

THÈSE

présentée à

l'Université Louis Pasteur de Strasbourg — École Doctorale MSII
Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection
UMR 7005 CNRS - ULP

par

M. Olivier Génévaux

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ LOUIS PASTEUR
DISCIPLINE: INFORMATIQUE

SIMULATION DE LIQUIDES À L'AIDE DES ÉQUATIONS DE
NAVIER-STOKES, ET VISUALISATION, À DESTINATION DE
L'INFOGRAPHIE

soutenue publiquement le 27 novembre 2006 devant le jury composé de:

M. Jean-Michel Dischler, Directeur de Thèse,
Professeur à l'Université Louis Pasteur, Strasbourg

M^{me} Marie-Paule Cani, Rapporteur externe,
Professeur à l'Institut National Polytechnique de Grenoble

M. Mathias Paulin, Rapporteur externe,
Maître de conférence des universités, Habilité à diriger des recherches,
Université Paul Sabatier, Toulouse

M. Pascal Schreck, Rapporteur interne,
Professeur à l'Université Louis Pasteur, Strasbourg

M^{me} Stéphanie Salmon, Examineur,
Maître de conférence des universités, Université Louis Pasteur, Strasbourg

Remerciements

Mes remerciements vont à :

- Monsieur Jean-Michel Dischler, qui a été le directeur de cette thèse, et qui s'est assuré de son bon déroulement aussi longtemps que nécessaire.
- Madame Marie-Paule Cani, Monsieur Mathias Paulin, et Monsieur Pascal Schreck, pour avoir accepté de juger ce travail.
- Madame Stéphanie Salmon, qui a bien voulu participer au jury de cette thèse, après m'avoir initié, avec patience et bienveillance, aux méthodes de résolution par éléments finis, et plus généralement, à une vision mathématique de la simulation de fluides.
- Monsieur Arash Habibi, dont les idées m'ont guidé dans les premiers travaux liés à la simulation de fluides, et qui m'a assisté de sa présence encourageante au cours des premiers mois de cette thèse.
- Monsieur Frédéric Larue, qui est à l'origine de l'application des harmoniques sphériques au problème de la visualisation de la réfraction.
- mes collègues et amis, pour leurs encouragements et leur bonne humeur.
- ma famille, pour son soutien durant cette thèse et depuis le début de mes études.

Table des matières

Table des figures	ix
Introduction	xv
1 Éléments de physique	1
1.1 Aperçu de la mécanique des milieux continus	1
1.1.1 Descriptions Eulérienne et Lagrangienne	2
1.1.2 Notions de mécanique et résultats nécessaires	4
Théorème de transport de Reynolds	4
Contraintes	5
Mesure de la vitesse de déformation	5
1.1.3 Lois de conservation	6
1.2 Équations de comportement du fluide	7
1.2.1 Loi de comportement	8
1.2.2 Équations de Navier-Stokes	9
1.2.3 Modèles simplifiés	12
1.2.4 Formulation alternative en vortacité	13
2 Simulation de fluides en infographie	15
2.1 Méthodes manuelles	16
2.1.1 Création des mouvements	16
2.1.2 Réaction d'objets à la cinétique du fluide	21
2.2 Méthodes Lagrangiennes	25
2.2.1 Particules sans interactions	25
2.2.2 Particules en interaction	28
2.2.3 Méthode SPH	35
Base de simulation	35
Extensions	38
2.2.4 Approche vortacité	40
2.3 Méthodes Eulériennes	42
2.3.1 Modèles simplifiés ou réduits	43
Simplification du fluide	43
Simplification de la configuration de simulation	45
Réduction dimensionnelle et extrapolation	50
2.3.2 Simulation physique non simplifiée	53
Simulation 3D du fluide en formulation vitesse–pression	53
Gestion de la turbulence et vortacité	58
Simulation du fluide par la vortacité	60

	Traitement de la surface pour le cas des liquides	63
	Domaine de simulation : discrétisation et modifications	68
2.3.3	Extensions	73
	Extension de la nature du fluide	74
	Bulles, gouttes et simulations diphasiques	76
2.3.4	Interaction avec des objets externes	81
2.3.5	Contrôlabilité	85
	Spécification des variables de simulation	85
	Points-clés	87
2.3.6	Accélération matérielle	93
2.4	Modèles Eulériens discrets	95
2.4.1	Automates cellulaires étendus	96
2.4.2	Modèles de Boltzmann sur réseau	98
2.5	Problématiques périphériques à la simulation	101
2.5.1	Emploi subordonné de simulations	102
2.5.2	Spécificités du rendu de fluides	103
	Ajout de détails	105
	Problèmes liés aux surfaces implicites	105
2.6	Bilan	107
3	Simulation MAC et interaction avec des solides déformables	111
3.1	Simulateur de fluide	112
3.1.1	Principe de la méthode MAC	112
3.1.2	Grille de discrétisation	113
3.1.3	Matérialisation du liquide	114
3.1.4	Conditions aux limites considérées	114
3.1.5	Résolution des équations de Navier-Stokes	116
	Algorithme général	116
	Détail des différentes étapes	119
3.1.6	Mise en œuvre du simulateur	122
	Gestion des conditions aux limites	123
	Gestion de la surface libre	126
	Trajectoires : marqueurs et caractéristiques	128
	Traitement de l'incompressibilité	131
3.1.7	Algorithme final	132
3.1.8	Résultats	132
3.2	Interaction avec des solides déformables	138
3.2.1	Modélisation masses-ressorts des solides	139
3.2.2	Interaction	140
3.2.3	Résultats	148
3.3	Bilan	153
4	Simulation MAC : Visualisation	157
4.1	Rendu direct de nuage de points par raytracing	158
4.1.1	Situation	158
	Modèle associé aux nuages de points	158
	Inadéquation des approches existantes	159
	Problème soulevé	160
4.1.2	Algorithme proposé	160
	Principe	160

	Recherche d'intersections	161
	Contrainte induite sur le raytracer-hôte	163
4.1.3	Surface	163
	Modèle de surface	163
	Normale	164
	Position de la surface	165
4.1.4	Construction des objets intermédiaires	166
	Construction de l'objet dual	166
	Normales	167
	Optimisation des nuages de points	168
4.1.5	Résultats	170
	Influence de la taille des primitives	172
4.1.6	Bilan	172
4.2	Visualisation rapide assistée par le matériel	175
4.2.1	Méthode de rendu	175
	Approche	176
	Données du rendu	177
	Algorithme de rendu	177
	Améliorations	179
4.2.2	Résultats	180
4.2.3	Bilan	183
5	Simulation par éléments finis	185
5.1	Principe de la simulation par éléments finis	186
5.2	Discrétisation en temps	187
	5.2.1 Méthode des caractéristiques	188
	5.2.2 Point fixe et linéarisation du terme advectif	188
5.3	Formulation variationnelle du problème de Stokes	189
	5.3.1 Problème fort	189
	5.3.2 Première forme variationnelle	191
	5.3.3 Intégration des conditions aux limites	191
	5.3.4 Forme variationnelle finale	192
5.4	Discrétisation en espace	194
	5.4.1 Problème discret	195
	5.4.2 Représentation du domaine	196
	5.4.3 Construction des espaces d'approximation	197
	Définition d'un élément fini	197
	Maillage et éléments	199
	Élément de référence	201
	5.4.4 Choix des espaces pour le problème de Stokes	202
	Interpolation de Lagrange dans un tétraèdre	203
	Élément de Taylor-Hood	204
	Élément mini	205
	5.4.5 Principe du calcul de la discrétisation	206
	Principe de construction des matrices	206
	Calcul sur un élément de référence	208
	Mise en œuvre de l'intégration	211
	5.4.6 Discrétisation des équations de Stokes	212
	Cas découplé	212
	Cas couplé	216

5.5	Résolution des systèmes linéaires	218
5.5.1	Introduction des conditions aux limites	218
5.5.2	Résolution du système mixte	219
5.5.3	Méthode d'Uzawa	220
5.6	Validation et premiers résultats	221
5.7	Bilan	235
6	Visualisation interactive de la réfraction	239
6.1	Rappels d'optique	240
6.1.1	Géométrie des rayons	241
6.1.2	Énergie portée par les rayons	241
6.1.3	Absorption	242
6.2	Travaux existants	242
6.2.1	Approches apparentées au lancé de rayons	243
6.2.2	Approches par projection de primitives	243
6.2.3	Approches hybrides	245
6.2.4	Approche avec précalcul	246
6.2.5	Acquisition d'une distorsion réelle	246
6.2.6	Bilan	246
6.3	Principe de la méthode	247
6.3.1	Modèle de réfraction adopté	247
6.3.2	Principe du rendu	249
	Gestion des données de distorsion	249
	Algorithme général	250
	Mise en œuvre de l'échantillonnage durant le précalcul	252
	Caractéristiques de l'approche	252
6.4	Harmoniques sphériques et fonctions directionnelles	253
6.4.1	Définition et propriétés des harmoniques sphériques	254
6.4.2	Compression	257
6.4.3	Décompression assistée par le matériel	261
6.4.4	Lissage	264
	Lissage fréquentiel	264
	Lissage spatial	266
6.5	Algorithme de rendu	266
6.5.1	Approche directe	268
6.5.2	Approche avec garantie de continuité	270
6.5.3	Rendu final	275
6.6	Résultats	276
6.7	Bilan	287
6.8	Extension aux <i>surface light fields</i>	289
6.8.1	Calcul des données de radiance	290
6.8.2	Visualisation des données	291
6.8.3	Résultats	291
6.8.4	Bilan	292
	Conclusion	295

A Notations et rappels mathématiques	301
A.1 Définition des opérateurs différentiels	301
A.2 Interprétation des opérateurs	303
A.3 Dérivée directionnelle	303
A.4 Identités et propriétés utiles	304
Bibliographie	309

Table des figures

1.1	Étude des contraintes internes à un matériau	5
1.2	Navier et Stokes	11
2.1	Primitives de flux	17
2.2	Advection des détails effectuée lors du rendu	20
2.3	Construction de flux par pavage	21
2.4	Masquage dynamique de champs statiques	23
2.5	Famille de potentiels de type Lennard-Jones	31
2.6	Grille de gestion de la convection pour la méthode MPS-MAFL	38
2.7	Limite de représentation des champs de hauteur	45
2.8	Division verticale des colonnes d'un champ de hauteur	47
2.9	Couplage Navier-Stokes 2D et propagation d'ondes	51
2.10	Interpolation de simulations	53
2.11	Principe de la méthode MAC	54
2.12	Évolution de la surface par <i>levelset</i>	64
2.13	Méthodes de correction des <i>levelsets</i>	65
2.14	Améliorations de l'advection	66
2.15	Évolution implicite de surface triangulée	67
2.16	Types de cellules d'une grille décalée hétérogène	71
2.17	Méthode semi-Lagrangienne pour un maillage mobile	72
2.18	Extrapolation à l'interface de deux fluides	79
2.19	Interaction fluide – solide par méthode <i>Rigid Fluid</i>	83
2.20	Contrôle de fumée pour atteindre une densité cible	90
2.21	Réseau D3Q19 pour méthodes de type Boltzmann sur réseau	99
2.22	Chronologie des principales méthodes	109
3.1	Grille décalée	113
3.2	Conditions aux limites pour une grille décalée	115
3.3	Résolution des équations de Navier-Stokes pour la méthode MAC	119
3.4	Méthode des caractéristiques	120
3.5	Masque de calcul du Laplacien pour la viscosité	121
3.6	Ambiguïté de la discrétisation à proximité des obstacles	123
3.7	Traitement des ambiguïtés de la grille	125
3.8	Statistiques sur la grille de simulation	126
3.9	Traitement de la vitesse de la surface	127
3.10	Friction parasite de l'air sur le liquide	129
3.11	Problème de l'interpolation linéaire à proximité d'un obstacle	130
3.12	Comparaison du traitement des caractéristiques	131

3.13	Algorithme de simulation MAC implémenté.	133
3.14	Exemple : boîte à fond percé	134
3.15	Exemple : jet et obstacles	134
3.16	Exemple : variation de la viscosité	135
3.17	Conservation du volume de liquide	137
3.18	Temps d'exécution du simulateur MAC	138
3.19	Carré modélisé par des masses et des ressorts	139
3.20	Répulsion des marqueurs par l'interaction	142
3.21	Interaction entre liquide et solides déformables	143
3.22	Interaction représentée par un champ de force non-uniforme	145
3.23	Convergence de la force d'interaction pour les masses	146
3.24	Reconstruction alternative du champ de force Eulérien	148
3.25	Exemple d'interaction : flottaison	149
3.26	Exemple d'interaction : jet	150
3.27	Exemple d'interaction : cubes	151
3.28	Exemple d'interaction : ricochet	151
3.29	Paramètres significatifs des exemples	152
3.30	Temps requis par l'interaction	152
3.31	Exemples de forces d'interaction	153
4.1	Dualité	161
4.2	Algorithme de recherche d'intersections	161
4.3	Partitionnement pour la recherche d'intersections	162
4.4	Double recherche de l'intersection de sortie	162
4.5	Dépendance du calcul d'intersection au point de vue	163
4.6	Ambiguïté du volume de l'objet	164
4.7	Discontinuité de la normale géométrique	165
4.8	Interpolation des normales	165
4.9	Interpolation de la position de la surface	166
4.10	Génération du dual	166
4.11	Calcul de la normale	168
4.12	Exemple de résultat du prétraitement	169
4.13	Extraction de la surface du nuage primal	169
4.14	Génération du dual mince	169
4.15	Étapes de prétraitement	170
4.16	Exemples d'animations	171
4.17	Temps de calcul	172
4.18	Influence de la taille des primitives	173
4.19	Illustration des composantes du rendu rapide	176
4.20	Illustration du rendu volumique par coupe	176
4.21	Définition des régions de rendu	177
4.22	Atténuation de l'illumination locale	179
4.23	Exemples de visualisation volumique	180
4.24	Comparaison du rendu volumique au rendu surfacique	181
4.25	Temps de calcul du rendu volumique	182
4.26	Vitesse de rendu	183
5.1	Exemples de maillages	196
5.2	Exemples d'éléments de type Lagrange	198
5.3	Fonctions de base globales	200

5.4	Correspondance locale-globale des degrés de liberté	201
5.5	Élément de référence	202
5.6	Tétraèdres de référence	203
5.7	Éléments pour le problème de Stokes	204
5.8	Algorithme de construction des matrices	207
5.9	Motifs de remplissage des matrices pour le problème de Stokes	207
5.10	Format de stockage CCS	208
5.11	Données nécessaires au calcul des matrices de l'élément mini	215
5.12	Méthode d'Uzawa	222
5.13	Problème-test	224
5.14	Tests de convergence (1)	225
5.15	Tests de convergence (2)	226
5.16	Tests de convergence (3)	227
5.17	Résultats imprécis pour le tétraèdre mini	228
5.18	Exemple de domaine non orthotrope	229
5.19	Problème test de la cavité entraînée	230
5.20	Problème test de la cavité entraînée — Vitesse	231
5.21	Problème test de la cavité entraînée — Pression	232
5.22	Comparaisons pour le problème de la cavité entraînée	233
5.23	Évolution de la consommation mémoire	234
6.1	Intersection d'un rayon de lumière avec un dioptre	240
6.2	Propagation d'un rayon de lumière dans un objet	247
6.3	Algorithme global	250
6.4	Modèle de la situation simulée	251
6.5	Échantillonnage de la fonction de distorsion	252
6.6	Système de coordonnées sphériques	254
6.7	Représentation des premières fonctions harmoniques sphériques	255
6.8	Influence de l'échantillonnage directionnel	259
6.9	Données compressées	260
6.10	Influence de la quantification des coefficients d'harmoniques	260
6.11	Structure des <i>shaders</i> de décompression	262
6.12	Nombre de passes requises par la décompression	263
6.13	Supports des calculs	264
6.14	Effets du lissage	264
6.15	Phénomène de Gibbs	265
6.16	Architecture du rendu	267
6.17	Comparaison de la continuité de l'interpolation	267
6.18	Discontinuités acceptables dans la réfraction	268
6.19	Étapes du rendu pour l'approche directe	269
6.20	Stratégie de remplissage des gouttières	270
6.21	Échantillonnage uniforme dérivé de la triangulation	271
6.22	Interpolation dans la texture	272
6.23	Zones d'interpolation	274
6.24	Stockage des triangles	274
6.25	Étapes du rendu pour l'approche avec garantie de continuité	274
6.26	Algorithme de rendu final	276
6.27	Résultats	277
6.28	Matériaux multiples	278
6.29	Atténuation variable	279

6.30	Émulation d'un changement de matériaux	279
6.31	Comparaison avec d'autres méthodes	280
6.32	Variation de la compression	282
6.33	Capture progressive	283
6.34	Vitesse de rendu	285
6.35	Utilisation mémoire	286
6.36	Textures directionnelles	290
6.37	Phénomène de Gibbs	292
6.38	Exemples de résultats	293
6.39	Variation de la compression	294
A.1	Calcul de la dérivée directionnelle	304

Introduction

Avec la généralisation du recours à l'infographie dans l'industrie des loisirs, la demande concernant la production de scènes d'un réalisme croissant a fortement augmenté durant les deux dernières décennies. Par conséquent, si des modèles géométriques simples présentant une apparence de matière plastique ont longtemps été perçus comme crédibles, les critères actuels du réalisme simulé ont évolué en termes d'exigences et requièrent une intégration convaincante de phénomènes naturels communément répandus.

L'intégration de tels phénomènes naturels, familiers au spectateur, oblige ainsi à recourir à des méthodes capables de fournir des résultats d'un réalisme quasi-parfait. C'est pourquoi les méthodes empiriques fortement manuelles ne suffisent plus à donner satisfaction à grande échelle ou dans des situations complexes.

Un tel constat s'applique tout particulièrement aux liquides. Visibles au quotidien — aussi bien sous la forme d'une rivière que d'une tasse de café — leurs mouvements caractéristiques sont toujours imprévisibles, sans pour autant être aléatoires. Cette thèse a donc pour objet l'étude et la mise au point de techniques permettant d'inclure des liquides dans des scènes réalisées en images de synthèse.

Cet objectif de réalisme rencontre deux problématiques complémentaires : d'une part la simulation des mouvements du fluide, d'autre part le rendu de ce même fluide. Bien que très différents par nature, ces aspects ont tous deux fait l'objet des travaux de cette thèse.

En préalable au développement de ces travaux, un exposé de la physique sous-jacente à la simulation des fluides est proposé dans le chapitre 1, détaillant la dérivation des équations de Navier-Stokes, communément considérées comme un modèle réaliste des fluides. C'est ce modèle que nous adopterons comme modèle de simulation des fluides.

Bien que le plus abouti et d'utilisation prédominante depuis une dizaine d'années dans le contexte de la simulation de fluides à destination de l'infographie, ce modèle de simulation n'est pas le seul.

Le chapitre 2 offrira un tour d'horizon des nombreuses méthodes qui ont été proposées pour produire des animations de synthèse de fluides, des plus simples, fortement empiriques et manuelles, à celles issues en droite ligne de la physique et des mathématiques appliquées. La tendance actuelle à recourir à ces techniques de simulation sophistiquées s'y reflète, tout comme la prédominance du cadre de simulation Eulérien, cadre dans lequel les travaux de cette thèse s'inscrivent.

Concernant la simulation proprement dite du fluide, notre intérêt s'est tout d'abord porté sur le problème que constitue l'interaction entre liquide à surface

libre et solides, placés au sein d'un environnement de simulation commun. La réalisation automatique d'une telle interaction présente un intérêt de premier ordre pour l'infographie, dans le but d'obtenir des scènes vivantes et visuellement convaincantes.

Le mode d'interaction que nous proposons à cet effet est décrit dans le chapitre 3. Il autorise le couplage entre solides déformables représentés par des objets masses-ressorts, Lagrangiens, et des liquides Eulériens simulés par la méthode *Markers And Cells* par le biais du calcul d'une force d'interaction réintroduite dans les deux simulations. L'interaction développée, si elle est empirique sur le plan physique, offre néanmoins satisfaction sur un plan purement visuel. Ce même chapitre décrit également en détails le simulateur chargé de la gestion des liquides, avant la présentation de l'interaction elle-même.

Le type de simulateur employé, *Markers And Cells* s'appuie avantageusement sur une représentation du liquide sous forme de nuage volumique de points pour autoriser des simulations de liquides à surfaces libres non contraintes dans leurs mouvements. Cette représentation déstructurée, si elle se révèle une force lors de la simulation des mouvements, se transforme pourtant en faiblesse dans le cadre de la visualisation des résultats, la représentation d'un objet par un nuage volumique de points restant floue et peu courante en infographie. Le chapitre 4 présente nos travaux de visualisation directe d'un liquide représenté sous cette forme, sans reconstruction de surface explicite. Deux situations sont étudiées, l'une dans le cadre d'une visualisation photo-réaliste, l'autre dans le cadre d'une prévisualisation réaliste et rapide des animations.

Dans l'objectif de proposer une interaction du fluide avec des solides, qui soit de meilleure qualité grâce à son ancrage sur une base physique robuste, un problème couplé de simulation des deux phases doit être posé, ce qui nécessite de pouvoir simuler le fluide dans des domaines de formes arbitraires. Or, le simulateur auquel nous avons eu initialement recours et qui s'appuie sur une grille Cartésienne n'offre pas cette fonctionnalité. Notre choix s'est alors porté vers un mode de résolution par éléments finis du fluide. Ce mode de simulation, à notre connaissance non encore employé pour simuler des fluides en infographie, permet en effet la gestion de domaines de simulation arbitraires et de conditions aux limites plus adaptées. La description de la méthode des éléments finis, ainsi que son application aux équations de Navier-Stokes, est exposée dans le chapitre 5. Néanmoins, devant la complexité de la méthode, seuls des premiers résultats simples, essentiellement de validation de la méthode, ont pu être obtenus. Ces premiers résultats quoiqu'encourageants, ne présentent pas encore un intérêt visuel direct dans un cadre d'infographie.

Parallèlement à ces travaux de simulation, nous nous sommes intéressés au problème de la visualisation interactive du phénomène visuel prépondérant pour les objets transparents comme les liquides, c'est-à-dire la réfraction. Nous avons à cet effet développé une méthode capable de simuler de matière réaliste la réfraction, pour des objets de forme quelconque représentés par un maillage, méthode apte à tenir compte de plusieurs déviations de la lumière. Cette approche, décrite dans le chapitre 6, s'appuie sur un précalcul échantillonnant la déviation de la lumière due à la présence de l'objet réfringent, puis sur une exploitation de ces données au cours du rendu. La faisabilité de la méthode tient à la représentation compressée de cette information directionnelle, et à sa décompression en temps réel à l'aide du matériel graphique programmable. En raison de sa dépendance à un précalcul, et à un volume de données générées, qui reste

important même compressé, cette méthode est cependant restreinte à des objets statiques, et n'est donc pas immédiatement applicable au rendu d'animations de liquides.

Chapitre 1

Éléments de physique

Avant de s'intéresser à la simulation numérique d'un phénomène, il convient de se préoccuper des lois qui le régissent. Ce chapitre a pour vocation d'introduire les équations qui seront à la base de la simulation ultérieure des fluides, les équations de Navier-Stokes. L'objectif n'est en aucun cas de commencer à traiter le problème de la résolution de ces équations.

Il convient également de préciser la portée de la modélisation effectuée : il s'agit de la dérivation des équations minimales représentant un fluide mathématiquement et physiquement idéalisé, sans aucune composante de modélisation physique supplémentaire, et qui sont communément admises comme un modèle satisfaisant des fluides. Le propos n'est pas de modéliser des phénomènes observables à une échelle macroscopique, comme une turbulence statistique.

1.1 Aperçu de la mécanique des milieux continus

Toute la construction de la mécanique des milieux continus qui va suivre, repose sur la supposition qu'un système physique est modélisable mathématiquement par un *continuum*, c'est-à-dire que les diverses quantités nécessaires à sa représentation peuvent être manipulées comme des fonctions à valeurs scalaires, vectorielles ou tensorielles différentiables presque partout — Dans le cas qui nous intéresse, on ne considérera pas de discontinuités, donc elles peuvent être considérées comme étant différentiables partout — Comme pour tout modèle, des limites existent : il est ainsi parfaitement clair que les hypothèses de continuité ne sont plus valides en deçà d'une certaine échelle lorsque les quantités macroscopiques manipulées, telles la vitesse ou la pression, perdent leur signification, en raison de la nature discrète de la matière. Ceci étant, si cette problématique de pertinence du modèle continu existe réellement en ingénierie, ces limites de validité ne seront jamais transgressées, ni même jamais approchées, dans le cadre de la synthèse d'images, où seules des situations à l'échelle humaine sont généralement envisagées.

On considère donc des milieux continus, définis dans l'espace Euclidien $E = \mathbb{R}^3$ muni d'un système de coordonnées Cartésien $(O, x, y, z) = (O, x_1, x_2, x_3)$. On suppose connues les définitions des opérateurs différentiels usuels. Dans le cas contraire, on invite le lecteur à se reporter à l'annexe A qui expose ces définitions et présente quelques identités utiles pour les développements ultérieurs.

1.1.1 Descriptions Eulérienne et Lagrangienne

Pour présenter les équations gouvernant la mécanique des milieux continus, il convient d'abord d'introduire les modes de représentations possibles du système modélisé. Les deux représentations, Lagrangienne et Eulérienne, correspondent respectivement à des mesures liées au milieu en mouvement et à des mesures fixes dans l'espace.

Si la représentation Lagrangienne est plus simple à appréhender en première intention, on peut d'ores et déjà postuler que la représentation Eulérienne est préférée dans le cadre des fluides, puisque l'on n'est généralement pas capable de suivre individuellement les différentes particules composant le fluide au cours du temps, lors de ses mouvements complexes. De plus, la notion de mouvement stationnaire est aisée à appréhender à l'aide de la description Eulérienne.

Définition des descriptions On considère un domaine matériel Ω_L composé d'un *continuum* matériel en mouvement, et on repère la configuration Ω_0 initiale de ce volume. Cette configuration permet d'identifier toutes les particules de matière $\eta \in \Omega_0$ et on peut, par suite, considérer leurs positions au temps t au travers de la fonction

$$\Phi(\eta, t) \quad (\eta, t) \in \Omega_0 \times [0, +\infty[$$

On peut noter que la transformation $\Phi_t(x) = \Phi(x, t)$ est bijective et différentiable, que son inverse l'est également, et qu'elle préserve l'orientation de l'espace en raison de contraintes physiques.

Cette représentation est la représentation Lagrangienne, où sont données toutes les trajectoires de toutes les particules composant le milieu étudié, celles-ci étant repérées par rapport à une configuration de référence Ω_0 . Les points d'observation du *continuum* sont les particules qui le composent, et par conséquent le système de coordonnées est mobile, lié au domaine en mouvement.

À l'inverse, dans la représentation Eulérienne, on considère un domaine géométrique fixe Ω_E au travers duquel évolue un milieu matériel continu en mouvement, ce qui définit des points d'observation du milieu mobile fixes au cours du temps $x \in \Omega_E$. La représentation Eulérienne du mouvement consiste alors en la donnée des vitesses du milieu mobile en ces points d'observation, à tout moment, c'est-à-dire à la donnée de la fonction

$$u(x, t) \quad (x, t) \in \Omega_E \times [0, +\infty[$$

décrivant la vitesse du milieu au point x à l'instant t .

Équivalence des descriptions Les deux descriptions précédentes sont parfaitement équivalentes. Supposons que l'on connaisse la description de Lagrange du mouvement, c'est-à-dire que Φ soit connue. La vitesse d'une particule η au temps t est donc définie comme

$$\hat{u}(\eta, t) = \frac{\partial \Phi}{\partial t}(\eta, t)$$

Par conséquent,

$$u(\Phi(\eta, t), t) = \hat{u}(\eta, t) = \frac{\partial \Phi(\eta, t)}{\partial t}$$

Et comme Φ est inversible

$$x = \Phi(\eta, t) \Leftrightarrow \Phi^{-1}(x, t) = \eta$$

Ce qui prouve que la description Eulérienne est déterminée par la description Lagrangienne :

$$u(x, t) = \frac{\partial \Phi(\Phi^{-1}(x, t), t)}{\partial t}$$

Inversement, si on suppose donnée la description Eulérienne, c'est-à-dire la fonction u connue, la description Lagrangienne, pour la particule η , *i.e.* sa trajectoire, en est déductible par l'équation différentielle suivante :

$$\frac{\partial \Phi(\eta, t)}{\partial t} = u(\Phi(\eta, t), t) \quad \Phi(x, 0) = \eta$$

Autres quantités La description faite jusqu'ici a porté sur les vitesses du milieu observé. On peut également étendre ces visions Eulérienne et Lagrangienne à d'autres quantités définies dans le milieu. Ainsi, il est possible de considérer

$$\hat{q}(\eta, t) \quad (\eta, t) \in \Omega_0 \times [0, +\infty[$$

n'importe quelle quantité \hat{q} au temps t portée par la particule repérée par η dans la configuration initiale Ω_0 , et simultanément

$$q(x, t) \quad (x, t) \in \Omega_E \times [0, +\infty[$$

la quantité portée par la particule présente à la position x au temps t . Ces deux quantités sont bien évidemment liées par les mouvements du milieu :

$$\hat{q}(\eta, t) = q(\Phi(\eta, t), t)$$

Dérivée convective On va maintenant définir la notion de dérivée convective, qui permet de lier les variations d'une quantité portée par les particules constitutives du champ dans ces deux représentations.

En étudiant les variations de \hat{q}_η :

$$\begin{aligned} \frac{d\hat{q}_\eta}{dt} &= \frac{\partial q}{\partial x} \cdot \frac{\partial \Phi_\eta^x}{\partial x} + \frac{\partial q}{\partial y} \cdot \frac{\partial \Phi_\eta^y}{\partial y} + \frac{\partial q}{\partial z} \cdot \frac{\partial \Phi_\eta^z}{\partial z} + \frac{\partial q}{\partial t} \\ &= \frac{\partial q}{\partial x} \cdot u_x + \frac{\partial q}{\partial y} \cdot u_y + \frac{\partial q}{\partial z} \cdot u_z + \frac{\partial q}{\partial t} \\ &= u \cdot \nabla q + \frac{\partial q}{\partial t} \end{aligned}$$

Cette dernière relation lie donc les visions Eulérienne et Lagrangienne des variations d'une même quantité, pour un milieu plongé dans un champ de vitesse u connu. L'opérateur de dérivée convective $\frac{D}{Dt}$, applicable à un champ exprimé en variables Eulériennes, peut alors être défini comme

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \cdot \nabla$$

Il permet d'exprimer les variations du champ Eulérien en fonction des variations subies par les particules constitutives du milieu mobile, représentées de manière

Lagrangienne. On reconnaîtra que le deuxième terme de l'opérateur n'est autre que la dérivée directionnelle dans le sens du courant. Une interprétation plus intuitive de la définition peut également être donnée en modifiant légèrement la relation :

$$\frac{\partial q}{\partial t} = -u \cdot \nabla q + \frac{d\hat{q}_\eta}{dt}$$

En un point fixe de l'espace, la variation $\frac{\partial q}{\partial t}$ d'une quantité au cours du temps est égale à la variation de la quantité portée par la particule présente en ce point, $\frac{d\hat{q}_\eta}{dt}$, plus la variation due au déplacement des particules, c'est-à-dire le déplacement du champ suivant sa vitesse locale $-u \cdot \nabla q$.

Il est à noter que la dérivée convective $\frac{D}{Dt}$ est également connue sous le nom de dérivée particulaire, matérielle, substantielle, ou encore dérivée totale. Il faut également remarquer que cette relation est valable pour n'importe quelle quantité afférente aux particules du milieu observé, ce qui inclut en particulier la vitesse u . Le transport d'une quantité par un champ vectoriel externe, c'est-à-dire le problème

$$\frac{\partial q}{\partial t} + u \cdot \nabla q = 0$$

est connu sous le nom d'advection ou de convection.

1.1.2 Notions de mécanique et résultats nécessaires

Théorème de transport de Reynolds

Pour établir les équations de conservation à la base de la mécanique des milieux continus, il est nécessaire de faire usage du théorème de transport de Reynolds, qui permet d'exprimer les variations d'une quantité sur un domaine mobile au cours du temps. On rappelle que $u : \Omega \times [0, +\infty[\rightarrow \mathbb{R}^3$ représente le champ de vitesse Eulérien.

Soit $V \subset \Omega_0$ un sous-domaine de l'espace initial et $V(t)$ sa forme à l'instant t :

$$V(t) = \{\Phi(\eta, t) \mid \eta \in V\}$$

Alors, pour tout champ différentiable $f : \Omega \times [0, +\infty[\rightarrow \mathbb{R}$, il est possible d'exprimer la variation de cette quantité dans le domaine :

$$\frac{d}{dt} \int_{V(t)} f(x, t) dx = \int_{V(t)} \frac{\partial f}{\partial t}(x, t) + \nabla \cdot (fu)(x, t) dx$$

Une autre expression de cette relation existe, après usage du théorème de Gauss, qui facilite son interprétation :

$$\frac{d}{dt} \int_{V(t)} f(x, t) dx = \int_{V(t)} \frac{\partial f}{\partial t}(x, t) dx + \int_{\partial V(t)} fu \cdot nds$$

où $\partial V(t)$ représente la surface fermée bordant le volume $V(t)$ munie d'un élément de surface ds de normale unitaire extérieure n . Il devient alors plus clair que la variation globale de la quantité f sur V est égale à la variation de f dans le domaine lui-même, additionnée au flux traversant la surface.

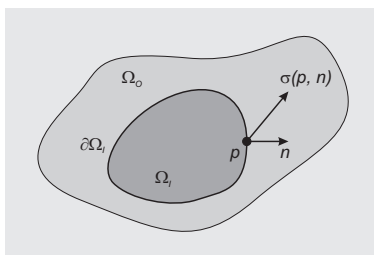


Figure 1.1: Étude des contraintes internes à un matériau.

Contraintes

On s'intéresse maintenant à appréhender les efforts internes au matériau présent dans un domaine Ω . Pour les considérer en un point p , on rend p extérieur comme suit : on définit un sous domaine arbitraire $\Omega_I \subset \Omega$ dont la frontière $\partial\Omega_I$ passe par p en présentant une normale unitaire extérieure n , comme illustré dans la figure 1.1.

On étudie alors les actions de contact de $\Omega_O = \Omega \setminus \Omega_I$ sur Ω_I , définies sur la surface $\partial\Omega_I$: elles forment un champ de forces surfaciques, appelé champ des vecteurs contraintes. Comme ces actions modélisent des forces de contact, purement locales, elles ne dépendent en fait pas de la forme générale de Ω_I , mais uniquement de n en chaque point p . En chaque point p , il existe donc *a priori* une infinité de vecteurs contraintes, dépendants de n . On peut toutefois montrer que tous ces vecteurs contraintes sont linéairement dépendants de n .

Ceci permet de définir l'existence, en chaque point, d'un tenseur des contraintes, appelé tenseur des contraintes de Cauchy, qui décrit localement l'état des contraintes internes au matériau. Il s'agit d'un tenseur du second ordre $\sigma = [\sigma]_{i,j=1,2,3}$. Le vecteur de contraintes dans une direction n est alors donné par $\sigma \cdot n$. Ce tenseur σ étant défini en tout point du domaine, il existe un champ de tenseur de rang deux décrivant l'état des contraintes internes du matériau. On peut noter que dans le cas de milieux où ne sont en action que des champs de forces, *i.e.* des torseurs de moments nuls, σ est symétrique, ce qui est le cas ici. Cette limite de modélisation n'est toutefois pas restrictive puisqu'elle n'exclut *a priori* que les actions à distance magnétiques, qui n'entrent pas dans le cadre de simples fluides.

Sur la frontière, les contraintes exercées par le milieu ambiant *i.e.* les forces extérieures de surface f_s , doivent être en équilibre avec les contraintes locales, c'est-à-dire que

$$\sigma \cdot n = f_s \quad \text{sur } \partial\Omega$$

Mesure de la vitesse de déformation

Pour pouvoir étudier la vitesse de déformation d'un matériau, il est nécessaire de pouvoir la mesurer. Pour cela, on introduit D , le tenseur des taux ou des vitesses de déformation, défini à partir du champ de vitesse Eulérien u .

Le tenseur ∇u qui mesure localement l'évolution de la vitesse peut être décomposé en sa partie symétrique D et sa partie antisymétrique W :

$$\nabla u = D + W$$

avec

$$D = \frac{1}{2}[\nabla u + \nabla^T u] \quad \text{et} \quad W = \frac{1}{2}[\nabla u - \nabla^T u]$$

Or on peut montrer que W est caractéristique d'un mouvement rigide, ce qui autorise à considérer D comme une mesure de la vitesse de déformation.

1.1.3 Lois de conservation

On va maintenant dériver les deux lois de conservation, de masse et de quantité de mouvement. À celles-ci peuvent être adjointes une ou plusieurs lois afférentes à la conservation de l'énergie du système et à sa dissipation, qui découlent des principes de la thermodynamique. Dans le cadre des travaux menés dans cette thèse, la variation d'énergie est jugée sans influence visible sur les résultats, et donc négligeable, ce qui permet d'ignorer ces lois supplémentaires. C'est généralement le cas en infographie, sauf lorsque sont simulés des phénomènes avec des variations énergétiques très intenses comme des feux ou des explosions.

Conservation de la masse La première loi de conservation est celle transcrivant le principe de conservation de la masse, qui postule que la matière ne disparaît ni n'apparaît arbitrairement. Ceci équivaut à poser que la variation de masse est nulle.

Donc, en considérant $\rho(x, t)$ la masse volumique :

$$\frac{d}{dt} \int_{V(t)} \rho(x, t) dx = 0$$

ce qui, en employant le théorème de transport, se transcrit en :

$$\int_{V(t)} \frac{\partial \rho}{\partial t}(x, t) + \nabla \cdot (\rho u)(x, t) dx = 0$$

Cette dernière relation étant vraie quel que soit le domaine $V(t)$, elle est vraie partout, ce qui amène à pouvoir formuler l'expression locale du principe de conservation de la masse :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \quad \text{sur } \Omega \times [0, +\infty[$$

Cette équation est également dénommée équation de continuité.

Conservation de la quantité de mouvement La seconde loi de conservation qui prend part à la modélisation des milieux continus est celle décrivant la conservation de la quantité de mouvement.

Le principe fondamental de la mécanique — la deuxième loi de *Newton* — pose que, dans un référentiel Galiléen, le torseur dynamique est égal à la somme des torseurs des actions extérieures au système. Deux relations en découlent donc, l'une sur les moments et l'autre sur les résultantes. La première permet d'établir la symétrie du tenseur de contraintes et ne sera pas détaillée ici. La deuxième exprime que les forces s'appliquant sur un domaine engendrent une variation de sa quantité de mouvement :

$$\frac{d}{dt} \int_{V(t)} \rho u(x, t) dx = \int_{V(t)} \rho f(x, t) dx + \int_{\partial V(t)} \sigma \cdot n ds$$

Les forces f s'exerçant sur un volume sont soit internes, soit de surface, et sont dans ce cas en équilibre avec les contraintes. Par application du théorème de Gauss, on peut réécrire cette relation uniquement sur le volume, en transformant le terme de surface :

$$\frac{d}{dt} \int_{V(t)} \rho u(x, t) dx = \int_{V(t)} \rho f(x, t) dx + \nabla \cdot \sigma(x, t) dx$$

En considérant le théorème de transport, on peut reformuler la variation de la quantité de mouvement :

$$\begin{aligned} & \frac{d}{dt} \int_{V(t)} \rho u(x, t) dx \\ &= \left[\frac{d}{dt} \int_{V(t)} \rho u_i(x, t) dx \right]_{i=1,2,3} \\ &= \left[\int_{V(t)} \frac{\partial \rho u_i}{\partial t}(x, t) + \nabla \cdot (\rho u_i u)(x, t) dx \right]_{i=1,2,3} \\ &= \int_{V(t)} \frac{\partial \rho u}{\partial t}(x, t) + \nabla \cdot (\rho u \otimes u)(x, t) dx \end{aligned}$$

en posant que $u \otimes v$ représente le tenseur de rang deux, obtenu par produit tensoriel de deux vecteurs : $u \otimes v = [u_i] \otimes [v_j] = [u_i \cdot v_j]_{i,j=1,2,3}$.

Finalement, on obtient la forme intégrale de la conservation de quantité de mouvement :

$$\int_{V(t)} \frac{\partial \rho u}{\partial t}(x, t) + \nabla \cdot (\rho u \otimes u)(x, t) dx = \int_{V(t)} \rho f(x, t) + \nabla \cdot \sigma(x, t) dx$$

Comme pour la conservation de la masse, on en déduit l'expression locale du principe de conservation de la quantité de mouvement :

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \otimes u) = \rho f + \nabla \cdot \sigma \quad \text{sur } \Omega \times [0, +\infty[$$

1.2 Équations de comportement du fluide

Les principes de la mécanique des milieux continus précédemment énoncés sont généraux : ils s'appliquent à tous les types de milieu, aussi bien solides que fluides. Cette section va maintenant caractériser ce qu'est un fluide, au travers de ses lois de comportement, pour aboutir aux équations de Navier-Stokes.

À ce stade, il est nécessaire de préciser que la loi de comportement qui sera présentée constitue une modélisation de la réalité, et décrit, par conséquent, un fluide hypothétique idéalisé. Dans la réalité, les fluides peuvent adopter des comportements beaucoup plus complexes, comme le font par exemple les fluides viscoélastiques ou pâteux. Néanmoins, on admet que la modélisation Newtonienne, modélisant correctement l'eau ou les gaz, est satisfaisante pour la majorité des fluides représentables en infographie. On peut également noter qu'il existe divers dérivés des équations de Navier-Stokes, selon la formulation adoptée, les composantes de modélisation ajoutées, ou les simplifications apportées, ce qui aboutit à de nombreux modèles différents.

1.2.1 Loi de comportement

Un fluide, ne comportant pas de configuration de référence, n'admet pas de mesure de déformation directe. En revanche, la vitesse de déformation est mesurable par le tenseur des taux de déformation D . La loi de comportement d'un fluide établit donc une relation entre les contraintes internes au fluide, représentées par le tenseur de contraintes σ , et D .

Fluide compressible Dans le cas d'un fluide Newtonien, la relation entre contraintes et déformations est affine, et le fluide doit être isotrope, ce qui implique une loi de comportement de la forme :

$$\sigma = 2\mu D + (\lambda \operatorname{Tr} D - p)I$$

où I représente le tenseur identité de rang deux, Tr la trace, et p la pression absolue, normalement dépendante de l'état thermodynamique local, *i.e.* des masse volumique et température locales. Les deux coefficients μ et λ sont les premier et deuxième coefficients de viscosité caractéristiques du fluide. μ est la viscosité dynamique, mesurée en Poiseuille, et de dimension physique $kg \cdot m^{-1} \cdot s^{-1}$.

Il est possible de recourir à l'hypothèse dite de Stokes, qui consiste à supposer que la puissance mécanique intérieure des forces de viscosité est uniquement due aux cisaillements, et non aux variations de volume. Cette hypothèse se traduit alors par le fait que λ et μ sont liés :

$$\lambda + \frac{2\mu}{3} = 0$$

ce qui induit une loi de conservation avec un unique paramètre μ :

$$\sigma = 2\mu D - \left(\frac{2}{3}\mu \operatorname{Tr} D - p \right) I$$

Dans le cas de fluides non-Newtoniens, la relation entre σ et D est gouvernée par un ensemble d'équations différentielles, rendant le problème encore beaucoup plus complexe.

Fluide incompressible On porte maintenant attention aux fluides Newtoniens incompressibles, qui nous intéressent particulièrement parce qu'ils modélisent bien les liquides. Dans ce cas, tout sous-domaine du fluide conserve son volume au cours du temps :

$$\int_{V(t)} dx = \int_{V(0)} dx \quad \forall t \in [0, +\infty[$$

ce qui implique que les variations de volume du fluide sont nulles au cours du temps :

$$\frac{d}{dt} \int_{V(t)} dx = 0 \quad \forall t \in [0, +\infty[$$

Cette relation peut alors être réexprimée grâce au théorème de transport :

$$\frac{d}{dt} \int_{V(t)} dx = \int_{V(t)} \nabla \cdot u dx = 0 \quad \forall t \in [0, +\infty[$$

Cette propriété étant vraie pour tout sous-domaine du fluide, on peut exprimer le principe d'incompressibilité localement par :

$$\nabla \cdot u = 0$$

En rappelant que $\nabla \cdot u = \text{Tr } D$, la loi de comportement des fluides incompressibles s'écrit donc :

$$\sigma = 2\mu D - pI$$

Il est utile de préciser que l'incompressibilité ne se rapporte pas à la pression, mais la non-compressibilité du fluide, *i.e.* l'absence de variation de son volume.

En faisant appel à la propriété de conservation de la masse, on peut également déduire que dans ce cas, si le fluide est initialement homogène, la masse volumique du fluide est constante $\rho(x, t) = \rho$. En effet :

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) &= 0 \\ \frac{\partial \rho}{\partial t} + u \nabla \rho + \rho \nabla \cdot u &= 0 \\ \frac{D\rho}{Dt} + \rho \nabla \cdot u &= 0 \\ \frac{D\rho}{Dt} &= 0 \end{aligned}$$

ce qui signifie que la masse volumique des particules de fluide est conservée au cours du temps.

1.2.2 Équations de Navier-Stokes

On supposera que le milieu est homogène, c'est-à-dire que la même loi de comportement s'applique à l'intégralité du domaine Ω .

Fluide compressible En insérant la loi de comportement d'un fluide compressible dans les équations générales de la mécanique des milieux continus, on obtient la relation suivante :

$$\begin{aligned} \frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \otimes u) &= \rho f + \nabla \cdot [2\mu D + (\lambda \text{Tr } D - p)I] \\ &= \rho f + \mu[\nabla \cdot \nabla u + \nabla \cdot \nabla^T u] + \nabla \cdot [(\lambda \nabla \cdot u)I] - \nabla \cdot [pI] \\ &= \rho f + \mu \Delta u + \mu \nabla \nabla \cdot u + \lambda \nabla \nabla \cdot u - \nabla p \\ &= \rho f + \mu \Delta u + (\lambda + \mu) \nabla \nabla \cdot u - \nabla p \end{aligned}$$

Or,

$$\begin{aligned} \frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \otimes u) &= \frac{\partial \rho}{\partial t} u + \rho \frac{\partial u}{\partial t} + u \nabla \cdot (\rho u) + \rho (u \cdot \nabla) u \\ &= u \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) \right] + \rho \left[\frac{\partial u}{\partial t} + (u \cdot \nabla) u \right] \end{aligned}$$

qui peut être simplifiée en considérant l'équation de continuité :

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \otimes u) = \rho \left[\frac{\partial u}{\partial t} + (u \cdot \nabla) u \right] = \rho \frac{Du}{Dt}$$

Couplée à l'équation de continuité, on obtient finalement les équations de Navier-Stokes compressibles :

$$\left\{ \begin{array}{l} \rho \left[\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right] = \mu \Delta u + (\lambda + \mu) \nabla \nabla \cdot u - \nabla p + \rho f \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \end{array} \right. \quad \text{sur } \Omega \times [0, +\infty[$$

Fluide incompressible Dans le cas incompressible, $\nabla \cdot u = 0$, ce qui implique l'équation de conservation de quantité de mouvement suivante :

$$\rho \left[\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right] = \mu \Delta u - \nabla p + \rho f$$

Par ailleurs, puisque la masse volumique est constante, elle peut être éliminée en posant $\nu = \mu/\rho$ et en opérant la substitution $p \leftarrow p/\rho$, pour obtenir les équations de Navier-Stokes incompressibles :

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nu \Delta u - \nabla p + f \\ \nabla \cdot u = 0 \end{array} \right. \quad \text{sur } \Omega \times [0, +\infty[$$

Ceci introduit une deuxième mesure de la viscosité, la viscosité cinématique ν , mesurée en Stokes, de dimension physique $m^2 \cdot s^{-1}$. Dans la suite de ce travail, et sauf indications contraires, on désignera ces dernières équations comme équations de Navier-Stokes, en adoptant toujours la convention que u , p et f représentent respectivement les champs de vitesse, de pression et de forces volumiques extérieures.

Forme adimensionnelle Une forme commune de ces équations de Navier-Stokes est une forme adimensionnée, qui permet de dégager la caractéristique significative de la situation à simuler, le nombre de Reynolds.

Pour cela, on introduit des quantités de référence et on normalise l'équation par rapport à elles. Il en ressort alors le nombre de Reynolds $Re = L_0 \cdot u_0/\nu$, qui est sans dimension, et qui découle de la viscosité du fluide ν , et des taille L_0 et vitesse u_0 caractéristiques de l'écoulement :

$$\left\{ \begin{array}{l} \frac{\partial \tilde{u}}{\partial \tilde{t}} = -(\tilde{u} \cdot \nabla)\tilde{u} + \frac{1}{Re} \Delta \tilde{u} - \nabla \tilde{p} + \tilde{f} \\ \nabla \cdot \tilde{u} = 0 \end{array} \right. \quad \text{sur } \Omega \times [0, +\infty[$$

Dans ce cas, \tilde{u} , \tilde{p} et \tilde{f} ne représentent plus des variables physiques, mais des variables relatives aux références, par exemple $u = \tilde{u} \cdot u_0$.

Conditions aux limites Pour formuler des problèmes mathématiques bien posés, c'est-à-dire des problèmes admettant une unique solution, les équations de Navier-Stokes précédemment décrites doivent être complétées par des conditions aux limites. Celles-ci comprennent d'une part les conditions initiales de vitesse $u_0 = u(x, 0) : \Omega \rightarrow \mathbb{R}^3$ et éventuellement de masse volumique $\rho_0 = \rho(x, 0) : \Omega \rightarrow \mathbb{R}$ dans le cas compressible et requièrent d'autre part des conditions aux limites sur la frontière du domaine $\partial\Omega$.



Figure 1.2: Claude Louis Marie Henri Navier, France (1785 – 1836) et George Gabriel Stokes, Irlande (1819 – 1903)

Du point de vue de la modélisation physique, Stokes a proposé une condition d'adhérence sur la totalité du bord du domaine, ce qui se traduit par la condition de Dirichlet :

$$u = 0 \quad \text{sur } \partial\Omega$$

Navier a, quant à lui, proposé une condition plus générale :

$$\begin{cases} \lambda_n [u \cdot n] + (1 - \lambda_n) [n \cdot \sigma \cdot n] = 0 \\ \lambda_t [u - (u \cdot n)n] + (1 - \lambda_t) [\sigma \cdot n - (n \cdot \sigma \cdot n)n] = 0 \end{cases} \quad \text{sur } \partial\Omega$$

avec $\lambda_n, \lambda_t \in [0, 1]$ et n la normale unitaire extérieure au domaine. Cette condition englobe la condition de glissement :

$$\begin{cases} u \cdot n = 0 \\ \sigma \cdot n - (n \cdot \sigma \cdot n)n = 0 \end{cases} \quad \text{sur } \partial\Omega$$

À faible vitesse ou dans des cas de viscosité élevée, la condition de Stokes semble expérimentalement plus satisfaisante. Cependant, dans le cas de viscosité faible, celle de Navier apparaît plus valable. On notera qu'il s'agit ici de l'expression de conditions aux limites physiques, et non de conditions aux limites de modélisation, comme on peut les rencontrer dans le cadre de la simulation.

Historique Les équations de Navier-Stokes ont été nommées ainsi en référence à leurs découvreurs, Claude Louis Marie Henri Navier (1785 – 1836) et George Gabriel Stokes (1819 – 1903) — figure 1.2 — qui les ont établies séparément.

Navier les a formulées en 1822. Pour établir ses équations, il a modifié les équations d'Euler afin de tenir compte des forces entre les molécules, sans pour autant bien appréhender correctement la notion de viscosité, mais il a malgré tout formulé des équations correctes *in fine*.

Stokes s'est intéressé ultérieurement au même problème de fluides visqueux, et a, lui, correctement dérivé ces équations en 1845, dans le cadre de l'étude du mouvement d'une petite sphère dans un fluide visqueux.

Même si seuls ces deux noms restent associés à ces équations, il faut souligner que d'autres savants les ont étudiées : Cauchy, Poisson, ou encore Saint-Venant qui a publié un travail analogue à celui de Stokes deux ans avant lui, sans pour autant que son nom soit resté associé à ces équations.

1.2.3 Modèles simplifiés

On peut identifier des situations physiques particulières dans lesquelles les équations de Navier-Stokes se simplifient, parfois drastiquement. On donne ici quelques exemples classiques d'équations qui sont souvent employées lorsqu'une modélisation complète n'est pas nécessaire.

Équations d'Euler Une première simplification possible concerne la nature même du fluide. Si celui-ci est supposé non visqueux, $\lambda = \mu = \nu = 0$, on obtient les équations d'Euler, en version compressible :

$$\begin{cases} \rho \left[\frac{\partial u}{\partial t} + (u \cdot \nabla)u \right] = -\nabla p + \rho f \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0 \end{cases} \quad \text{sur } \Omega \times [0, +\infty[$$

ou non :

$$\begin{cases} \frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\nabla p + f \\ \nabla \cdot u = 0 \end{cases} \quad \text{sur } \Omega \times [0, +\infty[$$

Équations de Navier-Stokes incompressibles stationnaires Plutôt que de simplifier la nature du fluide, il est possible de simplifier la situation simulée. Une situation particulière est celle où un régime stationnaire est atteint, ce qui se traduit par $\frac{\partial u}{\partial t} = 0$. Par conséquent, les équations de Navier-Stokes incompressibles stationnaires s'écrivent :

$$\begin{cases} -(u \cdot \nabla)u + \nu \Delta u - \nabla p + f = 0 \\ \nabla \cdot u = 0 \end{cases} \quad \text{sur } \Omega$$

Équations d'Oseen Une simplification peut être introduite en linéarisant le terme d'advection des équations de Navier-Stokes autour d'une vitesse particulière, pour donner naissance aux équations d'Oseen. Donc, étant donné un champ de vecteur $w \neq 0$ tel que $\nabla \cdot w = 0$:

$$\begin{cases} -(w \cdot \nabla)u + \nu \Delta u - \nabla p + f = 0 \\ \nabla \cdot u = 0 \end{cases} \quad \text{sur } \Omega$$

Équations de Stokes Plus drastiquement que précédemment, il est possible de linéariser les équations de Navier-Stokes autour d'un champ de vitesse nul, ce qui élimine complètement le terme advectif. Dans ce cas, on obtient les équations de Stokes :

$$\begin{cases} -\nu \Delta u + \nabla p = f \\ \nabla \cdot u = 0 \end{cases} \quad \text{sur } \Omega$$

Le problème de Stokes est au cœur du développement de méthodes de simulations de fluide, tant d'un point de vue théorique, en considérant le problème de Navier-Stokes comme une perturbation du problème de Stokes, que d'un point de vue pratique. En effet, comme on le verra dans le cadre de la résolution par éléments finis, le problème de la résolution des équations de Navier-Stokes complètes peut se ramener à la résolution d'une succession de problèmes de Stokes.

1.2.4 Formulation alternative en vorticit 

Les d rivations pr c dentes ont permis d' tablir les  quations de Navier-Stokes incompressibles en employant les variables que sont la vitesse u et la pression p . Cependant, il est possible d'introduire d'autres formulations de ces  quations. Une de ces autres formulations qui a  t  employ e en infographie, et qui use du concept de vorticit  va maintenant  tre pr sent e. On l'illustre dans le cas bidimensionnel.

Deux fonctions scalaires sont introduites, ψ et ω , repr sentant respectivement la fonction de courant, et la vorticit , d finies par :

$$\begin{cases} u_x = \frac{\partial \psi}{\partial y} \\ u_y = -\frac{\partial \psi}{\partial x} \end{cases}$$

et

$$\omega = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}$$

Ces deux fonctions sont donc li es par :

$$\omega = -\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2} = -\Delta \psi$$

Le premier int r t de cette formulation provient de la suppression de la contrainte de continuit , ou plut t de sa prise en charge implicite puisque tout champ u d riv  d'une fonction ψ satisfait automatiquement   cette contrainte :

$$\nabla \cdot u = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = \frac{\partial^2 \psi}{\partial x \partial y} - \frac{\partial^2 \psi}{\partial y \partial x} = 0$$

Ce probl me est compl t  par une  quation d crivant le transport de la vorticit , et qui est obtenue en d rivant les  quations de conservation de quantit  de mouvement pour u_y et u_x respectivement par rapport   x et y :

$$\begin{cases} \frac{\partial}{\partial y} \frac{\partial u_x}{\partial t} = \frac{\partial}{\partial y} \left[-(u \cdot \nabla) u_x + \nu \Delta u_x + \frac{\partial p}{\partial x} + f_x \right] \\ \frac{\partial}{\partial x} \frac{\partial u_y}{\partial t} = \frac{\partial}{\partial x} \left[-(u \cdot \nabla) u_y + \nu \Delta u_y + \frac{\partial p}{\partial y} + f_y \right] \end{cases}$$

et en les soustrayant l'une   l'autre pour obtenir :

$$\begin{aligned} \frac{\partial}{\partial t} \left[\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right] &= -(u \cdot \nabla) \left[\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right] - (\nabla \cdot u) \left[\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right] \\ &\quad + \nu \Delta \left[\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right] + \left[\frac{\partial^2 p}{\partial x \partial y} - \frac{\partial^2 p}{\partial y \partial x} \right] + \left[\frac{\partial f_x}{\partial y} - \frac{\partial f_y}{\partial x} \right] \end{aligned}$$

Cette relation peut  tre simplifi e en consid rant la d finition de ω et l'incompressibilit  du fluide :

$$\frac{\partial \omega}{\partial t} = -(u \cdot \nabla) \omega + \nu \Delta \omega + \left[\frac{\partial f_x}{\partial y} - \frac{\partial f_y}{\partial x} \right]$$

On voit alors apparaître un deuxième avantage à cette formulation : le terme de gradient de pression disparaît également.

La transcription en vorticit  des  quations de Navier-Stokes incompressibles bidimensionnelles s'apparente donc au probl me suivant :

$$\left\{ \begin{array}{l} \frac{\partial \omega}{\partial t} = -(u \cdot \nabla)\omega + \nu \Delta \omega + \left[\frac{\partial f_x}{\partial y} - \frac{\partial f_y}{\partial x} \right] \\ u = \left[\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right]^T \\ \omega = -\Delta \psi \end{array} \right. \quad \text{sur } \Omega \times [0, +\infty[$$

Le cas tridimensionnel est analogue, mais les fonctions de courant et de vorticit  sont alors vectorielles. La situation est toutefois rendue plus complexe du fait que cette paire de fonctions n'existe pas toujours.

Si ces formulations apportent une forme de simplification en s'abstrayant de la pression, il convient de remarquer que la sp cification de conditions aux limites devient complexe. En effet, ni la vorticit , ni sa d riv e ne sont g n ralement connues   l'avance aux bords d'un obstacle puisqu'elles sont d pendantes du flux, qui n'est bien  videmment pas connu   l'avance.

On notera qu'il existe  galement des m thodes reconstruisant directement le champ de vitesse   partir des  chantillons de la vorticit , par une op ration de convolution et sans passer par le calcul de la fonction courant.

Chapitre 2

Simulation de fluides en infographie

L'un des objectifs majeurs de l'informatique graphique est la simulation de phénomènes naturels. Les fluides, tant sous leur forme liquide que gazeuse, n'échappent pas à la règle : des travaux visant à l'intégration automatique ou semi-automatique de fluides dans des environnements virtuels existent depuis plus de vingt ans. L'intérêt porté à cette problématique a néanmoins explosé dans la dernière décennie, et plus particulièrement durant les cinq dernières années.

L'étude bibliographique réalisée ci-après se focalise essentiellement sur le problème de l'animation des fluides en infographie, et non sur celui de leur rendu. Cette orientation est justifiée par l'absence, en général, de structure propre du fluide à visualiser : le fluide n'est simulé que pour déplacer un objet ou agir sur lui. Deux exceptions sont à noter : la première concerne la surface des liquides, qui procède toutefois d'un rendu classique de surface, la deuxième les fumées et autres nuages, nécessitant l'emploi d'algorithmes capables de traiter ces médias participatifs. Mais même dans ces deux cas, il est difficile de distinguer des algorithmes de rendu spécifiques aux fluides. Cependant, et plus particulièrement dans le cas de modèles simples essentiellement empiriques, il arrive que simulation et rendu soient interdépendants, pour former une méthode composite de simulation associée à une forme de rendu particulier.

Dans cette étude bibliographique, seules les techniques prenant en compte une réelle modélisation des fluides dans leur globalité sont examinées. Ainsi, sont écartés les travaux ne modélisant que la surface d'un liquide et son évolution, sans considération de transport de fluide, c'est-à-dire les travaux modélisant directement les conséquences de phénomènes en ignorant leurs causes. Ceci exclut par conséquent la majorité des travaux concernant la simulation des grandes étendues d'eau, même si cette problématique coïncide avec le début de l'inclusion de liquides animés dans des images de synthèse [48, 119]. Pour les mêmes raisons, l'on a écarté certains travaux relatifs aux gouttes d'eau [49, 77].

Néanmoins, le spectre des techniques évoquées reste large, en termes d'échelles physiques — de la goutte d'eau à la portion de lac ou de mer — comme en termes de classes de méthodes. Les méthodes manuelles, dont le rôle est de permettre à un utilisateur humain de concevoir manuellement un champ de

vitesse plausible pour un fluide, seront présentées les premières dans la section 2.1. Ensuite seront détaillées les méthodes de simulation proprement dites, qui autorisent une simulation automatique d'un fluide dans son environnement. Parmi elles, les méthodes Lagrangiennes seront abordées dans la section 2.2, et les méthodes retenant actuellement le plus l'attention, qui sont les méthodes Eulériennes, dans la section 2.3. Après avoir détaillé les modèles complètement discrets dans la section 2.4, on terminera, dans la section 2.5 par des travaux périphériques à la simulation, qu'il s'agisse de l'emploi de simulations, ou de quelques problèmes spécifiques au rendu des fluides. L'ordre de présentation des différentes classes de simulation correspond globalement à l'évolution suivie par le domaine : depuis des méthodes empiriques et manuelles nécessitant très peu de ressources, jusqu'à de réels simulateurs automatiques comparables à ceux utilisés en ingénierie, et qui requièrent des calculs extrêmement lourds mais fournissent des résultats très réalistes.

Parmi les nombreux travaux présentés, on invite le lecteur à se focaliser sur la classe des méthodes Eulériennes procédant de simulations physiques non simplifiées, décrite dans la section 2.3.2, et sur ses extensions ayant trait à l'interaction avec des solides Lagrangiens, détaillées dans la section 2.3.4. C'est en effet dans ce cadre Eulérien que s'inscrivent les travaux de simulation présentés dans cette thèse, et c'est l'interaction entre fluide et solides qui les motive directement. On juge toutefois important de présenter dans sa totalité ce vaste domaine qu'est la simulation, car une présentation détaillée témoigne bien de son évolution.

2.1 Méthodes manuelles

Les approches manuelles décrites dans cette section font partie des toutes premières méthodes à avoir été proposées pour permettre l'inclusion de fluides dans des environnements virtuels. Ces méthodes ne cherchent pas à se substituer à un animateur humain mais à lui donner les moyens de composer de manière aisée une animation, tout en garantissant au résultat une certaine forme de vraisemblance. Puisque le contrôle de l'animateur sur le fluide est direct, ces méthodes travaillent à un niveau cinématique, toujours directement en lien avec le champ de vitesse du fluide.

De fait, ces méthodes ne s'apparentent pas réellement à des simulations du fluide et les ressources qu'elles requièrent restent légères. La contrepartie immédiate est qu'elles restent d'application limitée et ne présentent évidemment aucune forme d'automatisation ou de réaction à un quelconque changement dans l'environnement de la scène, et encore moins d'interaction réelle avec des objets plongés dans le fluide. Malgré tout, elles introduisent déjà une notion de composante stochastique physiquement réaliste qui est employée dès lors que la complexification visuelle d'une simulation simple est souhaitée. Néanmoins, ces modèles ne peuvent garantir un grand réalisme, ils restent essentiellement limités à la simulation de fluides gazeux, vus au travers du transport d'autres objets déstructurés et flous, tels que des nuages ou de la fumée.

2.1.1 Création des mouvements

Le modèle communément utilisé pour construire manuellement un champ de vitesse plausible du fluide est celui proposé par Wejchert et Haumann [176]. Dans

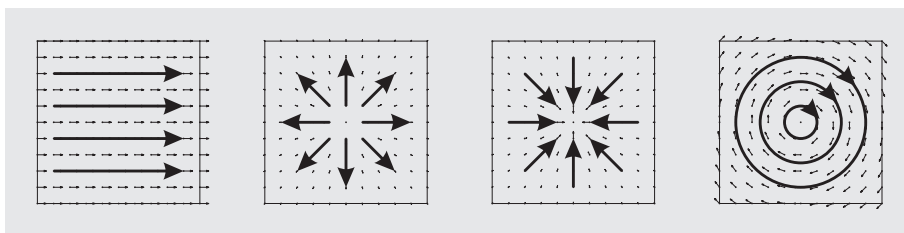


Figure 2.1: Primitives de flux proposées par Wejchert et Haumann [176] : flux uniforme, source, puits, et vortex.

ce modèle, le fluide est supposé incompressible, irrotationnel, et de viscosité nulle. Ces hypothèses correspondent à une modélisation simplifiée d'un fluide, tel que l'air à des vitesses faibles, en régime laminaire. Son champ de vitesse peut être alors être dérivé du gradient d'un champ de potentiel scalaire ϕ , ce dernier satisfaisant à l'équation de Laplace.

$$\begin{cases} u = \nabla\phi \\ \Delta\phi = 0 \end{cases}$$

La méthode proposée fait usage de la linéarité de ces équations pour spécifier que toute solution peut être décomposée en une combinaison linéaire d'autres solutions. En adoptant la démarche inverse, des solutions analytiques classiques sont proposées à l'utilisateur du système comme briques de base pouvant être additionnées pour former un champ final complexe. Les primitives proposées par la méthode et illustrées dans la figure 2.1 — le champ uniforme, le puits, la source et le vortex — restent cependant de nature très simple.

Le travail de conception, ainsi que le placement dans l'espace et l'ajustement des intensités relatives des différentes primitives composant le champ, restent donc complètement à la charge de l'animateur, la vraisemblance finale lui incombant ainsi pleinement. La méthode assure toutefois, par construction, que le champ produit se conforme aux hypothèses de base d'un fluide non-turbulent. Concernant l'intégration du fluide dans un environnement fixe présentant des obstacles, la méthode ne prévoit aucun mécanisme automatique pour que le flux se conforme aux contraintes naturelles de non-pénétration dans les objets. La solution proposée est le placement manuel de primitives dans le but de dévier la trajectoire du fluide, l'empêchant ainsi de franchir les frontières incriminées.

Cette méthode ne propose donc pas un solveur à proprement parler, la prise en compte de conditions aux limites n'étant pas possible automatiquement. Elle doit plutôt être perçue comme une technique assurant un minimum de crédibilité physique aux flux manuellement créés. Malgré cet inconvénient, le contrôle cinématique direct du fluide permet à l'animateur d'obtenir une vision immédiate des mouvements engendrés, et autorise des modifications extrêmement simples.

Turbulence de petite échelle Bien que des champs de vitesse réalistes puissent être créés à l'aide d'une telle technique, il reste difficile à un animateur humain de synthétiser manuellement un résultat comportant des structures de multiples échelles tout en conservant une apparence réaliste. Une solution est proposée par Stam et Fiume [154] par superposition d'un champ de vitesse

turbulent de petite échelle, généré stochastiquement, au champ de plus grande échelle, agencé lui, manuellement.

Afin de conserver une base physique à la turbulence, celle-ci est générée en filtrant un bruit blanc à l'aide du spectre énergétique de Kolmogorov. Le résultat de cette opération effectuée dans le domaine fréquentiel est ensuite reprojété dans le domaine spatial par transformée de Fourier inverse. Le spectre de Kolmogorov utilisé découle de l'analyse des turbulences pour un fluide simplifié, dans lequel l'énergie introduite à une fréquence donnée se propage aux fréquences supérieures à taux constant.

Pour plus de flexibilité et à des fins d'animation, cette synthèse spectrale n'est pas effectuée dans le seul domaine spatial, mais dans le domaine spatio-temporel. Cette synthèse complète permet ainsi de moduler la corrélation dans le temps des structures produites selon leur fréquence. Le spectre spatial est modulé par une Gaussienne dont l'écart-type est fonction de la fréquence spatiale : les perturbations de basses fréquences restent relativement corrélées dans le temps alors que les perturbations de hautes fréquences ne le sont pas du tout.

La synthèse effective est réalisée sur une grille régulière, et elle donne lieu à la création d'un champ de vitesse turbulent périodique. L'échelle de la turbulence reste toutefois limitée à la résolution de la grille. Si cette technique permet l'introduction automatique de phénomènes de petite échelle, très difficilement modélisables manuellement, la nature statistique de l'algorithme la rend en revanche nécessairement globale, la turbulence ne pouvant être cantonnée à une sous-partie de l'espace.

Ceci explique pourquoi cette technique est plus adaptée à la gestion de fumées transportées dans des gaz qu'à celle des liquides. Afin de matérialiser la fumée à transporter, deux approches sont proposées. La première, Eulérienne, est directe, simulant l'évolution d'une variable qui représente la densité de fumée à rendre sur une grille régulière. L'évolution temporelle est alors gouvernée par une équation d'advection et de diffusion. La deuxième technique, Lagrangienne, distribue la densité entre des Gaussiennes advectées par le champ de vitesse, la diffusion étant prise en compte par l'augmentation du rayon de celles-ci. Le rendu peut alors être effectué efficacement par raytracing sur cette collection de sphères, les calculs d'illumination pouvant être effectués analytiquement.

Sur la même base cinématique, Rudolf et Raczkowski [130] s'intéressent à modéliser des fumées denses et turbulentes. Ils proposent pour cela de remplacer la turbulence stochastique par un modèle à base de primitives tourbillonnaires mobiles et localisées dans l'espace. La probabilité de création de ces primitives, en fonction des densité et vitesse locales du fluide, est alors spécifiée par l'utilisateur. Chaque primitive de turbulence est modélisée par un vortex générant son propre champ de vitesse tourbillonnaire, et se trouve elle-même advectée par le fluide. Une forme de flottabilité liée à la densité locale, ainsi qu'une seconde turbulence, sont également adjoints au modèle pour favoriser le développement de panaches de fumée et pour brasser les zones calmes du fluide.

Bien qu'apportant plus de flexibilité qu'un modèle purement statistique, en permettant une localisation spatiale de la turbulence, ce modèle reste difficile à employer de par son côté empirique et complètement manuel. Néanmoins, il permet un contrôle localisé du mouvement à grande comme à petite échelle.

Au contraire de techniques précédemment évoquées, qui cherchent à obtenir un champ de vitesse final complexe en superposant des champs de petites échelles à un champ laminaire global, une autre approche a été proposée, où les

structures fines n'existent que visuellement. Sakas [132] propose ainsi de faire directement évoluer au cours du temps un champ détaillé de densité de fumée, en le représentant et le manipulant dans le domaine spectral. Cette représentation est initialisée à partir d'un champ fractal, le champ de densité effectif dans le domaine Euclidien étant reconstruit à chaque pas de temps, depuis la représentation fréquentielle, à l'aide d'une transformée de Fourier inverse.

Le champ de densité est soumis à un mouvement translationnel global. Eu égard à sa nature simple, l'effet de ce mouvement peut être réduit à un décalage de phase appliqué à la représentation spectrale du nuage. Outre ce déplacement, une perturbation turbulente est également appliquée aux ondes dont les longueurs excèdent un seuil donné, supposées être constitutives de tourbillons. Cette variation est prise en compte par un second décalage de phase, dérivé d'une cascade d'énergie de Kolmogorov.

La même idée, de séparation de l'échelle de simulation et de l'échelle de visualisation, est présentée par Ebert *et al.* [33]. La description des mouvements est analogue à celle proposée par Wejchert et Haumann [176], par combinaison manuelle de primitives cinématiques de haut niveau. Cependant, une totale liberté est laissée quant à la nature des primitives : elles ne sont pas contraintes à vérifier une hypothèse physique, et englobent jusqu'à des champs arbitraires discrétisés sur des grilles régulières.

Les effets de turbulence à petite échelle sont ajoutés durant l'affichage et ne sont constitués que par l'échantillonnage d'une fonction de densité de fumée turbulente finement définie lors du processus de rendu. La spécificité de l'approche est de ne pas chercher à advecter cette densité par le champ de vitesse de grande échelle au cours du temps pour ensuite réaliser un rendu classique. Au contraire, le processus inverse est mis en œuvre : la densité est définie dans l'espace au temps initial uniquement, et reste immobile. En revanche, la recherche d'une valeur de densité associée à une position et un temps donné ne relève plus d'une simple requête à la position désirée, ceci afin de tenir compte de l'évolution spatiale du champ. La valeur de densité renvoyée ne correspond pas à la position donnée dans la requête, mais à celle définie par l'extrémité de la trajectoire originare de la position donnée et engendrée par le champ de vitesse, comme illustré dans la figure 2.2.

En fait, cette méthode s'apparente à un traitement semi-Lagrangien du transport de la densité par le fluide, sans rééchantillonnage intermédiaire à chaque pas de temps mais en rebroussant systématiquement les caractéristiques jusqu'à l'instant initial. Le premier avantage d'une telle approche, tel que décrit par Sims [144], est de ne pas introduire de lissage excessif des structures visuelles fines. Le second est que le coût de rendu reste directement corrélé à la surface du fluide effectivement visible. En effet, le fluide ne fait pas évoluer la densité : seule une requête pour connaître la valeur de densité en un point de l'espace et à un instant donné, engendre effectivement un calcul. Les calculs d'évolution de la densité ne sont donc effectués qu'à la demande du rendu, pour les points du fluide réellement visibles.

Création par pavage Si les méthodes précédemment évoquées permettent la création manuelle de la composante dominante de flux, elles ne fournissent toutefois que peu de primitives munies d'une justification physique pour parvenir à ce résultat. Par ailleurs, ces primitives sont difficilement utilisables à

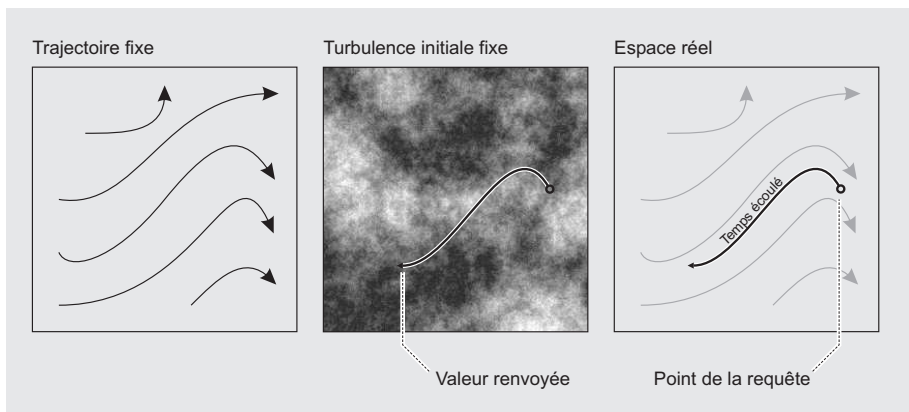


Figure 2.2: Advection des détails effectuée lors du rendu. Le champ de trajectoire fixe est rebroussé jusqu'à l'instant initial pour déterminer la valeur de turbulence qui aurait été advectée jusqu'au point de requête.

proximité d'obstacles présents dans le flux, puisque l'existence même de ceux-ci n'est qu'implicite, et résulte en fait de multiples compensations entre primitives. Cheney [20] propose donc un mode de construction de flux plus adapté au cas de fluides dans des espaces bidimensionnels confinés, et contraints par des conditions aux limites.

La construction n'est pas réalisée par superposition de primitives de flux élémentaires, mais par pavage de l'espace à l'aide d'un ensemble de carreaux portant des flux prédéfinis, comme présenté sur la figure 2.3. Pour apporter une vraisemblance physique, le flux se doit d'être continu et sans divergence. La problématique sous-jacente à une telle approche est, d'une part de générer un ensemble fini et restreint de carreaux satisfaisant à ces contraintes, et d'autre part de faciliter à un utilisateur la sélection des carreaux pouvant légalement être placés à un emplacement donné.

Pour assurer un nombre restreint de carreaux, il est proposé de limiter les vitesses de bords et de coins des carreaux à un petit ensemble de valeurs quantifiées. Par ailleurs, les carreaux symétriques sont supprimés. Sur chaque carreau, le remplissage est assuré automatiquement par une forme d'interpolation des quantités de bord. Pour simplifier la manipulation des carreaux ainsi que pour des raisons de compacité de représentation, le champ de vitesse est manipulé sous la forme d'une fonction de courant.

Malgré cette approche relativement restrictive, un très grand nombre de modèles de carreaux est rapidement généré. Il n'est donc pas envisageable de présenter en permanence à un utilisateur humain la totalité de l'ensemble des carreaux lors de la phase de construction manuelle. Outre ce problème lié à la cardinalité des ensembles de carreaux manipulés, se pose un problème de compatibilité. En effet, pour rendre l'édition pertinente, l'utilisateur peut définir des conditions aux limites sur les flux aux bords du domaine à paver avant de débiter le processus de construction. En raison de la contrainte d'incompressibilité et des conditions à satisfaire sur le bord, des cas de pavages partiels sans terminaison possible peuvent se rencontrer. Afin de prévenir de telles situations, un algorithme permettant la recherche automatique des carreaux compatibles

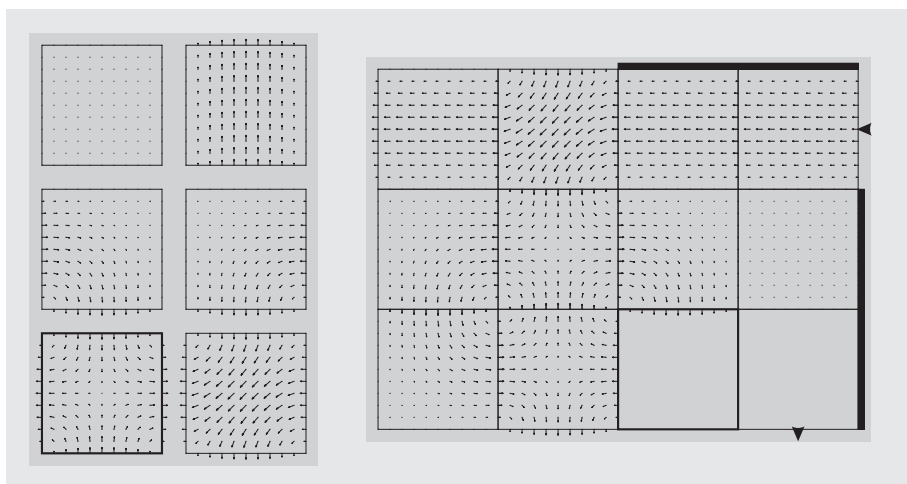


Figure 2.3: Construction de flux par pavage à base de carreaux prédéfinis. Gauche : un ensemble de carreaux générés. Droite : Un pavage en construction avec des conditions aux limites imposées. Pour remplir l'emplacement mis en valeur, un seul carreau est proposé par l'algorithme car c'est le seul permettant une terminaison globale du pavage, bien qu'un autre carreau satisfasse également aux contraintes locales.

avec une situation partiellement construite est donc proposé. Cet algorithme, employant des optimisations linéaires entières par méthode du simplexe, permet de déterminer le sous-ensemble des carreaux garantissant un non-blocage ultérieur du pavage. Le pavage partiel peut alors être augmenté de n'importe quel carreau proposé dans cette liste, soit automatiquement, soit manuellement. La construction complète du pavage est ainsi réalisée en itérant cette recherche jusqu'à détermination complète du pavage.

L'intérêt d'une telle approche par pavage est double. D'abord, la création des situations reste très simple et intuitive, tout en confinant l'intégralité des calculs à la phase de création, et non à la phase d'exploitation du flot créé, qui reste dans ce cas physiquement vraisemblable. Ensuite, la représentation du résultat construit reste extrêmement efficace du point de vue du stockage. Néanmoins, et même si des conditions aux limites peuvent être fixées avant le début du pavage, il faut souligner qu'une telle méthode ne constitue pas un solveur d'une équation sous-jacente à proprement parler. De plus, la question reste posée de la taille, et donc de la variété, des ensembles de carreaux modèles en regard des situations à gérer et de leur taille.

2.1.2 Réaction d'objets à la cinétique du fluide

Il est souvent intéressant, voire indispensable d'un point de vue visuel, de placer des objets dans un environnement rempli d'un fluide en mouvement. Lorsque ces objets présentent une physique propre, au contraire de la simple fumée entraînée passivement par le fluide, ils se doivent de réagir de manière cohérente aux mouvements du fluide, comme des arbres oscillant sous le vent. Si des simulations physiques complètes peuvent être réalisées, leur coût souvent élevé paraît peu justifiable dans un cas où la modélisation du fluide reste approxima-

tive et surtout figée, sans réelle interaction possible. Pour obtenir des résultats à coût réduit, des modèles simples tirant partie des informations de vitesse, et éventuellement de caractéristiques statistiques du fluide ont été développés. Ils représentent ainsi un bon complément aux modélisations cinématiques du fluide précédemment décrites.

Modèle de forces explicites Une première approche est proposée par Wejchert et Haumann [176] pour exploiter les champs de vitesse créés par superposition de primitives cinématiques. Des particules, ou des objets maillés plus complexes gérés par des systèmes masses-ressorts, peuvent être introduits dans le fluide. Plutôt que de transporter directement les objets, et afin de rendre les scènes plus réalistes, la force exercée par le fluide sur les objets est évaluée en fonction de la vitesse du fluide là où ils sont localisés, puis leur est appliquée. Notons que cette influence n'est pas réciproque, en raison de la nature statique du champ de vitesse créé, et qu'aucune interaction véritable ne peut prendre place entre solides et fluide.

Cette technique très directe a l'avantage d'être simple à mettre en œuvre. De plus, le mouvement des objets introduits reste généralement proche des trajectoires du fluide lui-même : la prévision des mouvements résultants est donc rapidement correcte, sans requérir de multiples ajustements. Toutefois, une telle approche requiert la création effective d'un champ de vitesse, et éventuellement son animation manuelle pour obtenir des mouvements complexes au cours du temps. Par conséquent, si elle semble bien adaptée pour gérer peu d'objets éparpillés dans une scène, vérifiant l'hypothèse de couplage non-bidirectionnel, elle se révèle moins attractive pour la gestion d'objets plus denses et plus complexes, soumis à un fluide moins laminaire.

Dans le cas d'une action sur un objet peu dense et déformable, comme un tissu, une telle technique se trouve donc limitée par l'aspect statique du champ de vitesse, qui ne permet pas de simuler des effets temporaires de masquage entre divers objets ou d'un objet sur lui-même. Keckeisen *et al.* [80] proposent une technique permettant de résoudre ce problème : chaque primitive est supposée émettre des particules qui lui sont propres, qui suivent son champ de vitesse et qui sont influencées par les collisions avec des objets déformables. Les particules fournissent donc une indication dynamique sur l'influence spatiale réelle de chaque primitive, comme illustré sur la figure 2.4. L'environnement est alors discrétisé par une grille, et chaque cellule de celle-ci cumule les vitesses issues des primitives, uniquement si elle se trouve traversée par une particule durant le pas de temps considéré. Cette grille représente alors un champ de vitesse incorporant des effets de masquage dynamiques. Par suite, il est possible de calculer les forces aérodynamiques s'appliquant aux objets à partir de ce champ.

Réponse à une turbulence statistique Aucune de ces deux techniques ne paraît bien adaptée à la simulation de l'interaction avec un fluide non laminaire, en raison de la difficulté à modéliser ce type de fluide en employant des primitives. Pour la gestion d'une telle situation, illustrée par des arbres soumis aux mouvements du vent, deux approches similaires ont été proposées.

Shinya et Fournier [142] proposent une méthode tirant partie d'une analyse modale des systèmes mis en présence du fluide, où plusieurs sous-approches sont considérées pour recréer des animations réalistes de différents végétaux dans des

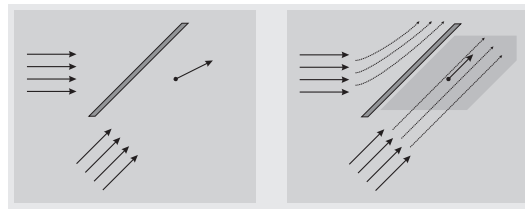


Figure 2.4: Détermination dynamique de l'influence des primitives d'un flot statique. Gauche : les primitives définies directement dans tous l'espace ne sont pas affectées par l'obstacle. Droite : l'emploi de particules sensibles aux collisions permet de détecter la masquage dû à l'obstacle.

scènes naturelles.

Dans le cas de structures très rigides, assimilables à des barres rigides, tels des bambous ou des branches d'arbres, les déformations des structures sont produites en excitant les différents modes de vibrations de la structure d'après les forces externes appliquées par le fluide. Ces modes sont déterminés analytiquement grâce à la forme particulièrement simple de la structure. Dans le cas d'une structure arborescente, des forces locales sont propagées de branche en branche, vers le tronc. Pour des objets plus souples, telle de l'herbe, il est proposé de ne pas déformer des modèles préexistants, mais de faire osciller directement, à l'aide d'oscillateurs amortis, certains paramètres d'un générateur paramétrique. Finalement, pour des objets existant sans structure géométrique définie, tels les feuillages d'arbre représentés à l'aide d'une texture plaquée sur géométrie grossière, une hypothétique réponse fréquentielle du feuillage est supposée et celle-ci est combinée aux mouvements du fluide, également représenté dans le domaine fréquentiel, pour produire une animation.

Stam [148] propose une autre approche, relativement similaire dans le contexte précis des arbres. Mais plutôt que d'employer une transmission approximative des forces entre les branches et de gérer chaque branche indépendamment, il est proposé de modéliser exactement la structure globale d'arbre. Si chaque branche reste représentée par une barre, la structure topologique de l'arbre est effectivement préservée, et un problème global de dynamique des structures est formulé : toutes les branches de l'arbre sont donc bien en interaction simultanée, et l'inconnue recherchée correspond aux déplacements en translation et rotation de chaque nœud de l'arbre. Les données d'un tel problème, outre celles de la géométrie de l'arbre, consistent en les forces appliquées à l'arbre.

La méthode proposée s'appuie sur une analyse modale du problème posé pour obtenir efficacement une solution. Le problème est donc reformulé dans le domaine fréquentiel, et les premiers modes vibratoires de la structure sont déterminés. Dans la base ainsi formée, forces appliquées et déplacements engendrés sont très simplement corrélés. C'est donc dans cette dernière base que la dynamique de l'arbre est simulée, avant d'être retransformée dans le domaine spatial.

Concernant les forces effectivement appliquées, plutôt que de les dériver d'un champ de vitesse environnant donné, il est proposé de les déduire d'un modèle de vent stochastique, avec l'avantage de pouvoir directement générer les forces nécessaires dans le domaine fréquentiel, sans conversion entre domaine spatial

et fréquentiel, et donc de gagner en efficacité. Le modèle utilisé présente une corrélation dans sa composante spatiale, les branches proches étant ainsi soumises à des forces similaires.

Réponse précalculée Une approche plus directe est proposée par Perbet et Cani [120] pour modéliser de manière interactive les mouvements d'une prairie sous l'influence du vent. Dans ce type d'approche, le fluide n'est en fait pas réellement modélisé, en tous les cas pas durant la phase d'utilisation de la technique. En revanche, une liaison directe entre application d'une primitive cinématique au fluide et déformation de l'herbe est préétablie durant une phase de précalcul. Ceci autorise alors une approximation des mouvements de la prairie sans recourir à aucune simulation effective, garantissant ainsi l'efficacité de la méthode.

Le vent est toujours modélisé de manière cinématique à l'aide d'une composition de primitives, placées et déplacées par l'utilisateur. Pour chacune de ces primitives, est préévalué l'effet du souffle induit sur un brin d'herbe, en termes de variation de direction et de courbure. Ce précalcul est effectué à l'aide d'un modèle physique, pour un intervalle de temps donné après que la primitive ait commencé à influencer le brin d'herbe. En un sens, ce précalcul transforme les primitives de mouvement du fluide en primitives de mouvement d'herbe. Durant le déroulement de l'animation, l'influence précalculée est appliquée, pour chaque primitive, aux portions de prairie lui étant assignées. Cette approche revient essentiellement à rejouer une animation enregistrée pour les zones de prairie affectées. Cette reprise est légèrement perturbée pour plus de vraisemblance et de variabilité. Afin d'augmenter encore la variété des animations, différentes caractéristiques physiques associées à différents types d'herbes peuvent être employées simultanément.

Cette approche est également généralisée à une modélisation de portion de prairie à plus grande échelle. S'appuyant sur des imposteurs pourvus de textures semi-transparentes, elle peut être employée pour animer les imposteurs eux-mêmes, mettant alors en mouvements un ensemble de brins d'herbe simultanément.

Une telle approche d'animation d'objets soumis à un fluide est rendue possible par la relative simplicité de la situation. En effet, la faible variabilité du modèle à simuler — un ou plusieurs brins d'herbe soumis à une primitive donnée — permet au précalcul de fournir des résultats simples, en l'occurrence le fléchissement engendré ainsi que sa direction. Si cette méthode est très efficace, une limitation reste toutefois posée en raison de l'absence de gestion des superpositions de primitives : la combinaison de celles-ci n'étant pas prise en compte, une et une seule primitive peut être active à un instant donné en chaque point de l'environnement.

Bilan Bien que simples et approximatives, ces différentes méthodes pour animer des objets plongés dans un gaz, permettent d'obtenir des résultats convaincants à un coût raisonnable. Néanmoins leur applicabilité reste limitée : les structures animées restent peu déformables et le fluide, de faible densité, n'exerce sur elles qu'une influence limitée. En outre, le fluide n'étant pas matérialisé et ne possédant pas de surface, aucune corrélation visuelle n'est requise entre les mouvements du fluide et ceux des solides. En aucun cas, les techniques proposées

ne permettent de considérer une réelle interaction entre fluide environnant et solides, bien qu'il soit envisageable d'émuler manuellement et de manière approximative un tel comportement.

2.2 Méthodes Lagrangiennes

Les méthodes purement cinématiques présentent l'avantage de la simplicité d'utilisation ainsi qu'une relative efficacité, mais elles ne permettent pas de faire émerger automatiquement des comportements non prévus par l'animateur. Elles requièrent donc un important travail manuel, la vraisemblance du résultat final incombant totalement à l'animateur. C'est pour de telles raisons que ce type de méthodes a plutôt été développé pour être appliqué à des fluides peu denses sans surface visible à l'image.

Dans le cas de fluides présentant une surface clairement identifiée et visible, des méthodes automatiques plus précises sont donc requises. Pour atteindre cet objectif, de nombreuses méthodes de simulation physique ont été présentées. Toutes ces simulations partagent un caractère dynamique, et l'évolution temporelle du fluide est gouvernée par des principes physiques tenant compte des forces mises en jeu. Les méthodes présentées ci-après sont fondées sur une telle approche, en adoptant un point de vue Lagrangien.

Dans ces méthodes Lagrangiennes, une grande diversité d'approches co-existe. La plus usuelle consiste à considérer le fluide au travers d'une collection de particules régies par la mécanique Newtonienne. Les différentes particules peuvent être employées soit pour matérialiser le fluide dans l'espace, soit de manière plus élaborée pour tenir un rôle dans la simulation en étant le siège de calculs d'interactions. Finalement, une approche plus récente, la simulation SPH, *Smoothed Particle Hydrodynamics*, apporte une réelle justification physique en réalisant la liaison entre ce type de systèmes et les lois exprimées de manière continue en mécanique des fluides. Une dernière classe de méthodes Lagrangiennes, moins usuelle, fait usage d'une formulation en vorticit , les entit s Lagrangiennes maintenant alors cette information pour garder au fluide un caract re tr s turbulent.

2.2.1 Particules sans interactions

Dans l'approche la plus simple, introduite par Reeves [129], le syst me est compos  de particules repr sentant chacune une portion finie de mati re. Chaque particule, assimilable   un point mat riel du point de vue m canique, est soumise   des forces pouvant varier au cours du temps. Les positions des diff rentes particules  voluent ind pendamment dans le temps en accord avec les lois de la m canique Newtonienne, bien que de mani re non encore formalis e.

Du point de vue du contr le, l'utilisateur a la possibilit  de sp cifier les lieux de cr ation des particules, ainsi que les param tres initiaux assign s aux particules nouvellement cr ees. Pour introduire une apparence visuelle finale plus naturelle, ces param tres ne sont pas sp cifi s exactement, la g n ration effective r pondant   une variation statistique simple dirig e par l'utilisateur sous la donn e de moyennes et de variances. Pour plus de contr le, une particule est dot e d'une dur e de vie maximale, apr s laquelle elle est supprim e du syst me. D'autres contraintes de position peuvent  galement provoquer une

suppression prématurée pour simuler des effets plus complexes. Dans le même but, il est également possible d'utiliser un modèle hiérarchique, afin de lier les lieux et paramètres de génération de particules à des particules elles-mêmes déjà existantes.

L'intérêt d'une telle méthode provient de la nature intrinsèquement déstructurée des nuages de particules formés des éléments simulés. En n'imposant aucune contrainte géométrique ou topologique entre les différentes particules, ce mode de simulation permet l'obtention de structures hautement déformables, à même de représenter explosions, feux d'artifice, ou fluides de manière réaliste. Toutefois, il est nécessaire de souligner que si un mécanisme de simulation est bien présent, l'animation résulte encore essentiellement de la conception des forces externes appliquées aux particules, aucun comportement ne pouvant émerger de lui-même. En revanche, la géométrie globale du système émerge, elle, de la simulation.

Pour qu'une telle approche soit viable, un grand nombre de particules différentes doivent être présentes dans la simulation afin de fournir l'illusion d'un *continuum* à l'image. Pour permettre l'utilisation d'un grand nombre de particules, il est proposé par Sims [143] d'employer une machine fortement parallèle dans ce but. Dans la mesure où les particules restent sans interaction, le même algorithme simple d'évolution temporel étant employé pour toutes les particules, cette méthode s'adapte bien à une architecture SIMD massivement parallèle.

Outre la démonstration de l'utilisation d'une machine massivement parallèle, Sims propose également de rendre le système hybride en mêlant contrôle dynamique à l'aide de forces, et contrôle cinématique des particules à l'aide de contrôleurs de trajectoires capables de modifier directement les trajectoires. Il est ainsi possible de contrôler plus aisément certaines trajectoires suivies par les particules, et de simuler, par exemple, la présence d'obstacles sur lesquels viennent rebondir les particules. Avec telle technique, il commence à être possible d'obtenir, de manière quasi-automatique, des comportements émergents physiquement réalistes, comme celui d'une chute d'eau perturbée par des obstacles.

Configuration de simulation spécifique Afin de rendre la simulation plus efficace, dans le cadre particulier de la simulation de chutes d'eau, Mallinder [98] propose de ne pas représenter les particules indépendamment. En considérant une source de particules en amont de la chute d'eau, les trajectoires de toutes les particules issues de celle-ci sont supposées pouvoir être interpolées entre les trajectoires des particules de vitesses minimale et maximale issues de ce point. Ces deux trajectoires extrêmes sont donc précalculées en tenant compte de l'environnement et échantillonnées, les vitesses limites étant données, pour un ensemble de sources discrétisant la largeur de la chute d'eau. Cette étape peut s'apparenter à la discrétisation d'un certain nombre de profils de la chute d'eau.

Afin d'éviter l'introduction de particules réelles, le rendu est effectué directement à partir de cette information. Lors de la phase de rendu, une forme de texture volumique aléatoire est plaquée le long de trajectoires aléatoires interpolées à partir des profils stockés. L'information chaotique de densité, extraite de la texture volumique, permet alors de simuler une variation cohérente de la présence d'eau dans le temps et dans l'espace.

En ne stockant qu'un nombre réduit de trajectoires, et en synthétisant de manière procédurale le fluide à partir de celles-ci, peu de ressources-mémoires sont nécessaires par rapport à une modélisation purement particulaire, mais la simplicité d'une telle méthode est perdue. Néanmoins, la validité de l'hypothèse d'interpolation des trajectoires, qui est plausible dans des cas simples, devient discutable dans des cas d'environnements plus complexes.

Optimisations Dans le cadre de simulations Lagrangiennes, la qualité des résultats ainsi que la performance de la méthode dépendent fortement du modèle retenu quant aux forces appliquées aux particules. Unbescheiden et Trembilski [169] dérivent ainsi un modèle de forces adapté à la simulation de l'évolution des gouttelettes d'eau composant un nuage, qui reste évaluable simplement. Cette force est définie en considérant une modélisation physique simplifiée du système réel et une forme de turbulence simple. Les résultats obtenus en considérant différentes simplifications sont comparés pour déterminer le point pour lequel la modélisation répond aux exigences de vraisemblance posées. L'objectif d'une telle démarche est de limiter au maximum le coût de simulation, en raison de contraintes de performance dictées par l'application de la technique en environnements immersifs. Pour la même raison, un nombre de particules réduit est employé, celles conservées formant la base de petits polygones déformables texturés autorisant un rendu à moindre coût.

Dans la même optique d'efficacité, et toujours dans un but d'application aux fumées, Holtkämper [71] propose l'emploi d'une approche à deux niveaux. Les particules sont regroupées en groupes rigides, chaque groupe étant simulé comme une particule unique. La réaction automatique des groupes de particules à des obstacles est automatiquement prise en compte en considérant une collision inélastique. Cette collision mettant en jeu tout un groupe est cependant un peu plus complexe qu'avec une particule isolée, puisqu'une variation de la vitesse de rotation du groupe est produite, ainsi qu'un changement de sa taille.

Lors du rendu, les différentes particules constitutives à chaque groupe sont projetées à l'écran et texturées. Afin de produire une apparence réaliste, elles sont munies d'une opacité variable. Pour assurer une cohérence temporelle et spatiale à cette variabilité au sein même du groupe, l'opacité attribuée est dérivée d'une fonction de turbulence fixe, échantillonnée en tenant compte des déplacements du groupe et des positions fixes des particules à l'intérieur de celui-ci. Du fait de leur rigidité, les cohérences spatiales et temporelles des différentes particules sont garanties.

Les simplifications opérées par ces deux dernières techniques permettent d'obtenir des résultats relativement convaincants en maintenant des performances proches du temps réel dans des cas simples. Néanmoins, ces simplifications, de même que les approximations visuelles considérées, ne restent valides que dans le cas de la représentation de fumées. Quand des fluides présentant une surface, comme des liquides, doivent être représentés, il reste difficile de limiter la simulation à un faible nombre de particules en raison de la faible pertinence des imposteurs, la surface d'un liquide présentant des effets optiques marquants difficilement approximables de manière convaincante.

Ces différentes approches, où des particules figurent un fluide, sont intéressantes de par la nature déstructurée de la représentation mise en œuvre. Celle-ci évite en effet d'induire des limitations quant aux mouvements simulables, liées

à la représentation choisie pour le fluide. De plus, les particules présentent une forme de discrétisation du fluide intuitive, leur trajectoire matérialisant naturellement celle du fluide dans son ensemble.

Néanmoins, dans les méthodes présentées jusqu'ici, les particules ne sont soumises qu'à des actions externes, le plus souvent prédéterminées par l'animateur. Dans ce cas, la simulation réalisée ne peut être que très partielle, la mécanique interne au fluide restant difficilement capturable par ce biais. Pour cette raison, les techniques présentées ont pour objectif soit la reproduction de gaz, où les interactions internes au fluide restent faibles à une échelle visuelle, les particules étant essentiellement utilisées pour réagir à un environnement, soit la reproduction de liquides fortement perturbés, comme dans le cas d'une cascade d'eau, où les forces externes de gravité et de réaction aux collisions dominent largement, du moins visuellement, celles internes au fluide.

2.2.2 Particules en interaction

Dans la veine des simulations Lagrangiennes où les particules n'ont pour rôle que de porter une information, principalement de présence de fluide, à travers leur position, des techniques plus évoluées ont été développées, où les particules prennent part au calcul des forces qui leur sont appliquées. Les particules ne sont alors plus uniquement affectées par des forces statiques prédéfinies par un animateur externe mais réagissent à des forces calculées dynamiquement en fonction de la configuration courante du fluide, ouvrant la porte à de réelles simulations dynamiques.

Modélisation des actions du fluide Cette démarche est présentée par Haumann et Parent [66] en employant une formalisation un peu plus générale : un certain nombre d'acteurs, regroupant masses ponctuelles et générateurs de forces, coexistent et s'échangent des informations dans un environnement de simulation commun. Les masses évoluent toujours en accord avec les lois de Newton, les forces issues des générateurs s'appliquant sur elles. Les forces effectives sont calculées en réponse à la configuration courante des particules, chaque type de générateur étant à même de répondre à une situation spécifique telle que la simulation de la traînée, de la gravité, ou encore de la réponse à une collision. Il est à noter que dans la formalisation proposée aucune restriction n'est posée quant à la nature ni au nombre des paramètres des générateurs de force. Ceci permet d'unifier aussi bien un générateur de gravité, agissant sur une seule masse, qu'un générateur de traînée considérant des triplets de masse comme éléments de surface en mouvement dans un fluide. Outre ce schéma d'évolution physique, la simulation reste modifiable de manière externe et arbitraire afin de simuler des contraintes simples pouvant être posées sur des acteurs, telles des contraintes de position.

À l'aide d'un tel système sont présentées des animations d'objets en mouvement dans des fluides. Elles mettent en jeu une modélisation masse-ressorts de solides, principalement soumis à une traînée aérodynamique due à leur vitesse par rapport au fluide ainsi qu'à la gravité. Des comportements totalement émergents sont obtenus par ce biais, comme le flottement d'un drapeau ou d'un rideau dans le vent, ou les mouvements erratiques d'une feuille de papier tournoyant dans l'air lors de sa chute.

Moyennant la description de forces entrant en ligne de compte dans les phénomènes aérodynamiques à simuler, cette technique jette ainsi les bases d'une réelle simulation physique automatique mettant en jeu des fluides. L'objectif de l'animateur n'est ainsi plus de décrire directement l'animation désirée, mais de décrire la situation physique sous-jacente, afin d'obtenir une simulation réaliste automatiquement. Il reste toutefois à noter que, dans la méthode proposée, la description du système physique et de ses interactions reste donnée *a priori*, par la description de l'interaction statique entre les différents acteurs de la simulation. Par ailleurs, le fluide n'est jamais directement simulé, seuls ses effets sur un objet qui y est plongé sont modélisés.

Simulation du fluide Miller et Pearce [103] proposent alors l'évolution logique consistant à étendre ce principe de particules interagissant à la simulation du fluide lui-même. La physique Newtonienne est toujours employée pour régir le mouvement des masses matérialisant le fluide et il est proposé de faire naître des forces entre toutes les paires de particules composant le fluide, dès lors que la distance entre les deux membres d'une paire est inférieure à un seuil donné. Une telle approche permet de modéliser des interactions localisées dans le fluide.

La forme proposée de la force comprend un terme d'attraction et de répulsion, corrélé à la distance entre les particules, ainsi qu'un terme de viscosité, corrélé à la différence de vitesses entre les particules. En faisant varier les différents paramètres de ces forces, il est ainsi possible de simuler différents types de matériaux, depuis les poudres jusqu'aux solides élastiques, en passant par les fluides. Dans ce dernier cas, la force est choisie pour amortir fortement les différences de vitesse tout en ne maintenant qu'une faible cohésion entre les particules. Outre ces forces internes au fluide, des forces externes peuvent être aisément ajoutées pour simuler la gravité, ou tout autre effet souhaité ou phénomène physique modélisé. La gestion des collisions avec des obstacles est, elle, prise en charge par des réponses impulsionnelles modifiant directement la position des particules.

Une telle méthode où toute particule peut se retrouver en interaction avec toute autre, peut toutefois s'avérer coûteuse, exhibant une complexité quadratique en nombre de particules. Or, il est nécessaire d'employer un grand nombre de particules pour donner l'illusion d'un *continuum* à cette méthode intrinsèquement discrète. Cette complexité peut cependant être réduite, en tirant profit de la localité des interactions. Un partitionnement spatial de la scène est alors employé pour extraire rapidement les particules candidates à une interaction non-nulle, celles-ci étant confinées à seulement quelques cellules contiguës du partitionnement. Une telle optimisation permet alors de ramener la complexité moyenne de l'algorithme à une complexité linéaire, en considérant une distribution spatiale relativement homogène des particules. Mais un tel partitionnement statique de l'espace n'a d'influence que sur la complexité moyenne, et non sur la complexité dans le pire des cas.

Si la démarche décrite permet la simulation, entre autres, de fluides visqueux, encore faut-il pouvoir afficher les résultats obtenus, afin d'en tirer parti. Dans la mesure où aucune connaissance *a priori* de la forme du fluide n'est disponible, aucune surface à afficher n'est connue. Il est donc proposé de faire usage des particules simulées comme du squelette d'une surface implicite. Cette approche reste extrêmement simple et directe, quoique pas nécessairement économique

en ressources. Par ailleurs, si une surface visualisable est bel et bien générée, il reste difficile d'en contrôler précisément la position par rapport à son squelette, et par rapport à un environnement extérieur.

Chiba *et al.* [23] proposent une approche similaire, en variant la forme des forces d'interaction. Celle qui est employée est librement inspirée des interactions de type *Van der Waals* prenant naissance à l'échelle moléculaire. Elle reste toutefois une combinaison d'attraction, de répulsion, et de viscosité entre particules, dont la magnitude décroît avec l'augmentation de la distance d'interaction. Si les forces d'attraction et de viscosité entre particules sont usuellement définies sur la seule base des positions et vitesses de particules, la force de répulsion employée fait, elle, appel à une mesure supplémentaire de densité locale des particules. Cette densité reste toutefois approximative, et découle d'un comptage du nombre de particules par cellule d'une grille partitionnant l'espace de simulation. Cette grille est par ailleurs employée pour simuler les collisions pouvant survenir entre particules : plus la densité de particules est mesurée comme importante, plus la probabilité de collision d'une particule en mouvement avec les particules peuplant la cellule est supposée élevée, la collision effective étant ensuite déterminée stochastiquement. Des collisions élastiques supplémentaires surviennent également avec les obstacles existant dans le domaine de simulation.

Dans ces deux approches, des animations de fluides réalistes sont obtenues, dans lesquelles le fluide synthétique adopte naturellement des réactions réalistes autour des obstacles définis dans le domaine de simulation. L'utilisation de nuages de points comme mode de représentation permet en outre d'obtenir des animations présentant des mouvements violents ou des environnements complexes, cas où la surface du fluide subit des modifications drastiques.

Extension de la nature des fluides simulés Considérant une approche de simulation similaire, Terzopoulos *et al.* [162] cherchent à étendre la simulation à des solides qu'il soit possible de liquéfier par fusion. Pour cela, il est proposé de superposer deux modèles distincts, mais toujours Lagrangiens, un premier modélisant le comportement de solide élastique, et un deuxième prenant en compte la simulation de fluides. La transition entre ces deux modèles est dictée par une température simulée en sus.

Les solides élastiques sont initialement maillés à l'aide d'un réseau de ressorts élastiques, en considérant un maillage hexaédrique de particules. La raideur des ressorts est directement déterminée par la température locale des deux particules connectées. Ainsi, au delà d'une certaine température, la raideur diminue jusqu'à s'estomper totalement. Parallèlement à ces interactions élastiques, des interactions de cohésion sont toujours simulées entre les particules. À ces deux types d'interaction s'ajoutent les forces liées à la non-pénétration des objets dans les obstacles, ainsi que d'éventuelles forces de frottement. Les interactions de cohésion s'inspirent du potentiel de Lennard-Jones qui modélise les interactions entre molécules neutres ou atomes, et qui engendre des forces attractives à longue distance mais répulsives à courte distance.

Outre cette double simulation structurelle, la chaleur est simulée en sus, et diffusée entre les particules en phase solide en suivant les ressorts. En employant une telle modélisation, des objets exhibant un comportement à la fois élastique et fluide, sont simulables. Initialement introduits froids dans l'environnement, et donc possédant des liaisons élastiques très raides dominant le modèle, ils ap-

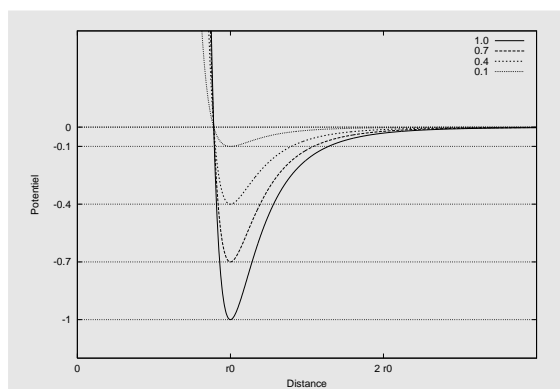


Figure 2.5: Échantillons de la famille de potentiels de type Lennard-Jones proposée par Tonnesen [166]. Les potentiels, indexés par leur profondeur, sont fonction de la différence entre le potentiel d'énergie inter-particulaire ϵ fixe et thermique ψ variable. La profondeur du puits de potentiel est proportionnelle à leur différence.

paraissent comme des solides. En les chauffant ultérieurement, cette élasticité disparaît progressivement, les différentes particules adoptant alors un comportement fluide. Si cette transition de fusion peut être ainsi réalisée sans difficulté, le modèle ne définit toutefois pas la transition inverse. En effet, l'arrangement régulier initialement employé dans la définition des interactions élastiques ne sauraient être valide pour reconstituer de nouvelles liaisons une fois les particules dispersées, après être passées par la phase liquide.

Dans la même voie de simulation d'objets susceptibles de fusion, Tonnesen [166] propose d'employer un modèle unifié plutôt que la superposition de deux modèles distincts. Pour cela, une famille de potentiels d'interaction inter-particules de type Lennard-Jones est considérée. Tous les potentiels engendrent des forces répulsives à faible distance mais attractives à plus longue distance. En revanche, la profondeur du puits de potentiel varie au sein de la famille, sans pour autant se déplacer. Une telle modélisation revient ainsi à considérer des particules cherchant toujours à conserver un espacement fixe — puisque la position du puits de potentiel ne varie pas — mais dont l'affinité avec cette position est plus ou moins grande, le puits étant plus ou moins profond, comme illustré dans la figure 2.5.

L'interaction exacte employée dépend alors de la température locale des particules : moins la température est élevée, plus le puits de potentiel est profond, et plus l'arrangement est stable, donc le matériau simulé rigide. Outre cette force de cohésion, la température des particules est donc simulée en considérant une diffusion, approximée par une relaxation locale des températures.

$$\phi_{\psi}(r) = \frac{-\epsilon + \psi}{n - m} \left(m \left(\frac{r_0}{r} \right)^n - n \left(\frac{r_0}{r} \right)^m \right)$$

Le modèle proposé est donc homogène et permet d'unifier la simulation des comportements solides et liquides. Cependant, la modélisation, bien qu'apparemment plausible, reste non réaliste du point de vue physique : elle ne fait qu'émuler, de manière détournée, la variation d'équilibre entre énergie cinétique et énergie potentielle inter-particulaire des objets, mais ne la simule pas correc-

tement.

Justification visuelle du mode de simulation En se focalisant uniquement sur la phase fluide et son comportement, Luciani *et al.* [95] et Habibi *et al.* [61] cherchent à définir, dans le contexte d'une simulation purement Lagrangienne, le modèle de force minimal susceptible de reproduire une collection de phénomènes physiques réellement observables. Il est montré que l'emploi d'une composante visqueuse, limitée dans l'espace, permet l'obtention de phénomènes tourbillonnaires comme les instabilités de Kelvin-Helmholtz, mais n'est pas suffisant pour faire émerger tous les comportements désirés, notamment ceux liés à des variations de pression dans le fluide. Pour enrichir le modèle proposé, une interaction élastique est donc superposée à un modèle purement visqueux. L'introduction d'une forme d'élasticité, toujours limitée dans l'espace, autorise alors la formation et le déplacement d'ondes de chocs dans le fluide, en cherchant à maintenir une densité homogène de particules dans l'espace, ce qui permet l'obtention de phénomènes plus complexes telles les allées tourbillonnaires de Von Karman. Dans l'approche proposée, la portée des interactions élastiques est choisie comme relativement importante, afin de capturer les variations de densité des particules sur une zone relativement large, comparativement à la portée de la composante visqueuse. Malgré un système de composition simple, il est donc prouvé, par l'exemple, que les seules interactions visco-élastiques sont suffisantes pour faire émerger automatiquement des comportements réalistes complexes.

Une dernière composante de la dynamique des fluides, la diffusion, est introduite directement durant l'étape de visualisation. Lors de leur déplacement, les particules laissent une trace sur leur passage. Les différentes contributions, s'estompant progressivement au cours du temps, sont cumulées en une mesure de densité globale du fluide, témoin de la position et des trajectoires de l'ensemble des particules durant les derniers instants. Le retour progressif de la densité perturbée à une valeur de référence simule alors un processus de diffusion du milieu. La mise en œuvre de cette technique est en fait une simulation physique simple supplémentaire, dans laquelle un réseau de masses liées entre elles et à une référence fixe par des interactions visco-élastiques, se retrouve soumis à des forces émanant des particules du fluide. La densité locale est dérivée de la déformation locale du réseau, celui-ci, de par les caractéristiques de ses interactions, simulant alors naturellement la diffusion. Dans la mesure où la résolution de cette mesure de densité peut être très supérieure à celle des particules composant le fluide, cette technique est également assimilable à une forme de raffinement de la représentation du fluide.

Extension à plusieurs fluides En suivant toujours la même approche particulière et afin de pouvoir exprimer des situations plus complexes, des travaux ont visé à l'utilisation de plusieurs types de particules simultanément. Ainsi, Murta et Miller [109] proposent d'enrichir ce type de simulation par l'emploi simultané de plusieurs familles de particules. Il devient alors possible d'exprimer plus de variété dans les interactions, celles-ci dépendant des familles d'origines des particules considérées. Toutefois, aucune utilisation concrète ne vient présenter de résultats mettant à profit cette différenciation possible. Par ailleurs, le contrôle de la densité des particules est abordé, en autorisant la séparation et la fusion de particules existantes, selon des critères tels que la survenue d'un

impact ou la faible densité locale pour la séparation, ou la présence contiguë de petites particules présentant des caractéristiques de vitesse similaires pour la fusion.

Steele *et al.* [156] reprennent la même idée de familles multiples de particules, mais l'illustrent effectivement en simulant différents matériaux au sein d'une même scène, certains matériaux pouvant relever des solides et d'autres des liquides visqueux. Les différentes forces appliquées simulent alors l'adhésion entre deux particules, la viscosité dans le cas où les particules font partie du même liquide, ou la friction dans le cas où une particule se situe à proximité d'un objet. L'adhésion entre particules fait usage de la notion de familles de particules, les interactions entre toutes les familles possibles étant choisies parmi trois types de comportement : l'attraction pure, la répulsion pure, ou bien une combinaison des deux, similaire à une interaction de type Lennard-Jones, répulsive en deçà d'un certain seuil mais attractive au delà.

Pour simuler la combustion violente survenant lors d'une explosion, Take-shita *et al.* [160] proposent également de modéliser la situation en employant deux familles de particules, l'une matérialisant l'air, l'autre matérialisant les matériaux enflammés. Les différentes forces, appliquées à toutes les particules quelle que soit leur nature, comprennent une forme de flottabilité liée à la température locale, de répulsion simulant la pression locale ainsi que de viscosité. Les particules portent également une température, qui évolue au cours du temps par simulation de la diffusion de la chaleur. Toutefois, aucune réaction ne s'opère entre les particules des deux familles, les particules d'air n'étant pas simulées pour faire office de comburant, mais pour matérialiser l'air environnant plus frais, aspiré par l'explosion, et ainsi moduler le déplacement en régulant la température locale. Pour jouer ce rôle, les particules d'air sont initialement disposées dans l'intégralité de l'environnement de simulation.

Ainsi, dans cette dernière simulation, les différentes particules ne jouent plus nécessairement un rôle symétrique, et se voient agrémentées de propriétés supplémentaires destinées à la simulation de phénomènes plus complexes. Cette logique peut être poussée plus avant, les particules n'étant pas nécessairement restreintes à la représentation d'une fraction de fluide, mais pouvant être associées à des phénomènes complets, de plus haut niveau.

Modélisation de plus haut niveau Une approche phénoménologique est proposée par Neyret [110] dans le cadre de la simulation de la formation de nuages. La méthode proposée substitue à un modèle physique complet une modélisation fondée sur l'emploi de bulles d'air chaud comme primitives de haut niveau représentatives dans la formation de nuages. La simulation est découplée en deux sous-modèles distincts : le premier, de la couche basse de l'atmosphère, est chargé de la création des bulles d'air chaud ; le deuxième, de l'atmosphère à partir de l'altitude du point de rosée, est chargé de l'évolution des nuages déjà formés.

Le premier modèle, chargé de simuler la naissance des bulles d'air chaud modélise le sol du domaine de simulation. Dans cette simulation bidimensionnelle, des particules représentant de petites fractions d'air chaud évoluent dans le domaine. Elles sont créées stochastiquement suivant le réchauffement engendré par le sol. Une fois créées, ces particules sont soumises à diverses forces simulant le transport d'air dans cette couche de surface. Les particules proches sont

agrégées et lorsque les forces s'appliquant sur une particule excèdent un seuil prédéterminé, une bulle d'air est considérée comme quittant ce sous-domaine de simulation. La particule en question est donc supprimée, et une bulle déjà formée est directement placée dans la deuxième simulation, celle de l'atmosphère, à l'altitude du point de rosée.

Dans cette deuxième simulation, les bulles composant le nuage sont en constante évolution : de nouvelles bulles sont constamment créées dans la zone la plus chaude du nuage, absorbant une partie de la chaleur de leurs voisines, puis sont déplacées. Suite à ce déplacement, une redistribution de la chaleur entre les bulles peut se produire pour limiter la redondance spatiale entre bulles. Afin de rendre encore plus complexe l'apparence visuelle des nuages, des vortex, représentés comme des entités Lagrangiennes supplémentaires, sont introduits sur la surface des bulles. Ces vortex sont créés en permanence sur la partie supérieure des bulles et simultanément entraînés par le mouvement des bulles elles-mêmes. Ces sous-structures sont alors peu ou prou assimilables à la surface du nuage.

Dans une telle simulation, les causes des phénomènes ne sont que très partiellement simulées, l'accent étant mis sur la simulation des conséquences directement observables. La polyvalence de la simulation est donc restreinte au profit d'une diminution des ressources requises par la simulation. L'approche Lagrangienne sied bien à un tel paradigme de simulation, car elle permet de focaliser la simulation sur les points jugés d'importance. Notons qu'il convient dans ce cas de choisir les conséquences simulées avec attention pour conserver au modèle le réalisme recherché. Par ailleurs, la versatilité de la simulation s'en trouve fortement réduite, car limitée au cadre précis choisi.

Toujours à haut niveau, Chiba *et al.* [22] proposent, dans le cadre d'une modélisation bidimensionnelle de fumées et de flammes, d'employer un système de particules pour matérialiser les vortex en mouvement dans le fluide. Ce système de particules est chargé de donner au champ de vitesse du fluide son caractère turbulent. Un modèle analytique de vortex est alors associé à chacune des particules, celles-ci étant soumises à diverses forces, dont certaines issues des vortex eux-mêmes. Dans la mesure où les particules sont sensibles aux collisions avec les obstacles et avec des forces externes, il est possible d'obtenir un champ turbulent assez facilement contrôlable, et qui s'adapte automatiquement à la géométrie de la scène.

Simulation parallèle des particules Dans le cadre d'une animation employant un système de particules, une des clés du réalisme réside dans le grand nombre de particules simulées pour donner l'illusion d'un *continuum*. Une technique pour conserver des performances acceptables dans un tel cas, est d'exploiter une forme de calcul parallèle, comme le propose Sims [143], à l'aide d'une machine massivement parallèle. Dans une approche plus moderne, un matériel graphique programmable peut être utilisé pour gérer la physique du système de particules en interactions. Dans un cas d'interactions simples à base de collisions élastiques, Kipfer *et al.* [84] proposent une telle implémentation, en employant un tri bitonique [8] conjointement à un partitionnement spatial de la scène par une grille régulière, pour détecter les interactions entre particules. Ce tri est choisi parce qu'il est bien adapté au tri en parallèle des différentes particules stockées dans des textures et indépendant des valeurs triées elles-mêmes. Néanmoins, les requêtes de proximité ne peuvent pas être parfaitement calculées,

mais seulement approximées en raison de la programmabilité restreinte du matériel graphique. Les réponses fournies sont toutefois suffisamment bonnes pour ne pas introduire de comportement visuellement aberrant dans la simulation.

2.2.3 Méthode SPH

Si les techniques particulières précédemment décrites sont à même de produire des animations de fluides d'un réalisme convaincant, elles présentent le défaut de n'établir de corrélation avec la physique continue que dans les résultats obtenus. Si ce défaut peut sembler anecdotique dans le cadre de l'infographie, il risque de se révéler réhibitoire lors de la mise au point d'animations, les nombreux et différents paramètres manipulables des forces d'interaction ne correspondant nullement à des quantités physiques connues telles la viscosité ou la densité. Trouver le jeu de paramètres requis pour obtenir un comportement précis se révèle alors difficile, car relevant d'une recherche manuelle complexe à effectuer. En outre, ces interactions locales ne permettent pas la réelle simulation d'une équation globale gouvernant l'évolution du fluide.

Les méthodes employant le formalisme SPH, *Smoothed Particle Hydrodynamics*, cherchent à remédier à ces problèmes en adoptant une approche alternative, dans laquelle les particules sont le support d'une discrétisation des champs physiques continus sous-jacents, et où il est possible de considérer des équations formulées sur les *continuum* discrétisés.

Base de simulation

Développée initialement dans le domaine de l'astrophysique par Lucy [96] et par Gingold et Monaghan [53], cette méthode a été introduite ultérieurement en informatique graphique par Desbrun et Cani [28] dans le contexte de l'animation d'objets hautement déformables, jusqu'à des matériaux quasi-liquides.

Le principe de la modélisation SPH est de regarder chaque particule comme un point échantillon des champs continus manipulés dans le problème traité. Pour pouvoir considérer ces champs continus, il est nécessaire de formuler un schéma d'interpolation. Celui-ci est construit en considérant que chaque particule possède une influence spatiale radiale positive décroissante en fonction de la distance. En tout point de l'espace r , une valeur interpolée $\langle q(r) \rangle$ peut alors être calculée, en sommant les contributions individuelles q_i liées à chaque particule de la discrétisation. Cette interpolation est contrôlée par le choix du noyau d'influence W_h . Pour que l'interpolation reste indépendante de la densité locale de particule, pouvant évoluer au cours du temps, un facteur de correction est ajouté à ce schéma d'interpolation sous la forme d'une pondération relative à la densité locale $\rho_i = \langle \rho(r_i) \rangle$. Celle-ci est calculée dynamiquement en considérant que chaque particule porte une masse fixe prédéfinie m_i , et en utilisant le schéma d'interpolation décrit.

$$\begin{aligned}\langle \rho(r) \rangle &= \sum_i m_i W_h(r - r_i) \\ \langle q(r) \rangle &= \sum_i m_i \frac{q_i}{\rho_i} W_h(r - r_i)\end{aligned}$$

L'intérêt d'un tel schéma d'interpolation est qu'il est extensible au calcul de quantités différentielles sur une base similaire. Ainsi, en considérant non

plus le noyau d'interpolation lui-même mais son gradient, calculé de manière analytique, il est possible d'interpoler le gradient du champ discrétisé de manière cohérente.

$$\langle \nabla q(r) \rangle = \sum_i m_i \frac{q_i}{\rho_i} \nabla W_h(r - r_i)$$

De manière analogue, il est possible d'exprimer un opérateur Laplacien, bien que dans ce cas il soit généralement préférable d'employer un schéma ne faisant intervenir que le gradient du noyau plutôt que son Laplacien, afin d'obtenir une plus grande stabilité.

Un intérêt fort de la méthode est que le seul paramètre de ce schéma d'interpolation est le choix de la fonction noyau. Celle-ci reste arbitraire sous deux conditions : sa normalisation d'une part, et la condition qu'en cas de raffinement de la discrétisation, cette fonction tende vers une fonction de Dirac. Afin de permettre des optimisations par partitionnement spatial, les noyaux choisis sont généralement nantis d'un support compact.

$$\int W_h(x) = 1$$

$$\lim_{h \rightarrow \infty} W_h(x) = \delta(x)$$

Une fois cette discrétisation mise en place, il est possible de dériver des forces cohérentes avec divers phénomènes physiques décrits par des équations exprimées à un niveau continu, incluant des termes différentiels. Il est à noter que ces forces ne dépendent pas de paramètres empiriques déterminés par un animateur externe, mais sont systématiquement dérivées des quantités physiques apparaissant dans les équations continues qui constituent le point de départ de la modélisation. Une telle technique représente donc le lien entre une modélisation physiquement correcte, non empirique, et les modèles particuliers. Néanmoins, le choix du noyau reste délicat, celui-ci n'étant pas sans influence sur les forces effectivement générées. Après choix d'un noyau adapté, il se révèle que dans une modélisation classique des fluides, la combinaison des forces de cohésion et de celles de répulsion dues à la pression résulte alors en une interaction similaire à celle de Lennard-Jones, couramment employée dans les systèmes de particules plus empiriques.

Stora *et al.* [158] suivent cette approche pour simuler un matériau assimilable à un liquide extrêmement visqueux, la lave. Dans ce cas, la viscosité est directement corrélée à la température locale du milieu. Cette dernière est prise en compte par la simulation d'une équation de diffusion, également traitée avec la méthode SPH.

Müller *et al.* [105] enrichissent ce type de modélisation des fluides, en incluant des effets de tension de surface. Une force supplémentaire est donc appliquée aux particules proches de la surface du liquide, en plus de la répulsion liée à la pression et de la viscosité. Cette force est logiquement calculée à partir de la courbure locale de la surface, à l'aide d'une évaluation SPH s'appuyant sur la présence des particules. Il est également proposé d'employer des noyaux différents selon les termes à évaluer. Ainsi, les calculs liés à la répulsion utilisent un noyau favorisant le non-regroupement des particules, ceux liés à la viscosité, un noyau ne pouvant provoquer d'accélération parasites.

Gestion de l'incompressibilité Bien que découlant en ligne relativement directe d'une modélisation physique cohérente, les méthodes précédemment évoquées peinent à garantir une caractéristique essentielle des liquides, la conservation de la masse, qui se traduit en général par une conservation du volume. Même si les variations découlant de la reconstruction d'une surface sont ignorés, ces méthodes n'incorporent pas de mécanisme spécifique destiné à garantir l'incompressibilité du fluide et se satisfont de la tendance naturelle du système à recouvrer une densité uniforme, sous l'action des forces de cohésion, *i.e.* la pression, engendrées par la variation de densité.

Clavet *et al.* [26] proposent une approche pour tenter de pallier ce défaut, en introduisant une interaction de cohésion plus fine entre particules. La technique proposée consiste à calculer deux mesures locales différentes de densité, à l'aide de deux noyaux légèrement différents, et à calculer l'interaction de cohésion à partir de ces deux valeurs. La cohésion dépend alors linéairement de la pression usuelle, et de manière quadratique de la pression plus locale. Une telle approche empêche la formation d'agglomérats denses et localisés de particules dont la densité calculée serait, à tort, considérée comme la densité cible au repos du fluide. En effet, un tel type d'agglomérat ne peut être stable vis à vis des deux mesures de densité. L'on constate également que cette méthode engendre comme effet de bord des effets de tension de surface, sans nécessiter un calcul explicite de courbure, du fait de sa réaction à l'anisotropie des configurations de particules proches de la surface.

L'approche la plus aboutie pour garantir l'incompressibilité dans le cadre de méthodes de type SPH est proposée par Premože *et al.* [123]. Il emploie la méthode MPS — *Moving Particle Semi-Implicit* — qui garantit l'incompressibilité mais au prix de calculs supplémentaires. Cette technique suit la méthode de projection de Chorin et résout les équations de Navier-Stokes en deux étapes : une première considérant l'évolution due uniquement à la conservation de la quantité de mouvement du fluide, puis une seconde, de projection dans l'espace des champs-solutions à divergence nulle, satisfaisant à la contrainte d'incompressibilité.

La première étape fait usage d'une simulation SPH classique, similaire à celle suivie dans les méthodes précédemment évoquées. Une fois les particules déplacées, la contrainte d'incompressibilité n'est plus satisfaite. La correction, en termes de vitesse, à apporter aux particules peut toutefois être reliée à la pression, calculable en considérant la résolution d'un système linéaire lié à la configuration courante du fluide. Le système en question est donc formé et résolu, puis la correction est apportée aux particules, pour former le nouveau champ satisfaisant à la contrainte d'incompressibilité. Cette méthode de projection correspond exactement à celle suivie dans le cas des méthodes Eulériennes actuellement les plus populaires, sous réserve d'une application dans un cadre SPH Lagrangien plutôt que dans le cadre plus classique d'une simulation Eulérienne par différences finies.

Cette méthode respecte bien la contrainte de conservation de volume mais ne permet pas d'introduire ou de supprimer des particules dans la simulation sans la perturber et ainsi violer la contrainte d'incompressibilité. Il n'est donc pas possible, en l'état, d'inclure des conditions aux limites intéressantes du point de vue graphique dans la simulation, tels des sources ou des puits de fluide. Une extension de la méthode MPS remédiant à cette limitation est donc également proposée par Premože *et al.*, sous la forme de la méthode MPS-MAFL, *MPS* -

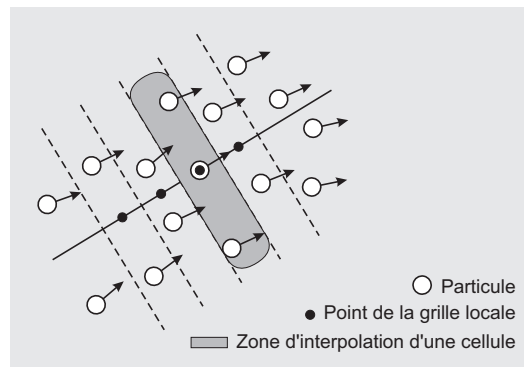


Figure 2.6: Grille locale utilisée par la méthode MAFL dans les calculs de convection par un schéma de différenciation en amont. La grille locale, orientée dans le sens du courant est construite en interpolant les vitesses des particules proches.

Meshless Advection using Flow direction Local grid initialement développée par Yoon *et al.* [186].

Dans cette approche, la méthode MPS précédemment décrite est utilisée pour faire évoluer la simulation sur un pas de temps, en termes de position comme de propriétés portées, puis la position des particules est corrigée pour tenir compte des conditions aux limites et redistribuer spatialement les particules. Finalement, les valeurs des propriétés portées par les particules déplacées sont corrigées en conséquences. Cette correction est requise en raison du déplacement arbitraire, non physique, subi par celles-ci. Elle est gérée à l'aide d'un schéma de convection Eulerien appliqué sur une grille monodimensionnelle, alignée localement avec le flux, et dynamiquement générée autour de chaque particule, comme illustré sur la figure 2.6.

Extensions

Plusieurs extensions, que l'on va maintenant décrire, ont été proposées au modèle SPH afin de permettre des simulations de situations plus complexes.

Simulations multi-phasiques Par exemple, il est possible de réaliser des simulations bi-phasiques ou multi-phasiques, où plusieurs fluides immiscibles sont simulés dans un environnement commun. Comme ces fluides présentent des caractéristiques physiques différentes, des phénomènes de tension de surface doivent être incorporés dans les simulations. Les informations de courbure et de normales à l'interface, nécessaires au calcul des forces de tension étant simplement calculables en utilisant le formalisme SPH, cette première extension reste relativement immédiate, puisqu'elle se ramène au calcul d'une force supplémentaire à proximité de la surface, et à son application.

Premože *et al.*, réalisent également un couplage entre les différentes phases prenant place durant la correction due à la conservation de masse, c'est-à-dire durant le calcul de la pression. Si les différentes phases présentent une densité similaire, un seul calcul d'incompressibilité, commun, est réalisé, englobant simultanément tous les fluides. Dans le cas contraire, pour une interface entre

fluides de densités fortement dissemblables, les calculs d'incompressibilités sont réalisés itérativement jusqu'à convergence en vitesses et positions des particules.

Müller *et al.* [107], emploient des forces de tension de surface à l'interface entre phases et proposent également de généraliser la viscosité, pour qu'elle puisse agir simultanément sur les phases au sein d'un seul calcul global. Par ailleurs, afin de simuler des matériaux dont la densité varie en fonction de la température locale, l'évolution de cette dernière est également prise en compte au travers d'une équation de diffusion. En outre, une gestion spéciale est mise en place dans le cas d'un gaz environnant un liquide, afin de gérer de manière plus précise et plus efficace la formation et l'évolution de bulles de gaz prisonnières du fluide. Les particules associées au gaz sont créées et détruites dynamiquement, restant confinées à la frontière du liquide pour plus d'efficacité en évitant une simulation complète : aucune simulation de l'environnement gazeux éloigné du fluide n'est donc effectuée, celui-ci étant supposé négligeable. Néanmoins, des poches de particules de gaz emprisonnées au milieu du liquide peuvent toujours se former, matérialisant précisément des bulles.

Interaction avec des objets déformables Outre ces simulations ne mettant en jeu que des fluides, des simulations couplant fluides avec objets solides ont également été réalisées.

Ainsi, Müller *et al.* [106] proposent de modéliser l'interaction de solides déformables triangulés avec des liquides représentés par la méthode SPH. Les solides sont simulés à l'aide d'une simulation par éléments finis et donc représentés par un maillage volumique. Pour calculer une interaction, une force de traction et de répulsion, analogue à une interaction de Lennard-Jones est définie entre fluide et solide. L'idée principale de la méthode est d'exprimer cette force entre toute particule de fluide proche de l'objet et la surface de l'objet lui-même de manière intégrale, et d'approximer cette intégrale par quadrature Gaussienne sur chaque triangle de la surface de l'objet. Cette interaction entre deux points, le point de Gauss et la particule de fluide, est dès lors aisément calculable. La force répulsive ainsi définie a également pour rôle une émulation de la condition d'imperméabilité de la frontière des solides. Outre cette force de répulsion, une force de viscosité, destinée à émuler la friction entre fluide et solide à l'interface, est gérée sur le même mode.

Extension des matériaux simulés Keiser *et al.* [81], au contraire, proposent d'unifier le traitement des fluides et des solides, fondant l'intégralité de la simulation, tant pour les fluides que pour les solides, sur la méthode SPH. L'approche adoptée consiste à considérer les états fluide et solide comme les deux *extrema* d'un *continuum*. Chaque état limite répond alors à des lois propres, et toutes les particules sont vues comme appartenant à ce *continuum*, subissant donc la superposition des interactions fluides et solides, à des degrés divers, selon l'état local du matériau. Les différentes forces impliquées comprennent alors l'élasticité, la plasticité, la viscosité et la pression, ainsi que la tension de surface. Dans l'approche proposée, les paramètres de ces interactions sont interpolés entre les valeurs fournies pour les deux états limites fluide et solide selon un paramètre local de température. Cette dernière évolue au cours du temps par diffusion au sein du matériau, cette diffusion étant elle-même simulée par la méthode SPH. L'intérêt d'une telle modélisation est évidemment de pouvoir réaliser dynami-

quement et de manière complètement automatique des transitions entre états au cours de la simulation.

Procédant de la même idée, celle de l'enrichissement des forces d'interaction internes au matériau, Clavet *et al.* [26] étendent le champ des matériaux simulables en prenant pour base des liquides. Élasticité, viscosité et plasticité sont ainsi combinées aux forces internes classiques pour permettre la simulation de fluides fortement visqueux. Cette complexification des forces mises en jeu s'accompagne alors d'un schéma d'évolution temporel à base de prédiction et de relaxation, se rapprochant légèrement d'un schéma implicite, plus stable qu'une méthode d'Euler explicite.

Bilan La formulation SPH, relativement récente, représente le dernier développement des méthodes Lagrangiennes. Elle fournit ainsi une justification théorique aux expressions de forces employées dans les simulations, estompant l'empirisme associé aux systèmes de particules. Néanmoins, il faut à noter que si le principe est justifié théoriquement, il reste, sinon systématiquement, du moins souvent, employé de manière aveugle, sans se préoccuper réellement de la physique sous-jacente. C'est notamment le cas pour les dernières extensions présentées, les enrichissements simulant des matériaux non-physiques restant, eux, totalement empiriques, bien qu'inspirés par des principes physiques. Un autre point critique quant à l'utilisation concerne les limites de la méthode qui ne présente généralement pas de garantie d'incompressibilité. En outre, la question de la pertinence de la simulation, relativement à la densité d'échantillonnage, reste sans réponse.

2.2.4 Approche vorticité

Un dernier type de méthodes Lagrangiennes, assez peu développé en infographie, concerne les méthodes mettant en œuvre une approche par simulation de la vorticité. Dans cette approche, empruntée relativement récemment à la communauté de Mécanique des Fluides, les champs de vitesse et de pression du fluide ne sont pas simulés directement, mais déduits d'une quantité intermédiaire, la vorticité, dont l'évolution est, elle, simulée explicitement. Il faut souligner que cette approche découle d'une modélisation physique réaliste et d'une reformulation des équations sous-jacentes, et ne procède pas d'une approximation des phénomènes turbulents pouvant se produire dans le fluide.

On rappellera que la vorticité est définie comme le rotationnel du champ de vitesse $\omega = \nabla \times u$, et que, lors de la reformulation des équations de Navier-Stokes en employant la vorticité, la contrainte d'incompressibilité devient implicite, tout comme les forces liées à la pression, comme détaillé dans la section 1.2.4. Par conséquent, une seule quantité nécessite d'être simulée, et celle-ci n'est pas contrainte, au contraire de la formulation plus usuelle en vitesse et en pression. Cette reformulation entraîne donc une simplification apparente du problème, simplification qui a son revers car le champ de vitesse n'est alors plus directement disponible, et doit être reconstruit par une opération non-triviale. La représentation Lagrangienne de la vorticité se justifie par le fait que la vorticité est en partie advectée par le fluide lui-même, et qu'elle tend à être spatialement concentrée dans les zones fortement tourbillonnaires.

Les deux méthodes proposées par Angelidis et Neyret [2] et par Park et Kim [118] procèdent en discrétisant la vorticité, qui est une quantité vectorielle

pour les simulations 3D, par un ensemble de primitives Lagrangiennes. Lors de la simulation, ces primitives évoluent dans l'espace et les quantités de vorticit  port es sont mises   jour.   chaque pas de temps, le champ de vitesse correspondant est recalcul  en employant une convolution de la vorticit  avec un noyau de Biot-Savart. Ce champ est ensuite employ  pour d placer les primitives discr tisant la vorticit .

Il est   remarquer que Gamito *et al.* [50] proposent  galement une approche similaire, qui sera d taill e dans la section 2.3.2, la reconstruction du champ de vitesse  tant dans ce cas effectu e par r solution d'une  quation diff rentielle sur une grille Eul rienne. Au contraire, les m thodes propos es par Angelidis et Neyret et par Park et Kim. sont purement Lagrangiennes et emploient une reconstruction directe, utilisant des noyaux de convolution.

Si dans le principe ces deux m thodes adoptent une approche similaire, la repr sentation effective diff re drastiquement dans le cas de simulations tridimensionnelles : pour Park et Kim la vorticit  est int gralement port e par des particules peuplant l'int gralit  du domaine de simulation, alors qu'Angelidis et Neyret emploient un nombre restreint de courbes comme primitives. Ce dernier choix est justifi  par le fait que la vorticit  tend   se structurer en filaments, souvent ferm s, dans l'espace, et par la vis e de la m thode. En effet, si Park et Kim adoptent une simulation directe et relativement compl te, Angelidis et Neyret introduisent un certain nombre d'approximations afin de s'approcher d'un syst me d'animation interactif.

La premi re approximation concerne le noyau de convolution employ  pour reconstruire le champ de vitesse   partir de la vorticit  : le noyau de Biot-Savart est remplac  par une approximation de forme plus simple int grable analytiquement pour plus de performance. La seconde approximation consiste en une gestion dynamique de primitives courbes, supports de la simulation. Le niveau de discr tisation de ces courbes, *i.e.* la pr cision de l'approximation et la quantit  de calculs n cessaires   la reconstruction du champ de vitesse, reste ainsi limit , au prix toutefois, d'une  volution plus complexe que dans le cas de simples primitives particuli res. Finalement, la viscosit  est int gralement mod lis e par une att nuation de la vorticit  port e par les primitives au cours du temps. En fait, la technique propos e ne mod lise pas exactement la vorticit  mais emploie plut t la circulation de celle-ci le long des courbes tenant lieu de primitives, ce choix offrant l'opportunit  de plusieurs optimisations n cessaires   la r alisation d'un simulateur interactif. Malgr  ces quelques simplifications et adaptations, la gestion de l' volution des filaments de vorticit  reste loin d' tre triviale. Une cons quence est que seules sont consid r es des conditions aux limites simples, de non-p n tration avec des plans fixes.

Malgr  les simplifications op r es, des r sultats d'une tr s bonne qualit  visuelle peuvent  tre obtenus   des co ts de calcul faibles, autorisant une m thode interactive dans des situations simples. Cette performance s'explique principalement par le fait qu'un nombre relativement restreint de primitives est suffisant pour engendrer un flot complexe et ainsi produire des animations plaisantes. Les primitives choisies sont donc bien   m me de capturer l'essence visuelle des animations demand es. En outre, les primitives Lagrangiennes offrent   l'utilisateur un mode de contr le relativement direct et intuitif, localisant ais ment les zones turbulentes dans l'espace.

L'approche suivie par Park et Kim est plus directe et l'int gralit  de l'espace est initialement discr tis e r guli rement   l'aide de particules ponctuelles. Les

termes de convection des particules et de modification de la vorticit   port  e, sont   valu  s en reconstruisant le champ de vitesse et son gradient aux positions des particules. Pour cela, la convolution par le noyau de Biot-Savart est discr  t  s  e en employant les positions des particules comme points de Gauss. Par cons  quent, le champ de vitesse peut   tre reconstruit comme combinaison lin  aire d'un ensemble de fonctions de base port  es par les particules. Dans la mesure o   ces fonctions de base sont sol  no  idales, le champ g  n  r   satisfait intrins  quement    la contrainte d'incompressibilit  . Pour   viter la complexit   quadratique associ  e au fait que toutes les particules contribuent au champ de vitesse en tout point de l'espace, un algorithme hi  rarchique, la m  thode multip  le rapide, est employ  e lors de la reconstruction. La diffusion est, quant    elle, simul  e par un processus d'  change de vorticit   entre particules voisines, approximant l'op  rateur de diffusion Laplacien.

Les conditions aux limites de non-p  n  tration sont g  r  es par la m  thode des panneaux. Celle-ci consid  re un ensemble de vortex plac  s sur la fronti  re des objets, et d  termine les vorticit  s individuelles requises pour annuler le champ de vitesse    la surface de l'objet calcul  , sans prise en compte de conditions aux limites. Il est    noter que le calcul des vorticit  s associ  es aux conditions aux limites n'est pas simple, puisque toutes influent en tout point de l'espace. Les influences   tant interd  pendantes, un syst  me lin  aire doit donc   tre r  solu    chaque pas de temps. Les conditions aux limites d'adh  rence sont r  solues par   mission de particules de vorticit      proximit   des bords du domaine de simulation. La vorticit   port  e par ces particules est corr  l  e    celle d  termin  e par des conditions de non-p  n  tration, mais il est possible de la moduler pour simuler des perturbations plus ou moins importantes du fluide par la fronti  re.

Une telle approche a pour int  r  t principal de permettre l'obtention de deux facteurs difficiles    concilier. Tout d'abord, le champ de vitesse g  n  r   satisfait, par construction,    la contrainte d'incompressibilit  . Ensuite, les r  sultats ne souffrent pas de la dissipation num  rique associ  e    la phase d'advection Eul  rienne, en raison de la formulation Lagrangienne employ  e. N  anmoins, les conditions aux limites restent relativement complexes    imposer, et limit  es dans leur champ d'application puisque d  finies uniquement sur la vorticit  , et non sur la vitesse ou la pression.

2.3 M  thodes Eul  riennes

   l'oppos   des m  thodes Lagrangiennes qui suivent l'  volution du fluide    travers des points mobiles au cours du temps, on trouve les m  thodes Eul  riennes, o   les points de discr  tisation employ  s sont fixes au cours du temps. Ainsi, cette approche mod  lise plus directement le caract  re continu des fluides simul  s, et la discr  tisation effective des   quations d'  volution peut   tre r  alis  e de mani  re plus naturelle, sur un support immobile. En revanche, la d  finition d'un sch  ma de discr  tisation fixe peut   galement introduire des contraintes de repr  sentativit  , un liquide ne pouvant, plus n  cessairement, adopter n'importe quelle configuration spatiale sans violer des hypoth  ses de base de la repr  sentation choisie.

Parmi les m  thodes Eul  riennes, il est possible de distinguer tout d'abord les m  thodes relevant de mod  les simplifi  s ou restreints quant    leur repr  sentativit  , d  crites dans la section 2.3.1. Cependant, la plupart des mod  les propos  s

et employés, détaillés dans la section 2.3.2, découlent des simulations des équations de Navier-Stokes ou de leur dérivées. Ces modèles connaissent par ailleurs de nombreuses extensions destinées à augmenter leur champ d'application qui sont détaillées dans la section 2.3.3. En outre, le contrôle de ces méthodes par un animateur, problématique propre à l'infographie, a récemment fait l'objet d'un nombre conséquent de travaux. Cette facette des simulations Eulériennes est présentée dans la section 2.3.5. Finalement, comme décrit dans la section 2.3.6, l'efficacité de ces méthodes peut être sensiblement augmentée par l'utilisation du matériel graphique au cours de la résolution numérique des équations.

2.3.1 Modèles simplifiés ou réduits

En raison de la difficulté inhérente à la résolution des équations de Navier-Stokes tridimensionnelles complètes, en termes de complexités de méthodes comme en termes de ressources requises, les simulations Eulériennes proposées se restreignent parfois sur deux points. D'une part, l'emploi de modèles simplifiés peut-être préféré à des simulations physiquement complètes. D'autre part, la dimension des problèmes peut être réduite, et une extrapolation peut éventuellement prendre place pour émuler la dimension perdue. Ces deux types de simplifications s'accompagnent alors logiquement d'une restriction du champ d'application de la méthode correspondante.

Simplification du fluide

Une première approche, assez rarement adoptée, consiste à constituer un modèle original, inspiré de diverses lois empruntées à la physique. Ainsi, Bashforth et Yang [7] composent un modèle personnel de dynamique des gaz pour simuler des explosions. La base du modèle développé est la prise en compte de particules de fluide virtuelles échangées entre cellules d'une discrétisation de l'espace. Ces particules permettent par suite de considérer le transfert de masse et d'énergie entre les cellules, sous l'action de différentes forces.

Ce modèle empirique est confronté à un ensemble de tests physiques réels, et éventuellement ajusté à l'aide de termes empiriques supplémentaires pour pallier des comportements jugés erronés. Du point de vue de la simulation elle-même, une notion d'adaptation spatiale est introduite, les cellules pouvant être regroupées au sein d'un *octree* et être de tailles variables. En outre, une adaptation du pas de temps est considérée pour augmenter la stabilité de la résolution, de même qu'une procédure de retour en arrière et de reprise automatique en cas de détection de divergence numérique de la simulation. Des objets mobiles peuvent être inclus dans la simulation en plus du fluide. L'interaction entre le fluide et les solides voxelisés est modélisée d'une part par l'application de forces dérivées de la pression sur l'objet, et d'autre part par l'obstruction des cellules de la grille occupée par les objets.

Malgré des résultats confirmant une certaine adéquation du modèle développé avec des phénomènes réels observables, cette approche n'offre pas toute satisfaction. En effet, si la recherche d'un modèle simplifié, cohérent et efficace est sans nul doute un objectif valable, l'approche empirique employée ne permet pas d'accorder une grande confiance aux résultats générés dans des situations arbitraires, non-confrontables à une réalité-témoin. Cependant, si une seule appréciation visuelle est considérée suffisante, ce modèle simple peut faire illusion.

Une alternative consiste à s'appuyer sur des modèles physiques existants. Une contrainte forte liée à l'emploi des équations de Navier-Stokes est l'incompressibilité du fluide, qui n'autorise pas de méthode de résolution simple. Witting [180] propose donc d'employer un modèle inspiré de la météorologie, où le fluide est artificiellement considéré comme compressible. Outre la vitesse du fluide et la pression, d'autres quantités sont simulées comme une température, mais les équations simulées suivent essentiellement le système suivant :

$$\begin{cases} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nu \Delta u - \nabla p + f \\ \frac{\partial p}{\partial t} = \nabla \cdot u \end{cases}$$

En employant cette simplification, la résolution devient à la fois beaucoup plus simple et beaucoup plus efficace que dans le cas d'une incompressibilité. En effet, la recherche d'une solution devient alors locale et ne nécessite plus la résolution d'un système linéaire mettant en jeu la totalité des points de discrétisation, comme dans les méthodes traitant un fluide incompressible décrites dans la section 2.3.2. L'introduction de la compressibilité est en fait équivalente à l'introduction d'une vitesse de propagation de l'information dans le milieu simulé. Le mode de résolution effectivement employé est un schéma combinant méthode de Runge-Kutta d'ordre 4 en temps, et différences finies centrées d'ordre 4 en espace.

Cette approximation de la modélisation du fluide est très intéressante pour l'application finale visée : l'animation réaliste de fluides dans un cadre non-photoréaliste. En effet, cette méthode employée dans un cadre bidimensionnel, est rapide tout en restant stable. Elle permet aisément d'advecter des images et se contrôle avec une relative facilité par l'imposition de conditions aux limites sur la vitesse ou la température. Les approximations employées dans le modèle du fluide n'induisent pas de comportements visiblement gênants, puisque dans un cadre non-photoréaliste l'utilisation des fluides reste essentiellement artistique et fortement tributaires des intentions des animateurs.

Approche linéaire multi-grilles Dans le cadre de fluides stationnaires très simplifiés, Weimer et Warren [175] proposent une approche originale et très efficace, inspirée des méthodes multi-grilles. L'objectif est de calculer le champ de vitesse sur une grille de résolution arbitraire, étant donné un premier champ imposé sur une grille très grossière qui fournit une forme de conditions aux limites. Suivant le même processus que pour les méthodes de subdivision de surface, une séquence de champs de plus en plus fins est générée par raffinement, en partant de la grille grossière de départ. Il faut souligner qu'il ne s'agit en aucun cas d'une simple forme d'interpolation, mais bien de la résolution des équations sous-jacentes à un modèle linéaire de fluide simplifié. À la limite de la subdivision, le champ généré satisfait aux équations continues.

La méthode exposée par Weimer et Warren permet de déterminer le schéma de subdivision à employer pour fournir des approximations de plus en plus denses. Les conditions aux limites appliquées correspondent aux valeurs prescrites de la grille la plus grossière et les séquences raffinées représentent les grilles obtenues par subdivision dyadique. La relation existant entre deux grilles successives est dérivée en considérant la notion d'opérateur de prédiction parfait.

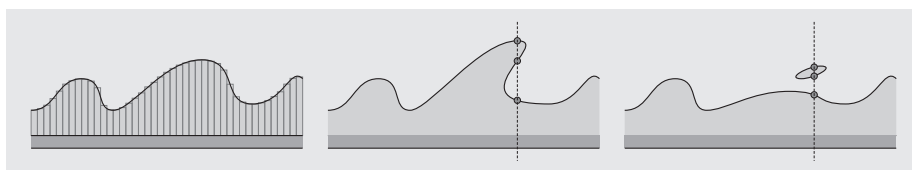


Figure 2.7: Limites de représentation des champs de hauteur. Seul le cas illustré à gauche est représentable, les deux autres présentant des lieux où la surface est multivaluée verticalement.

Contrairement au cas usuel de la subdivision de splines polynomiales, où le processus de subdivision est purement local, la subdivision induite est dans le cas considéré, globale. Afin de permettre néanmoins l'emploi d'une règle de subdivision locale efficace, les auteurs proposent alors de calculer des masques de calcul approximatifs qui exhibent des supports locaux. Les coefficients des masques sont, dans ce cas, calculés en employant un algorithme d'optimisation numérique. La précision du schéma de subdivision qui en résulte découle alors de la taille des masques employés.

Une telle technique est particulièrement intéressante puisqu'une fois le masque calculé, la génération du champ est très efficace : pour chaque point d'une grille raffinée, seul un voisinage local de la grille immédiatement moins raffinée est nécessaire au calcul. En outre, la conception d'un champ est relativement aisée puisque dérivant directement des valeurs fournies par l'utilisateur au niveau de résolution le plus faible. La dérivation d'un tel schéma n'en présente pas moins des limitations. Tout d'abord, les équations gouvernant le fluide doivent être linéaires, ce qui induit des simplifications sensibles sur le fluide. Ensuite, il n'est possible de considérer que des problèmes stationnaires, empêchant l'adéquation de la technique aux animations.

Simplification de la configuration de simulation

Dans le cadre de la simulation des liquides et non des gaz, l'emploi de modèles approximatifs se révèle délicat puisque la surface, qui est un élément-clé du fluide, peut être directement observée. Il est nécessaire d'en choisir un mode de représentation durant la simulation : ce choix relève alors d'un compromis entre facilité de représentation et contraintes imposées aux mouvements du fluide. Dans la mesure où ce choix s'accompagne d'une perte de représentativité, on peut aussi choisir d'employer un modèle de fluide simplifié.

Une forme de représentation naturelle souvent employée est celle d'un champ de hauteur, ce qui permet de considérer un problème de transport de fluide bidimensionnel. La simulation de la hauteur d'eau sur une grille autorise ensuite la construction triviale de la surface du liquide, pour émuler une étendue d'eau tridimensionnelle relativement calme. Toutefois, il faut souligner qu'une telle supposition, si elle permet de rendre les méthodes de simulations relativement simples, limite très fortement les mouvements du fluide : il n'est alors plus possible de former de vagues s'enroulant et se brisant, ou même de considérer des projections d'eau, comme l'illustre la figure 2.7.

Cette approche a été introduite par Kass et Miller [78] en infographie, en remplaçant les équations de Navier-Stokes par celles de propagation d'une onde.

Elles apparaissent par simplification des équations de propagation des vagues en milieu peu profond. L'équation finale employée permet donc d'exprimer la variation au cours du temps de la hauteur d'eau au dessus d'un terrain de profondeur variable.

La discrétisation du domaine employée est régulière, et le mode de résolution proposé en tire avantage en employant la méthode ADI — *Alternating-Direction Implicit* — qui consiste à substituer à la résolution d'un seul système global représentant le problème bidimensionnel, la résolution alternée de deux systèmes correspondant aux deux problèmes monodimensionnels qui découle de la séparation de l'opérateur Laplacien selon les deux axes de la discrétisation.

L'intérêt d'un tel schéma est qu'il est assez stable, puisqu'il emploie une méthode implicite, tout en restant performant, puisque le coût de résolution reste proportionnel au nombre de points discrétisant la surface. En effet, les systèmes linéaires mis en œuvre sont tridiagonaux, symétriques et limités au nombre de points d'échantillonnage dans une dimension. En complément de cette stratégie de résolution, une procédure de correction s'assure également que la conservation de volume est vérifiée, et qu'il n'est pas possible de générer des hauteurs de surface négatives.

Une approche similaire est proposée par O'Brien et Hodgins [115] : le liquide simulé est toujours considéré comme borné verticalement par une surface univaluée. La simulation mise en œuvre pour faire évoluer le système est fondée sur l'échange de fluide entre les différentes colonnes adjacentes. Celles-ci sont supposées être reliées par des tuyaux permettant l'échange de fluide à chaque pas de temps. Le débit de ces liaisons est alors simulé explicitement en tenant compte de l'accélération induite par la différence de pression entre les colonnes reliées par les tuyaux. La pression locale à chaque colonne est, elle, directement dépendante du volume de fluide contenu. Un tel modèle reste valide tant que cette dernière hypothèse du modèle est vérifiée, c'est-à-dire tant que le fluide reste lent et que la pression hydrostatique domine la pression dynamique. En outre, le transfert de fluide est limité par le volume effectif de fluide contenu dans les colonnes.

En fait, la méthode proposée correspond peu ou prou à une description algorithmique de la résolution de l'équation de propagation des ondes, en utilisant un algorithme de résolution n'employant qu'une seule itération. En effet, en considérant p la pression, v le volume de liquide dans une colonne, a l'accélération du débit dans un tuyau, et q le débit dans un tuyau, la méthode suit les relations de proportionnalité¹ suivantes :

$$p \propto v \quad a \propto \nabla p \quad \frac{\partial q}{\partial t} \propto a \quad \frac{\partial v}{\partial t} \propto \nabla \cdot q$$

donc :

$$\frac{\partial q}{\partial t} \propto \nabla v \quad \frac{\partial v}{\partial t} \propto \nabla \cdot q$$

et finalement :

$$\frac{\partial^2 v}{\partial t^2} \propto \Delta v$$

Toutefois, un tel modèle se trouve fortement limité, en l'état, par la représentation univaluée de la surface adoptée. En effet, seuls des mouvements de

¹La proportionnalité est dénotée par le symbole \propto .

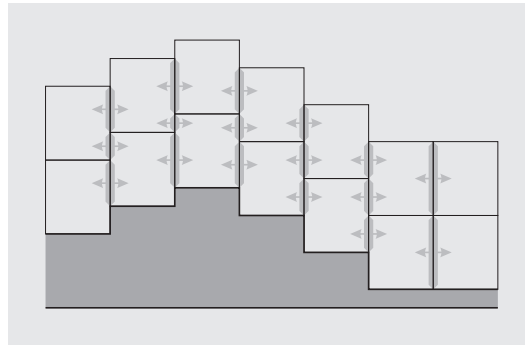


Figure 2.8: Division verticale des colonnes d'un champ de hauteur. Des interfaces verticales de taille variable apparaissent entre chaque paire de cellules contiguës horizontalement.

faible amplitude et traduisant une faible agitation du liquide sont simulables. Pour pallier cet inconvénient, il est proposé d'adjoindre à cette modélisation un système de particules. Celui-ci, à même de prendre en compte des topologies complexes, est employé pour modéliser la formation et l'évolution de gouttes se détachant de la composante principale du fluide. Pour cela, dès lors que la variation de hauteur d'une colonne excède un seuil prédéterminé, des particules sont générées et adoptent un mouvement balistique dont les paramètres initiaux sont dérivés de la vitesse locale du fluide au point de leur naissance. Une fois créées, ces particules n'interagissent pas entre elles, mais convoient néanmoins une quantité de fluide prélevée dans leur colonne de création et restituée dans la colonne où ces particules entrent ultérieurement en collision avec la surface du fluide.

Bien qu'incapable de remplacer une modélisation tridimensionnelle complète, ces particules se révèlent un substitut adéquat pour simuler de petites projections de liquide, notamment quand des objets sont introduits dans la simulation et engendrent des forces importantes sur le fluide lors de collisions avec la surface. Même si une forme d'interaction réelle entre objets et fluide est exprimée, les mouvements adoptables à la fois par le fluide et les objets restent limités, en raison de la nature fortement contrainte de la surface du fluide : seuls des objets flottants peuvent être introduits. En outre, le calcul de la force d'interaction reste dépendant des hypothèses simplificatrices concernant la configuration spatiale du fluide.

Cette dernière modélisation du fluide reste limitée à un terrain plat. Pour relaxer cette hypothèse, Mould et Yang [108] autorisent une représentation plus fine du fluide. Chaque colonne est éventuellement divisée en de multiples segments, donnant naissance à un ensemble de cellules, mais pas à une grille structurée, ni même à une grille conforme, comme l'illustre la figure 2.8 puisque le découpage vertical de chaque colonne est réalisé indépendamment. Des tuyaux sont ensuite considérés à chaque interface entre cellules de colonnes différentes. Cette représentation est adoptée pour permettre une variation de la topographie du lit de l'étendue de liquide.

L'algorithme gouvernant l'évolution du système est similaire à celui proposé par O'Brien et Hodgins, exception faite des transferts de fluide qui ne sont plus

réalisés entre colonnes mais entre cellules. La description verticale du fluide est donc affinée, puisque la vitesse horizontale du fluide peut varier avec la profondeur.

Le modèle de gestion de gouttes à base de système de particules est également amélioré, en autorisant la fragmentation stochastique des gouttelettes, et en affinant les conditions de leur création. La gestion des objets solides reste, elle, très similaire, une interaction limitée entre fluide et solides pouvant exister quand l'objet flotte à la surface du liquide, mais celle-ci étant remplacée par une simple action du fluide sur le solide dès lors que ce dernier est submergé. Il est en outre proposé d'adopter une gestion similaire des bulles présentes dans le liquide.

Si l'ensemble de ces modifications autorise un plus grand réalisme, la gestion de tous ces phénomènes périphériques à la simulation principale reste essentiellement empirique et fortement dépendante de la modélisation simplifiée adoptée pour le fluide. Néanmoins, malgré ses simplifications, ce modèle reste pertinent vis-à-vis de la physique, ce qui est démontré lors d'une phase de validation qualitative.

Sur la même base de simulation, Holmberg et Wünsche [70] proposent d'affiner la visualisation des résultats en reconstruisant une surface B-Spline à partir de la hauteur des différentes colonnes. En obtenant une représentation lisse et continue, il devient possible de concilier des simulations rapides, employant de petites grilles, et néanmoins affichées de manière suffisamment détaillée pour être plausibles sur le plan visuel.

Simulation par perturbation de modèle stationnaire Malgré les simplifications opérées sur les modèles de simulation, l'obtention de résultats par les méthodes précédemment décrites peut rester coûteuse, principalement en raison de la taille des grilles à manipuler pour traiter des environnements réalistes. Dans une situation particulière, celle de la simulation de ruisseaux, Neyret et Praizelin [112] proposent une approche alternative permettant de concilier un bon réalisme avec un coût de simulation modique. Plutôt que de chercher à simuler complètement le fluide lui-même comme dans les approches précédentes, une simulation physico-empirique de l'élément visuel le plus significatif, c'est-à-dire des perturbations visibles de la surface, est employée.

L'idée principale est de calculer et de capturer ces perturbations engendrées par l'environnement durant une phase de précalcul, et de restreindre la tâche effectuée durant l'affichage à une animation simple des perturbations capturées.

Les perturbations considérées, les ondes de choc générées en réponse à des obstacles dans le flux de liquide en mouvement, sont déterminées en usant d'un solveur Eulérien simple. Celui-ci simule alors un fluide stationnaire non visqueux et irrotationnel, *i.e.* un fluide potentiel, dans l'environnement donné. Ce dernier n'est appréhendé qu'en deux dimensions, la profondeur du flux étant supposée sans grandes variations. Les ondes de choc dans le fluide sont ensuite recherchées dans le champ de vitesse ainsi calculé. Étant donné la configuration particulière des ruisseaux, un certain nombre de connaissances hydrologiques sont *a priori* connues quant à la forme et aux points de naissance des ondes : elles s'apparentent à des courbes dont les trajectoires forment un angle connu avec le flux de fluide et prennent naissance sur des isolignes caractéristiques du champ de vitesse. Leur recherche ne présente donc pas de difficulté majeure.

Pour accroître le réalisme, des vagues supplémentaires sont ajoutées aux perturbations engendrées directement par les chocs. La construction de ces vagues secondaires s'appuie également sur les fronts détectés et sur le champ de vitesse calculé. Les perturbations capturées s'apparentent donc à un ensemble de courbes discrétisées dérivant dans le liquide.

Afin d'atteindre l'objectif d'animation interactive des ruisseaux, les perturbations de la surface du ruisseau sont animées directement. Pour cela, des perturbations localisées du champ de vitesse du fluide sont introduites aléatoirement dans le flux en amont du ruisseau et advectées par le flux précalculé. Ces perturbations sont modélisées comme des sources ou des vortex à supports compacts, dont les champs de vitesse sont incompressibles, pour se conformer aux hypothèses de fluide potentiel. Elles modifient directement les perturbations de la surface, capturées en altérant les courbes précalculées à leur passage. En raison de la configuration spécifique considérée, l'action de la perturbation peut être efficacement simulée par un processus de relaxation le long de la courbe, d'amont en aval.

Cette approche essentiellement phénoménologique permet ainsi l'obtention de résultats à la fois réalistes et interactifs, mais reste évidemment très spécifique. En effet, la majeure partie des hypothèses employées est propre à la configuration des ruisseaux, pour lesquels les perturbations de la surface sont assimilables à des phénomènes quasi-stationnaires pouvant être animés par une perturbation ultérieure. Dans une situation moins contrainte, il est peu probable de parvenir à caractériser aussi bien des éléments visuellement prépondérants et peu sujets aux variations au cours du temps.

Une autre situation très spécifique pour laquelle une méthode *ad-hoc* a été développée, est celle correspondant aux mouvements résultant de l'interaction d'un tissu avec un flux d'air. La technique proposée par Ling *et al.* [92] pour simuler cette configuration s'appuie sur la méthode des panneaux, et considère la pièce de tissu flottant dans un fluide potentiel, de vitesse connue.

Dans le cas de cette méthode, développée pour l'étude aérodynamique de profils alaires, l'intégralité du volume d'air environnant le tissu n'est pas explicitement simulé, ce qui donne lieu à une simulation drastiquement simplifiée. Dans la simulation, seule la surface de contact entre le tissu et le fluide, assimilable au tissu lui-même, ainsi que le sillage engendré, sont modélisés. Par conséquent, l'espace de simulation est intrinsèquement réduit d'une dimension par rapport à l'espace physique contenant les objets.

Dans la méthode proposée, la surface du tissu ainsi que celle du sillage sont discrétisés en un ensemble de panneaux, un vortex indépendant étant associé à chaque panneau. La méthode cherche ensuite à déterminer les différentes intensités des vortex requises pour satisfaire aux conditions aux limites existant à la surface du tissu. A cet effet, un système linéaire est assemblé, considérant la vitesse induite par chaque vortex d'intensité inconnue sur chaque panneau.

Les cas quasi-stationnaire et instationnaire sont traités par deux variantes similaires de cette technique. Les différences existant entre ces deux déclinaisons concernent la géométrie des vortex considérés, plus complexe dans le sillage pour le cas instationnaire, et la procédure employée pour calculer la force à appliquer au tissu en fonction des intensités des vortex. Une fois les forces calculées, elles sont appliquées à un modèle de solide élastique régissant le déplacement du tissu. Le choix d'un modèle ou de l'autre dépend d'un compromis entre vitesse et précision : le cas stationnaire est moins coûteux mais suppose évidemment

une simplification sévère des interactions réelles.

Dans la mesure où le calcul des intensités des vortex est directement dépendant de la géométrie du tissu, qui est elle-même influencée par les mouvements de l'air, une réelle interaction prend place entre le tissu et le fluide. La méthode présente également l'avantage de focaliser la simulation sur l'objet en mouvement, le tissu, en évitant de gaspiller des ressources pour simuler un grand volume de fluide sans intérêt pour l'animation. Reste qu'une telle méthode demeure extrêmement spécifique. En effet, elle suppose la configuration particulière d'un objet immergé unique en interaction avec un fluide possédant une vitesse uniforme à une certaine distance de lui.

Réduction dimensionnelle et extrapolation

Si pour des raisons de performance il n'est pas toujours souhaitable d'employer un modèle de simulation tridimensionnel complet, et si la perte de réalisme due à un modèle simplifié n'est pas tolérable, il est possible d'employer un modèle de simulation de dimension réduite mais considérant néanmoins un fluide modélisé par les équations de Navier-Stokes. La dimension ignorée peut ensuite être extrapolée en vue de retrouver des résultats tridimensionnels.

La simplification la plus drastique, introduite initialement par Chen et da Victoria Lobo [17], consiste à considérer, pour une étendue de liquide, une simulation des équations de Navier-Stokes bidimensionnelles, et à dériver la hauteur locale de la surface à partir de la pression. L'environnement de simulation considéré reste par ailleurs très simple, puisque le fond de l'étendue de liquide est supposé plan.

Le solveur employé considère une version pénalisée des équations de Navier-Stokes.

$$\begin{cases} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \nabla p + \frac{1}{Re} \Delta u \\ \epsilon p + \nabla \cdot u = 0 \end{cases}$$

Une discrétisation explicite de ces équations est opérée sur une grille décalée pour la vitesse et la pression. La contrainte d'incompressibilité est relaxée, ce qui rend la pression linéairement dépendante de la divergence locale du champ de vitesse. Les valeurs des deux variables à un pas de temps donné, sont donc directement calculables à partir de leurs valeurs au pas de temps précédent, sans qu'aucune résolution de système linéaire ne soit requise, la pression étant évaluée par l'entremise du paramètre de pénalisation ϵ choisi.

Cette pression locale est par la suite employée pour définir la hauteur de la surface du plan d'eau. En employant un véritable solveur, tout simpliste qu'il soit sur le point de l'incompressibilité, il est possible d'intégrer très facilement des conditions aux limites aussi bien internes qu'externes, pour simuler des environnements arbitraires. Des conditions évoluant au cours du temps pour simuler des obstacles ou des objets mobiles peuvent être introduites tout aussi aisément. Néanmoins, il n'est pas question dans ce dernier cas d'une interaction réelle entre objet externe et fluide, le mouvement de l'objet étant présupposé connu.

Ce mode de simulation a par la suite été étendu afin de pouvoir prendre en compte des environnements présentant un terrain avec des variations d'altitude [18]. Cette extension est réalisée en employant un modèle à deux niveaux. Dans

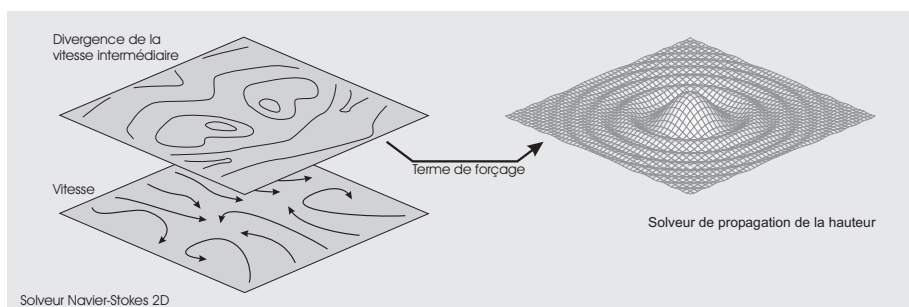


Figure 2.9: Couplage d'une simulation Navier-Stokes 2D par projection avec un modèle de propagation d'ondes gérant la surface d'un champ de hauteur. Un terme de forçage dépendant de la divergence du champ de vitesse intermédiaire est introduit dans le modèle de surface.

un premier temps, un algorithme fait s'écouler le liquide sur le terrain en considérant sa topographie. Cette phase, proche d'un algorithme de ligne de partage des eaux, permet de déterminer les différents bassins dans lesquels sont ensuite simulées des étendues de liquide à l'aide de la simulation précédemment décrite. Bien qu'à même de prendre en compte un environnement non plat, une telle technique ne saurait être assimilée à une véritable simulation tridimensionnelle. En effet, même si la première phase de l'algorithme s'assure de conserver globalement le volume du fluide s'épanchant sur le terrain, elle ne saurait simuler correctement les mouvements d'un liquide sur celui-ci.

Dans un objectif similaire de simulation rapide mais néanmoins plus réaliste de plans d'eau, Klein *et al.* [85] accolent à la résolution bidimensionnelle des équations de Navier-Stokes un modèle de propagation d'ondes pour gérer la surface du liquide, comme décrit dans la figure 2.9. La variation de hauteur de la surface du plan d'eau est simulée par un dérivé de la méthode introduite par Kass et Miller et décrite à la section 2.3.1, qui résout l'équation de propagation des ondes par la méthode ADI. Le reste du fluide est simulé en résolvant les équations bidimensionnelles de Navier-Stokes selon la méthode de projection de Chorin décrite dans la section 2.3.2. Contrairement à la technique précédente utilisant une pénalisation, ce solveur garantit que l'équation de continuité est tout à fait satisfaite, au prix toutefois de la résolution d'un système linéaire de Laplacien supplémentaire.

Dans cette méthode, il est supposé que les vagues sont sans influence notable sur le transport de fluide. En revanche, la composante principale du fluide est supposée influencer la formation et l'évolution des vagues. Un couplage monodirectionnel entre les modèles de surface et de composante principale du fluide est donc employé pour former la simulation complète. Pour cela, la hauteur du fluide, outre son évolution naturelle calculée en résolvant l'équation de propagation d'ondes, est perturbée par une quantité dérivée de la divergence du champ de vitesse intermédiaire calculé durant la résolution de l'équation de Navier-Stokes. La logique gouvernant cette heuristique est que cette divergence représente une variation potentielle de la quantité de fluide locale, et constitue donc un bon indicateur de la variation effective de hauteur de fluide.

En employant une telle forme de couplage, il est possible de combiner, à peu

de frais, la simulation d'un transport de matière réel dans la scène, grâce à la simulation des équations de Navier-Stokes, avec la simulation des effets typiques des vagues, telles leur réflexion ou leur réfraction en réaction à un terrain non plan, sans pour autant recourir à une simulation tridimensionnelle complète.

Extrapolation de champs de vitesse *a posteriori* Dans les techniques précédemment décrites, l'extrapolation prend place pour déduire la position de la surface d'un liquide, en présupposant la configuration globale de la scène. C'est cette hypothèse de configuration spatiale particulière qui permet de ne considérer qu'une simulation bidimensionnelle. Dans le cas d'un gaz, aucune surface n'est présente et aucune configuration particulière n'est inférable, la simulation devant être réellement tridimensionnelle pour générer des mouvements plausibles. Dans cette situation, pour simuler de larges explosions, Rasmussen *et al.* [127] proposent une technique permettant néanmoins d'opérer une réduction dimensionnelle et ainsi d'accélérer les calculs.

Plutôt que d'employer une très grande simulation tridimensionnelle, il est proposé de calculer une série de simulations bidimensionnelles indépendantes, et d'interpoler les résultats de ces simulations pour recouvrer la simulation désirée. Selon la configuration attendue du phénomène simulé, les coupes simulées peuvent être arrangées linéairement, cylindriquement, ou dans toute configuration plus appropriée. D'un point de vue pratique, les simulations des coupes sont réalisées en simulant les équations d'Euler pour un fluide incompressible, à l'aide de la méthode de projection de Chorin. Densité de fumée et température sont également simulées par une simple advection. Ce mode de simulation n'est cependant pas spécifique, la simulation des coupes pouvant résulter de tout type de simulateur.

Toutefois, la seule interpolation des résultats calculés ne saurait être suffisante, sous peine de ne créer qu'un mouvement bidimensionnel extrudé et interpolé, plongé dans un espace 3D, sans circulation de fluide entre les coupes adjacentes. Il est donc nécessaire de recréer la troisième dimension manquante du mouvement pour produire des résultats réalistes. Pour cela, un champ de vitesse aléatoire réellement tridimensionnel, créé par synthèse spectrale, est superposé au champ de vitesse interpolé, comme illustré dans la figure 2.10. Afin de conférer un certain réalisme au champ aléatoire, l'emploi d'un spectre de Kolmogorov lors de cette synthèse est recommandé. Ce champ aléatoire est également animé, afin de varier la turbulence du champ global résultant, et d'estomper au maximum le caractère interpolé des mouvements.

En opérant de la sorte, le volume de calculs est réduit de manière drastique puisqu'en lieu et place d'une très grande simulation 3D, seules quelques simulations 2D sont requises. Outre un gain de rapidité d'exécution, il est possible de dépasser largement, en termes de taille, une simulation pleinement tridimensionnelle grâce à une substantielle économie de mémoire. Les résultats obtenus, dans le cadre de la simulation d'explosions de grande envergure déplaçant d'importantes quantités de fumée, apparaissent largement plausibles en raison du faible couplage existant dans ce cas entre les deux échelles de mouvements présentes, très différentes. Dans cette situation, une turbulence stochastique de petite échelle est un bon modèle des déplacements locaux de la fumée, sans couplage notable sur les mouvements généraux, de grande ampleur. Une telle approche est cependant d'un usage plus délicat dans le cas d'un liquide à sur-

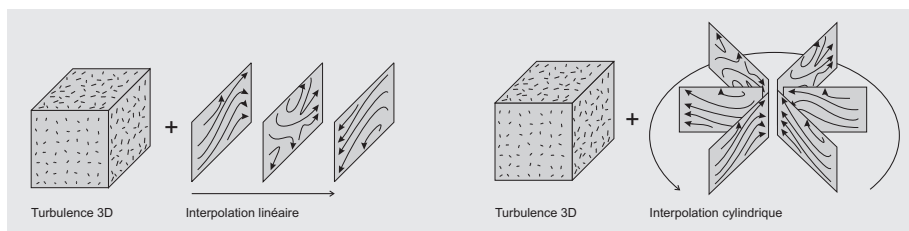


Figure 2.10: Interpolations linéaire et cylindrique de coupes bidimensionnelles et superposition d'une vitesse stochastique tridimensionnelle afin d'émuler la troisième dimension du mouvement.

face libre, cette dernière étant plus prompte à trahir la nature essentiellement bidimensionnelle des mouvements considérés.

2.3.2 Simulation physique non simplifiée

Afin d'obtenir le spectre de situations le plus étendu, il ne faut ni procéder à une simplification des équations modélisant le comportement du fluide, ni se limiter à une configuration géométrique de simulation particulière. Concernant l'aspect de modélisation, les équations de Navier-Stokes dans leur intégralité sont alors requises pour la simulation des liquides. Dans le cas des gaz, les équations d'Euler incompressibles, *i.e.* les équations de Navier-Stokes sans le terme de viscosité, peuvent également être employées. Dans cette dernière situation, l'hypothèse d'incompressibilité est ainsi souvent conservée, même si une réelle compressibilité est parfois considérée pour rendre compte des ondes de choc se propageant dans le milieu.

La différence entre gaz et liquides est plus marquée pour ce qui a trait à la gestion de la configuration de simulation. En effet, pour le cas des gaz, ou pour celui d'un liquide occupant la totalité de son environnement de simulation, seule la simulation du champ de vitesse en lui-même importe, afin d'advecter diverses quantités ou de déplacer des objets. La configuration de simulation est alors figée dans le temps et clairement prédéfinie, objets mobiles éventuels mis à part. En revanche, dans le cas usuel pour un liquide où une surface libre est simulée, la gestion de celle-ci devient complexe, puisqu'elle est par définition mobile et qu'elle fait varier les conditions aux limites à appliquer au cours du temps. Se pose alors également un problème de représentation quant à cette interface : il est évidemment souhaité que la représentation choisie ne restreigne pas la palette des situations simulables.

Cette différence drastique de situation explique pourquoi la simulation Eulérienne a été employée pour la simulation de gaz, dès 1984 par Kajiyama et von Herzen [76], alors que les premières simulations tridimensionnelles de liquide à surface libre ne sont apparues qu'en 1996 avec l'introduction de la méthode MAC par Foster et Metaxas [45].

Simulation 3D du fluide en formulation vitesse–pression

La simulation Eulérienne de fluides décrits par les équations de Navier-Stokes a réellement pris son essor en infographie avec l'introduction en 1996 par Fos-

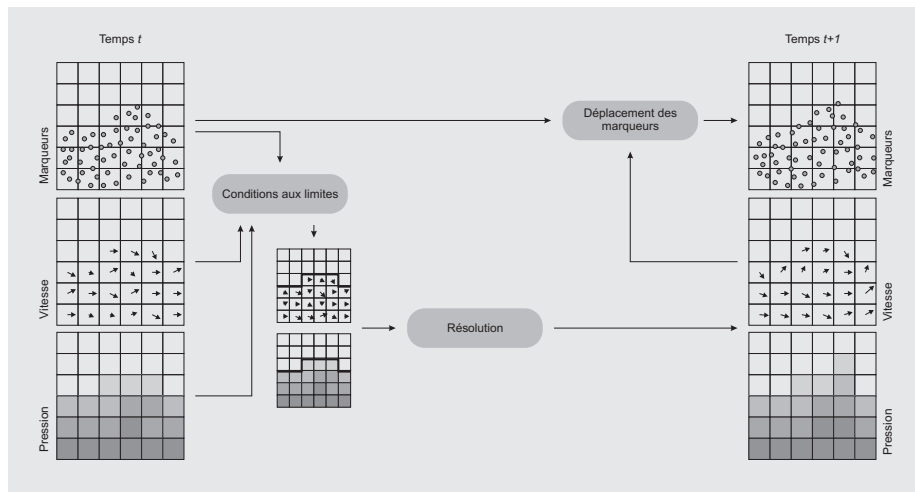


Figure 2.11: Principe de la méthode MAC. Des particules Lagrangiennes sont utilisées pour repérer le fluide dans une grille Eulerienne et poser des conditions aux limites à sa surface. Les vitesse et pression sont ensuite calculées par une simulation Eulerienne. Finalement, les particules sont advectées par le champ de vitesse calculé.

ter et Metaxas [45] de la méthode MAC — *Markers And Cells* — initialement développée par Harlow et Welch [64] dans le domaine de la physique en 1965. L'intérêt suscité par cette méthode, dont le déroulement est schématisé dans la figure 2.11, provient de sa capacité à représenter et à faire évoluer très simplement un liquide à surface libre au cours du temps. Pour ce faire, la méthode repose sur deux visions complémentaires du fluide : l'une Eulerienne, destinée à la simulation proprement dite du champ de vitesse et de pression, et l'autre Lagrangienne, uniquement destinée à permettre le repérage du fluide dans l'espace.

Ainsi, une grille parallélépipédique régulière fixe discrétise l'espace de simulation et définit un échantillonnage des champs de vitesse et de pression. Notons que celle-ci ne présente pas un arrangement collocated usuel des différentes variables mais un arrangement décalé, pour plus de stabilité. Sur cette grille, l'évolution temporelle est calculée par un schéma explicite en temps, employant une discrétisation par différences finies de l'équation de conservation de la quantité de mouvements, suivi par une procédure de relaxation de la divergence pour satisfaire à la contrainte d'incompressibilité du fluide. En considérant uniquement des obstacles dont les parois sont alignées avec les faces des voxels de la grille, des conditions aux limites physiquement significatives, comme la non-pénétration du fluide dans les obstacles, peuvent être aisément incorporées dans la simulation.

Toutefois, une telle méthode ne permet que la simulation de l'évolution temporelle des champs de vitesse et de pression et ne constitue qu'une moitié de la simulation d'un fluide à surface libre. La seconde partie de la simulation a pour rôle le suivi au cours du temps du fluide dans l'environnement. Afin de ne limiter en rien les mouvements simulables, une représentation du fluide par un nuage dense de marqueurs ponctuels est adoptée. Ces marqueurs, dénués de masse, sont advectés par le champ de vitesse calculé. Ils n'entrent pas directe-

ment dans le calcul de l'évolution du fluide, leur seul rôle étant de permettre la localisation du fluide dans l'espace.

L'interaction entre ces deux sous-parties du modèle est réalisée par le truchement des conditions aux limites découlant de la surface libre du liquide. Il devient en effet trivial de repérer les cellules de la grille contenant une partie de la surface du fluide puisqu'il s'agit des cellules contenant un marqueur ou plus, jouxtant au moins une cellule qui en est entièrement dépourvue. À chacune de ces cellules, il est possible d'affecter des conditions aux limites transitoires sur la vitesse, simulant ainsi la présence d'une surface libre. L'analyse du nuage de marqueurs pour la détection de la surface et l'imposition des conditions aux limites résultantes est réalisée à chaque pas de simulation. D'un certain point de vue, cette modélisation Lagrangienne peut s'apparenter à une augmentation artificielle de la résolution de la grille. Néanmoins, il convient de souligner que du point de vue de la composante Eulérienne du simulateur, la résolution visible du liquide reste celle définie par la grille.

Le choix de la représentation du liquide sous une forme Lagrangienne prend tout son sens lors de mouvements importants de la surface, puisqu'aucune intervention particulière n'est requise pour la gestion des modifications topologiques du liquide. Celui-ci peut subir des mouvements violents engendrant séparation en de multiples parties et reconnections, ou encore repli sur lui-même, sans que la représentation choisie ne s'y oppose, aussi bien dans des simulations bidimensionnelles que tridimensionnelles. De plus, la présence de sources de fluides ou puits dans l'environnement est alors aisément simulable en associant création ou destruction des marqueurs à l'imposition de conditions aux limites sur la vitesse en ces lieux.

Malgré leur couplage, les deux composantes de ce modèle restent indépendantes et dédiées à la résolution de deux problématiques relativement distinctes : la simulation des mouvements d'une part, et le repérage du liquide d'autre part. Pour preuve, dans le cadre de la simulation de fumées, Foster et Metaxas [47] n'emploient que la composante Eulérienne de ce modèle afin de calculer le mouvement de fumées. L'environnement est alors supposé rempli d'un gaz et la détermination d'une surface n'est plus requise. En revanche, la température du milieu est simulée, en plus de la vitesse et de la pression du gaz porteur, afin d'apporter plus de réalisme au calcul de la flottaison engendrée par la variation de température. Cette simulation de la température reste simple à mettre en œuvre puisque très similaire à celle du fluide, les deux comprenant des composantes diffusives et convectives.

Ce mode de simulation permet au final, de manière totalement automatique, l'obtention de simulations extrêmement variées et réalistes, dans des environnements arbitraires. Mais la simulation reste relativement coûteuse, car la résolution exige des ressources substantielles, en raison de l'emploi d'une ou plusieurs grilles tridimensionnelles stockant les variables de la simulation. Par ailleurs, l'emploi d'un schéma d'intégration explicite restreint fortement la taille des pas de temps garantissant une simulation stable. La borne de stabilité est en effet proportionnelle au pas de la grille, ainsi qu'à la vitesse maximale adoptée par le fluide dans l'intégralité du domaine de simulation. Cette double dépendance signifie que l'emploi de grilles de taille importante se révèle rapidement très coûteux.

Algorithme de simulation stable Pour pallier ce problème, Stam [149, 150, 151, 153] propose un algorithme de simulation inconditionnellement stable. Il est alors possible de choisir des pas de temps arbitrairement grands, ce qui ouvre, même pour de petits domaines, la possibilité de réaliser des simulations interactives.

Cette méthode constituant en partie la base du simulateur auquel une extension est apportée dans cette thèse, seul un aperçu en est donné ici, une explication détaillée étant disponible dans la section 3.1.5.

L'algorithme proposé s'appuie sur une méthode de projection décrite initialement par Chorin [24] en 1968 pour séparer la simulation d'un pas de temps $u^t \rightarrow u^{t+1}$ en deux phases : la première fait évoluer le champ de vitesse en accord avec l'équation de conservation de quantité de mouvement, et la seconde projette ce champ mis à jour dans l'espace des fonctions solénoïdales pour satisfaire à la contrainte d'incompressibilité, à l'aide de la résolution d'une équation de Poisson :

$$\nabla \cdot u^t = 0 \xrightarrow{\text{accélération}} \nabla \cdot u^* \neq 0 \xrightarrow{\text{projection}} \nabla \cdot u^{t+1} = 0$$

Contrairement à la technique décrite précédemment, les termes de l'équation de conservation de quantité de mouvement ne sont pas tous simulés en une seule opération. Au contraire, l'influence de chaque terme de l'équation est résolu indépendamment, séquentiellement, et de manière optimale : les termes externes, de diffusion et de convection sont alors résolus respectivement par intégration d'Euler, par une méthode implicite, et par une advection semi-Lagrangienne. Le terme lié à la pression est, quant à lui, ignoré puisque l'étape de projection a pour corollaire de la faire disparaître.

L'emploi de méthodes de résolution stables pour chacun des termes permet ainsi d'obtenir une méthode globalement robuste. En outre, dans le cas d'un domaine partout périodique, un mode de résolution alternatif, plus compact mais pas nécessairement plus rapide, est proposé. Il agit toujours selon le même principe, mais tire parti de transformées de Fourier pour résoudre plus simplement les étapes de diffusion et de projection, dans le domaine fréquentiel.

Dans tous les cas, le solveur obtenu reste suffisamment stable pour permettre une simulation interactive lorsque de petits domaines sont utilisés. Selon le degré désiré, il est également possible de varier la précision de résolution des systèmes linéaires employés. Il est par exemple proposé de n'employer que quelques itérations d'une relaxation de Gauss-Seidel, lorsque seule une apparence globale de réalisme est visée sans chercher des résultats précis [153].

Malheureusement, et même si des techniques de résolution précises sont employées, cette méthode introduit, de par l'usage d'une advection semi-Lagrangienne, une forte dissipation numérique, qu'il n'est pas possible de supprimer. Toutefois, le résultat reste généralement plus que satisfaisant pour l'exploitation visuelle opérée en infographie. À noter que le solveur ainsi défini est immédiatement capable de traiter de la même manière la simulation d'équations d'advection et de diffusion de composés supplémentaires dans le fluide. Ceci signifie que la simulation de fumée, par exemple, est possible sans extension particulière du simulateur. Il en est de même pour la déformation d'une image soumise aux mouvements du fluide. De par ses qualités d'efficacité et de stabilité, cette technique, souvent référencée sous le nom de « *Stable Fluids* », constitue actuellement la base de simulation la plus couramment employée, sous cette forme,

ou avec de légères variantes pour ce qui est du traitement des différents termes de l'équation de conservation de quantité de mouvement.

Méthodes hybrides Quoique très peu employées en infographie, d'autres méthodes hybrides, associant grille et particules, existent en physique. Dans le but de simuler du sable comme un fluide, Zhu et Bridson [189] proposent ainsi d'employer les méthodes PIC —*Particle In Cell*— ou FLIP —*FLuid Implicit Particle*— Dans la méthode MAC, les marqueurs sont uniquement chargés d'identifier la position du fluide, sans prendre part au calcul totalement réalisé sur une grille. Dans les méthodes PIC ou FLIP, le fluide est principalement matérialisé par les particules, et celles-ci interviennent effectivement dans le calcul pour gérer la composante advective des équations.

Dans la méthode PIC, à chaque pas de temps, la grille est réinitialisée par transfert des valeurs portées par les particules, puis devient support de la simulation de toute l'équation, exceptée la composante advective. Après ce calcul, les valeurs portées par les particules sont interpolées à partir de la grille ayant évolué, puis les particules sont convectées par le fluide, gérant ainsi la composante convective. La méthode FLIP suit une démarche similaire, mais permet de limiter très fortement la dissipation numérique, en n'employant la grille que pour évaluer les variations des quantités portées par les particules, et non les quantités elles-mêmes.

Simulation de fluide compressible Si l'hypothèse d'incompressibilité du fluide est toujours vérifiée pour les liquides, et souvent admissible dans le cas de gaz se déplaçant à faible vitesse, elle s'avère inexacte dans le cas de gaz subissant des mouvements très violents. Pour traiter cette situation, particulièrement dans le cadre de la simulation d'explosions, Yngve *et al.* [185] emploient une simulation de gaz compressibles, à même de mieux gérer les forts gradients de pression générés par la situation.

La technique proposée repose sur une discrétisation Eulérienne à l'aide d'une grille régulière des équations de Navier-Stokes compressibles, enrichies d'une équation matérialisant les transferts d'énergie. L'évolution temporelle est réalisée suivant un schéma particulier où les composantes convectives et non-convectives sont prises en compte séparément, pour plus de stabilité. En outre, la simulation emploie une méthode de limitation du flux lors du calcul des composantes convectives de la masse et de l'énergie du fluide. Cette technique, permettant de stabiliser la simulation, consiste à pondérer les quantités transférées lors du déplacement du fluide par les quantités locales effectives. Ceci signifie qu'à vitesse égale, la masse transférée par une cellule contenant peu de fluide sera faible, alors que pour une cellule en contenant beaucoup, elle sera importante, ce qui tend à éviter toute divergence.

En employant une telle technique, il devient possible de simuler correctement la propagation dans l'environnement des ondes de choc résultant d'une explosion, au moins visuellement. Cette technique est également couplée à une réelle interaction avec des solides, à l'aide d'une procédure détaillée dans la section 2.3.4.

Gestion de la turbulence et vorticit 

Un probl me soulev  par les m thodes de simulation pr c demment d crites, et particuli rement par toutes celles pr sentant une parent  avec la m thode *Stable Fluids* est la pr sence d'une dissipation num rique forte. Souvent involontaire et purement num rique, elle peut parfois  tre sciemment ajout e pour stabiliser une m thode. Si elle se r v le rarement g nante du point de vue quantitatif, en g n ral n glig  en infographie, elle se manifeste  galement sur le plan visuel, par un fluide d'apparence tr s visqueuse, dont l'amplitude des mouvements d cro t rapidement. Cette situation est particuli rement visible dans le cas de liquides   surfaces libres ou de fum es tourbillonnantes, la dissipation excessive rendant donc difficile l'obtention de s quences r alistes de longue dur e.

Traitement du terme advectif Ce terme fait l'objet d'une attention particuli re car il est sans nul doute le plus prompt   engendrer une dissipation num rique parasite. De mani re plus g n rale, ces m thodes peuvent d'ailleurs s'appliquer au transport de n'importe quelle quantit  dans le domaine de simulation, et pas uniquement au transport de la vitesse. Outre un traitement par discr tisation centr e [45] ou un traitement par caract ristiques [149], on trouve l'emploi d'un traitement par diff renciation vers l'amont [44], ou par l'utilisation d'une m thode de Lax-Wendroff [38].

Confinement de la vorticit  Pour pallier le ph nom ne d'att nuation, Fedkiw *et al.* [39] proposent d'employer une technique de confinement de vorticit . Cette technique, invent e relativement r cemment en physique par Steinhoff et Underhill [157] a pour but le renforcement coh rent des tourbillons du fluide, afin de combattre toute dissipation excessive.

Pour cela, une nouvelle force externe est appliqu e au fluide, afin d'amplifier la vorticit  existante. Cette force est elle-m me directement calcul e,   partir de la vorticit  d j  pr sente dans le champ, pour ne pas cr er de turbulences artificielles, mais pour renforcer celle d j  pr sente. Le calcul effectif de la force f_{conf} est tr s simple et purement local.

$$\omega = \nabla \times u \quad \eta = |\omega| \quad n = \frac{\nabla \eta}{|\nabla \eta|} \quad f_{conf} = \epsilon h n \times \omega$$

La force est perpendiculaire   la fois au tourbillon ω et au gradient de la norme du tourbillon η . Elle est aussi directement proportionnelle au tourbillon. Dans la mesure o  cette force reste artificielle, elle est contr l e par un param tre ϵ laiss    la discr tion de l'utilisateur. La d pendance de la force au pas de la grille h est destin e   fournir une approximation coh rente lorsque la grille est raffin e.

Si cette technique permet l'amplification de la vorticit  existante dans une simulation, elle ne se r v le toutefois pas n cessairement suffisante pour propager la turbulence au sein de l'environnement. Dans ce but, Selle *et al.* [136] pr sentent une approche plus Lagrangienne, employant des particules, pour assurer une propagation effective de la vorticit .

L'approche propos e consiste, en sus de la simulation Eul rienne de base manipulant vitesse et pression,   simuler s par ment la vorticit .

$$\frac{\partial \omega}{\partial t} + (u \cdot \nabla) \omega - (\omega \cdot \nabla) u = \nu \Delta \omega + \nabla \times f$$

Toutefois, plutôt que d'adopter une simulation Eulérienne pour cette dernière, une approche Lagrangienne séparée est mise à profit pour éviter les problèmes de dissipation numérique. La simulation de la vorticit   ne saurait cependant   tre compl  tement d  coupl  e de celle du champ de vitesse puisqu'elle requiert la connaissance de ce dernier. En connaissant le champ de vitesse du fluide, l'  volution de la vorticit   port  e par chaque particule est alors relativement simple    calculer, puisqu'elle se ram  ne essentiellement    la r  solution d'une   quation diff  rentielle ordinaire ind  pendante pour chaque particule.

   chaque pas de temps, un champ global de vorticit   est alors reconstituable dans tout l'espace    partir des particules de vorticit  . Deux approches sont possibles pour r  introduire cette vorticit   dans la simulation Eul  rienne. La plus simple consiste    pond  rer la force de confinement de vorticit   d  crite pr  c  demment par la magnitude du champ de vorticit   d  riv   des particules. La deuxi  me approche consiste    tirer parti de toute l'information de vorticit   calcul  e, et d'en tirer directement et analytiquement la force de confinement. Chaque particule porteuse de vorticit   engendre alors autour d'elle une force de confinement qui lui est propre.

En employant cette technique Lagrangienne, la vorticit   est efficacement transport  e dans tout l'environnement, et la turbulence semble se propager naturellement. Afin d'assurer une simulation coh  rente, les particules sont introduites aux sources de fluide, ou    proximit   des objets pour simuler une interaction turbulente du fluide avec ceux-ci. Une telle technique   quivaut alors    une forme de mod  lisation de ph  nom  nes physiques plus complexes.

Mod  lisation explicite de la turbulence L'introduction de la force de confinement de vorticit   dans la simulation trouve en partie sa raison d'  tre dans la mod  lisation des effets des ph  nom  nes d'  chelles trop fines pour   tre r  solus par une grille fixe. Dans le cadre de la simulation d'une combustion, Barrero *et al.* [4, 5], envisagent une autre approche pour capturer les multiples   chelles de ce ph  nom  ne complexe. La logique gouvernant l'approche propos  e est que le comportement du fluide est essentiellement autosimilaire    de multiples   chelles.

Une simulation multi-  chelle hi  rarchique est donc propos  e, o   l'environnement de simulation, au lieu d'  tre appr  hend   par une grille r  guli  re, l'est par un *octree*. Au sein de chaque cellule de l'*octree*, le champ est suppos     tre la superposition d'une composante d  terministe et d'une composante simul  e, l'une ou l'autre pouvant   ventuellement   tre nulle, et l'une ou l'autre pouvant   ventuellement   tre d  finie comme le r  sultat d'une autre simulation    plus petite   chelle. La simulation proc  de hi  rarchiquement, en employant les r  sultats de la simulation    un niveau donn   comme conditions aux limites de la simulation aux niveaux suivants. Apr  s une r  solution successive    tous les niveaux, le champ est int  gralement connu dans tout l'espace et    toutes les   chelles.

Dans ce contexte, la turbulence est g  r  e par deux approches diff  rentes. La premi  re consiste    l'introduire    l'aide des champs d  terministes en consid  rant un atlas de ph  nom  nes turbulents pr   tablis. La turbulence est alors choisie par l'utilisateur et exprim  e comme combinaison de ces primitives d  terministes, et ce    diff  rentes   chelles. La deuxi  me approche emploie une forme auto-similaire des   quations de Navier-Stokes pour mod  liser le comportement du fluide   

une échelle plus fine que celle des cellules de la grille. Lors de la simulation, cette modélisation est employée à chaque feuille de l'arbre, afin d'approximer le comportement du fluide plus finement que la résolution maximale employée.

Une telle approche présente l'intérêt de pouvoir combiner, dans un seul modèle, une gestion manuelle de la turbulence, essentiellement utilisée à des fins de modélisation, et une gestion algorithmique de la turbulence, destinée à une simulation moins dissipative et plus réaliste.

Une autre gestion de la turbulence est celle proposée par Yoshida et Nishita [187] dans le cadre de la simulation de fumée dans l'atmosphère. Le champ de vitesse du fluide est considéré comme un champ moyen complété par une composante turbulente, et l'évolution du fluide est déduite de l'équation de Reynolds, qui découle des équations de Navier-Stokes mais incorpore directement les effets de la composante aléatoire. Par ailleurs, il est proposé d'employer le modèle de turbulence $k - \epsilon$, plus stable qu'une simulation directe de la turbulence. Ce modèle décrit d'une part l'équilibre entre dissipation et production énergétique au sein du fluide, et d'autre part sa diffusion turbulente. Pour accroître encore la turbulence du fluide, il est proposé d'introduire des vortex Lagrangiens créés aléatoirement et advectés par le fluide dans la simulation.

Simulation du fluide par la vortacité

Les utilisations précédemment décrites de la vortacité consistent principalement en une amplification, ou un transport, de la turbulence. Plutôt que de ne mettre à profit cette quantité qu'en complément d'une simulation employant principalement la vitesse comme variable, il est possible de la placer au centre la simulation, et de la considérer comme variable principale. La vitesse est ensuite dérivée de la vortacité, directement ou par l'intermédiaire de la fonction de courant. Plus de détails quant à cette formulation des équations de Navier-Stokes sont disponibles dans la section 1.2.4 qui énonce le double intérêt de cette approche : la simulation est simplifiée par l'absence de la pression, et, dans le cas bidimensionnel, la simulation vectorielle est remplacée par une simulation scalaire. En revanche, le traitement des conditions aux limites peut se révéler moins intuitif.

Modélisation bidimensionnelle Paradoxalement, cette approche quelque peu détournée de la simulation d'un fluide fut l'une des premières proposées en infographie, par Yaeger *et al.* [184], pour simuler l'atmosphère de la planète Jupiter. Dans la technique proposée, deux approximations sont effectuées pour rendre la simulation plus simple : le domaine de simulation est considéré comme bidimensionnel cylindrique *i.e.* périodique sur une dimension, et non comme sphérique, et une simplification est opérée dans le modèle d'évolution de la vortacité afin de stabiliser la méthode. Les forces considérées dans l'évolution du système correspondent alors à la viscosité ainsi qu'à la force de Coriolis. Une méthode de résolution pseudospectrale est employée pour la résolution effective des équations. Ce choix engendre une résolution simple dans le domaine fréquentiel, tirant profit des conditions aux limites choisies. La simulation reste efficace en faisant usage de transformées de Fourier rapides pour opérer la conversion de, et vers, le domaine fréquentiel. L'évolution en temps est prise en charge par une méthode de type Adams-Bashforth.

L'emploi d'une méthode en vorticit   pose le probl  me de la d  termination des conditions initiales. En effet, un champ de vorticit   ne pr  sente pas d'interpr  tation intuitive, et sa cr  ation *ex-nihilo* se r  v  le compliqu  e. En m  me temps, ce champ initial poss  de une importance cruciale puisqu'il conditionne largement l'animation r  sultante. Afin de conserver une certaine plausibilit  , une technique tirant partie des photographies disponibles de la surface de Jupiter, ult  rieurement advect  es durant l'animation, a   t   employ  e. Tout d'abord, un profil de vitesse horizontal moyen est d  termin   manuellement.    la vorticit   ainsi induite est superpos  e une vorticit   de petite   chelle, d  termin  e manuellement en prenant pour base les vortex visibles sur les clich  s disponibles. Ces vortex sont suppos  s elliptiques et leurs param  tres de taille et de puissance sont extraits manuellement. Sans se r  v  ler quantitativement exact, ce champ initial se r  v  le n  anmoins visuellement compatible avec l'advection ult  rieure du clich   employ   pour sa d  termination. Le travail de construction de ce champ initial reste toutefois laborieux puisque plusieurs centaines, voire plus d'un millier, de vortex sont extraits manuellement pour construire un champ satisfaisant.

Plut  t que d'employer une r  solution int  gralement sur grille, comme la m  thode pr  c  dente, Gamito *et al.* [50] proposent, toujours pour une simulation bidimensionnelle, une m  thode hybride : le transport de la vorticit   est assur   par des particules convoyant directement la vorticit  , mais la reconstruction du champ de vitesse *via* la fonction de courant passe par une grille. L'objectif de ce transport Lagrangien est une plus grande efficacit  , le recours    un ensemble restreint de particules porteuses r  duisant sensiblement le nombre de points de discr  tisation pour lesquels la vorticit   doit   tre simul  e. En effet, il est raisonnable de supposer que les particules ont tendance    se concentrer naturellement dans les r  gions o   la dynamique est la plus turbulente, r  partissant ainsi plus efficacement les ressources dans les zones les plus pertinentes de la simulation.

En raison de la vision Lagrangienne adopt  e, la simulation de la vorticit   ω port  e par les particules mobiles prend une forme tr  s simple.

$$\frac{d\omega}{dt} = \nu \Delta \omega \quad \frac{dx_i}{dt} = u(x_i, t)$$

Les particules de positions x_i sont advect  es par le champ de vitesse, et la vorticit   n'est plus affect  e que par un terme diffusif. Ce dernier n'est pas simul   directement mais plut  t   mul   par une proc  dure stochastique, perturbant    chaque pas de temps la position des particules de mani  re al  atoire.

Une fois les particules mises    jour, un champ de vorticit   global est reconstruit, chaque particule contribuant aux points-  chantillons de la grille les plus proches. La fonction de courant est ensuite d  duite de cette vorticit   en r  solvant une   quation de Poisson avec des conditions aux limites homog  nes, *i.e.* aucun flux n'est autoris      travers les bords du domaine. De la fonction de courant est ensuite tir   le champ de vitesse.

Ce mode de simulation est employ   afin de simuler un panache de fum  e bi-dimensionnel. Un mod  le de jet est utilis  , o   des particules de vorticit   contrarotatives sont cr  ees de part et d'autre de la base du jet. La vorticit   qui leur est initialement assign  e est indirectement fonction de la vitesse d'  jection. Comme dans les autres m  thodes impliquant une formulation en vorticit  , la d  termination des conditions aux limites reste d  licate. Si dans cette configuration extr  mement simple et contrainte, ce travail est possible, il reste toutefois d  licat    effectuer dans un cas plus g  n  ral.

Cette méthode procède d'une démarche similaire à la méthode purement Lagrangienne proposée par Park et Kim [118], et décrite dans la section 2.2.4. Dans les deux cas, la vorticit  est simul e en recourant   des particules porteuses puis un champ de vitesse est reconstruit. La diff rence se situe dans cette reconstruction, directement   partir des particules en utilisant un noyau de Biot-Savart pour Park et Kim, et par l'interm diaire d'une grille et de la d termination explicite de la fonction de courant dans le cas de Gamito *et al.*. En revanche, elle est plus  loign e de la m thode propos e par Selle *et al.* [136], d crite dans la section 2.3.2, qui simule la vorticit  avec des particules pour d terminer ensuite un terme de confinement de vorticit  dans une simulation Eul rienne usuelle, et qui ne fait pas intervenir la notion de fonction de courant.

Mod lisation tridimensionnelle Une derni re approche s'appuyant sur une formulation en vorticit  et sur une r solution Eul rienne est propos e par Elcott *et al.* [34]. Elle appara t comme particuli rement int ressante en raison de ses capacit s   op rer sur des maillages simpliciaux arbitraires, et   produire des r sultats d nu s de dissipation parasite.

Pour cela, elle s'appuie sur l'association d'une manipulation de quantit s int gr es et sur le cadre de l'analyse ext rieure discr te. Une telle approche permet un traitement   la fois ais  et rigoureux des quantit s diff rentielles n cessaires   la r solution du probl me, alors d finies en tenant compte de la topologie du maillage sous-jacent. La manipulation effective de ces quantit s d coule directement de la subdivision topologique en simplexes des maillages manipul s,   l'aide de deux op rateurs : la d riv e ext rieure, et l' toile de Hodge, qui d rivent tr s simplement des simplexes de diff rentes dimensions du maillage consid r . En pratique, la d riv e ext rieure est repr sentable par une s rie de matrices d'adjacence encodant les relations d'adjacence entre les diff rentes entit s topologiques du maillage. L' toile de Hodge diagonale choisie est assimilable   un vecteur dont les entr es d notent une certaine m trique pour les simplexes du maillage.

En utilisant ces deux op rateurs, il devient possible de consid rer une discr tisation des op rateurs diff rentiels continus, parfaitement compatible avec le maillage, c'est- -dire ne s'appuyant que sur des quantit s port es par le maillage. De plus, en manipulant le fluide par son flux   travers les faces, *i.e.* une quantit  int gr e, il est possible de produire un algorithme d'advection de la vorticit  non dissipatif. L'algorithme de r solution global proc de classiquement, en simulant l' volution de la vorticit  par une advection et une  ventuelle diffusion, puis en extrayant le champ de vitesse de la fonction courant d riv e de la vorticit  alors calcul e.

Si le cadre math matique de d rivation de cette m thode reste complexe, l'algorithme   mettre en  uvre n'est toutefois pas plus d licate que pour d'autres solveurs Eul riens. En outre, elle se r v le peu c teuse   l'utilisation, et capable de traiter sans dissipation parasite des fluides dans des domaines arbitraires. Dans le cadre de cette m thode, l'imposition de conditions aux limites sur les flux au bord du domaine reste possible directement, ou sans grande modification.

Traitement de la surface pour le cas des liquides

Dans le traitement de liquides présentant une surface libre, la simulation de cette surface est un élément prépondérant du réalisme, pour la simulation numérique proprement dite, et, dans le cas de l'infographie, pour la visualisation réaliste. Du point de vue de la simulation, la surface doit être définie précisément, et évoluer d'une manière suffisamment contrôlée, pour assurer une bonne conservation de la masse du fluide. Une définition précise permet également une prise en compte de phénomènes d'interface fins, telle la tension de surface. Du point de vue de la production d'images, la surface issue de la simulation doit présenter une apparence très lisse mais doit également être capable d'exhiber des détails fins, tout en étant susceptible de subir des modifications topologiques importantes, pour présenter une évolution réaliste.

En raison des contraintes liées à ces modifications topologiques importantes, les solutions tirent généralement partie des surfaces implicites. Une solution directe, dans le cadre de simulations MAC, est de reconstruire une surface implicite *a posteriori*, indépendamment pour chaque pas de temps, en tirant avantage des marqueurs comme squelette. En suivant cette approche, l'obtention d'une surface lisse et temporellement cohérente tout au long de l'animation reste toutefois délicate, un bruit résiduel restant généralement présent en raison du manque d'homogénéité locale des marqueurs.

Traitement par *Levelsets* Afin d'obtenir des résultats de bien meilleure qualité, Foster et Fedkiw [44] proposent d'intégrer directement l'évolution de la surface à la simulation, en employant une technique fondée sur les *levelset* introduits par Osher et Sethian [137]. L'idée de la méthode consiste à associer la surface du fluide à l'isocontour $\phi = 0$ d'une fonction scalaire $\phi(x, t)$ représentant la distance à la surface, définie sur l'intégralité du domaine spatial et dans le temps. Cette fonction est advectée par le champ de vitesse du fluide, ce qui se traduit par le problème de transport suivant :

$$\frac{\partial \phi}{\partial t} + (u \cdot \nabla) \phi = 0 \quad \phi(x, 0) = \phi_0(x)$$

Le champ initial ϕ_0 est défini comme la distance signée à la surface initiale du fluide extraite du nuage de marqueurs, et lissée. À tout moment, la surface reste définie par l'isocontour $\phi = 0$, comme illustré dans la figure 2.12. Les modifications topologiques que peut subir le fluide sont alors automatiquement prises en compte par cette reformulation.

Ce problème est simulé de manière Eulérienne, sur une grille fine, à l'aide d'une méthode de différenciation en amont. Ce choix, plutôt que l'emploi d'une méthode semi-Lagrangienne, est dicté par la volonté de limiter au maximum la dissipation numérique affectant la position de la surface. Toutefois, malgré cette procédure, une perte de volume parasite est encore notable. Pour la combattre, une procédure de correction de la position de la surface, illustrée dans la figure 2.13, est adjointe à la méthode : les marqueurs du fluide continuent à être simulés en parallèle, et sont autorisés à rectifier la valeur de ϕ , en repoussant la position de la surface vers l'extérieur du fluide si nécessaire, dans les zones de forte courbure de la surface. L'idée est que dans ces zones les particules sont de meilleurs indicateurs de la position, au contraire des zones plus lisses, où l'isocontour brut est plus fiable. Afin de produire une évolution de qualité de la

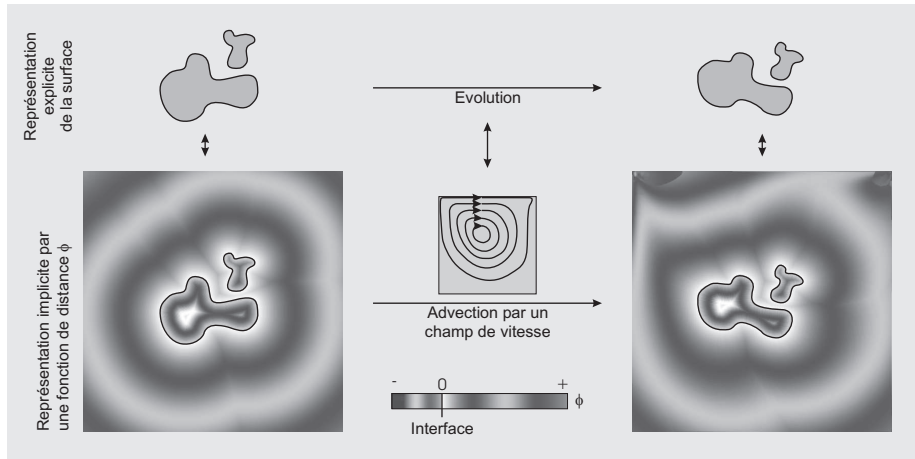


Figure 2.12: Évolution de la surface par *levelset*. La surface est représentée par une fonction scalaire $\phi(x, t)$ matérialisant la distance à la surface. Son évolution se ramène alors au problème d'advection de ϕ sous l'influence du champ de vitesse de fluide.

surface, la résolution de l'équation de transport de la surface est effectuée avec un pas de temps plus fin que celui employé pour la simulation du champ de vitesse. Périodiquement, la fonction ϕ peut également être réinitialisée comme la distance signée à la surface courante, pour éviter qu'elle ne présente trop de distorsions.

En usant de cette méthode, il est alors possible de simuler une surface restant visuellement lisse au cours du temps, grâce à la modélisation implicite, et ne souffrant pas d'une dissipation de masse excessive, grâce à la complémentarité des marqueurs. Cette technique hybride ouvre ainsi la voie à un rendu animé de liquides de grande qualité.

Cette idée est étendue par Enright *et al.* [36], dans le but de limiter encore plus les déviations importunes de la surface, et donne lieu à la méthode « *Particle Levelset* ». La correction précédemment proposée est conçue pour limiter la disparition inappropriée de fluide en repoussant la surface du fluide vers l'extérieur de celui-ci. Cependant, elle ne peut juguler une expansion non désirée de celle-ci. Dans l'extension proposée, de nouvelles particules de correction sont introduites à l'extérieur du fluide, et jouent un rôle de correction parfaitement analogue, mais cette fois dans le but de limiter ces expansions parasites. La surface du fluide se trouve donc contrainte des deux côtés simultanément, comme l'illustre la figure 2.13.

Pour mettre en œuvre cette technique, une vitesse de convection de ces particules externes doit être définie, et pour ne pas entraver le mouvement naturel de la surface, elle doit impérativement être cohérente avec les mouvements du fluide. La définition de la vitesse choisie recourt à une extrapolation de la vitesse de la surface du liquide selon l'équation suivante :

$$\frac{\partial u}{\partial \tau} + (n \cdot \nabla)u = 0 \quad n = \frac{\nabla \phi}{|\nabla \phi|}$$

Cette équation correspond à l'extrapolation de la vitesse le long de la normale à la surface n , en usant d'un pseudo-temps τ . Cette extrapolation de la vitesse

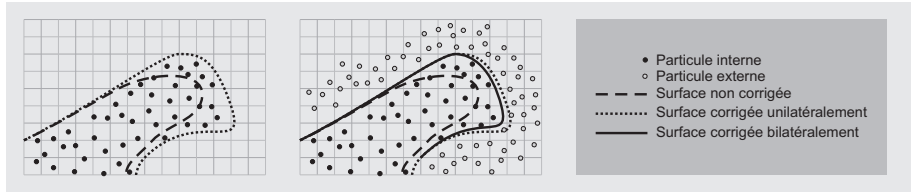


Figure 2.13: Méthodes de correction des *levelsets*. L'isocontour $\phi = 0$ représentant la surface est déplacé par des particules qui modifient les valeurs de ϕ localement dans les zones de forte courbure. Gauche : seules des particules internes au fluide sont ajoutées, empêchant des contractions parasites du fluide. Droite : la correction est effectuée des deux côtés de la surface

permet également de fournir d'une manière naturelle les conditions aux limites sur la vitesse pour la surface libre. Pour une extrapolation encore plus judicieuse, Rasmussen *et al.* [126] proposent de projeter également ce champ de vitesse externe au fluide sur l'espace des fonctions à divergence nulle. Cette correction, bien que non-requise explicitement puisque localisée hors du fluide, a pour effet de préserver une meilleure distribution des particules externes en cas de collision entre deux parties de la surface. La fusion résultante devient ainsi plus douce et plus réaliste.

Cette coercion bilatérale de la surface permet d'obtenir des résultats très réalistes, où des structures comme des films fins apparaissent et sont correctement maintenues dans le temps : elles ne sont ni perdues, ni empâtées.

Accroissement de la précision En adoptant ce type de représentation implicite de la surface, la simulation de son évolution se transforme en un problème d'advection d'une quantité par un champ externe. Deux méthodes différentes ont été introduites en infographie afin d'augmenter la précision de ces transports : Kim *et al.* [82] proposent d'employer la méthode BFEC — *Back and Forth Error Compensation and Correction* — et Takahashi *et al.* [159] ou Song *et al.* [147] la méthode CIP — *Constrained Interpolation Profile* — toutes deux illustrées dans la figure 2.14 et détaillées ci-dessous. Dans la mesure où l'advection de l'interface est un problème tout à fait analogue à celui du traitement de la composante convective du fluide, ces méthodes sont en fait applicables aux deux problèmes.

L'idée de la méthode BFEC introduite par Dupont et Liu [32] en physique, est de calculer dynamiquement, et localement, l'erreur commise lors d'une étape d'advection, et d'en tenir compte afin d'améliorer les résultats. L'hypothèse employée est que l'erreur est relativement homogène, quelle que soit l'advection effectuée. L'erreur e est alors mesurée en comparant le résultat de deux advections opposées, normalement sans effet, avec le champ initial q^n . Une étape d'advection corrigée L^* est alors très simplement implémentée à l'aide de trois étapes d'advection normales L .

$$\begin{cases} q^{n+1} = L^*(u, q^n) = L(u, q^n - e) \\ e = -\frac{1}{2}(q^n - L(-u, L(u, q^n))) \end{cases}$$

Sans nécessiter aucune implémentation supplémentaire, une telle technique

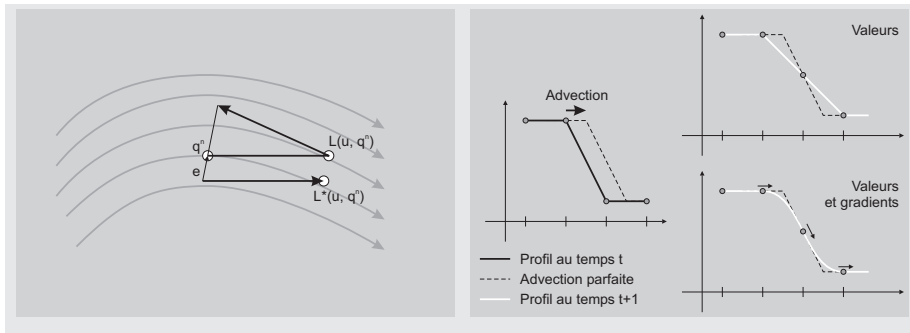


Figure 2.14: Améliorations de l'advection. Gauche : méthode BFEC. L'erreur e est approximée par deux advections de sens contraires puis compensée. Droite : méthode CIP. Une diffusion numérique survient lors de l'advection par rapport au profil exact figuré en pointillé. La prise en compte des gradients lors de l'advection permet une meilleure capture du profil.

permet une augmentation sensible de la précision du schéma d'advection, en réduisant l'erreur de précision temporelle et spatiale. Néanmoins, selon que l'advection concerne le champ de vitesse lui-même, ou la surface, il convient d'appliquer cette méthode avec précaution à proximité de l'interface, afin de ne pas introduire de mouvements parasites, et de ne pas perturber la surface.

Dans le cas de la méthode CIP, proposée par Yabe *et al.* [183], l'idée est d'accroître l'ordre d'interpolation des champs advectés lors du rééchantillonnage inhérent à une méthode semi-Lagrangienne. Il est proposé d'employer un polynôme de degré 3 pour représenter le champ sur une maille d'interpolation, le polynôme étant fonction des valeurs du champ aux coins de la maille, mais également de ses dérivées spatiales. Par conséquent, cette technique demande le maintien, et l'advection, des valeurs des dérivées en plus de celles du champ. L'équation de transport des différentes dérivées spatiales $q_\xi = \frac{\partial q}{\partial \xi}$ découle directement de celle du champ q :

$$\frac{\partial q_\xi}{\partial t} + (u \cdot \nabla) q_\xi = - \frac{\partial u}{\partial \xi} \cdot \nabla q$$

À l'exception du terme de forçage, au second membre, cette équation représente une advection par le champ de vitesse du fluide. Après prise en compte de ce terme supplémentaire, toutes les advections, du champ et des dérivées, peuvent donc être effectuées simultanément : le polynôme est évalué pour obtenir la valeur du champ, de même que sa dérivée analytique pour obtenir la valeur de la dérivée. Cette forme d'interpolation plus précise est étendue au cas d'un espace à plusieurs dimensions, en employant plusieurs interpolations monodimensionnelles successives. Elle permet la réduction de l'erreur en espace et se révèle très intéressante du fait qu'elle autorise une augmentation de la précision des advections tout en restant purement locale : elle évite l'emploi d'interpolations à supports larges, plus grands qu'une maille, toujours problématiques sur les bords du domaine.

En recourant à ces techniques, des simulations multiphasiques réalistes sont produites, avec une dissipation numérique fortement réduite à l'interface. Les interfaces entre phases sont beaucoup mieux préservées, et les volumes mieux

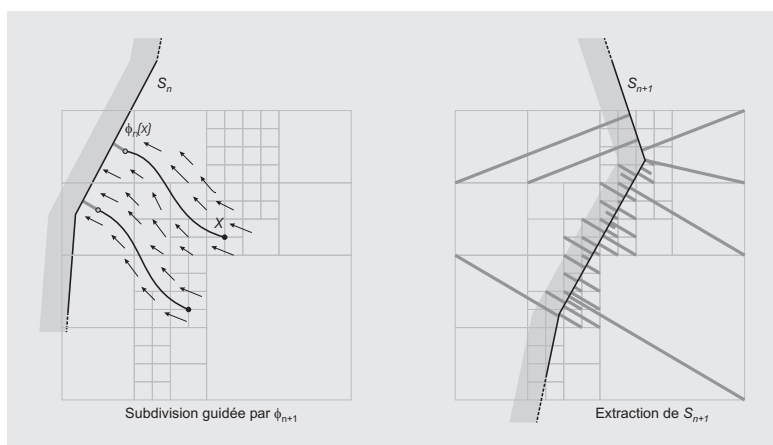


Figure 2.15: Évolution implicite de surface triangulée par méthode d'extraction semi-Lagrangienne. Une approximation de la fonction de distance ϕ_{n+1} est calculée adaptativement sur un *octree* par une méthode semi-Lagrangienne et la valeur de ϕ_n advectée est calculée exactement à partir de la représentation explicite maillée de la surface. Une fois l'*octree* construit, la surface est extraite par un algorithme de type *Marching-Cubes*.

conservés. Cette meilleure représentation de l'interface ouvre également la voie à la simulation de phénomènes fins telle la tension de surface. Par ailleurs, le fluide reste plus turbulent, ce qui contribue à renforcer l'impression visuelle de faible viscosité généralement recherchée. Les interactions avec des objets sont également plus vivantes car moins amorties.

Représentation explicite non-contrainte Si une représentation implicite de la surface par *levelset* offre de nombreux avantages, elle présente l'inconvénient de n'engendrer aucune cohérence temporelle explicite : les surfaces peuvent être facilement extraites à chaque pas de temps, mais ne sont pas corrélées. Dans un objectif d'infographie, cette limitation peut être gênante car il est naturel de chercher à associer à la surface des propriétés liées au rendu, comme des textures ou des couleurs.

Bargteil *et al.* [3] proposent une méthode hybride de suivi de la surface au cours du temps, qui répond à ce critère de cohérence temporelle, sans pour autant limiter les mouvements ou les modifications topologiques. Dans cette méthode, la surface du fluide est directement représentée comme un maillage triangulé. En revanche, son évolution est effectuée en considérant une vision implicite de la surface mue par une méthode semi-Lagrangienne. L'idée centrale de cette méthode, illustrée dans la figure 2.15, est de ne pas considérer de manière séquentielle, mais de manière simultanée, les problèmes d'advection et d'extraction de la fonction de distance.

Considérant deux surfaces S_n et S_{n+1} séparées par un pas de temps, et leurs représentations implicites sous forme de fonctions de distance ϕ_n et ϕ_{n+1} , un algorithme hiérarchique, s'appuyant sur un *octree* procède en calculant ϕ_{n+1} en chacun des nœuds de l'arbre à partir de ϕ_n . La subdivision récursive de l'arbre est directement contrôlée grâce aux valeurs de ϕ_{n+1} , qui fournissent une esti-

mation directe de l'erreur commise sur la position de S_{n+1} . Pour chaque valeur à évaluer, le calcul de la fonction de distance advectée ϕ_{n+1} est effectué par un traitement semi-Lagrangien guidé par le champ de vitesse du fluide. Toutefois, la distance à la surface S_n une fois la caractéristique rebroussée, est évaluée exactement et non pas interpolée d'une discrétisation de ϕ_n . Cette évaluation exacte de la distance à la surface est possible parce que cette dernière est accessible sous la forme explicite d'une collection de triangles. Par conséquent, aucune dissipation n'est introduite comme dans les méthodes précédemment décrites où seule une représentation discrétisée et interpolée de la fonction de distance est accessible. Finalement, l'extraction effective est réalisée par un algorithme de *Marching-Cubes*, sur la base des valeurs de ϕ_{n+1} stockées dans l'*octree*, une fois ce dernier construit à un niveau de précision jugé suffisant. Durant cette procédure d'extraction, aucune interpolation de ϕ_{n+1} n'est employée, mais de réelles recherches d'intersections sont effectuées en réemployant le même procédé d'évaluation de la fonction de distance que celui décrit ci-dessus.

C'est donc l'intégration de l'advection semi-Lagrangienne de la représentation implicite de la surface au cœur d'un algorithme d'extraction d'isosurfaces adaptatif, qui autorise le maintien d'une surface triangulée absolument non contrainte dans ses mouvements. Grâce à cette méthode, la dissipation numérique se trouve fortement réduite puisque la méthode opère sans discrétisation fixe. Outre l'obtention directe de surfaces triangulées, cette méthode présente l'avantage de mettre automatiquement en corrélation la nouvelle surface S_{n+1} avec celle du pas de temps précédent S_n . En effet, chaque sommet de S_{n+1} a, par définition, été généré par l'algorithme d'extraction à partir d'un antécédent localisé sur la surface S_n . Tout champ défini sur la surface de S_n peut donc être rééchantillonné sur S_{n+1} en considérant cette relation.

Domaine de simulation : discrétisation et modifications

La base des simulations Eulériennes est constituée par le maillage de l'environnement. Celui-ci délimite l'environnement de simulation, et sa résolution détermine la précision des résultats. Dans le cas de simulations tridimensionnelles, la taille du maillage augmente très rapidement en nombre de cellules, conduisant à des simulations nécessitant une grande quantité de mémoire et des ressources de calcul importantes. De plus, dans une vision usuelle de la simulation Eulérienne, la grille de simulation est fixe et définie à l'avance, ce qui limite l'ampleur des simulations réalisables, et impose de connaître à l'avance les grandes lignes du mouvement du fluide.

Optimisations Afin de pallier ces limitations, plusieurs techniques ont été proposées. Ainsi, Shah *et al.* [138] proposent de rendre la grille mobile, pour autoriser son resserrement et engendrer un meilleur taux d'utilisation. Cette technique est surtout intéressante dans les situations où un phénomène relativement localisé est amené à subir des déplacements importants, dans un domaine large ou non borné.

Dans ce cas, les équations de Navier-Stokes sont simulées sur une grille de taille prédéfinie, globalement mobile en translation. Dans ce cadre, le principe de l'invariance Galiléenne spécifie que les solutions aux équations de Navier-Stokes dans des repères fixes et mobiles sont assez simplement corrélées : une compensation de la vitesse de la grille est nécessaire, et un terme supplémentaire

contrebalançant les accélérations de la grille intervient dans les advections, ces deux termes pouvant être pris en compte de manière assez simple. La trajectoire choisie pour la grille est calquée sur la trajectoire suivie par le centre de la quantité portée par le fluide.

Afin de diminuer encore les ressources requises, une deuxième technique, plus agressive, est proposée en excluant de la simulation toutes les cellules dont le mouvement est jugé très homogène au regard d'un critère spécifié par l'utilisateur. Pour pallier l'éviction des cellules du domaine, des conditions aux limites de Neumann sont alors ajoutées dynamiquement à la simulation.

La combinaison de ces deux procédures permet un accroissement sensible des performances, en réduisant significativement le nombre de cellules de grille nécessaires à la simulation, la précision des résultats ne semblant pas être fortement affectée de ces optimisations. Néanmoins, il est à noter que l'approximation du flux par des conditions aux limites introduit indéniablement une modification de la solution, même si l'aspect général des mouvements reste préservé.

Rasmussen *et al.* [126] proposent une approche différente pour satisfaire à la même fin de réduction de ressources mais dans le cadre d'un liquide mobile. Plutôt que de traduire la grille, et de modifier les équations, ils proposent de considérer la grille de simulation comme une fenêtre logique repositionnable et redimensionnable sur une grille virtuelle plus grande. En fait, les calculs restent uniquement localisés dans la fenêtre, aucun liquide à simuler n'étant présent hors de cette fenêtre. Si le liquide menace de déborder hors de la fenêtre au cours de la simulation, la fenêtre est redimensionnée et traduite en conséquence. La translation de la fenêtre n'est définie que par incréments entiers, ce qui évite tout rééchantillonnage et donc toute interpolation forcément dissipative, seul un décalage des valeurs dans la grille étant requis.

Cette approche ne présente toutefois aucun intérêt dans le cadre de simulations de grande ampleur, très disséminées dans l'espace, la fenêtre logique redevenant rapidement équivalente à la définition d'une grande grille. Dans le cadre de liquides essentiellement mus par la gravité, il est supposé que la pression est d'influence limitée, ce qui permet de relaxer la nature globale de la simulation. Il est donc proposé de gérer la simulation sur deux échelles. À haut niveau, des blocs de calcul sont disposés dans l'espace, en se recouvrant légèrement. À petite échelle, chacun de ces blocs est maillé par une grille régulière sur laquelle une simulation Eulérienne classique peut-être effectuée. Le calcul sur ces petites grilles a lieu dans un ordre déterminé par le flux global, d'amont en aval. Afin d'assurer un transfert d'information entre deux blocs de calcul, la zone de recouvrement entre blocs de la simulation en aval est peuplée de conditions aux limites dérivées de la simulation en amont. Le recouvrement avec une grille en aval donne lieu à des conditions de Dirichlet plus arbitraires.

Grâce à ces techniques, il devient possible de cerner assez précisément des liquides exhibant des mouvements complexes et des géométries tourmentées sans grande déperdition de ressources de calcul. Cette méthodologie apporte également une grande flexibilité pour appréhender des mouvements non prévus, puisqu'il est possible de rajouter dynamiquement de nouveaux blocs de calculs, si nécessaire.

Représentation hiérarchique Si les techniques présentées jusqu'ici contribuent à optimiser les ressources, elles ne s'appliquent *a priori* qu'à des grilles

possédant des mailles régulières. Elles se révèlent donc peu adaptées à la représentation de phénomènes d'échelles très différentes au sein d'une même simulation. Pour apporter une solution à ce problème, Losasso *et al.* [94] proposent l'emploi d'une discrétisation complètement hiérarchique en utilisant un *octree*. Pour la simulation du champ de vitesse, l'algorithme de résolution par la méthode de projection de Chorin employé n'est pas modifié par le passage à une discrétisation hiérarchique.

Concernant la discrétisation, la logique présidant à l'arrangement décalé des variables de la méthode MAC est préservée : la pression est échantillonnée au centre des cellules, et la composante orthogonale de la vitesse au centre de chaque face. En revanche, pour plus de facilité, les quantités scalaires autres que la pression sont définies aux nœuds du maillage, donc aux coins des mailles. Il est à noter que l'équilibrage de l'*octree* n'est pas requis. Cependant, la valeur portée en tout nœud non-conforme se voit contrainte pour maintenir la continuité des champs à l'interface des cellules de l'*octree*.

Concernant la résolution, la méthode semi-Lagrangienne utilisée pour traiter l'advection ne requiert aucune modification. Le principal changement concerne donc la discrétisation des opérateurs différentiels intervenant lors de la projection pour assurer l'incompressibilité : les calculs de la divergence du champ de vitesse, et des Laplacien et gradient de la pression, se trouvent donc légèrement modifiés. Dans le cas du Laplacien, il faut veiller à ce que la discrétisation adoptée aboutisse à une matrice symétrique définie positive, pour des raisons d'efficacité de résolution.

L'adaptativité de résolution ainsi gagnée permet une bonne représentation des structures fines du fluide en toutes situations. Le critère de subdivision de l'*octree* dépend du fluide simulé : pour les gaz il fait intervenir la proximité des obstacles, la vorticit , et la densit  de fum e advect e ; pour les liquides le crit re logique est la proximit    la surface. Des simulations d taill es et rapides sont possibles gr ce   cette approche hi rarchique. Les r sultats obtenus pr sentent cependant une pr cision  quivalente   ceux issus d'une grille r guli re, la pr cision restant limit e par la m thode semi-lagrangienne.

Maillages t tra driques Malgr  la possibilit  de capturer plus finement des obstacles dans le fluide, cette approche hi rarchique reste limit e   l'emploi de conditions aux limites orthotropes, align es avec les faces des mailles. Pour relaxer une telle hypoth se, Feldman *et al.* [42] proposent l'int gration de cellules t tra driques et transitionnelles   une grille r guli re compos e de cellules hexa driques. Ces nouveaux types de cellules suivent la m me logique de discr tisation que les cellules cubiques originales de la m thode MAC : les quantit s scalaires sont  chantillonn es au centre des cellules, et les quantit s vectorielles au centre des faces, dans la direction de la normale, comme illustr  dans la figure 2.16. La pr sence des cellules de transition est requise pour permettre la conformit  du maillage de simulation.

Cette discr tisation est employ e dans le cadre d'une simulation par la m thode de projection de Chorin. Par cons quent, et comme pour la m thode d crite pr c demment, des discr tisations des op rateurs diff rentiels mis en jeu lors de la projection sont fournies. Elles restent tr s simples, et co ncident parfaitement avec les discr tisations par diff rences finies usuelles dans les cas hexa driques. En outre, elles assurent que la projection garantissant l'incom-

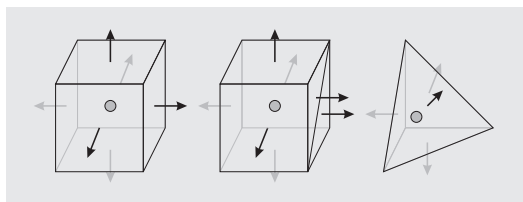


Figure 2.16: Types de cellules d'une grille décalée hétérogène : cellules cubiques, de transition, et tétraédriques. Les points de discrétisation des différentes composantes de la vitesse et de la pression ne sont pas collocalés.

pressibilité met en jeu des matrices symétriques définies positives, pouvant être résolues efficacement.

Par ailleurs, une forme d'interpolation du champ de vitesse est nécessaire afin de pouvoir résoudre la composante advective par une méthode semi-Lagrangienne. Dans le cas des cellules hexaédriques, une interpolation trilineaire est utilisée. Dans le cas des autres cellules, une forme d'interpolation employant la méthode des moindres carrés mobiles est proposée. Deux variantes sont possibles, plus ou moins précises, selon le degré, zéro ou un, de l'interpolation. Dans les deux cas, la méthode cherche à approximer la vitesse locale, au sens des moindres carrés, étant donné les vitesses normales aux cellules environnantes et un noyau de lissage. Le taux de lissage est fourni par l'utilisateur.

En employant une grille définie à l'aide de ces cellules, hexaédriques dans la majorité du volume et tétraédriques sur les bords, il devient possible d'exprimer très simplement des conditions aux limites se conformant à une géométrie de frontière maillée arbitraire. L'emploi de ces cellules hexaédriques assure une efficacité de calcul plus grande qu'un maillage uniquement tétraédrique. Ce type de grille hybride permet donc d'allier meilleure efficacité et représentation plus fidèle du domaine.

Maillages mobiles Afin d'accroître encore la souplesse de ce type de grilles hybrides, Feldman *et al.* [43] introduisent le support d'une formulation ALE — *Arbitrary Lagrangian and Eulerian* — pour autoriser des frontières mobiles dans le temps. Une notion de vitesse de grille \bar{u} est donc introduite dans les équations de Navier-Stokes non visqueuses pour tenir compte des mouvements propres du maillage :

$$\begin{cases} \frac{\partial u}{\partial t} = -((u - \bar{u}) \cdot \nabla)u - \nabla p + f \\ \nabla \cdot u = 0 \end{cases}$$

Pour tenir compte de cette vitesse de grille, la méthode semi-Lagrangienne de traitement du terme convectif est légèrement modifiée : au lieu de rebrousser les trajectoires et d'interpoler le champ de vitesse dans le maillage courant, cette étape est réalisée dans le maillage du pas de temps précédent. En revanche, les points de départ des trajectoires rebroussées restent définis par les positions d'échantillonnage découlant du maillage courant, comme illustré dans la figure 2.17. L'intérêt d'une telle approche, par rapport à un traitement du terme convectif dans le maillage précédent suivi d'un rééchantillonnage, est d'éviter l'introduction de la dissipation numérique supplémentaire associée.

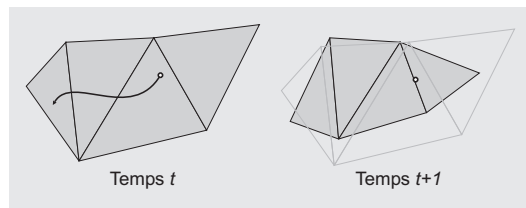


Figure 2.17: Méthode semi-Lagrangienne pour un maillage mobile : pour tenir compte de la variation de position du maillage, l'advection est calculée en employant le champ de vitesse, et donc le maillage, du pas de temps précédent.

En employant une telle méthode, il devient possible de considérer des domaines de simulation mobiles et se déformant au cours de la simulation. Toutefois, ces déformations ne doivent pas être trop importantes, la régularité du maillage étant, comme pour toute méthode employant une grille, un facteur déterminant de la qualité des solutions. Afin de pallier ces problèmes de dégradation progressive d'un maillage mobile et de lever les limitations imposées aux mouvements possibles, Klingner *et al.* [86] proposent d'étendre cette approche en remaillant de manière dynamique le domaine au cours de la simulation, grâce à une approche apparentée à celle proposée par Alliez *et al.* [1]. Un tel remaillage permanent induit toutefois un coût supplémentaire non négligeable, mais permet alors possible de considérer des déplacements importants d'objets dans le domaine, ou des variations substantielles du domaine lui-même.

Si le remaillage laisse une plus grande liberté dans la variation du domaine, il permet également de concentrer la précision de représentation à proximité des zones de la simulation jugées plus particulièrement pertinentes, qui peuvent varier dans le temps, en adaptant localement la finesse des mailles au cours de temps. Cette flexibilité accrue permet ainsi un gain en finesse de représentation ainsi qu'en coût d'exécution, par rapport à un maillage statique.

Simulations sur des surfaces courbes Les méthodes présentées précédemment s'attachent à améliorer le calcul d'une solution définie dans un espace Cartésien. Cependant, il peut être intéressant, d'un point de vue visuel, de restreindre l'évolution du fluide à une surface bidimensionnelle plongée dans un espace tridimensionnel. Cette problématique de simulation dans un domaine de géométrie et de topologie arbitraire a également été étudiée. Ce type de simulation, appliqué à un domaine restreint par rapport à un espace de dimension plus élevée, permet de focaliser les ressources de simulation, mais permet surtout de contraindre simplement les mouvements du fluide à suivre une géométrie donnée. Cette contrainte peut ainsi se révéler utile pour simuler l'écoulement approximé d'un fluide en couche mince adhérent à une surface quelconque, ou encore pour simuler de manière bidimensionnelle l'atmosphère d'une planète.

Stam [152] propose d'appliquer une résolution suivant la méthode « *Stable Fluids* », mais en coordonnées curvilignes. Une forme de paramétrisation globale continue de la surface est donc requise pour pouvoir appréhender le domaine de simulation. Pour cette raison, les domaines de simulation considérés sont des surfaces de subdivision de Catmull-Clark, pour lesquelles cette paramétrisation existe naturellement. Le domaine de simulation complet est donc appréhendé

sous la forme d'un patchwork de quadrilatères, correspondant au niveau racine de la subdivision. Chaque quadrilatère donne par suite naissance à une grille régulière. Sous couvert d'une résolution identique, ceci permet d'assurer une discrétisation compatible de l'intégralité de la surface. Il convient toutefois d'être attentif aux transformations nécessaires pour tenir compte des orientations relatives entre grilles locales, dès lors qu'une même quantité doit être considérée dans deux grilles différentes, et ce, à toutes les étapes de la résolution.

Exceptées ces différences, l'algorithme de résolution ne varie pas dans sa démarche par rapport à son implantation Cartésienne. Néanmoins, tous les opérateurs différentiels voient leurs définitions enrichies par la présence du tenseur métrique, permettant de tenir compte de la déformation locale due à la géométrie de la surface.

Considérant la même situation de surface arbitraire, Shi et Yu [139] adoptent une approche différente, qui autorise l'emploi de tout maillage triangulé comme domaine de simulation. La simulation emploie toujours la méthode de projection de Chorin associée à un traitement semi-Lagrangien de l'advection, mais le fluide est cette fois supposé non-visqueux.

La discrétisation est relativement inattendue puisque la vitesse est supposée constante sur chaque triangle du maillage, alors que la pression est interpolée linéairement. Pour réaliser la projection garantissant l'incompressibilité, le problème de Poisson est discrétisé directement sur le maillage triangulaire, en considérant que le champ de vitesse dérive d'un potentiel linéairement interpolé sur chaque triangle. L'advection, outre la contrainte de suivre une trajectoire tangente au maillage, doit composer avec l'approximation discontinue choisie pour la vitesse. Ainsi, une forme de reconstruction du champ de vitesse est d'abord considérée, en chaque sommet du maillage, en interpolant les vitesses définies sur les triangles du voisinage. Le calcul de la trajectoire est ensuite effectué en employant ce champ auxiliaire continu.

Si une approche plus algorithmique et plus géométrique que pour la méthode précédente permet de considérer une classe d'objets plus large, il convient de remarquer que la phase d'advection reste moins satisfaisante, car définie au travers d'approximations et d'interpolations successives.

2.3.3 Extensions

Les méthodes de simulations du fluide, liquide ou gazeux, décrites dans la précédente section traitent de la simulation Eulérienne du fluide, et uniquement du fluide, dans un cadre relativement restreint. En outre, le seul contrôle disponible sur le fluide est son nombre de Reynolds. Si ces simulations donnent lieu à des animations très intéressantes, elles peuvent, du point de vue de l'infographie, sembler peu variées. Afin d'enrichir la variété des situations simulables, plusieurs extensions ont été proposées, allant de la modification des propriétés des fluides, qui ne sont alors plus nécessairement fluides, à une meilleure gestion des bulles et des gouttes apparaissant durant la simulation, en passant par la prise en compte automatique de l'interaction avec d'autres objets de la scène virtuelle. Ce dernier type d'extension, quoique n'ayant pas reçu une large attention, mais constituant la motivation des travaux de cette thèse, est détaillé dans la section 2.3.4.

Extension de la nature du fluide

Si un objectif récurrent des méthodes déjà détaillées est de présenter le moins de dissipation numérique possible, afin de simuler plus correctement des fluides de faible viscosité, il existe des situations où un jeu sur la viscosité permet la manipulation de fluides visuellement intéressants. Ainsi, Carlson *et al.* [15] proposent de moduler dynamiquement la viscosité selon un paramètre de température pour pouvoir simuler la fusion d'objets solides. Dans une approche semblable à celle suivie par Tonnesen [166], décrite à la section 2.2.2, les solides et les liquides sont considérés comme deux états d'un même modèle paramétré par la température : les solides sont donc simulés comme des fluides froids, nantis d'une viscosité extrêmement élevée, au contraire des liquides, chauds et faiblement visqueux.

Ce matériau, dans tous ces états, est donc simulé à l'aide des équations de Navier-Stokes tenant compte d'une viscosité variable. La viscosité ν est fonction de la température locale T , dont l'évolution est régie par une équation d'advection-diffusion.

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nabla \cdot (\nu \nabla u) - \frac{1}{\rho} \nabla p + f \\ \nabla \cdot u = 0 \\ \nu = \nu(T) \\ \frac{\partial T}{\partial t} = k \Delta T - u \cdot \nabla T \end{array} \right.$$

La simulation numérique de ces équations emploie une approche MAC, et met à profit une approche de type *Stable Fluids*, légèrement modifiée. Afin de gérer correctement les hautes viscosités, il est fait usage d'une résolution implicite de la composante diffusive. Dans la mesure où le fluide ne remplit pas l'intégralité du domaine de simulation, une résolution matricielle est employée, où l'opérateur de diffusion est discrétisé par différences finies en tenant bien compte de la viscosité locale variable, plutôt qu'une résolution spectrale comme dans la méthode *Stable Fluids* originale. Cette forme de résolution pose cependant un problème lors de l'emploi de hautes viscosités : elle tend à supprimer l'inertie des différentes composantes connexes du fluide en mouvement. Une procédure de correction est donc suggérée, où l'inertie calculée et mémorisée avant que la diffusion ne soit appliquée est réintroduite ultérieurement. Sa conservation est ainsi assurée explicitement. Une autre modification mineure suggérée consiste en la modification de l'interpolation du champ de viscosité, afin de rendre dominantes les faibles viscosités et de ne pas solidifier trop rapidement les fluides en mouvement jouxtant un fluide froid et rigide. La simulation de la température est assurée par une combinaison de différenciation en amont, pour la convection, et d'une résolution implicite, pour le terme diffusif.

Pour simuler des fluides visco-élastiques, Goktekin *et al.* [54] ajoutent un terme d'élasticité aux équations de Navier-Stokes. Celui-ci est logiquement défini en égard aux déformations subies par le matériau en raison de son élasticité, mesurées à l'aide du tenseur des déformations élastiques ϵ^E .

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nu \Delta u + \mu \nabla \cdot \epsilon^E - \frac{1}{\rho} \nabla p + f \\ \nabla \cdot u = 0 \end{array} \right.$$

Cependant, dans la mesure où le fluide est extrêmement mobile, et peut subir des modifications topologiques, il n'est pas possible de calculer le terme d'élasticité à partir de la déformation d'une configuration initiale de référence. Une approche incrémentale est donc adoptée, et les déformations élastiques sont évaluées par intégration, compte tenu de leur variation au cours du temps. En fait, dans la mesure où le matériau considéré exhibe également une part de plasticité, mesurée par ϵ^P , la mesure de la déformation élastique est un peu plus complexe à calculer. Le calcul reste toutefois possible à partir des propriétés intrinsèques du matériau et de la variation de la déformation totale $\dot{\epsilon}$ qui est mesurable à partir du champ de vitesse du fluide.

$$\begin{aligned}\epsilon &= \epsilon^E + \epsilon^P \\ \epsilon &= \epsilon_0 + \int_0^t \dot{\epsilon} dt \\ \dot{\epsilon} &= \frac{1}{2} (\nabla u + \nabla u^T)\end{aligned}$$

Une dernière difficulté subsiste dans cette formulation : le tenseur des déformations élastiques, représentant une propriété matérielle du fluide, doit être stocké et advecté avec le fluide. Pour cela, une advection semi-Lagrangienne est employée, comme pour les autres propriétés du fluide. Ce tenseur, de rang deux, est alors stocké selon une extension du schéma décalé introduit avec la méthode MAC.

Une autre extension possible des simulateurs de fluide, proposée par Zhu et Bridson [189], est le traitement de matériaux granulaires comme le sable. L'idée principale est de considérer le sable comme un *continuum* fluide incompressible et sans viscosité, auquel est ajouté un comportement supplémentaire pour tenir compte de la nature particulière du matériau simulé, tantôt adhérent, tantôt glissant sur lui-même.

Dans le calcul d'un pas de temps, une première étape consiste donc en la simulation de la composante fluide. Ensuite, les contraintes locales sont calculées pour l'intégralité du sable, et comparées à un critère de rigidité locale, pour délimiter les différentes composantes connexes du sable jugées rigides. Chacune de ces composantes est ensuite projetée dans l'espace des déplacements rigides. Le reste du sable, jugé en condition de glissement, subit pour sa part des forces de friction.

Cette modification de la résolution est complétée par l'intégration de conditions aux limites exprimant la friction du sable avec son environnement, conditions qui dérivent de la loi de friction de Coulomb. Cette forme de condition aux limites est nécessaire pour permettre au sable de se stabiliser et d'adopter une position statique, sans pour autant rendre l'environnement en tout point instantanément adhérent, comme avec des conditions de Dirichlet.

La simulation découlant de cette modélisation est réalisée en employant la méthode de projection de Chorin alliée à un traitement PIC ou FLIP de la composante convective. La démarche employée par ces derniers a été évoquée dans la section 2.3.2.

Ces trois extensions présentées permettent ainsi d'étendre la souplesse des simulateurs de fluide Eulériens et de donner naissance à des matériaux existants ou non, mais toujours susceptibles de répondre dans leur évolution aux attentes de réalisme d'un animateur. Ce réalisme est dans une certaine mesure garanti

par l'introduction dans les modèles de termes physiquement cohérents, et par une résolution adéquate lors de la simulation. Sans affirmer que le contrôle de ces matériaux se révèle intuitif, le nombre de paramètres les décrivant semble suffisamment restreint pour les rendre utilisables.

Bulles, gouttes et simulations diphasiques

Lors d'une discrétisation Eulérienne, et même en recourant à l'emploi d'une structure hiérarchique, il reste difficile de préserver des structures fines telles les bulles ou les gouttes. Ces structures fines apportent pourtant beaucoup au réalisme des scènes liquides simulées. C'est pourquoi plusieurs techniques ont été développées dans le but de les faire apparaître et de les maintenir plus correctement. À cet effet, il devient intéressant de considérer la tension de surface, composante importante de leur formation, largement ignorée dans les simulations de plus grande échelle. La simulation d'un problème réellement diphasique peut dès lors se justifier, la deuxième phase étant souvent constituée de l'air environnant.

Représentation Lagrangienne des petites inclusions Dans le cas de simulations spatialement étendues, la difficulté principale est de maintenir l'information de présence des bulles ou des gouttes, les forces de tension de surface étant dans ces cas faibles au regard de l'importance des mouvements présents. L'approche classique, similaire à celle proposée par O'Brien et Hodgins [115] dans le cadre d'une simulation Eulérienne restreinte où le corps du fluide est représenté par un champ de hauteur, consiste alors en l'adjonction d'un système de simulation Lagrangien pour les bulles ou les gouttes, en plus de la représentation Eulérienne choisie pour la composante principale du fluide.

Takahashi *et al.* [159] utilisent un système de particules balistiques pour représenter les gouttes se détachant d'un liquide, et étendent cette approche par l'ajout de mousse à la surface. Le simulateur de fluide est fondé sur une approche de type VOF — *Volume Of Fluid* — Outre les données intrinsèques à la simulation, comme la vitesse ou la pression, la fraction du volume de chaque cellule de la grille occupée par le liquide est alors simulée, ce qui permet de localiser l'interface. En considérant le domaine comme peuplé d'un seul fluide de caractéristiques variables, les mouvements de l'air environnant le liquide sont donc naturellement simulés, et il n'est pas nécessaire de traiter l'interface de manière spécifique pour coupler les deux fluides. La résolution emploie une méthode CIP, comme décrite dans la section 2.3.2, pour simuler les fluides et leur interface commune sans trop de dissipation numérique.

À cette modélisation du corps du fluide est ajouté un système de particules destiné à modéliser les gouttes en vol au dessus du fluide, ainsi que la mousse à la surface du liquide. Des particules sont éjectées du fluide dans l'air, quand la surface du liquide présente une forte courbure. Ces particules sont alors soumises à la gravité, et à la viscosité de l'air. Simuler cette dernière est possible car l'air est explicitement modélisé par le simulateur. Lorsqu'une particule rencontre ultérieurement la surface du fluide, elle est supposée donner naissance à une fraction de mousse, flottant à la surface. Pour le reste de sa vie, limitée dans le temps, elle reste contrainte à surnager à la surface du liquide, déplacée par une force de viscosité engendrée par le fluide. Ces particules, indicatrices de gouttes

ou de mousse, sont ensuite employées lors de la phase de rendu pour figurer ces effets.

Si ce type de modélisation permet d'agrémenter visuellement les mouvements violents de fluide de manière plaisante et réaliste, il n'en reste pas moins que cette technique se limite essentiellement à un post-traitement visuel. Ainsi, aucun fluide n'est effectivement transporté dans le domaine de simulation, ce qui n'est pas nécessairement gênant, du moins dans le cadre de la simulation de grands environnements, pour lesquels il est légitime de supposer que le transport induit par des gouttes est négligeable.

Greenwood et House [57] présentent une technique sensiblement similaire dans le cas de la représentation de bulles d'air incluses dans un liquide à surface libre, spécifiquement pour les simulateurs employant une représentation du fluide à l'aide de la méthode *Particle Levelset*, décrite à la section 2.3.2. La création des bulles est fondée sur l'emprisonnement par le fluide de particules représentant l'air environnant. En fait, ces particules sont précisément les particules externes au fluide destinées à prévenir les expansions parasites de la surface, ce qui explique la dépendance particulière au mode de représentation *Particle Levelset*. Une fois engoutties, ces particules prennent le statut de bulles sphériques, et le *levelset* est corrigé afin que le simulateur de fluide perçoive un fluide plein, sans cavité. La taille des bulles est déterminée aléatoirement, selon une moyenne et un écart-type dépendant de l'information de distance à la surface initialement associée à la particule, lors de son usage de correction du *levelset*. À leur retour à la surface, les bulles sont stochastiquement supprimées. Durant leur vie, elles subissent diverses forces, provenant du fluide et éventuellement d'interactions entre elles et avec leur environnement, selon leur position par rapport à la surface du fluide. Néanmoins, elles ne sont en aucun cas autorisées à s'échapper du fluide, et ne sont pas capables d'appliquer des forces au fluide en retour, ni d'engendrer aucun transport réel de fluide dans le simulateur.

Kim *et al.* [83] proposent d'améliorer la technique précédente, pour le cas des gouttes s'échappant du volume de fluide principal. Toujours gérées comme des entités Lagrangiennes sphériques, elles voient leurs volumes découler d'une mesure de la différence de volume entre la représentation implicite sous forme de *levelset* et la représentation particulaire. Cette fraction du fluide mal représentée est mesurée par une intégration de Monte-Carlo de la présence du fluide sous ces deux représentations, dans les cellules de la grille où elles sont en désaccord. Une fois générées, les particules ne sont pas déplacées selon un mode balistique, mais sont advectées selon un champ de vitesse d'un fluide virtuel, simulé par une méthode hybride FLIP – PIC. Une restitution des vitesse et masse portées par les particules survient lorsque celles-ci regagnent le corps du fluide.

Le rendu des particules générées est assuré par le biais d'une surface implicite, définie en les prenant pour squelette, à l'aide d'un processus de *ray-marching* adaptatif. Un processus de densification du nuage de particules, durant la phase de rendu, est également mis en œuvre, afin de contrôler visuellement l'épaisseur des films de liquide produits. Les particules générées sont également employées pour induire un terme de force destiné à déplacer un brouillard de fines particules autour du fluide, représentées comme un *continuum*. L'évolution de ce brouillard est gouvernée par les équations d'Euler incompressibles, pour ce qui concerne son déplacement, et par une loi d'advection et de diffusion, pour ce qui concerne sa densité.

Dans l'optique d'une simulation plus précise d'un volume de fluide restreint, Song *et al.* [147] proposent une technique alternative, mettant à profit une simulation diphasique. Ils couplent l'apparition et la disparition de gouttes, ou symétriquement de bulles, à une modification des valeurs du *levelset* employé pour représenter la surface du liquide. Ces particules, représentant une réelle fraction de fluide, sont porteuses d'une masse et d'un volume qui sont déterminés, à la création de la particule, à partir des valeurs du *levelset*, en considérant un modèle de distribution sphérique. Ils peuvent être amenés à évoluer dans le temps, si plusieurs particules sont fusionnées car elles se recouvrent partiellement. La création des particules est effectuée pour de petites régions de fluide ou de vide isolées, ou dans le cas d'une variation de masse purement numérique. Ces particules évoluent ensuite sous l'action de diverses forces, en engendrant simultanément une réaction sur le fluide Eulérien traversé, jusqu'à rencontrer l'interface où une restitution du fluide transporté a lieu, en modifiant la valeur du *levelset*. Dans le cas où l'agrégation d'un nombre suffisant de particules permet à nouveau une représentation Eulérienne significative, cette restitution peut également avoir lieu avant même le retour à l'interface.

Simulation multiphasique et tension de surface Outre la conservation de la masse, l'autre aspect important de l'incorporation dans une simulation de bulles ou de gouttes est la prise en compte de la tension de surface. Cette tension correspond aux forces qui s'exercent à l'interface du fluide, forces dues à l'anisotropie de la configuration géométrique locale qui perturbe la régularité de la cohésion des molécules. Les méthodes présentées ci-après s'attachent à la reproduction précise de l'interface entre fluides ainsi qu'à la reproduction de cette force. La nécessité d'une capture précise de l'interface ne permet donc que des simulations d'échelle restreinte, focalisées sur ces structures fines.

Afin de capturer la dynamique des grandes bulles déformables dans un liquide, Hong et Kim [72] proposent d'employer une simulation VOF, simulant conjointement le liquide et l'air contenu dans les bulles. Le recours à une tel mode de simulation garantit naturellement une réelle interaction entre les deux phases, puisqu'elles sont toutes deux vues comme un même fluide de densité variable. Une force liée à la tension de surface est également intégrée à la simulation.

Cependant, plutôt que de calculer explicitement cette force à l'interface après reconstruction de celle-ci, une méthode alternative, heuristique, est proposée. La force est calculée pour l'ensemble du volume, en chaque cellule, directement à partir des valeurs locales de quantité de fluide disponibles grâce à l'utilisation d'une méthode VOF. Considérant une cellule, la valeur moyenne de la présence du fluide dans son voisinage est utilisée comme valeur de référence de placement d'une surface hypothétique ne subissant aucune contrainte. Ensuite, la valeur effective de la force est calculée en fonction de la déviation de la surface effective par rapport à cette surface hypothétique. Dans le cas d'une cellule non proche de l'interface, les deux valeurs comparées sont nécessairement égales, ce qui n'engendre aucune force.

Finalement, la tension de surface se trouve exprimée comme une force volumique, dérivée simplement de l'indicateur de présence du fluide, et ne prend une valeur non-nulle que dans un voisinage de la surface. En outre, son calcul ne nécessite pas de reconstruction de surface, ce qui évite des valeurs peu

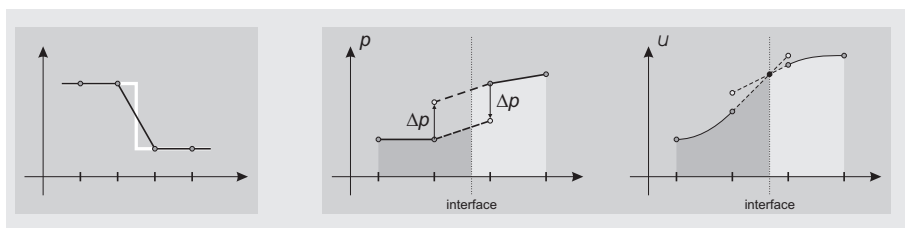


Figure 2.18: Extrapolation à l'interface de deux fluides. Gauche : les discontinuités ne peuvent être correctement capturées lors d'un échantillonnage. Droite : Méthode *Ghost Fluid* où la pression et la vitesse sont extrapolées séparément pour chacun des deux fluides à travers l'interface, permettant ainsi de réintroduire la discontinuité.

stables découlant d'interfaces tourmentées. Par ailleurs, l'emploi d'une méthode VOF, stockant exactement en chaque cellule la quantité de fluide qui s'y trouve, permet à tout moment une analyse parfaite de la masse présente dans le domaine. Ce taux de remplissage local des cellules est également mis à profit pour combattre toute dissipation artificielle engendrée par le schéma numérique : les quantités de fluide par cellule sont artificiellement saturées pour toutes les cellules éloignées des interfaces, et celles-ci sont ensuite déplacées via l'application d'un facteur de correction aux cellules proches, pour conserver la masse totale. Ceci permet à la simulation de régénérer, à chaque pas de temps, un champ de présence de fluide proche d'un champ binaire, sauf à l'interface, et de combattre ainsi toute dissipation numérique.

Pour renforcer le réalisme de la simulation, de petites bulles sphériques indéformables sont simulées en sus et sont gérées comme des particules externes. Une fois créées, elles ne sont pas en interaction avec le fluide, et évoluent tout à fait indépendamment de lui. Si cette méthode présente une grande simplicité dans sa mise en œuvre, elle reste néanmoins essentiellement approximative, et empirique. Une deuxième méthode, reposant sur une prise en compte plus exacte des discontinuités à l'interface du fluide est présentée par Hong et Kim [74], en s'inspirant de la *Ghost Fluid Method* introduite par Fedkiw *et al.* [40] en physique. Cette technique est appliquée dans le contexte d'un simulateur diphasique de fluides incompressibles, employant la méthode de projection de Chorin et une représentation de l'interface par *levelset*.

L'idée centrale de la méthode est d'extrapoler les quantités intervenant dans les calculs réalisés à proximité de l'interface, séparément pour chacune des phases. Ceci permet de mieux capturer la discontinuité en évitant le lissage qui survient si le traitement de la discontinuité n'est qu'implicite, comme illustré dans la figure 2.18. Dans le cas de la manipulation de la pression, l'extrapolation est réalisée en considérant la corrélation entre le saut de pression à l'interface et la courbure de l'interface. Par suite, cette extrapolation permet de considérer une discrétisation plus fidèle de l'équation de Poisson impliquée dans la projection garantissant l'incompressibilité, en tenant compte de la discontinuité. De manière similaire est appliqué un traitement sensible à la discontinuité du gradient de la vitesse à l'interface, pour des fluides de viscosités différentes.

Associée à une discrétisation hiérarchique par *octree*, cette méthode permet un traitement physiquement bien fondé des phénomènes fins liés à la tension de surface, malgré des différences élevées de viscosité et de densité entre les diffé-

rentes phases. Il est ainsi possible d'obtenir des animations très convaincantes où de nombreuses bulles sont présentes, et où se forment des nappes minces de liquide. Ces deux phénomènes sont tous deux de bons indicateurs d'une gestion correcte de la tension de surface. Il est à noter qu'aucune force n'est explicitement prise en compte, mais que le seul traitement précis du saut de pression à l'interface suffit à la faire apparaître.

Toujours pour simuler des bulles de volumes importants, Zheng *et al.* [188] proposent, dans le contexte d'un simulateur de fluide s'appuyant sur les équations de Navier-Stokes, de modifier d'une part la représentation de la présence du fluide, et d'autre part d'employer un modèle de tension de surface pris en compte implicitement pour assurer une stabilité numérique inconditionnelle. Pour la simulation des mouvements du fluide est employée une approche diphasique, avec un modèle de fluide continu à l'interface, qui lisse les discontinuités de propriétés aux interfaces.

La représentation introduite, pour le fluide et les bulles, s'appuie et étend une approche de type *levelset*, et consiste à adjoindre à la fonction de distance à l'interface une fonction indicatrice d'un partitionnement de l'espace. Chaque bulle peut ainsi être marquée comme un élément différent de la partition, ce qui permet de définir implicitement des interfaces d'épaisseurs infinitésimales à la frontière de régions marquées comme différentes.

La création des bulles est assurée de manière externe, ou par détection des vides naissant dans le corps du fluide. Pour simuler l'évolution des bulles formées, l'épaisseur de l'interface de chacune est simulée au cours du temps. Cette simulation reste simple, puisqu'une épaisseur unique est considérée pour l'ensemble du film bordant chaque région de l'espace. La fusion de bulles adjacentes est alors gouvernée par la comparaison de l'épaisseur avec un seuil donné.

Dans le but de simuler le processus d'ébullition, caractérisé par la formation de bulles, Mihalef *et al.* [102] proposent d'adjoindre à une simulation de fluide Eulérienne diphasique, une simulation de la température, en incluant un phénomène de transfert de masse du liquide vers le gaz, pour tenir compte du phénomène de vaporisation du liquide. Les bulles n'y sont pas simulées comme des inclusions dans le volume du fluide, mais comme des poches de gaz à part entière, présentes dans le fluide.

La simulateur de fluide mis en œuvre à la base de cette technique est constitué d'une méthode hybride combinant une simulation VOF, pour garantir une bonne conservation de la masse au cours du temps, et un suivi de la surface par une méthode de type *levelset*, pour mieux approximer la surface et en particulier sa normale. Aux équations liées à la simulation des mouvements du liquide et du gaz sont ainsi ajoutées des équations décrivant aussi bien la diffusion de la chaleur que la variation de volume du liquide ou du gaz liée à la vaporisation. Grâce au couplage réel existant ainsi entre fluide et bulles, ces dernières engendrent des mouvements significatifs de circulation dans le fluide. De plus, la gestion du transfert de la chaleur autorise une formation et une croissance réaliste des bulles au cours de la simulation.

Dans l'optique de simuler le déplacement de gouttes de liquide en contact avec une surface rigide, Wang *et al.* [170] s'intéressent à la gestion précise du front de contact des différentes interfaces entre liquide, air, et solide. Pour cela, au sein d'un simulateur employant une représentation de la surface par *levelset* et la méthode de projection de Chorin, ils recourent à la définition d'une surface virtuelle engendrant une courbure artificielle pour émuler les forces de tension

à l'œuvre, à l'endroit de la triple interface.

La prise en compte des forces n'est toutefois pas directe, mais comme dans la méthode précédemment décrite, elle est dépendante du saut de pression à l'interface entre air et liquide, ce saut étant corrélé à la courbure locale et au coefficient de tension de surface. La prise en compte de ces forces est donc réalisée par l'adjonction de conditions aux limites de Neumann durant la projection garantissant l'incompressibilité du fluide. La courbure artificielle, et donc la position de la surface virtuelle, est dictée par la relation d'équilibre existant entre les interfaces mises en œuvre, et qui est connue puisque dépendante des différents matériaux simulés et de leurs affinités. Le principal travail effectué correspond donc à la création du *levelset* virtuel capable de présenter la bonne courbure à l'interface. Cette procédure se révèle relativement laborieuse, surtout dans le cas où la surface solide de contact du liquide est courbe, et non plane.

Néanmoins, cette méthode permet d'obtenir des effets visuels très intéressants, où l'affinité des liquides avec les solides peut être très différenciée, et très réalistes. Il est possible d'étendre encore le réalisme en considérant un modèle de contact dynamique, qui tient compte de l'état de mouillage du solide pour en modifier dynamiquement l'affinité : les gouttes ont ainsi naturellement tendance à emprunter les trajectoires déjà mouillées par les gouttes précédentes, là où la friction du solide se révèle moins forte. Un autre effet, plus inattendu, est la présence d'une capillarité implicite, dès lors que les solides présentent des caractéristiques hydrophiles. Les surfaces de liquides présentent alors simultanément la formation d'un ménisque. Cependant, ces simulations restent très coûteuses à cause de la nécessité impérative d'une très fine résolution pour garantir une bonne capture de l'interface.

2.3.4 Interaction avec des objets externes

Afin de donner vie aux fluides dans une scène, il faut qu'une interaction automatique avec les solides, déformables ou non, soit possible. Dans les simulations Eulériennes précédentes, une réelle interaction n'est jamais prise en compte et seuls sont possibles des couplages dans un seul sens. Ainsi, la réaction du fluide à des mouvements prédéterminés d'objets externes est possible, et relativement simple, en introduisant des conditions aux limites transitoires dans la simulation du fluide. Inversement, le fluide peut lui-même influencer des objets, comme c'est le cas dès le début de l'emploi de la simulation MAC [45], en dérivant un modèle simple de forces à appliquer à l'objet. Néanmoins, dans ce cas, les deux simulations ne sont jamais couplées, l'une des deux étant toujours prédéterminée et ne réagissant pas à l'autre.

On notera que l'on se focalise ici uniquement sur l'interaction automatique de solides Lagrangiens avec un fluide représenté de manière Eulérienne. En effet, la problématique d'une telle interaction avec les autres classes de simulation de fluide n'existe pas réellement. Dans le cas des modèles de fluides manuels, il n'est pas question de simulation automatique, et aucune interaction automatique avec des objets ne peut donc y avoir lieu. Au mieux on peut trouver une réaction aux mouvements du fluide, comme décrit dans la section 2.1.2. Dans le cas des approches Lagrangiennes, la similarité de représentation entre fluide et solide supprime tout problème de prise en compte de l'interaction.

On mentionnera également, en anticipant sur leur présentation dans la section 2.4.2, que les modèles discrets de type *Lattice-Boltzmann* sont capables de

traiter une interaction bidirectionnelle avec des solides Lagrangiens [173]. La procédure proposée pour cela consiste à tenir compte du mouvement de l'interface mobile lors de la phase d'advection des paquets de particules et à modifier le calcul de la vitesse en conséquence. Lors de cette modification, les forces et couples s'exerçant sur l'objet sont simultanément déterminés, ce qui autorise de réelles interactions.

Dans un contexte Eulérien classique, peu de techniques ont été proposées pour prendre en compte une réelle interaction avec des objets externes dont le mouvement n'est pas connu à l'avance mais est en partie influencé par les mouvements du fluide. Dans le cas où ces objets sont eux-mêmes régis par une simulation physique, une réelle interaction est pourtant une nécessité. C'est dans ce contexte que s'inscrivent les travaux décrits dans la section 3.2, qui proposent une forme d'interaction automatique du fluide avec des objets déformables. On précise à ce propos que cette méthode est antérieure à la majorité des travaux décrits ci-après, ainsi qu'aux dernières améliorations de la simulation Eulérienne de fluide, ce qui explique que l'on puisse trouver son approche relativement primitive.

Une première technique pour calculer une réelle interaction entre fluide Eulérien et solides Lagrangiens est proposée par Yngve *et al.* [185], pour le cas des fluides compressibles. Cette interaction est réalisée en imposant à l'objet des forces dérivées de la pression dynamique du fluide, et en recalculant à chaque pas de temps des conditions aux limites pour le fluide d'après la voxelisation des objets dans la grille de calcul. Afin de limiter les effets de fortes discontinuités temporelles dans la présence de ces conditions aux limites, une forme de lissage temporel est appliquée.

Takahashi *et al.* [159] suivent une approche similaire où les forces appliquées à un objet dérivent de la pression du fluide à l'interface, et où le solide fixe des conditions aux limites au fluide. Les conditions considérées sont de Neumann sur la pression, et de Dirichlet sur la vitesse. Cependant, contrairement au cas précédent, ce traitement est appliqué sans aucune forme de lissage temporel, et dans le cadre d'une simulation de fluides incompressibles.

Dans le cas d'objets rigides, Carlson *et al.* [16] proposent une technique tirant partie de cette caractéristique de rigidité du déplacement de l'objet. L'idée essentiellement développée est de superposer les solides au fluide dans le domaine de simulation en les considérant comme du fluide, et d'imposer des déplacements rigides dans les zones de recouvrement du champ de vitesse. Ceci signifie que le champ de vitesse du fluide, dans les zones de recouvrement, est en adéquation avec un déplacement légal des solides. Les mouvements des solides peuvent alors être dérivés de ceux du fluide dans ces zones, ce qui assure bien un réel couplage bidirectionnel.

Au cours d'un pas de temps, l'évolution de la simulation, illustrée dans la figure 2.19, et des solides, suit l'enchaînement suivant :

$$\nabla \cdot u^t = 0 \xrightarrow{\text{fluide}} \nabla \cdot u^* = 0 \xrightarrow{\text{interaction}} Du^{**} \neq 0 \xrightarrow{\text{rigidité}} Du^{**} = 0 \quad u^{t+1}$$

La technique commence par une mise à jour de l'intégralité des zones mobiles à l'aide d'un solveur usuel dédié au fluide ($u^t \rightarrow u^*$). Ensuite, les forces propres aux solides, comme les collisions subies durant le pas de temps, sont appliquées aux portions du champ de vitesse recouvertes par des solides. Une

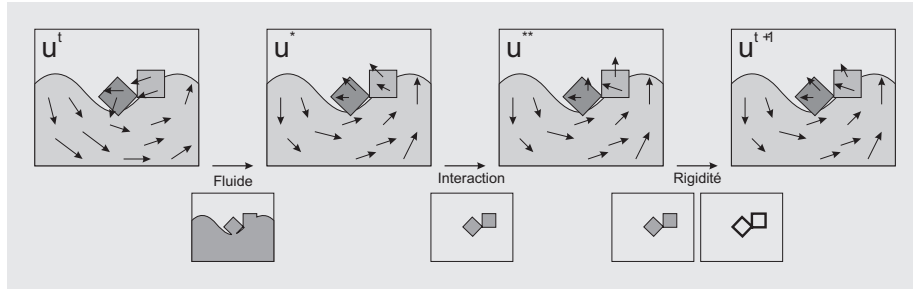


Figure 2.19: Interaction fluide – solide par méthode *Rigid Fluid*. Les solides sont traités comme le fluide dans un premier temps, avant d’introduire une force d’interaction puis une contrainte de rigidité du mouvement sur les zones de la grille qu’ils recouvrent. Les paramètres des mouvements des solides sont déterminés simultanément à la contrainte de rigidité. La portée des différents traitements est figurée dans la ligne du bas.

force engendrée par la différence de densité induite par la présence des solides dans le fluide est ajoutée. Le champ de vitesse est donc modifié dans les zones solides en accord avec toutes ces forces ($u^* \rightarrow u^{**}$). Finalement, une contrainte de rigidité $D(u) = \frac{1}{2} (\nabla u + \nabla u^T) = 0$ est imposée au champ dans les régions de recouvrement et le champ est corrigé pour y répondre ($u^{**} \rightarrow u^{t+1}$). Cette correction est directe et nécessite peu de calculs : elle consiste en une simple intégration des vitesses et couples sur les domaines des objets. Elle permet également de déterminer l’évolution des vitesses linéaires et angulaires des objets. L’interaction entre fluide et solides est bien assurée par le biais du champ de vitesse, même si aucune force n’est explicitement appliquée du fluide sur le solide. En effet, le mouvement des solides en interaction est directement recalculé depuis le champ de vitesse des zones de recouvrement entre fluide et solides.

Dans cette technique, les contraintes d’incompressibilité et de rigidité sont prises en compte de manière séquentielle, ce qui peut se révéler problématique, la correction appliquée pour satisfaire à la deuxième contrainte pouvant venir détruire, au moins partiellement, le respect de la première contrainte. Pour résoudre ce problème, Klingner *et al.* [86] proposent une prise en compte simultanée des deux contraintes, dans le cadre d’une simulation de fluide par méthode de projection de Chorin sur un maillage non-structuré s’adaptant aux frontières du domaine fluide. L’étape de projection classique est alors modifiée afin que le vecteur de pression, qui n’est autre que le multiplicateur de Lagrange destiné à satisfaire l’incompressibilité, tienne également la contrainte de rigidité à l’interface entre fluide et solides.

En adoptant une démarche de couplage similaire, Chentanez *et al.* [21] proposent de simuler des solides déformables en interaction avec un fluide. Les solides sont simulés en considérant un modèle élastique linéaire, dont l’évolution est calculée à l’aide d’un schéma implicite de Newmark, et le fluide emploie la méthode de projection de Chorin. Un réel problème couplé d’interaction entre le modèle du solide déformable et le fluide incompressible est alors posé, en considérant un problème composite liant évolution du solide et calcul de la pression dans le fluide pour en assurer l’incompressibilité. La résolution de ce problème remplace dans ce cas l’étape d’incompressibilité usuelle du fluide.

Les résultats obtenus grâce à cette technique apparaissent très réalistes. La

méthode est en outre stable, en raison de la formulation implicite employée, même en cas de pas de temps important, quoiqu'elle puisse se révéler coûteuse. En effet, le problème couplé à résoudre est d'une taille plus importante que les problèmes élémentaires constitutifs, puisque le problème composite rassemble tous les degrés de liberté liés au fluide et tous les degrés de liberté liés au solide. Par ailleurs, sa discrétisation n'est pas symétrique, ce qui en limite les performances de résolution numérique.

Structures minces Un type de structure avec lequel il est naturel de considérer une interaction avec le fluide, est celui des textiles, ou plus généralement des objets déformables fins. Cette situation est relativement complexe dans la mesure où les deux sous-systèmes sont alors très fortement interdépendants. De plus, la capture de structures fines, et même infiniment fines, par une simulation Eulérienne ne va pas sans poser problème.

Keckeisen *et al.* [80] proposent une approche simple de ce problème, dans le cadre de l'inclusion de surfaces triangulées imperméables au fluide et déformables. Préalablement à l'application de l'algorithme permettant le calcul d'un pas de temps du simulateur du fluide, la procédure suivante est exécutée : la vitesse du fluide aux sommets du maillage de la surface est enregistrée, pour calculer ensuite les forces aérodynamiques que le fluide exerce sur chaque triangle de la surface. Simultanément, en chaque cellule de la grille intersectée par un morceau de surface, la vitesse du fluide est directement manipulée afin d'en supprimer toute composante normale à la surface. La normale à la surface est déterminée à l'échelle de la grille de simulation comme la moyenne des normales des différents triangles présents dans la cellule.

À l'issue de cette procédure, des forces sont exercées du fluide sur les solides, et le fluide est approximativement contraint à ne pas traverser la surface, ce qui permet de réaliser une interaction entre les deux sous-systèmes. Cette solution simple et n'entraînant pas de modification du simulateur de fluide, reste toutefois très approximative, en particulier quant à l'imperméabilité de la surface. Elle n'est pas réellement garantie par des conditions aux limites maintenues au cours de l'intégralité de la résolution, mais uniquement revalidées de manière algorithmique au début de chaque pas de temps.

L'approche suivie par Guendelman *et al.* [58] pour traiter ce même problème de structures minces, déformables ou non comme le sont les textiles, apparaît comme nettement plus cohérente. Cette approche requiert toutefois des modifications plus profondes du simulateur de fluide, ce dernier reposant sur la méthode de projection de Chorin.

La technique présentée s'appuie principalement sur la définition d'une forme de test de visibilité dynamique tenant compte de la présence d'objets extérieurs au fluide. Si, lors d'une interpolation ou d'une différenciation pour un point de référence, une valeur quelconque est requise en un autre point voisin, un test de visibilité, effectué par lancé de rayon robuste, est utilisé. Celui-ci permet d'évincer toute valeur masquée par une surface, aussi fine soit-elle, et donc de marquer réellement une délimitation dans la résolution, là où existe une frontière physique. Dans le cas d'objets minces mobiles, si un nœud de la grille change de côté par rapport à la surface mobile, il se trouve invalidé et les quantités qui y sont stockées sont réinterpolées à partir des valeurs des nœuds voisins, en utilisant un schéma itératif de Jacobi.

Par ailleurs, la gestion des conditions aux limites de Neumann sur la pression, durant l'étape de projection assurant l'incompressibilité du fluide, est rendue plus précise par rapport aux mouvements de l'objet, afin de limiter au maximum toute fuite de fluide à travers la surface. Pour cela, une étape de prédiction de la future position de l'objet est réalisée, dans le but d'estimer très précisément sa vitesse au cours du prochain pas de temps. Les conditions de Neumann sont alors imposées sur la pression, pour que le fluide adopte exactement cette vitesse prédite à l'interface. Parallèlement, et afin de calculer les forces appliquées sur les solides par le fluide, un deuxième problème incorporant la présence du solide est résolu. En outre, dans le cas de la simulation d'un liquide représenté par un *levelset* enrichi de particules, une gestion attentive de l'évolution de la surface est réalisée.

Un des intérêts de cette méthode est que l'interaction entre fluide et solides est réalisée en imposant des conditions aux limites sur le fluide, et en appliquant des forces aux solides. Ceci permet de rendre relativement arbitraire le mode de simulation des solides et de le découpler du solveur de fluide.

2.3.5 Contrôlabilité

Dès lors que la simulation de fluide Eulérienne est détournée de sa destination initiale qu'est l'ingénierie, pour être employée à destination de l'infographie, une nouvelle problématique très concrète se fait jour : la contrôlabilité des animations. En effet, dans le cadre d'une utilisation artistique, l'objectif désiré n'est pas systématiquement une simulation obéissant parfaitement aux règles de la physique, mais plutôt une animation correspondant à l'idée préconçue d'un animateur et offrant une forte apparence de réalisme.

La problématique sous-tendant la contrôlabilité est de fournir des outils permettant l'expression de cette idée avec simplicité, sans toutefois imposer des contraintes trop drastiques aux solveurs, c'est-à-dire sans détruire l'apparence de réalisme dont ils sont les garants. Ce problème est beaucoup plus présent que dans les méthodes Lagrangiennes, et ceci pour deux raisons : d'une part, une discrétisation Eulérienne offre bien moins d'intuitivité que sa contrepartie Lagrangienne, et d'autre part la manipulation même des résultats se trouve fortement compliquée. Cette problématique de contrôlabilité n'existe pas, en tous cas pas sous cette forme, dans les simulations Lagrangiennes, pour lesquelles le moyen d'induire des mouvements se fait assez naturellement par l'adjonction de forces de contrôle appliquées aux entités Lagrangiennes. En revanche, la définition de ces forces ne constitue pas nécessairement une tâche aisée.

Selon le degré et la finesse du contrôle à exercer, plusieurs techniques différentes ont été développées. Elles peuvent être séparées en deux familles, selon le type de contrôle offert : celle opérant de manière explicite sur les variables de la simulation, directement ou à travers l'imposition de conditions aux limites, et celle mettant en œuvre des procédés indirects, par définition d'images ou de situations-clés, des *keyframes*, guidant la simulation de loin en loin.

Spécification des variables de simulation

La technique la plus directe, toujours employée, consiste à poser des conditions aux limites prises en compte naturellement par le simulateur. Suivant une

approche héritée de l'ingénierie où un phénomène bien particulier et bien quantifié est en général étudié, une spécification statique des conditions aux limites est toujours possible. Ce traitement reste cependant délicat pour deux raisons : l'intuitivité d'une telle procédure est, en général, proche de zéro et sa mise en œuvre est extrêmement fastidieuse. De plus, ce mode de contrôle laisse peu de place à l'équilibre entre contrainte externe absolue et liberté due aux calculs réalistes des mouvements.

Pour passer outre ces limitations, il est nécessaire de spécifier ces conditions d'une manière plus expressive, en relation avec des comportements de plus haut niveau. C'est l'approche proposée par Foster et Metaxas [46] qui définit un certain nombre de contrôleurs permettant d'influencer un simulateur MAC. Ces contrôleurs se présentent comme des procédures de plus haut niveau que des conditions aux limites basiques, et sont munis de paramètres explicites afin d'engendrer des comportements connus. Fontaines, vagues, tension de surface approximative, ou encore objet animé se déplaçant dans le fluide peuvent ainsi être inclus plus aisément dans une simulation. Il est à noter que la prise en compte effective des contrôleurs n'est pas toujours réalisée en spécifiant des conditions aux limites, mais parfois en imposant la modification directe de valeurs dans la simulation, ou en incluant de nouvelles forces, artificielles et conçues *ad-hoc*. En outre, il est logiquement proposé de faire varier de manière dynamique les caractéristiques intrinsèques du fluide simulé, comme sa densité ou sa viscosité.

Dans la même veine, afin de guider le parcours du fluide dans son environnement le long de trajectoires préétablies, Foster et Fedkiw [44] proposent l'inclusion d'objets virtuels se déplaçant le long de ces trajectoires, imposant par conséquent plus ou moins leurs mouvements au fluide. L'intérêt de cette approche indirecte est évidemment de ne pas spécifier une contrainte absolue au fluide, mais plutôt de l'influencer fortement, tout en présentant une approche intuitive. Rasmussen *et al.* [126] adoptent une approche semblable, en superposant à la simulation un ensemble de particules porteuses de valeurs et capables de mixer ou de substituer les valeurs qu'elles portent aux données natives manipulées par le simulateur dans leur voisinage. Les mouvements, valeurs portées, et influences de ces particules peuvent alors être préétablis simplement et intuitivement par l'animateur.

Dans tous les cas, il s'agit essentiellement pour ces méthodes de fournir à l'animateur un outil tangible et intelligible lui permettant de contraindre les mouvements du fluide, sans qu'il ait à se préoccuper d'une multitude de conditions aux limites. Ainsi des entités bien définies, et aisément manipulables, comme des nuages de particules ou des objets, se révèlent de bons candidats comme vecteurs de transfert d'une information de l'animateur vers le simulateur.

Néanmoins l'application de ces contrôles peut rester délicate, de par leur effet global et parfois drastique sur le déroulement de la simulation. En effet, d'essai en essai, rien ne garantit la préservation d'une structure jugée intéressante ou la suppression d'un phénomène parasite, l'animation étant complètement recalculée à chaque correction. Une autre approche est donc explorée par Pighin *et al.* [122] et par Schpok *et al.* [135], dans laquelle l'animateur se détache plus sensiblement de la notion de conditions aux limites liées aux variables simulées, pour manipuler directement le résultat de la simulation et les structures qui y sont présentes.

Pighin *et al.* proposent ainsi un contrôle *a posteriori* des animations par le passage d'une représentation Eulérienne à une représentation Lagrangienne, plus facilement manipulable par un animateur. Dans la représentation Lagrangienne est mis en œuvre un modèle employant des fonctions à base radiale — *Radial Basis Functions* — advectées le long de trajectoires. Ce modèle capture alors l'évolution au cours du temps des champs utiles simulés, comme une densité externe ou une température.

Une fois effectuée une simulation Eulérienne classique, une procédure d'extraction automatique fournissant trajectoires, rayons et poids des différentes fonctions est exécutée, délivrant une approximation de la simulation Eulérienne, mais sous la forme plus aisément manipulable d'un ensemble de trajectoires. Celles-ci sont ensuite déformées manuellement par l'utilisateur afin de mieux répondre à l'animation souhaitée, en altérant la position de certaines particules. Simultanément, et après chaque modification, le système effectue une correction, afin de préserver au mieux l'incompressibilité du fluide mise à mal par la modification externe arbitraire. Pour cela, une simulation SPH, analogue à celle décrite à la section 2.2.3 est employée à l'intérieur du pas de temps modifié, en associant les particules SPH aux fonctions à base radiale. En outre, une propagation dans le temps de la modification de position est réalisée par l'emploi d'une interpolation hiérarchique à base de B-Spline.

Ce système permet donc de dériver automatiquement un ensemble de manipulateurs du fluide capturant les grandes lignes de son évolution au cours du temps. La manipulation de ces contrôles extraits permet ainsi d'apporter des corrections à une animation déjà calculée et de la rendre plus conforme à une idée préliminaire.

Schpok *et al.* proposent de suivre une démarche globalement similaire. Ils ne cherchent pas à capturer le flot de fluide dans son intégralité, mais plutôt au travers de ses caractéristiques les plus importantes, sous la forme des vortex, flots uniformes, puits et sources qui y sont présents. L'idée est alors de simuler le fluide, tout en permettant à l'utilisateur d'altérer simultanément la présence et les paramètres de ces entités de plus haut niveau. Par réaction, cette modification engendre une nouvelle configuration du fluide, ce qui entraîne de nouveaux mouvements ultérieurs du fluide. À chaque pas de temps, les entités sont donc extraites automatiquement ainsi que leurs paramètres, puis éventuellement modifiées manuellement par l'utilisateur s'il le juge nécessaire. Dans le cas d'une modification, les variables de simulation sont altérées en accord avec les variations subies par le modèle analytique extrait. Les primitives sont supposées localisées dans l'espace afin de fournir un contrôle local de l'animation. Pour être utilisable, cette technique doit donc définir, pour chaque primitive de haut niveau, un modèle paramétrique facilement manipulable ainsi que les procédures de conversion entre ce modèle et une représentation discrétisée. L'interaction avec le simulateur étant supposée interactive, les modèles aussi bien que les procédures d'extraction doivent rester simples, et la simulation être de taille relativement réduite.

Points-clés

Si les modes d'interactions précédemment évoqués permettent de placer des contraintes ponctuelles sur les mouvements du fluide ou de l'influencer à dessein dans des zones bien délimitées, ils ne permettent pas, ou très difficilement, la

conception directe d'une animation dont le déroulement est imaginé à l'avance, spécialement dans le cas d'animations peu naturelles d'effets spéciaux spectaculaires. En effet, sans poser un jeu de contraintes tellement dense qu'il enlèverait toute liberté au simulateur, la mise en place d'une série de transitions précises se révèle quasiment impossible dans la pratique, tant les simulateurs sont chaotiques et les contrôles difficiles à mettre en œuvre. Dans le cas de postures du fluide connues à l'avance à divers instants, la problématique de contrôlabilité consiste donc à pouvoir spécifier des états-clés, des *keyframes*, et à laisser au simulateur le soin de créer des transitions naturelles entre eux. Cette problématique est importante dans le contexte de l'infographie, domaine où les simulations de fluides peuvent être sollicitées pour produire des animations extravagantes mais ayant l'apparence du réalisme, dans le cadre d'effets spéciaux non-réalistes.

Gestion de la situation initiale Si la plupart des solutions de *keyframing* sont conçues pour opérer tout au long de l'animation, le problème de la définition de la configuration initiale de simulation, en un sens la première des situations-clés, reste posé, spécialement dans le cas des liquides. En effet, dans des cas où la géométrie initiale du fluide est tourmentée, concevoir cette situation s'apparente à émuler manuellement sa formation depuis une situation plus simple.

Dans le cadre de la modélisation de vagues susceptibles de se briser, Mihalef *et al.* [101] proposent une approche permettant de créer cette situation initiale avec une relative facilité. Étant donné la relative homogénéité de toutes les situations initiales possibles, ils proposent que l'utilisateur recoure à une bibliothèque de profils de vagues, afin de construire le front de vagues initial qu'il désire. Une fois le front 3D construit de manière satisfaisante, un simulateur tridimensionnel classique de fluides assure le reste de la simulation à partir de cette situation initiale.

Pour peupler la bibliothèque de profils, des simulations bidimensionnelles de l'évolution de vagues sont effectuées, selon diverses conditions initiales de vagues périodiques. Cette approche permet d'inclure dans la bibliothèque des profils de vagues variés, y compris de vagues déjà en train de se briser. Si aucun profil satisfaisant n'est proposé, il est toujours possible d'enrichir la bibliothèque de manière dynamique par de nouvelles simulations. Dans la mesure où les simulations stockées ne sont que bidimensionnelles, le temps de création de la bibliothèque ne se révèle pas prohibitif. L'intérêt d'une telle approche est de permettre une création rapide et aisée de la situation initiale, en recourant à des modèles qu'un utilisateur peut facilement identifier et manipuler. Cependant, ce type de stratégie ne peut s'appliquer qu'à des situations présentant une faible variabilité, qui rendent pertinente la constitution d'une bibliothèque.

Références ponctuelles Néanmoins, il ne suffit pas de spécifier les conditions initiales de la simulation pour obtenir une animation complexe, spécialement si elle met en scène des mouvements non-naturels. Pour obtenir l'équivalent d'une animation plus traditionnelle par points-clés, deux types similaires d'approches ont été proposées : *via* un contrôle statique d'une part, ou par des procédés d'optimisation numérique d'autre part. En général, toutes les techniques opèrent en incluant des forces supplémentaires adaptées pour diriger le fluide vers le point-clé suivant, celui-ci étant souvent connu comme une forme géométrique

que le fluide est censé occuper à un point précis dans le temps. En général il n'est pas souhaité que le fluide atteigne pleinement et exactement sa cible mais plutôt qu'il en prenne globalement l'apparence, tout en conservant un comportement de fluide.

Hong et Kim [73] suivent cette approche en définissant la force de contrôle comme le gradient d'un champ de potentiel c , celui-ci étant dérivé de la géométrie de la cible à atteindre.

$$\begin{cases} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \nabla p + \nu \Delta u + f - \nabla c \\ \nabla \cdot u = 0 \end{cases}$$

Ceci signifie que la contrainte exercée sur le fluide est uniquement dépendante de sa position dans l'espace. Peu de contraintes s'exercent sur le champ de contrôle : il est simplement nécessaire qu'il atteigne approximativement un minimum sur la cible pour diriger efficacement le fluide vers celle-ci. En conséquence, le champ de contrôle reste relativement générique et peut être calculé soit par une transformée en distance de la cible, soit par résolution d'une équation de diffusion $\Delta c = 0$ adaptée, soit par toute autre procédure spécifique à une situation particulière. L'intérêt de cette approche relativement simple est qu'elle n'engendre aucun surcoût dans la simulation elle-même. En effet, les champs de contraintes ne dépendent pas du fluide, et peuvent, par conséquent, être précalculés. La prise en compte de la force de contrainte peut, pour sa part, être combinée trivialement avec les autres forces externes.

Dans le cadre de la simulation d'une fumée advectée par un fluide incompressible et non-visqueux, Fattal et Lischinski [38] proposent de spécifier les *keyframes* comme des champs de densité de fumée à atteindre aux instants-clés. Pour atteindre ces cibles, un terme de forçage est ajouté aux équations de Navier-Stokes, et un autre à l'équation de transport de la fumée. Contrairement à la méthode précédemment décrite, ces termes ne sont pas statiques et dépendent de σ^* , la prochaine *keyframe* à atteindre, comme de σ , la configuration courante :

$$\begin{cases} \frac{\partial u}{\partial t} = -(u \cdot \nabla)u - \nabla p + \nu \Delta u + f + \nu_f F(\sigma, \sigma^*) - \nu_d u \\ \nabla \cdot u = 0 \\ \frac{\partial \sigma}{\partial t} = -(u \cdot \nabla)\sigma + \nu_g G(\sigma, \sigma^*) \end{cases}$$

Dans l'équation d'évolution du fluide porteur, le premier terme ajouté est destiné à faire converger le fluide vers sa cible. Il suit le gradient d'une version lissée de la densité cible, et il est proportionnel à la densité courante. Le deuxième terme ajouté, terme de viscosité, a pour tâche d'empêcher l'accumulation d'énergie. Le terme ajouté à l'équation d'advection de la densité est inclus pour permettre de converger vers des configurations inatteignables par simple advection du champ de densité initial, c'est-à-dire non atteignable par reparamétrisation de la densité initiale. Il est défini sous la forme d'un opérateur de diffusion de l'erreur existante entre la densité courante et celle définie par la *keyframe*-cible. Il est possible d'équilibrer l'influence de ces différentes forces par le choix des facteurs ν_f , ν_d et ν_g . Par rapport à la technique précédente, le contrôle possible est donc plus fin, puisque les configurations à atteindre peuvent

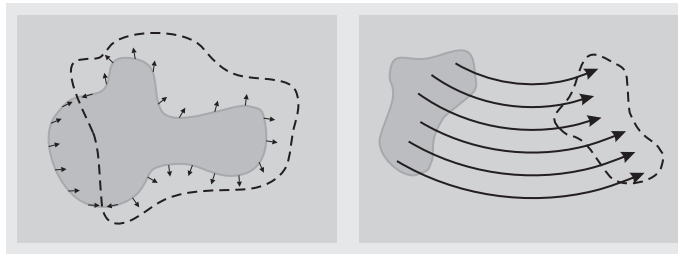


Figure 2.20: Contrôle de fumée pour atteindre une densité cible. Gauche : forces appliquées sur la frontière de fumée dans le cas de petits déplacements. Les forces sont dirigées vers le point le plus proche de la frontière de la densité cible. Droite : dans le cas de grands déplacements, une transformation rigide déplace directement le volume de fumée en court-circuitant l'étape d'advection du solveur.

être spécifiées avec exactitude sur le plan quantitatif, et que le contrôle s'opère de manière dynamique et non plus statique.

Références denses Les techniques déjà évoquées ont pour objet de faire converger le fluide entre des points-clés espacés dans le temps. Cependant, une autre situation existe, où les références à atteindre sont connues à chaque pas de temps. Dans ce cas, l'objectif est de faire approximativement suivre au fluide les mouvements prédéterminés d'un objet de référence, éventuellement déformable. Pour cela, Shi et Yu proposent deux techniques différentes, selon que le fluide à animer représente un gaz transportant une fumée [140] ou un liquide incompressible à surface libre [141].

Dans le cas des gaz, l'approche suivie consiste à appliquer des contraintes sur la vitesse du fluide afin de faire coïncider la surface de la fumée avec la surface de l'objet cible. Pour rendre la manipulation aisée, les deux objets, fumée et cible, sont représentés par des surfaces implicites discrétisées. Deux cas, dont le traitement est illustré dans la figure 2.20, sont distingués, selon que la transition à effectuer est jugée comme un mouvement important ou non, en fonction de la taille du recouvrement entre état courant et état cible.

Dans tous les cas, une procédure destinée à faire converger la surface du gaz vers la surface de la cible est exécutée en imposant des vitesses à la surface de la fumée. La composante normale de cette contrainte à la surface de la fumée assure alors la convergence proprement dite, en étant approximativement proportionnelle à l'écart existant avec la cible. La composante tangentielle est simultanément bornée pour réduire les risques de non-convergence. Notons que pour laisser le plus de liberté possible au simulateur, la totalité du bord de la fumée n'est en fait pas contrainte. Au contraire, le jeu de contraintes à employer pour converger est déterminé itérativement, en augmentant le nombre de contraintes jusqu'à satisfaire à la tolérance.

Une telle étape prend en charge de petits déplacements, ainsi que les déformations non-rigides que peut subir l'objet cible. Dans le cas d'un grand déplacement, le traitement de la composante rigide est prise en compte séparément : l'objet cible est supposé être un container rigide, déplaçant avec lui l'intégralité de la fumée qu'il contient. Des contraintes de vitesse sur le fluide, égales au mouvement rigide, sont donc appliquées sur l'intégralité du volume de fu-

mée contenu dans l'objet cible. De plus, cette fumée est directement déplacée, sans passer par l'étape d'advection normale. Bien que cette technique puisse directement s'appliquer à un solveur de fluide incompressible, il est proposé de rendre le fluide compressible, afin de regagner la stabilité éventuellement perdue par l'adjonction de contraintes. La relaxation de l'hypothèse d'incompressibilité n'est, dans ce cadre, pas toujours problématique et ce pour deux raisons : une variation de masse légère n'est que difficilement détectable pour un gaz, et les animations visées ne sont, par essence, pas réalistes.

En revanche, dans le cas de liquides, où la variation de masse est aisément perceptible en raison de la surface libre marquant une délimitation visuelle nette, cette technique ne se révèle plus adaptée. Shi et Yu [141] proposent donc une autre technique reposant sur l'introduction de deux forces, tendant l'une à restaurer la forme de la cible, l'autre à restaurer la vitesse de la cible. L'inclusion de cette dernière est justifiée par le fait qu'en cas d'accélération franche de la cible, la force dépendant de la forme n'est, à elle seule, pas suffisante pour offrir une compensation efficace. Cette force est définie comme une force visqueuse, tendant à aligner la vitesse courante avec la vitesse de la cible, en tout point du fluide. Si le recouvrement de la cible par le fluide n'est pas total, la vitesse vers laquelle le fluide est contraint correspond à la vitesse du point le plus proche de la cible.

La force destinée à faire converger la forme du liquide vers la forme de la cible est plus complexe. En effet, elle est spécialement conçue pour ne pas être altérée par l'application de la contrainte d'incompressibilité du liquide. À cet effet, elle satisfait elle-même à une contrainte d'incompressibilité. Elle est donc calculée comme le gradient d'un champ scalaire F satisfaisant à l'équation de Laplace.

$$\begin{cases} f = \nabla F \\ \nabla \cdot \nabla F = \Delta F = 0 & \text{sur } \Omega_L \\ \nabla F = f_{\partial\Omega_L} & \text{sur } \partial\Omega_L \end{cases}$$

Cette force f définie sur tout le volume du liquide Ω_L est donc intégralement déterminée par ses valeurs $f_{\partial\Omega_L}$ au bord du domaine, *i.e.* à la surface du liquide. Ces valeurs de surface sont déterminées pour pointer vers le point le plus proche de la surface cible tout en satisfaisant à une contrainte de compatibilité vis-à-vis de l'incompressibilité de la force exprimée sur l'intégralité du liquide.

Contrôle par optimisation Toutes les méthodes présentées précédemment, se sont attachées à la définition de forces adaptées au contrôle du fluide. Même si les valeurs des contraintes sont calculées dynamiquement, le type de contraintes lui-même reste complètement prédéterminé. À l'opposé de cette approche, dans le cadre de la simulation de gaz, Treuille *et al.* [167] proposent l'emploi d'un cadre d'optimisation non-linéaire générique pour diriger la simulation de manière dynamique et automatique. Dans ce cas, le problème du contrôle se trouve reformulé sous la forme de la minimisation d'une fonction objectif. Ceci permet de se limiter à considérer un modèle de contrôle très général : des générateurs de forces linéaires ou tourbillonnaires localisés dans l'espace par des Gaussiennes. L'optimiseur employé, un algorithme de quasi-Newton à mémoire limitée, se charge alors de déterminer les paramètres optimaux de ces générateurs, incluant éventuellement leur position, dans le but de minimiser la fonction objectif, c'est-à-dire d'atteindre les *keyframes* demandées.

Si le principe est simple et attractif, sa réalisation reste complexe. En particulier, l'optimiseur requiert, pour approcher le minimum de la fonction objectif, d'évaluer le gradient de la fonction par rapport aux divers paramètres possibles. Or la fonction objectif dépend dynamiquement de l'état de la simulation, qui dépend lui-même de celui des contrôles, et ce, tout au long du temps de simulation. Considérant une formalisation de l'état de toute la simulation, les auteurs présentent un moyen d'évaluer ce gradient très précisément, en le simulant parallèlement à la simulation usuelle. Pour cela, ils dérivent systématiquement et analytiquement l'évolution de tous les gradients nécessaires lors de l'emploi d'une méthode de type « *Stable Fluids* ».

Le calcul d'une animation contrôlée est donc ramené à une boucle au cours de laquelle sont affinés les paramètres des forces. À chaque itération, l'intégralité de la simulation doit être recalculée, en considérant les paramètres en vigueur. De même, le gradient de la simulation doit être simulé en parallèle pour chaque contrôle actif. Malgré des optimisations supplémentaires, scindant essentiellement l'optimisation globale en séries de sous-optimisations moins coûteuses, l'obtention d'une animation contrôlée requiert un nombre de simulations élémentaires prohibitif. On ne peut donc employer une telle technique que pour de petites grilles. En outre, la convergence du processus d'optimisation n'est nullement garantie, celui-ci pouvant tomber dans un minimum local de la fonction objectif. Néanmoins, cette approche présente l'avantage de ne dépendre d'aucune forme de force de contrainte préétablie.

Cette approche est devenue réellement utilisable grâce à son extension et son optimisation par McNamara *et al.* [99]. L'extension porte sur le domaine d'utilisation, pour qu'il englobe les liquides à surface libre représentés par un *levelset*. Cette avancée est réalisée en exprimant une forme de dérivée, requise pour le contrôle, correspondant à l'opération de redistanciation employée lors de la réinitialisation périodique du *levelset* à une fonction de distance à la surface. La difficulté de cette expression est due au fait que cette opération est essentiellement algorithmique, dépendante d'un tri par tas. En outre, un nouveau type de contrôle permettant une variation de la masse est inclus pour faciliter l'optimisation.

Parallèlement, l'utilisation de la méthode de l'adjoint permet de réduire drastiquement le volume de calculs à effectuer, au prix toutefois, d'une grande dépense de mémoire puisque l'intégralité des états de la simulation doit alors être stocké. En effet, la méthode de l'adjoint va opérer ses calculs en retraversant tous les états de la simulation à l'envers, depuis l'état final jusqu'à l'état initial. Néanmoins, elle se révèle extrêmement efficace pour prendre en compte de très nombreuses contraintes : le nombre de celles-ci peut être plus important que dans le cas de la technique d'optimisation précédente, et ceci de plusieurs ordres de grandeur. Les besoins en mémoire croissant linéairement avec la durée des animations, cette technique nécessite cependant le partitionnement temporel de la simulation ou l'emploi de méthodes plus évoluées de réduction des séquences à optimiser. Toutefois, les résultats obtenus présentent une très bonne similarité avec les *keyframes* spécifiées, et les animations peuvent réellement bénéficier d'un degré de contrôle élevé, employant de très nombreuses contraintes.

Si toutes les méthodes présentées pour autoriser une forme de contrôlabilité permettent l'obtention de résultats, aucune ne définit exactement ce que devrait être le résultat idéal. Toutes les méthodes intègrent donc un mécanisme pour guider le fluide vers sa cible, mais toujours par intention, en laissant à

la dynamique du fluide toute latitude pour assurer la transition, et pour garantir un aspect visuel « fluide » réaliste. À l'exception des méthodes reposant sur une optimisation, dans lesquelles cet aspect est englobé et masqué par la fonction objectif dictant l'optimisation, aucune méthode n'énonce précisément de problème à résoudre avec exactitude.

2.3.6 Accélération matérielle

Notoirement coûteuses en ressources de calcul, les simulations de fluide ont rapidement recouru à la puissance des processeurs graphiques programmables afin d'accélérer leur traitement. Cette utilisation du matériel graphique, s'est faite selon deux approches. La première cherche à tirer partie de l'adéquation entre discrétisation régulière de domaine de simulation et régularité des textures. En général, cette approche conduit à la transcription complète de l'algorithme de simulation en opérations effectuées par le matériel graphique, en tirant un parti maximum d'opérations graphiques pour optimiser les calculs mais également en admettant parfois des compromis pour conserver une bonne efficacité, c'est-à-dire conserver une transcription directe. Cette approche est essentiellement destinée à produire des simulations à destination de l'infographie.

La deuxième approche tend à ne considérer le matériel graphique que comme une machine SIMD à très grande bande passante et à l'utiliser dans ce sens, davantage dans un but de calcul scientifique que d'infographie pure. Dans ce cas, l'intérêt est essentiellement porté sur l'implémentation matérielle d'une technique de résolution numérique, la simulation de fluide servant plus d'exemple marquant que de but à atteindre.

L'approche générale mise en œuvre est toujours similaire quelle que soit la technique. Certaines textures stockent les données à manipuler, et des passes de rendu vers d'autres textures sont employées pour effectuer les calculs souhaités. Les codes à exécuter sont donc localisés dans les *pixel shaders*, et sont généralement appliqués à l'intégralité des textures. C'est cette approche qui garantit une grande performance, en exhibant un parallélisme massif au niveau des différents pixels. Quelle qu'en soit la destination, l'avènement d'un support matériel pertinent a été essentiellement déclenché par la possibilité de traiter et de stocker des données en virgules flottantes par le matériel graphique. En effet, la quantité d'opérations à effectuer lors de la résolution, ainsi que la difficulté à borner les valeurs manipulées, empêchent l'utilisation d'une arithmétique quantifiée. Cette section, relative à l'exécution de simulations Eulériennes ne mentionne pas les simulations discrètes de type Lattice-Boltzmann, également accélérées par le matériel graphique, mais qui sont détaillées dans la section 2.4.2 dédiée intégralement aux modèles discrets.

Dans le cas d'une destination visuelle, l'algorithme implémenté est souvent une déclinaison de l'algorithme « *Stable Fluids* », décrit dans la section 2.3.2, qui fait appel à une résolution en deux temps : il commence par simuler les termes de l'équation de conservation de quantité de mouvement, avec des méthodes respectivement adaptées à chaque terme, puis réalise une étape de projection pour garantir l'incompressibilité du fluide, ce qui nécessite la résolution d'une équation de Poisson.

Harris *et al.* [65], Wu *et al.* [181] ainsi que Scheidegger *et al.* [134] adoptent cette approche, et représentent directement la discrétisation sous-jacente à la résolution comme une texture. En fait, l'accès simultané aux données, en lecture

et en écriture, étant impossible, deux textures sont employées alternativement, l'une comme source et l'autre comme destination de chaque calcul. Dans le cas de domaines tridimensionnels, toutes les coupes bidimensionnelles de la grille selon un axe sont regroupées dans une seule texture, ceci pour permettre la mise à jour simultanée de toute la grille sans difficulté. Dans tous les cas, la régularité des calculs est pleinement exploitée, chaque passe de rendu qui effectue des calculs étant appliquée à l'intégralité de la texture contenant la grille. Les données stockées dans la grille incluent la vitesse, la pression, et toute autre quantité dont la simulation est souhaitée, comme une densité auxiliaire, ou la température locale, si la simulation du fluide est complétée par d'autres phénomènes auxiliaires.

Même si quelques différences subsistent dans la méthode de traitement de certains opérateurs de l'équation de conservation de quantité de mouvement, par discrétisation explicite ou par une méthode lagrangienne pour l'advection notamment, leurs approches restent cependant très similaires. Elles ne mettent majoritairement en œuvre que des opérations locales, et très régulières. Le cas des obstacles, brisant la régularité de la grille, est traité par une analyse préalable de la grille avant simulation, pour identifier les emplacements invalides de la grille et les conditions à leur appliquer. Avant le calcul, un remplissage dynamique de ces valeurs à partir des valeurs valides voisines est alors effectué, à l'aide d'une passe de rendu adéquate.

L'étape la plus délicate à réaliser à l'aide du matériel graphique est la résolution de l'équation de Poisson sur la pression. En effet, elle ne requiert plus seulement des opérations locales, mais également des opérations globales, affectant simultanément la totalité des cellules de la grille. Les architectures graphiques ne disposant pas d'opération d'accumulation native pour le calcul du résidu, l'implémentation d'algorithmes de résolution numériques usuels, comme le gradient conjugué, se trouve sérieusement compliquée. On préfère donc l'emploi d'algorithmes de relaxation de Jacobi, ou d'une de ses variantes, qui peuvent opérer localement et simultanément en chaque point du maillage. Le contrôle de la convergence nécessitant également une opération d'accumulation, il est soit totalement ignoré, l'algorithme de résolution effectuant un nombre prédéterminé d'itérations, soit effectué seulement de loin en loin et le moins souvent possible, à l'aide d'une opération de réduction.

Qualitativement, le gain de vitesse obtenu par le support matériel croît nettement avec la taille de la simulation : dans les plus grandes simulations pouvant être exécutées, il peut aisément atteindre un ordre de grandeur. Néanmoins, il reste très dépendant de l'implantation, ainsi que de l'architecture matérielle employée lors de l'exécution, ce qui limite fortement l'intérêt d'une comparaison quantitative. Ces simulations s'approchent toutefois d'une exécution interactive pour de petites grilles, et présentent des résultats visuellement satisfaisants tout à fait similaires à leur contrepartie non accélérée. Néanmoins, il faut noter que ces simulations ne peuvent s'exécuter sur de grands domaines en raison des limitations liées à la mémoire graphique disponible et à la taille des textures manipulables, ce qui est regrettable eu égard à leur très bon passage à l'échelle. En outre, seuls des fluides occupant la totalité de leur environnement sont simulés, ce qui réduit l'intérêt pratique de telles méthodes.

Dans un but de calcul plus scientifique, l'accent est mis sur l'implémentation d'une méthode de résolution, la simulation de fluide n'étant qu'un exemple de l'emploi de la technique. Dans ce contexte, Goodnight *et al.* [55] proposent

une implémentation matérielle d'un algorithme de résolution multi-grilles de l'équation de Poisson avec des conditions aux bords. Les différents niveaux de raffinement de la grille sont alors naturellement manipulés sous la forme de différentes textures de résolutions variables. Cette méthode est donc destinée à un domaine régulièrement échantillonné. Les opérateurs de restriction et de prolongation sont simples, effectuant essentiellement un rééchantillonnage des grilles. L'opérateur de lissage, *i.e.* la résolution du Laplacien, est implémenté par relaxation de Jacobi en employant une discrétisation centrée de l'opérateur continu. Ce solveur multi-grilles est employé pour simuler un fluide bidimensionnel en formulation vorticité-courant autour d'un profil d'aile. Bien que le domaine ne soit pas directement rectangulaire, mais un disque déformé par une transformation de Joukowski, il reste toutefois relativement simple, et régulier.

Dans le cas de domaines plus compliqués, des maillages non structurés sont souvent employés en ingénierie. En conséquence, l'approche mettant directement en bijection la discrétisation du domaine de simulation avec l'espace des textures stockant les données, n'est plus possible. L'accent est alors mis sur la représentation directe des opérateurs discrétisés, des matrices creuses, et sur les opérations afférentes requises lors de la résolution d'un système linéaire associé. Bolz *et al.* [12] décrivent dans cette optique un mode de stockage des matrices creuses, qu'ils emploient pour implémenter un solveur par gradient conjugué et un solveur multi-grilles. Krüger et Westermann [87] décrivent également une implémentation d'une méthode du gradient conjugué et d'une méthode de Gauss-Seidel, mais généralisent le type de matrices utilisables : diagonales, bandes, ou creuses. Dans ces deux travaux, les deux opérations primordiales identifiées dans ces processus sont le produit matrice-vecteur et la réduction d'un vecteur *i.e.* la combinaison de toutes les composantes par une opération arbitraire, utilisée par exemple pour calculer le résidu de la méthode itérative. Ces techniques de résolution sont ensuite employées dans des simulations bidimensionnelles de fluide. Tout comme dans les simulations à destination uniquement visuelle, il est possible d'atteindre des vitesses de simulation interactives sur des grilles de taille raisonnable.

Bien qu'à même de produire des accélérations sensibles par rapport à une implémentation purement logicielle, toutes ces approches de résolution assistées par le matériel restent limitées. Leur principal défaut est de rester confinées à l'heure actuelle à des simulations de taille modeste, en raison de limites liées à la mémoire disponible et aux contraintes de son adressage. Or c'est sur les simulations de grande taille que tout le potentiel de parallélisation est exploité. En outre, s'il est possible d'atteindre des performances élevées, c'est au prix d'un important travail de conversion des structures manipulées, et parfois, de certaines concessions sur l'algorithmique mise en œuvre. En effet, l'absence d'opérations de *scatter* et d'*accumulation* rend difficile, ou impossible, la programmation de certaines approches sans dégradation de performances.

2.4 Modèles Eulériens discrets

À la marge des méthodes Eulériennes classiques précédemment décrites, existe un ensemble de méthodes conservant le caractère Eulérien de points d'échantillonnage fixes dans l'espace, mais dont les mécanismes de simulation ne découlent pas d'une discrétisation d'équations modélisant un *continuum* de

fluide. Ces méthodes adoptent au contraire un point de vue discret. Au sein de ces méthodes se détachent deux familles : les méthodes relativement empiriques, peu ou pas justifiées physiquement, employant des automates cellulaires étendus, et les méthodes s'appuyant sur le formalisme des modèles Botzmann sur réseau, dont les capacités de simulation physiques ont été démontrées.

2.4.1 Automates cellulaires étendus

Cette première famille de méthodes regroupe des méthodes très simples, dont le principal objectif est de permettre une simulation extrêmement rapide.

L'efficacité de ces algorithmes découle de la définition même des automates cellulaires. Ceux-ci sont définis comme un maillage de cellules distribuées régulièrement dans l'espace, chaque cellule possédant un état pris dans un ensemble fini prédéterminé. À chaque pas de temps, l'état de toutes les cellules est mis à jour simultanément, en considérant un ensemble de transitions également prédéterminées. Pour chaque cellule, la transition choisie est dépendante des états des cellules de son propre voisinage. La forme de ce voisinage, toujours le même pour toutes les cellules, ainsi que les différentes transitions, déterminent alors *a priori* l'évolution globale suivie par l'automate.

En considérant une telle approche, Dobashi *et al.* [31] proposent de simuler la formation et l'évolution de nuages. L'automate résultant comprend alors quatre variables Booléennes par cellule : deux précisant l'état de la cellule, la présence d'humidité et la présence de nuage déjà formé, et deux précisant si un nuage est en train de se former ou de disparaître. La transition correspondant à l'extinction est nantie d'un délai, pour éviter des cycles de formation et de destruction trop courts. Afin d'augmenter la dynamique de formation des nuages générés, les variables sont perturbées aléatoirement dans les zones vierges de nuages formés. Ainsi, l'automate ne peut atteindre un état stable dénué de tout nuage. Dans la mesure où la variable matérialisant la présence de nuage dans une cellule est Booléenne, cette information binaire, bien que suffisante pour contrôler l'évolution, n'est pas assez précise pour permettre une visualisation de bonne qualité de la structure diaphane des nuages. Il est donc proposé de reconstruire une valeur de densité continue à partir de cette variable binaire, en employant une méthode analogue à la construction du potentiel définissant une surface implicite : la densité globale est considérée comme la somme de densités élémentaires générées en chaque cellule, un noyau radial assurant l'étalement de chaque contribution élémentaire. La densité associée à chaque cellule dérive, elle, directement de la densité moyenne calculée à partir de la variable de présence binaire sur un voisinage de la cellule.

Bien qu'avantageuse de par le faible coût associé à la seule manipulation de quelques variables binaires durant l'évolution correspondant à un pas de temps, cette méthode présente néanmoins un intérêt restreint de par son manque de contrôle. En effet, le seul contrôle direct est limité à la spécification des fonctions de densité de probabilité, contrôlant la perturbation des variables à l'origine de l'émergence des nuages. Tout autre effet, tel un déplacement des nuages, doit être obtenu par une modification des règles de transition et par l'introduction d'une anisotropie dans les voisinages considérés lors des transitions : ce type de contrôle, non séparé de la mécanique globale d'évolution, est par conséquent difficile à exercer.

Si la technique précédente ne fait émerger de variable continue que lors

d'un post-traitement à la simulation, il est possible de considérer ces variables continues directement durant l'évolution en relaxant quelque peu le formalisme des automates cellulaires. Thalmann [163] propose donc d'étendre ce formalisme, afin d'augmenter son pouvoir d'expressivité et de permettre une définition plus aisée des automates. L'extension proposée consiste à permettre l'inclusion de variables réelles dans les états des cellules de l'automate. Les transitions peuvent alors dépendre des valeurs prises par ces variables, et les modifier en retour. Ce formalisme étendu, très général, est illustré par la simulation d'une surface d'eau légèrement perturbée. La perturbation est alors transmise de proche en proche par l'automate, simulant la propagation d'une ou plusieurs vagues de petite amplitude. Cependant, la fonction de transition effective permettant cette évolution n'est pas explicitement décrite.

Ce formalisme reste ainsi très général, la seule contrainte encore héritée du formalisme des automates cellulaires stricts étant la régularité de la disposition des cellules et l'application simultanée et systématique de la même procédure de transition à toutes les cellules. Il est à remarquer qu'aucune technique de simulation précise n'est proposée, et seul est montré que dans une situation simple, un résultat visuellement satisfaisant peut être obtenu par cette mécanique d'évolution simple. Néanmoins, il reste reconnu que le développement d'une fonction de transition peut s'avérer complexe et empirique.

En employant ce formalisme étendu autorisant des quantités réelles, une simulation plus complexe est proposée par Wei *et al.* [171] afin de reproduire la fusion d'objets solides et leur évolution ultérieure sous forme liquide. La transition n'est pas modélisée comme réversible. L'automate considère deux variables Booléennes indiquant si l'état d'une cellule est liquide ou solide, et trois variables réelles décrivant l'une la fraction de liquide présente dans chaque cellule, l'autre l'énergie de la cellule, permettant de déterminer l'état de la matière présente de la cellule, et la troisième la température locale de la cellule. Les règles considérées modélisent deux phénomènes différents : le transfert de la chaleur et l'évolution de l'objet.

Le transfert de la chaleur est simulé par une diffusion simple de la température entre cellules voisines : la température locale tend à s'aligner sur la température moyenne du voisinage. Pour cela, un pas de relaxation de la température est effectué à chaque pas de temps. L'évolution de l'objet est plus complexe, requérant un grand nombre de règles différentes pour simuler la transition solide-liquide et le déplacement de matière. Ce dernier point est le plus délicat et il est géré en scindant le déplacement en deux composantes : l'une alignée avec la gravité, toujours présente si l'objet est en phase liquide, et l'autre perpendiculaire, plus complexe. Cette dernière composante assure la redistribution de la matière dans les voisinages des cellules, en corrélation directe avec la configuration du voisinage considéré. Un nombre conséquent de règles statiques est donc nécessaire pour simuler ce comportement. Une attention particulière est cependant toujours accordée à leur conception pour assurer une conservation de la matière au cours de l'évolution temporelle.

Tout en étant capable de simuler la fusion et l'évolution d'un objet sans requérir de grandes ressources, cette méthode ne permet pas d'obtenir des résultats très réalistes, en partie à cause de la modélisation très empirique et approximative du déplacement du fluide. En effet, sans prise en compte d'un champ de vitesse dans le fluide, aucune inertie n'est possible, et le déplacement se trouve essentiellement approximé par une forme de diffusion anisotrope, prin-

cipalement dirigée dans le sens de la gravité tout en étant contrainte par la configuration spatiale de l'objet. Même si l'expressivité théorique du formalisme est équivalente à celle des machines de Turing, l'expressivité pratique, accessible à un utilisateur, reste limitée, le nombre de règles et de variables croissant rapidement pour atteindre une complexité telle que des phénomènes complexes ne peuvent être modélisés.

En outre, le contenu de ces règles d'évolution doit être déterminé *ex nihilo* par l'utilisateur, sans appui possible sur les équations régissant la mécanique continue sous-jacente.

2.4.2 Modèles de Boltzmann sur réseau

Dans le même esprit de modélisation discrète, mais ayant atteint un stade beaucoup plus abouti, existent les modèles dits *Boltzmann sur réseau* ou LBM — *Lattice Boltzmann Method* — toujours assimilables à des automates cellulaires manipulant des variables réelles. L'idée sous-jacente au développement de ces modèles est de conserver la nature discrète en temps et en espace des automates cellulaires, mais d'autoriser la manipulation de quantités statistiques moyennes, directement durant la simulation, plutôt que de les calculer lors d'un post-traitement comme illustré dans la méthode de Dobashi *et al.* précédemment présentée. Cette méthode de Boltzmann, développée initialement par McNamara et Zanetti [100] en 1988, présente des résultats nettement moins affectés par un bruit statistique que des méthodes équivalentes n'employant que des variables Booléennes. Elle a été introduite dans le domaine de l'infographie par Wei *et al.* [174, 172, 173] en 2003. Il est à noter que cette classe de méthodes permet la simulation effective des équations de Navier-Stokes, et n'en constitue pas simplement une approximation visuelle.

Méthode de simulation La vision sous-tendant la technique LBM est de considérer un maillage régulier, et de faire circuler des paquets de particules de fluide sur ce maillage, entre les différents nœuds. Pour des raisons d'isotropie, le réseau employé est celui dénommé D3Q19, représenté dans la figure 2.21, qui considère un ensemble $\{e_i\}_i$ de 19 directions, et qui est plongé dans l'espace 3D. Toutefois, plutôt que de considérer les paquets de particules de manière individuelle et explicite, comme dans une modélisation totalement discrète, le système maintient en chaque nœud r et pour chaque direction de déplacement possible la densité de particules associée $\{f_i\}_i$, qui est une quantité réelle. Ce type de modélisation correspond à l'équation de Boltzmann qui exprime l'évolution d'un fluide dans l'espace des phases. À partir de ces densités sont calculables la densité de fluide et la vitesse moyenne du fluide au point de discrétisation associé à la cellule.

$$\rho(r, t) = \sum_i f_i(r, t) \quad (2.1)$$

$$u(r, t) = \frac{1}{\rho(r, t)} \sum_i f_i(r, t) e_i \quad (2.2)$$

Il est donc possible de recalculer des quantités macroscopiques significatives à partir de la simulation de paquets de particules évoluant à une échelle microscopique.

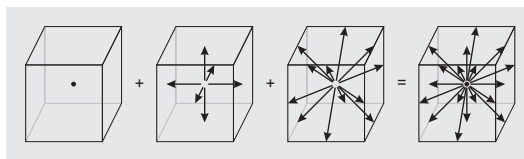


Figure 2.21: Réseau D3Q19 pour méthodes de type Boltzmann sur réseau. Les vitesses sont définies pour produire des déplacements entiers entre les nœuds du maillage durant un pas de temps.

À chaque pas de temps, les paquets de particules se déplacent sous l'effet des collisions se produisant entre les différents paquets au niveau des nœuds du maillage, et sous l'effet de leur propre inertie. Ce déplacement est transcrit en deux procédures, l'une redistribuant statistiquement les particules entre les différentes vitesses possibles en chaque nœud, sans considérer ou simuler explicitement de collision, l'autre advectant les densités sur le maillage. Afin de ne pas nécessiter une simulation explicite des collisions, le modèle BGK — *Bhatnagar, Gross, Krook* — est employé, qui consiste en un modèle de relaxation à un pas, pour lequel les densités des particules tendent à se rapprocher d'un état d'équilibre local :

$$f_i(r, t^+) = f_i(r, t) - \frac{1}{\tau}(f_i(r, t) - f_i^{eq}(r, t))$$

où τ représente le paramètre contrôlant la relaxation, et t^+ le pseudo-temps après la relaxation.

La densité à l'équilibre est calculée dynamiquement en fonction de constantes liées à la topologie de réseau lui-même :

$$f_i^{eq}(\rho, u) = \rho (A + B(e_i \cdot u) + C(e_i \cdot u)^2 + Du^2)$$

Le transport inertiel prend, lui, une forme particulièrement simple puisque les vitesses considérées sont précisément définies à partir de la topologie du réseau sous-jacent, ce qui signifie que les paquets de particules se déplacent exactement entre nœuds adjacents :

$$f_i(r + e_i, t + 1) = f_i(r, t^+)$$

Afin de parvenir à simuler des fluides de faible viscosité sans pour autant introduire des instabilités numériques, une viscosité artificielle supplémentaire est introduite. Celle-ci, découlant du modèle de sous-maille de Smagorinsky correspond à la modélisation des phénomènes d'échelles plus fines que la grille elle-même. Dans les faits, cette viscosité supplémentaire est traduite par une modification locale de la constante de relaxation employée durant la phase de traitement des collisions. Dans la mesure où cette viscosité supplémentaire dépend directement de la distribution locale de vitesse, la constante de relaxation n'est alors plus identique dans la totalité du domaine.

Un des intérêts de ce type de modèle réside dans le fait que différentes formes de conditions aux limites peuvent être incluses aisément. Ainsi, les conditions de périodicité du domaine, de frontière libre, ou encore d'obstacles imperméables alignés avec les axes du réseau ne demandent aucun traitement substantiel supplémentaire, mais seulement une gestion correcte de la topologie du réseau,

une copie de distributions de particules ou encore l'application d'une règle locale simple aux distributions des nœuds-frontières. Dans le cas d'objets imperméables, mobiles ou présentant des géométries qui ne sont pas directement compatibles avec le réseau sous-jacent, une procédure légèrement plus complexe est employée. Celle-ci n'est plus purement locale mais considère plusieurs nœuds de part et d'autre de la frontière entre fluide et solide. Néanmoins ce traitement reste toujours à une échelle locale et le couplage entre fluide et solide reste unidirectionnel, le solide imposant des conditions aux limites au fluide [174, 172]. Pour autoriser cette réelle interaction, une procédure différente est proposée [173] : pour les cellules localisées à l'interface entre fluide et solide, l'étape d'advection des densités est modifiée afin de tenir compte du mouvement de l'interface. Ce faisant, la vitesse du fluide est modifiée et cette modification est employée pour calculer les forces et les couples s'exerçant sur l'objet.

Simulation hybride Ce type de méthode n'est pas cantonné à la simulation des équations de Navier-Stokes, mais peut également être appliqué à la simulation d'autres modèles proches, tel celui gouvernant la propagation des vagues en eaux peu profondes. Thürey *et al.* [164] proposent ainsi une modélisation hybride du fluide, combinant deux simulations de type LBM, avec un modèle tridimensionnel évoluant selon les équations de Navier-Stokes pour un fluide à surface libre, superposé à un modèle bidimensionnel s'appuyant sur les équations de propagation de vagues en eaux peu profondes. Il devient ainsi possible de simuler à peu de frais de larges étendues d'eau, grâce à la simulation bidimensionnelle, sans pour autant imposer de restriction aux mouvements du fluide, grâce au recours à une simulation tridimensionnelle si nécessaire.

Pour faire fonctionner cette simulation hybride, des conditions aux limites sont imposées dans chacun des modèles dans l'espace correspondant au recouvrement des deux simulations. Elles sont chacune calculées à partir du modèle opposé, en tenant compte de corrections d'échelles pour rendre compatibles les quantités transférées d'une simulation à l'autre : hauteur du fluide et distribution des vitesses.

Un troisième modèle est par ailleurs adjoint à la simulation hybride du corps du fluide, pour gérer la création et la propagation de gouttes de petites tailles. Ces gouttes, supposées sphériques, sont gérées comme des entités Lagrangiennes supplémentaires voyageant dans l'air environnant sous l'effet de la gravité et d'un modèle de friction, et sont générées d'une manière stochastique en relation avec le degré de turbulence de petite échelle mesuré dans les simulations LBM du corps du fluide. Ces gouttes ne transportent pas de matière, et restent une adjonction purement visuelle à la simulation.

De par leur filiation avec les automates cellulaires, les méthodes LBM présentent des propriétés fortes de régularité de traitement pour toutes les cellules du réseau ainsi que des propriétés de localité des données nécessaires aux calculs effectués. En outre, ces calculs restent simples. Elles représentent donc de bonnes candidates pour être accélérées par le matériel graphique. L'implantation matérielle de cette méthode de calcul ne présente pas de difficulté majeure dans le choix d'une organisation des données, le réseau cubique employé se projetant naturellement sur un ensemble de textures bidimensionnelles, les différentes distributions étant regroupées dans différents ensembles de textures. Li *et al.* [91] présentent en détails une telle implémentation matérielle sur une architecture

à programmabilité limitée, ainsi qu'un moyen de garantir l'absence de dépassement de capacité des types de données bornés disponibles. Sur une architecture plus moderne capable de manipuler une arithmétique à virgule flottante, l'implémentation est plus aisée et l'accélération observée lors du passage à une version accélérée matériellement peut atteindre un ordre de grandeur sur de grands réseaux [173]. Compte tenu de la rapidité intrinsèque de la méthode, des grilles de tailles significatives peuvent alors être simulées interactivement.

Simulation sur des domaines courbes Dans les approches précédemment décrites, le réseau employé, D3Q19, illustré dans la figure 2.21, est tridimensionnel et régulier. Au contraire, Fan *et al.* [37], pour simuler un fluide bidimensionnel défini sur une surface courbe maillée plongée dans un espace 3D, proposent d'employer directement le maillage de la surface comme réseau sous-jacent à la simulation. Pour pouvoir adopter cette approche, deux problèmes sont pris en compte : l'irrégularité topologique du voisinage de chaque nœud, et la mise en correspondance des vitesses dans tout un voisinage.

Pour pallier l'irrégularité topologique du réseau, la méthode *Unstructured LBM* est employée. Elle permet de découpler topologie du voisinage et distributions des vitesses employées dans le modèle. Un modèle bidimensionnel isotrope composé de neuf vitesses, défini dans l'espace tangent en tout nœud constitue ainsi la base du modèle employé. Bien que d'expressions différentes et légèrement plus complexes car faisant intervenir le maillage lui-même, les équations régissant l'évolution des différentes populations de particules restent cependant toujours locales, comme dans le modèle LBM régulier. Les vitesses étant toutefois définies dans un espace tangentiel à la surface en chaque sommet du maillage, une procédure est nécessaire pour mettre en corrélation les vitesses définies en des points différents du maillage. Elle s'appuie sur la projection locale des voisinages dans un espace commun et sur la réorientation subséquente des différentes directions de propagation pour retrouver un voisinage plan et des orientations communes. Les procédures d'évolution temporelle déduites de la méthode Unstructured LBM sont ensuite appliquées. Cette procédure en deux temps est appliquée à toutes les cellules du réseau indépendamment.

Outre sa performance, découlant de la modélisation discrète LBM sur laquelle elle s'appuie, cette méthode présente l'avantage de pouvoir être appliquée à n'importe quel maillage triangulaire, quelle que soit sa topologie. En effet, les opérations employées, uniquement locales, ne nécessitent aucune paramétrisation globale pour mettre en correspondance les vitesses définies aux différents nœuds.

2.5 Problématiques périphériques à la simulation

En infographie, outre la simulation directe des fluides eux-mêmes, deux problématiques connexes gravitent autour du domaine de la simulation des fluides. La première concerne l'emploi de ces modes de simulation de fluides comme base de résolution de nouveaux problèmes, ou de la simulation de phénomènes connexes ne mettant pas en jeu les seuls fluides. La deuxième est fondamentalement liée à la destination finale des simulations qu'est la production d'images, et concerne donc le rendu.

2.5.1 Emploi subordonné de simulations

Une preuve de la maturité à laquelle est parvenue la simulation du fluide en infographie, particulièrement dans le cas des simulations Eulériennes, est son utilisation comme support d'autres techniques ou en tous cas comme composant de simulations plus larges, bien au-delà de sa propre finalité.

Les deux principales applications périphériques des simulations de fluide, la simulation de feu ou d'explosions incandescentes, et la formation de nuages, restent toutefois très proches d'une finalité brute de dynamique des fluides.

Simulation de combustions et de formations de nuages Pour simuler des explosions, Feldman *et al.* [41] emploient une modélisation à base de fluides incompressibles et d'un jeu sur la divergence, pour modéliser l'expansion initiale des gaz. Cette modélisation est adoptée car elle est plus efficace que la simulation d'un fluide réellement compressible. En sus, une simulation couplée de la température du fluide est réalisée, et un système de particules est employé pour simuler la dispersion du combustible et de la suie. Nguyen *et al.* [113] emploient pour leur part un simulateur de fluide incompressible comme base de simulation de flammes. En fait, deux simulations couplées à leur interface sont utilisées, dans lesquelles les fluides représentent le comburant non encore oxydé et le milieu environnant. La *Ghost Fluid Method*, décrite dans la section 2.3.3, est employée pour gérer avec précision la discontinuité des fluides à l'interface. Afin de pouvoir simuler des réactions plus complexes entre plusieurs réactifs transportés par un fluide incompressible, Ihm *et al.* [75] proposent de simuler effectivement les concentrations de ces réactifs dans un fluide porteur en mouvement. Les espèces chimiques considérées subissent les effets d'une advection par le fluide, d'une diffusion, ainsi que des réactions possibles en fonction d'une équation-bilan et d'une vitesse de réaction données. En retour, la température du fluide, également simulée, est affectée par la réaction chimique, ce qui autorise une simulation couplée du fluide avec la réaction. Beaucoup plus simplement, et sans couplage, Stam et Fiume [155] proposent, pour simuler des feux ou des phénomènes gazeux, d'employer un simulateur cinématique simple afin de déplacer des parcelles Gaussiennes de fumée ou de flammes et de les diffuser simultanément.

La simulation de phénomènes moins violents, comme la formation et l'évolution de nuages, tire également bénéfice de la simulation de fluide. Dans ce cas, l'évolution de la quantité de vapeur et la température, toutes deux requises pour faire correctement apparaître la condensation et donc les nuages, sont simulées. La simulation de fluide sous-jacente peut y être relativement simple, soit en utilisant un automate discret qui simule un fluide compressible comme proposé par Miyazaki *et al.* [104], soit en recourant à une simulation complète de fluide incompressible par différences finies, comme le proposent entre autres Harris *et al.* [65]. Il est à noter que c'est dans ce cadre de simulation de nuages qu'apparaît la première simulation de fluide Eulérienne à destination de l'infographie, proposée par Kajiyama et von Herzen [76]. Bien qu'elle soit fortement restreinte en terme de taille par les capacités de calcul alors disponibles, elle considère déjà une simulation de fluide incompressible non-visqueux, avec simulation conjointe de la température et de la vapeur d'eau.

Dans tous ces cas, l'objectif se résume à introduire dans une simulation de fluide standard des termes de forçage liés à l'application spécifique désirée, afin

d'induire des transports de fluide plus réalistes que dans le cas d'un fluide inerte et isolé ou d'un mouvement spécifié d'avance par un animateur. Par ailleurs, les fumées, flammes, suie, nuages ou autres, prennent alors naissance naturellement : il est ainsi possible pour un animateur d'obtenir des animations très réalistes sans un travail manuel pénible.

Autres applications Néanmoins, ces applications restent très proches, même dans leur finalité, d'une simulation de fluide classique. Quelques usages plus insolites existent cependant. Dans un but de rendu non-photoréaliste, et plus précisément de création interactive de tableaux numériques, Baxter *et al.* [9] emploient une simulation tridimensionnelle Eulérienne des équations de Navier-Stokes à forte viscosité. La simulation se charge de gérer le comportement de la peinture appliquée sur une toile. La surface libre de la peinture est capturée par une méthode de type *levelset*, complétée par un lissage ainsi que par quelques corrections adaptées à la simulation d'un fluide très visqueux. L'interaction avec l'utilisateur se fait par imposition de conditions aux limites de Dirichlet sur la vitesse, en fonction des mouvements du pinceau virtuel. L'interactivité étant essentielle dans cette application, la simulation emploie à la fois une méthode de résolution stable et de petits domaines de simulation.

Hadap et Magnenat-Thalmann [62] proposent une autre application, insolite, à savoir la simulation des mouvements d'une chevelure comme ceux d'un *continuum*, en s'appuyant sur une simulation de fluide SPH. La chevelure est donc considérée comme un milieu fluide, à densité variable de cheveux, celle-ci faisant directement varier la pression. Cependant, les cheveux restent modélisés indépendamment et sont discrétisés en chaînes cinématiques, les points de discrétisation étant associés aux particules SPH. Ces derniers reçoivent alors des forces de la simulation de fluide, qu'ils sont à même de transférer aux cheveux modélisés unitairement. Ceci permet de simuler les interactions entre cheveux à plus haut niveau et plus efficacement que dans un cas de modélisation explicite des collisions. En outre, une deuxième simulation de fluide, également SPH et représentant l'air environnant, est couplée à la simulation des cheveux, par un simple terme de viscosité, pour susciter de réelles interactions entre cheveux et atmosphère.

Une dernière application peu classique est proposée par Nixon et Lobb [114], qui tirent profit d'une simulation de fluide à une fin d'animation d'objets déformables. Il est en effet proposé de simuler un fluide compressible, sur une grille régulière, à l'intérieur d'une enveloppe masses-ressorts. Les deux simulations sont couplées : la surface impose alors des conditions aux limites de Dirichlet au fluide interne, pendant que celui-ci exerce une pression sur les masses de la surface. L'intérêt d'une telle modélisation est de permettre une conservation approximative de la masse de l'objet et une vraie dynamique interne, grâce au simulateur de fluide volumique. Simultanément, il est possible, en faisant varier les paramètres de la simulation de surface, d'obtenir des comportements dynamiques relativement différents.

2.5.2 Spécificités du rendu de fluides

Si l'animation de fluides a reçu une attention particulière de la part de la recherche en infographie, le rendu de ces fluides n'a assurément pas engendré autant de travaux dédiés, ce qui s'explique sans nul doute par le fait que ces

fluides ne présentent pas de caractéristiques intrinsèques qui les différencient d'autres objets communément rendus en infographie. Ceci ne préjuge toutefois aucunement de la complexité et du coût des procédures de rendu elles-mêmes. Le cas du rendu de nuages ou de fumées, objets volumétriques de densités variables et provoquant des dispersions multiples de la lumière, l'illustre très bien, puisque les phénomènes mis en jeu dans la représentation d'un verre fumé sont les mêmes sans pour autant qu'il s'agisse d'un fluide. À ce titre, chercher à détailler toutes les techniques employées pour le rendu de fluides revient à évoquer tout le spectre du rendu lui-même. Il est possible de distinguer deux classes particulières de rendu des fluides en infographie, selon le rapport d'importance entre la représentation de la surface du fluide et son volume.

Dans le premier cas, où la surface prédomine comme dans le cas d'un liquide telle de l'eau claire, c'est la représentation géométrique de celle-ci qui est essentielle, le rendu du volume lui-même pouvant alors être considéré comme bien maîtrisé car homogène. Pour les simulations particulières, l'approche canonique consiste en la définition d'une surface implicite en considérant les particules comme squelette. Dans le cas de simulations Eulériennes, une approche très restrictive dans la simulation pour maintenir un maillage triangulé a longtemps été de mise, jusqu'à l'introduction de la méthode MAC, permettant de se limiter à une représentation particulière du volume de fluide, puis à l'introduction des techniques de *levelset*. Ces dernières ont ouvert la voie à une représentation vraiment lisse donc vraiment exploitable, et sans contrainte, de la surface des liquides à surface libre.

Dans le deuxième cas, où la surface est peu définie, mais où le volume est primordial, tel le cas d'un nuage où plus généralement des gaz, l'accent est mis sur le traitement de ce volume. Selon le degré de réalisme requis, les phénomènes simulés peuvent alors s'échelonner de la simple atténuation volumique à la diffusion multiple de la lumière. Les simulations Lagrangiennes se prêtent alors plus naturellement à une approche par *splatting*, où sont directement combinées à l'écran les empreintes des particules, alors que les approches Eulériennes adoptent plus aisément une approche indirecte où un champ de densité discrétisé et simulé constitue la donnée d'un algorithme de rendu ultérieur. Dans ce dernier cas, la synergie entre simulation et rendu est généralement moins profonde que dans le cas d'un liquide à surface libre, où la surface profondément ancrée dans la simulation tout comme dans le rendu, engendre des contraintes dans les deux processus. Bien évidemment, la combinaison de ces deux cas de rendu, où la surface et le volume sont tous deux importants, est possible, comme dans le cas d'une eau trouble.

De fait, et en termes de techniques, la simulation et le rendu sont donc relativement orthogonaux. Dans certains cas, des conversions de représentation peuvent cependant être nécessaires, la plus courante étant sans aucun doute le passage d'une représentation implicite à un maillage triangulé par application d'un algorithme de *marching-cubes*. Bien évidemment, si ces conversions restent possibles, la qualité, mais surtout la performance, dictent le niveau d'intégration du rendu avec la simulation, ce qui engendre des contraintes sur la simulation et le rendu en terme de représentation du fluide, et même parfois de méthodes de simulation.

Outre le problème du rendu, d'autres problématiques se posent avec une importance particulière dans le cas du traitement de fluides. Leur manifestation particulièrement sensible dans le contexte des fluides procède généralement de

la représentation adoptée lors de la simulation et de contraintes liées à celle-ci. Ce sont elles qui sont maintenant détaillées.

Ajout de détails

Quel que soit le mode adopté, toute simulation reste coûteuse en ressources et il en résulte que seules les simulations de résolutions limitées en taille de grille ou en nombre de particules, peuvent être exécutées. Or la résolution maximale qu'il est possible d'obtenir dans la pratique n'est pas toujours suffisante dans le cadre de la production d'images de haute qualité. Plusieurs techniques ont été proposées afin d'accroître le niveau de détails visuels des simulations.

Une technique simple proposée par Kunimatsu *et al.* [88] pour densifier la géométrie de la surface d'un liquide à peu de frais, est d'appliquer un processus de subdivision, *a posteriori* de l'extraction de celle-ci. N'ayant pas affaire à une surface lisse comme celle d'un liquide, mais à une surface tourmentée de lave en fusion chargée de scories, Stora *et al.* [158] proposent une approche où la complexification est assurée par le placage d'une texture et d'une *displacement map* sur la surface d'une lave simulée par méthode SPH. La nécessité de cohérence temporelle dans les détails est alors résolue en associant un placage local à chacune des particules définissant la surface.

Cette problématique de placage d'une texture en mouvement, *i.e.* de la convection d'une texture par le fluide au cours du temps, est en fait centrale dans l'ajout de détails. Comme rapidement décrit par Sims [144], le problème principal qui émerge de l'advection d'une texture est le flou introduit dans celle-ci en raison de sa déformation grandissante. En conséquence, il préconise, plutôt que d'advecter itérativement une texture à chaque pas de temps lors d'une animation, c'est-à-dire de provoquer un rééchantillonnage cumulatif à chaque pas de temps, de faire évoluer le champ sous-jacent et de calculer la distorsion cumulée induite. Le rééchantillonnage est ensuite effectué à partir de la texture initiale et du champ de déformation cumulé, évitant ainsi les rééchantillonnages successifs. Dans la même idée d'advection de textures dans le but d'ajouter des détails fins à une animation plus grossière, Neyret [111] propose un mécanisme pour advecter des textures procédurales. L'accent est toujours mis sur la prévention de toute dissipation visuelle. Afin d'obtenir un flot très complexe sur le plan visuel, plusieurs textures différentes périodiquement régénérées sont superposées, mais en affinant leur combinaison : les poids de combinaison sont déterminés localement eu égard aux déformations engendrées par le champ. Plutôt que de combiner des résultats de fonctions turbulentes, il est proposé de combiner les paramètres de la turbulence elle-même avant d'appliquer l'algorithme de génération procédurale. Par ailleurs, les paramètres de la turbulence sont animés à l'aide d'une cascade d'énergie de Kolmogorov.

Si ces techniques permettent l'advection de textures par le fluide, leur emploi reste confiné au volume du fluide lui-même. Dans le cas de quantités définies sur la surface d'un liquide, la méthode proposée par Bargteil *et al.* [3], décrite à la section 2.3.2 peut être employée.

Problèmes liés aux surfaces implicites

Dans le cas des liquides une représentation implicite permet de définir très simplement une surface, mais un contrôle précis est loin d'être évident dans

le cas d'une définition de la surface par squelette. C'est le cas d'une partie des simulation Lagrangiennes décrites à la section 2.2 où des simulations MAC maintiennent un nuage de marqueurs pour repérer la position du fluide. En particulier, une problématique récurrente est la conservation du volume d'un liquide bordé par une telle surface implicite. Quelques travaux existent pour apporter une réponse à ce problème de variation parasite du volume.

Murta et Miller [109] et Tong *et al.* [165] proposent des approches similaires : une procédure itérative est chargée de calculer le volume des diverses composantes connexes du fluide et d'ajuster en conséquence les paramètres des noyaux de potentiel associés aux particules composant le squelette, jusqu'à converger vers un volume de référence. L'évaluation du volume repose alors sur une procédure numérique employant une subdivision hiérarchique de l'espace pour estimer le volume réel. Un problème, dans ce cas, est la détermination et la gestion des composantes connexes de fluide, sur lesquelles il faut pratiquer des gestions de volumes indépendantes. La variation des paramètres des noyaux peut alors être réalisée localement pour chaque particule comme proposé par Tong *et al.*, ou globalement pour chaque composante, comme décrit par Murta et Miller.

Si ces procédures permettent de garantir une vraie conservation de volume, elles restent extrêmement coûteuses, et ne peuvent par conséquent s'appliquer à d'importants nuages de particules. Des procédures locales, plus économiques, existent également. Desbrun et Cani [27] proposent, dans un contexte d'objets déformables, un mode de contrôle du volume où le champ unitaire associé à chaque particule du squelette varie. Cette variation est déterminée en introduisant un biais dans le champ élémentaire de chaque particule en fonction de la variation de volume mesurée autour de celle-ci, ainsi que de la vitesse de la variation. Ceci implique que le volume bordant chaque particule soit mesuré en permanence, au moins approximativement. Si cela est possible pour un objet modérément déformé, cette phase peut s'avérer ardue dans le cadre d'un fluide très mobile et sujet à de grandes déformations. Une autre approche proposée par Desbrun et Cani [29] consiste à dériver la position de la surface implicite d'un potentiel échantillonné sur une grille régulière d'une manière Eulérienne, et non directement à partir d'un squelette Lagrangien. La variation du potentiel est alors définie de manière à approcher les mouvements d'une forme-cible, tout en étant à même d'incorporer une contrainte de conservation de volume. Steele *et al.* [156] proposent plutôt une correction itérative en déplaçant les particules constitutives du squelette. La relaxation est fonction de la densité locale de particules, et tend à restaurer la densité locale initiale de la simulation.

Notons que cette problématique de garantie d'incompressibilité n'est pas présente, sous cette forme, pour les méthodes maintenant la surface par un *levelset*, puisque dans ce cas la position de la surface est simulée directement et n'est pas extrapolée à partir d'un squelette interne. Dans ce cas, l'incompressibilité est garantie lors de la simulation elle-même, en contraignant le champ de vitesse du fluide à être solénoïdal. Néanmoins, comme détaillé dans la section 2.3.2, le volume réel du fluide peut être sujet à variation en raison de la dissipation numérique.

Un autre problème des surfaces implicites générées à partir d'un squelette composé de points, est qu'elles sont par définition très lisses, et par conséquent difficilement exploitables, lorsque des détails fins, présents sur une surface, doivent être représentés. Dans le contexte d'un simulateur Lagrangien capable de gérer la fusion d'objets solides détaillés, Keiser *et al.* [81] proposent donc de

ne pas recourir à une surface habillant directement un nuage de particules SPH, mais plutôt d'adopter une représentation explicite liée à la surface implicite. Cette représentation explicite est maintenue grâce à un échantillonnage de la surface par des particules de surface orientées. Ces dernières matérialisent d'emblée la surface avec détails et entourent les particules SPH de calcul, présentes dans le volume du fluide ou du solide. La surface particulière est mise à jour en calquant ses déplacements sur ceux des particules de calcul proches, puis en subissant les diverses forces destinées à maintenir une bonne approximation de la surface. Outre cette évolution, les particules orientées de surface peuvent subir des rééchantillonnages en cas de modifications topologiques de la surface.

2.6 Bilan

Au travers de l'exposé des méthodes de simulation de fluide, il apparaît que l'évolution subie a suivi la même logique que dans d'autres domaines de l'infographie : d'abord, une modélisation simple et empirique des phénomènes dominants a été réalisée. Puis, devant le manque, à la fois de généralité et d'automatisme de ces méthodes, un glissement vers la modélisation des causes s'est opéré, se rapprochant par là-même de la physique et d'une réelle simulation des fluides, jusqu'à intégrer complètement cette discipline. De cette évolution se dégagent principalement trois classes de méthodes.

La première regroupe les méthodes manuelles, qui s'appuient sur des modèles spécifiques, s'inspirant ou non de la physique. Si ces méthodes, développées principalement avant l'introduction de réelles simulations, peuvent se révéler efficaces pour obtenir des résultats réalistes, elles ne restent généralement que des aides fournies à l'animateur et ne peuvent prétendre réellement au titre de simulateurs.

La simulation Lagrangienne a durablement constitué la réponse de choix au problème de la simulation de fluides en infographie car elle est très facile à appréhender et à mettre en œuvre, sous la forme de systèmes de particules plus ou moins évolués. Initialement sans automatisme, puis employant des interactions inspirées de la physique, ce mode de simulation a d'ailleurs longtemps incarné la seule voie pour simuler des liquides non contraints présentant une surface libre. En considérant des interactions fondées sur quelques considérations physiques, une forme certaine d'automatisme dans la simulation a été acquise. Le recours à des paramètres non physiques, donc sans référence mesurable, ne confère toutefois pas à ce type de simulation une forte intuitivité. La popularité de ce type de méthodes a été quelque peu estompé par les méthodes Eulériennes, moins empiriques. Mais, les méthodes Lagrangiennes ont également connu une forme de renaissance grâce à l'introduction des méthodes SPH qui apportent de réelles justifications théoriques aux simulations particulières. Il demeure cependant vrai que les méthodes Lagrangiennes recourent généralement à quelques paramètres abstraits, comme la donnée des noyaux de la méthode SPH. De plus, dans le cas où une reconstruction effective de la surface d'un liquide est souhaitée, se pose inévitablement la problématique de sa définition et de son extraction. Néanmoins, ces méthodes tirent très bien parti de la grande flexibilité d'une représentation à base de particules, pour offrir des mouvements variés et réalistes en toutes situations, tout en restant d'une approche relativement simple.

La simulation Eulérienne s'est popularisée plus tardivement, car elle nécessite de plus grandes ressources en mémoire et en capacité de calcul, et se révèle moins intuitive. Ces facteurs expliquent pourquoi l'approche Eulérienne n'a d'abord été employée que pour des simulations simplifiées ou fortement contraintes dans leurs mouvements. Elles ont toutefois connu un essor majeur avec la simulation des équations de Navier-Stokes tridimensionnelles pour un liquide à surface libre en recourant à la méthode MAC, puis avec l'introduction de la représentation de la surface par un *levelset* advecté par le fluide. L'emploi de cette dernière technique marque alors un réel essor de la simulation de fluide Eulérienne, en proposant une solution particulièrement attractive sur le plan visuel. Les simulations Eulériennes sont également attractives par le degré d'automatisme poussé, voire total, qu'elles présentent dans le cas où les conditions aux limites disponibles sont adéquates. De plus, elles restent intuitives grâce au nombre restreint de paramètres physiques bien définis alors employés, pour lesquels des références concrètes existent. Il est à noter que les simulateurs Eulériens discrets de type LBM sont en forte parenté avec les modèles plus classiques, tout en étant plus rapides, au détriment toutefois de leur intuitivité. Initialement cantonnés à la simulation des gaz, sans surface, ces modèles permettent désormais la simulation de liquides à surface libre. De ce fait, ils commencent à devenir compétitifs en termes de résultats visuels face aux modèles de simulations Eulériens plus classiques.

À l'issue d'une sélection nécessairement subjective, les différents travaux que l'on peut considérer comme les plus marquants dans l'évolution de chaque approche sont regroupés dans la figure 2.22.

Si les deux classes de méthodes, Eulérienne et Lagrangienne, autorisent des simulations réalistes, le coût en est généralement plus élevé dans le cadre Eulérien, en ressources nécessaires comme en complexité de méthodes à appliquer. Cependant, alliée à une représentation de la surface évoluée par *levelset*, cette approche génère des résultats plus convaincants, en termes de surface produite comme en termes des mouvements du fluide.

Quel que soit le mode de simulation du fluide, nombre de problématiques périphériques ont également été abordées pour lesquelles des solutions satisfaisantes ont été apportées, telles l'extension des propriétés du fluide, l'interaction automatique avec des objets externes, ou encore la contrôlabilité. L'efficacité des méthodes Eulériennes a également été accrue par le recours à des simulations plus stables et plus précises, même si elles restent coûteuses. Pour les deux types de simulations, il est d'ailleurs possible, dans le cas de solveurs pas trop complexes et de petites simulations, de recourir au matériel graphique pour accélérer la résolution, et ainsi d'obtenir des simulations interactives, même si celles-ci restent encore très éloignées d'un point de vue qualitatif des simulations précalculées les plus poussées.

Le fait que ce type de simulation a dépassé le cadre strict des fluides et sert désormais de base à des applications plus vastes est la preuve, s'il le fallait, que la simulation de fluide en infographie a atteint une certaine forme de maturité, ou tout au moins une appréciable maîtrise technique.

Qu'il s'agisse des simulations Lagrangiennes ou Eulériennes, il est aisé de noter un fort glissement, naturel, vers des simulations physiques de plus en plus poussées. Celles-ci s'appuient désormais sur un recyclage de plus en plus rapide des méthodes développées dans les communautés de physique et de mathématiques : les méthodes les plus efficaces ne sont que la transcription exacte de

<i>Méthodes manuelles</i>		
1991	Composition à base de primitives	Wejchert et Haumann [176]
1993	Turbulence statistique de petite échelle	Stam et Fiume [154]
<i>Méthodes Lagrangienne</i>		
1983	Système de particules découplées	Reeves [129]
1989	Système de particules en interactions	Miller et Pearce [103]
1991	Unification modélisation liquide / solides	Tonnesen [166]
1996	Modélisation SPH	Desbrun et Cani [28]
2003	Fluide SPH incompressible	Premože <i>et al.</i> [123]
2004	Interaction fluide SPH / solides déformables	Müller <i>et al.</i> [106]
2005	Modélisation SPH unifiée liquide / solides	Keiser <i>et al.</i> [81]
	Modélisation en vorticit�	Park et Kim [118]
<i>Méthodes Eul�rienne</i>		
1984	Premi�re simulation en vitesse	Kajiya et von Herzen [76]
1986	Premi�re simulation en vorticit�	Yaeger <i>et al.</i> [184]
1990	Propagation d'ondes et heighfield	Kass et Miller [78]
1995	Navier-Stokes 2D et champ de hauteur	Chen et da Vitoria Lobo [17]
1996	Sch�ma MAC 3D	Foster et Metaxas [45]
1999	M�thode « <i>Stable Fluids</i> »	Stam [149]
2001	Repr�sentation de la surface par <i>levelset</i>	Foster et Fedkiw [44]
2003	M�thode <i>Lattice Boltzmann</i>	Wei <i>et al.</i> [174]
2004	M�thode de l'adjoint pour le contr�le	McNamara <i>et al.</i> [99]
	Interaction avec des solides rigides	Carlson <i>et al.</i> [16]
	Maillage hi�rarchique par <i>octree</i>	Losasso <i>et al.</i> [94]
2005	Interaction avec des structures minces	Guendelman <i>et al.</i> [58]
	Maillage t�tra�drique	Feldman <i>et al.</i> [42]
	Maillage mobile	Feldman <i>et al.</i> [43]
2006	Surface triangul�e non contrainte	Bargteil <i>et al.</i> [3]
2007	Vorticit� conservative sur maillage arbitraire	Elcott <i>et al.</i> [34]

Figure 2.22: Chronologie des principales avanc es de la simulation de fluide en infographie.

celles proposées dans ces deux communautés, utilisées au final pour produire des images plutôt que des données numériques. Cet aboutissement est néanmoins logique dans la mesure où l'objectif de réalisme est également la base, et ce depuis toujours, du travail de ces communautés. Cependant, l'emprunt de ces techniques ne résout pas l'intégralité des questions, puisque des problématiques annexes, d'importance pour l'infographie, subsistent, comme la contrôlabilité d'une simulation par un animateur extérieur ou la génération de surfaces visuellement réalistes pour les liquides. En outre, des adaptations s'imposent généralement, en raison notamment d'un équilibre entre performance et précision, qui diffère de celui de l'ingénierie.

Au vu des avancées significatives déjà réalisées et du niveau de qualité atteint, il est raisonnable de considérer que les techniques actuelles sont en mesure de fournir des animations convaincantes couvrant une majorité des animations les plus simples et les plus courantes. Les cas plus spécifiques et plus complexes, tels les domaines peu réguliers ou mobiles ou encore les interactions avec d'autres simulateurs présentent encore un potentiel de développement. Par ailleurs, les liquides exhibent toujours une viscosité apparente excessive, en raison de l'absence de détails très fins de la surface associée à une représentation de résolution limitée et à la dissipation numérique toujours présente.

Pour espérer apporter des solutions à ces questions et parvenir à un degré de réalisme encore plus élevé, il semble nécessaire d'assumer pleinement l'héritage mathématique des méthodes employées, afin de pouvoir en tirer un plein profit. Une approche trop pragmatique et trop utilitaire de la simulation conduirait sans nul doute à un manque d'innovation et à des solutions moins nettes et moins robustes. Bien évidemment, cela ne s'oppose en rien, dans une étape ultérieure, à la pratique d'éventuelles simplifications, approximations, ou adaptations en remettant cette problématique dans le contexte de l'infographie, mais dans ce cas avec une bonne maîtrise de l'effet de ces choix.

Les travaux menés dans le cadre de cette thèse s'inscrivent en partie dans la problématique de la simulation de l'interaction d'un fluide avec des objets déformables, pour des simulateurs Eulériens. Une technique simple, s'intégrant à un mode de simulation MAC, a été développée dans ce but. Elle est présentée, avec le simulateur qui l'environne dans le chapitre 3. Cette technique, qui ne recourt pas à une voxelisation des solides, fait partie des pionnières dans ce domaine.

De manière plus générale, l'absence d'écriture d'un problème couplé à l'interface entre liquide et solides prohibe une résolution rigoureuse de ce même problème. Or l'écriture de ce problème et sa résolution sont difficiles à mettre en œuvre dans le cas d'une simulation opérant sur une grille structurée, qui ne permet pas une représentation exacte de l'interface. Une solution possible à ce problème est d'exploiter une méthode de résolution par éléments finis, méthode de choix en ingénierie de par sa capacité à représenter des domaines arbitraires. C'est ce qui motive principalement l'étude de cette méthode, décrite dans le chapitre 5.

Chapitre 3

Simulation MAC et interaction avec des solides déformables

Ce chapitre présente une extension apportée à la simulation MAC, autorisant une interaction automatique du liquide avec des objets déformables. Avant l'exposé de cette méthode, le simulateur MAC lui servant de base est également décrit en détails ; sont présentées la méthode de résolution qu'il emploie ainsi que son implantation.

Justification du cadre de simulation L'objectif affiché, en terme de simulation, est de fournir un modèle où peuvent interagir de manière automatique un liquide et des solides. À cet effet, il est nécessaire de faire un choix quant aux modes de simulation de ces derniers. Ce choix est crucial, car il va non seulement largement déterminer les contraintes s'appliquant aux deux milieux, mais encore conditionner les données et les opérations accessibles dans les deux simulateurs indépendants, et par là-même, le mode de couplage effectif qu'il sera possible de réaliser.

À titre d'exemple, et comme évoqué précédemment dans la section 2.2 traitant des méthodes de simulation Lagrangienne, si un mode de représentation identique à base de particules est adopté pour le fluide comme pour les solides, un couplage simple peut être très aisément mis en œuvre. Mais, si cette représentation unifiée présente une simplicité indéniable dans l'interaction, elle n'autorise pas les simulations les plus satisfaisantes pour la phase fluide, en regard des solutions Eulériennes disponibles.

Le cadre de simulation traité est celui de solides déformables de taille significative, en interaction avec un liquide à surface libre. En terme d'échelle, on suppose un liquide approximativement d'un ordre de grandeur supérieur à celui des solides présents. Néanmoins, les tailles des objets et du fluide restent toujours comparables. Une telle situation induit deux contraintes : tout d'abord, le liquide est l'élément visuel prépondérant de la scène et doit être simulé du mieux possible. Ensuite, les deux entités étant de taille similaire, en tous cas d'une taille non négligeable l'une devant l'autre, une réelle interaction est nécessaire, et un couplage unidirectionnel n'est pas satisfaisant.

L'analyse de l'objectif d'interaction, laisse apparaître assez clairement que la complexité de simulation du fluide est largement supérieure à celle de la simulation des solides. Il paraît donc logique d'opter pour une plus grande latitude dans le choix du mode de simulation du fluide, pour ensuite choisir une représentation des objets et établir une forme d'interaction automatique satisfaisante. En fonction de cette démarche et de la présence d'une surface libre à simuler, le choix du mode de simulation du liquide se porte sur la méthode MAC, *i.e.* une méthode Eulérienne. Le choix adopté pour la modélisation des solides déformables sera présenté et justifié ultérieurement, en même temps que sera proposé le mode d'interaction associé.

3.1 Simulateur de fluide

Après une présentation du principe général des simulations MAC, le mode adopté de résolution des équations de Navier-Stokes sera expliqué, et l'implémentation du simulateur sera détaillée. Il est à noter que ce simulateur ayant été réalisé avant l'introduction en infographie de la majorité des techniques raffinant le processus de simulation ou la représentation du fluide, il reste relativement basique, bien que fonctionnel, comparé aux techniques actuelles devenues plus performantes.

3.1.1 Principe de la méthode MAC

Comme décrit dans la section 2.3.2, l'introduction en infographie par Foster et Metaxas [45] de la méthode MAC — *Markers And Cells* — proposée initialement par Harlow et Welch [64], a ouvert la voie aux simulations Eulériennes de liquides à surface libre, sans aucune contrainte de mouvements. La méthode MAC s'appuie sur une double représentation du fluide, d'où son nom de *Markers And Cells* : l'une, Eulérienne, destinée au calcul des champs de vitesse et de pression du fluide sur une grille fixe, et l'autre Lagrangienne, comme un nuage de marqueurs sans masse, advectés par le champ de vitesse du fluide. Ce nuage est employé pour localiser la position du liquide, changeante au cours du temps, dans l'environnement de simulation. Puisque le fluide est matérialisé par un nuage volumique de marqueurs ponctuels, aucune hypothèse de représentation ne vient restreindre les mouvements autorisés, et le fluide peut évoluer librement dans son environnement. La simulation des équations de Navier-Stokes est réalisée par une méthode de différences finies.

Comme l'illustre la figure 2.11, les deux représentations coopèrent pour faire évoluer le fluide. De la position des marqueurs, il est possible de déduire, pour le pas de temps courant, et à la résolution de la grille Eulérienne près, la position du liquide dans l'environnement. Ainsi le domaine de simulation occupé par le fluide pour le pas de temps courant est déterminé. Simultanément, la position de la surface du liquide est repérée, ce qui permet de spécifier les conditions aux limites adéquates sur la grille pour la résolution des équations de Navier-Stokes. Cette résolution permet alors de faire évoluer les champs de vitesse et de pression. Ensuite, les marqueurs du liquide sont advectés, complétant le calcul du pas de temps.

Les deux modèles du fluide sont donc étroitement couplés : les marqueurs permettent de spécifier de manière dynamique le domaine de simulation Eu-

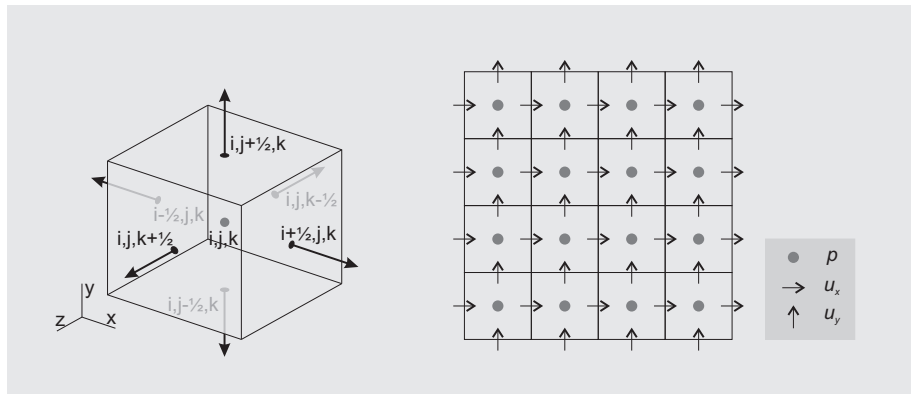


Figure 3.1: Grille décalée. Gauche : détail de la cellule (i, j, k) avec ses faces adjacentes. Les quantités scalaires sont discrétisées au centre des cellules, alors que les discrétisations des différentes composantes des vecteurs sont décalées aux centres des faces qui leur sont orthogonales. Droite : illustration des positions de discrétisation des différentes quantités pour une coupe x - y : les échantillons de la pression sont figurés par des disques, ceux de la vitesse par des flèches.

lérien, la représentation Eulérienne quant à elle détermine les mouvements du fluide et déplace les marqueurs. Par ailleurs, la méthode découple la représentation du liquide, maintenue avec finesse grâce aux marqueurs, de la résolution proprement dite des équations, effectuée avec la résolution plus grossière de la grille.

3.1.2 Grille de discrétisation

Une particularité de la méthode MAC est de faire usage d'une grille de discrétisation décalée pour la pression et les différentes composantes de la vitesse. Au lieu de discrétiser tous les champs aux mêmes points, la discrétisation est décalée, comme illustré dans la figure 3.1, avec les quantités scalaires définies au centre des cellules, et les différentes composantes des vecteurs, échantillonnées au centre des faces qui leur sont orthogonales. Cette forme de discrétisation n'est pas spécifique à la méthode MAC, bien qu'elle ait été introduite par elle [64].

Le choix de cet arrangement, qui peut paraître surprenant initialement, n'est pas arbitraire : il autorise une discrétisation des équations avec un support très compact, tout en conservant bien la masse et la quantité de mouvement. De plus, il permet de supprimer les oscillations en damier parasites de la pression, qui peuvent survenir pour un arrangement de variables coïncidant.

Le choix de cette grille et de sa résolution est crucial puisqu'il détermine le domaine dans lequel le fluide pourra évoluer au cours de son animation. En effet, aucun mécanisme de déplacement ou d'actualisation de la grille au cours de la simulation n'est employé, mais surtout, la résolution choisie détermine la finesse avec laquelle sera capturé le fluide.

Pour désigner les différents points de la grille, on utilise la notation usuelle, fixant une indexation globale de tous les points de discrétisation et employant des indices fractionnaires. Les indices entiers correspondent alors au centre des

cellules, et les demi-incréments au centre des faces, comme schématisé dans la figure 3.1.

3.1.3 Matérialisation du liquide

La détermination de la configuration du liquide dans la grille est assurée par les marqueurs sans masse advectés par le champ de vitesse. Les cellules de la grille sont partitionnées en deux groupes : celles dépourvues de tout marqueur, et celles qui en contiennent au moins un. Ce partitionnement détermine directement le domaine de simulation du liquide, mobile au cours du temps. Les cellules composant sa surface, qui sont le lieu de l'imposition des conditions aux limites relatives à la surface libre, sont déterminées comme étant les cellules pleines adjacentes par une face à au moins une cellule vide, c'est-à-dire les cellules pleines voisines à au moins une cellule vide au sens de la 6-connexité.

Il est à noter que les marqueurs déterminent, par leur seule présence, le statut des cellules de la grille. Aucune forme d'historique du statut, ou de gestion plus complexe, n'est employée. Par conséquent, il est impératif que le nuage de marqueurs soit suffisamment dense pour assurer une bonne cohérence temporelle du repérage du fluide. Pour empêcher les apparitions transitoires de vides parasites dans le liquide, le nombre moyen de marqueurs par cellule doit donc être élevé. Par ailleurs, cette contrainte impose une liaison entre densité du nuage de marqueurs et résolution de la grille : plus cette dernière est élevée, et donc plus la taille des cellules est petite, plus le nombre de marqueurs employé doit être élevé afin de maintenir un bon repérage.

Il n'est cependant requis que la distribution des marqueurs dans l'espace soit parfaitement homogène pour la simulation du fluide. Néanmoins, le champ de vitesse du fluide portant les particules étant incompressible, la distribution reste dans les faits globalement homogène, même si l'on peut constater des variations d'autant plus importantes que les mouvements du fluide sont violents. On notera que toute tentative de correction de la densité des marqueurs se révèle délicate, puisqu'une telle opération s'apparente à un rééchantillonnage de la fonction de présence du fluide, fonction qui est précisément connue au travers de la position des marqueurs eux-mêmes.

Il est également possible, et même très intéressant d'un point de vue visuel, d'introduire ou de supprimer des marqueurs au cours de la simulation. Ainsi, des générateurs de marqueurs, généralement couplés à des conditions de Dirichlet, représentent des sources de fluide. D'une manière analogue, il est possible de définir des zones de suppression des marqueurs, figurant des puits. Celles-ci sont généralement définies sur le bord du domaine pour simuler un domaine non borné.

3.1.4 Conditions aux limites considérées

La simulation numérique d'un fluide prend tout son intérêt en infographie lorsque celui-ci évolue dans un environnement complexe parsemé d'obstacles, situation dans laquelle un animateur aurait du mal à animer le liquide manuellement avec un mouvement naturel. La prise en compte de ces obstacles est réalisée en considérant un domaine de simulation de forme adaptée, domaine muni de conditions aux limites adéquates sur ses frontières. En fait, plutôt que

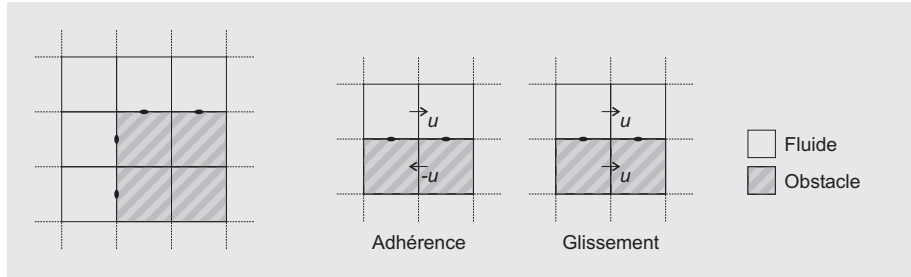


Figure 3.2: Conditions aux limites pour une grille décalée. Gauche : si les obstacles sont alignés avec les faces de la grille, la spécification d'une condition de non-pénétration revient à imposer une vitesse nulle pour les points de discrétisation de la frontière de l'obstacle — Droite : les conditions d'adhérence $u = 0$ et de glissement $u \cdot n = 0$ sont imposables en considérant les paires de points de la vitesse tangentielle de part et d'autre de la frontière de l'obstacle. La valeur de vitesse interne à l'obstacle est alors définie comme l'opposé, respectivement la copie, de la vitesse réelle du fluide.

de considérer des grilles de forme arbitraire, une grille englobant l'environnement avec ses obstacles est employée, les obstacles correspondant alors à des cellules particulières dans lesquelles le fluide ne peut pénétrer. Avec une grille décalée, la définition des conditions aux limites concernant les obstacles se révèle très naturelle, sous réserve que les frontières du domaine coïncident avec les bords des cellules. Cette coïncidence n'autorisant que des frontières localement alignées avec les axes, les obstacles courbes doivent être voxelisés.

Pour simuler des obstacles imperméables dans la grille, des conditions aux limites d'adhérence — $u = 0$ — aussi bien que de glissement — $u \cdot n = 0$ — peuvent être imposées simplement, comme illustré dans la figure 3.2. En effet, dans le cas d'un alignement des obstacles avec la grille, la composante normale de la vitesse à une frontière du domaine est localisée naturellement sur cette frontière. En revanche, ce n'est pas le cas de la composante de la vitesse tangentielle aux obstacles qui, elle, fait intervenir la paire de points de discrétisation de part et d'autre de la frontière. La prise en compte d'une condition aux limites est alors implémentée en calculant dynamiquement la valeur portée par le point extérieur au domaine en fonction de la valeur du point intérieur. Dans le cas d'une condition d'adhérence, une vitesse nulle est attendue à la frontière, et l'opposé de la vitesse du fluide est employé comme valeur externe au domaine. Dans le cas d'une condition de glissement, la vitesse est copiée pour simuler une vitesse tangentielle libre sur la frontière. La condition d'adhérence simule une friction de l'obstacle sur le fluide, au contraire de la condition de glissement.

À des fins de modélisation, il est également possible d'imposer des conditions de Dirichlet arbitraires à la surface d'un obstacle ou dans le corps du fluide. En outre, une condition particulière est employée afin de gérer une frontière ouverte du domaine. Pour simuler l'ouverture de la grille, une double condition de Neumann est adoptée sur la pression $\frac{\partial p}{\partial n} = 0$, et sur la vitesse $\frac{\partial u}{\partial n} = 0$. En fait, la condition sur la vitesse est émulée par une condition de Dirichlet dont la valeur varie dynamiquement selon la vitesse du fluide dans la cellule voisine.

3.1.5 Résolution des équations de Navier-Stokes

Pour assurer l'évolution des champs de vitesse et de pression, les équations de Navier-Stokes sont simulées sur la grille Eulérienne matérialisant le domaine de simulation. Pour résoudre ces équations, il n'est pas fait usage de l'approche directe proposée par Foster et Metaxas [45] — qui discrétise la totalité de l'équation de continuité de mouvement avant d'employer un processus de relaxation pour satisfaire à la contrainte d'incompressibilité — mais d'une variante de l'approche proposée par Stam [149] qui repose sur la méthode de projection de Chorin.

Algorithme général

La méthode de projection de Chorin [24], est une méthode de résolution des équations de Navier-Stokes incompressibles, qui sépare la simulation d'un pas de temps $u^t \rightarrow u^{t+1}$ en deux étapes, en introduisant un champ de vitesse intermédiaire u^* .

$$\nabla \cdot u^t = 0 \xrightarrow{\text{accélération}} \nabla \cdot u^* \neq 0 \xrightarrow{\text{projection}} \nabla \cdot u^{t+1} = 0$$

Le champ intermédiaire u^* est obtenu en simulant l'évolution du champ de vitesse u^t due à l'équation de conservation de quantité de mouvement, sans prise en compte de la pression, et en levant la contrainte d'incompressibilité. Le champ final u^{t+1} satisfaisant à cette contrainte d'incompressibilité est ensuite obtenu par projection du champ intermédiaire dans l'espace des fonctions solénoïdales.

Projection La projection peut être déduite de la décomposition d'Helmholtz, spécifiant que tout champ vectoriel w peut être décomposé en la combinaison d'un champ solénoïdal v et du gradient d'un champ scalaire q :

$$w = v + \nabla q \quad \text{tel que } \nabla \cdot v = 0$$

En considérant la divergence de cette relation, il vient :

$$\begin{aligned} \nabla \cdot w &= \nabla \cdot v + \Delta q \\ \nabla \cdot w &= \Delta q \end{aligned}$$

ce qui permet d'exprimer l'opération de projection P sur l'espace des fonctions solénoïdales comme :

$$P : w \rightarrow v = w - \nabla q \quad \text{avec } \Delta q = \nabla \cdot w \text{ et } \frac{\partial q}{\partial n} = 0 \text{ sur } \Gamma_D$$

où Γ_D désigne la partie du bord du domaine à laquelle sont appliquées des conditions de Dirichlet sur u , et n la normale extérieure au domaine. L'origine de ces conditions aux limites de Neumann sur la pression sera explicitée plus loin.

En appliquant cette opération de projection à l'équation de conservation de quantité de mouvement, et en considérant que $Pu = u$ et que $P\nabla p = 0$, l'équation suivante est obtenue :

$$\frac{\partial u}{\partial t} = P [-(u \cdot \nabla)u + \nu \Delta u + f]$$

qui induit l'algorithme de simulation sur un pas de temps précédemment décrit.

Plus prosaïquement, il est également possible de dériver le même algorithme en observant la discrétisation temporelle des équations de Navier-Stokes :

$$\begin{aligned}\frac{\partial u}{\partial t} + \nabla p &= -(u \cdot \nabla)u + \nu \Delta u + f \\ \frac{u^{t+1} - u^t}{\delta t} + \nabla p^t &= -(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t\end{aligned}$$

En considérant la divergence de cette équation :

$$\nabla \cdot \left[\frac{u^{t+1} - u^t}{\delta t} + \nabla p^t \right] = \nabla \cdot [-(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t]$$

Ce qui, compte tenu de la contrainte d'incompressibilité, se traduit par :

$$\begin{aligned}\delta t \Delta p^t &= \nabla \cdot u^t + \delta t \nabla \cdot [-(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t] \\ \delta t \Delta p^t &= \nabla \cdot [u^t + \delta t [-(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t]]\end{aligned}$$

En posant $u^* = u^t + \delta t [-(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t]$,

$$\delta t \Delta p^t = \nabla \cdot u^*$$

Parallèlement,

$$\begin{aligned}\frac{u^{t+1} - u^t}{\delta t} + \nabla p^t &= -(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t \\ u^{t+1} &= u^t + \delta t [-(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t] - \delta t \nabla p^t\end{aligned}$$

Ce qui, d'après la définition de u^* , amène :

$$u^{t+1} = u^* - \delta t \nabla p^t$$

En considérant la définition de u^* sous la forme $\frac{u^* - u^t}{\delta t} = -(u^t \cdot \nabla)u^t + \nu \Delta u^t + f^t$, la simulation d'un pas de temps peut alors s'écrire sous la forme :

$$\left\{ \begin{array}{l} \frac{\partial u^*}{\partial t} = -(u^* \cdot \nabla)u^* + \nu \Delta u^* + f^t \quad \text{avec } u^*(0) = u^t \\ u^* = u^*(\delta t) \\ \Delta p = \frac{1}{\delta t} \nabla \cdot u^* \quad \text{avec } \frac{\partial p}{\partial n} = 0 \\ u^{t+1} = u^* - \delta t \nabla p \end{array} \right.$$

Les conditions aux limites de Dirichlet sur la vitesse, présentes sur Γ_D , sont reportées à l'identique dans le calcul de u^* . La condition aux limites de Neumann sur la pression est liée à ce report, pour les conditions sur la composante normale de la vitesse $u \cdot n|_{\Gamma_D} = u_{\Gamma_D}$. En effet, d'après les relations précédentes :

$$u^{t+1} \cdot n = u^* \cdot n - \delta t \frac{\partial p}{\partial n}$$

Or, à cause du report des conditions aux limites de u sur u^* :

$$u^{t+1} \cdot n = u^* \cdot n = u_{\Gamma_D} \quad \text{sur } \Gamma_D$$

et donc :

$$\frac{\partial p}{\partial n} = 0 \quad \text{sur } \Gamma_D$$

Accélération Pour le calcul de l'accélération, Stam [149] propose d'utiliser une méthode de décomposition d'opérateur, ce qui amène à résoudre chaque terme de l'équation de conservation de quantité de manière indépendante et séquentielle :

$$\begin{aligned}\frac{\partial v_0}{\partial t} &= -(v_0 \cdot \nabla)v_0 && \text{avec } v_0^t = u^t \\ \frac{\partial v_1}{\partial t} &= \nu \Delta v_1 && \text{avec } v_1^t = v_0^{t+1} \\ \frac{\partial v_2}{\partial t} &= f^t && \text{avec } v_2^t = v_1^{t+1} \\ u^* &= v_2^{t+1}\end{aligned}$$

Il propose ensuite d'employer une méthode semi-Lagrangienne pour le terme d'advection, et une méthode implicite pour le terme de diffusion, dans le but de proposer un solveur inconditionnellement stable, sans contrainte sur la taille des pas de temps. L'approche qui est adoptée dans ce travail est plutôt celle employée par Foster et Fedkiw [44], qui font également usage d'une méthode semi-Lagrangienne pour traiter le terme advectif mais d'une méthode explicite pour la diffusion.

Ce choix, s'il engendre une limite à la taille des pas de temps stables, est motivé par l'application visée d'interaction avec des solides déformables. En effet, l'intérêt d'une résolution implicite, requérant la formation et la résolution d'un système linéaire, est d'autoriser un pas de temps suffisamment important pour compenser la résolution de ce système linéaire. Dans le cas de l'inclusion de l'interaction avec des objets déformables, les pas de temps choisis sont dans tous les cas petits, en raison des interactions très raides qui peuvent apparaître. Par conséquent, le mode de résolution implicite se révèle peu intéressant. La résolution explicite par différence centrée de la composante visqueuse présente également l'avantage d'être très simple.

En revanche, le traitement semi-Lagrangien du terme d'advection par la méthode des caractéristiques, est conservé, car il permet de ne pas introduire une discrétisation de l'opérateur d'advection qui couple les différentes composantes de la vitesse [45]. Ceci autorise alors une gestion performante des conditions aux limites, comme décrit ci-après, gestion qui serait sinon impossible, du moins plus complexe, en cas de couplage des composantes.

La gestion de la surface libre du fluide n'a pas été évoquée dans cette résolution. Sa prise en compte relevant de la définition du domaine de simulation et des conditions aux limites, elle n'intervient pas directement dans l'algorithme de résolution des équations de Navier-Stokes. La méthode de résolution des équations de Navier-Stokes choisie, pour un pas de temps, est résumée dans la figure 3.3. Il est à noter que le pas de temps fixe choisi par l'utilisateur est subdivisé de manière adaptative pour respecter la condition de stabilité CFL — Courant, Friedrichs, Levy — suivante :

$$\delta t \leq \kappa \frac{\delta x}{|u|_\infty}$$

avec $0 \leq \kappa \leq 1$ choisi par l'utilisateur, et $|u|_\infty$ la vitesse maximale atteinte dans le domaine. Cette condition spécifie que les déplacements admissibles du fluide sur un pas de temps doivent rester petits par rapport à la taille des cellules de la grille.

- A1. Résolution de $\frac{\partial v_0}{\partial t} = -(v_0 \cdot \nabla)v_0$ avec $v_0^t = u^t$ par la méthode des caractéristiques
- A2. Résolution de $\frac{\partial v_1}{\partial t} = \nu \Delta v_1$ avec $v_1^t = v_0^{t+1}$ par différences centrées
- A3. Résolution de $\frac{\partial v_2}{\partial t} = f^t$ avec $v_2^t = v_1^{t+1}$ par la méthode d'Euler avant
— $u^* = v_2^{t+1}$
- P1. Calcul de la pression par résolution de $\Delta p = \frac{1}{\delta t} \nabla \cdot u^*$ avec $\frac{\partial p}{\partial n} = 0$ sur Γ_D par la méthode du gradient conjugué
- P2. Application de la correction : $u^{t+1} = u^* - \delta t \nabla p$

Figure 3.3: Méthode de résolution des équations de Navier-Stokes sur un pas de temps. Les étapes notées A correspondent à la prise en compte des différents termes d'accélération, et celles indiquées par P, à la projection pour satisfaire à la contrainte d'incompressibilité.

Détail des différentes étapes

Caractéristiques [A1] La méthode des caractéristiques est employée pour résoudre l'équation $\frac{\partial u}{\partial t} = -(u \cdot \nabla)u$. Cette méthode cherche à simplifier la résolution de l'équation aux dérivées partielles, en la ré-exprimant sous la forme d'une équation différentielle ordinaire le long de courbes dans l'espace des paramètres appelées caractéristiques.

Lorsque l'équation de transport à résoudre est réécrite sous la forme

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = 0$$

il devient clairement visible qu'elle coïncide avec la dérivée convective, ce qui équivaut à spécifier un transport du champ sous sa propre vitesse, sans autre action :

$$\frac{Du}{Dt} = 0$$

Les caractéristiques de l'équation sont par conséquent déterminées par les trajectoires des particules, et la vitesse portée par les particules est constante au cours du temps.

Soit $X(x, s; t)$, la position à l'instant t de la particule se trouvant en x au temps s . Cette fonction décrivant la trajectoire d'une particule obéit à l'équation différentielle ordinaire suivante :

$$\begin{cases} \frac{dX(x, s; t)}{dt} = u(X(x, s; t)) \\ X(x, t; t) = x \end{cases}$$

Sur le pas de temps $[t\delta t, (t+1)\delta t]$, en posant $\chi(x) = X(x, (t+1)\delta t; t\delta t)$, il est alors possible d'approximer la dérivée convective par :

$$\frac{Du}{Dt}(x) \approx \frac{u^{t+1}(x) - u^t \circ \chi(x)}{\delta t}$$

où la variation de la vitesse portée par une particule est exprimée en considérant son déplacement au cours du pas de temps. Le schéma de calcul suivant émerge alors :

$$u^{t+1}(x) = u^t \circ \chi(x)$$

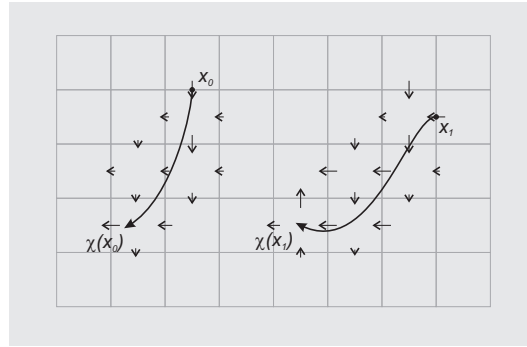


Figure 3.4: Méthode des caractéristiques : pour chaque point de discrétisation x de chaque composante de la vitesse, la caractéristique est calculée et rebroussee sur un pas de temps jusqu'au point $\chi(x)$. La valeur $u^t \circ \chi(x)$, interpolée à partir de l'échantillonnage de u^t constitue alors la valeur de $u^{t+1}(x)$.

Celui-ci se ramène intégralement au problème de suivi de trajectoires, correspondant à la résolution de l'équation différentielle ordinaire présentée précédemment. Pour chaque point de discrétisation x , la caractéristique est rebroussee sur un pas de temps, et la valeur présente au pied de la caractéristique au pas de temps précédent, $u^t \circ \chi(x)$, est transportée au point de discrétisation x à l'instant $t + 1$. Dans la mesure où le point $\chi(x)$ ne correspond généralement pas à un point de la grille, une interpolation est nécessaire au calcul de $u^t \circ \chi(x)$.

En raison de la nature décalée de la grille qui a été choisie, il est nécessaire de remonter une caractéristique pour chaque point-échantillon de chaque composante de la vitesse, comme l'illustre la figure 3.4.

Cette méthode est considérée comme une méthode semi-Lagrangienne dans la mesure où elle fait intervenir à la fois la notion de grille comme support de la discrétisation, et la notion Lagrangienne de trajectoires des particules. Elle est équivalente à une reparamétrisation de u^t en fonction χ .

Viscosité [A2] Le terme de diffusion $\frac{\partial u}{\partial t} = \nu \Delta u$ est résolu par un schéma de différences centrées explicites. Dans un souci de clarté, ce paragraphe use de la notation suivante : $u = [u_x, u_y, u_z]^T = [u, v, w]^T$. En considérant le point $(i + \frac{1}{2}, j, k)$ correspondant à un point de discrétisation de la composante u_x , les gradients discrétisés à gauche Δ_x^- et à droite Δ_x^+ , dans la dimension x s'écrivent, au temps t :

$$\Delta_x^- = \frac{u_{i+\frac{1}{2},j,k}^t - u_{i-\frac{1}{2},j,k}^t}{\delta x} \quad \Delta_x^+ = \frac{u_{i+\frac{3}{2},j,k}^t - u_{i+\frac{1}{2},j,k}^t}{\delta x}$$

Pour les deux autres dimensions :

$$\begin{aligned} \Delta_y^- &= \frac{u_{i+\frac{1}{2},j,k}^t - u_{i+\frac{1}{2},j-1,k}^t}{\delta x} & \Delta_y^+ &= \frac{u_{i+\frac{1}{2},j+1,k}^t - u_{i+\frac{1}{2},j,k}^t}{\delta x} \\ \Delta_z^- &= \frac{u_{i+\frac{1}{2},j,k}^t - u_{i+\frac{1}{2},j,k-1}^t}{\delta x} & \Delta_z^+ &= \frac{u_{i+\frac{1}{2},j,k+1}^t - u_{i+\frac{1}{2},j,k}^t}{\delta x} \end{aligned}$$

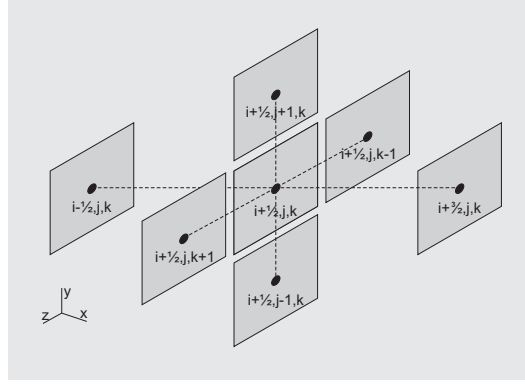


Figure 3.5: Masque de calcul du Laplacien pour la viscosité. Entrent dans le calcul d'un point, le point lui-même ainsi que ses six voisins. Illustré sur la composante u_x de la vitesse.

où δx représente le pas de la grille, qui pour plus de simplicité est choisi identique dans toutes les dimensions. Ceci amène à la discrétisation suivante :

$$\frac{u_{i+\frac{1}{2},j,k}^{t+1} - u_{i+\frac{1}{2},j,k}^t}{\delta t} = \nu \left[\frac{\Delta_x^+ - \Delta_x^-}{\delta x} + \frac{\Delta_y^+ - \Delta_y^-}{\delta x} + \frac{\Delta_z^+ - \Delta_z^-}{\delta x} \right]$$

et finalement :

$$\begin{aligned} u_{i+\frac{1}{2},j,k}^{t+1} = u_{i+\frac{1}{2},j,k}^t + \frac{\nu \cdot \delta t}{\delta x^2} [& u_{i+\frac{1}{2},j-1,k}^t + u_{i+\frac{1}{2},j+1,k}^t \\ & + u_{i+\frac{1}{2},j,k-1}^t + u_{i+\frac{1}{2},j,k+1}^t \\ & + u_{i-\frac{1}{2},j,k}^t + u_{i+\frac{3}{2},j,k}^t - 6 \times u_{i+\frac{1}{2},j,k}^t] \end{aligned}$$

Pour un point donné, les points mis à contributions dans le calcul de son évolution sont les six voisins dans la grille ainsi que lui-même, comme illustré dans la figure 3.5.

Le cas des autres composantes de la vitesse est identique, en veillant à traiter correctement le décalage des trois grilles supportant les trois composantes de la vitesse :

$$\begin{aligned} v_{i,j+\frac{1}{2},k}^{t+1} = v_{i,j+\frac{1}{2},k}^t + \frac{\nu \cdot \delta t}{\delta x^2} [& v_{i-1,j+\frac{1}{2},k}^t + v_{i+1,j+\frac{1}{2},k}^t \\ & + v_{i,j+\frac{1}{2},k-1}^t + v_{i,j+\frac{1}{2},k+1}^t \\ & + v_{i,j-\frac{1}{2},k}^t + v_{i,j+\frac{3}{2},k}^t - 6 \times v_{i,j+\frac{1}{2},k}^t] \\ w_{i,j,k+\frac{1}{2}}^{t+1} = w_{i,j,k+\frac{1}{2}}^t + \frac{\nu \cdot \delta t}{\delta x^2} [& w_{i-1,j,k+\frac{1}{2}}^t + w_{i+1,j,k+\frac{1}{2}}^t \\ & + w_{i,j-1,k+\frac{1}{2}}^t + w_{i,j+1,k+\frac{1}{2}}^t \\ & + w_{i,j,k-\frac{1}{2}}^t + w_{i,j,k+\frac{3}{2}}^t - 6 \times w_{i,j,k+\frac{1}{2}}^t] \end{aligned}$$

L'évolution des trois composantes est totalement découplée.

Forces externes [A3] Le terme $\frac{\partial u}{\partial t} = f$ concernant les forces volumiques externes appliquées au fluide est le plus simple. Pour le résoudre, la méthode d'Euler avant est utilisée, donc :

$$\frac{\partial u}{\partial t} \approx \frac{u^{t+1} - u^t}{\delta t} = f$$

et

$$u^{t+1} = u^t + \delta t \cdot f$$

en tout point de la discrétisation spatiale, pour chacune des composantes.

Projection [P1–P2] La projection consiste essentiellement en la résolution du problème suivant :

$$\begin{cases} \Delta p = \frac{1}{\delta t} \nabla \cdot u^* & \text{sur } \Omega \\ \frac{\partial p}{\partial n} = 0 & \text{sur } \Gamma_D \\ p = p_{atm} & \text{sur } \Gamma_S \end{cases}$$

où l'on a posé que la pression des cellules de surface Γ_S est égale à la pression atmosphérique p_{atm} .

Le traitement standard pour ce type de problème elliptique est appliqué, à savoir l'usage d'un solveur linéaire recourant à la méthode du gradient conjugué pour résoudre le système discrétisé.

3.1.6 Mise en œuvre du simulateur

La mise en œuvre proprement dite du simulateur et des algorithmes précédemment évoqués est maintenant détaillée, et plus particulièrement l'aspect le plus délicat, à savoir la gestion des conditions aux limites.

Comme précédemment évoqué, l'intérêt d'un simulateur de fluide est la prise en compte automatique de l'environnement de simulation. À cet effet, la démarche suivie s'attache à n'imposer aucune contrainte ni de taille ni de positionnement des obstacles dans la grille de simulation, excepté la coïncidence des frontières des obstacles avec les faces de la grille.

Le but de l'implantation réalisée est également de présenter les performances maximales. Cet objectif implique le recours à des grilles régulières, donc à des tableaux, pour stocker les différentes valeurs, afin de conserver aux calculs et aux accès-mémoires une régularité maximale. En outre, on cherche à minimiser les calculs, et en particulier à éviter autant que possible tout test. L'objectif, en terme d'implantation, est donc d'adopter une conception minimisant les défauts de cache aussi bien que les erreurs de prédiction de branchement.

D'une manière générale, la structure sous-jacente au simulateur est composée de quatre grilles tridimensionnelles scalaires, stockant les différentes composantes de la vitesse et la pression. Ces quatre grilles ne présentent pas une taille identique, mais sont dimensionnées de la manière suivante :

$$\begin{array}{l} p : n_x \times n_y \times n_z \\ u_x : n_x + 1 \times n_y \times n_z \\ u_y : n_x \times n_y + 1 \times n_z \\ u_z : n_x \times n_y \times n_z + 1 \end{array}$$

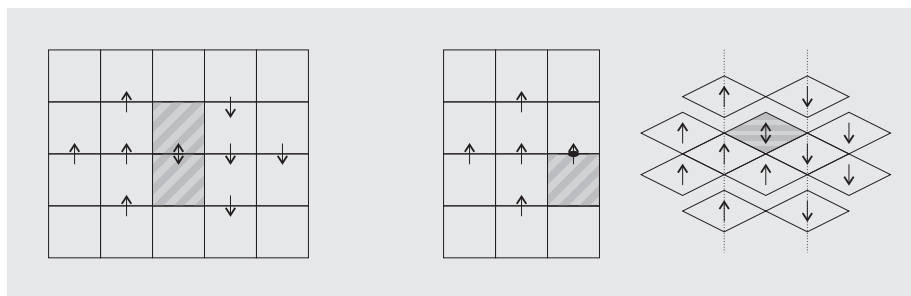


Figure 3.6: Définition ambiguë d'une valeur de la grille lors de l'application des conditions aux limites pour le Laplacien, à proximité des obstacles. La situation décrite à gauche peut être indirectement gérée en subdivisant la grille. Les deux situations de droite, l'une bidimensionnelle et l'autre tridimensionnelle, requièrent un traitement explicite adéquat.

Certaines étapes du calcul du champ de vitesse au pas de temps $t + 1$ nécessitant des valeurs non locales au pas de temps t , il est nécessaire d'employer deux jeux de grilles stockant la vitesse. La grille correspondant à la pression ne requiert, elle, pas de dédoublement en raison du mode de calcul choisi. D'autres informations sont également stockées pour optimiser l'implémentation, comme des informations d'adjacence entre cellules et faces.

Toutefois, cette recherche de performance par la régularité des calculs se fait au détriment de la souplesse de stockage et d'accès aux informations du simulateur. La régularité des calculs peut ainsi rendre extrêmement compliqué, et donc extrêmement coûteux, la prise en compte exacte de certaines conditions aux limites. Dans la mesure où l'application visée est celle de la production d'images, dont le résultat est essentiellement évalué sur le plan qualitatif et non quantitatif, on s'autorise à pratiquer quelques choix ou approximations arbitraires afin de maximiser les performances au détriment de la précision de résolution. Au vu des résultats qui sont présentés dans les sections suivantes, ces choix ne semblent pas constituer une entrave majeure au réalisme visuel. On n'est toutefois pas en mesure de réaliser une comparaison avec une implémentation de référence, et l'on ne peut donc quantifier précisément la dégradation des résultats.

Gestion des conditions aux limites

Dans une optique de rapidité et de régularité, et afin de prendre en compte les conditions aux limites, on opte pour l'approche consistant à placer dans la grille des valeurs particulières pour certains points de discrétisation internes aux obstacles proches des frontières, comme le suggèrent Harlow et Welch [64] et comme évoqué dans la section 3.1.4. Les calculs eux-mêmes opèrent alors aveuglément sur leurs données et ne tiennent plus compte de manière explicite de leur environnement : on s'appuie sur les valeurs particulières, calculées dynamiquement et injectées dans la grille aux positions adéquates pour en émuler la prise en compte. Dans ce cas, avant tout calcul susceptible d'accéder à ces valeurs émulant les conditions aux limites, il est nécessaire de les rafraîchir en accord avec les données valides les plus récentes présentes dans la grille.

Or, ces valeurs particulières sont localement dépendantes de la vitesse du

fluide. Certaines valeurs devraient alors différer selon le point en cours de calcul, comme illustré dans la figure 3.6. L'approche triviale reposant sur une mise à jour de ces valeurs dynamiques émulant les conditions aux limites, avant le calcul effectif du pas de temps et pour l'intégralité de la grille, ne peut donc être appliquée sur une structure de grille basique.

Gestion à la volée Une solution simple consiste à traiter conditionnellement les calculs en déterminant, à la volée, les valeurs relatives aux conditions aux limites. Cette approche a l'inconvénient de briser la régularité souhaitée des calculs et d'induire des tests pour analyser le voisinage. Elle entre donc en contradiction avec le type d'approche que l'on s'est fixé. De plus, elle se révèle relativement inefficace car elle nécessite de déterminer, à chaque utilisation de chaque valeur, si celle-ci doit être calculée dynamiquement ou recherchée directement dans la grille.

Gestion avec précalcul Une autre approche est alors adoptée pour gérer plus efficacement ces points problématiques. Elle consiste à traiter tous les points dont le voisinage est non-ambigu dans la grille, comme attendu, et à isoler le traitement des points ambigus hors de la grille, pour leur appliquer une procédure adaptée, ce qu'illustre la figure 3.7. Une analyse de la grille préalable à la simulation permet la classification des points entre ces deux familles. L'application de chacune des deux procédures de calcul, à chaque pas de temps, est alors réalisée de manière optimale, sans devoir analyser le voisinage de chaque point.

Détails de la gestion avec précalculs Cette approche est possible en raison de plusieurs caractéristiques de la prise en compte des conditions aux limites. Tout d'abord, au vu du mode de résolution choisi, ce problème de définition ambiguë de valeurs de grille survient essentiellement dans le cadre du calcul du Laplacien. Durant les autres étapes de la simulation, les seules valeurs requises dans les calculs sont externes aux obstacles ou définies sur leurs bords, et ne présentent aucune ambiguïté dans leur définition : elles peuvent donc être retrouvées dans la grille régulière sans problème. Ensuite, et comme décrit précédemment, la résolution du Laplacien emploie un arrangement de calcul fixe, réduit, et découplé pour chaque composante de vitesse. Finalement, les obstacles sont fixes dans la grille.

On peut donc identifier, avant la simulation, les points de la grille pour lesquels la discrétisation du Laplacien inclut des valeurs dynamiques liées aux conditions aux limites : il s'agit de tout point dont un des voisins est localisé à l'intérieur d'un obstacle ou sur la surface d'un obstacle. Cette analyse permet ainsi de partitionner les points de discrétisation en deux groupes : le groupe des points dont tous les voisins sont non-ambigus et le groupe des autres. Un calcul optimal est alors employé pour chacun des deux groupes.

Dans le cas de non-ambiguïté complète du voisinage, le calcul est réalisé directement à partir de la grille régulière, en employant le masque de discrétisation régulier précédemment décrit, les valeurs requises se trouvant localisées aux points voisins attendus dans la grille régulière. L'objectif de parfaite régularité des calculs et des accès-mémoires est ainsi atteint pour ces points.

Le cas des calculs requérant des valeurs dynamiques pourrait être résolu par analyse du voisinage lors des calculs. Néanmoins, puisque le choix des conditions

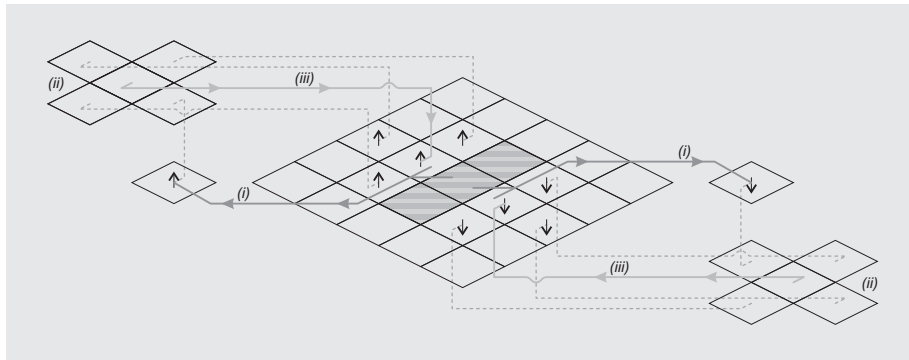


Figure 3.7: Traitement des ambiguïtés de la grille. Les valeurs ambiguës de la grille sont allouées en dehors de la grille ainsi qu'une description du voisinage nécessaire au calcul. Le calcul peut alors être effectué séquentiellement : toutes les valeurs ambiguës sont générées selon leur condition aux limites associée (i) puis les calculs sont réalisés dans chaque voisinage isolé (ii) et les valeurs calculées sont placées dans la grille (iii). Illustration sur une coupe de la grille pour la composante de vitesse perpendiculaire en considérant des conditions de glissement sur l'obstacle.

aux limites est statique, une possibilité d'optimisation supplémentaire existe : ces points sont gérés, pour leur calcul, hors de la grille. Pour chaque point de discrétisation possédant un voisinage ambigu, les cellules porteuses de ces valeurs ambiguës sont allouées hors de la grille, et une structure est créée qui décrit le voisinage de calcul et la destination du résultat. En outre, est mémorisé le moyen de remplir ces voisins isolés à partir de la grille, en tenant compte des conditions aux limites. Cette approche permet de déporter toute la phase d'analyse du voisinage dans une phase de précalcul préalable à la simulation. Lors d'un pas de temps de la simulation, toutes les valeurs multi-définies peuvent ainsi être remplies différemment dans chaque voisinage dans lequel elles apparaissent, et les calculs être ensuite réalisés avec une grande régularité et des accès-mémoires avec une seule indirection, sans aucun test. Cette démarche est illustrée dans la figure 3.7. Il est à noter que ce traitement n'est effectué que dans le cas de valeurs ambiguës. Il n'est pas nécessaire de l'appliquer à tous les points dont la valeur est dynamiquement calculée : ceux de valeurs non-ambiguës sont directement gérés dans la grille.

Cette approche par discrimination préalable est efficace, car il est raisonnable de supposer que le nombre de traitements par indirection reste faible devant le nombre de traitements réguliers dans la grille principale. À titre d'exemple, la figure 3.8 présente la répartition du traitement des différentes cellules pour quelques exemples d'environnements. Ainsi, et même si cela entraîne des limitations, il apparaît logique de chercher à optimiser le calcul de loin le plus communément effectué. Cette technique est également rendue possible en raison de la discrétisation très simple du Laplacien, et particulièrement du non-couplage des trois composantes de la vitesse. Elle se révélerait bien sûr beaucoup plus complexe, voire inefficace, s'il fallait traiter une discrétisation directe du terme advectif qui présente un support de discrétisation plus large et des composantes couplées, comme le font Foster et Metaxas [45].

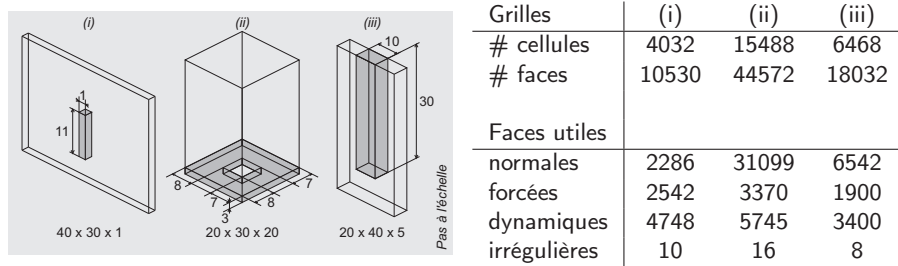


Figure 3.8: Statistiques concernant la distribution des faces pour trois exemples de grilles. Sont mentionnées les dimensions de la partie utile, figurée, de la grille. Pour plus de clarté, les conditions aux limites de bordure sont masquées. L'environnement (i) émule un environnement bidimensionnel en considérant une grille utile munie de conditions de glissement, d'une seule cellule d'épaisseur. L'environnement (ii) présente une boîte avec un plancher troué et l'environnement (iii) correspond à un U — Les faces répertoriées comme « normales » correspondent à un calcul régulier dans la grille. Les faces « forcées » et « dynamiques » correspondent à des faces dont les valeurs sont imposées par le simulateur, pour prendre en compte les conditions aux limites, respectivement de manière statique et de manière dynamique par rapport à une autre face. Les faces « irrégulières » correspondent aux faces nécessitant une valeur ambiguë dans leur calculs. Leur nombre est très restreint puisque pour la plupart, les faces dynamiques sont non-ambiguës et peuvent être gérées dans la grille. La différence entre la somme des catégories et la taille de la grille provient des éléments non utilisés du bord de la grille.

Bords du domaine Par ailleurs, et de manière classique, afin que les bords de la grille ne posent pas de problème et que les supports de discrétisation n'y requièrent aucune modification, le domaine de simulation se voit agrandi et bordé d'une rangée d'obstacles munis de conditions aux limites adéquates sur tous ses bords. La simulation se déroule alors uniquement à l'intérieur de cette grille élargie sans plus avoir à se soucier des bords de la grille.

Gestion de la surface libre

La localisation du liquide dans la grille est réalisée en s'appuyant sur la présence ou non de marqueurs dans les cellules. Les cellules pleines de liquide jouxtant au moins une cellule vide, sont considérées comme représentant la surface du liquide. Ces cellules possèdent deux types de faces : celles jouxtant une autre cellule pleine, qui correspond à une face interne au liquide, et celles formant la frontière du liquide, mitoyennes des cellules vides. Ces faces, frontières du domaine fluide avec le vide environnant, sont les lieux où les conditions aux limites liées à la surface libre sont imposées au fluide.

Leur vitesse est calculée afin que la contrainte d'incompressibilité discrétisée soit satisfaite localement pour chacune des cellules de surface. Cette condition n'est toutefois pas suffisante pour déterminer d'une manière unique les vitesses, si une cellule comporte plusieurs faces-frontières, c'est-à-dire si elle est mitoyenne de plusieurs cellules vides.

Pour fixer les conditions aux limites d'une cellule de surface donnée, une

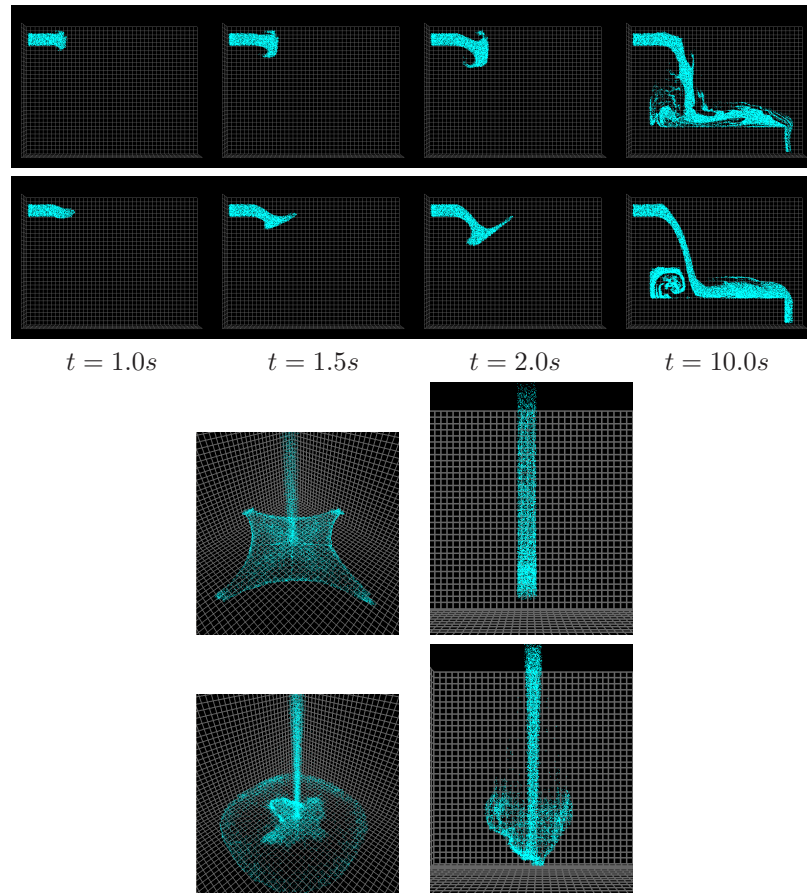


Figure 3.9: Comparaison entre le traitement isotrope et le traitement par axe de la vitesse de la surface. La procédure isotrope correspond à la ligne supérieure, celle par axe à la ligne inférieure. La procédure isotrope provoque une dispersion du liquide plus importante que la procédure par axe, ce qui résulte en un fluide moins bien délimité. Le premier groupe d'images appartient à une animation bidimensionnelle, le deuxième, à une animation tridimensionnelle.

solution est de calculer le flux de liquide entrant ou sortant de la cellule à partir des vitesses des faces internes, et de le redistribuer également entre les faces-frontières pour l'annuler globalement. Néanmoins, cette solution présente l'inconvénient de disperser le liquide dans toutes les directions, comme illustré dans la figure 3.9. Une autre procédure moins isotrope est donc préférée, dans laquelle les axes sont traités indépendamment, et seules les faces-frontières opposées aux faces internes se voient modifiées. Pour une telle face-frontière, la vitesse imposée correspond alors à la vitesse de sa face interne opposée, additionnée à une compensation de la divergence globale de la cellule. Cette solution correspond à celle proposée par Foster et Metaxas [45] complétant celle de Harlow et Welch [64] dans le cas bidimensionnel.

Limitations Implémentée sous la forme décrite, la procédure calculant les vitesses des faces-frontières du fluide ne garantit pas une incompressibilité parfaite dans tous les cas, puisqu'il existe quelques configurations où aucune face frontière à déterminer ne s'oppose à une face interne au liquide, et où aucun traitement n'est donc appliqué. Le respect strict de la contrainte d'incompressibilité serait néanmoins possible en réintroduisant le calcul isotrope dans ces cas.

En outre, le mode de traitement des conditions aux limites choisi, destiné à produire les performances de simulation maximales, se révèle peu adapté pour une gestion stricte des conditions aux limites mobiles découlant de la surface libre. En particulier, il n'est pas possible, au cours de la simulation, de déterminer de manière simple et efficace les conditions aux limites associées à une face de la grille. Ceci interdit par conséquent de tenir parfaitement compte des conditions aux limites de Dirichlet liées aux obstacles, dans les calculs relatifs à la surface libre. Pour la même raison, il n'est pas possible de supprimer complètement la friction parasite de l'environnement sur le liquide lors du calcul du Laplacien, comme illustré dans la figure 3.10, le support fixe de celui-ci employant des valeurs externes au fluide. Néanmoins, et comme les résultats présentés ultérieurement le montreront, ces approximations, certes préjudiciables lors d'une évaluation quantitative, ne compromettent pas significativement la qualité visuelle globale des résultats et peuvent donc se justifier dans le cadre de l'infographie.

Trajectoires : marqueurs et caractéristiques

Le déplacement des marqueurs matérialisant le liquide, tout comme la méthode des caractéristiques, repose sur la résolution de l'équation différentielle ordinaire décrivant la trajectoire d'une particule advectée par le champ de vitesse du fluide. Cette procédure nécessite l'emploi d'un algorithme d'intégration, ainsi que l'interpolation de la vitesse connue sur la grille. Marqueurs et caractéristiques sont toutefois traités selon deux variantes différentes.

Marqueurs Le rôle des marqueurs est de déterminer la présence, ou non, du fluide dans chaque cellule de la grille. *In fine* cette information de présence n'est définie qu'avec la résolution de la grille de calcul, bien inférieure à celle du nuage de marqueurs. Par conséquent, il est raisonnable de postuler que les déplacements indépendants des marqueurs ne nécessitent pas une précision importante,

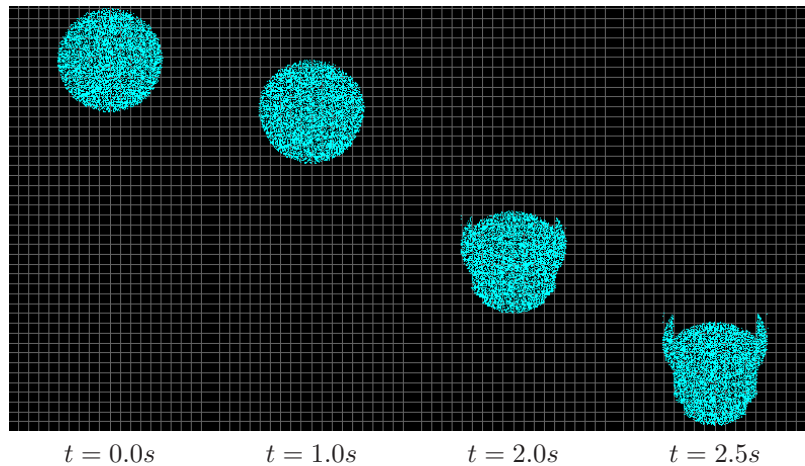


Figure 3.10: Friction parasite de l'air sur le liquide.

puisque c'est leur comportement collectif, plutôt qu'individuel, qui importe vis-à-vis du simulateur. En outre, le nombre des marqueurs étant généralement important, il est préférable de choisir une procédure de mise à jour du liquide, relativement simple et rapide. À cet effet, une intégration d'Euler avant à un seul pas est employée.

Une forme d'interpolation de la vitesse, simple et rapide, sinon précise, est également utilisée. En effet, il n'est pas fait usage de l'interpolation linéaire naturelle définie pour chaque composante de la vitesse par sa grille propre, mais d'une interpolation discontinue, définie sur la base des cellules. Décrite dès l'introduction des grilles décalées par Welch *et al.* [177], une interpolation linéaire aveugle, ne tenant aucun compte particulier des obstacles, conduit à des pertes de liquides en déplaçant des marqueurs à l'intérieur des obstacles, et ce, même si les conditions aux limites d'imperméabilité sont parfaitement respectées, comme dans la situation illustrée par la figure 3.11. Ce problème est lié aux décalages présents entre les grilles discrétisant les différentes composantes de la vitesse et la définition des obstacles.

Plutôt que de recourir à une gestion complexe de l'interpolation, analysant le voisinage dans la grille, l'interpolation est simplifiée et réalisée sur la base des cellules. Sur une cellule, chacune des composantes de la vitesse est linéairement interpolée entre les deux faces correspondantes. Ceci induit une interpolation linéaire continue entre les cellules le long de la composante considérée, et une interpolation constante selon les deux autres axes, discontinue à l'interface des cellules. Ce mode d'interpolation choisi supprime totalement les problèmes liés à l'interpolation puisque son support coïncide exactement avec les cellules constituant la base de définition des obstacles. Les résultats obtenus présentent une qualité visuelle suffisamment correcte pour être exploitée. Si cette approximation peut paraître hasardeuse dans un cas général, il faut rappeler que les seules variations de trajectoire des marqueurs affectant les résultats sont celles des particules à proximité de la surface, pour lesquelles des imprécisions sont déjà commises dans les calculs de vitesse. Par conséquent, ce mode d'interpolation s'avère adapté à la situation.

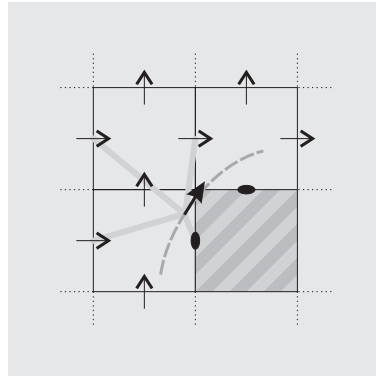


Figure 3.11: Problème de l'interpolation linéaire à proximité d'un obstacle. Malgré le respect des contraintes d'imperméabilité, l'interpolation linéaire engendre des trajectoires pénétrant les obstacles. Pour le point repéré sur la trajectoire sont figurés les points de discrétisation interpolés pour une des deux composantes de la vitesse.

Étant donné le très grand nombre de marqueurs à déplacer tout au long d'une simulation, et la compacité du code mis en jeu, il est possible et intéressant d'optimiser la procédure de déplacement des marqueurs. Le code mis en jeu pour chaque marqueur à déplacer comporte deux parties distinctes. La première est le calcul de la cellule contenant le marqueur ainsi que le calcul de sa position relative à l'intérieur de la cellule nécessaire à l'interpolation. La seconde est constituée de l'interpolation elle-même. Or, en maintenant directement la position des marqueurs sous une forme relative, c'est-à-dire sous la forme de l'indice de la cellule occupée et de la position relative dans la cellule, il est possible de supprimer intégralement la phase de conversion. Par rapport à une version manipulant les positions absolues des particules, un temps d'exécution approximativement divisé par quatre est alors obtenu, au coût d'un tiers d'espace-mémoire supplémentaire.

Caractéristiques Pour le traitement des caractéristiques, deux algorithmes d'intégration ont été testés : une intégration d'Euler avant, et une intégration par méthode de Runge-Kutta d'ordre deux. Dans les deux cas, les méthodes ne sont pas adaptatives vis-à-vis du champ de vitesse sous-jacent, et procèdent en subdivisant la trajectoire en un nombre de pas prédéfini. Comme illustré dans la figure 3.12, les différences qualitatives entre ces deux procédures restent minimales étant donné le nombre important — une vingtaine — de pas d'intégration effectués, par rapport aux déplacements subis et à la résolution de la grille. Dans le cas où la caractéristique analysée est amenée à franchir la frontière du domaine de simulation, au niveau d'un obstacle ou à la surface du liquide, on arrête son suivi et la vitesse advectée est celle constatée au point d'arrêt partiel.

Le suivi de la caractéristique nécessite également l'interpolation du champ de vitesse discrétisé, comme pour les marqueurs, mais les problèmes potentiels sont légèrement différents. Si les trajectoires peuvent toujours pénétrer les obstacles en employant une interpolation trilineaire, cette situation ne pose plus nécessairement problème : la trajectoire se trouve écourtée mais il est raisonnable de supposer qu'une valeur proche de la valeur attendue est tout de même advectée.

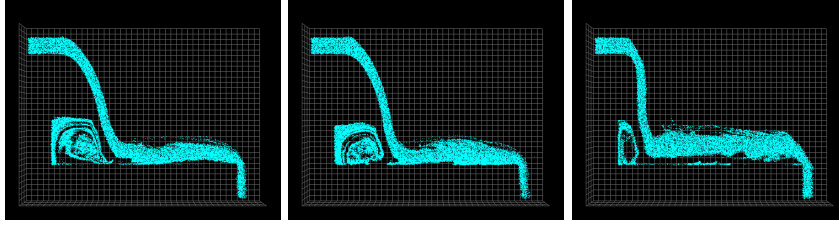


Figure 3.12: Comparaison du traitement des caractéristiques. De gauche à droite : Méthode RK2 avec interpolation hybride, méthode d'Euler avant avec interpolation hybride et méthode RK2 avec interpolation locale aux cellules.

En outre, contrairement au cas des marqueurs, aucun risque de déperdition du fluide n'est présent.

Cependant, à proximité de la surface, le mode d'interpolation trilineaire tend à être pollué par les valeurs de vitesse non définies extérieures au liquide. C'est pourquoi, pour les cellules composant la surface, le mode d'interpolation choisi est le même que celui employé pour les marqueurs, local à chaque cellule, et ainsi non affecté par les valeurs voisines non-pertinentes. Le choix d'une interpolation hybride est donc le fruit d'un compromis. Il est en effet préférable d'employer, quand faire se peut, l'interpolation trilineaire complète plutôt que l'interpolation locale plus robuste, mais moins précise.

La différence entre l'emploi, pour les caractéristiques, de cette interpolation hybride, et l'emploi de la seule interpolation locale, est présentée dans la figure 3.12.

Traitement de l'incompressibilité

Le traitement de l'incompressibilité demande la résolution de l'équation de Poisson suivante pour la pression :

$$\begin{cases} \Delta p = \frac{1}{\delta t} \nabla \cdot u^* & \text{sur } \Omega \\ \frac{\partial p}{\partial n} = 0 & \text{sur } \Gamma_D \\ p = p_{atm} & \text{sur } \Gamma_S \end{cases}$$

Pour cela le système est discrétisé sur la base de la grille décalée. Le système linéaire $Lp = u$ résultant de la discrétisation du Laplacien est résolu par la méthode du gradient conjugué. Or, comme le mentionnent Foster et Fedkiw [44], muni de ces conditions aux limites, le système discrétisé présente une structure creuse particulière, ce qui est attendu puisqu'il s'agit de la discrétisation d'un Laplacien sur une grille tridimensionnelle régulière.

En considérant une cellule C d'indice $c = (i, j, k)$, et l'ensemble des indices de ses cellules adjacentes pleines de liquides $N_F(C)$:

$$\sum_{m \in N_F(C)} (p_m - p_i) = \frac{\delta x}{\delta t} \sum_{n \in F} (u_{i+n}^* - u_{i-n}^*)$$

avec

$$F = \{(1/2, 0, 0), (0, 1/2, 0), (0, 0, 1/2)\}$$

$$N_F(C_c) = \{c + l \mid C_{c+l} \text{ plein et } l \in \{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}\}$$

Le système discrétisé se présente donc sous la forme suivante : une ligne i correspond à la discrétisation du Laplacien d'une cellule C_i particulière de la grille, L_{ii} correspond à l'opposé du nombre de cellules adjacentes remplies de liquide, et un élément $L_{ij} = 1$ si la cellule C_j , adjacente à C_i , est remplie de liquide, et $L_{ij} = 0$ dans le cas contraire. L'élément u_i est, lui, égal à la discrétisation de la divergence de la cellule C_i .

Au vu de cette structure particulière connue à l'avance, le système peut donc employer une construction, un mode de stockage et une résolution optimisés. En particulier, la représentation adoptée de la matrice L ne stocke que les valeurs des éléments diagonaux ainsi que les positions, pour chaque ligne, des éléments non-nuls puisque leur valeur est connue à l'avance.

3.1.7 Algorithme final

L'algorithme de résolution complet est détaillé dans la figure 3.13. Il correspond à la résolution des équations de Navier-Stokes par la méthode résumée dans la figure 3.3, en tenant compte des conditions aux limites. Cette prise en compte est réalisée en analysant leur influence sur la grille préalablement à la simulation, puis en mettant à jour la grille en conséquence durant chaque pas de temps quand cela est nécessaire. Les différentes étapes de mise à jour de la grille n'ont alors plus à se préoccuper des conditions aux limites et opèrent avec une parfaite régularité sur l'intégralité de la grille.

3.1.8 Résultats

On présente maintenant quelques résultats obtenus à l'aide du simulateur décrit précédemment. L'objectif de cette présentation n'est pas d'étudier en détails le comportement du simulateur mais de donner un bref aperçu des capacités, et des limites, de ce mode de simulation.

Les figures 3.14 et 3.15 présentent deux environnements dans lesquels le fluide adopte automatiquement un comportement naturel, grâce au simulateur. Même si les situations restent relativement simples, l'animation est intéressante du point de vue visuel, et ne pourrait être produite, avec ces détails, qu'au prix d'un travail très important par un animateur humain.

S'il est possible de varier l'environnement pour changer du tout au tout le résultat d'une animation, on peut également altérer le comportement du fluide en modifiant sa viscosité. Comme l'illustre la figure 3.16, des animations très différentes sont obtenues par ce biais. Le fluide apparaît ainsi tantôt comme de l'eau, tantôt comme de la glu.

Remarques Bien que les possibilités de simulation soient vastes, et le comportement réaliste du fluide visuellement intéressant, plusieurs critiques peuvent être formulées sur les résultats.

Tout d'abord, les animations manquent singulièrement de vie. D'où l'objectif de faire interagir automatiquement le fluide avec des objets mobiles externes.

Analyse préalable à la simulation

- A1. Repérage des obstacles dans la grille à partir de la description de la scène fournie par l'utilisateur.
- A2. Analyse du voisinage de chaque cellule, duplication des cellules ambiguës et mise en place des liens nécessaires aux calculs ultérieurs.
- A3. Mise en place dans la grille des conditions initiales de vitesse.
- A4. Génération des marqueurs aux positions initiales et détection de la position de la surface.

À chaque pas de simulation

- S1. Calcul des vitesses libres des cellules de surfaces.
- S2. Réinitialisation de la grille non occupée par le liquide.
- S3. Application des conditions aux limites à la grille. Recalcul des valeurs de vitesses dynamiques à proximité des obstacles.
- S4. Traitement de l'advection.
- S5. Application des forces externes.
- S6. Traitement de la viscosité.
- S7. Application des conditions aux limites. Recalcul des valeurs de vitesses dynamiques à proximité des obstacles.
- S8. Réinitialisation de la grille non occupée par le liquide.
- S9. Projection pour tenir compte de l'incompressibilité
- S10. Application des conditions aux limites à la grille. Recalcul des valeurs de vitesses dynamiques à proximité des obstacles.
- S11. Déplacement, création et destruction des particules.
- S12. Détection de la position de la surface dans la grille.

Figure 3.13: Algorithme de simulation MAC implémenté.

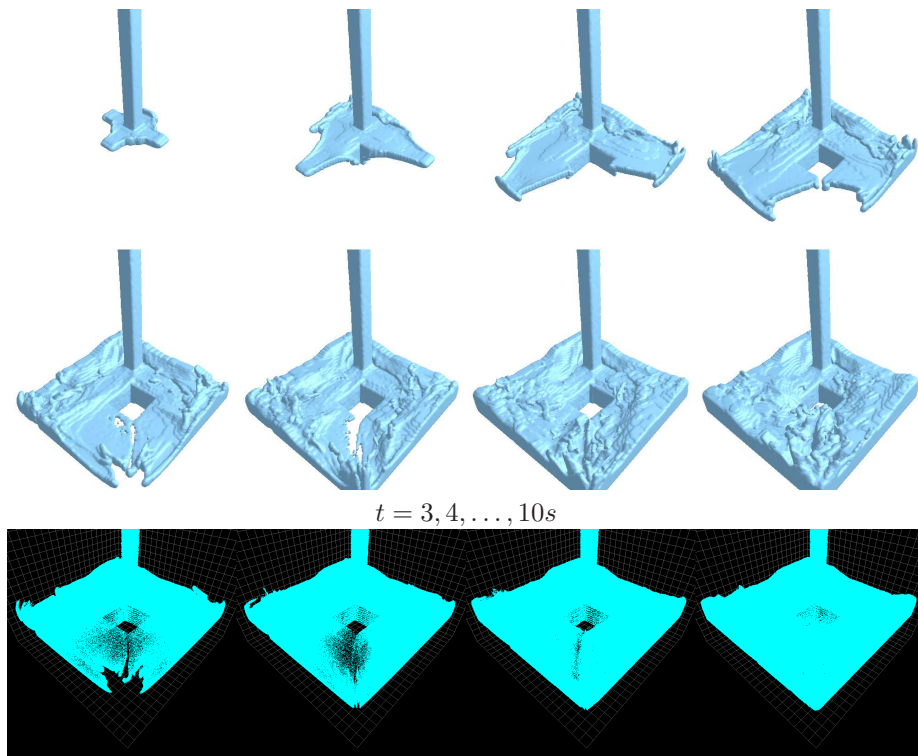


Figure 3.14: Exemple : boîte à fond percé. Une surface est reconstruite à partir des données du simulateur. Le fluide est introduit en continu dans un coin de la boîte et évolue sous l'influence de la gravité. La deuxième partie de l'animation est présentée sous forme reconstruite et sous forme brute.

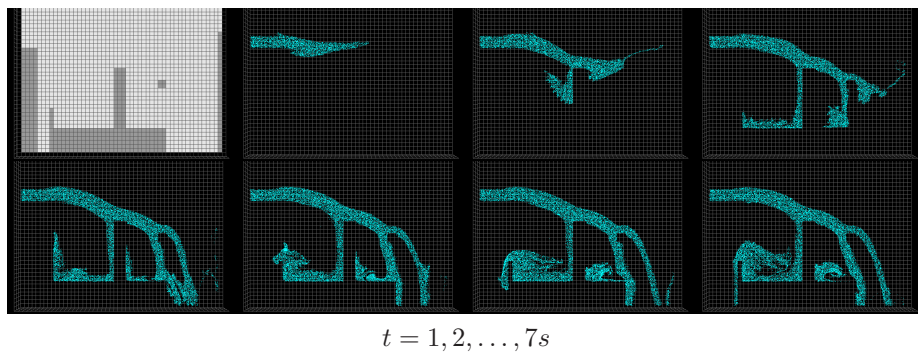


Figure 3.15: Exemple : jet et obstacles. L'environnement de simulation est figuré dans la première image. Le fluide est introduit en continu avec une vitesse imposée sur la gauche du domaine.

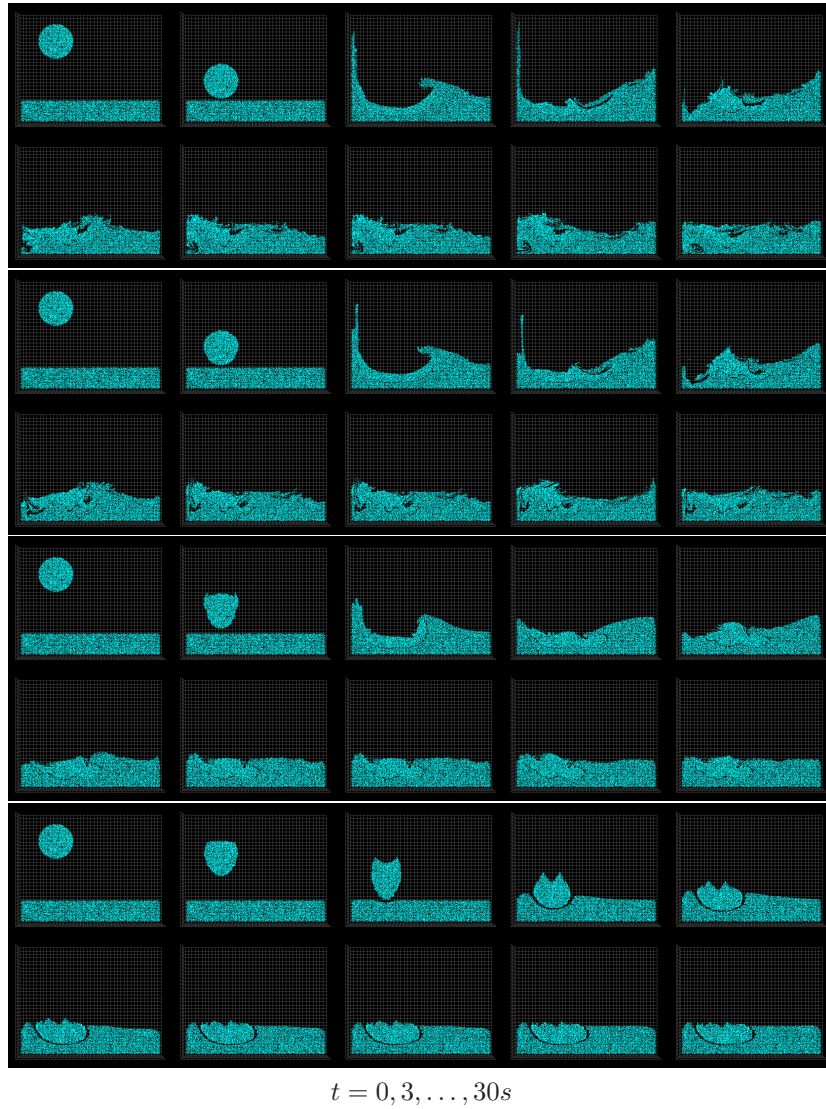


Figure 3.16: Exemple : variation de la viscosité. De haut en bas : $\nu = 0, 0.1, 1, 10$.

Le but visé est alors d'augmenter le dynamisme et l'attrait des situations simulées. Une technique autorisant ce type d'interaction du fluide avec des solides déformables représentés sous forme d'objets masses-ressorts, est présentée dans la section 3.2.

Si cette première remarque relève d'une perception humaine, et ne peut réellement être qualifiée de défaut — on ne peut exiger d'un simulateur de fluides de simuler autre chose que des fluides — d'autres défauts, plus tangibles, sont repérables au sein de ces résultats.

On peut remarquer, par exemple dans la figure 3.16, que l'uniformité de la densité des particules n'est pas nécessairement préservée tout au long de la simulation. Plus les mouvements sont violents, plus est grand le risque de voir se développer des zones pauvres en marqueurs, mais suffisamment peuplées pour que le simulateur y voie un fluide sans bulle. Ces zones, qui prennent naissance à la surface du fluide, sont ensuite relativement préservées en raison de la contrainte d'incompressibilité. Ceci est dû à la manière dont le simulateur perçoit le liquide, perception liée à la résolution de la grille et au critère binaire de présence ou d'absence de marqueurs, et non à la réelle densité de marqueurs.

Par conséquent, le volume du liquide varie au cours du temps, de manière parasite, ce qui est illustré dans la figure 3.17. On constate que l'augmentation artificielle du volume est bien corrélée à la violence des mouvements du liquide, un saut dans le volume se produisant lorsque la boule de fluide fusionne avec le liquide déjà présent au fond du bac. Après ce saut, la variation de volume dépend directement de l'activité de la surface : dans le cas d'une viscosité élevée, la variation devient rapidement nulle, contrairement au cas d'un liquide moins visqueux.

La variation de volume que l'on vient de décrire affecte la simulation elle-même. Elle ne parasite néanmoins pas trop les résultats, et ceci pour deux raisons : premièrement, la variation de densité de marqueurs est relativement faible et localisée dans l'espace, et ne remet donc pas en cause la représentation du fluide par les marqueurs. Deuxièmement, s'il est difficile pour un observateur humain de percevoir les variations de volume du liquide en ayant pour seule référence l'animation elle-même, cela devient quasiment impossible dans le cas d'une scène où le liquide est introduit délibérément, comme dans la figure 3.14.

Cependant, il existe un autre problème, également lié au manque d'homogénéité des marqueurs, qui se trouve illustré dans la même figure. Il survient lorsqu'une surface doit être explicitement reconstruite à partir du nuage de points pour permettre la visualisation du fluide. On note que l'on emploie ici un algorithme de *Marching Cubes* [93] sur une grille échantillonnant une version lissée de la densité de marqueurs. Si l'on examine cette figure en détails, on s'aperçoit à un instant de l'animation que le fluide représenté par sa surface est de manière peu explicable absent du coin de la boîte le plus proche de l'observateur.

Cette incohérence locale entre la présence des marqueurs et la surface finale découle du manque d'homogénéité de la densité de marqueurs. Il est dès lors impossible de trouver des paramètres d'extraction de surface, satisfaisants sur le plan visuel tout en assurant un bon respect de la cohérence temporelle, dans l'ensemble du domaine. Il serait évidemment possible de rendre l'algorithme d'extraction de surface plus tolérant pour capter cette portion de fluide, mais au prix d'un empâtement des surfaces générées, pour un résultat global de qualité finalement inférieure. Une augmentation significative du nombre de particules, qui apparaît comme une solution immédiate, ne constitue pas une option viable,

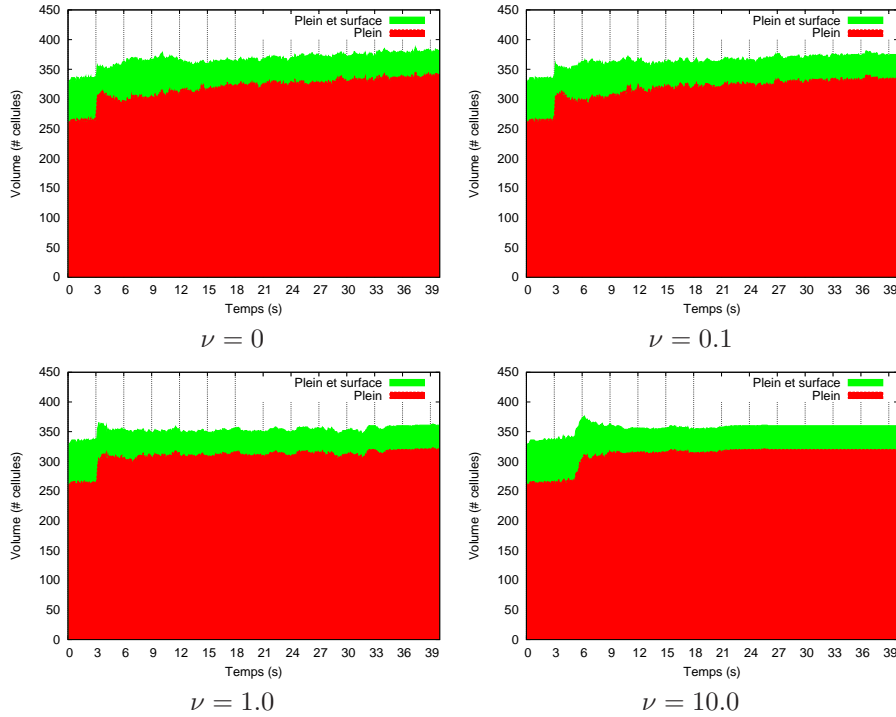


Figure 3.17: Conservation du volume de liquide pour les différentes situations de la figure 3.16. Les repères dans le temps indiquent les instants des différents clics.

puisqu'approximativement trois millions de particules sont déjà utilisées pour cette simulation relativement simple.

Les avancées récentes de représentation du liquide, employant les méthodes de type *levelset* et leurs extensions, décrites dans la section 2.3.2, apportent une meilleure réponse à ce problème. Malheureusement, on ne dispose pas de ces techniques dans le simulateur qui a été implémenté.

À titre d'information uniquement, les temps d'exécution pour deux simulations différentes sont présentés dans la figure 3.18. Ces temps sont à considérer avec réserve, eu égard au nombre de paramètres différents influant sur la simulation : les deux cas présentés ne sauraient être généralisés et ne restent représentatifs que des simulations desquelles ils sont issus. Les temps sont obtenus pour un processeur de type Pentium 4 2.6 GHz. La première situation présentée correspond à la chute d'un disque de fluide tombant dans le bac partiellement rempli de la figure 3.16 — une petite simulation bidimensionnelle — et la seconde au remplissage progressif d'une boîte percée par un liquide, dans la figure 3.14. On notera que dans ce deuxième cas, c'est le déplacement des marqueurs qui monopolise la quasi-totalité du temps de calcul, ceux-ci étant très nombreux, de l'ordre de cinq millions après douze secondes de simulation. On précise également que les temps rapportés n'incluent que la simulation, hors visualisation ou export de données.

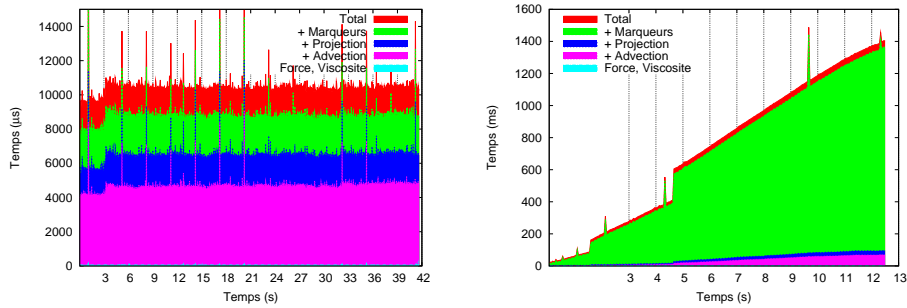


Figure 3.18: Temps d'exécution du simulateur MAC. Gauche : temps correspondant au disque tombant dans le bac avec $\nu = 1.0$: 1200 cellules et 28000 particules. Droite : Boite percée : 12000 cellules, 0 à 5M particules. Les temps reportés pour les deux simulations correspondent à un pas de temps de $1/24$ de seconde. Les repères dans le temps marquent les clichés qui composent les figures illustrant les simulations.

3.2 Interaction avec des solides déformables

Cette section présente l'extension proposée aux simulateurs MAC pour permettre une interaction automatique du liquide avec des objets déformables.

Contexte Une solution au problème de l'interaction est de traiter les problèmes du fluide et du solide séparément, en employant un couplage par les conditions aux limites. Les solides fixent alors des conditions aux limites sur la vitesse du fluide, et simultanément le fluide impose des forces aux solides. Les objets se retrouvent donc essentiellement voxelisés dans le maillage de simulation et imposent des conditions à l'échelle de celui-ci. Cependant, cette approche ne peut être mise en œuvre de manière vraiment satisfaisante dans le cadre d'un simulateur MAC à grille régulière fixe, puisque les conditions aux limites du fluide sont dans ce cas contraintes à coïncider avec les faces de la grille : elles ont alors des difficultés à matérialiser fidèlement l'interface entre fluide et solides.

Pour plus de détails sur les différents moyens qui ont été proposés en infographie pour réaliser cette interaction automatique entre un fluide simulé de manière Eulérienne et des objets externes, le lecteur est invité à se reporter à la section 2.3.4.

Objectifs de l'interaction développée L'approche que nous proposons cherche à éviter toute forme de voxelisation des objets visant à imposer des conditions aux limites au fluide, et s'appuie pour cela sur une interaction directe entre le liquide et les solides. Par ailleurs, on désire introduire des objets déformables, visuellement plus intéressants que des objets strictement rigides.

La représentation adoptée pour ces objets déformables, puis l'interaction proposée proprement dite, sont décrites dans les sections suivantes. Si ces deux points sont présentés successivement, il va sans dire qu'ils sont étroitement liés.

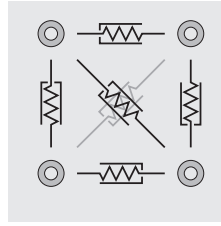


Figure 3.19: Carré modélisé par des masses et des ressorts. Quatre masses sont disposées aux quatre coins du carré, et un maillage complet d'interactions est employé.

3.2.1 Modélisation masses-ressorts des solides

La modélisation choisie pour les objets déformables est celle des objets masses-ressorts au sens large, c'est-à-dire de solides représentés par un système de particules muni d'interactions dynamiques prédéterminées. Cette forme d'interaction assez générale est celle employée, entre autres, par Terzopoulos et Fleischer [161] pour modéliser des objets déformables.

Description du modèle Un solide déformable continu est représenté par un ensemble de masses ponctuelles $\{S_i\}_i$, de masses m_i et de positions x_i , évoluant chacune indépendamment sous l'effet des forces f_i qui leur sont appliquées, en accord avec la deuxième loi de Newton :

$$m_i \cdot \frac{d^2 x_i}{dt^2} = f_i$$

La force f_i regroupe toutes les forces s'appliquant au point P_i , ce qui inclut les sollicitations externes à l'objet, comme la gravité ou la réponse aux collisions, mais également les forces internes à l'objet modélisé.

Dans la mesure où l'on souhaite modéliser des objets déformables, mais possédant tout de même une forme propre, on emploie pour modèle de forces internes un modèle visco-élastique avec des interactions statiques, c'est-à-dire que les paires de masses en interaction au sein du solide sont fixées à l'avance et ne varient pas. Les paramètres de ces interactions modélisant la cohésion des différentes masses sont également déterminés avant le début de la simulation.

Comme son nom l'indique, le modèle des forces de cohésion adopté est la superposition d'un modèle élastique linéaire et d'un modèle visqueux. Pour une paire de masses $(S_i, S_j)_{i \neq j}$, l'action que S_j applique à S_i présente l'intensité suivante :

$$|c_{S_i, S_j}| = k \cdot (|x_i - x_j| - d) + z \cdot \frac{d}{dt} |x_i - x_j|$$

avec pour paramètres :

- $k \geq 0$ la raideur de la partie élastique
- $d \geq 0$ la distance au repos entre les deux points P_i et P_j
- $z \geq 0$ le coefficient de viscosité.

Intuitivement, la partie élastique de l'interaction tend à restaurer la distance au repos entre les deux particules, et la partie visqueuse cherche à annuler la vitesse relative des deux particules. La force est alignée avec la direction portée

par la paire de masses :

$$c_{S_i, S_j} = |c_{S_i, S_j}| \cdot \frac{x_j - x_i}{|x_j - x_i|}$$

Par définition, cette force est symétrique et donc cohérente avec la troisième loi de Newton, qui spécifie que les actions et réactions sont d'intensité et de directions égales mais de sens opposés.

La topologie du maillage des interactions ainsi que les paramètres des interactions déterminent conjointement les propriétés physiques de l'objet. Un exemple d'objet simple ainsi modélisé est présenté dans la figure 3.19.

Justification du choix de ce modèle La raison principale qui pousse à adopter ce mode de simulation pour les objets solides, est que les deux milieux, liquide et solide, présentent alors une forme Lagrangienne similaire, sous les formes respectives des marqueurs et des masses. Comme on le verra, cette similarité est fortement exploitée dans la forme proposée d'interaction entre fluide et solides. Une telle modélisation masses-ressorts est également choisie en raison de ses caractéristiques avantageuses : elle est extrêmement simple à mettre en œuvre, elle est souple, et elle est raisonnablement performante, si les objets ne sont pas trop densément maillés.

Ces avantages s'accompagnent néanmoins d'un inconvénient : cette modélisation, bien qu'inspirée de la physique, reste empirique, et ne correspond pas à une discrétisation rigoureuse des équations de la mécanique des milieux continus pour un solide élastique. Étant donné que la visée poursuivie est graphique, et ne cherche pas à prévoir des valeurs physiques réalistes, ce défaut n'est ici pas rédhibitoire.

Soulignons que lorsque l'on considère les objets uniquement comme des ensembles de masses, la simulation physique en ignore toute notion de surface. Ce point n'est pas problématique dans l'objectif de l'interaction avec le liquide, ce dernier ne présentant pas non plus de surface clairement définie.

On peut également justifier le choix de ce mode de simulation pour les solides par la qualité de représentation qu'il offre, comparable à celle offerte par les marqueurs du simulateur de fluide MAC. Il n'est pas utile, dans ce contexte, de recourir à un mode de simulation plus précis qui serait alors sous-exploité, et gaspillerait sans doute des ressources sans pour autant offrir une simulation de meilleure qualité.

3.2.2 Interaction

Principe Pour coupler directement le modèle du fluide et celui des solides, on propose de calculer une force d'interaction directe entre les deux milieux. Cette force est ensuite réintroduite symétriquement dans les deux modèles. Une telle approche garantit une réelle interaction réciproque, ainsi qu'une cohérence de réaction, entre les solides et le fluide.

Pour calculer cette force d'interaction, on tire parti de la représentation Lagrangienne des deux milieux : le liquide est représenté par ses marqueurs, le solide par ses masses. En considérant ces deux représentations discrètes, l'interaction complète entre les deux systèmes physiques peut alors être calculée comme une somme d'interactions point-à-point pour toutes les paires mixtes

composées d'un marqueur du liquide et d'une masse d'un solide. Les masses et les marqueurs agissent donc comme les points de discrétisation mobiles de cette interaction.

On notera que, dans ce cas, le solide apparaît pour le fluide comme une perturbation dynamique des forces externes, et *vice versa*. Le solide évolue alors dans le domaine du fluide en s'y superposant, sans le restreindre explicitement d'aucune manière.

Force d'interaction Il est nécessaire de choisir une forme à cette force d'interaction. On rappelle qu'elle a pour rôle de modéliser du mieux possible l'intégralité de l'interaction entre solides et fluide. Elle doit donc avoir une action tentant d'empêcher d'une part la pénétration des objets par les marqueurs, et de favoriser d'autre part condition naturelle d'égalité des vitesses des deux milieux aux interfaces. Par ailleurs, il faut remarquer que ces interactions doivent être nécessairement limitées dans l'espace, au vu de leur caractère purement local. Ces considérations amènent alors à choisir une forme particulière d'interaction visco-élastique, limitée dans l'espace, et dont l'action élastique ne peut être que répulsive.

Étant donné $M = \{M_j\}_j$ l'ensemble des marqueurs du fluide, et $S = \{S_i\}_i$ l'ensemble des masses des différents solides présents dans la simulation, pour une paire mixte $(E_i, E_j)_{(E_i \in M \wedge E_j \in S) \vee (E_i \in S \wedge E_j \in M)}$, l'intensité de la force est définie comme :

$$|f_{E_i, E_j}| = R_{E_i, E_j} + V_{E_i, E_j}$$

avec le terme de répulsion élastique linéaire :

$$R_{E_i, E_j} = \begin{cases} r \cdot \left[\frac{|x_j - x_i|}{D_r} - 1 \right] & \text{si } |x_j - x_i| \leq D_r \\ 0 & \text{sinon} \end{cases}$$

et le terme de viscosité :

$$V_{E_i, E_j} = \begin{cases} v \cdot \frac{d}{dt} |x_j - x_i| & \text{si } |x_j - x_i| \leq D_v \\ 0 & \text{sinon} \end{cases}$$

en fonction des paramètres suivants :

- $D_r \geq 0$ la distance de répulsion maximale
- $D_v \geq 0$ la distance d'interaction visqueuse maximale
- $r \geq 0$ l'intensité de répulsion maximale
- $v \geq 0$ le coefficient de viscosité.

Cette interaction correspond à une interaction visco-élastique classique limitée dans l'espace, pour laquelle le rayon d'action élastique est égal à la distance au repos de la partie élastique. Afin d'offrir un contrôle plus flexible de la force, on autorise deux rayons différents pour limiter les deux termes dans l'espace. Logiquement, la force d'interaction est portée par la droite passant par les points E_i et E_j :

$$f_{E_i, E_j} = |f_{E_i, E_j}| \cdot \frac{x_j - x_i}{|x_j - x_i|}$$

Le choix des rayons d'interaction D_r et D_v est fortement influencé par la façon dont les solides sont maillés, comme le montre la figure 3.20 : ces rayons

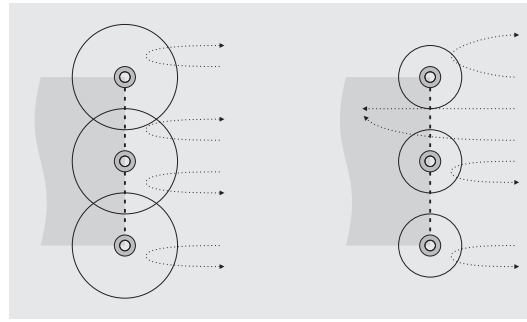


Figure 3.20: Répulsion des marqueurs par l'interaction. À gauche, l'interaction couvre suffisamment la surface de l'objet pour empêcher les marqueurs de pénétrer dans le solide. À droite, les interactions sont trop limitées et pas assez raides pour repousser efficacement les marqueurs hors du solide.

définissent en effet implicitement la zone d'interaction des objets avec le fluide, et par conséquent la pseudo-surface que présentent les solides au simulateur de fluide. En particulier, dans un objectif d'imperméabilité des solides, ces rayons doivent être choisis de taille suffisante pour éviter la formation de trous dans le volume d'interaction des solides, tout en approximant correctement leur surface. En outre, l'intensité de la répulsion doit être assez forte pour prévenir efficacement l'intrusion de marqueurs dans le corps de l'objet et rendre ainsi les objets imperméables aux marqueurs. Toutefois, il faut souligner qu'il n'est pas possible de garantir une parfaite imperméabilité des objets puisque celle-ci n'est fondée que sur une modélisation par intention : aucune raideur de l'interaction élastique, si élevée soit-elle, ne peut prévenir la pénétration d'une particule possédant une vitesse suffisante. On notera également que, du point de vue de l'interaction, on ne peut attribuer de réelle surface aux solides : les marqueurs pénètrent plus ou moins profondément la zone d'interaction, selon la configuration locale.

Pertinence de l'interaction Avec cette représentation, et comme illustré dans la figure 3.21, la représentation Lagrangienne du fluide tient lieu d'interface entre le fluide Eulérien et le calcul de la force d'interaction. Le choix des marqueurs comme représentants Lagrangiens du liquide, et donc comme support de l'interaction, est tout à fait naturel : ils sont précisément introduits dans le schéma MAC comme une forme de discrétisation de la présence du fluide dans l'environnement. Par définition, ils constituent donc l'information disponible la plus précise que l'on puisse trouver sur la localisation du liquide. En s'appuyant sur eux, on exploite du mieux possible cette information dans le calcul de l'interaction.

Cette précision de localisation est un point important dans la mesure où la position de la surface du liquide varie et influe directement sur l'interaction. En effet, comme on s'intéresse au cas d'un fluide à surface libre et à des objets de taille non négligeable par rapport le fluide, les solides peuvent se trouver partiellement immergés. Il serait par trop simplificateur de supposer que le voisinage du solide est composé intégralement, ou non, de fluide. En adoptant cette interaction entre masses et marqueurs, on est capable de localiser l'interaction à

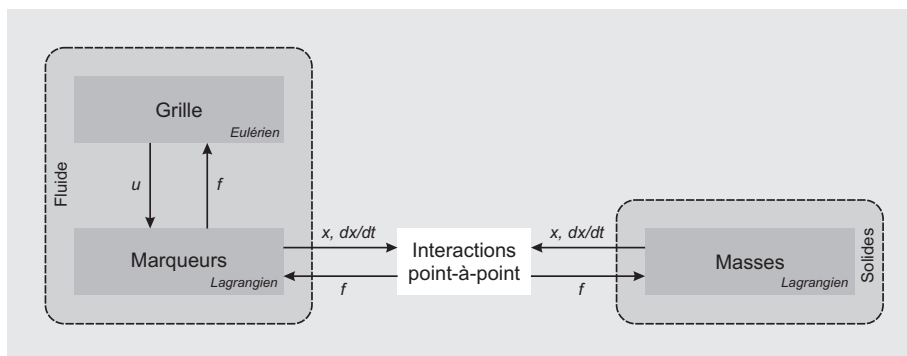


Figure 3.21: Interaction entre liquide et solides déformables.

l'échelle de ces points, donc assez précisément dans l'espace.

On précise d'emblée que cette modélisation de l'interaction est empirique, et que la forme du couplage est construite à dessein : aucune réalité macroscopique ne vient étayer cette construction. Pour autant, cette forme d'interaction n'est pas sans fondement, et peut trouver une justification à une échelle microscopique. En effet, à cette échelle, des forces d'interaction existent entre les deux milieux : on en choisit une modélisation, qui peut être discutée, et l'on extrapole ensuite l'échelle. En cela, la construction que l'on propose procède selon une démarche similaire à celle employée par Tonnesen [166] dans laquelle les interactions de Lennard-Jones, d'échelle moléculaire, influent sur des particules de matière de taille bien plus importante que des molécules.

Ce choix de force concorde également avec une vision continue du domaine de simulation, où, à grande échelle, l'interface peut être vue comme une zone de transition amalgamant les deux milieux. Comme les forces visco-élastiques sont capables de simuler aussi bien des fluides [61] que des solides déformables [161], elles sont de bonnes candidates pour modéliser le comportement de cette zone de transition hybride.

Prise en compte de la force d'interaction La quantité que l'on a définie précédemment, et à laquelle on a abusivement donné le nom de force, correspond en fait à une densité de force entre une masse ponctuelle et l'élément de volume de fluide représenté par son marqueur. Elle est donc définie pour une masse et par unité de volume du fluide. Il reste à reconstruire effectivement les forces subies par les masses et le fluide, et à les réintroduire dans les simulateurs respectifs.

À cet effet, on part de l'hypothèse que les marqueurs du fluide constituent un échantillonnage uniforme de l'espace occupé par le fluide, c'est-à-dire que la densité spatiale de marqueurs est uniforme. Cette hypothèse semble raisonnable en raison de l'incompressibilité du fluide, pour peu que les marqueurs soient initialement distribués de manière homogène. On peut d'ailleurs constater dans les différentes animations précédemment présentées, par exemple celles de la figure 3.16, que cette propriété reste généralement vérifiée sauf dans le cas de mouvements violents. Toutefois, même dans ces cas, les perturbations restent relativement faibles.

On suppose donc connue η la densité spatiale de marqueurs dans le liquide, supposée homogène. Si sa valeur n'est pas effectivement connue, elle peut être calculée en évaluant le ratio entre nombre de marqueurs présents dans la simulation et volume de la grille de simulation rempli de liquide.

La force d'interaction subie par une masse S_i d'un solide, correspond donc à :

$$F_{S_i} = \int_{V_{int}} f_{S_i,x} dx$$

où V_{int} représente le volume d'interaction. En s'appuyant sur une intégration de Monte-Carlo et l'hypothèse d'uniformité de la distribution des marqueurs dans le liquide, qui autorise à considérer les marqueurs comme des échantillons équiprobables, il est possible d'exprimer la force sur la base des interactions point-à-point avec les marqueurs présents dans V_{int} :

$$F_{S_i} \approx \frac{|V_{int}|}{N} \sum_{j=1}^N f_{S_i,M_j}$$

Or, puisque la densité de marqueurs est supposée constante, on a également l'approximation suivante : $N \approx \eta \cdot |V_{int}|$, qui permet d'écrire :

$$\begin{aligned} F_{S_i} &\approx \frac{|V_{int}|}{\eta \cdot |V_{int}|} \sum_{j=1}^N f_{S_i,M_j} \\ &\approx \frac{1}{\eta} \sum_{j=1}^N f_{S_i,M_j} \end{aligned}$$

Cette dernière quantité F_{S_i} représente la force que subit la masse S_i de la part de l'intégralité du fluide avec laquelle elle est en interaction. Sa prise en compte est alors triviale, puisqu'elle peut être directement intégrée à l'évolution de S_i , comme n'importe quelle autre force.

Pour traiter le cas du liquide, on se focalise d'abord sur un point de discrétisation de la grille de simulation, et sur une masse unique. On s'intéresse donc à l'interaction F_{D,S_i} de tout le fluide qui est contenu dans le volume de contrôle $CV(D)$ associé à un point de discrétisation D , avec une unique masse S_i . La valeur discrétisée par D correspond alors à la densité de force moyenne expérimentée par le fluide sur $CV(D)$:

$$F_{D,S_i} = \frac{1}{|CV(D)|} \int_{CV(D)} f_{x,S_i} dx$$

Comme précédemment, en utilisant une intégration de Monte-Carlo dont le support correspond aux marqueurs présents dans $CV(D)$, ainsi que l'hypothèse d'uniformité de la distribution de ces marqueurs, cette expression peut se réécrire sous la forme :

$$F_{D,S_i} \approx \frac{1}{|CV(D)|} \frac{|CV(D)|}{N} \sum_{j=1}^N f_{M_j,S_i}$$

D'un point de vue pratique, la sommation des forces ne nécessite cependant pas l'analyse de tous les marqueurs : l'interaction étant limitée à un rayon d'action

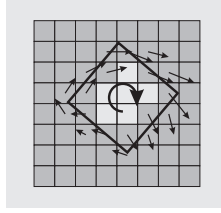


Figure 3.22: Interaction représentée par un champ de force non-uniforme. La force d'interaction n'est pas uniforme dans l'espace, et dépend localement des mouvements des solides et du liquide.

restreint, seuls les marqueurs suffisamment proches, susceptibles d'induire une interaction non-nulle, nécessitent d'être examinés.

On introduit ensuite η en remarquant que $N \approx \eta \cdot |CV(D)|$:

$$F_{D,S_i} \approx \frac{1}{\eta \cdot |CV(D)|} \sum_{j=1}^N f_{M_j,S_i}$$

Comme $|CV(D)|$ est constant et correspond au volume d'une cellule de la grille, la valeur de la densité de force totale en D , induite par la masse S_i , correspond à :

$$F_{D,S_i} \approx \frac{1}{\eta \cdot \delta x^3} \sum_{j=1}^N f_{M_j,S_i}$$

Finalement, l'interaction totale discrétisée en D se compose de toutes les interactions élémentaires issues de toutes les masses $S_i \in S$ présentes dans la simulation. En tenant également compte de la nature décalée de la grille de simulation, la force d'interaction pour D s'exprime donc comme :

$$F_D = \sum_{S_i \in S} F_{D,S_i} \cdot n_D$$

où n_D correspond à la normale à la face associée à D .

La force d'interaction F_D s'intègre alors naturellement à l'étape de prise en compte des forces externes au fluide, notée S4 dans la figure 3.13. Dans ce cas, le champ de force n'est toutefois plus uniforme, comme le montre la figure 3.22.

En examinant les expressions finales décrivant les forces d'interaction réintroduites dans les simulateurs, on remarque qu'elles sont bien indépendantes de la densité de marqueurs, ainsi que de la résolution de la grille, excepté pour le bruit variable introduit par l'intégration de Monte-Carlo. Cette convergence, à la limite, est illustrée par la figure 3.23 dans laquelle on analyse les forces subies par une masse en variant la densité de marqueurs.

On notera que l'interaction dépend du maillage des solides : un maillage deux fois plus dense génère deux fois plus de forces. On considère cependant que la modélisation masses-ressorts adoptée pour les solides ne procède pas d'une logique de discrétisation d'un objet continu, mais constitue une modélisation intrinsèquement discrète. À ce titre, on traite deux variantes plus ou moins denses d'un même objet comme deux solides différents, l'interaction étant alors naturellement différente pour les deux.

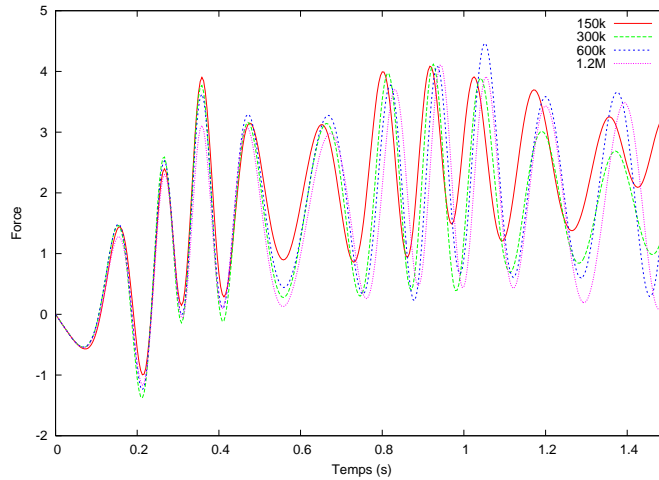


Figure 3.23: Convergence de la force d'interaction pour les masses. Force subie par la masse de la simulation de flottaison — illustrée dans la figure 3.25 — pour quatre densités différentes de marqueurs. Bien que des différences se développent en raison du caractère dynamique de la simulation, les différents calculs sont en bon accord.

Même s'il est possible de tenir compte de la subdivision plus ou moins grande du maillage des solides, en considérant une densité de force par unité de volume de fluide et par unité de volume de solide, il n'est pas nécessairement judicieux de le faire. En effet, un raffinement du maillage des solides s'accompagne généralement de la modification des paramètres de l'interaction, notamment de ses rayons, ce qui périmé de toute manière la notion de convergence de l'interaction.

Flexibilité de l'interaction Puisque l'interaction proposée est essentiellement empirique et ne découle pas directement d'un phénomène physique, on s'autorise à la modifier pour la rendre plus flexible. Le but de ces modifications est d'offrir un plus grand contrôle à un animateur pour lui faciliter la tâche de mise au point d'une animation, et d'en accélérer le calcul.

À l'usage, on se rend compte qu'il est difficile d'obtenir l'équilibre visuel voulu entre les mouvements des deux milieux. Cette constatation tend à prouver que le modèle d'interaction adopté ne répond pas entièrement aux lois de la physique, ce qui était prévisible eu égard à sa nature empirique, même s'il peut se révéler satisfaisant dans la pratique. Afin d'obtenir une plus grande flexibilité, on propose d'introduire une modification drastique et d'autoriser une interaction asymétrique. Deux facteurs multiplicatifs de l'interaction, ι_F et ι_S , un pour chaque milieu, sont ainsi mis à disposition de l'animateur et l'interaction « artistique » prend la forme :

$$\begin{aligned}\widetilde{F}_{S_i} &= \iota_S \cdot F_{S_i} \\ \widetilde{F}_D &= \iota_F \cdot F_D\end{aligned}$$

Choisir $\iota_S \neq \iota_F$ rend évidemment caduque toute considération physique de conservation pour la situation simulée, mais peut se révéler très utile dans un but artistique. En fait, le choix de ces coefficients ne détruit pas réellement la

physique sous-jacente, mais revient à modifier l'interaction ainsi qu'à changer de manière corrélée un certain nombre de paramètres du système, externes à l'interaction. L'introduction de ι_S et de ι_F offre donc à l'utilisateur un moyen de régler avantageusement l'équilibre de l'interaction par des paramètres propres, sans devoir manipuler de paramètres externes.

Dans leur forme générale, les forces reconstruites sont proportionnelles au nombre de marqueurs présents dans la simulation. Or, les marqueurs n'étant pas propres aux calculs d'interaction, il peut arriver que l'animateur désire altérer le nombre des marqueurs sans modifier l'interaction, ou inversement modifier l'interaction sans altérer les marqueurs. Il peut pour cela compter sur les corrections apportées par le facteur de densité de marqueurs η , mais on propose également de calculer la force d'interaction en n'employant qu'un sous-ensemble des marqueurs. Dans ce cas, η ne représente alors plus la densité de marqueurs du liquide, mais la densité de marqueurs supports de l'interaction.

Ce découplage entre représentation du fluide et calcul de l'interaction peut être mis à profit dans deux situations. Premièrement, il permet de minimiser le changement des propriétés dynamiques de l'interaction lors d'une augmentation du nombre de marqueurs; par exemple pour le passage d'une animation de basse résolution, destinée à la mise au point, à une animation finale de qualité, de résolution plus importante. Deuxièmement, et comme il sera montré dans la section 3.2.3 présentant des résultats de cette interaction, les calculs peuvent être coûteux si le nombre de marqueurs est grand. Le calcul de l'interaction sur un sous-ensemble des marqueurs autorise dans ce cas une réduction des coûts.

Calcul pratique Le principe de l'interaction étant simple, peu d'éléments sont à décrire pour son implémentation qui requiert seulement de savoir calculer les forces d'interaction point-à-point, et de les sommer.

Pour calculer les interactions point-à-point, il faut pouvoir évaluer la vitesse relative d'un marqueur et d'une masse. Par conséquent, les marqueurs doivent conserver, outre leur position actuelle, la position qu'ils ont occupée au pas de temps précédent. On note qu'il serait possible de se passer de cette information et de la dériver à la demande du champ de vitesse courant du fluide, mais on préfère éviter ce calcul supplémentaire.

Pour calculer la force d'interaction d'une manière efficace, on tire parti de sa localité spatiale. Le calcul se fait alors en deux temps : pour un marqueur donné, un test de proximité est d'abord réalisé avec chaque objet, et les forces ne sont calculées avec les différentes masses de l'objet que si le test révèle une possible interaction. Ce test de proximité est implémenté par un test d'inclusion du marqueur dans la boîte englobante de l'objet, élargie de la taille de l'interaction. Cette approche est retenue en raison de sa simplicité, mais des stratégies plus complexes d'accélération de la recherche des interactions non-nulles pourraient être implémentées, notamment par l'emploi de modes plus évolués de partitionnement de l'espace.

Pour gérer les collisions des objets entre eux, on a recours à un modèle de répulsion visco-élastique des masses, similaire à celui de l'interaction entre masses et marqueurs. Cette gestion des collisions n'est pas optimisée, et donc approximativement de complexité quadratique en nombre de masses. La gestion des collisions avec les obstacles est réalisée à l'aide d'une répulsion élastique par rapport aux obstacles de l'environnement.

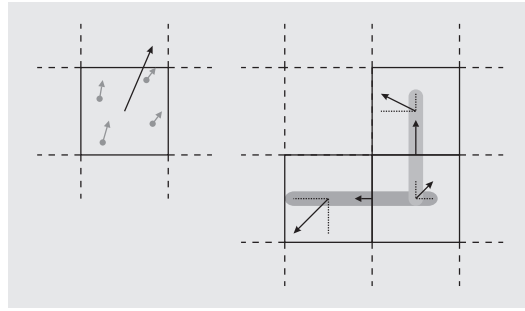


Figure 3.24: Reconstruction alternative du champ de force Eulérien. Les forces portées par les marqueurs sont sommées sur la base des cellules, pour toutes leurs composantes, en une seule étape, comme figuré à gauche. Ensuite, les forces de la discrétisation décalée, portées par les faces, sont dérivées des deux cellules adjacentes, pour chaque composante.

Pour tenir compte de la force d'interaction dans le fluide, on propose une variante de calcul destinée à gérer le décalage des différentes composantes dans la grille. Tout d'abord, les forces des marqueurs sont sommées sur la base des cellules de la grille discrétisant les scalaires, ce qui permet de reconstruire simultanément toutes les composantes du champ de force vectoriel. Ensuite, pour chaque point de discrétisation décalé associé à une face, la force est déterminée en considérant ce champ dans les deux cellules adjacentes, comme figuré sur la figure 3.24. Cette méthode, revient à lisser spatialement la force d'interaction sur deux cellules de la grille. Le comportement final ainsi obtenu par cette méthode est tout à fait satisfaisant et analogue à un calcul directement décalé. C'est d'ailleurs cette variante qui est employée dans les exemples présentés.

3.2.3 Résultats

Les figures 3.25–3.28 illustrent les résultats que l'on peut obtenir en mettant en œuvre l'interaction proposée. Ces divers résultats, qui paraissent réalistes, sont obtenus de manière totalement automatique après spécification des paramètres de la simulation.

Dans la mesure où les paramètres des interactions n'ont pas de signification physique et sont largement variables d'une simulation à l'autre, ils sont volontairement omis. Ils sont déterminés par un processus itératif de mise au point, jusqu'à déterminer un résultat visuellement convaincant et conforme à l'idée initiale à illustrer.

Pour la recherche de ces paramètres, on ne peut donner de règle générale, mais il faut veiller à choisir avec attention r et D_r , l'intensité de répulsion maximale et le rayon de répulsion, afin de repousser efficacement les marqueurs de l'intérieur des objets. Si l'intérieur des objets se trouve noyé par le fluide, aucun vide ne se développe dans le volume du liquide du point de vue du simulateur de fluide, et l'émulation de la poussée d'Archimède ne peut se développer comme dans le cas de la figure 3.25. Cette force de flottaison prend en effet naissance à partir de la différence de pression du fluide autour de l'objet : liée à la gravité, elle presse davantage les marqueurs sur l'objet en-dessous qu'au dessus de lui, le faisant ainsi émerger.

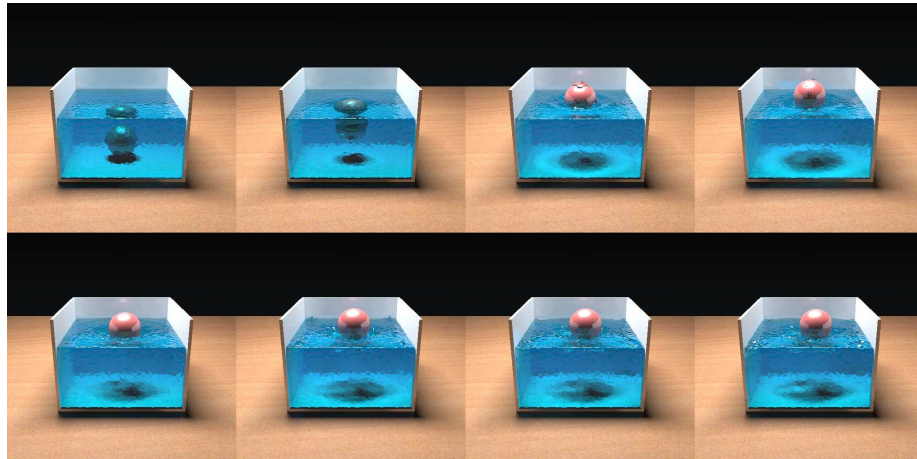


Figure 3.25: Exemple d'interaction : flottaison. La masse et le fluide sont initialement au repos. Toutefois, la différence de pression dans le fluide, due à la gravité, pousse davantage les marqueurs contre le solide en dessous qu'au-dessus de lui, ce qui le fait émerger.

La figure 3.26 présente un jet entrant en collision avec un cube composé d'un millier de masses. Cette simulation prouve qu'il est possible de localiser l'interaction de manière suffisamment précise pour déformer le cube à l'échelle des masses qui le composent. L'interaction est également illustrée, puisque le jet est dévié par le cube tout en le propulsant. Les deux figures 3.27 et 3.28 illustrent également cette interaction automatique avec des jeux entre solides et liquide. Dans la première, deux cubes de masses différentes interagissent par l'intermédiaire du liquide : celui-ci étant incompressible et confiné, il transfère le mouvement d'un cube à l'autre, comme dans un système de freins. Dans la seconde, on lance une boule représentée par une masse dans un bac de liquide, et on observe un ricochet : la boule rebondit sur la surface du liquide tout en la mettant en mouvement.

On donne dans la figure 3.29 quelques caractéristiques des simulations correspondant à ces animations. Les temps correspondent à des calculs réalisés avec un processeur Athlon XP 1800+ pour simuler $1/24$ de seconde, les pas de temps effectifs étant limités à $1/240$ de seconde au maximum. L'adoption de pas de temps réduits, pour éviter de faire diverger la simulation, est une conséquence des interactions très raides nécessaires entre fluide et solides. Les temps mentionnés comprennent la simulation et l'export des données brutes du simulateur *i.e.* le nuage de marqueurs et la position des solides, mais ne prennent en compte ni la reconstruction d'une surface, ni son rendu à l'aide du logiciel *Radiance*.

Le temps requis par l'interaction est extrêmement variable d'une simulation à l'autre, comme l'illustre la figure 3.30 et il dépend largement du nombre de masses introduites dans l'environnement. Comme on peut le voir, la plus grande partie du temps de calcul est employée au calcul des forces point-à-point. Au sein même d'une simulation, le temps requis peut varier, selon la configuration géométrique, comme illustré par la simulation des deux cubes dans la boîte en U. Initialement, les marqueurs sont loin des solides, et le temps de calcul des forces est minimal, mais on observe un accroissement du temps de calcul dès

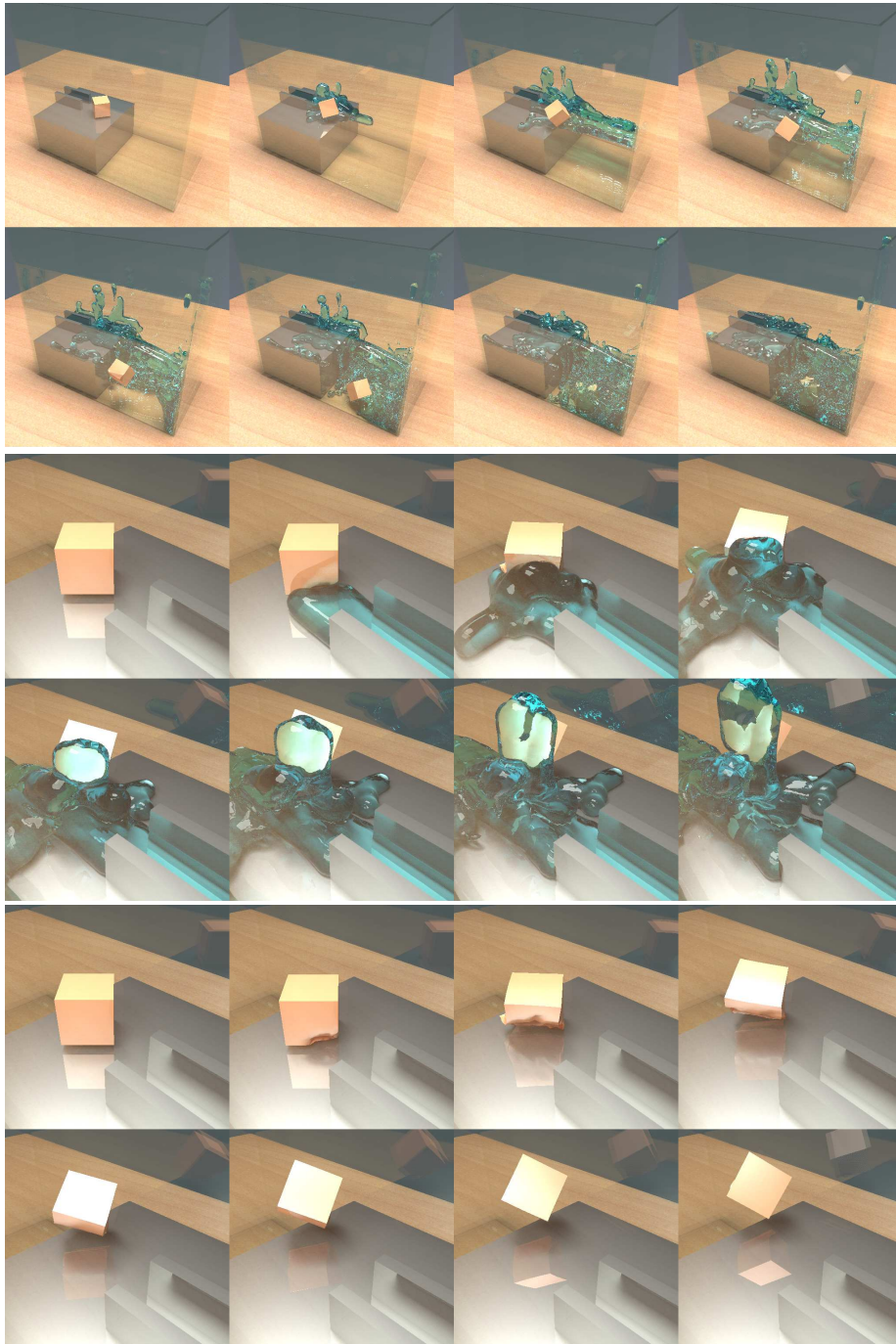


Figure 3.26: Exemple d'interaction : jet. Animation générale, et gros plan sur le cube en contre-champ durant les premiers pas de temps. Le gros plan est présenté avec, et sans le liquide, pour mieux apprécier la déformation du cube sous l'impact.

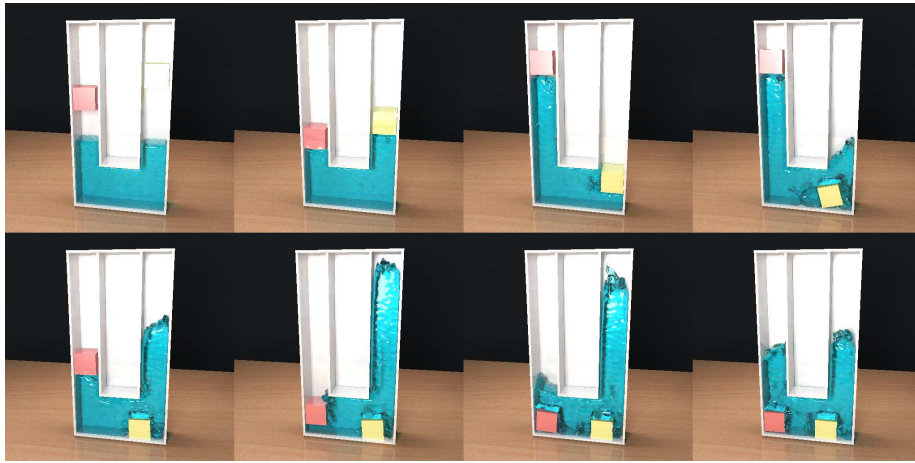


Figure 3.27: Exemple d'interaction : cubes. Le cube de droite est plus lourd que celui de gauche. Le liquide est mis en mouvement sous la pression exercée par les cubes, mais repousse également un cube sous la pression de l'autre en transmettant le mouvement.

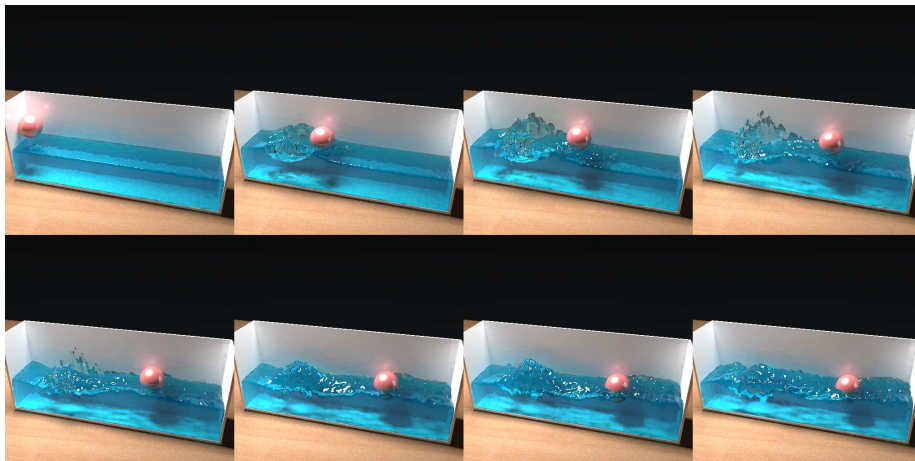


Figure 3.28: Exemple d'interaction : ricochet. La balle rebondit sur la surface du liquide tout en la mettant en mouvement.

Exemple	Flottaison	Jet	Cubes	Ricochet
Figure	3.25	3.26	3.27	3.28
Taille grille	40 × 50 × 40	40 × 40 × 20	20 × 40 × 5	100 × 40 × 20
Nb. total de marqueurs	4 676 065	800 000 / s	250 000	1 600 000
Nb. de masses	1	1000	2000	1
Marqueurs d'interaction	1/8	1/8	1/1	1/16
Temps de calcul moyen	45.4 s	12 s	8 s	23.2 s

Figure 3.29: Paramètres significatifs des exemples. Le nombre de marqueurs d'interaction est exprimé en fonction du nombre total de marqueurs. Le temps de calcul correspond à une moyenne réalisée à partir du temps de calcul complet des animations, pour une progression de 1/24 de seconde. Les pas de temps effectifs sont limités à 1/240 de seconde.

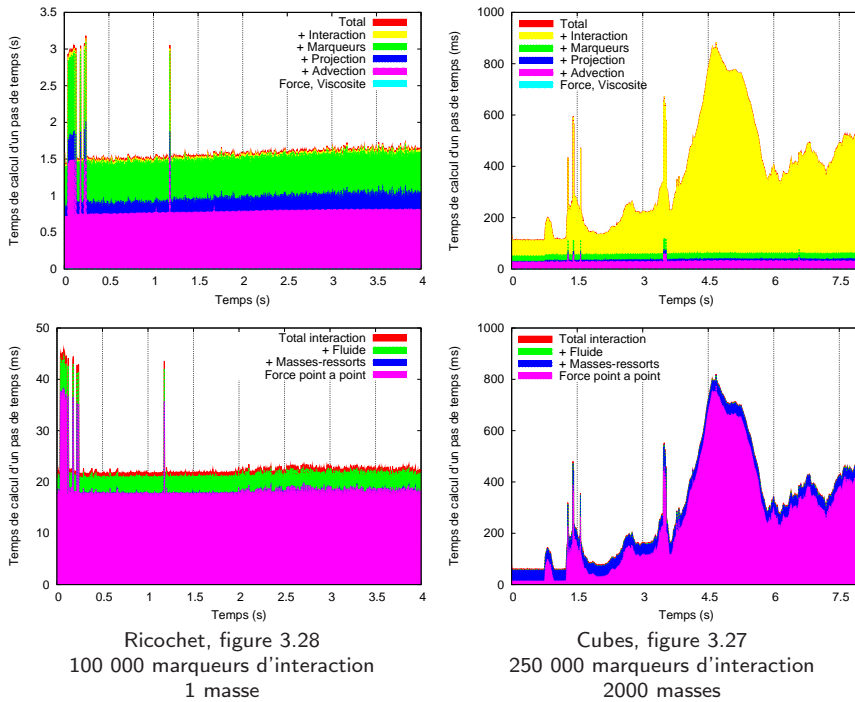


Figure 3.30: Temps requis par l'interaction pour deux simulations. Les graphiques du haut comparent le temps requis par l'interaction au temps du reste de la simulation, ceux du bas décomposent l'interaction entre le calcul des forces point-à-point et la réintroduction des forces dans les deux milieux. Les repères verticaux marquent les instants illustrés des animations. Les temps indiqués correspondent au calcul de 1/240 seconde : les pics correspondent à des instants où le pas de temps est subdivisé de manière adaptative pour satisfaire à la condition CFL.

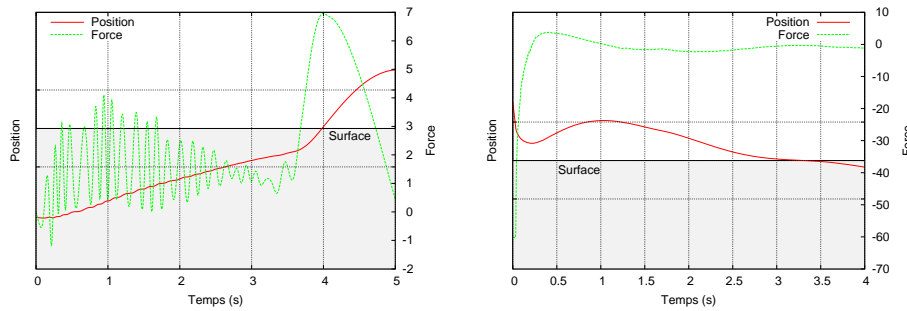


Figure 3.31: Exemples de forces d'interaction correspondant à la simulation de flottaison — figure 3.25, à gauche — et de ricochet — figure 3.28, à droite — La position de la masse est présentée, tout comme la force totale subie, en fonction de l'instant de simulation. La force inclut donc l'interaction et la gravité. Les repères horizontaux marquent la position moyenne de la surface du fluide au repos, ainsi que les rayons d'interaction : le solide commence à émerger quand sa position égale le repère inférieur, et ne perçoit plus aucune force quand il dépasse le repère supérieur. Ces repères sont approximatifs, en raison des mouvements propres de la surface.

lors que les interactions sont effectives.

Comme illustré dans la figure 3.31, qui montre la force subie par la boule dans la simulation de flottaison — figure 3.25 — on constate que la force d'interaction est relativement peu stable dans cet exemple et a une forte tendance à osciller. Dans ce cas, l'interaction est fortement répulsive, sans viscosité. La force d'interaction a donc du mal à représenter correctement cette situation proche de l'équilibre, en s'appuyant sur l'échantillonnage de la force à partir des marqueurs, en tous les cas pour la taille de grille et les paramètres de simulation qui ont été employés. On notera que cette oscillation ne coïncide ni avec la résolution de la grille, ni avec un mouvement périodique visible des marqueurs. Néanmoins, ces oscillations étant de fréquence importante, elles ne sont pas nécessairement perceptibles sur la trajectoire du solide à cause de son inertie : globalement, le solide adopte tout de même une trajectoire plausible. En outre, et comme on peut le voir sur la deuxième partie de la figure, dans le cas de mouvements plus vifs, comme lors du ricochet — figure 3.28 — ces oscillations ne sont pas toujours observées.

3.3 Bilan

On cherche maintenant à dresser le bilan du modèle d'interaction présenté. Afin de replacer ces travaux dans leur contexte, et d'expliquer leur approche que l'on peut juger primitive, on rappelle qu'il constitue en infographie une des premières tentatives pour considérer l'interaction de manière automatique, dans une simulation de fluide Eulérienne [51].

Même s'il intègre le cadre d'une méthode de simulation physique de fluide éprouvée, le modèle choisi ne constitue qu'un modèle empirique de l'interaction, construit à dessein. À ce titre, il ne prétend pas simuler l'interaction entre liquide et solides d'une manière réaliste sur le plan quantitatif, mais d'une manière plausible sur le plan visuel. La démarche adoptée, depuis le choix d'une

modélisation *ad-hoc* jusqu'à la conception de la forme d'interaction, en passant par le choix de la modélisation des solides, suit des techniques qui ont fait leurs preuves en infographie, principalement dans le domaine des systèmes de particules. Du point de vue de la synthèse d'images, on peut affirmer sans grande réserve que les résultats obtenus par une telle approche sont probants. Dans l'objectif d'un réalisme visuel, cette méthode présente un intérêt certain pour augmenter l'attrait des simulations de liquide.

Néanmoins, cette méthode n'est pas exempte de critique. En effet, si elle se révèle simple, c'est surtout en raison d'une démarche de modélisation « par intention », toutes les propriétés physiques reposant sur les paramètres de force choisis par l'animateur. L'animateur espère ensuite que cette force est suffisamment bien définie pour faire émerger les comportements attendus. Par conséquent, cette méthode, par elle-même, ne garantit pas forcément le réalisme dans l'interaction, même si elle s'y emploie.

Concernant en particulier l'imperméabilité des objets, aucune contrainte formelle n'est posée pour la respecter : les objets introduits dans le domaine de simulation sont vus comme une forme de perturbation externe de la simulation du fluide, et non comme une restriction du domaine de simulation. Les objets ne pourraient d'ailleurs pas être rendus rigoureusement imperméables, puisque l'interaction repose sur le fait même que des particules approchent suffisamment près des masses pour déclencher une interaction. D'ailleurs, du point de vue de la simulation, les solides ne possèdent pas de surface et sont réduits à un assemblage de masses ponctuelles, ce qui empêche toute définition claire de l'imperméabilité.

Cette approche empirique engendre également un long et fastidieux processus de mise au point des animations, dans lequel il est nécessaire d'ajuster les paramètres de la simulation de manière très incrémentale. Compte tenu de la durée des temps de calcul des simulations, la conception d'une animation se révèle alors inefficace, d'autant plus que les paramètres choisis doivent être satisfaisants pour toute la durée de l'animation. En ce sens, on peut dire que cette méthode n'atteint pas l'objectif de faciliter la tâche d'un animateur, même si elle autorise l'inclusion de l'interaction, jusque-là absente.

Si ces remarques concernent la méthode dans sa globalité, on peut également formuler d'autres observations, sur des points plus précis.

Tout d'abord, et même si aucune voxelisation des objets n'est réalisée, les forces perçues par le fluide s'en tiennent à la résolution de la grille de simulation. La non-voxelisation des solides reste justifiée par l'absence de sauts induits dans l'imposition de conditions aux limites, mais au détriment d'une réelle imperméabilité des solides. Du point de vue des solides, l'interaction, perçue à l'échelle des masses, est plus fine que celle des cellules de la grille.

Ensuite, le mode d'interaction proposé a du mal à capturer une interaction très fine entre liquide et solides. Si l'obtention d'une animation comportant des mouvements violents et réalistes est relativement aisée, l'obtention de jeux subtils entre liquide et solides relève, elle, de la gageure, sinon du domaine de l'impossible. Ainsi, il est difficile de réaliser la simulation d'un objet affleurant à la surface du liquide et oscillant au gré des vagues. Une modélisation trop grossière, ou mal adaptée, de l'interaction comme du liquide, en est certainement la cause.

Finalement, dans le domaine d'application de la méthode proposée, on n'a pas étudié le problème de l'interaction avec un liquide représenté par *levelset*,

représentation du fluide qui s'est largement imposée comme étant l'une des meilleures. Il ne semble toutefois pas exister d'incompatibilité entre la méthode proposée et cette représentation, surtout si le fluide adopte une représentation hybride à base de particules et d'un *levelset*.

En reconsidérant globalement les limitations de l'approche suivie, on peut rattacher la plupart des défauts de la méthode d'interaction proposée essentiellement à deux points.

Tout d'abord, aucun problème clair, au sens mathématique du terme, n'a été posé, et le résultat découle uniquement d'une démarche de modélisation empirique « par intention ». La modélisation étant conçue dans cette optique, elle engendre bien évidemment des résultats qui peuvent être satisfaisants, comme on a pu le voir, mais la méthode reste limitée à une émulation de simulation physique réelle. Les résultats obtenus ne sont donc pas toujours réalistes, et la méthode n'est pas toujours robuste. En outre, l'absence de problème clairement posé implique l'absence de solutions bien définies. Il faut donc, à l'image de la physique, réellement poser le problème de l'interaction fluide-structure, et ne plus se contenter de l'émuler.

Ensuite, il n'est pas possible, avec une simulation MAC, de représenter fidèlement des domaines arbitraires, ce qui entrave considérablement la possibilité de modéliser correctement l'interaction entre deux milieux. On notera que l'on s'est référé ici aux simulateurs employant des grilles régulières : des modes de simulations plus récents offrent la possibilité de réaliser des simulations dans des domaines arbitraires, ou représentés hiérarchiquement.

Ces deux conclusions nous conduisent à explorer le domaine des simulations par éléments finis, travail qui sera détaillé dans le chapitre 5. On attend alors, de l'utilisation rigoureuse de cette méthode, des simulations sinon immédiatement performantes, du moins robustes.

Chapitre 4

Simulation MAC : Visualisation

Comme on a pu le voir dans le chapitre précédent, la force de l'approche MAC, pour simuler un liquide à surface libre, est de ne pas maintenir de représentation explicite de la surface du liquide, mais de se reposer sur un nuage volumique de marqueurs. L'absence de représentation explicite de la surface pose cependant problème dès lors qu'il faut visualiser ces simulations.

Approches fréquemment suivies Pour traiter le problème d'obtention de surface, l'approche usuelle est de définir une surface implicite dont le squelette est constitué des marqueurs.

Une autre approche plus récente, et qui fait référence dans le cadre de l'infographie, consiste à ne pas utiliser les seuls marqueurs mais à faire évoluer directement la surface du liquide comme une fonction implicite, par une formulation de type *levelset*. Cette avancée majeure dans la représentation des surfaces de liquides a été décrite dans la section 2.3.2.

Que la surface implicite découle des marqueurs ou qu'elle soit associée à un *levelset*, elle peut être visualisée directement, ou alors on peut en extraire une surface triangulée explicite en utilisant un algorithme de type *Marching-Cubes*.

Situations étudiées Si ce type de représentation implicite par *levelset* se révèle visuellement très intéressant, il introduit néanmoins un coût supplémentaire dans la simulation, tant au niveau de son implémentation qu'au niveau de son utilisation. L'emploi de surfaces définies *a posteriori* pour chaque pas de temps se révèle également coûteux et pose des problèmes de cohérence, comme décrit dans la section 3.1.8. Au vu des résultats plus que satisfaisants qui sont obtenus dans le domaine du rendu de nuages de points décrivant des surfaces [133, 191], s'appuyer sur une visualisation plus directe du nuage de marqueurs, sans passer par une représentation intermédiaire, semble une approche intéressante.

Ce chapitre présente des travaux que nous avons réalisés pour visualiser les résultats de simulations MAC, et qui s'appuient sur cette utilisation directe du nuage de marqueurs matérialisant le liquide. Deux problèmes distincts sont abordés : la visualisation réaliste par raytracing d'une animation et la visualisation rapide dédiée au prototypage.

4.1 Rendu direct de nuage de points par raytracing

Cette section présente un algorithme que nous avons développé pour pouvoir intégrer aux techniques de rendu photo-réaliste des objets définis comme des nuages de points volumiques.

Au préalable, on admet que l'opération primitive de tout rendu photo-réaliste consiste en la simulation de la propagation de la lumière, qui se traduit par le calcul d'intersection entre une demi-droite et la surface de l'objet à visualiser. Par ailleurs, la performance d'un rendu dépend directement de la vitesse de cette opération.

Il est utile de rappeler deux propriétés particulières des volumes de liquide auxquels on s'intéresse. Premièrement, ils sont définis comme un nuage de points volumique et pas uniquement surfacique. Ensuite, de par leur nature de liquide, ces objets sont représentés avec une forme de transparence, et donc la lumière se propage dans leur volume. Par conséquent, l'algorithme de rendu considère des rayons externes à l'objet, aussi bien que des rayons internes.

Le problème posé est donc de considérer un algorithme capable de réaliser ces intersections de manière efficace aussi bien dans le cas de rayons internes qu'externes à l'objet.

Après avoir décrit le modèle affecté aux nuages de points, et avoir expliqué pourquoi les approches traditionnelles ne sont pas adaptées pour traiter ces données efficacement, l'algorithme que nous proposons sera décrit dans la section 4.1.2. Une variante dans la représentation de la surface sera également présentée dans la section 4.1.3. La construction des objets nécessaires au fonctionnement de l'algorithme proposé sera ensuite détaillé dans la section 4.1.4, avant la présentation de résultats dans la section 4.1.5.

4.1.1 Situation

Modèle associé aux nuages de points

En raison de l'absence de définition intrinsèque d'un objet modélisé par un nuage de points, il est nécessaire d'en fixer une. On choisit la définition communément admise — hors celle recourant à une surface implicite — que le volume de l'objet est assimilable à une collection de primitives centrées sur les points du nuage. On retient également la primitive la plus simple qui soit : une sphère. L'objet X est alors défini comme l'union de toutes les sphères S_i d'un rayon r donné, centrées sur les points $x_i \in N_X$ composant le nuage N_X donné :

$$X = \bigcup_{x_i \in N_X} S_i = \bigcup_{x_i \in N_X} \{x : |x - x_i| \leq r\}$$

Notons que le nuage de points est nécessairement fini, et qu'en conséquence l'objet X généré par une telle définition peut toujours être borné par une boîte englobante de dimension finie.

Dès lors que l'on considère des notions telles que l'intérieur ou l'extérieur de l'objet X , c'est la définition suivante qui prévaut, et qui permet d'exprimer le critère d'appartenance d'un point à l'objet, à savoir :

$$p \in X \iff \exists x \in N_X : |p - x| \leq r$$

On considère également que chaque point du nuage porte une orientation, qui correspond à la normale de la surface attendue au nuage, proche du point. Pour les points internes aux nuages, cette orientation est sans importance et non définie.

Cette normale est employée de deux manières distinctes. Dans un premier temps, elle est utilisée pour définir la normale qui sera renvoyée à l'algorithme de rendu hôte. Dans un deuxième temps, on fera complètement abstraction de la définition volumique et cette seconde définition, surfacique, sera employée dans le cadre des calculs d'intersection en considérant l'objet comme une collection de disques et non plus de sphères. On exposera ultérieurement le mode de calcul de cette orientation.

Inadéquation des approches existantes

Compte tenu de la définition précédemment adoptée pour les objets, deux procédures peuvent être envisagées pour résoudre le problème du calcul d'intersection. Après leur description sera détaillé le problème de leur performance.

La première approche, découlant de la définition donnée, consiste à calculer effectivement, pour chaque sphère définie par le nuage, ses intersections avec le rayon. Une fois les paires d'intersections déterminées, l'union de tous les intervalles ainsi définis permet de déterminer les intersections effectives du rayon avec l'objet.

La seconde approche relève de la recherche d'une racine de fonction sur un intervalle, et consiste à déterminer la présence d'intersections par l'échantillonnage, le long du rayon, de la fonction décrivant l'appartenance ou non d'un point à l'objet. Une intersection avec la surface existe nécessairement entre deux points-échantillons si la réponse au prédicat d'appartenance y trouve une réponse opposée. Dans ce cas, la technique ne garantit pas de trouver toutes les intersections et la précision de la technique est liée à la fréquence d'échantillonnage choisie.

Dans le cas considéré, où l'objet est assimilable à une collection de primitives, l'efficacité de ces modes de calculs est habituellement gagnée grâce à un partitionnement spatial des primitives, dans le but de restreindre les calculs d'intersection à un sous-ensemble extrêmement réduit de l'ensemble des primitives. Le principe en est le suivant : avant tout calcul d'intersection, un partitionnement de l'espace est considéré, et pour chaque cellule de la partition, la liste des primitives présentes dans la cellule est construite. Ensuite, lors d'un calcul d'intersection, les primitives listées dans chacune des cellules percées successivement par le rayon sont examinées, jusqu'à trouver une intersection effective. Ainsi seules les primitives enregistrées dans les cellules effectivement traversées par un rayon sont testées, les autres, qui ne sont manifestement pas en mesure d'intersecter le rayon, sont ignorées.

Intuitivement, une telle optimisation peut se révéler efficace pour accélérer les techniques décrites dans le cas d'un rayon voyageant à l'extérieur de l'objet : toutes les cellules externes à un objet étant vides, le rayon peut traverser ce vide sans calcul autre que celui du parcours des cellules. En revanche, cette optimisation se trouve mise en défaut dans le cas d'un rayon interne à l'objet : à l'intérieur d'un objet muni de la modélisation choisie, toutes les cellules enregistrent un nombre non-négligeable de primitives qui y sont présentes, engendrant par là-même des calculs. Bien qu'une optimisation non négligeable soit

déjà gagnée, l'intérieur de l'objet ne peut donc être traversé aussi rapidement que l'extérieur.

Problème soulevé

Pour obtenir un calcul performant, il est nécessaire de ne considérer que la surface du nuage, et non son volume, les calculs d'intersections se concentrant naturellement dans la seule zone potentiellement intéressante qu'est la surface, et les volumes intermédiaires étant ignorés. Le problème est qu'avec le modèle volumique d'objet adopté, aucune définition explicite de la surface n'est disponible, et celle-ci doit être détectée par la recherche d'une transition entre intérieur et extérieur.

Comme on l'a vu, la détection de cette transition est efficace pour l'entrée dans l'objet, puisque le partitionnement spatial de l'objet permet d'ignorer efficacement toute zone externe dépourvue de primitive. Par contre, évaluer une transition de sortie d'un objet équivaut à trouver le point où cesse l'intérieur du volume, ce qui implique de détecter une transition vers le vide environnant. L'absence de description explicite du vide environnant en interdit logiquement la recherche explicite, ce qui a pour effet d'empêcher le recours à un partitionnement spatial pour en accélérer la recherche. Pour évaluer la sortie de l'objet, on est donc contraint d'évaluer explicitement la poursuite du volume.

On note bien que cette dissymétrie ne provient pas de la définition par collection de primitives, mais de la primitive choisie. Si des cubes au lieu des sphères sont employés pour primitives — c'est-à-dire si la distance Euclidienne $|\cdot|_2$ est remplacée par la distance $|\cdot|_\infty$ — le vide environnant l'objet peut être défini comme une collection explicite de parallélépipèdes.

Ce problème ne se pose pas dans le cas de nuages de points uniquement distribués sur la surface des objets, puisqu'aucune primitive n'est alors présente à l'intérieur même de l'objet. Dans ce cas, le partitionnement spatial décrit peut être employé efficacement à l'intérieur comme à l'extérieur de l'objet, tel que le décrivent Schaufler et Jensen [133], les deux environnements étant analogues.

4.1.2 Algorithme proposé

L'algorithme que nous proposons permet de résoudre ce problème de performance lors du parcours de l'intérieur des nuages de points, et de rendre la traversée de l'intérieur de l'objet aussi performante que celle de son extérieur.

Principe

La technique proposée s'appuie sur la constatation que chercher l'intersection de sortie d'un objet équivaut à chercher l'entrée dans le vide environnant. Or, la détermination d'une intersection d'entrée dans un nuage de points peut être rendue performante en recourant à un partitionnement spatial.

L'approche développée repose donc sur la création d'un objet dual \bar{X} dont le rôle est de modéliser explicitement le vide dans lequel l'objet primal X est plongé. Dans ce cas, le vide primal est équivalent au volume de l'objet dual, et le volume de l'objet primal est équivalent au vide dual. Par suite, l'entrée d'un rayon dans l'objet primal équivaut alors à sa sortie de l'objet dual, et *vice versa*, comme illustré dans la figure 4.1.

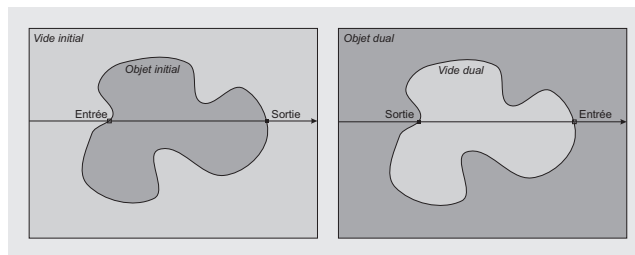


Figure 4.1: Dualité

Précalcul

- P1. Partitionnement de l'objet primal X
- P2. Génération de l'objet dual \bar{X} et calcul des normales de X et de \bar{X}
- P3. Partitionnement de l'objet dual \bar{X}

Recherche d'intersections

- I0. Si le rayon est externe à l'objet X , I1, sinon I2.
- I1. Détection d'une entrée dans l'objet primal, par recherche d'une entrée dans l'objet primal
- I2. Détection d'une sortie de l'objet primal, par recherche d'une entrée dans l'objet dual

Figure 4.2: Algorithme de recherche d'intersections.

La représentation de l'objet dual \bar{X} est choisie de même nature que celle de l'objet primal X , c'est-à-dire par un nuage de points $N_{\bar{X}}$. Ceci en autorise une création simple ainsi qu'un algorithme globalement homogène. Un fois créé, l'objet dual est partitionné spatialement, tout comme l'objet primal.

En modélisant ainsi le vide, il est possible de créer un algorithme de recherche d'intersections aussi rapide à l'extérieur qu'à l'intérieur de l'objet, algorithme dans lequel la recherche d'une intersection est toujours ramenée à la recherche d'une intersection d'entrée. La recherche est alors conduite avec l'optique la plus adaptée. Ainsi la recherche d'une intersection d'entrée dans le nuage primal — le passage du vide primal au volume primal — est effectuée rapidement dans la vision primale — alors que la recherche d'une intersection de sortie du nuage primal — le passage du volume primal au vide primal — est conduite efficacement dans la vision duale.

L'algorithme ainsi mis en place est illustré dans la figure 4.2. La recherche précise des intersections est décrite ci-après.

Recherche d'intersections

Étant donné un des deux objets, primal ou dual, la recherche d'une intersection d'entrée est accélérée par le partitionnement spatial précalculé. Elle repose sur le parcours ordonné des différentes cellules du partitionnement percées par le rayon, depuis son origine. Pour chacune, sont calculées les intersections avec les primitives qui y sont enregistrées. L'intersection la plus proche de l'origine, trouvée dans la première région en contenant une, est l'intersection d'entrée

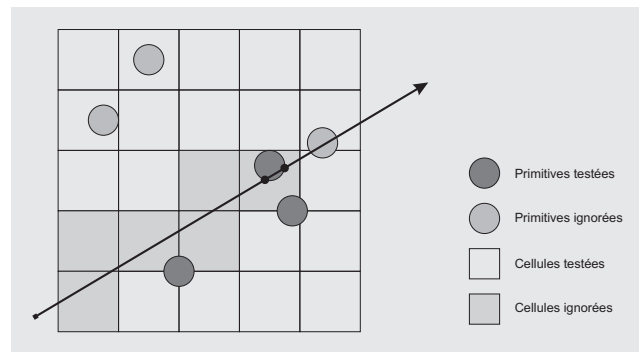


Figure 4.3: Partitionnement pour la recherche d'intersections. Seules les primitives des cellules traversées par le rayon subissent des tests d'intersection, les autres sont ignorées.

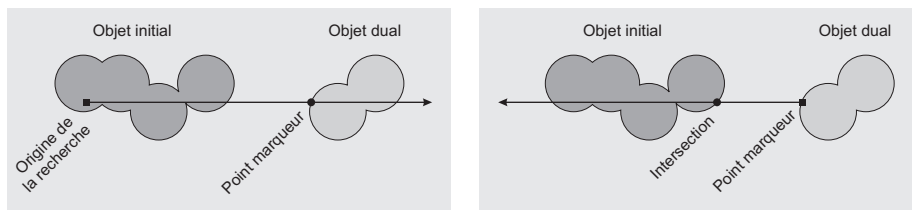


Figure 4.4: Double recherche de l'intersection de sortie. L'objet dual permet de rechercher un point marqueur externe proche de la sortie attendue. Ensuite, une recherche dans le sens inverse, à partir du point marqueur, permet de localiser la sortie attendue.

recherchée, comme illustré dans la figure 4.3.

Pour déterminer une intersection d'entrée dans l'objet primal, *i.e.* dans le cas I1 de l'algorithme, cette technique est immédiatement utilisable et fournit le résultat attendu.

Toutefois, en considérant une définition de l'objet dual de même nature que celle de l'objet primal, sous forme de collection de sphères, il n'est pas possible de faire exactement coïncider les surfaces des deux objets. Par conséquent, la dualité des entrées et sorties n'est pas parfaitement respectée. Compte tenu de ce fait, deux variantes sont proposées pour calculer une intersection de sortie, *i.e.* le cas I2 de l'algorithme : l'une incluant une procédure corrigeant ce manque de cohérence, l'autre non.

Dans la première variante, illustrée par la figure 4.4, l'objet dual est simplement utilisé comme marqueur de l'espace vide, et permet de s'approcher rapidement de la surface de l'objet primal. Une première recherche d'entrée dans l'espace dual est donc employée, selon le rayon, pour fournir un point marqueur, externe à l'objet primal mais proche de l'intersection désirée. Ce point marque alors le premier point du rayon trouvé hors du volume de l'objet initial depuis l'origine de la recherche. Partant de ce point, une deuxième recherche d'intersection, mais avec l'objet primal et dans la direction opposée est ensuite effectuée, et celle-ci produit le point attendu de sortie de l'objet initial. On notera que si aucun point marqueur intermédiaire n'est trouvé, ce point intermédiaire est

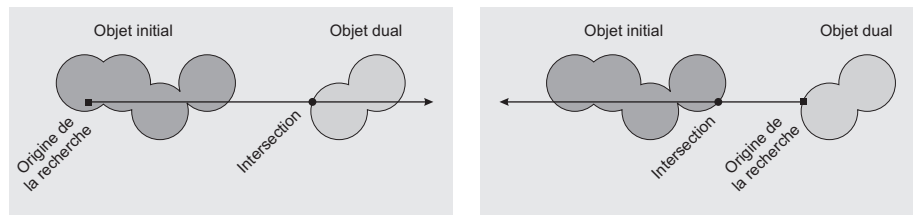


Figure 4.5: Dépendance du calcul d'intersection au point de vue.

défini par intersection du rayon avec la boîte englobante de l'objet primal. Ce cas, qui peut se produire à proximité de la boîte englobante si l'objet dual est peu dense, ne change rien à la recherche d'intersection.

La deuxième variante consiste à se satisfaire du point marqueur comme intersection de sortie. Dans ce cas, même si la dualité n'est pas géométriquement respectée, elle est supposée exacte. Cette variante est intéressante par la symétrie qu'elle fait apparaître entre la détection des intersections d'entrée et celle des intersections de sortie. Elles se résument toutes deux à la recherche d'une intersection d'entrée, dans l'objet primal ou l'objet dual.

L'adoption de cette deuxième variante engendre toutefois une position d'intersection légèrement variable en fonction du point de vue de l'observateur, comme l'illustre la figure 4.5. Cette hypothèse est raisonnable si l'objet dual est suffisamment dense, pour que le point marqueur, trouvé à l'aide de l'objet dual, soit toujours proche du point de sortie attendu. Dans ce cas, le manque de précision engendré est compensé par un gain de vitesse, puisque l'étape de la recherche finale est complètement supprimée.

On notera que l'on a, pour le moment, déterminé uniquement la position de l'intersection : le calcul de la normale sera discuté ultérieurement.

Contrainte induite sur le raytracer-hôte

L'inclusion d'une telle recherche d'intersection dans le contexte d'un algorithme de lancé de rayon introduit une seule et unique contrainte : connaître pour chaque rayon s'il voyage à l'intérieur ou l'extérieur de l'objet. Le raytracer incorporant la technique proposée doit donc être en mesure de fournir cette information rapidement, sans coût de calcul supplémentaire, lors du calcul de chaque rayon.

4.1.3 Surface

Modèle de surface

Comme précisé lors de la définition de l'objet dérivé d'un nuage de points, on considère également une définition alternative, à base de disques plutôt que de sphères. L'objectif de cette définition alternative est de représenter plus fidèlement la surface du nuage de points. L'obtention des directions associées aux points sera détaillée ultérieurement, dans la section 4.1.4.

Le recours à des disques, dans le contexte d'un nuage volumique de points, a comme principal inconvénient de ne pas autoriser la définition d'un volume. En effet, assimilables eux-mêmes à des éléments de surface, ils ne peuvent définir

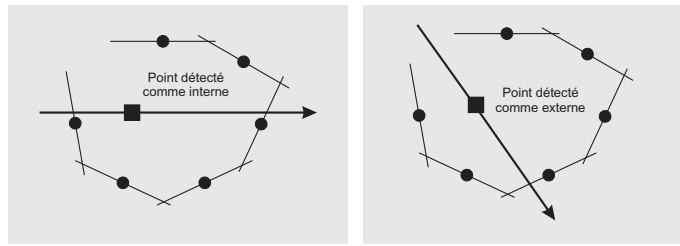


Figure 4.6: Ambiguïté du volume de l'objet. Le volume de l'objet ne peut être défini sans ambiguïté dès lors que la surface n'est pas fermée.

correctement qu'une surface. Toutefois, par conception, l'algorithme d'intersection que nous proposons n'est pas fondé sur une définition du volume des nuages, mais sur une définition de leur surface. Il est donc, sans aucune modification, capable de traiter de manière satisfaisante des objets représentés par une collection de disques. Dans ce cas, les disques les plus extérieurs doivent être orientés pour approximer correctement la surface, l'orientation des disques intérieurs étant sans aucune importance.

Si l'algorithme proposé peut intrinsèquement se satisfaire de disques, un problème d'ambiguïté de définition de l'objet subsiste néanmoins. La caractérisation de l'intérieur et de l'extérieur s'effectuant alors en examinant les intersections avec la surface, un point de l'espace peut se retrouver interne ou externe à l'objet en fonction des rayons. Cette situation survient si la surface définie par les disques n'est pas complètement fermée, comme dans la figure 4.6.

L'avantage évident des disques, par rapport aux sphères, est qu'ils offrent une meilleure approximation de la surface attendue du nuage de points. Ils empâtent moins que les sphères la surface générée et celle-ci est donc plus fidèle à la surface attendue. Le revers de l'usage des disques est l'apparition plus fréquente de trous dans la surface, puisqu'à rayon égal, la surface des disques projetée à l'image est nécessairement plus petite que celle des sphères.

Normale

Que le nuage soit associé à une collection de disques ou à une collection de sphères, il reste primordial, du point de vue du rendu, qu'une normale suffisamment lisse soit associée à la surface. En effet, c'est elle qui conditionne essentiellement l'apparence finale des points visibles, à travers les calculs d'illumination locale comme à travers les directions des rayons propagés. Une utilisation directe des normales géométriques dérivées des primitives est donc impossible, puisque les normales seraient, à l'image, discontinues et constantes par morceaux, comme le montre la figure 4.7.

Une technique capable de fournir des normales plus lisses est donc requise. Pour cela, on reprend la technique proposée par Schauffer et Jensen [133] dans le cadre de nuages surfaciques de points associés à une collection de disques, technique qui interpole les normales associées aux différents disques.

En considérant l'intersection détectée d'un rayon avec un disque, la normale lissée est calculée en considérant tous les disques percés par le rayon dans un voisinage de l'intersection détectée, comme figuré dans la figure 4.8. La direction de la normale n est alors évaluée par pondération des différentes normales n_i

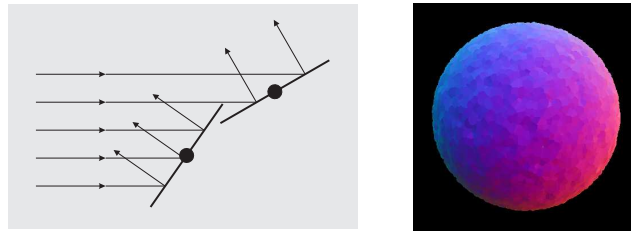


Figure 4.7: Discontinuité de la normale géométrique. L'image de droite présente les coordonnées de la normale codées sous forme de couleur.

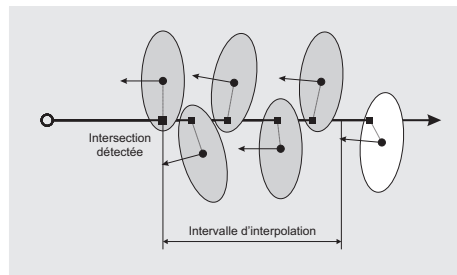


Figure 4.8: Interpolation des normales proposée par Schaufler et Jensen [133]. Sont interpolées les normales des disques percés par le rayon sur un intervalle donné après l'intersection détectée. La pondération associée à chaque normale est fonction de la distance des échantillons au rayon, dans le plan des disques.

correspondantes aux disques percés, en fonction de la proximité des points du nuage au rayon :

$$n = \frac{\sum_i n_i \cdot (r - d_i)}{\sum_i (r - d_i)}$$

où les distances d_i des points au rayon sont mesurées dans les plans des disques.

Dans le cas des sphères, on propose de lisser les normales associées aux points situés dans un voisinage sphérique de l'intersection détectée.

Position de la surface

S'il est tentant d'appliquer la technique de lissage précédente à la position même des intersections, pour produire une surface plus continue et plus lisse, il n'est absolument pas judicieux de le faire. En effet, tout lissage repousse nécessairement l'intersection vers l'intérieur de l'objet. Même si la migration de l'intersection est faible en termes de distance, elle a des conséquences importantes car elle peut entraîner des comportements erronés dans la propagation ultérieure de rayons.

Ces erreurs apparaissent pour les rayons renvoyés dans leur milieu d'origine, et sont la conséquence d'intersections détectées avec des disques proches, intersections qui ne sont pas attendues, comme le présente la figure 4.9. Ce problème se rapproche de celui de la gestion des imprécisions numériques traditionnelles des calculs d'intersections, mais se trouve ici exacerbé par le caractère flou de la surface, et le fait qu'elle est composée d'un nombre important de sous-primitives.

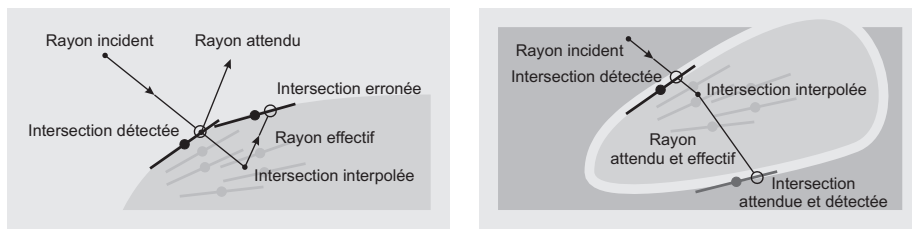


Figure 4.9: Interpolation de la position de la surface. Si la position de la surface est interpolée, les points d'intersection migrent dans l'objet primal. Cela se traduit par des intersections erronées pour les rayons repartant vers l'extérieur, comme figuré à gauche. Ce n'est pas le cas pour des rayons internes, comme figuré à droite, puisque ceux-ci considèrent alors les primitives duales.

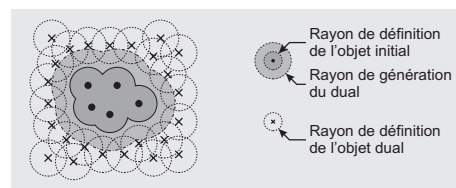


Figure 4.10: Génération du dual. L'objet dual est construit en échantillonnant l'espace environnant le volume primal, ici stochastiquement. La région échantillonnée correspond à une restriction du vide environnant pour éviter la superposition des objets primal et dual.

On peut noter, dans la même figure, que ce problème ne se pose pas pour les rayons franchissant l'interface, puisqu'aucun disque proche n'existe et ne peut venir interférer, que l'origine du rayon soit repoussée ou non.

4.1.4 Construction des objets intermédiaires

Construction de l'objet dual

Après avoir décrit l'usage qui est fait de l'objet dual, on en expose maintenant la construction. L'objet dual étant conçu pour matérialiser le vide environnant l'objet primal, sa construction s'apparente à un échantillonnage de l'espace extérieur au volume du nuage de points.

Il est absolument nécessaire au bon déroulement de l'algorithme que les objets primal et dual ne s'interpénètrent pas. Par contre, un intervalle n'appartenant ni à l'un, ni à l'autre, ne remet pas en cause l'algorithme, sauf éventuellement en termes de précision s'il est de taille importante. Dans le cas de sphères de rayon r , aucun point dual ne peut donc se trouver à moins d'une distance $2r$ d'un point primal. C'est d'ailleurs cette définition qui est employée pour définir le volume dual à échantillonner, comme l'illustre la figure 4.10. Dans le cas de disques, il est possible de diminuer cette distance minimale.

On notera que, cette phase de génération du dual relevant du précalcul, on peut recourir à des définitions du volume dual plus complexes, sans pénaliser ultérieurement la vitesse du rendu lui-même. On peut par exemple définir le volume dual comme l'extérieur d'un objet implicite dont le squelette est composé

des points primaux, dès lors que les contraintes de non-pénétration des deux objets sont respectées. Une telle définition peut permettre la génération d'un dual plus lisse, ce qui peut se révéler intéressant selon l'utilisation qui en est faite, simple marqueur du vide ou surface de sortie.

Pour réaliser l'échantillonnage effectif, on propose d'employer un échantillonnage régulier ou stochastique. On préférera en général la version stochastique dont le caractère progressif permet de l'interrompre à tout moment en conservant un échantillonnage partiel pertinent. Bien évidemment, un tel échantillonnage n'est pas en mesure d'apporter une quelconque garantie quant à la détection des vides, mais il se révèle parfaitement satisfaisant à l'usage. Si l'échantillonnage régulier peut théoriquement offrir des garanties sur la taille des vides détectés en fonction de la taille de son pas, un échantillonnage fournissant des garanties de détection intéressantes reste généralement trop coûteux à obtenir. Qu'elle soit stochastique ou régulière, la génération d'échantillons opère par réjection, en générant tous les échantillons potentiels dans la boîte englobante, et en ne conservant que ceux qui sont satisfaisants. Dans le cas de la double recherche d'intersection, le grossissement de la boîte englobante n'est pas impératif lors de la génération du dual, l'objet primal ne devant pas nécessairement être bordé par l'objet dual : la boîte englobante, également testée pour déterminer l'intersection marqueur de la sortie, assure la fermeture de l'espace bordant l'objet primal.

Bien évidemment, l'échantillonnage du dual doit être suffisamment dense au regard des paramètres de l'algorithme qui l'emploie, pour remplir efficacement sa mission de marquage du vide.

Normales

Comme évoqué précédemment, une direction est associée à chaque point des objets, primal ou dual. Cette direction est employée pour calculer la normale à la surface et éventuellement pour orienter les disques lorsque ces primitives sont utilisées. Toutefois, la donnée initiale de l'algorithme étant constituée d'un nuage de points seulement caractérisés par leurs positions, cette information doit être calculée. Bien qu'il soit possible de mettre en place des algorithmes plus complexes, un mode de calcul simple et local des normales est utilisé.

En un point x du nuage de points C , primal ou dual, la direction de la normale n est calculée à l'aide de la formule suivante :

$$n = \sum_{\substack{q \in C \\ |p-q| < r_C}} \frac{p-q}{|p-q|}$$

Sous réserve que le nuage C soit homogène, la direction calculée pointe vers l'extérieur de l'objet, puisque des points ne sont présents, approximativement, que dans une demi-sphère centrée autour de p . Cette situation est illustrée dans la figure 4.11.

Un résultat moins sensible au manque d'homogénéité des marqueurs peut être obtenu en s'appuyant sur les points du nuage dual \bar{C} :

$$n = \sum_{\substack{q \in \bar{C} \\ |p-q| < r_C}} -\frac{p-q}{|p-q|}$$

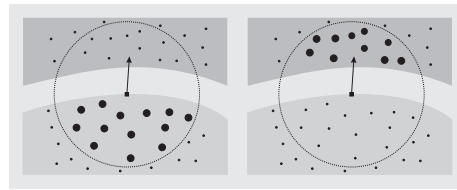


Figure 4.11: Calcul de la normale. Un voisinage sphérique est considéré autour de chaque point pour le calcul. À gauche sont employés les points primaux, à droite les points duaux. Ces derniers étant plus éloignés ils constituent un pôle d'attraction de la normale mieux défini.

La meilleure stabilité de ce mode de calcul est liée au plus grand éloignement des points q considérés. Étant situés plus loin, ils forment en effet un pôle d'attraction plus localisé dans l'espace, donc de meilleure qualité et moins sensible au manque d'homogénéité. Dans les deux cas, le rayon de calcul doit être choisi d'une taille suffisante pour prendre en compte un voisinage peuplé de points.

Si les normales sont jugées trop désorganisées à l'issue de leur calcul, on autorise également l'application d'un lissage spatial itératif. Dans ce cas, on recourt à un limiteur de variation pour éviter de dénaturer totalement le calcul initial par les applications successives du lissage.

Optimisation des nuages de points

De par sa conception même, l'algorithme de recherche d'intersection n'examine que les primitives proches des surfaces attendues des objets primal ou dual. Par conséquent, les points internes aux nuages n'interviennent en rien dans l'algorithme, et il est tout à fait possible de les éliminer sans modifier la surface générée. Cette élimination est intéressante dans la mesure où elle est à la fois simple et rapide à mettre en œuvre, et capable de réduire fortement le nombre total de points du nuage. Puisque ces points sont ignorés, l'optimisation ne porte alors pas sur la vitesse de recherche des intersections, mais sur la réduction de la mémoire pour représenter l'objet à visualiser.

On fonde alors l'élimination de l'intérieur du nuage de points primal sur un critère de proximité avec les points de l'objet dual. Un point primal n'est alors conservé que s'il est plus proche d'au moins un point dual qu'un seuil donné, c'est à dire qu'il est détecté proche du vide environnant, donc qu'il appartient à la surface, comme présenté dans la figure 4.13.

On notera bien que ce prétraitement d'amincissement du nuage primal ne permet pas d'extraire à proprement parler une surface, mais d'éliminer les points réellement internes jamais employés par l'algorithme d'intersection. Cet amincissement ne permet toutefois pas de se ramener au cas d'une surface échantillonnée par des points : la bande de points conservée garde une épaisseur non négligeable.

Bien qu'il soit possible de procéder symétriquement sur le nuage dual, après sa génération complète, il est plus efficace de générer directement un dual restreint à sa surface. Lors de la génération, un point échantillon n'est alors admis dans le dual que s'il remplit simultanément une condition d'éloignement du nuage primal — pour être représentatif du vide — et de proximité du nuage

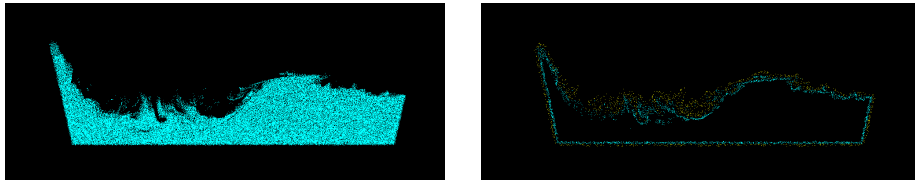


Figure 4.12: Exemple de résultat du prétraitement. À gauche, une coupe dans un nuage initial, à droite, les nuages primal et dual de surface qui en sont extraits.

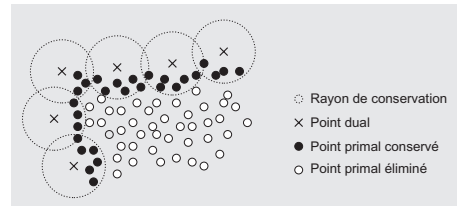


Figure 4.13: Extraction de la surface du nuage primal. L'extraction est réalisée en considérant un critère de proximité au nuage dual.

primal — pour être représentatif de la surface — On notera que la génération directe de ce dual mince peut alors être accélérée par cette limitation spatiale, en partitionnant l'espace au préalable et en examinant la distance minimale de chaque cellule au nuage primal, comme le présente la figure 4.14. Les cellules lointaines ne pouvant contenir de marqueurs de l'objet dual, elles sont rapidement ignorées, ce qui permet alors de concentrer la génération dans la zone de l'espace la plus pertinente. Un exemple de ces objets primal et dual fins est présenté dans la figure 4.12

On propose également un dernier traitement : densifier le nuage primal en y insérant de nouveaux échantillons, sur la base de la définition de son volume. L'intérêt de cette densification est d'autoriser une réduction de la taille des primitives pour mieux approximer la surface attendue. Ce dernier traitement est bien évidemment optionnel, mais il permet d'émuler un nuage plus dense. Si ce

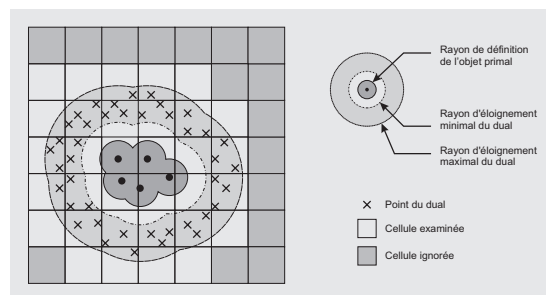


Figure 4.14: Génération du dual mince. L'espace correspondant à la surface du dual, défini par un intervalle de distance au nuage primal, est seul échantillonné. L'échantillonnage est rendu plus efficace en considérant un partitionnement qui permet d'ignorer l'espace trop éloigné du nuage primal pour contenir des points duaux.

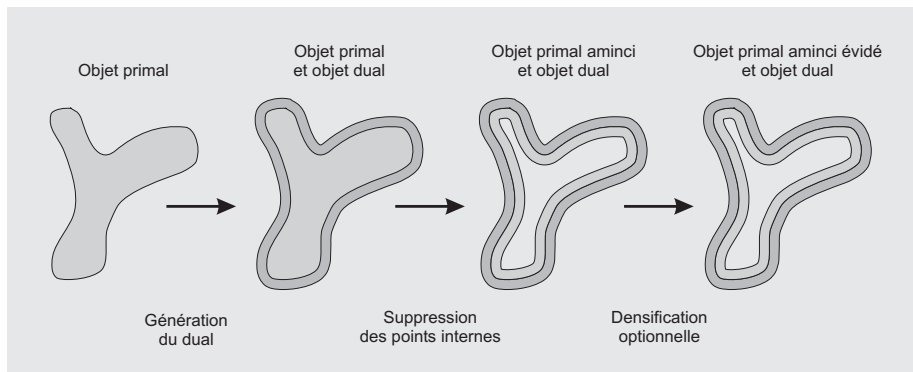


Figure 4.15: Étapes de prétraitement du nuage de points primal, hors calcul des normales et partitionnement. La dernière étape, de densification du nuage primal, est optionnelle et laissée à la discrétion de l'utilisateur, s'il juge le nuage initial trop lâche.

traitement est, dans son résultat, similaire à l'exécution d'une simulation avec un plus grand nombre de marqueurs, il reste possible de l'appliquer *a posteriori* de la simulation, sans nécessiter de nouvelle simulation, ce qui peut se révéler plus flexible. En outre, cette densification ne prend place qu'à la frontière du nuage primal, au contraire d'un accroissement uniforme de la densité de marqueurs dans la simulation, qui opère également dans le volume des nuages de points. En ce sens, ce post-traitement de densification fait meilleur usage du capital de marqueurs qui lui est alloué que le simulateur. Cependant, même limitée à la frontière, la génération reste coûteuse en raison des nombreux tests de proximité à effectuer pour déterminer si un point candidat peut être retenu ou non.

Les traitements appliqués pour un nuage primal comprennent donc deux phases : la génération du dual et l'optimisation du nuage primal, comprenant la suppression des points internes ainsi qu'une éventuelle densification du nuage. Ils sont résumés dans la figure 4.15.

4.1.5 Résultats

La figure 4.16 présente deux exemples d'animation rendues à l'aide de la méthode qui vient d'être présentée. Elle illustre bien que la méthode est tout à fait capable de produire des résultats satisfaisants tant que les marqueurs sont distribués d'une manière homogène, au début de l'animation. En revanche, dès lors que cette homogénéité est mise à mal, des trous peuvent apparaître dans la surface, comme on peut le voir dans la première animation. En effet, un seul jeu de paramètres est employé pour toute la durée de l'animation, ce qui rend la méthode de rendu difficile à appliquer sur l'intégralité d'une animation. Un fort bruit peut également se développer, comme à la surface du liquide, dans la deuxième animation. Dans ce cas, les normales perdent leur cohérence spatiale, et la surface n'est plus approximée de manière correcte. En outre, et bien que cela ne soit que peu visible sur une série d'images fixes, ce bruit est temporellement instable, ce qui le rend d'autant plus gênant. Pourtant, dans cette deuxième animation, 2 668 000 marqueurs sont exportés du simulateur, et les objets primal et dual sont composés d'environ 300 000 points chacun. À

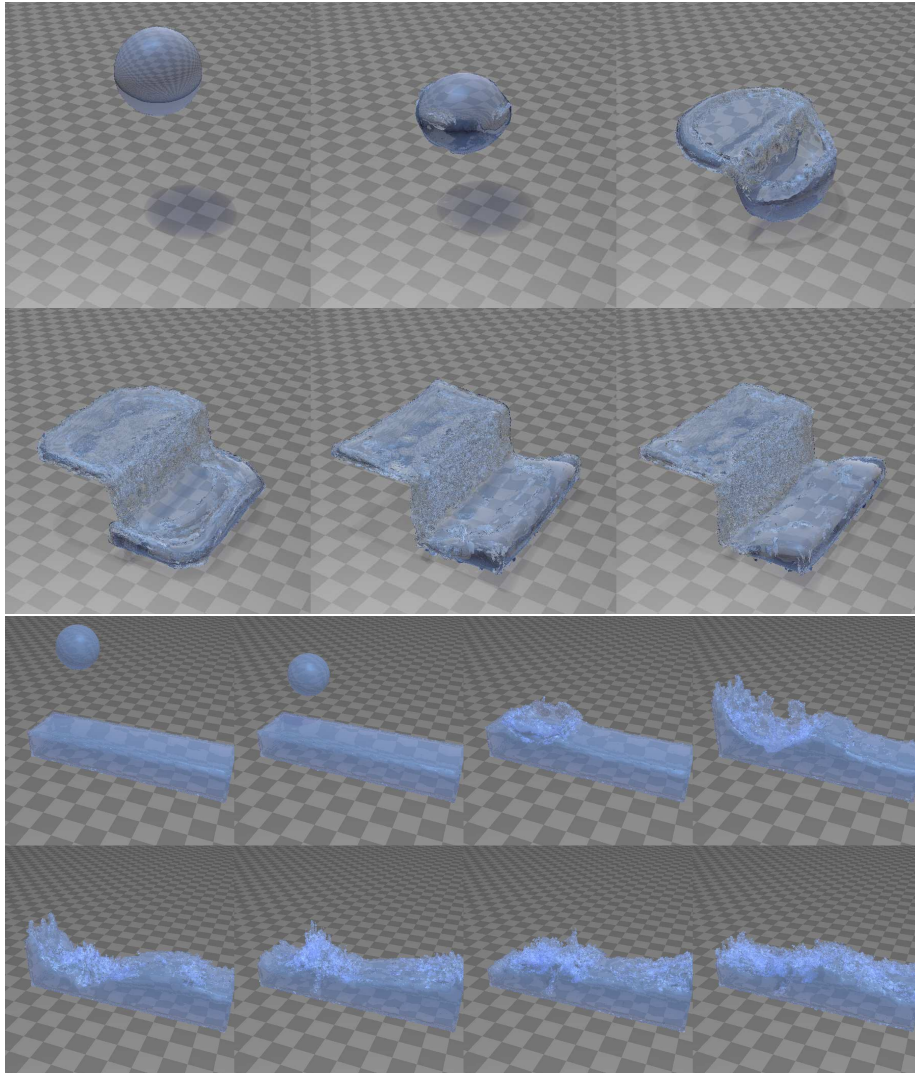


Figure 4.16: Exemples de résultats pour des animations réelles. Les surfaces visualisées présentent des défauts en raison du trop faible nombre de marqueurs se développant par endroits. Dans ces zones, les disques ne sont pas jointifs et approximativement mal la surface attendue.

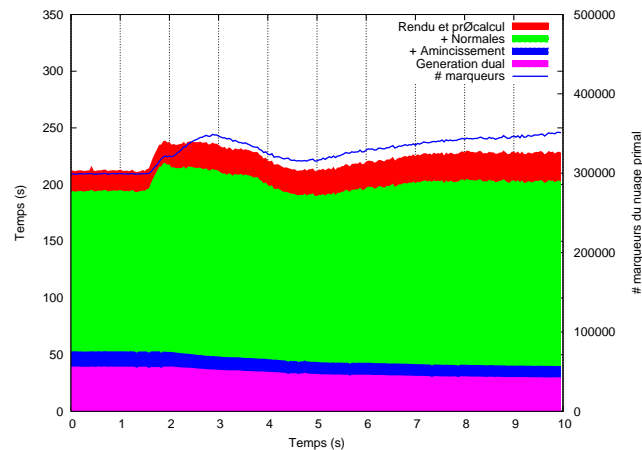


Figure 4.17: Temps de calcul pour la deuxième animation de la figure 4.16. Le nombre de marqueurs exportés de la simulation est constant, mais le fluide changeant de configuration, le nombre de marqueurs constitutifs de la surface primale varie légèrement. Le temps de calcul est largement dominé par le précalcul, et plus particulièrement par le lissage des normales, quatre passes de lissage étant effectuées.

titre de comparaison, une surface composée de 100 000 à 250 000 triangles peut être extraite des mêmes données par l'algorithme de *marching-cubes*.

Pour cette deuxième animation, on présente également les temps des calculs effectués, dans la figure 4.17. Ces temps correspondent à une animation de 240 images d'une résolution de 600×600 , calculée en approximativement 15 heures sur un Pentium 4 cadencé à 2.6GHz. On peut constater que le précalcul prédomine ici largement par rapport au temps de rendu lui-même, notamment en raison de l'importance du lissage des normales employé. Durant toutes les opérations de recherche de voisins, un partitionnement spatial par grille régulière est employé.

Influence de la taille des primitives

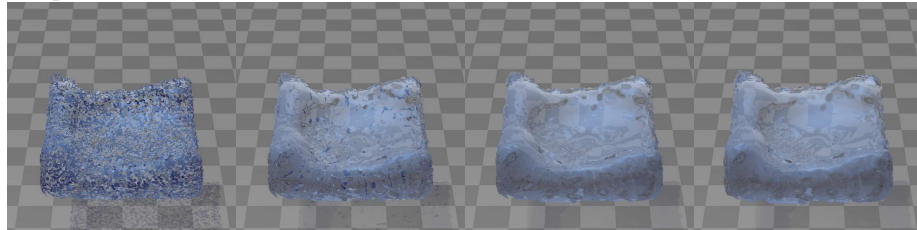
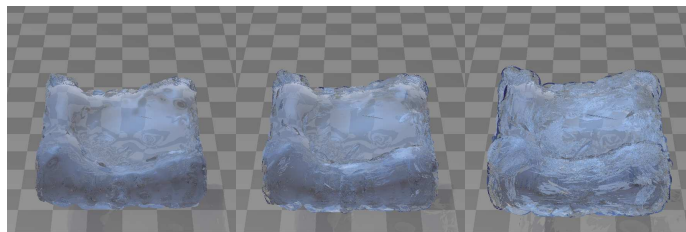
Que l'on emploie des disques ou des sphères, la taille des primitives joue un rôle déterminant dans la qualité du résultat. Logiquement, un seul intervalle de tailles est admissible pour un nuage donné, comme illustré par la figure 4.18. Si les primitives sont de trop petit rayon, le nuage apparaît discontinu. Si elles possèdent un rayon trop important, la surface se retrouve au mieux empâtée et ne correspond plus réellement à l'image attendue, ou pire encore, se trouve complètement dénaturée dans le cas des disques.

Notons qu'à rayon égal, les sphères fournissent une meilleure couverture de la surface, celle-ci présentant alors moins de trous. Cependant, la surface est plus difficilement approximée, et l'objet est sensiblement grossi par rapport à celui obtenu par l'usage de disques.

4.1.6 Bilan

Si l'idée d'employer directement les marqueurs comme éléments de visualisation est séduisante, elle se révèle très délicate à mettre en œuvre dans le

Disques

 $r = 0.01$ $r = 0.02$ $r = 0.03$ $r = 0.04$  $r = 0.05$ $r = 0.07$ $r = 0.10$

Sphères

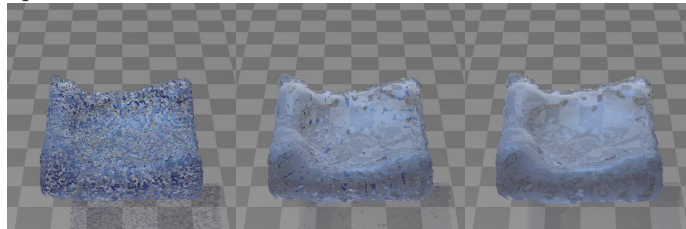
 $r = 0.01$ $r = 0.02$ $r = 0.03$

Figure 4.18: Influence de la taille des primitives. Seule une plage de tailles, dépendante de la géométrie du nuage, autorise un résultat visuellement satisfaisant. En deçà, la surface est fragmentée, au-delà, elle se trouve dénaturée.

contexte du raytracing, comme en attestent les résultats présentés. Si le liquide peut être visualisé, la méthode est loin d'être assez robuste pour présenter des images d'une qualité photo-réaliste.

La difficulté que rencontre la méthode est double. La première difficulté découle du type même des données employées : dans la mesure où les disques fournissent une représentation très floue et non structurée de la surface, il est difficile, voire impossible, de parvenir à une visualisation de la surface précise et dénuée de parasites. Ce défaut est amplifié par le manque d'homogénéité des nuages de points, qui est de plus variable dans le temps, puisque provenant de données réelles d'une simulation.

La seconde difficulté est liée au calcul des normales, et donc à l'orientation des disques. Le mode de calcul employé n'est de loin pas assez performant, et ne garantit aucune cohérence temporelle. Dès lors que la surface du liquide devient tourmentée, la normale n'est plus du tout approximée correctement, qu'une phase de lissage soit appliquée ou pas. Les disques se recouvrent alors d'une manière anarchique sans plus matérialiser aucune surface fermée. Ces zones présentent alors des résultats très parasités, voire une représentation complètement altérée comme on a pu le montrer dans les résultats.

Bien évidemment, ces problèmes s'estompent quand les objets sont représentés avec une densité de points plus importante, puisque la surface se trouve naturellement mieux approximée. Comme on peut le voir au début des animations, quand les points fournissent de meilleures approximations, la méthode peut produire des résultats satisfaisants. Cependant, les nuages de points étant tridimensionnels et non bidimensionnels, et provenant d'une simulation et non d'une acquisition, une augmentation sensible de la densité de points constitue difficilement une option viable. En effet, augmenter le nombre de points signifie augmenter d'une manière très significative le temps de simulation. En outre, et même si cette augmentation était réalisée, il serait difficile de maintenir une bonne homogénéité des marqueurs au cours du temps sans introduire de contrôle plus poussé dans le simulateur. Il faut également rappeler que les temps, déjà longs, de précalcul au rendu, seraient encore accrus. En outre, l'emploi d'une représentation plus détaillée ne résout pas fondamentalement les problèmes liés à l'échantillonnage ponctuel du fluide, mais les repousse. Le problème sous-jacent de la reconstruction robuste d'une information continue et satisfaisante à partir d'échantillons discrets n'est ainsi pas mieux résolu.

Par opposition à ces défauts, on peut tout de même remarquer que cette visualisation directe des marqueurs autorise une visualisation sans omission des données issues du simulateur. Contrairement à une démarche d'extraction de surface définie à partir des marqueurs, pour laquelle des parties minces de liquide peuvent être perdues, tout marqueur exporté du simulateur est visualisé, sinon avec une qualité parfaite, du moins avec une matérialisation effective.

Si l'on fait abstraction de ces problèmes liés aux données, on peut dresser un bilan propre à l'algorithme de recherche d'intersection lui-même : cet algorithme permet de visualiser des nuages volumiques de points d'une manière relativement rapide, en se concentrant sur la surface des objets, grâce à la notion d'objet dual introduite qui permet de supprimer toute évaluation réelle du volume. Il pourrait ainsi se révéler intéressant d'étudier l'application du même principe de matérialisation du vide environnant aux surfaces implicites, pour en accélérer la visualisation. Il faut cependant souligner que cette relative rapidité de rendu est gagnée au prix d'un précalcul important.

En conclusion, l'algorithme, s'il est fonctionnel, ne présente pas une robustesse suffisante pour traiter les données issues de simulations réelles. Au vu des résultats obtenus, on peut également remettre en cause la pertinence, pour la visualisation, d'un emploi direct — sans aucune forme de reconstruction de surface — des marqueurs du liquide. En effet, en se rappelant que l'usage initial de ces marqueurs est de déterminer le statut des cellules de la grille de simulation, on peut estimer qu'ils n'engendrent de représentation du liquide et de sa surface qu'à une échelle proche de celle de la grille de simulation elle-même. La résolution effective de la représentation est alors insuffisante pour produire une visualisation de qualité. Le grand nombre de marqueurs des nuages, découlant du caractère volumique des nuages, est trompeur sur ce point : seule une fraction des marqueurs est réellement utile à la définition d'une surface.

4.2 Visualisation rapide assistée par le matériel

Si l'objectif de la visualisation photo-réaliste finale est évident, l'objectif de la visualisation de prototypage reste à préciser. Le rôle qu'on lui assigne est la présentation d'un résultat visuel suffisamment significatif pour pouvoir juger de son adéquation avec un objectif à atteindre, au coût le plus faible possible. Il s'agit en somme d'un compromis entre un rendu strictement interactif et un rendu réaliste sans contrainte de temps. Ce type de méthode trouve sa place dans l'étape de mise au point des animations : comme on l'a déjà évoqué dans le bilan du mode d'interaction entre liquide et solides, la conception des animations est très itérative et il n'est pas envisageable d'introduire de longs délais de rendu pour juger de la qualité de chaque essai de simulation. Ces prévisualisations intermédiaires se doivent donc d'être rapides, tout en restant significatives.

La méthode proposée dans ce cadre ne prétend pas constituer une avancée dans le domaine du rendu, mais plutôt une réponse circonstanciée au problème de la visualisation rapide de simulations MAC. En outre, cette méthode a été conçue alors que la programmabilité du matériel graphique, sur laquelle la méthode s'appuie, était plus que limitée, le cadre de ce travail étant constitué du *shader model 1.4*.

4.2.1 Méthode de rendu

Lorsqu'un liquide à surface libre est observé, on peut noter deux classes d'effets visuels : celle propre à la surface du liquide, comme son illumination, et celle liée à son volume, comme l'atténuation de la lumière traversant le volume.

Afin d'obtenir un résultat utile pour évaluer la qualité d'une animation, on postule que les deux classes d'effets doivent être figurées avec un minimum de vraisemblance. On s'attache en particulier à vouloir représenter des effets de transparence qui incluent une atténuation variable de la lumière traversant le milieu liquide. Combinée avec un modèle d'illumination locale de la surface, une assez bonne appréciation du volume du liquide et de ses mouvements peut être obtenue, comme l'illustre la figure 4.19. Dans l'optique de visualisation de simulations MAC avec des interactions, il est également intéressant de pouvoir figurer correctement des scènes avec des objets autres que le liquide.

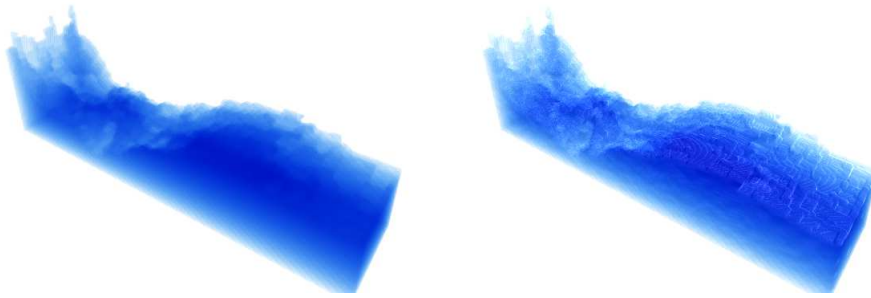


Figure 4.19: Illustration des composantes du rendu rapide. Un modèle d'illumination locale, même simple, appliqué à la surface du liquide contribue à une meilleure perception des mouvements du liquide.

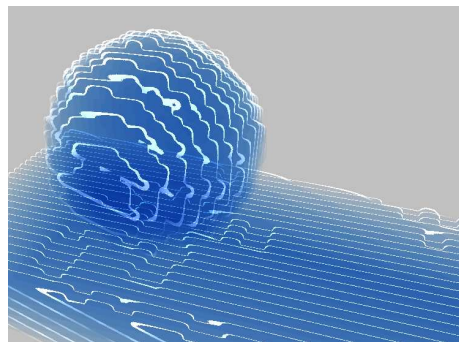


Figure 4.20: Illustration du rendu volumique par coupe. Un nombre volontairement faible de coupes est employé pour distinguer chaque coupe individuellement. L'illumination locale de la surface, ainsi que l'atténuation de la lumière en fonction de la distance parcourue dans le liquide, sont visibles.

Approche

Une approche possible consiste bien évidemment à reconstruire une surface à partir des marqueurs exportés du simulateur, comme on a pu le décrire précédemment. Si cette opération peut être réalisée dans des délais tout à fait acceptables, la visualisation de la surface extraite reste problématique par rapport au critère d'atténuation de la lumière traversant le volume.

C'est pourquoi on adopte une approche apparentée au rendu volumique, qui permet de gérer avec une grande simplicité et un bon réalisme l'atténuation de la lumière traversant le liquide. En outre, la topologie du liquide peut alors être complètement ignorée dans le rendu, ce qui constitue un deuxième avantage. La méthode que nous proposons pour offrir une prévisualisation rapide consiste donc en une méthode de rendu volumique par coupes, accélérée par le matériel, incluant un modèle d'illumination locale simple à proximité de la surface, comme l'illustre la figure 4.20.

On rappelle que le principe d'un rendu volumique par coupes [14] consiste à afficher une série de coupes prises dans un volume de données, de l'arrière-plan vers l'avant-plan, en composant les résultats à l'image par transparence.

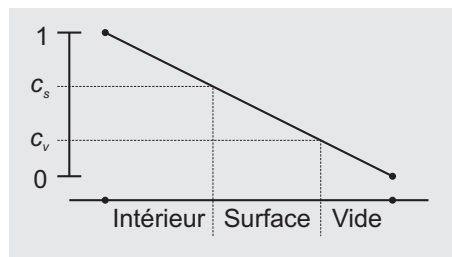


Figure 4.21: Définition des régions de rendu. Les régions sont définies par des intervalles, pour la valeur interpolée de présence de liquide.

En dérivant des données l'opacité et la couleur des pixels de chaque coupe, il est possible de reconstituer une image et de simuler une surface sans avoir à se préoccuper de la topologie réelle des structures incluses dans les données. Ce procédé s'apparente à l'évaluation simultanée, par échantillonnage, de l'absorption de la lumière par les données, le long de tous les rayons composant l'image.

Données du rendu

Pour effectuer le rendu souhaité, il est nécessaire de déterminer si un point de l'espace appartient au liquide ou non. S'il se trouve proche de la surface, la normale à celle-ci est également requise dans le calcul d'illumination. Ces deux informations, de normale et de présence du liquide, sont précalculées et stockées sous la forme d'une texture 3D à quatre composantes définie sur tout l'espace des données. Les trois composantes de la couleur représentent la normale, la composante alpha représente la valeur de présence du liquide. Cette texture tridimensionnelle est échantillonnée en considérant une interpolation trilineaire durant le rendu.

Le calcul de la présence du liquide reprend la caractérisation déjà utilisée par le simulateur : une cellule de la grille définie par la texture est considérée comme pleine de liquide si au moins un marqueur y est présent. Les autres cellules représentent alors le vide environnant. Cette information binaire de présence est ensuite codée comme les deux valeurs extrêmes représentables par la composante de transparence de la texture. Interpolée dans l'espace, cette valeur de présence autorise alors la discrimination des trois régions d'intérêt de l'espace — l'intérieur, la surface, et l'extérieur du liquide — par comparaison avec deux seuils, c_s et c_v , comme l'illustre la figure 4.21.

La normale à la surface, nécessaire à l'évaluation d'un modèle d'illumination, est calculée en considérant le gradient de la densité de marqueurs, éventuellement lissé spatialement.

Algorithme de rendu

Comme on l'a déjà mentionné, le rendu se fait par composition de coupes dont les valeurs d'opacité et de couleurs sont calculées à partir de la texture décrite précédemment. Les différentes coupes dans le volume de données sont réalisées orthogonalement à la ligne de vue centrale de l'observateur. L'accès aux

données pour les pixels de la coupe est réalisé de manière implicite, en raison de leur stockage dans une texture volumique.

Avant de décrire le traitement réalisé pour chaque pixel de chaque coupe, on rappelle qu'il est nécessaire d'en composer le résultat avec les rendus précédents déjà stockés dans l'image. Cette étape est prise en charge matériellement par l'étape d'*alpha-blending* du *pipeline* graphique. Dans le cas étudié ici, selon la région de la coupe, intérieur du liquide ou surface, un filtrage simulant l'atténuation — un comportement multiplicatif — ou une éventuelle adjonction de lumière — un comportement additif — doit être appliqué.

Ce comportement est obtenu par une composition additive dont les paramètres, notés $[\cdot]$ sont choisis pour réaliser l'opération suivante :

$$res = c_{src} \cdot [a_{src}] + c_{dst} \cdot [c_{src}]$$

où les variables c_{src} et c_{dst} représentent respectivement la couleur calculée dans la coupe, et la couleur déjà présente à l'image, et a_{src} la composante de transparence calculée dans la coupe.

Associé à ce mode de composition, le *pixel shader* exécuté pour calculer la coupe produit une sortie $\langle c_{src}, a_{src} \rangle$ conditionnelle. Dans le cas d'un point intérieur au volume de liquide, la sortie prend la valeur $\langle c_{liquid}, 0 \rangle$ qui correspond au facteur d'atténuation de la lumière traversant le liquide, *i.e.* sa couleur. Une atténuation pure est donc effectuée :

$$res = c_{liquid} \cdot 0 + c_{dst} \cdot c_{liquid} = c_{dst} \cdot c_{liquid}$$

Si l'on considère un point en surface du liquide, la sortie prend la valeur $\langle 1, l \rangle$ où l représente l'illumination locale calculée, ce qui additionne l'illumination locale à la couleur déjà présente dans l'image :

$$res = 1 \cdot l + c_{dst} \cdot 1 = c_{dst} + l$$

Cette approche engendre cependant une restriction : seule une illumination blanche peut être considérée, puisqu'elle n'est décrite que par le seul scalaire l , passé du *pixel shader* à l'unité de composition dans la composante de transparence. Le modèle d'illumination utilisé est choisi purement diffus, ce choix étant dicté par les limitations en nombre d'instructions de l'architecture-cadre adoptée. On rappelle que la normale est interpolée linéairement à partir des données présentes dans la texture 3D.

En plus de fournir les informations d'atténuation et d'évaluer l'illumination locale, le *pixel shader* contrôle l'affichage effectif du point de la coupe : si le point considéré appartient au vide environnant le liquide, l'affichage du pixel de la coupe est annulé.

Fonctionnant en tandem avec ce *pixel shader*, un *vertex shader* calcule les directions de l'observateur et de la source lumineuse. Ces directions sont alors interpolées automatiquement entre les sommets des coupes pour être fournies au *pixel shader*. Face au nombre restreint de calculs réalisables avec l'architecture-cadre, seules des interpolations des directions entre les sommets peuvent être réalisées, et il n'est pas possible de calculer ces directions à chaque pixel, ou même de les renormaliser après interpolation. Ceci impose donc que la source lumineuse et l'observateur soient suffisamment éloignés de l'objet à visualiser pour que cette approximation reste valide.

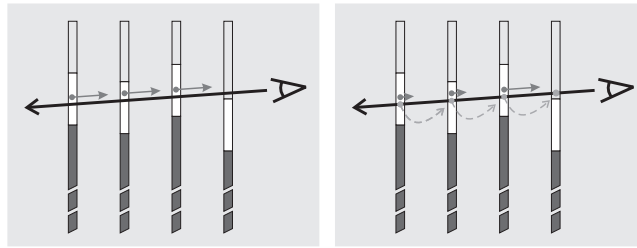


Figure 4.22: Atténuation de l'illumination locale. Un facteur d'atténuation pondérant l'illumination locale est calculé en fonction du point courant, ainsi que du point se projetant sur le même pixel, mais appartenant à la coupe suivante.

Améliorations

Tel que décrit ci-dessus, l'algorithme souffre de deux limitations. La première est liée à la précision des données contenues dans l'image, stockées sur 8 bits uniquement. Cette quantification impose qu'entre deux coupes successives, l'intensité de la couleur doit varier d'au moins une unité pour produire une atténuation effective : en deçà, l'atténuation devient nulle et l'objet complètement transparent. Dans le cas de plus de 256 coupes, une composante de couleur se trouve donc complètement absorbée, même avec l'atténuation la plus faible possible.

Une solution simple existe pour résoudre ce problème, en concentrant l'atténuation sur un sous-ensemble des coupes. L'atténuation due à plusieurs coupes est alors reportée et évaluée sur une coupe unique. Cette réduction du nombre de coupes porteuses de l'atténuation permet alors de masquer le problème de quantification, mais au prix d'une baisse de la qualité de représentation du liquide sur ses bords. Le modèle d'illumination locale reste, lui, évalué pour chacune des coupes.

La seconde limitation concerne la prise en compte de l'illumination locale. Dans la mesure où elle est calculée sur une bordure du volume d'épaisseur finie, un pixel peut recevoir des contributions provenant de plusieurs coupes, comme le présente la figure 4.22. Dans ce cas, les contributions successives peuvent s'accumuler de manière excessive si la surface est regardée sous un angle rasant. Pour combattre ce phénomène, une pondération de l'illumination locale est employée. Elle est calculée en considérant les deux valeurs de présence du liquide, au point courant, et au point de la coupe suivante correspondant au pixel en train d'être calculé. Elle est définie comme la différence des deux valeurs de présence, restreinte à l'intervalle $[0, 1]$. Ainsi, pour un angle rasant, les deux valeurs sont similaires, et la différence, proche de 0, produit une atténuation importante. En raison de l'architecture-cible et de ses contraintes en nombre d'instructions, il n'est pas possible d'employer un modèle de détection de surface plus évolué.

Une dernière amélioration mineure est apportée à l'ergonomie de la visualisation, en modulant le nombre de coupes utilisées pour le rendu. Ainsi, quand l'animateur décide de déplacer le point de vue pour examiner l'animation sous un autre angle, il est possible de rendre cette manipulation interactive, en n'affichant que peu de coupes. Si cette réduction de précision s'accompagne d'une sévère perte de qualité, elle autorise un déplacement fluide et un examen de l'animation bien plus agréable qu'avec un déplacement non-interactif, ou qu'avec une

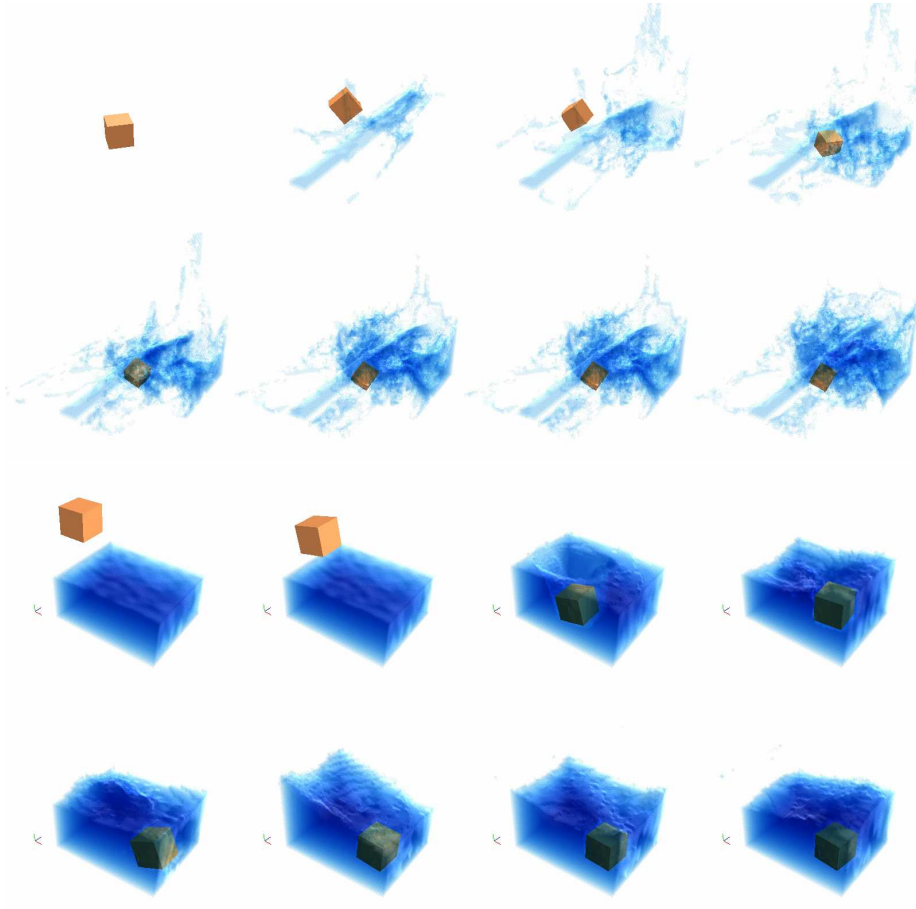


Figure 4.23: Exemples de visualisation volumique. La première animation correspond à celle déjà présentée sous un autre angle dans la figure 3.26.

totale omission des données pendant le déplacement.

4.2.2 Résultats

La figure 4.23 illustre le type de résultats qu'il est possible d'obtenir avec cette méthode de rendu. Dans cette figure, le premier résultat correspond à l'animation présentée avec un rendu photo-réaliste dans la figure 3.26, mais vue sous un angle différent. S'il n'est bien évidemment pas question de comparer directement les images générées, la visualisation volumique reste suffisamment fidèle pour que l'on puisse se rendre compte des mouvements du fluide, et juger de la qualité de l'animation.

La figure 4.24 illustre pour sa part le bénéfice qu'il est possible de retirer de cette forme de visualisation volumique, pour juger d'une animation. Comparativement à un rendu surfacique sans transparence, l'éclairage de la surface ainsi que la prise en compte de l'atténuation volumique permettent de mieux appréhender les mouvements internes au fluide. On voit ainsi nettement la ca-

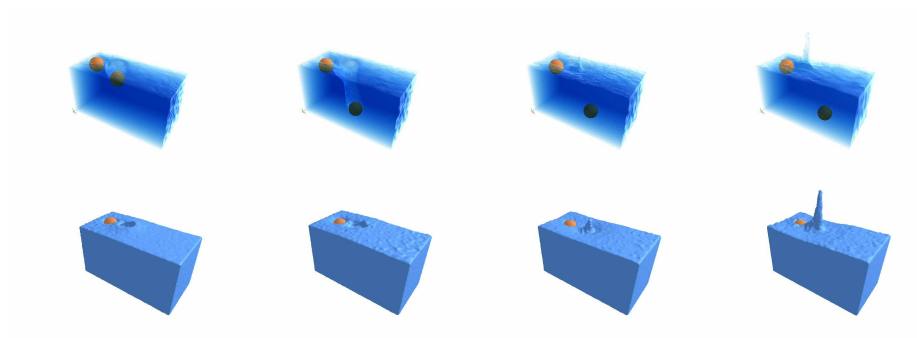


Figure 4.24: Comparaison du rendu volumique au rendu surfacique. Le rendu volumique permet de mieux appréhender les mouvements de l'intérieur du liquide.

tivité engendrée par la balle dans le liquide, l'entrée étant seule visible selon le point de vue adopté, si seule la surface est visualisée. On notera également qu'à condition de rendre à l'image le liquide après les autres objets présents dans la scène, l'atténuation des objets par le liquide est correctement prise en compte.

La figure 4.25 illustre une autre animation, ainsi que le temps nécessaire à son rendu. L'animation comporte 240 images, chacune comptant 2 668 000 points, ce qui représente des données d'un volume de 7.2 GB réparties en 240 fichiers de 30.5 MB. La texture 3D employée a pour résolution $128 \times 64 \times 32$, et une passe de lissage spatial est employée dans le calcul des normales. La taille de grille choisie ne fait pas apparaître de trous parasites dans le liquide, au contraire de la grille de résolution double. Le rendu est effectué à l'aide de 1024 coupes pour une image d'une taille finale de 800×600 pixels approximativement. Le matériel sur le quel ont été réalisées les mesures est composé d'un Pentium 4 cadencé à 2.6 GHz, et d'une Radeon 9800 Pro.

Il est aisé de constater qu'une moitié du temps — 56% en moyenne — est consacrée au chargement des données en mémoire. Le prétraitement et le rendu se partagent le reste du temps à parts à peu près égales, occupant en moyenne 18% et 24% du temps total. Le calcul de la texture inclut la discrimination entre le liquide et son environnement par comptage des points dans la grille ainsi que le calcul de la normale et son lissage. Le rendu inclut la génération de l'image et sa sauvegarde.

Excepté quelques aléas de mesure, le temps de calcul reste logiquement stable au cours de l'animation, le rendu étant réellement indépendant des données à visualiser. Au total, la prévisualisation complète de l'animation est effectuée en 5 minutes 24 secondes. Ce temps est tout à fait comparable à celui de la simulation elle-même, qui est de 5 minutes 49 secondes.

Dans cette figure, il est également possible de noter quelques effets de voxelisation du liquide, découlant de la résolution de la texture 3D utilisée comme support de l'algorithme. Bien que problématiques dans l'objectif d'une visualisation photo-réaliste, ces légers défauts ne posent pas problème pour évaluer les mouvements du liquide.

Comme il est prévu, et comme le confirme la figure 4.26, le temps requis par le rendu lui-même est linéaire en nombre de coupes, ainsi qu'en nombre de pixels à afficher.

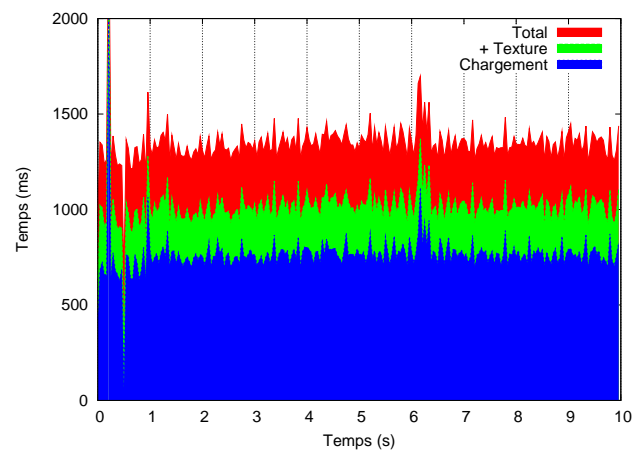
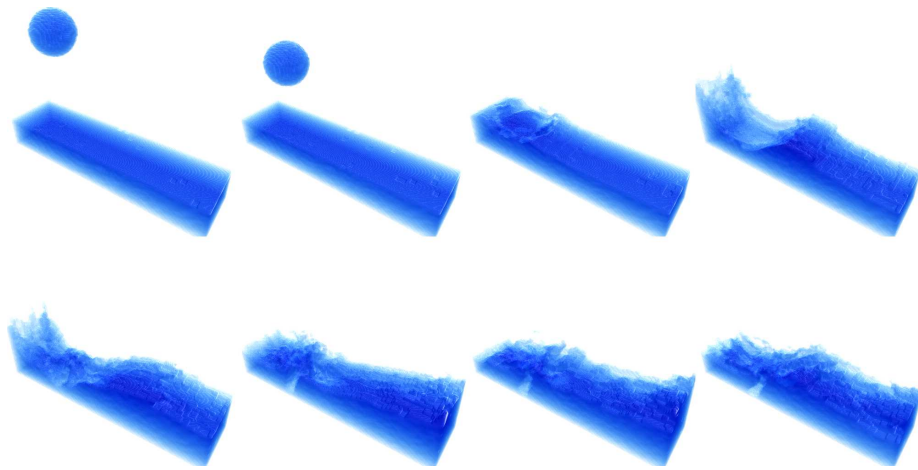


Figure 4.25: Temps de calcul du rendu volumique.

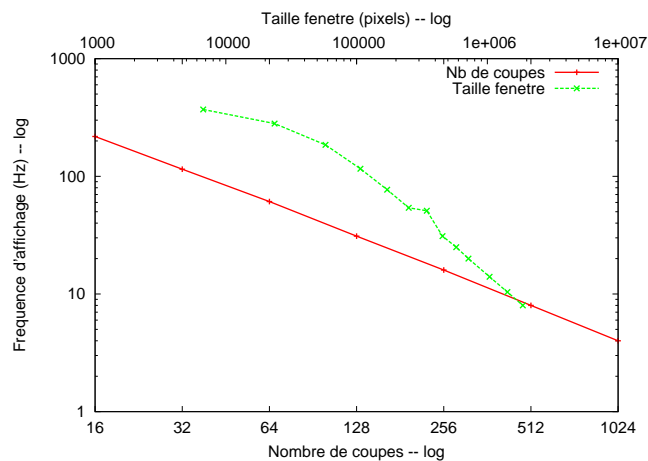


Figure 4.26: Vitesse de rendu. Influence du nombre de coupes, et du nombre de pixels à afficher. Le taux de couverture de l'image, par la boîte englobante de l'objet volumique à afficher est de 58%.

4.2.3 Bilan

La méthode que nous avons proposée satisfait, au vu de ses résultats, aux objectifs qui ont guidé sa conception, à savoir une rapidité suffisante pour ne pas interrompre la conception de l'animation par de longues phases de rendu, et un résultat assez significatif pour pouvoir juger de l'animation. Elle ne prétend cependant pas se targuer d'une qualité de rendu suffisante pour être appliquée hors du champ du prototypage des animations, ni proposer autre chose qu'une solution de visualisation adaptée aux données issues de simulations MAC.

On note que la qualité des résultats pourrait être considérablement améliorée en adoptant pour base une technique de rendu volumique beaucoup plus performante et précise fondée sur la pré-intégration [35], et en mettant à profit la programmabilité sans cesse améliorée du matériel graphique.

Par ailleurs, les performances de la méthode pourraient certainement être plus élevées si l'on intégrait davantage la simulation et la visualisation, ce qui éviterait les entrées-sorties coûteuses. Toutefois, cette intégration de la simulation et de la visualisation se ferait au détriment de la flexibilité du processus de mise au point des animations.

Chapitre 5

Simulation par éléments finis

Ce chapitre présente les travaux que nous avons réalisés pour proposer une simulation de liquides fondée sur la méthode des éléments finis. Si cette méthode a déjà été employée dans le cadre de la simulation de solides à des fins d'animation en informatique graphique [116], elle ne l'a pas été, à notre connaissance, dans le cadre de la simulation de fluides. Le principal avantage qu'offre cette méthode, est sa capacité à traiter naturellement des domaines de simulation de géométries arbitraires, pour envisager à long terme un meilleur traitement des interactions d'un fluide avec des solides déformables. D'autres perspectives existent également, comme l'obtention d'une représentation explicite de la surface dans un but de visualisation, ou la suppression des contraintes liées à une discrétisation uniforme de l'espace.

Les avantages de cette méthode ayant pour revers un coût de calcul bien plus important que celui des méthodes employant des grilles régulières, et un abord plus ardu que celui d'une discrétisation par différences finies, les travaux présentés dans ce chapitre ne sont pas encore en mesure de produire des images visuellement intéressantes du point de vue de l'infographie, et encore moins des animations.

Après un rapide aperçu de la méthode des éléments finis, présenté dans la section 5.1, les sections suivantes détailleront la méthode de résolution des équations du fluide, jusqu'à présenter les premiers résultats validant le simulateur implémenté, dans la section 5.6.

Au cours de la description de l'application de la méthode des éléments finis aux équations de Navier-Stokes, on s'est attaché à fournir une présentation détaillée des différentes étapes par lesquelles elle chemine, afin de rendre compréhensible une démarche moins courante que la méthode des différences finies. On prend notamment le parti de ne pas éluder les développements intermédiaires des calculs, afin de conserver un caractère progressif à la présentation. Il ne s'agit pas ici de proposer un exposé exhaustif des méthodes par éléments finis, et encore moins de la théorie sur laquelle elles sont fondées. Il est clair que l'on n'évoquera pas les aspects mathématiques fondamentaux d'existence ou d'unicité des solutions, inhérents à la méthode et à son application aux équations de Navier-Stokes. Pour une introduction plus détaillée de la méthode des éléments finis, on invite le lecteur à se reporter à des ouvrages de référence, par exemple [128].

5.1 Principe de la simulation par éléments finis

La méthode par éléments finis est une technique permettant de calculer une solution approchée à une équation aux dérivées partielles. Cette méthode n'est pas une procédure de résolution en elle-même, mais une méthode de discrétisation des équations, c'est-à-dire le moyen de franchir le pas séparant l'équation aux dérivées partielles, continue, d'un système d'équations, discret, qu'il est ultérieurement possible de résoudre numériquement à l'aide d'une méthode adaptée.

Démarche de la méthode Très schématiquement, pour générer ce système, la méthode par éléments finis procède en requérant que l'approximation recherchée de l'équation aux dérivées partielles satisfasse l'équation en un sens intégral, à la manière des méthodes variationnelles. Une équation différentielle

$$l(u) - f = 0 \quad \text{dans } \Omega$$

se trouve donc traduite par le problème :

$$\begin{aligned} &\text{Trouver } u \text{ tel que} \\ &\int_{\Omega} [l(u) - f] \cdot v \, dx = 0 \quad \forall v \in V \end{aligned}$$

ce qui équivaut à demander que l'erreur commise par l'approximation, $l(u) - f$, soit orthogonale à un espace-test V considéré¹, et donc que l'approximation soit optimale par rapport à cet espace.

La méthode considère ensuite un espace test $V_h \subset V$ de dimension finie pour extraire de cette relation un système de $\dim(V_h)$ équations. Parallèlement, la solution est, elle aussi, approximée et recherchée dans un deuxième espace, U_h , également de dimension finie. En décomposant l'approximation recherchée, u , sur une base $(\phi_i)_i$ de l'espace U_h ,

$$u = \sum_i u_i \phi_i$$

dans chacune des équations précédemment extraite, un système algébrique mettant en jeu les inconnues u_i est construit. Si l'équation aux dérivées partielles est linéaire, un système linéaire en découle également :

$$[L] \cdot [u] = [F]$$

qui doit ensuite être résolu numériquement pour conduire à l'approximation recherchée. Dans l'approche que l'on emploie, celle de Galerkin, l'espace V_h est choisi identique à U_h .

C'est cette procédure qui assure la formation d'un système d'équations de taille nécessaire et suffisante pour pouvoir déterminer l'approximation de la solution de manière unique. La force de l'approche par éléments finis est de lier l'espace d'approximation de la solution, U_h , à un maillage du domaine de simulation, ce qui permet de tenir compte de sa forme ainsi que des conditions aux limites à considérer, points éludés dans cette présentation rapide.

¹Pour le produit scalaire $\langle u, v \rangle = \int_{\Omega} u \cdot v \, dx$ de $L^2(\Omega)$

Maillage de simulation Au contraire des grilles régulières employées par les méthodes de différences finies, la topologie du maillage utilisé par la méthode des éléments finis n'est que très peu contrainte : le maillage doit essentiellement former une partition du domaine et être géométriquement conforme. Hors de ces restrictions, le maillage utilisé peut être arbitraire, inclure divers types de cellules y compris des simplexes ou des éléments courbes, ce qui permet d'approximer le domaine de simulation, avec une aisance beaucoup plus grande que dans le cas d'une grille régulière. Par conséquent la solution recherchée peut être approximée avec une grande précision.

On ne tirera toutefois pas, ici, parti de toutes ces possibilités, et l'on se contentera dans ce travail d'utiliser des éléments simples et non courbes, qui se révèlent suffisants pour l'usage que l'on en requiert. On tirera cependant parti de la possibilité de mailler des domaines polyédriques de formes arbitraires. Par contre, la problématique de création du maillage, extrêmement vaste en elle-même, n'est pas du tout abordée ici, et le maillage de simulation est supposé faire partie des données du problème.

Application aux équations de Navier-Stokes De cette approche émerge un mode opératoire que l'on peut résumer ainsi :

1. Traduction du problème fort en sa forme variationnelle
2. Choix d'un type d'élément – Maillage du domaine de simulation avec ce type d'élément
3. Calcul de la discrétisation
4. Résolution du système d'équations résultant

L'application de cette méthode à la résolution des équations de Navier-Stokes est décrite dans les sections suivantes. Toutefois, le problème qui sera effectivement résolu ne sera pas celui de Navier-Stokes, qui est non-linéaire et dépendant du temps, mais celui de Stokes, qui est linéaire et stationnaire. La section 5.2 montrera le type de discrétisation temporelle adopté, qui permet de réaliser cette transformation. Sur la base du problème de Stokes, la section 5.3 établira la formulation variationnelle des équations considérées. Ensuite, la section 5.4 présentera les espaces d'approximations considérés pour la solution, découlant du type des éléments finis choisis, en relation avec le maillage de simulation, et les discrétisations des équations en découlant, pour aboutir au problème discret. La résolution numérique du problème linéaire discret produit est discutée dans la section 5.5.

5.2 Discrétisation en temps

Pour que la discrétisation produite par la méthode des éléments finis engendre un système d'équations linéaires, l'équation aux dérivées partielles traitée doit être également linéaire, ce qui n'est pas le cas des équations de Navier-Stokes. Celles-ci sont en outre dépendantes du temps, ce qui signifie qu'il est nécessaire de considérer une discrétisation en temps. Cette section présente deux méthodes pour opérer une discrétisation en temps, qui permettent toutes deux de ramener la résolution du problème des équations de Navier-Stokes à celle d'un problème de Stokes généralisé.

5.2.1 Méthode des caractéristiques

L'idée sous-jacente à la méthode des caractéristiques a déjà été présentée dans la section 3.1.5, dans le contexte des simulations MAC. On rappelle brièvement que cette technique s'appuie sur le fait que la dérivée totale apparaît dans les équations de Navier-Stokes :

$$\begin{cases} \frac{Du}{Dt} + \nu \Delta u - \nabla p = f \\ \nabla \cdot u = 0 \end{cases}$$

et qu'il est possible d'approximer ce terme en considérant la trajectoire du fluide, *i.e.* les caractéristiques χ du champ de vitesse sur un pas de temps $[t, t + 1[$:

$$\frac{Du}{Dt}(x) \approx \frac{u^{t+1}(x) - u^t \circ \chi^t(x)}{\delta t}$$

De cette approximation découle alors le problème suivant, pour calculer le pas de temps $t + 1$:

$$\begin{cases} \frac{u^{t+1} - u^t \circ \chi^t}{\delta t} + \nu \Delta u^{t+1} - \nabla p^{t+1} = f^{t+1} \\ \nabla \cdot u^{t+1} = 0 \end{cases}$$

Le problème peut alors se réécrire sous la forme :

$$\begin{cases} \frac{1}{\delta t} u^{t+1} + \nu \Delta u^{t+1} - \nabla p^{t+1} = f^{t+1} + \frac{1}{\delta t} [u^t \circ \chi^t(x)] \\ \nabla \cdot u^{t+1} = 0 \end{cases}$$

Avec cette méthode, la simulation d'un pas de temps des équations de Navier-Stokes s'apparente à la résolution d'un problème de Stokes généralisé, au prix de l'évaluation d'un large nombre de caractéristiques χ .

On notera que l'on fait référence à cette méthode pour justifier de l'intérêt de la résolution du problème de Stokes. On ne l'a toutefois pas mise en œuvre, car comme on le verra ultérieurement, les performances de simulation obtenues ne sont pas encore suffisantes pour entreprendre la résolution de problèmes non-stationnaires.

5.2.2 Point fixe et linéarisation du terme advectif

Une autre technique, plus directe que la précédente, consiste à linéariser directement le terme d'advection et à utiliser une méthode de point fixe pour approcher itérativement la solution du problème non-linéaire de Navier-Stokes.

Dans ce cas, il est nécessaire de considérer le problème auxiliaire d'Oseen, qui représente la linéarisation des équations de Navier-Stokes autour d'un champ w tel que $\nabla \cdot w = 0$:

$$\begin{cases} \frac{\partial u}{\partial t} - (w \cdot \nabla)u + \nu \Delta u - \nabla p = f \\ \nabla \cdot u = 0 \end{cases}$$

et qui, après discrétisation en temps, aboutit au problème :

$$L_{Oseen}(w; u, p) \equiv \begin{cases} \frac{1}{\delta t} u^{t+1} - (w \cdot \nabla) u^{t+1} + \nu \Delta u^{t+1} - \nabla p^{t+1} = f^{t+1} + \frac{1}{\delta t} u^t \\ \nabla \cdot u^{t+1} = 0 \end{cases}$$

qui est également un problème de Stokes généralisé.

Sur un pas de temps, la solution recherchée au problème de Navier-Stokes, $L_{Oseen}(u; u, p)$, s'approxime alors par une méthode de point fixe en considérant une séquence de problèmes linéaires de type :

$$\begin{aligned} w_0 &= u^t \\ \left\{ \begin{array}{l} L_{Oseen}(w_i; u_i, p_i) \\ w_{i+1} = u_i \end{array} \right. & \quad i = 0, \dots, n \\ (u^{t+1}, p^{t+1}) &= (u_n, p_n) \end{aligned}$$

Avec cette méthode, la simulation d'un pas de temps des équations de Navier-Stokes s'apparente donc à la résolution d'une séquence de problèmes de Stokes généralisés. Il en est de même pour la résolution du problème de Navier-Stokes stationnaire, comme on le verra dans les résultats.

5.3 Formulation variationnelle du problème de Stokes

Que l'on recoure à l'une ou l'autre des méthodes de discrétisation en temps précédemment présentées, le problème à résoudre correspond à un problème de Stokes généralisé :

$$\begin{cases} l(u) - \nu \Delta u + \nabla p = f \\ \nabla \cdot u = 0 \end{cases}$$

où $l(u)$ correspond aux termes linéaires supplémentaires par rapport au problème de Stokes classique. Dans la suite de la présentation, et pour plus de simplicité, on se concentrera sur le problème de Stokes uniquement, sans ce terme qui est d'un traitement similaire à celui du Laplacien. On reviendra brièvement sur son inclusion à l'occasion de la résolution numérique, en discutant à ce propos les conséquences de son inclusion.

5.3.1 Problème fort

La première étape de la méthode des éléments finis est la mise sous forme variationnelle du problème considéré. Dans cette étape, on va multiplier les équations aux dérivées partielles du problème initial par des fonctions-tests, et l'on va opérer une intégration par partie. Celle-ci est destinée d'une part à affaiblir les contraintes de différentiabilité sur la fonction recherchée, et d'autre part à faire apparaître des conditions aux limites naturelles, qui possèdent une interprétation physique.

On considère donc un domaine Ω de frontière $\Gamma = \partial\Omega$, dont la normale extérieure est notée n et avec $\{\Gamma_1, \Gamma_2, \Gamma_3\}$ une partition de Γ . Sur ce domaine,

on pose le problème de Stokes suivant :

$$\begin{cases} -\nu\Delta u + \nabla p = f \\ \nabla \cdot u = 0 \end{cases} \quad \text{sur } \Omega$$

muni des conditions aux limites

$$\begin{cases} u = u_D & \text{sur } \Gamma_1 \\ -\nu \frac{\partial u}{\partial n} + p \cdot n = 0 & \text{sur } \Gamma_2 \\ u \cdot n = u_D \cdot n & \text{sur } \Gamma_3 \\ -\nu \frac{\partial u}{\partial n} \cdot \tau_i = 0 \Big|_{i=1,2} & \text{sur } \Gamma_3 \end{cases}$$

La première condition aux limites, posée sur Γ_1 , correspond à une condition de Dirichlet sur la vitesse, qui permet de modéliser aussi bien la paroi d'un obstacle qu'une frontière artificielle du domaine dont la vitesse est forcée. La deuxième condition, posée sur Γ_2 , correspond à la modélisation d'une frontière de sortie. Il s'agit d'une condition de Neumann sur la vitesse. Finalement, les deux dernières conditions, posées sur Γ_3 correspondent à une condition de Dirichlet sur la composante normale de la vitesse permettant d'imposer l'imperméabilité d'une frontière, et à une condition de Neumann tangente à la frontière du domaine. Dans ce cas (n, τ_1, τ_2) forme une base orthonormée définie sur la surface frontière du domaine, Γ , et $d\tau_1$ et τ_2 correspondent à des vecteurs tangents. La fonction u_D est connue et constitue une donnée du problème.

On a mentionné précédemment que l'on emploie l'approche de Galerkin, qui consiste à considérer un espace d'approximation de la solution identique à l'espace des fonctions-tests. Cela n'est pas tout-à-fait exact dans le cas où les conditions aux limites de Dirichlet ne sont pas homogènes, c'est-à-dire que $u_D \neq 0$. En effet, dans ce cas, $u = u_D \neq 0$ doit être satisfait sur Γ_1 , or les fonctions tests se doivent d'être nulles là où des conditions de Dirichlet sont appliquées : $v|_{\Gamma_1} = 0$, ce qui est incohérent.

Pour regagner la cohérence recherchée, il faut considérer le problème homogène correspondant au problème non-homogène de départ, que l'on obtient en posant $w = u - u_D$, et qui porte alors sur le champ w plutôt que sur u . Pour plus de simplicité, on conserve cependant le problème originel, et l'on incorporera les conditions aux limites durant l'étape de résolution du système linéaire discrétisé. On note bien que cette prise en compte discrète des conditions aux limites est la contrepartie directe de l'homogénéisation.

Lorsque l'on énonce que l'on applique la méthode de Galerkin, il faut donc entendre que les deux espaces considérés, celui d'approximation et celui des fonctions-tests, sont identiques à l'exception de leur valeur sur la frontière du domaine, mais que l'on pourrait se placer de manière équivalente dans le cadre où l'hypothèse d'égalité est satisfaite, en considérant l'équation homogénéisée. C'est à cause de cette différence sur la frontière que l'on note V l'espace des fonctions-tests de la vitesse et \tilde{V} l'espace de la solution de la vitesse, même si l'on énonce que l'espace d'approximation et l'espace-test sont choisis identiques.

5.3.2 Première forme variationnelle

Étant donné la définition précédente, la forme variationnelle du problème s'écrit :

$$\begin{aligned} & \text{Trouver } (u, p) \in \tilde{V} \times Q \text{ tel que :} \\ & \left\{ \begin{array}{l} \int_{\Omega} [-\nu \Delta u + \nabla p] \cdot v \, dx = \int_{\Omega} f \cdot v \, dx \\ \int_{\Omega} [\nabla \cdot u] q \, dx = 0 \end{array} \right. \quad \forall (v, q) \in V \times Q \\ & \text{avec les conditions aux limites} \end{aligned}$$

On modifie maintenant cette expression en utilisant les formules de Green suivantes, qui correspondent à des relations d'intégration par partie en dimensions supérieures à un :

$$\begin{aligned} \int_{\Omega} [\Delta u] \cdot v \, dx &= - \int_{\Omega} \nabla u : \nabla v \, dx + \int_{\Gamma} v \cdot \frac{\partial u}{\partial n} \, ds \\ \int_{\Omega} [\nabla p] \cdot v \, dx &= - \int_{\Omega} p \nabla \cdot v \, dx + \int_{\Gamma} p v \cdot n \, ds \end{aligned}$$

Ces relations font usage de la notation \cdot pour décrire le produit scalaire des vecteurs, et $:$ pour décrire le produit scalaire des tenseurs de rang deux².

On obtient alors la formulation variationnelle suivante :

$$\begin{aligned} & \text{Trouver } (u, p) \in \tilde{V} \times Q \text{ tel que :} \\ & \left\{ \begin{array}{l} \nu \int_{\Omega} \nabla u : \nabla v \, dx - \int_{\Omega} p \nabla \cdot v \, dx + \int_{\Gamma} -\nu v \cdot \frac{\partial u}{\partial n} + p v \cdot n \, ds = \int_{\Omega} f v \, dx \\ \int_{\Omega} [\nabla \cdot u] q \, dx = 0 \end{array} \right. \\ & \quad \forall (v, q) \in V \times Q \\ & \text{avec les conditions aux limites} \end{aligned}$$

5.3.3 Intégration des conditions aux limites

Le terme de bord

$$\int_{\Gamma} -\nu v \cdot \frac{\partial u}{\partial n} + p v \cdot n \, ds$$

est ensuite décomposé selon la partition de la frontière pour y introduire les conditions aux limites. Donc :

$$\begin{aligned} \int_{\Gamma} -\nu v \cdot \frac{\partial u}{\partial n} + p v \cdot n \, ds &= \int_{\Gamma_1} -\nu v \cdot \frac{\partial u}{\partial n} + p v \cdot n \, ds \\ &+ \int_{\Gamma_2} -\nu v \cdot \frac{\partial u}{\partial n} + p v \cdot n \, ds \\ &+ \int_{\Gamma_3} -\nu v \cdot \frac{\partial u}{\partial n} + p v \cdot n \, ds \end{aligned}$$

²On rappelle que le produit scalaire de deux tenseurs de rang deux est défini comme $p : q = [p]_{i,j} : [q]_{i,j} = \sum_{i,j} p_{i,j} \cdot q_{i,j}$

Sur Γ_1 , où sont posées des conditions de Dirichlet, on impose que la fonction-test v soit nulle. Par conséquent :

$$\int_{\Gamma_1} -\nu v \cdot \frac{\partial u}{\partial n} + pv \cdot n \, ds = \int_{\Gamma_1} v \cdot \left[-\nu \frac{\partial u}{\partial n} + pn \right] \, ds = \int_{\Gamma_1} 0 \, ds = 0$$

Sur Γ_2 , la condition aux limites naturelle que l'on a posée assure naturellement que le terme de bord soit nul :

$$\int_{\Gamma_2} -\nu v \cdot \frac{\partial u}{\partial n} + pv \cdot n \, ds = \int_{\Gamma_2} v \cdot \left[-\nu \frac{\partial u}{\partial n} + pn \right] \, ds = \int_{\Gamma_2} 0 \, ds = 0$$

Sur Γ_3 , la double condition aux limites agit de manière similaire. Étant donnée la base orthonormée (n, τ_1, τ_2) , v peut se décomposer comme :

$$v = (v \cdot n)n + (v \cdot \tau_1)\tau_1 + (v \cdot \tau_2)\tau_2$$

Dans la mesure où une condition de Dirichlet normale est imposée sur la vitesse, on impose que la fonction-test satisfasse $v \cdot n = 0$, ce qui entraîne que :

$$\begin{aligned} \int_{\Gamma_3} -\nu v \cdot \frac{\partial u}{\partial n} + pv \cdot n \, ds &= \int_{\Gamma_3} v \cdot \left[-\nu \frac{\partial u}{\partial n} + pn \right] \, ds \\ &= \int_{\Gamma_3} [(v \cdot \tau_1)\tau_1 + (v \cdot \tau_2)\tau_2] \cdot \left[-\nu \frac{\partial u}{\partial n} + pn \right] \, ds \\ &= \sum_{i=1,2} \int_{\Gamma_3} (v \cdot \tau_i)\tau_i \cdot \left[-\nu \frac{\partial u}{\partial n} + pn \right] \, ds \\ &= \sum_{i=1,2} \int_{\Gamma_3} (v \cdot \tau_i) \left[-\nu \frac{\partial u}{\partial n} \cdot \tau_i \right] + (v \cdot \tau_i)p(n \cdot \tau_i) \, ds \end{aligned}$$

Comme les conditions naturelles sur Γ_3 éliminent le terme sur la vitesse, alors que $n \cdot \tau_i = 0$ par définition de la base, il vient que le terme de bord est également nul sur Γ_3 :

$$\int_{\Gamma_3} -\nu v \cdot \frac{\partial u}{\partial n} + pv \cdot n \, ds = 0$$

Pour résumer, le terme de bord est complètement nul sous les conditions suivantes sur v :

$$\int_{\Gamma} -\nu v \cdot \frac{\partial u}{\partial n} + pv \cdot n \, ds = 0 \quad \forall v \in V : v = 0 \text{ sur } \Gamma_1 \text{ et } v \cdot n = 0 \text{ sur } \Gamma_3$$

5.3.4 Forme variationnelle finale

La formulation variationnelle finalement retenue peut donc faire omission du terme de bord, pour devenir

$$\begin{aligned} &\text{Trouver } (u, p) \in \tilde{V} \times Q \text{ tel que :} \\ &\left\{ \begin{array}{l} \nu \int_{\Omega} \nabla u : \nabla v \, dx - \int_{\Omega} p \nabla \cdot v \, dx = \int_{\Omega} f v \, dx \\ - \int_{\Omega} [\nabla \cdot u] q \, dx = 0 \end{array} \right. \quad \forall (v, q) \in V \times Q \end{aligned}$$

et où les contraintes sur les différentes fonctions — nullité sur une partie de la frontière pour v ou valeur prescrite sur la frontière pour u — sont intégrées aux espaces V et \tilde{V} comme on va le poser immédiatement. On a également changé le signe de la deuxième équation pour faire apparaître une symétrie avec le terme incluant la pression de la première équation.

Il reste donc à déterminer à quels espaces appartiennent les différentes fonctions manipulées pour décrire complètement la formulation variationnelle du problème. Pour cela, on examine les différents termes écrits précédemment, et on part du principe que pour qu'un terme du type $\int_{\Omega} uv \, dx$ ait un sens, il faut que u et v soient de carrés intégrables sur Ω , c'est-à-dire que $u, v \in L^2(\Omega)$. Par conséquent, il faut que :

$$\nabla u, v, \nabla v, f, p, q \in L^2(\Omega)$$

Avec la définition usuelle suivante

$$H^1(\Omega) = \{\varphi \in L^2(\Omega) : \frac{\partial \varphi}{\partial x_i} \in L^2(\Omega) \text{ avec } i = 1, 2, 3\}$$

et en intégrant les contraintes de nullité sur la frontière, là où sont posées les conditions de Dirichlet, on peut définir les espaces V , \tilde{V} et Q comme :

$$\begin{aligned} V &= \{v \in H^1(\Omega)^3 : v = 0 \text{ sur } \Gamma_1 \text{ et } v \cdot n = 0 \text{ sur } \Gamma_3\} \\ \tilde{V} &= \{v \in H^1(\Omega)^3 : v = u_D \text{ sur } \Gamma_1 \text{ et } v \cdot n = u_D \cdot n \text{ sur } \Gamma_3\} \\ Q &= \begin{cases} L_0^2(\Omega) = \{q \in L^2(\Omega) : \int_{\Omega} q \, dx = 0\} & \text{si } \Gamma_2 = \emptyset \\ L^2(\Omega) & \text{si } \Gamma_2 \neq \emptyset \end{cases} \end{aligned}$$

La définition de l'espace Q varie légèrement selon la présence ou non de Γ_2 . En effet, si $\Gamma_2 = \emptyset$, la seule intervention de la pression dans le problème se fait par le biais de son gradient, ce qui signifie qu'elle n'est définie qu'à une constante près. Pour éliminer cette constante, on ajoute donc la contrainte supplémentaire que $\int_{\Omega} p \, dx = 0$. Au contraire, si $\Gamma_2 \neq \emptyset$, p est liée à u sur Γ_2 , ce qui rend cette contrainte supplémentaire superflue.

Pour ce qui concerne les paramètres, les exigences du problème sont reportées sur les données :

$$\begin{aligned} f &\in L^2(\Omega)^3 \\ u_D &\in L^2(\Gamma_1)^3 \cap L^2(\Gamma_3)^3 \end{aligned}$$

On remarquera que les conditions aux limites essentielles, de Dirichlet, doivent être explicitement prises en compte — bien qu'elles soient intégrées dans les définitions des espaces \tilde{V} et Q et donc quelque peu masquées — alors que les autres conditions aux limites, les conditions naturelles, sont traitées naturellement par la formulation posée, d'où leur nom.

On précise également que l'on peut montrer que ce problème variationnel est bien posé, et qu'il admet une solution unique. Cette solution, si elle est suffisamment régulière, se trouve également être une solution du problème fort de Stokes considéré au départ. Bien que ces points soient primordiaux, on admettra ici ces résultats sans chercher à les démontrer.

Formulation abstraite Afin d'en simplifier la manipulation ultérieure, on réécrit le problème que l'on vient de poser sous une forme abstraite. En introduisant les formes bilinéaires suivantes :

$$\begin{aligned} a : \tilde{V} \times V &\longrightarrow \mathbb{R} \\ (u, v) &\longmapsto \nu \int_{\Omega} \nabla u : \nabla v \, dx \end{aligned}$$

$$\begin{aligned} b : (\tilde{V} \cup V) \times Q &\longrightarrow \mathbb{R} \\ (v, q) &\longmapsto - \int_{\Omega} \nabla \cdot v q \, dx \end{aligned}$$

ainsi que la forme linéaire :

$$\begin{aligned} l : V &\longrightarrow \mathbb{R} \\ v &\longmapsto \int_{\Omega} f v \, dx \end{aligned}$$

le problème variationnel peut être réécrit sous la forme :

$$\begin{aligned} &\text{Trouver } (u, p) \in \tilde{V} \times Q \text{ tel que :} \\ &\begin{cases} a(u, v) + b(v, p) = l(v) \\ b(u, q) = 0 \end{cases} \quad \forall (v, q) \in V \times Q \end{aligned}$$

5.4 Discrétisation en espace

Partant de la formulation variationnelle définie dans la section ci-dessus, la méthode des éléments finis va produire un système d'équations linéaires. Ce système est dérivé en remplaçant dans la formulation variationnelle précédente les espaces V et Q infinis par deux espaces de dimensions finies $V_h \subset V$ et $Q_h \subset Q$. Ces espaces ne sont cependant pas choisis au hasard. Au contraire, la méthode lie leur définition à un maillage du domaine de simulation, ce qui va permettre de tenir compte de la géométrie effective du domaine. Par ailleurs, la définition va également tenir compte des conditions aux limites de Dirichlet à imposer dans le problème, en faisant apparaître des degrés de liberté — que l'on peut considérer pour le moment comme les variables du problème discret — correspondant exactement à ces conditions.

La dérivation abstraite du problème discret de Stokes est présentée la première, dans la section 5.4.1. Les deux sections suivantes 5.4.2 et 5.4.3 décrivent comment sont construits les espaces d'approximation discrets de manière générale dans la méthode des éléments finis. La section 5.4.4 présente ensuite les espaces que nous avons employés pour discrétiser le problème de Stokes. Finalement, la mise en œuvre des calculs de la discrétisation est détaillée dans son principe général dans la section 5.4.5, et spécifiquement pour le problème de Stokes dans la section 5.4.6.

Dans cette section, la présentation sera effectuée en ne considérant que des éléments finis tétraédriques, maillant un espace géométrique de dimension trois, pour autoriser une description plus concrète. Toutefois, comme on le verra dans la section 5.6, d'autres éléments ont été testés et le cas de problèmes en dimension deux a également été testé, à des fins de validation.

5.4.1 Problème discret

Comme précisé précédemment, la discrétisation s'appuie sur l'introduction dans le problème variationnel d'espaces $V_h \subset V$, $\tilde{V}_h \subset \tilde{V}$ et $Q_h \subset Q$, de dimensions finies, et construits à partir d'un maillage \mathcal{T}_h . Une fois les espaces choisis, le problème se réécrit alors :

$$\begin{aligned} & \text{Trouver } (u_h, p_h) \in \tilde{V}_h \times Q_h \text{ tel que :} \\ & \begin{cases} a(u_h, v_h) + b(v_h, p_h) = l(v_h) \\ b(u_h, q_h) = 0 \end{cases} \quad \forall (v_h, q_h) \in V_h \times Q_h \end{aligned}$$

Dans la mesure où V_h et Q_h sont de dimensions finies, et munis de leur bases respectives $\{\phi_i\}_{1 \leq i \leq n}$ et $\{\psi_j\}_{1 \leq j \leq m}$, ce problème est, par passage à la base, équivalent au problème suivant :

$$\begin{aligned} & \text{Trouver } (u_h, p_h) \in \tilde{V}_h \times Q_h \text{ tel que :} \\ & \begin{cases} a(u_h, \phi_i) + b(\phi_i, p_h) = l(\phi_i) & \forall \phi_i : i = 1, \dots, n \\ b(u_h, \psi_j) = 0 & \forall \psi_j : j = 1, \dots, m \end{cases} \end{aligned}$$

On décompose également u_h et p_h , les approximations recherchées, sur les bases respectives des espaces dans lesquels on les recherche — en rappelant que $V_h = \tilde{V}_h$ sauf en ce qui concerne les frontières du domaine —

$$\begin{aligned} u_h &= \sum_{k=1}^n u_k \phi_k \\ p_h &= \sum_{l=1}^m p_l \psi_l \end{aligned}$$

et on reporte ces décompositions dans les équations précédentes. En tirant parti de la bilinéarité de a et de b , le problème peut alors être réécrit sous la forme :

$$\begin{aligned} & \text{Trouver } U_h = [u_k] \in \mathbb{R}^n \text{ et } P_h = [p_l] \in \mathbb{R}^m \\ & \begin{cases} \sum_k u_k \cdot a(\phi_k, \phi_i) + \sum_l p_l \cdot b(\phi_i, \psi_l) = l(\phi_i) & \forall \phi_i : i = 1, \dots, n \\ \sum_k u_k \cdot b(\phi_k, \psi_j) = 0 & \forall \psi_j : j = 1, \dots, m \end{cases} \\ & \text{avec les conditions aux limites discrètes} \end{aligned}$$

Les conditions aux limites discrètes mentionnées correspondent aux conditions de Dirichlet imposées sur la frontière, et doivent être vérifiées par les degrés de liberté — grossièrement les inconnues du problème discret — associés aux nœuds — grossièrement les points d'échantillonnages des inconnues — résidant sur ces frontières. À supposer que les fonctions de base choisies s'y prêtent — et elles sont construites pour satisfaire à cette exigence — les conditions aux limites discrètes correspondent alors à des relations du type $u_i = u_{D,i}$.

La construction de fonctions de bases satisfaisantes sur ces points est discutée à la fin de la section 5.4.6, la prise en compte numérique des conditions aux limites, dans la section 5.5.

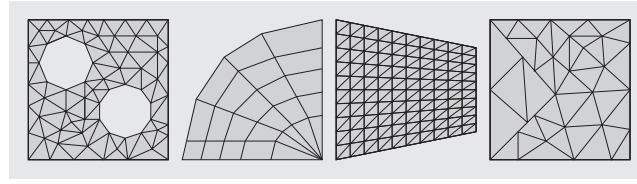


Figure 5.1: Exemples de maillages. De gauche à droite : maillage uniquement composé de triangles, maillage avec différents types d'éléments, maillage régulier et maillage non-conforme.

Le problème discret émanant du problème variationnel apparaît finalement sous la forme d'un système linéaire muni de la structure :

$$\begin{bmatrix} A_h & B_h \\ B_h^T & 0 \end{bmatrix} \cdot \begin{bmatrix} U_h \\ P_h \end{bmatrix} = \begin{bmatrix} F_h \\ 0 \end{bmatrix}$$

en définissant les différentes quantités comme suit :

$$\begin{aligned} [A_h]_{i,j} &= a(\phi_j, \phi_i) \\ [B_h]_{i,j} &= b(\phi_i, \psi_j) \\ [F_h]_i &= l(\phi_i) \end{aligned}$$

Ce problème est également complété par les conditions aux limites discrètes que l'on a évoquées.

5.4.2 Représentation du domaine

Pour définir les espaces V_h et Q_h , et leurs bases, nécessaires à la discrétisation, la méthode s'appuie sur un maillage \mathcal{T}_h du domaine de simulation Ω . Les contraintes qui s'exercent sur ce maillage sont relativement faibles : ses différents éléments $\{K_i\}_i$, non-vides et en nombre fini, doivent former une partition de Ω . Le maillage doit également être géométriquement conforme, ce qui signifie que considérant deux éléments $K_i, K_j \in \mathcal{T}_h$ avec $i \neq j$, leur intersection satisfait la condition suivante : $K_i \cap K_j$ est soit vide, soit un sommet commun de la triangulation, soit une arête commune de la triangulation, soit une face commune de la triangulation. En particulier, ces conditions autorisent l'emploi de maillages non-structurés. Plusieurs maillages, présentant des caractéristiques diverses, sont illustrés dans la figure 5.1.

Le rôle de chaque élément de la partition va être de servir de support à la définition d'une approximation par morceau des espaces de fonctions manipulés. Ceci explique pourquoi les éléments sont choisis de forme simple, usuellement des simplexes ou des cubes, éventuellement légèrement déformés. La possibilité de recourir à des simplexes permet alors de mailler n'importe quel domaine sans contrainte d'orientation de grille comme dans les méthodes de différence finies.

On caractérise un maillage \mathcal{T}_h par sa finesse h , *via* le diamètre de ses éléments :

$$h = \max_{K \in \mathcal{T}_h} h_K$$

$$h_K = \text{diam}(K) = \max_{x_1, x_2 \in K} |x_1 - x_2|_2$$

Cette notion de finesse est nécessaire pour mesurer la convergence des approximations calculées par la méthode des éléments finis, vers la solution exacte de l'équation aux dérivées partielles à résoudre. Cette mesure ne constitue cependant pas une mesure de précision dans l'absolu mais plutôt une mesure relative au sein d'une séquence de maillages de raffinements différents $\{\mathcal{T}_h\}_{h>0}$.

Dans le cadre de ce travail, on ne manipulera que des éléments simples, des simplexes ou cubes, rectilignes et non courbes. Ceci a pour conséquence que l'on se restreindra à des domaines polygonaux. Un seul type d'éléments par maillage constituera une autre restriction.

5.4.3 Construction des espaces d'approximation

Les espaces d'approximations employés dans la méthode des éléments finis s'appuient sur un maillage \mathcal{T}_h , en considérant des approximations par morceaux, sur chacune des cellules. Pour définir formellement cet espace d'approximation, un élément fini est associé à chaque cellule du maillage.

Définition d'un élément fini

Intuitivement, un élément fini est un triplet (K, P, Σ) destiné à la représentation de l'espace vectoriel de fonctions P sur le domaine géométrique K . Pour capturer l'espace P , qui est usuellement choisi comme un espace de fonctions polynomiales, un nombre fini de degrés de liberté, regroupés dans l'ensemble Σ est employé. Ces degrés de liberté matérialisent en un sens le moyen que l'on se donne de représenter des éléments de P .

À chaque degré de liberté particulier de Σ , on peut associer une fonction particulière, dite fonction de forme. Collectivement, ces fonctions forment une base de P . À ce titre, on peut dès à présent noter qu'elles vont jouer un rôle important dans la discrétisation, en fournissant le support de l'approximation recherchée.

Formellement, suivant la définition donnée par Ciarlet [25], un élément fini est un triplet (K, P, Σ) pour lequel :

1. K est un sous-ensemble, connecté, compact, Lipschitzien, de \mathbb{R}^d et d'intérieur non vide.
2. P est un espace vectoriel de fonctions $p : K \rightarrow \mathbb{R}^m$.
3. Σ est un ensemble $\{\sigma_1, \dots, \sigma_{n_{sh}}\}$ de n_{sh} formes linéaires agissant sur les éléments de P et tel que l'application linéaire

$$p \in P \longmapsto [\sigma_1(p), \dots, \sigma_{n_{sh}}(p)] \in \mathbb{R}^{n_{sh}}$$

soit bijective.

Avec cette définition, Σ forme une base de l'espace $\mathcal{L}(P, \mathbb{R})$ des applications linéaires de P dans \mathbb{R} . Les formes linéaires $\{\sigma_i\}_i$ sont alors appelées les degrés de liberté.

La propriété exigée sur Σ signifie que les $\{\sigma_i\}_i$ doivent être nécessaires et suffisants pour représenter n'importe quel élément $p \in P$. Dans ce cas, l'élément est dit unisolvent. Ceci signifie que la donnée d'un vecteur $[\sigma_1(p), \dots, \sigma_{n_{sh}}(p)] \in \mathbb{R}^{n_{sh}}$ équivaut à la donnée d'une fonction continue de P définie sur K .

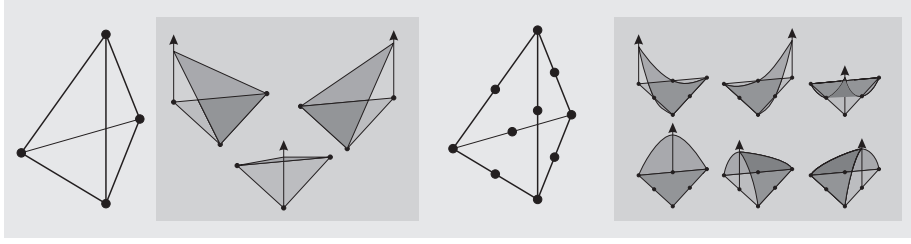


Figure 5.2: Exemples d'éléments de type Lagrange. Les éléments tétraédriques de Lagrange de degrés 1 et 2 pour la dimension 3 sont illustrés avec leurs nœuds. Les fonctions de forme associées sont présentées pour les cas bidimensionnels analogues des triangles de degrés 1 et 2.

Par suite de cette définition, il existe $(\theta_1, \dots, \theta_{n_{sh}})$ une base de P définie par

$$\sigma_i(\theta_j) = \delta_i^j \quad 1 \leq i, j \leq n_{sh}$$

dont les fonctions $\{\theta_i\}_i$ sont appelées fonctions de forme. C'est à partir de ces fonctions de formes qu'est construit, par morceaux et sur chaque élément d'un maillage, l'espace de fonctions associé au maillage.

Éléments finis de type Lagrange Pour plus de clarté, on présente deux exemples d'éléments finis, les éléments finis de Lagrange de degré un ou deux, aptes à représenter une fonction à valeur scalaire sur des domaines tétraédriques. Ils serviront ultérieurement de base à la définition des éléments effectivement utilisés pour discrétiser le problème de Stokes.

Un élément de type Lagrange choisit pour P un espace de polynômes de degré k donné, et définit ses degrés de liberté comme les valeurs prises par les polynômes $p \in P$ en un certain nombre de points du domaine.

Formellement, cela signifie qu'il existe un ensemble de points $\{a_i\}_i$ dans K tel que

$$\forall p \in P \quad \sigma_i(p) = p(a_i) \quad i = 1 \leq n_{sh}$$

Ces points sont appelés nœuds de l'élément fini. Par définition, chacune des n_{sh} fonctions de forme $\{\theta_i\}_i$ correspondantes, est un polynôme de degré k s'annulant sur tous les nœuds sauf un :

$$\delta_i^j = \sigma_i(\theta_j) = \theta_j(a_i)$$

Ceci signifie qu'une fonction p est représentée par ses valeurs prises en certains points de l'élément, les nœuds. Les fonctions de forme associées, dites nodales, associées aux degrés de liberté sont les polynômes d'interpolation de Lagrange en 1D, ou leur extension multidimensionnelle sur le domaine K pour les dimensions supérieures, d'où la dénomination des éléments.

Exemples On désigne par \mathbb{P}_k l'espace des polynômes en plusieurs variables x_1, \dots, x_d à coefficients réels de degré total inférieur à k :

$$\mathbb{P}_k = \left\{ p(x) = \sum_{\substack{0 \leq i_1, \dots, i_d \leq k \\ i_1 + \dots + i_d \leq k}} \alpha_{i_1, \dots, i_d} x^{i_1} \dots x^{i_d} \quad : \alpha_{i_1, \dots, i_d} \in \mathbb{R} \right\}$$

En dimension 3, on peut définir le tétraèdre \mathbb{P}_1 Lagrange par :

$$\left\{ \begin{array}{l} K = \text{tétraèdre} \subset \mathbb{R}^3 \\ P = \mathbb{P}_1 = \{\alpha_0 + \alpha_1 \cdot x + \alpha_2 \cdot y + \alpha_3 \cdot z : \alpha_i \in \mathbb{R}\} \\ \Sigma = \{\sigma_i(p) = p(a_i) : \{a_i\}_i = \{\text{sommets du tétraèdre}\}\} \\ \quad = \{\text{valeurs aux sommets du tétraèdre}\} \end{array} \right.$$

ou le tétraèdre \mathbb{P}_2 Lagrange par :

$$\left\{ \begin{array}{l} K = \text{tétraèdre} \subset \mathbb{R}^3 \\ P = \mathbb{P}_2 = \left\{ \begin{array}{l} \alpha_0 + \alpha_1 \cdot x + \alpha_2 \cdot y + \alpha_3 \cdot z \\ + \alpha_4 \cdot x^2 + \alpha_5 \cdot xy + \alpha_6 \cdot xz + \alpha_7 y^2 + \alpha_8 yz + \alpha_9 z^2 : \alpha_i \in \mathbb{R} \end{array} \right\} \\ \Sigma = \{\sigma_i(p) = p(a_i) : \{a_i\}_i = \{\text{sommets du tétraèdre et milieux des arêtes}\}\} \\ \quad = \{\text{valeurs aux sommets du tétraèdre et aux milieux des arêtes}\} \end{array} \right.$$

Dans ces exemples, les espaces de fonctions P représentables sur le domaine K à l'aide des degrés de liberté sont respectivement des polynômes de degré total inférieur à 1 ou 2. Les fonctions représentées sont alors interpolées entre les valeurs qu'elles prennent sur les nœuds des éléments. Les valeurs prises aux nœuds représentent les degrés de liberté Σ . Les fonctions de forme vérifient également la propriété suivante : elles prennent chacune la valeur 1 en un nœud différent de K et la valeur 0 sur tous les autres nœuds, d'où leur appellation de fonctions nodales. Une illustration de ces deux éléments est disponible dans la figure 5.2.

Bien que l'on n'ait illustré que ces éléments de type Lagrange, on notera que les éléments finis ne se restreignent pas à ces exemples, mais peuvent être bien plus riches dans leur représentation, *i.e.* dans leurs degrés de liberté. Ils peuvent par exemple inclure des valeurs de dérivées, de flux à travers des faces, ou de circulation le long d'arêtes comme degrés de libertés, et ne pas se contenter de valeurs ponctuelles.

Maillage et éléments

En s'appuyant sur un maillage \mathcal{T}_h , et en associant un élément fini à chaque cellule du maillage, il est possible de construire un espace d'approximation adapté à un domaine géométrique Ω , par morceau sur chaque cellule $K \in \mathcal{T}_h$. L'espace d'approximation global est défini de manière à être, sur chaque élément du maillage, égal à l'approximation locale assurée sur l'élément.

Il est habituel de chercher à construire un espace d'approximation global avec une continuité des fonctions à l'interface des éléments. Dans le cas des éléments de Lagrange, un tel espace peut être obtenu simplement. La continuité globale est en effet garantie par le partage des valeurs des degrés de liberté situés à l'interface entre cellules, à condition de considérer des éléments adjacents compatibles entre eux, c'est-à-dire des éléments dont les nœuds coïncident sur l'interface commune. Dans le cadre de ce travail, où les éléments sont symétriques et tous de même nature, cette condition est automatiquement remplie si le maillage est géométriquement conforme.

Les degrés de liberté représentant les inconnues du problème discret à résoudre, leur partage assure un couplage entre les différents éléments du maillage.

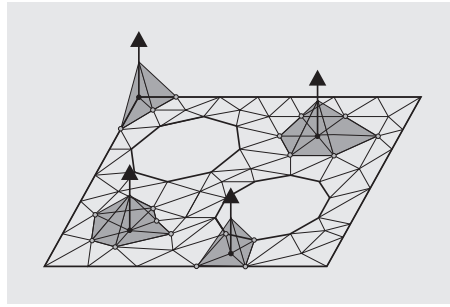


Figure 5.3: Fonctions de base globales. Les fonctions de base globales sont définies par morceaux, à partir des fonctions locales sur chaque élément. Quatre fonctions différentes sont illustrées ici, pour le cas d'éléments \mathbb{P}_1 Lagrange. Le support de chacune des fonctions est réduit aux éléments adjacents au nœud sur lequel la fonction est non-nulle.

On peut alors distinguer les degrés de liberté à deux niveaux différents : au niveau global du maillage, et au niveau local des éléments. L'ensemble des degrés de liberté, au niveau global du maillage, s'associe alors à toutes les inconnues du problème global.

Pour des éléments de type Lagrange, à l'échelle du maillage, des fonctions de forme globales $\{\bar{\varphi}\}_i$ peuvent alors être considérées en appliquant la caractérisation des polynômes de Lagrange : la prise d'une valeur 1 sur un nœud et 0 sur tous les autres nœuds. Par construction, il est clair que la restriction de ces fonctions globales aux différents éléments K du maillage est soit nulle, soit égale à une des fonctions de forme locales des éléments $\{\varphi_{K,j}\}_{K,j}$. Ces fonctions globales s'expriment donc par morceaux comme un assemblage des fonctions locales définies sur chaque élément. Par construction elles forment une base de l'espace d'approximation global. Comme l'illustre la figure 5.3, ces fonctions de forme globales ont pour support tous les éléments adjacents au nœud sur lequel elles prennent leur valeur unitaire.

Puisque ces fonctions globales forment une base de l'espace d'approximation global défini par le maillage, ce sont elles qui vont être employées pour discrétiser le problème variationnel. En outre, l'égalité sur chaque élément, des fonctions de forme globales avec leur contrepartie locale, autorise à décomposer les calculs à effectuer sur tout le maillage en une succession de calculs locaux sur les différents éléments. Les fonctions globales ne sont toutefois jamais manipulées pour effectuer des calculs comme on le verra dans la suite : elles ne sont considérées qu'à une échelle logique. Tous les calculs, réalisés élément par élément sont reportés sur les fonctions locales, qui coïncident localement avec les fonctions globales.

Ceci implique que, durant la mise en œuvre de la méthode, il est nécessaire de connaître la correspondance entre degrés de liberté locaux des éléments et degrés de liberté globaux du problème, comme le présente la figure 5.4. Cette correspondance n'est toutefois pas complexe à calculer : elle découle directement de la topologie du maillage de simulation et du type d'élément choisi.

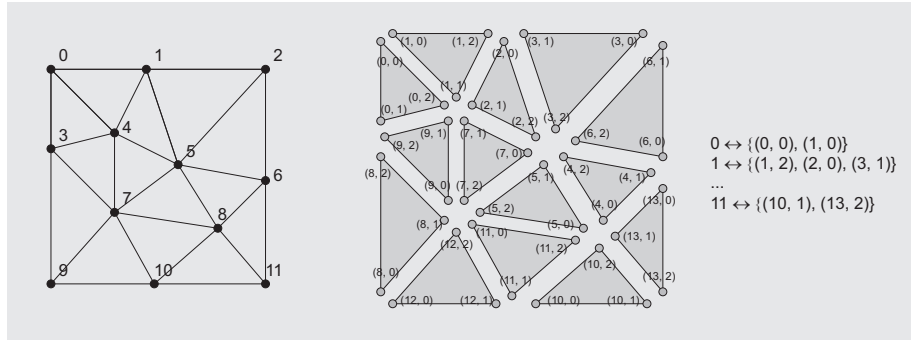


Figure 5.4: Correspondance locale-globale des degrés de liberté.

Élément de référence

Comme énoncé précédemment, un élément fini différent est associé à chaque cellule du maillage. Pour engendrer tous les éléments de même nature du maillage, on fait usage d'un élément de référence $(\hat{K}, \hat{P}, \hat{\Sigma})$ et d'un ensemble de transformations géométriques

$$T_K : \hat{K} \rightarrow K \quad \forall K \in \mathcal{T}_h$$

L'intérêt d'une telle définition est de pouvoir considérer tous les éléments du maillage au travers d'un seul représentant. Il est par conséquent possible d'exprimer tous les calculs nécessaires ayant trait à des éléments arbitraires (K, P, Σ) sur un seul et même élément de référence $(\hat{K}, \hat{P}, \hat{\Sigma})$, en tenant compte d'une transformation géométrique générale.

Pour le cas d'un élément de type Lagrange, où les degrés de liberté correspondent à la valeur prise en une série de points $\Sigma = \{p(a_i) : a_i \in K\}$, étant donné une transformation géométrique T_k connue, on définit l'élément fini réel (K, P, Σ) à partir de l'élément de référence $(\hat{K}, \hat{P}, \hat{\Sigma})$ comme :

$$\left\{ \begin{array}{l} K = T_k(\hat{K}) \\ P = \left\{ p = \hat{p} \circ T_k^{-1} : \hat{p} \in \hat{P} \right\} \\ \Sigma = \left\{ p(a_i) : a_i = T_k(\hat{a}_i), \hat{a}_i \in \hat{K} \right\} \end{array} \right.$$

On profite de cette réécriture pour introduire les fonctions de forme de l'élément $\{\varphi_i\}_i$ — qui forment une base de P — dans la définition de l'élément. Ceci est utile, car leur présence sera mise à profit dans les calculs ultérieurs de la discrétisation. La définition de l'élément devient alors :

$$\left\{ \begin{array}{l} K = T_k(\hat{K}) \\ P = \left\{ p = \sum_i b_i \varphi_i : \varphi_i = \hat{\varphi}_i \circ T_k^{-1} : \hat{\varphi}_i \in \hat{P} \right\} \\ \Sigma = \left\{ p(a_i) : a_i = T_k(\hat{a}_i), \hat{a}_i \in \hat{K} \right\} \end{array} \right.$$

Bien que les transformations T_k puissent être de formes générales sous certaines conditions, on ne s'intéressera dans ce travail qu'à des transformations affines, les plus simples qui soient.

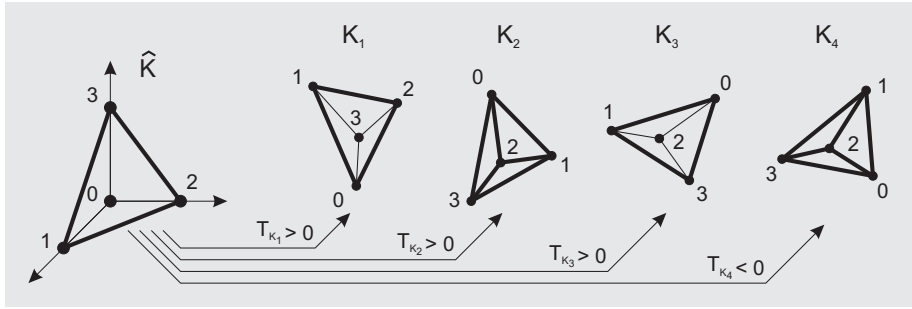


Figure 5.5: Éléments de référence. Les différents éléments $\{K_i\}_i$ d'un maillage peuvent être engendrés en considérant un élément de référence \hat{K} et un ensemble de transformations $\{T_{K_i}\}_i$. Les transformations $T_{K_1}, T_{K_2}, T_{K_3}$ préservent l'orientation, au contraire de T_{K_4} .

Une remarque est à formuler quant à l'utilisation d'éléments de référence. Les transformations employées doivent être inversibles et préserver l'orientation partout sur le domaine de définition : les éléments réels ne peuvent donc être trop déformés par rapport aux éléments de référence. En outre, les numérotations des degrés de liberté des éléments réels par rapport à l'élément de référence doivent être compatibles, sans quoi les calculs ne peuvent être effectués correctement. Dans le cas de transformations affines, l'orientation de la transformation est liée à la numérotation des sommets, comme illustré dans la figure 5.5. Il faut donc veiller à référencer correctement les sommets dans les éléments réels par rapport à l'élément de référence choisi.

Compte-tenu de l'association entre degrés de liberté locaux des éléments et globaux du maillage, et de la possibilité par conséquent d'identifier une fonction de base globale à une fonction de base locale sur chaque élément, si tous les éléments du maillage dérivent d'un élément de référence, tous les calculs requis peuvent se ramener à des calculs sur l'élément de référence muni de ses fonctions de base locales.

L'élément de référence étant alors l'archétype de tous les autres éléments, c'est lui seul que l'on décrira par la suite, sachant qu'il engendre tous les autres éléments. C'est également sur lui seul, muni d'une transformation T_k arbitraire, que l'on décrira les calculs effectués, avec ses fonctions de base locales.

5.4.4 Choix des espaces pour le problème de Stokes

La formulation du problème de Stokes employée est une formulation qui fait intervenir des variables de nature différente, la vitesse et la pression, de manière liée, d'où l'appellation de problème mixte. Cette liaison entre vitesse et pression a pour effet de restreindre le choix des espaces d'approximation qui ne peuvent donc être choisis séparément l'un de l'autre. Pour guider le choix d'une paire d'espaces compatibles, dite paire stable, une condition nécessaire et suffisante, la condition Ladyzenskaya-Babuška-Brezzi ou condition inf-sup doit être remplie. Elle garantit alors que le problème discret extrait de la formulation variationnelle est bien posé, c'est-à-dire qu'il admet une solution unique.

Parmi les paires d'espaces stables existantes, on en a testé deux, que l'on

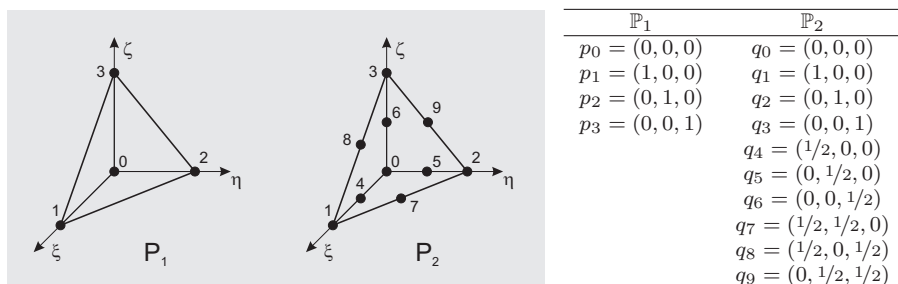


Figure 5.6: Tétrahédres de référence \mathbb{P}_1 , à gauche, et \mathbb{P}_2 , à droite. Les deux tétrahédres sont inscrits dans le cube unité. Les coordonnées des nœuds de référence sont rappelées dans le tableau.

décrit par les éléments finis de référence qui les matérialisent, représentés dans la figure 5.7. Les deux éléments permettent de définir des champs continus en vitesse et en pression, s'ils sont associés aux mailles d'un maillage conforme du domaine de simulation.

Interpolation de Lagrange dans un tétraèdre

Afin de simplifier la présentation des éléments, on va commencer par dériver les bases nodales de degré 1 et 2 pour des tétraèdres, avant de présenter les éléments qui en font usage.

On considère pour cela sur le tétraèdre de référence \widehat{K} représenté dans la figure 5.6, et défini comme

$$\widehat{K} = \{(\xi, \eta, \zeta) \in \mathbb{R}^3 : 0 \leq \xi, \eta, \zeta \wedge \xi + \eta + \zeta \leq 1\}$$

Sur ce tétraèdre de référence, on choisit comme degrés de liberté pour représenter \mathbb{P}_1 les valeurs aux sommets $\{p_i\}_i$. Le calcul des fonctions de forme $\{\theta_j^1\}_j$, de Lagrange de degré 1, obéit alors à la caractérisation :

$$\theta_j^1(p_i) = \delta_i^j$$

ce qui amène aux fonctions de base suivantes :

$$\begin{aligned} \theta_0^1 &= 1 - \xi - \eta - \zeta \\ \theta_1^1 &= \xi \\ \theta_2^1 &= \eta \\ \theta_3^1 &= \zeta \end{aligned}$$

et qui correspondent aux coordonnées barycentriques dans le tétraèdre. On pose d'ailleurs $\lambda = 1 - \xi - \eta - \zeta$.

Pour représenter \mathbb{P}_2 , on choisit pour degrés de liberté les valeurs aux sommets ainsi que les valeurs aux milieux des arêtes. Les sommets et milieux des arêtes sont nommés $\{q_i\}_i$. Le calcul des fonctions de forme $\{\theta_j^2\}_j$, de Lagrange de degré 2, obéit à la même caractérisation que précédemment :

$$\theta_j^2(q_i) = \delta_i^j$$

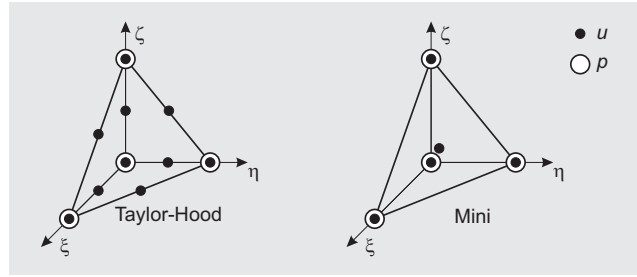


Figure 5.7: Élément pour le problème de Stokes. À gauche : élément de Taylor-Hood, également dit $\mathbb{P}_2/\mathbb{P}_1$. À droite : élément mini, également dit $\mathbb{P}_1 - \text{bulle}/\mathbb{P}_1$. Sont figurés les différents degrés de liberté en vitesse et en pression. Chaque degré de liberté de la vitesse est triple pour tenir compte du caractère vectoriel de la vitesse.

ce qui amène aux fonctions de base suivantes :

$$\begin{aligned}
 \theta_0^2 &= \lambda(2\lambda - 1) & \theta_4^2 &= 4\lambda\xi \\
 \theta_1^2 &= \xi(2\xi - 1) & \theta_5^2 &= 4\lambda\eta \\
 \theta_2^2 &= \eta(2\eta - 1) & \theta_6^2 &= 4\lambda\zeta \\
 \theta_3^2 &= \zeta(2\zeta - 1) & \theta_7^2 &= 4\xi\eta \\
 & & \theta_8^2 &= 4\xi\zeta \\
 & & \theta_9^2 &= 4\eta\zeta
 \end{aligned}$$

Élément de Taylor-Hood

Le premier élément considéré pour discrétiser le problème de Stokes, dit de Taylor-Hood, ou $\mathbb{P}_2/\mathbb{P}_1$ et représenté à gauche dans la figure 5.7, repose sur une approximation polynomiale de degré 2 pour la vitesse et de degré 1 pour la pression. Cet élément peut être considéré comme la superposition d'un triple tétraèdre \mathbb{P}_2 Lagrange pour la vitesse, et d'un tétraèdre \mathbb{P}_1 Lagrange pour la pression.

Formellement, l'élément fini mixte de Taylor-Hood correspond donc à

$$(\widehat{K}, [\mathbb{P}_2]^3, \widehat{\Sigma}_u) / (\widehat{K}, \mathbb{P}_1, \widehat{\Sigma}_p)$$

avec

$$\begin{aligned}
 \widehat{\Sigma}_u &= \{\sigma_{i,j} : p \in [\mathbb{P}_2]^3 \mapsto p(q_i) \cdot e_j, j = 1, 2, 3\} \\
 \widehat{\Sigma}_p &= \{\sigma_i : p \in \mathbb{P}_1 \mapsto p(p_i)\}
 \end{aligned}$$

en se référant à la figure 5.6 pour définir la position des points $\{p_i\}_i$ et $\{q_i\}_i$ et où e_j représente la composante j de la base Cartésienne de l'espace Euclidien.

Pour les deux variables, les fonctions de forme sont nodales. Les différentes fonctions de base associées aux nœuds de cet élément sont définies comme :

$$\begin{cases} u : \phi_{i,j} = \theta_i^2 e_j & i = 0 \dots, 9; j = 0, 1, 2 \\ p : \psi_i = \theta_i^1 & i = 0, \dots, 3 \end{cases}$$

ce qui représente un total de 34 degrés de liberté par élément. Pour pouvoir ordonner les degrés de liberté de la vitesse dans un vecteur, il est nécessaire de

leur assigner une numérotation linéaire. On choisit pour cela une numérotation corrélative des différentes composantes de la vitesse, *i.e.* la numérotation définie par $k = 3 \times i + j$. Pour la clarté d'écriture, on emploiera l'une ou l'autre notation, ϕ_k ou $\phi_{i,j}$ selon le contexte pour désigner une fonction de base arbitraire ou une fonction associée à une coordonnée bien particulière.

Pour un maillage composé de ces éléments, l'espace d'approximation engendré correspond à celui des fonctions continues dont la restriction à chaque élément du maillage est un polynôme de degré 2 pour chaque composante de la vitesse, et de degré 1 pour la pression.

Élément mini

Le second élément considéré, dit élément-mini, ou \mathbb{P}_1 -bulle/ \mathbb{P}_1 et représenté à droite dans la figure 5.7 considère une approximation polynomiale de degré 1 enrichie par une fonction bulle pour la vitesse, et de degré 1 pour la pression. Il sort donc du cadre d'une stricte interpolation de Lagrange.

La fonction bulle est une fonction polynomiale de degré 4, nulle sur les bords de l'élément. C'est sa présence qui assure la stabilité de la paire d'espaces. Elle est définie comme le produit des coordonnées barycentriques :

$$b(\xi, \eta, \zeta) = 256 \prod_{i=1}^4 \theta_i^1 = 256(1 - \xi - \eta - \zeta) \xi \eta \zeta$$

Elle est de plus conçue pour atteindre la valeur 1 au barycentre de l'élément. Par conséquent, son degré de liberté σ_b est défini comme :

$$\begin{aligned} \sigma_b : \mathbb{P}_1 \oplus \text{vect}(b) &\longrightarrow \mathbb{R} \\ p &\longmapsto p(\beta) - \sum_{i=1}^4 \frac{1}{4} p(p_i) \end{aligned}$$

en désignant par $\text{vect}(V)$ l'espace vectoriel engendré par V .

Formellement cet élément mixte peut être défini comme :

$$(\widehat{K}, [\mathbb{P}_1 \oplus \text{vect}(b)]^3, \widehat{\Sigma}_u) / (\widehat{K}, \mathbb{P}_1, \widehat{\Sigma}_p)$$

avec

$$\begin{aligned} \widehat{\Sigma}_u &= \{\sigma_{i,j} : p \in [\mathbb{P}_1 \oplus \text{vect}(b)]^3 \mapsto p(p_i) \cdot e_j, j = 1, 2, 3\} \cup \{\sigma_b \cdot e_j, j = 1, 2, 3\} \\ \widehat{\Sigma}_p &= \{\sigma_i : p \in \mathbb{P}_1 \mapsto p(p_i)\} \end{aligned}$$

en se référant à la figure 5.6 pour définir la position des points $\{p_i\}_i$ et où e_j représente la composante j de la base Cartésienne de l'espace Euclidien.

Les fonctions de base correspondant à cet élément sont donc définies comme

$$\left| \begin{array}{ll} u : & \phi_{i,j} = \theta_i^1 e_j \quad i = 0, \dots, 3; j = 0, 1, 2 \\ & \phi_{4,j} = b e_j \quad j = 0, 1, 2 \\ p : & \psi_i = \theta_i^1 \quad i = 0, \dots, 3 \end{array} \right.$$

L'élément de référence comprend donc un total de 19 degrés de liberté. Le même type de numérotation linéaire corrélative des composantes est employé pour ordonner les degrés de liberté correspondant à la vitesse : $k = 3 \times i + j$.

Pour un maillage composé de ces éléments, l'espace d'approximation engendré correspond à celui des fonctions continues dont la restriction à chaque élément du maillage est une combinaison linéaire d'un polynôme de degré 1 et de la bulle, un polynôme de degré 4 s'annulant sur toutes les faces de l'élément, pour chaque composante de la vitesse, et de degré 1 pour la pression.

L'intérêt de cet élément est d'employer moins de degrés de liberté que l'élément Taylor-Hood, tout en restant stable. En revanche, le taux de convergence de l'approximation vers la solution exacte en fonction de la résolution du maillage est plus faible.

5.4.5 Principe du calcul de la discrétisation

À supposer qu'un espace d'approximation ait été choisi, une base de fonctions est disponible pour effectuer la discrétisation. On s'intéresse donc au calcul effectif des quantités A_h, B_h et F_h telles qu'elles ont été définies à l'issue de la section 5.4.1. On verra d'abord comment ce calcul est décomposé sur chaque élément du maillage, pour faire apparaître des expressions à intégrer localement sur chaque élément, puis on détaillera comment sont réalisés ces calculs en pratique.

Principe de construction des matrices

Chaque élément des matrices à calculer nécessite d'évaluer une intégrale sur tout le domaine Ω . Le principe du calcul étant le même pour tous les éléments à calculer, on l'illustre en supposant que l'on doit calculer une matrice \bar{M}_h de terme général $[\bar{M}_h]_{i,j} = m(\bar{\varphi}_i, \bar{\varphi}_j) \in \mathcal{L}(V \times V, \mathbb{R})$ sur un domaine Ω maillé par \mathcal{T}_h .

Décomposition sur le maillage Pour effectuer le calcul, le maillage \mathcal{T}_h est mis à profit, et plus particulièrement sa propriété de constituer une partition du domaine Ω . Elle permet en effet de ramener l'évaluation d'une intégrale sur Ω à l'évaluation d'une série d'intégrales sur tous les éléments K du maillage \mathcal{T}_h :

$$[\bar{M}_h]_{i,j} = \int_{\Omega} m(\bar{\varphi}_i, \bar{\varphi}_j) \, dx = \sum_{K \in \mathcal{T}_h} \int_K m(\bar{\varphi}_i, \bar{\varphi}_j) \, dx$$

Le calcul est par conséquent effectué sur la base des éléments $K \in \mathcal{T}_h$ en introduisant les matrices élémentaires \bar{M}_h^K :

$$[\bar{M}_h^K]_{i,j} = \int_K m(\bar{\varphi}_i, \bar{\varphi}_j) \, dx$$

qui rassemblent toutes les évaluations relatives à l'élément K et qui sont assemblées dans la matrice globale \bar{M}_h .

$$\bar{M}_h = \sum_{K \in \mathcal{T}_h} \bar{M}_h^K$$

On note que les matrices locales \bar{M}_h^K mettent en jeu des opérateurs linéaires, ou bilinéaires, et que ces opérateurs sont appliqués aux fonctions de base $\bar{\varphi}$ dont le support est très restreint. Par conséquent, la majorité des intégrales sur K est nulle. Par construction, seules les fonctions de base associées à des nœuds

- M1 Calcul de la structure de la matrice globale \bar{M}_h
- M2 Allocation de matrice globale \bar{M}_h

- M3 Pour chaque élément $K \in \mathcal{T}_h$
 - M3.1 Calcul de la matrice élémentaire M_h^K
 - M3.2 Assemblage de M_h^K dans \bar{M}_h

Figure 5.8: Algorithme de construction des matrices.

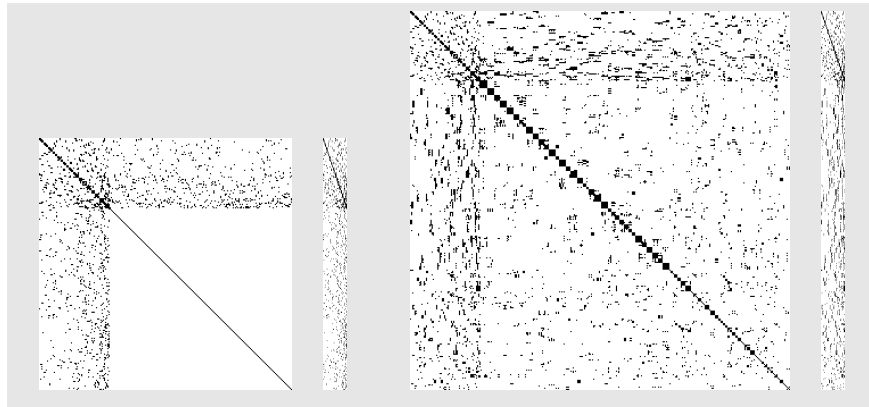


Figure 5.9: Motifs de remplissage des matrices pour le problème de Stokes. Les deux séries de matrices correspondent aux paires de matrices A et B pour un maillage non structuré, à gauche pour l'élément mini, et à droite pour l'élément de Taylor-Hood. Un point noir indique un élément potentiellement non-nul.

de l'élément K sont non-nulles sur K . Comme ces fonctions sont répertoriées pour chaque élément K , le calcul effectif des éléments de \bar{M}_h^K peut se concentrer directement sur les éléments non-nuls.

En tenant également compte des remarques liées à l'implémentation formulées ci-dessous, la construction des matrices s'opère alors en adoptant la démarche résumée dans la figure 5.8.

Mise en œuvre On soulignera deux points dans la mise en œuvre de cette procédure de construction des matrices.

Tout d'abord, on ne gère pas, dans la pratique, des matrices élémentaires \bar{M}_h^K de la dimension de \bar{M}_h durant le calcul : de petites matrices élémentaires denses M_h^K sont employées, qui regroupent uniquement les fonctions de base associées à K , c'est-à-dire toute l'information pertinente non-nulle. C'est d'ailleurs le calcul de ces matrices que l'on présente dans la suite. Dans ce cas, lors de l'assemblage, c'est-à-dire lors de l'addition à la matrice globale de cette matrice locale correspondant à un élément, il est nécessaire de mettre correctement en corrélation les éléments de cette matrice avec ceux de la matrice globale \bar{M}_h . Toutefois, aucune nouvelle information n'est requise, puisque cette corrélation est connue par la biais de la double numérotation, locale et globale, des degrés de liberté.

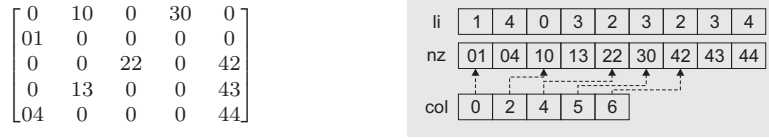


Figure 5.10: Format de stockage — *Compressed Column Storage*. Exemple de stockage de la matrice figurée à gauche. Le tableau nz stocke les valeurs de la matrice. Les valeurs $li[i]$ correspondent aux indices de ligne des éléments $nz[i]$, et col regroupe les positions de débuts de colonnes dans nz ou li .

Ensuite, la matrice \bar{M}_h est nécessairement très creuse, comme l'illustre la figure 5.9, ce qui oblige à adopter un format de stockage adapté. En effet,

$$[\bar{M}_h]_{i,j} = 0 \iff \exists K : \bar{\varphi}_i|_K \neq 0 \wedge \bar{\varphi}_j|_K \neq 0$$

Un élément (i, j) de la matrice est non-nul, si et seulement si, il existe un élément $K \in \mathcal{T}_h$ auquel les fonctions $\bar{\varphi}_i$ et $\bar{\varphi}_j$ sont toutes deux associées. Par construction, très peu d'éléments satisfont à ce critère.

Heureusement, la structure de la matrice est d'une part statique, et peut d'autre part être déterminée avant de commencer son remplissage, ce qui autorise le format de représentation de matrice creuse le plus performant, de type CCS — *Compressed Column Storage* — qui impose de connaître la structure de la matrice à l'avance [6].

Dans ce format, illustré dans la figure 5.10, tous les éléments non-nuls de la matrice sont compactés dans un tableau nz , suivant un balayage par colonne. Un deuxième tableau li , de même taille est également alloué, et stocke, pour chaque élément non-nul, son indice de ligne dans la matrice. Finalement, un tableau auxiliaire col regroupe les positions de début des colonnes de la matrice, dans le tableau des valeurs. Le coût d'une telle représentation est donc linéaire en nombre d'éléments non-nuls. Bien évidemment, aucune modification de la structure de la matrice n'est possible après son allocation. Comme dit précédemment, ce point n'est pas un problème puisque la structure de la matrice est statique et peut être calculée à l'avance. La structure de la matrice dépend à la fois de la topologie du maillage, et du type d'éléments employés.

Cette discussion sur la nature creuse du problème, n'a évidemment pas de sens dans le cas du calcul de F_h , le vecteur de second membre du système linéaire, qui est lui, dense. Le mode de calcul par calcul élémentaire puis assemblage n'est cependant pas remis en cause dans son traitement.

Calcul sur un élément de référence

Les calculs que l'on doit mettre en œuvre lors de la discrétisation sont des calculs d'intégration sur les domaines K des éléments finis du maillage, éléments tous engendrés par un élément de référence et une série de transformations T_K . C'est logiquement sur l'élément de référence que l'on ramène tous les calculs d'intégration car le domaine y est plus simple, et les fonctions de base y sont connues, comme on l'a vu lors de la présentation des éléments.

Transformation Par commodité, les transformations T_k sont habituellement exprimées par la donnée des positions $\{x_i\}_i$ de nœuds de l'élément réel, ainsi

que de fonctions d'interpolation de la déformation $\{\theta_i\}_i$ sur le domaine \widehat{K} . On reconnaîtra dans cette approche la donnée d'un élément fini géométrique de Lagrange. Dans ce cas, la déformation s'exprime comme :

$$T_K : \widehat{K} \longrightarrow K$$

$$\xi \longmapsto \sum_{i=1}^{n_{\text{geo}}} x_i \theta_i(\xi)$$

On remarquera que le degré des fonctions $\{\theta_i\}_i$ employées dans la déformation n'est pas contraint à être identique à celui employé pour l'approximation des fonctions dans les éléments. En particulier, dans ce travail, on choisit des transformations affines, c'est-à-dire que l'on considère uniquement les transformations induites par les déplacements des quatre sommets du tétraèdre de référence, avec une interpolation linéaire de la déformation induite sur \widehat{K} , et ce, même dans le cas de l'élément de Taylor-Hood.

Changement de variables Une fois explicité le modèle de transformation utilisé, on s'intéresse à la réalisation d'un calcul élémentaire d'intégrale du type :

$$\int_K m(\bar{\varphi}_i(x), \bar{\varphi}_j(x)) dx$$

En identifiant les restrictions des fonctions de base globales $\bar{\varphi}_i|_K$ et $\bar{\varphi}_j|_K$ aux fonctions de base locales de K :

$$\begin{aligned} \bar{\varphi}_i|_K &= \varphi_i && \text{sur } K \\ \bar{\varphi}_j|_K &= \varphi_j && \text{sur } K \end{aligned}$$

on peut écrire,

$$\int_K m(\bar{\varphi}_i(x), \bar{\varphi}_j(x)) dx = \int_K m(\varphi_i(x), \varphi_j(x)) dx$$

En employant le changement de variable $x \leftarrow \xi$ découlant de la transformation T_K , il est possible de réécrire l'intégrale sur le domaine de référence \widehat{K} :

$$\int_K m(\varphi_i(x), \varphi_j(x)) dx = \int_{\widehat{K}} m(\varphi_i \circ T_K(\xi), \varphi_j \circ T_K(\xi)) |\mathcal{J}(\xi)| d\xi$$

avec $\mathcal{J}(\xi)$ le déterminant de la matrice Jacobienne $J(\xi)$ de la transformation T_K :

$$\left| \begin{array}{l} T_K : \widehat{K} \longrightarrow K \\ \xi = \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} \longmapsto \begin{bmatrix} x(\xi, \eta, \zeta) \\ y(\xi, \eta, \zeta) \\ z(\xi, \eta, \zeta) \end{bmatrix} \end{array} \right.$$

$$\mathcal{J}(\xi) = |J(\xi)| = \left| \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \right| = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{vmatrix}$$

On note qu'un tel changement de variable contraint la transformation à être un \mathcal{C}^1 -difféomorphisme. On pose également que la transformation doit partout préserver l'orientation :

$$\mathcal{J}(\xi) > 0 \forall \xi \in \widehat{K}$$

Par définition de l'élément de référence et de la transformation T_K ,

$$\begin{aligned} T_K : \xi &\mapsto x \\ \varphi(x) &= \widehat{\varphi} \circ T_K^{-1}(x) \\ \varphi \circ T_K(\xi) &= \widehat{\varphi} \circ T_K^{-1} \circ T_K(\xi) = \widehat{\varphi}(\xi) \end{aligned}$$

ce qui permet de remplacer les fonctions de base $\{\varphi_i\}_i$ de l'élément K par les fonctions $\{\widehat{\varphi}_i\}_i$ associées à l'élément de référence \widehat{K} . On peut donc réécrire l'intégrale sous la forme finale que l'on utilise pour réaliser les évaluations :

$$\int_{\widehat{K}} m(\widehat{\varphi}_i(\xi), \widehat{\varphi}_j(\xi)) |J(\xi)| d\xi$$

qui n'emploie plus que des quantités de référence, en fonction des variables de référence.

On notera que le fait de se limiter à des transformations de type affine autorise deux simplifications : tout d'abord, la matrice Jacobienne est constante, et *a fortiori* son déterminant, donc $\mathcal{J}(\xi) = \mathcal{J}$. Ensuite, comme on fixe une condition de préservation de l'orientation à T_K , la contrainte $\mathcal{J} > 0$ revient à poser la seule contrainte que la numérotation des nœuds des éléments réels préserve l'orientation induite par la numérotation des nœuds de l'élément de référence.

Avec la numérotation des sommets de l'élément \mathbb{P}_1 représentée dans la figure 5.6, la matrice Jacobienne J , constante sur K , s'écrit :

$$J = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{bmatrix}$$

Changement d'opérateurs différentiels Dans le paragraphe précédent, on a vu comment ramener l'intégration sur un élément arbitraire à une intégration sur l'élément de référence. Cependant, certaines des expressions m à intégrer dans la méthode des éléments finis contiennent des opérateurs différentiels qui n'ont pas été touchés par le changement de variable, et qui se présentent toujours sous la forme

$$\nabla_x q(\xi) = \left[\frac{\partial q(\xi)}{\partial x}, \frac{\partial q(\xi)}{\partial y}, \frac{\partial q(\xi)}{\partial z} \right]^T$$

qui pose problème. Pour permettre l'intégration, on ramène ces opérateurs à des opérateurs différentiels où les dérivations sont effectuées par rapport aux coordonnées de référence ξ .

Pour cela, en considérant la dérivation d'une composition de fonctions, on sait que

$$\begin{bmatrix} \frac{\partial q}{\partial \xi} \\ \frac{\partial q}{\partial \eta} \\ \frac{\partial q}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \\ \frac{\partial q}{\partial z} \end{bmatrix}$$

Or la matrice qui apparaît n'est rien d'autre que la transposée de la matrice Jacobienne J de la transformation T_K , donc

$$\nabla_{\xi} q = J^T \cdot \nabla_x q$$

Par conséquent, il est possible d'exprimer toute dérivée Cartésienne *via* une combinaison linéaire des dérivées par rapport aux coordonnées de référence ξ en utilisant la relation :

$$\nabla_x q = J^{-T} \cdot \nabla_{\xi} q$$

La matrice J^{-T} existe nécessairement en raison de la contrainte posée sur T_K que $\mathcal{J} > 0$. Par commodité, on pose $[j] = [J]^{-T}$.

Mise en œuvre de l'intégration

À l'aide des techniques de réécriture d'intégrales précédemment décrites, on peut toujours ramener l'intégration à celle d'une fonction écrite en variables du domaine de référence, avec des opérateurs différentiels par rapport à ces mêmes variables de référence. Pour réaliser effectivement l'intégration, on a implémenté trois techniques.

Intégration analytique La première consiste à intégrer manuellement les expressions, en s'aidant éventuellement d'un logiciel de calcul formel. Cette approche est rendue possible par la restriction que l'on s'est fixée de ne considérer que des transformations affines de l'élément de référence. Dans ce cas, les matrices Jacobiennes et les déterminants afférents sont en effet constants, ce qui limite la complexité des expressions à des polynômes de faibles degrés.

Dans un cas de transformation non-affine, cette technique n'est pas réalisable car la matrice jacobienne devient non-constante, tout comme son déterminant, et l'expression devient trop complexe à manipuler analytiquement.

Cette approche manuelle est relativement prompte à introduire des erreurs, et pénible à mettre en œuvre dans les faits. Elle se trouve cependant être la plus rapide lors de l'exécution, puisque la grande majorité des calculs d'intégration est effectuée hors du code, et la construction d'une matrice se résume alors à évaluer une série d'expressions algébriques.

Intégration symbolique Une seconde approche, plus automatique, a donc été implémentée. Elle consiste toujours à réaliser l'intégration analytiquement, mais au lieu de réaliser cette opération manuellement, elle consiste à la faire réaliser symboliquement, de manière interne, par le code de calcul. Dans la mesure où les fonctions de base injectées dans les expressions sont polynomiales, la manipulation symbolique des expressions à calculer, polynomiales, reste simple.

Une fois l'expression à intégrer mise sous la forme d'un polynôme employant les variables de référence, l'intégration est conduite en employant la formule d'intégration analytique suivante connue sur l'élément de référence :

$$\int_{\widehat{K}} \xi^i \eta^j \zeta^k d\xi = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} \xi^i \eta^j \zeta^k d\zeta d\eta d\xi = \frac{i! j! k!}{(i+j+k+3)!}$$

En raison des limites de la manipulation des expressions polynomiales implémentée, cette approche requiert tout de même la détermination manuelle d'une

forme générale simple de l'expression à intégrer. En employant des systèmes symboliques plus puissants, cette approche peut toutefois prouver son intérêt par son automatisme.

Intégration numérique Finalement, on a implémenté le mode classique d'évaluation numérique des intégrales par quadrature [124]. Dans ce cas, une intégrale se trouve évaluée par une somme pondérée de valeurs prises par la fonction à intégrer en certains points du domaine :

$$\int_{\Omega} f(\xi) d\xi \approx \sum_k \omega_k \cdot f(\xi_k)$$

Dans le cas de fonctions f polynomiales, ce mode d'évaluation peut même être exact, si un nombre suffisant de points de quadrature est employé. Pour déterminer les poids $\{\omega_k\}_k$ et les points d'évaluation $\{\xi_k\}_k$, on se réfère aux règles standards tabulées, ou bien l'on dérive des règles de degré arbitraires en considérant un produit de règles de Gauss monodimensionnelles sur \hat{K} . Dans ce dernier cas, les règles obtenues ne sont toutefois pas optimales, en nombre de points utilisés ou encore en symétrie.

Cette méthode est la plus coûteuse des trois méthodes d'intégration, car elle nécessite rapidement de très nombreuses évaluations de la fonction à intégrer. En revanche, c'est la plus simple car il suffit de savoir évaluer une expression numériquement, c'est-à-dire simplement de la transcrire dans le code du solveur, pour pouvoir l'intégrer. C'est également la seule forme d'intégration disponible quand la forme de la fonction à intégrer n'est pas connue, par exemple dans le cas du second membre, où les forces externes peuvent éventuellement être choisies arbitraires.

5.4.6 Discrétisation des équations de Stokes

On s'intéresse maintenant à appliquer la démarche d'intégration précédemment décrite dans le cas général, aux termes issus des équations de Stokes.

Pour effectuer les calculs de discrétisation, on va distinguer deux cas, selon que des conditions de Dirichlet normales sont imposées ou non sur la vitesse, c'est-à-dire en fonction de la présence ou non de la frontière Γ_3 . Dans le cas où il est nécessaire d'imposer une condition de ce type, il faut que la discrétisation fasse explicitement apparaître des degrés de liberté pour la vitesse normale, dont il sera possible d'imposer les valeurs ultérieurement, lors de la résolution numérique du système linéaire recherchant la solution.

Comme une discrétisation faisant apparaître ces degrés de liberté particuliers peut être dérivée de la discrétisation où les degrés de liberté sont associés aux composantes Cartésiennes de la vitesse, on présentera au premier lieu la discrétisation Cartésienne, puis celle incorporant les conditions normales sur la vitesse.

Cas découplé

On s'intéresse ici au cas $\Gamma_3 = \emptyset$, c'est-à-dire que les seules conditions de Dirichlet imposées au problème le sont sur toutes les composantes de la vitesse. Dans ce cas, les degrés de liberté Cartésiens des éléments, précédemment décrits avec les éléments, sont adaptés pour imposer ces conditions.

On commence par rappeler que les fonctions de forme de l'espace d'approximation associées à la vitesse s'écrivent :

$$\phi_k = \phi_{a,b} = \gamma_a e_b$$

où γ_a , à valeur scalaire, donne la forme effective de la fonction et e_b représente la composante b de la base Cartésienne de l'espace Euclidien.

Pour obtenir A_h , on doit calculer des termes de la forme :

$$[A_h]_{i,j} = a(\phi_i, \phi_j) = \int_K [\nabla \phi_i] : [\nabla \phi_j] dx$$

et on rappelle que pour u , à valeur dans \mathbb{R}^3 :

$$[\nabla u]_{i,j} = \frac{\partial u \cdot e_j}{\partial x_i}$$

Étant donné que les fonctions de base ne sont non-nulles que sur une seule coordonnée :

$$[\nabla \gamma_a e_b]_{i,j} = \frac{\partial \gamma_a e_b \cdot e_j}{\partial x_i} = \frac{\partial \gamma_a}{\partial x_i} \delta_b^j = \begin{cases} 0 & \text{si } b \neq j \\ \frac{\partial \gamma_a}{\partial x_i} & \text{si } b = j \end{cases}$$

Si l'on revient sur le terme à évaluer, en posant $\phi_i = \gamma_a e_b$ et $\phi_j = \gamma_c e_d$,

$$\begin{aligned} [\nabla \phi_i] : [\nabla \phi_j] &= \sum_{k,l} \frac{\partial \gamma_a}{\partial x_k} \delta_b^l \frac{\partial \gamma_c}{\partial x_k} \delta_d^l \\ &= \sum_l \left[\left[\sum_k \frac{\partial \gamma_a}{\partial x_k} \frac{\partial \gamma_c}{\partial x_k} \right] \delta_b^l \delta_d^l \right] \\ &= \begin{cases} \sum_k \frac{\partial \gamma_a}{\partial x_k} \frac{\partial \gamma_c}{\partial x_k} & \text{si } b = d \\ 0 & \text{sinon} \end{cases} \end{aligned}$$

ce qui implique deux choses. D'abord, seuls les termes comportant deux fonctions de base opérant sur la même coordonnée sont non-nuls. Ensuite, le terme est identique pour toutes les coordonnées. Par conséquent, la matrice élémentaire \bar{A} possède une structure par blocs 3×3 de la forme

$$A_h = \begin{bmatrix} \bar{A}_{h;1,1} & \dots & \bar{A}_{h;1,N} \\ \vdots & & \vdots \\ \bar{A}_{h;N,1} & \dots & \bar{A}_{h;N,N} \end{bmatrix} \quad \text{avec} \quad \begin{cases} A_{h;i,j} = \tilde{A}_{i,j} I_3 \\ [\tilde{A}]_{i,j} = \int_K \nabla \gamma_i \cdot \nabla \gamma_j dx \end{cases}$$

en rappelant que les degrés de liberté des différentes composantes de la vitesse sont ordonnés de manière contiguë, et avec N le nombre de fonctions de formes $\{\gamma_i\}_i$ différentes pour une composante de la vitesse dans l'élément — $N = 10$ pour l'élément de Taylor-Hood, et $N = 5$ pour l'élément mini — En fait, on a là une discrétisation découplée du Laplacien pour les trois composantes de la vitesse, et l'on a trois fois le même problème de Laplacien monodimensionnel, pour les trois composantes de la vitesse.

Pour effectuer l'intégration, on rapporte maintenant cette intégrale sur l'élément de référence, et on fait apparaître les fonctions de base définies en variables de référence :

$$\begin{aligned} \int_K \nabla \gamma_i \cdot \nabla \gamma_j \, dx &= \int_{\widehat{K}} \nabla_x \widehat{\gamma}_i \cdot \nabla_x \widehat{\gamma}_j \, \mathcal{J} d\xi \\ &= \int_{\widehat{K}} (j \nabla_\xi \widehat{\gamma}_i)^T \cdot (j \nabla_\xi \widehat{\gamma}_j) \, \mathcal{J} d\xi \\ &= \int_{\widehat{K}} (\nabla_\xi \widehat{\gamma}_i)^T j^T j (\nabla_\xi \widehat{\gamma}_j) \, \mathcal{J} d\xi \end{aligned}$$

Dans le cas où l'on désire employer une intégration analytique, manuelle ou intégrée au code, on réécrit ce terme sous la forme

$$\begin{aligned} \widetilde{A}_{i,j} &= \sum_{1 \leq s,t \leq 3} M_{s,t} \int_{\widehat{K}} \frac{\partial \widehat{\gamma}_i}{\partial \xi_s} \frac{\partial \widehat{\gamma}_j}{\partial \xi_t} \, \mathcal{J} \, d\xi && \text{avec } M = j^T j \, \mathcal{J} \\ &= \sum_{1 \leq s,t \leq 3} M_{s,t} K_{s,t}^{i,j} && \text{avec } K_{s,t}^{i,j} = \int_{\widehat{K}} \frac{\partial \widehat{\gamma}_i}{\partial \xi_s} \frac{\partial \widehat{\gamma}_j}{\partial \xi_t} \, d\xi \end{aligned}$$

qui permet d'isoler la transformation des fonctions de base.

Cette expression peut ensuite être évaluée selon n'importe laquelle des techniques d'intégration décrites précédemment. À titre de référence, on donne une expression analytique de $K_{s,t}^{i,j}$ pour l'élément mini dans la figure 5.11.

On procède selon une démarche similaire pour le calcul de la matrice \bar{B}_h de terme général

$$[B_h]_{i,j} = b(\phi_i, \psi_j) = \int_K \psi_j \nabla \cdot \phi_i \, dx$$

Comme dans le cas précédent, les fonctions de base de la vitesse correspondent aux composantes Cartésiennes, donc :

$$\nabla \cdot \phi_i = \nabla \cdot (\gamma_a e_b) = \sum_k \frac{\partial \gamma_a e_b \cdot e_k}{\partial x_k} = \sum_k \frac{\partial \gamma_a}{\partial x_k} \delta_b^k = \frac{\partial \gamma_a}{\partial x_b}$$

Donc :

$$\int_K \psi_j \nabla \cdot \phi_i \, dx = \int_K \psi_j \frac{\partial \gamma_a}{\partial x_b} \, dx$$

On rapporte ensuite l'intégrale à calculer sur le domaine de référence et on introduit les fonctions de base de référence :

$$\int_K \psi_j \frac{\partial \gamma_a}{\partial x_b} \, dx = \int_{\widehat{K}} \widehat{\psi}_j \frac{\partial \widehat{\gamma}_a}{\partial x_b} \, \mathcal{J} \, d\xi$$

Pour faire apparaître les opérateurs différentiels en variables de référence, on réécrit le terme comme

$$\begin{aligned} \int_{\widehat{K}} \widehat{\psi}_j \frac{\partial \widehat{\gamma}_a}{\partial x_b} \, \mathcal{J} \, d\xi &= \int_{\widehat{K}} \widehat{\psi}_j e_b \cdot (j \nabla_\xi \widehat{\gamma}_a) \, \mathcal{J} \, d\xi \\ &= \mathcal{J} e_b \cdot j L_s^{j,a} && \text{avec } L_s^{j,a} = \int_{\widehat{K}} \widehat{\psi}_j \frac{\partial \widehat{\gamma}_a}{\partial \xi_s} \, d\xi \end{aligned}$$

Dans le cas d'un élément mini, $K_{s,t}^{i,j}$ correspond à :

$$\begin{bmatrix} \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} & \begin{bmatrix} -\frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} -\frac{1}{6} & 0 & 0 \\ -\frac{1}{6} & 0 & 0 \\ -\frac{1}{6} & 0 & 0 \end{bmatrix} & \begin{bmatrix} \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{6} & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & -\frac{1}{6} & 0 \\ 0 & -\frac{1}{6} & 0 \\ 0 & -\frac{1}{6} & 0 \end{bmatrix} & \begin{bmatrix} 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & -\frac{1}{6} \\ 0 & 0 & -\frac{1}{6} \\ 0 & 0 & -\frac{1}{6} \end{bmatrix} & \begin{bmatrix} 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{6} \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 4096 & 2048 & 2048 \\ 2835 & 2835 & 2835 \\ 2048 & 4096 & 2048 \\ 2835 & 2835 & 2835 \\ 2048 & 2048 & 4096 \\ 2835 & 2835 & 2835 \end{bmatrix} \end{bmatrix}$$

écrite sous la forme d'une matrice de matrices $[[K]_{s,t}(i,j)]_{i,j}$.
 Calculée analytiquement et écrite sous la forme $L_s^{j,a} = [[L]_s(a,j)]_{a,j}$, $L_s^{j,a}$ correspond à :

$$\begin{bmatrix} \begin{bmatrix} -\frac{1}{24} \\ -\frac{1}{24} \\ -\frac{1}{24} \end{bmatrix} & \begin{bmatrix} -\frac{1}{24} \\ -\frac{1}{24} \\ -\frac{1}{24} \end{bmatrix} & \begin{bmatrix} -\frac{1}{24} \\ -\frac{1}{24} \\ -\frac{1}{24} \end{bmatrix} & \begin{bmatrix} -\frac{1}{24} \\ -\frac{1}{24} \\ -\frac{1}{24} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{24} \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} \frac{1}{24} \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} \frac{1}{24} \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} \frac{1}{24} \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ \frac{1}{24} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \frac{1}{24} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \frac{1}{24} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \frac{1}{24} \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ \frac{1}{24} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ \frac{1}{24} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ \frac{1}{24} \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ \frac{1}{24} \end{bmatrix} \\ \begin{bmatrix} \frac{315}{16} \\ \frac{16}{315} \\ \frac{16}{315} \end{bmatrix} & \begin{bmatrix} -\frac{16}{315} \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ -\frac{16}{315} \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ -\frac{16}{315} \end{bmatrix} \end{bmatrix}$$

Figure 5.11: Données nécessaires au calcul des matrices de l'élément mini, calculées analytiquement.

Il reste ensuite à réaliser l'intégration proprement dite par l'un des moyens proposés. La figure 5.11 contient $L_s^{j,a}$ calculée analytiquement, pour le cas de l'élément mini.

Dans le cas du vecteur F_h , la seule transformation particulière à réaliser est de ramener l'intégration sur l'élément de référence. Si la force est de forme inconnue, la seule technique disponible consiste à approcher l'intégrale numériquement par quadrature. Si la force est connue explicitement, par exemple parce qu'elle est interpolée entre des valeurs aux sommets du maillage, il est possible de dégager une expression analytique, comme on l'a fait pour les autres matrices. La situation est toutefois plus simple car dans ce cas aucun opérateur différentiel n'intervient, et les termes à évaluer sont simplement de la forme

$$[F_h]_i = \int_K f \cdot \phi_i \, dx = \int_{\hat{K}} [f \circ T_K] \cdot \hat{\phi}_i \, d\xi$$

Cas couplé

Dans le cas où les fonctions de base correspondent aux composantes Cartésiennes, un certain nombre de simplifications ont rendu possible le découplage des différentes composantes de la vitesse dans le problème du Laplacien. Cependant, dans le cas où des conditions de Dirichlet normales sont imposées sur la vitesse, c'est-à-dire sur une seule des composantes, les fonctions de base ne peuvent plus être choisies pour être alignées avec les axes du repère mais doivent être choisies pour faire explicitement apparaître la vitesse normale dans le calcul.

Base choisie Dans ce cas, on suppose connue, en chaque sommet $q_i \in \Gamma_{3h}$, la discrétisation de Γ_3 , une base de l'espace $(n, \tau_1, \tau_2)(q_i)$ orthonormée et directe. On verra ultérieurement comment construire la normale à partir du maillage.

On construit alors la famille de fonctions $\{\Phi_{i,j}\}_{i,j}$ en remplaçant les fonctions Cartésiennes associées à la frontière Γ_3 par des fonctions localement orientées avec les normales aux points $q_i \in \Gamma_{3h}$. On rappelle qu'à n'importe quel nœud du maillage q_i sont associées les fonctions $\{\phi_{i,j}\}_j$.

On définit ensuite les fonctions de bases $\Phi_{i,i}$ associées au nœuds q_i comme suit :

$$\left\{ \begin{array}{l} \begin{bmatrix} \Phi_{i,0} \\ \Phi_{i,1} \\ \Phi_{i,2} \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z \\ \tau_{1x} & \tau_{1y} & \tau_{1z} \\ \tau_{2x} & \tau_{2y} & \tau_{2z} \end{bmatrix} \cdot \begin{bmatrix} \phi_{i,0} \\ \phi_{i,1} \\ \phi_{i,2} \end{bmatrix} \\ \Phi_{i,j} = \phi_{i,j} \end{array} \right. \quad \text{si } q_i \in \Gamma_{3h} \quad \text{sinon}$$

ce qui correspond à effectuer un changement de repère entre la base canonique et le repère $(n, \tau_1, \tau_2)(q_i)$ pour les fonctions de base associées à Γ_3 . Les autres fonctions ne sont pas touchées. Avec ce choix de base, les fonctions $\{\Phi_{i,0}\}_{q_i \in \Gamma_{3h}}$ représentent directement les vitesses normales à la frontière sur Γ_3 , ce qui permet d'appliquer les conditions aux limites imposées durant la résolution numérique.

Modification des matrices Modifier ainsi les fonctions de base nécessite de calculer les matrices avec ces fonctions. Pour ce faire on s'appuie sur les calculs déjà réalisés et sur le fait que les fonctions $\{\Phi_i\}_i$ sont liées aux fonctions $\{\phi_i\}_i$ par un changement de base. En particulier, en nommant la matrice de

changement de base X_i associée à q_i :

$$X_i = \begin{bmatrix} n_x & n_y & n_z \\ \tau_{1x} & \tau_{1y} & \tau_{1z} \\ \tau_{2x} & \tau_{3y} & \tau_{3z} \end{bmatrix}$$

on peut exprimer la matrice A avec cette nouvelle base en modifiant la construction de chacun des blocs de la manière suivante :

$$A_{i,j} = \begin{cases} \tilde{A}_{i,j} I_3 & \text{si } q_i \notin \Gamma_{3h} \wedge q_j \notin \Gamma_{3h} \\ \tilde{A}_{i,j} X_i & \text{si } q_i \in \Gamma_{3h} \wedge q_j \notin \Gamma_{3h} \\ \tilde{A}_{i,j} X_j^T & \text{si } q_i \notin \Gamma_{3h} \wedge q_j \in \Gamma_{3h} \\ \tilde{A}_{i,j} X_i X_j^T & \text{si } q_i \in \Gamma_{3h} \wedge q_j \in \Gamma_{3h} \end{cases}$$

Ces relations découlent directement de la bilinéarité de l'opérateur à discrétiser. On notera que si l'on pose $X_i = I_3$ si $q_i \notin \Gamma_{3h}$, on peut généraliser la construction des blocs de la matrice comme

$$A_{i,j} = \tilde{A}_{i,j} X_i X_j^T$$

On notera également que dans ce cas, la discrétisation des différentes composantes du Laplacien est couplée, et que l'on a affaire à un réel problème de Laplacien tridimensionnel.

Il faut procéder de même pour la matrice B_h , mais dans ce cas, seule une des deux familles de fonctions de base est modifiée. En regardant la matrice $B_h^\phi = B_h$ que l'on a précédemment calculée avec une structure de blocs 3×1 :

$$B_h^\phi = B_h = \begin{bmatrix} B_{h;1,1}^\phi & \cdots & B_{h;1,M}^\phi \\ \vdots & & \vdots \\ B_{h;N,1}^\phi & \cdots & B_{h;N,M}^\phi \end{bmatrix} \quad B_{h;i,j}^\phi \in \mathcal{M}_{3,1}$$

on peut exprimer la matrice B_h^Φ associée aux fonctions $\{\Phi_i\}_i$, de même structure, en calculant chaque bloc par la relation :

$$B_{h;i,j}^\Phi = X_i B_{h;i,j}^\phi$$

De même pour le vecteur F_h qui compose le second membre :

$$F_h^\phi = F_h = \begin{bmatrix} F_{h;1}^\phi \\ \vdots \\ F_{h;N}^\phi \end{bmatrix} \quad F_{h;i}^\phi \in \mathcal{M}_{3,1}$$

et qui se trouve transformé par la relation

$$F_{h;i}^\Phi = X_i F_{h;i}^\phi$$

Construction de la normale Pour construire les fonctions de base alignées avec les normales, on a évidemment besoin de connaître ces normales. Pour les

calculer, on reprend la construction employée par le code NSP1B3 [67], destinée à obtenir des normales qui satisfont à la propriété

$$\int_{\Gamma_h} v_h \cdot n_h = 0 \quad \forall v_h \in V_h \text{ tel que } b(u_h, q_h) = 0 \quad \forall q_h \in Q_h$$

analogue à la formule de Green

$$\int_{\Gamma} v \cdot n = 0 \quad \forall v \text{ tel que } \nabla \cdot v = 0$$

Un moyen d'obtenir de telles normales est de les définir comme

$$n_{i,j} = \frac{1}{C} \int_{\Omega} \frac{\partial \phi_i}{\partial x_j} dx$$

où $n_{i,j}$ représente la composante j de la normale associée au point q_i , ϕ_i est la fonction de base associée au nœud q_i , et C est une constante destinée à normaliser le vecteur n_i .

5.5 Résolution des systèmes linéaires

La discrétisation du problème de Stokes grâce à la méthode des éléments finis aboutit au système linéaire

$$\begin{bmatrix} A_h & B_h \\ B_h^T & 0 \end{bmatrix} \cdot \begin{bmatrix} U_h \\ P_h \end{bmatrix} = \begin{bmatrix} F_h \\ 0 \end{bmatrix}$$

qui est caractéristique d'un problème de point-selle.

Au vu de la taille des problèmes, qui peuvent aisément atteindre plusieurs dizaines de milliers d'inconnues, il est impossible d'inverser la matrice. Il est donc nécessaire de chercher une solution par des méthodes de résolution itératives, comme la méthode du gradient conjugué [6, 131] ou la méthode GMRES [131]. On reviendra sur la résolution proprement dite du système après avoir d'abord discuté la prise en compte des conditions aux limites dans la résolution.

5.5.1 Introduction des conditions aux limites

Pour tenir compte des conditions aux limites de Dirichlet à imposer sur la vitesse, il faut enlever de la famille des fonctions de base employée $\{\phi_i\}_i$ ou $\{\Phi_i\}_i$, les fonctions qui correspondent à des composantes imposées. Pour un nœud $q_i \in \Gamma_{1h}$, cela revient à éliminer les trois fonctions $\{\phi_{i,j}\}_{j=0,1,2}$. Pour un nœud $q_i \in \Gamma_{3h}$, seule la fonction $\Phi_{i,0}$ doit être supprimée. Aux équations ainsi perdues, on substitue toutes les équations de la forme $u_i = u_{D,i}$ correspondant aux conditions aux limites définies sur chaque composante contrainte. On notera que l'on a exactement construit les bases des espaces d'approximation pour pouvoir ainsi représenter directement ces composantes imposées et opérer la substitution décrite.

Du point de vue de la mise en œuvre pratique, il n'est pas question de modifier la création des matrices elles-mêmes pour tenir compte de cette restriction des bases. On applique en fait directement cette démarche au niveau discret, sur les matrices. Pour cela, on notera tout d'abord que les conditions à imposer

ne le sont que sur la vitesse, et non sur la pression. Pour la commodité de la présentation, on va permuter les degrés de liberté afin de former deux groupes distincts : celui des degrés de liberté non-contraints, dont les indices sont compris entre 0 et P , et ceux contraints, dont les indices sont compris entre P et N , N étant le nombre total de degrés de liberté de la vitesse. En permutant le contenu du système à résoudre de manière analogue, la structure par blocs suivante se dégage alors du système :

$$\begin{bmatrix} A_{h;1,1} & A_{h;1,2} & B_{h;1} \\ A_{h;1,1} & A_{h;1,2} & B_{h;2} \\ B_{h;1}^T & B_{h;2}^T & 0 \end{bmatrix} \cdot \begin{bmatrix} U_{h;1} \\ U_{h;2} \\ P_h \end{bmatrix} = \begin{bmatrix} F_{h;1} \\ F_{h;2} \\ 0 \end{bmatrix}$$

avec par définition le bloc $U_{h;2}$ qui correspond aux variables à contraindre. Puisque les valeurs de toutes ces variables sont connues *a priori* par le biais des conditions aux limites, on peut les introduire dans le système, qui se réécrit alors comme :

$$\begin{bmatrix} A_{h;1,1} & 0 & B_{h;1} \\ 0 & I & 0 \\ B_{h;1}^T & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} U_{h;1} \\ U_{h;2} \\ P_h \end{bmatrix} = \begin{bmatrix} F_{h;1} - A_{h;1,2} \cdot U_D \\ U_D \\ -B_{h;2}^T \cdot U_D \end{bmatrix}$$

où le vecteur U_D regroupe les valeurs connues des conditions aux limites.

Dans la pratique, il n'est absolument pas nécessaire de permuter la matrice pour opérer cette modification : elle peut être réalisée dans le système original, la permutation n'étant effectuée que pour la clarté de la présentation. Par son action, cette modification du système correspond exactement à la contrepartie discrète de la prise en compte des conditions aux limites au niveau du problème continu, en ramenant le problème non-homogène à un problème homogène.

Les conditions aux limites introduites, on peut replaquer la structure du système originel sur ce nouveau système en posant

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \cdot \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}$$

avec

$$A = \begin{bmatrix} A_{h;1,1} & 0 \\ 0 & I \end{bmatrix} \quad B = \begin{bmatrix} B_{h;1} \\ 0 \end{bmatrix} \quad F = \begin{bmatrix} F_{h;1} - A_{h;1,2} \cdot U_D \\ U_D \end{bmatrix} \quad G = -B_{h;2}^T \cdot U_D$$

pour retrouver la structure de discrétisation d'un problème de point-selle.

On peut noter que la prise en compte des conditions aux limites n'altère pas les propriétés de la matrice A par rapport à la matrice A_h : une matrice symétrique reste symétrique, et une matrice définie positive reste définie positive.

Pour résoudre ce système, on a le choix entre plusieurs techniques en fonction du problème considéré.

5.5.2 Résolution du système mixte

La première approche est de résoudre directement le système mixte mettant en jeu la vitesse et la pression. Le problème de cette stratégie est que la matrice mixte $\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}$, si elle est éventuellement symétrique, n'est pas définie positive. Ceci empêche alors d'employer la méthode du gradient conjugué pour résoudre le système, et il est nécessaire de recourir à la méthode GMRES, plus coûteuse.

Dans le cas d'un système non-symétrique, *i.e.* si A est non symétrique, cette résolution du système mixte reste toutefois la seule technique que l'on ait expérimenté. Ce cas non symétrique survient si l'on ne résout pas le problème de Stokes, mais celui d'Oseen dans le but de résoudre les équations de Navier-Stokes. Dans ce cas, la forme bilinéaire a discrétisée pour obtenir A_h n'est en effet plus :

$$a(u, v) = \nu \int_{\Omega} \nabla u : \nabla v \, dx$$

mais

$$a(u, v) = \nu \int_{\Omega} \nabla u : \nabla v \, dx + \int_{\Omega} [(w \cdot \nabla)u] \cdot v \, dx - \frac{1}{\delta t} \int_{\Omega} u \cdot v \, dx$$

et se trouve donc être non-symétrique. Pour le problème d'Oseen, l'inclusion du dernier terme est conditionnée au fait de résoudre le problème stationnaire ou non de Navier-Stokes.

Pour le cas d'un traitement par caractéristiques de la discrétisation en temps des équations de Navier-Stokes, la forme bilinéaire a s'apparente à :

$$a(u, v) = \nu \int_{\Omega} \nabla u : \nabla v \, dx + \frac{1}{\delta t} \int_{\Omega} u \cdot v \, dx$$

et la matrice A_h reste dans ce cas symétrique et définie positive.

5.5.3 Méthode d'Uzawa

Dans le cas d'une matrice A symétrique, qui découle du problème de Stokes ou du traitement par caractéristiques de la non-linéarité du problème de Navier-Stokes, il est cependant possible d'utiliser une autre méthode de résolution : celle d'Uzawa.

Si l'on considère le système à résoudre :

$$\begin{cases} AU + BP = F \\ B^T U = G \end{cases}$$

on peut, grâce à la première équation, exprimer U en fonction de P par :

$$U = A^{-1}[F - BP]$$

En insérant cette expression dans la deuxième équation, il vient :

$$\begin{aligned} B^T A^{-1}[F - B \cdot P] &= G \\ \iff [B^T A^{-1} B]P &= B^T A^{-1} F - G \\ \iff MP &= B^T A^{-1} F - G \end{aligned}$$

ce qui fait apparaître un système linéaire réduit, mettant uniquement en jeu la pression P , avec la matrice $M = B^T A^{-1} B$.

La méthode d'Uzawa consiste donc à résoudre ce système réduit sur la pression. Bien qu'il puisse être résolu par n'importe quelle méthode, ce système présente une caractéristique très intéressante : si la matrice A est symétrique définie positive, M l'est également, ce qui autorise à employer la méthode de gradient conjugué pour résoudre le système.

Si la méthode paraît simple au premier abord, il faut cependant tenir compte du fait que la matrice M ne peut être calculée explicitement, et ce pour deux raisons. La première est qu'elle est pleine, et que sa dimension — égale au nombre de degrés de liberté en pression uniquement — même réduite par rapport à la taille du problème initial, ne la rend pas calculable. La seconde est qu'un calcul explicite requerrait une inversion de la matrice A , ce qui n'est pas envisageable.

Pour pallier ce problème, il est nécessaire de considérer la matrice M sous la forme implicite de sa définition $M = B^T A^{-1} B$, dans l'algorithme de résolution itératif employé pour résoudre le système mettant en jeu la pression. Quel que soit le choix de l'algorithme itératif employé, pour résoudre le système $Ax = b$, les méthodes itératives requièrent dans leur déroulement d'effectuer un produit matrice-vecteur de la forme $q = Ap$. Dans le cadre de la méthode d'Uzawa, cette opération équivaut à calculer $q = B^T A^{-1} Bp$. À chaque itération de l'algorithme de résolution, le calcul de ce produit est donc décomposé en trois opérations :

1. Calcul de $q_1 = Bp$
2. Résolution de $Aq_2 = q_1$
3. Calcul de $q = B^T q_2$

Ceci signifie qu'il est nécessaire de résoudre un système linéaire mettant A en jeu à chaque itération de l'algorithme de résolution itératif de la pression. On peut employer n'importe quelle méthode pour la résolution de A mais on préférera généralement celle du gradient conjugué préconditionné, étant donné sa nature symétrique définie positive. Le préconditionnement que l'on a choisi dans ce cas est un préconditionnement SSOR, qui est simple à mettre en œuvre et ne requiert aucun stockage significatif supplémentaire.

La totalité de la résolution par la méthode d'Uzawa est résumée dans la figure 5.12. L'algorithme ne demande le stockage d'aucune matrice supplémentaire par rapport aux matrices originales du système, et ne requiert par conséquent pas plus d'espace-mémoire.

La résolution par la méthode d'Uzawa revient donc à une double résolution itérative imbriquée. Cette méthode reste par conséquent coûteuse, mais se trouve parfois plus performante qu'une résolution directe du système mixte par la méthode GMRES. On a, par contre, constaté qu'elle est également moins robuste en ce qui concerne la pression, et qu'elle ne parvient pas toujours à bien converger vers la solution recherchée.

La comparaison entre une résolution directe du système mixte et la méthode d'Uzawa est difficile, les deux n'opérant pas sur le même système linéaire. Implémentées sans préconditionnement, elles ne parviennent pas à converger de manière fiable vers les solutions recherchées, ce qui ne permet pas de les comparer avec rigueur, que l'étude porte sur la précision des résultats, ou sur la vitesse de convergence.

5.6 Validation et premiers résultats

Vont maintenant être présentés les premiers résultats que l'on a pu obtenir avec le solveur que l'on a implémenté, fondé sur la méthode de résolution par éléments finis précédemment décrite. Il s'agit principalement de résultats de validations numériques, dont le rôle est de juger de la correction de l'implantation. Il ne faut donc pas encore attendre d'images visuellement probantes, et encore

Pour résoudre le système

$$\begin{cases} AU + BP = F \\ B^T U = G \end{cases}$$

par la méthode d'Uzawa :

- 1 Calcul du second membre $R = B^T A^{-1} F - G$ du système réduit
 - 1.1 Résolution de $As = F$
 - 1.2 Calcul de $R = B^T s - G$

- 2 Résolution itérative du système réduit $MP = R$

Quand il est nécessaire de calculer $q = Mp$

 - 2.1 Calcul de $q_1 = Bp$
 - 2.2 Résolution itérative de $Aq_2 = q_1$, éventuellement avec préconditionnement
 - 2.3 Calcul de $q = B^T q_2$

- Une fois P calculé
 - 3 Calcul de $U = A^{-1}[F - BP]$
 - 3.1 Calcul de $Q = F - BP$
 - 3.2 Résolution itérative de $AU = Q$

Figure 5.12: Méthode d'Uzawa pour la résolution numérique du système linéaire résultant de la discrétisation d'un problème de point-selle.

moins d'animations de fluides comme on a pu en présenter dans le chapitre 3 dédié à la simulation MAC. Les travaux réalisés constituent néanmoins un premier pas vers cet objectif.

Le rôle de ces validations numériques rigoureuses du code implémenté est primordial, car c'est le seul et unique moyen disponible pour mettre au point le solveur. En effet, comme on a pu s'en apercevoir par la description qui en a été faite, la résolution par éléments finis est la confluence de la résolution d'une vaste gamme de sous-problèmes susceptibles d'erreurs.

Problème de Stokes Comme on l'a décrit dans le début de ce chapitre, la simulation des équations de Navier-Stokes se transpose en la résolution d'une succession de problèmes de Stokes. On valide ici numériquement le solveur du problème de Stokes que l'on implémenté.

Le principe des validations numériques effectuées est le suivant : on choisit une solution analytique au problème de Stokes (u, p) , de laquelle on dérive les données nécessaires pour poser un problème au simulateur, c'est-à-dire les conditions aux limites et le second-membre f des équations. Connaissant ainsi un problème de Stokes et sa solution analytique, il est possible de comparer solution analytique et solutions approchées au problème calculées par le simulateur.

On rappelle toutefois que le problème discret que résout numériquement le simulateur ne correspond pas à la recherche de la solution exacte, mais à la recherche d'une approximation dont la proximité avec la solution est liée au

maillage du domaine. La validation repose donc sur l'étude de la convergence de la solution approchée vers la solution analytique, pour une famille de maillages $\{\mathcal{T}_h\}_{h>0}$ progressivement raffinés. La convergence de la solution fournie par le simulateur est alors comparée au taux de convergence théorique dépendant de l'élément choisi, qui se présente sous la forme

$$\begin{aligned} |u_h - u|_{L_2(\Omega)} &\leq c_1 h^{k_1} & c_1 > 0 \\ |p_h - p|_{L_2(\Omega)} &\leq c_2 h^{k_2} & c_2 > 0 \end{aligned}$$

où (u, p) représente la solution exacte au problème et (u_h, p_h) représente la solution approchée associée au maillage \mathcal{T}_h de finesse h . Les constantes c_1 et c_2 sont alors uniquement dépendantes du domaine Ω . La norme $L^2(\Omega)$ employée pour la mesure est définie comme :

$$\begin{aligned} |u_h - u|_{L_2(\Omega)} &= \left[\sum_{i=1}^d \int_{\Omega} [(u_h - u) \cdot e_i]^2 dx \right]^{\frac{1}{2}} \\ |p_h - p|_{L_2(\Omega)} &= \left[\int_{\Omega} (p_h - p)^2 dx \right]^{\frac{1}{2}} \end{aligned}$$

avec d la dimension de l'espace géométrique considéré.

Dans la section précédente, seuls les éléments tétraédriques mini et de Taylor-Hood ont été présentés en détails. Nous avons également implémenté, exactement sur le même principe, des éléments triangulaires et quadrangulaires en dimension deux, ainsi que des éléments hexaédriques en dimension trois, tous de Taylor-Hood, ainsi que des quadrangles mini. La raison de cette multiplication des éléments est de permettre une validation plus complète.

Le cas-test de référence sur lequel on s'appuie est celui proposé par Bercovier et Engelman [11], défini comme suit pour un domaine bidimensionnel :

$$\left\{ \begin{array}{l} v(x, y) = -256 x^2 (x - 1)^2 y (y - 1) (2y - 1) \\ u(x, y) = \begin{bmatrix} v(x, y) \\ -v(y, x) \end{bmatrix} \\ p(x, y) = \left[x - \frac{1}{2} \right] \cdot \left[y - \frac{1}{2} \right] \\ f(x, y) = \begin{bmatrix} -\nu v(x, y) + \left[y - \frac{1}{2} \right] \\ \nu v(y, x) + \left[x - \frac{1}{2} \right] \end{bmatrix} \end{array} \right.$$

Dans cet exemple, le champ de vitesse tourne autour du point $[\frac{1}{2}, \frac{1}{2}]$ et s'annule sur les bords du domaine $[0, 1]^2$. On l'a extrudé le long de la troisième coordonnée pour le cas d'un domaine tridimensionnel, comme illustré par la figure 5.13.

On présente les résultats que l'on peut obtenir avec ces différents éléments dans les figures 5.14 à 5.16. Pour des éléments de Taylor-Hood, l'erreur attendue est de l'ordre de $o(h^3)$ pour la vitesse et $o(h^2)$ pour la pression, et $o(h^2), o(h^2)$ pour l'approximation mini. Comme on peut le constater d'après les différentes mesures, le simulateur fournit des approximations convergentes lorsque la grille est raffinée. La convergence est généralement de la forme attendue, ce qui permet de supposer que l'implantation réalisée est correcte.

Pour les éléments de Taylor-Hood, les taux de convergence théoriques sont généralement atteints. Ils sont même parfois dépassés si les maillages de test

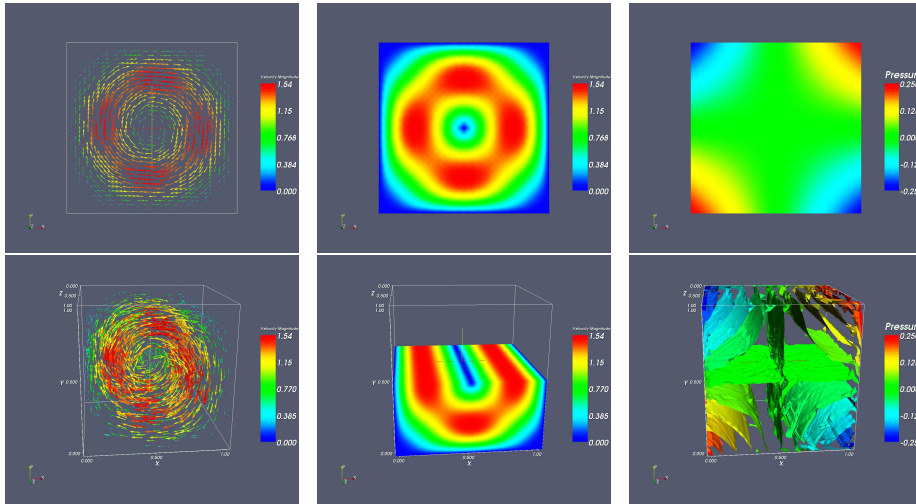


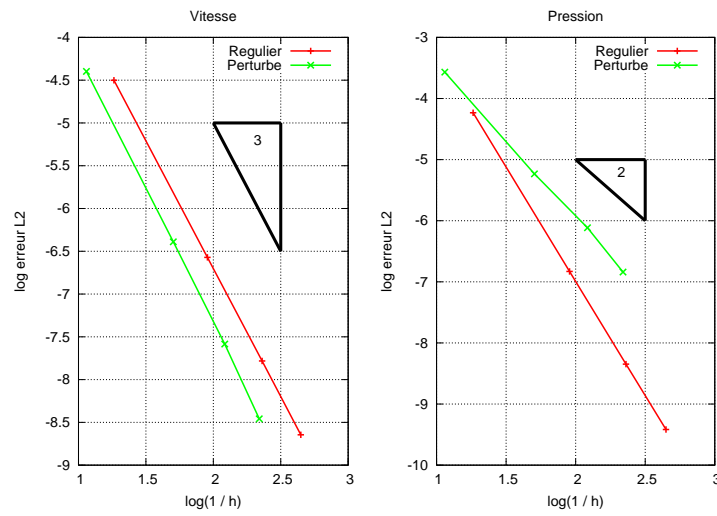
Figure 5.13: Solutions trouvées pour le problème-test proposé par Bercovier et Engelman [11]. Le test tridimensionnel correspond au test bidimensionnel extrudé. La première série d'images illustre une solution obtenue sur un maillage en quadrangles, la seconde sur un maillage en tétraèdres, avec des éléments de Taylor-Hood. Pour le test tridimensionnel, la pression est figurée par ses isosurfaces, et une coupe médiane du volume est représentée pour la vitesse.

employés sont structurés. Du fait de leur structure particulière, une forme de super-convergence peut en effet survenir lors de leur emploi. En revanche, le cas des éléments mini se révèle plus problématique, pour la convergence de la pression, celle-ci étant en effet plus faible que prévue. On peut également noter que l'erreur commise est sensiblement moins importante pour les problèmes 2D que pour les problèmes 3D, et qu'il est plus difficile de calculer avec exactitude la pression, l'erreur associée à la pression étant sensiblement plus grande que celle associée à la vitesse.

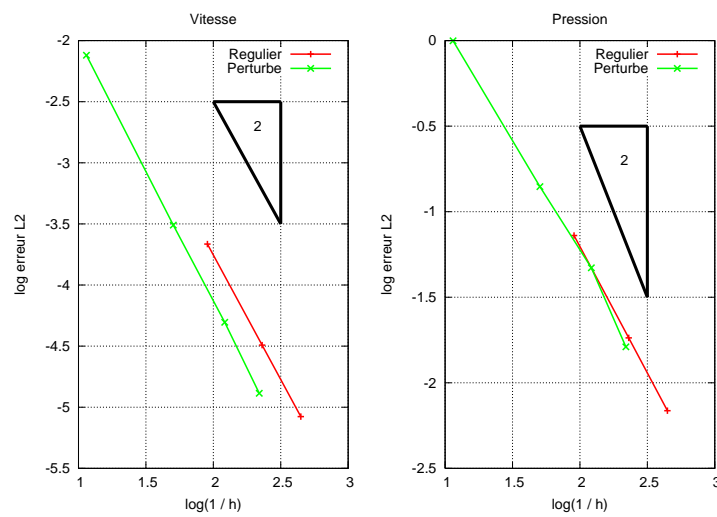
Néanmoins, on considérera avec réserve les résultats concernant les éléments jugés les moins performants, car dans leur cas, la résolution numérique peine très largement à converger. Ceci signifie que la recherche de l'approximation elle-même n'est pas parfaite, et qu'une erreur supplémentaire à celle de discrétisation est introduite dans la solution approchée, ce qui peut expliquer que la convergence évaluée soit plus faible que la convergence théorique. Ces problèmes de convergence empêchent ainsi une évaluation réellement indiscutable des performances des éléments. En particulier, on note que lors de la résolution itérative du problème discret, et ce, quelle que soit la méthode de résolution choisie, Uzawa ou système mixte, la convergence se fait de plus en plus faible au cours de la résolution, jusqu'à atteindre un point de stagnation.

Ainsi, le cas le moins satisfaisant est celui des solutions calculées par l'élément tétraédrique mini. Comme on peut le voir dans la figure 5.17, la pression calculée présente de très gros défauts, ce qui n'est pas le cas de la vitesse. Pourtant, la solution illustrée dans la figure correspond déjà à un maillage d'une résolution non-négligeable, puisque comprenant 96 000 éléments.

Malgré ces résultats qui semblent entrer en contradiction avec la théorie, dans



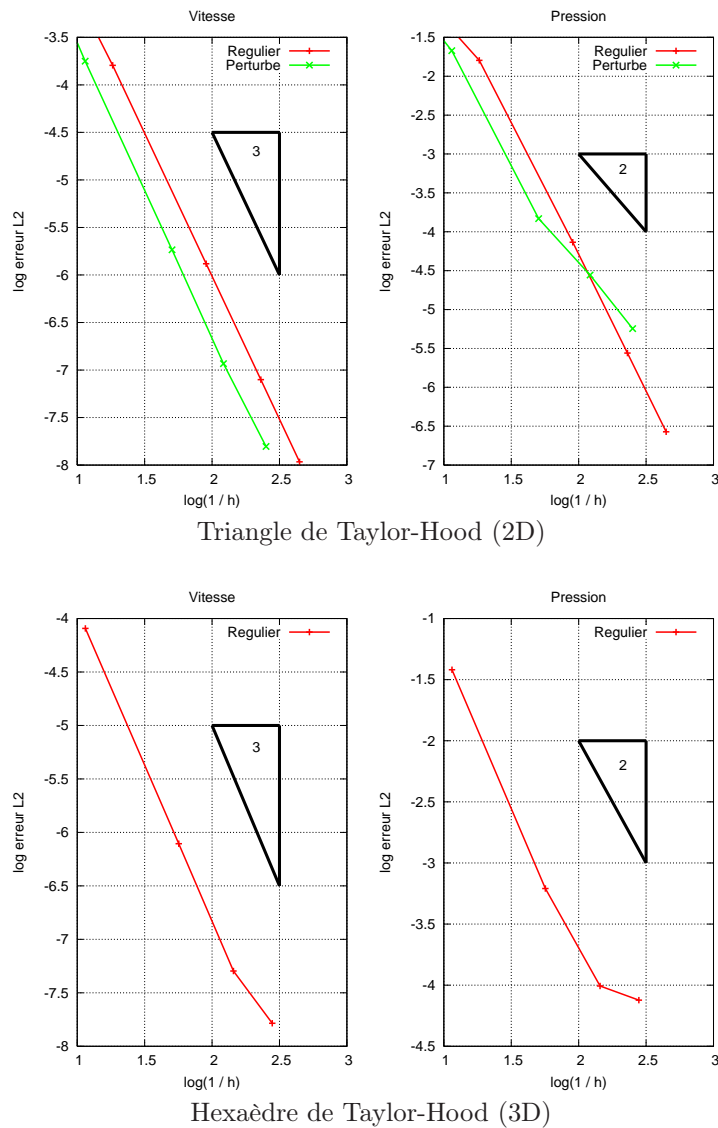
Quadrangle de Taylor-Hood (2D)



Quadrangle mini (2D)

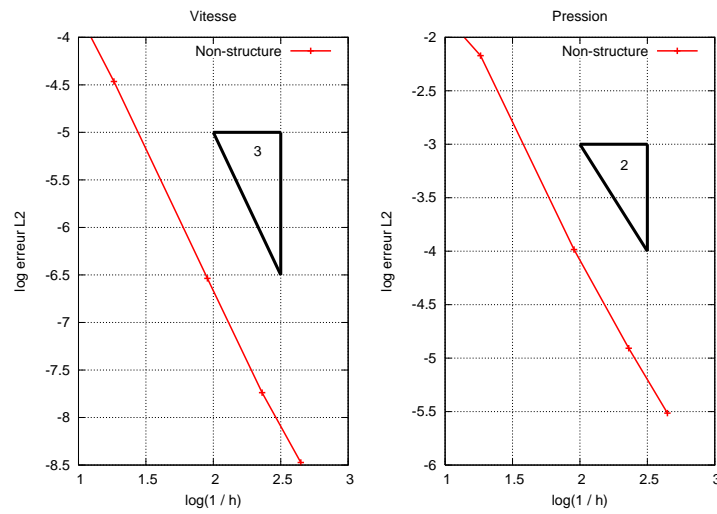
Elément	Situation	Convergence u	Convergence p
Quadrangle TH	Structuré régulier	2.98	3.07
Quadrangle TH	Structuré perturbé	3.15	2.5
Quadrangle mini	Structuré régulier	2.03	1.47
Quadrangle mini	Structuré perturbé	2.15	1.36

Figure 5.14: Tests de convergence (1) — Voir la légende de la figure 5.16

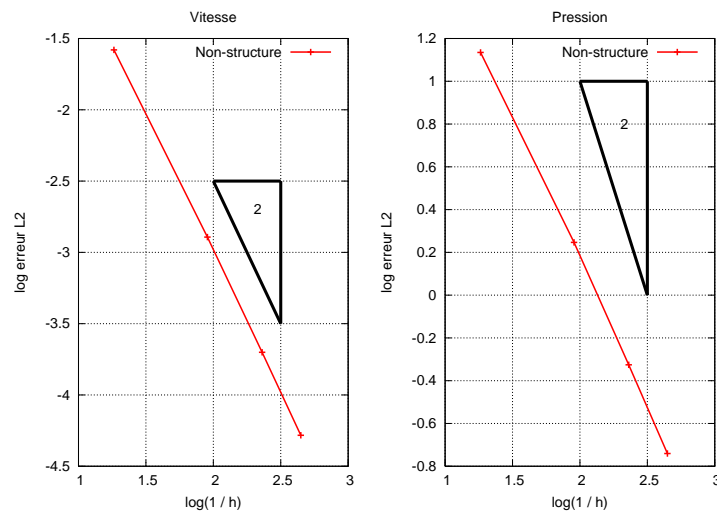


Elément	Situation	Convergence u	Convergence p
Triangle TH	Structuré régulier	2.93	2.88
Triangle TH	Structuré perturbé	3.13	2.55
Hexaèdre TH	Structuré régulier	2.72	2.03

Figure 5.15: Tests de convergence (2) — Voir la légende de la figure 5.16



Tétraèdre de Taylor-Hood (3D)



Tétraèdre mini (3D)

Élément	Situation	Convergence u	Convergence p
Tétraèdre TH	Non-structuré	2.85	2.07
Tétraèdre mini	Non-structuré	1.94	1.35

Figure 5.16: Tests de convergence (3) — Tests de convergence pour différents types d'éléments sur des domaines carrés ou cubiques. La pente de convergence théorique est figurée pour chaque graphique en plus des mesures. Pour les simulations bidimensionnelles, deux maillages structurés sont utilisés, l'un parfaitement régulier, l'autre dont les sommets internes sont légèrement perturbés. Le maillage du cube par hexaèdres emploie un maillage régulier, celui par tétraèdre un maillage non structuré. Le tableau donne les taux de convergence calculés par une méthode des moindres carrés à partir des mesures significatives. Les éléments de Taylor-Hood convergent avec les taux attendus, ce qui n'est pas le cas de la pression des éléments mini.

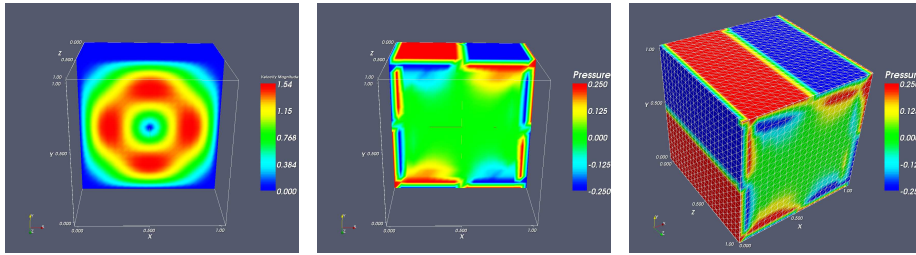


Figure 5.17: Résultats imprécis pour le tétraèdre mini. Coupes médianes dans le domaine, perpendiculaires à l'axe d'extrusion, avec la vitesse représentée à gauche et la pression au milieu. La peau du maillage de calcul est figurée à droite. La mauvaise convergence de la résolution numérique du système linéaire engendre une mauvaise convergence de la pression. La vitesse est pour sa part peu affectée.

le cas des éléments mini, on suppose, comme on l'a évoqué mais sans pouvoir en apporter la preuve, que ce problème de convergence est la manifestation de la faible performance de la résolution numérique, et non une remise en cause de l'exactitude de l'implantation réalisée. À l'appui de cette affirmation, on souligne que toutes les simulations utilisent un mode de calcul identique, très automatisé et dont la seule différence se situe dans la description des fonctions de base des éléments, ce qui amène à penser que les implantations des éléments et du simulateur ne sont pas la cause de la convergence trop faible.

Compte tenu des remarques que l'on a formulées, la validation ne peut être que partielle, et légèrement ambiguë, puisque l'on ne parvient pas toujours à retrouver les solutions attendues avec la précision que l'on est en droit d'attendre. Si ce problème se manifeste essentiellement avec les éléments mini, il reste fâcheux, car c'est le type d'élément qui paraît préférable pour l'infographie. Comparé à l'élément de Taylor-Hood, pourvu lui d'une meilleure convergence théorique, l'élément mini présente une économie appréciable en termes de degrés de liberté.

Pour aboutir à une validation définitive des performances des éléments ainsi que de la correction de l'implantation réalisée, il faudrait pouvoir disposer d'une méthode plus performante de résolution numérique du problème discret. Mais obtenir une telle méthode est complexe, en raison de la forme de point-selle de la discrétisation du problème de Stokes [10].

Imperméabilité Même si le simulateur n'est pas parfaitement validé, et reste globalement peu robuste, on peut tout de même s'en servir pour illustrer certains avantages de la méthode par éléments finis.

L'une des raisons qui nous ont poussé à étudier l'approche par éléments finis est illustrée dans la figure 5.18, qui met en jeu une situation avec des frontières d'orientations arbitraires, non-capturables par une grille Cartésienne. Dans cet exemple, une boule est placée dans un tube de section carré, un flux est imposé en entrée et en sortie, et des contraintes d'imperméabilité sont imposées sur le reste de la frontière du domaine, boule comprise. Comme l'illustre la figure, la contrainte d'imperméabilité est parfaitement respectée sur l'intégralité de la frontière. Quantitativement, l'erreur commise sur le flux traversant la surface

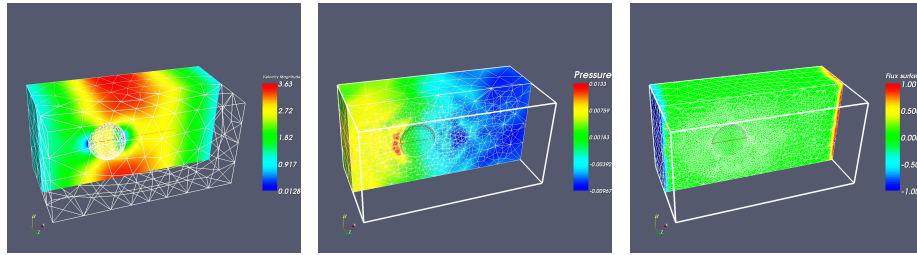


Figure 5.18: Exemple de domaine non orthotope : une boule est insérée dans un tuyau de section carrée. La vitesse du fluide est imposée à l'entrée et la sortie du tuyau. Le calcul est réalisé avec des éléments tétraédriques de Taylor-Hood. Des conditions de glissement $u \cdot n = 0$ sont imposées sur le milieu du domaine. De gauche à droite : vitesse, pression, et flux signé à travers la frontière du domaine.

frontière du centre du domaine Γ_c , est quasi-nulle :

$$\int_{\Gamma_c} (u \cdot n)^2 dx = 2.18 \cdot 10^{-5}$$

La mesure est restreinte au centre du domaine Γ_c , pour ignorer l'entrée et la sortie de fluide spécifiées dans le test.

Équations de Navier-Stokes stationnaires On a également testé le code réalisé dans le cadre de la résolution des équations de Navier-Stokes stationnaires, avec le problème-test bidimensionnel classique dit « de la cavité entraînée » représenté dans la figure 5.19. Ce problème considère une cavité carrée, sur la frontière de laquelle est imposée la vitesse. Celle-ci est nulle sur trois côtés, et tangente à la frontière sur le quatrième côté. L'objet du test est alors de faire varier la viscosité du fluide et d'étudier, entre autres, le comportement du vortex principal qui se forme, notamment par le déplacement de la position de son centre.

La méthode retenue pour résoudre ce problème est celle du point fixe sur les équations de Navier-Stokes stationnaires, qui demande de résoudre une série de problèmes de Stokes linéarisés comme le décrit la section 5.2.2. On utilise pour les simulations un maillage régulier du carré unité, par 40×40 quadrangles de Taylor-Hood, pour un total de 13 122 degrés de liberté en vitesse, et 1 681 degrés de liberté en pression.

Ce problème ne disposant pas d'une solution analytique, on compare les résultats obtenus par notre simulateur avec des résultats de simulations numériques publiés par Burggraf [13]. Une comparaison qualitative des lignes de courant dérivées du champ de vitesse est présentée dans la figure 5.20, et de la pression dans la figure 5.21. Deux autres comparaisons quantitatives sont également disponibles dans la figure 5.22. Elles représentent le profil de la vitesse horizontale le long de la coupe verticale médiane du domaine $x = 1/2$, ainsi que le déplacement du centre du vortex principal, en fonction de la viscosité. On constate que l'on obtient dans tous les cas des résultats en bon accord avec cette référence. Cette validation sur un cas non-linéaire permet également de supposer que l'implémentation réalisée est correcte.

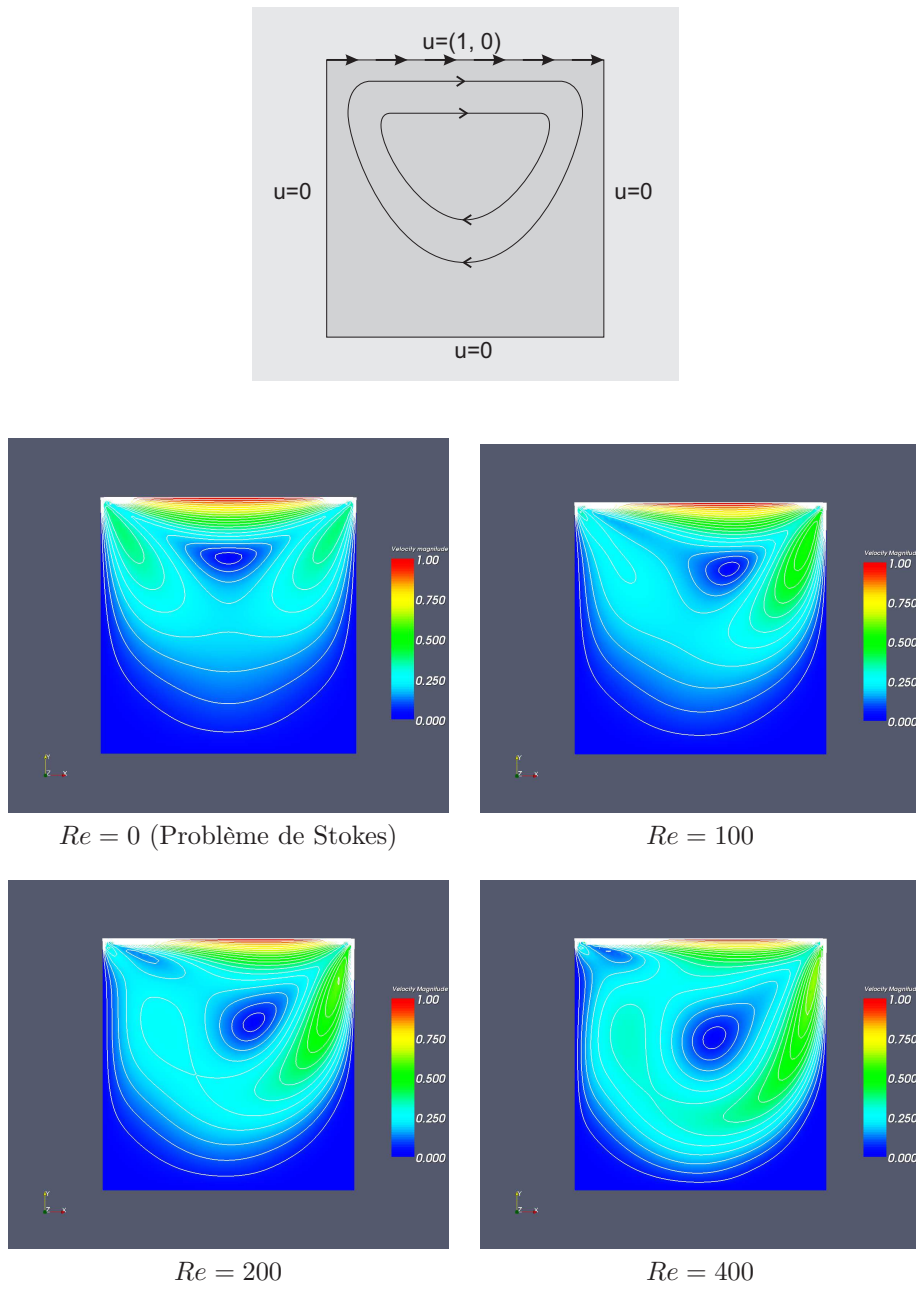


Figure 5.19: Problème test de la cavité entraînée. Situation simulée et norme du champ de vitesse obtenu pour $Re = 0$, $Re = 100$, $Re = 200$, et $Re = 400$.

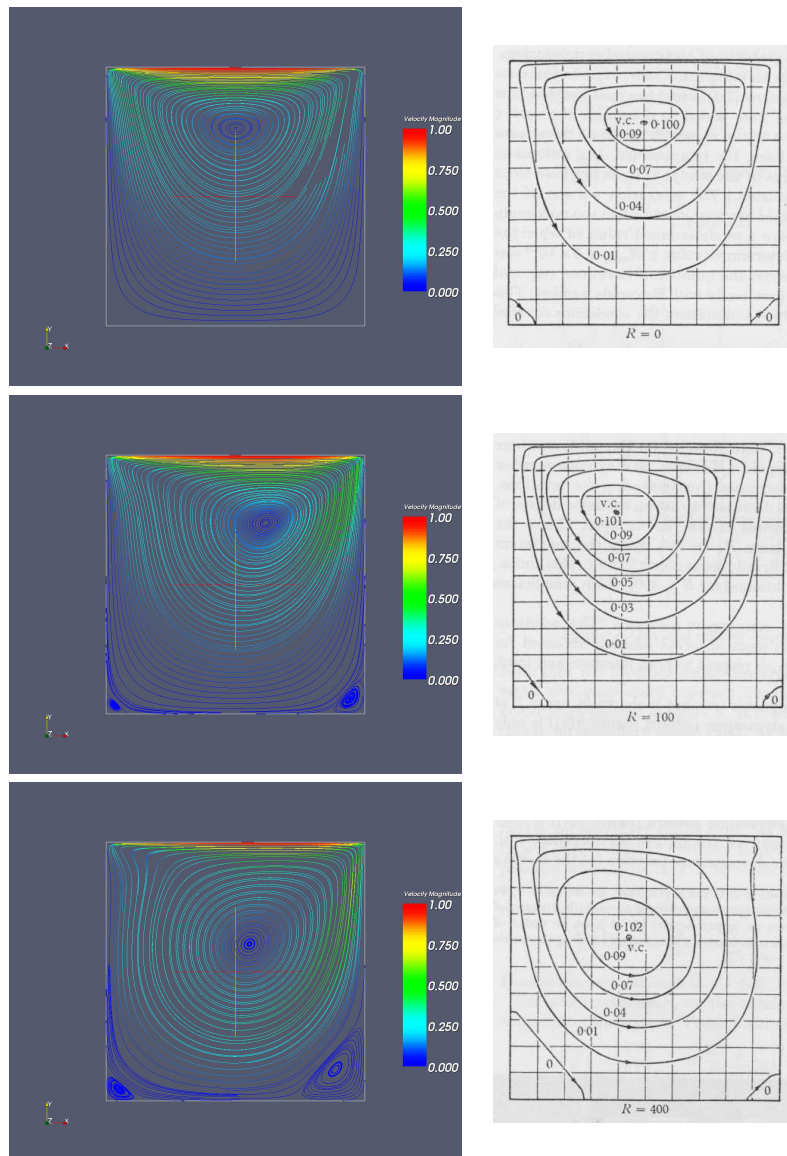


Figure 5.20: Problème test de la cavité entraînée — Vitesse. De haut en bas, $Re = 0$ (problème de Stokes), $Re = 100$, $Re = 400$. À gauche : lignes de courant dérivées du champ de vitesse calculé par le simulateur. À droite : données de référence extraites de Burggraf [13]. On ne tiendra pas compte des valeurs mentionnées dans la référence qui ne représentent pas la vitesse. Sous réserve d'une symétrie par rapport à l'axe vertical, les résultats sont en bon accord.

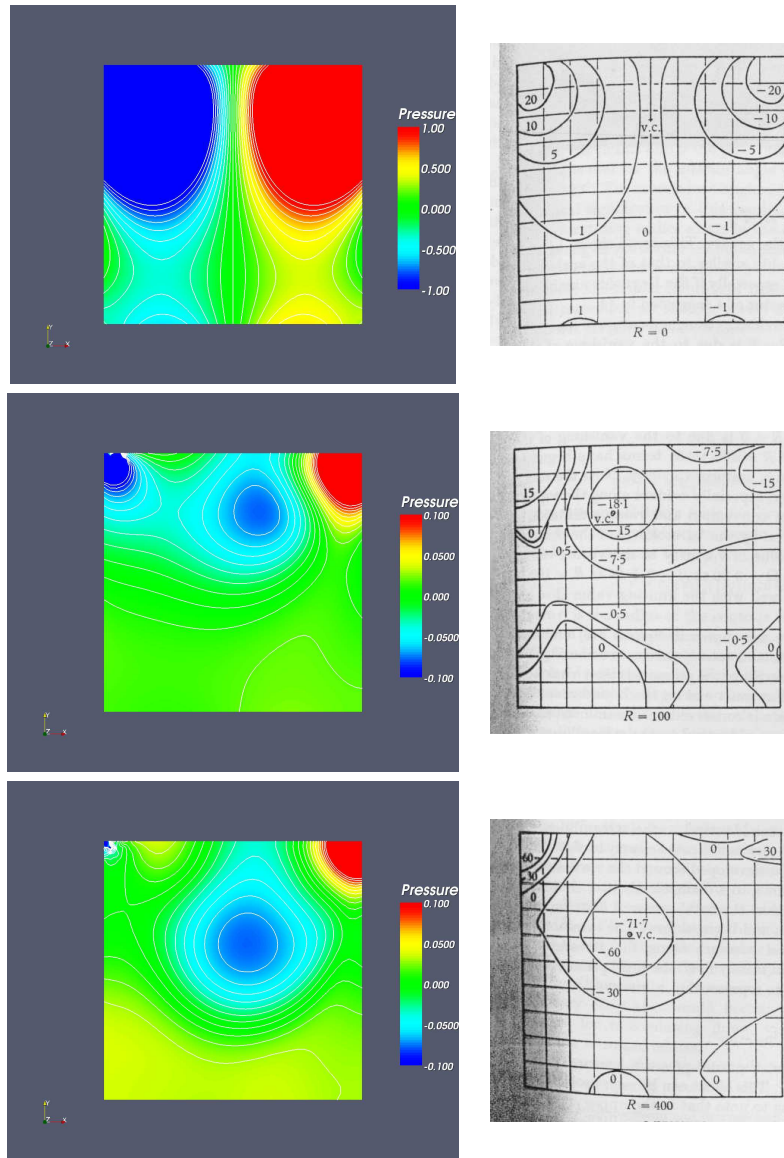


Figure 5.21: Problème test de la cavité entraînée — Pression. De haut en bas, $Re = 0$ (problème de Stokes), $Re = 100$, $Re = 400$. À gauche : champ de pression calculé par le simulateur. À droite : données de référence extraites de Burggraf [13]. On notera le bon accord qualitatif par rapport aux données extraites de Burggraf [13], sous réserve d'une symétrie par rapport à l'axe vertical. L'échelle de couleur de la pression correspond à une partie de la plage de pression totale, ce qui fait apparaître des zones plates aux coins du domaine.

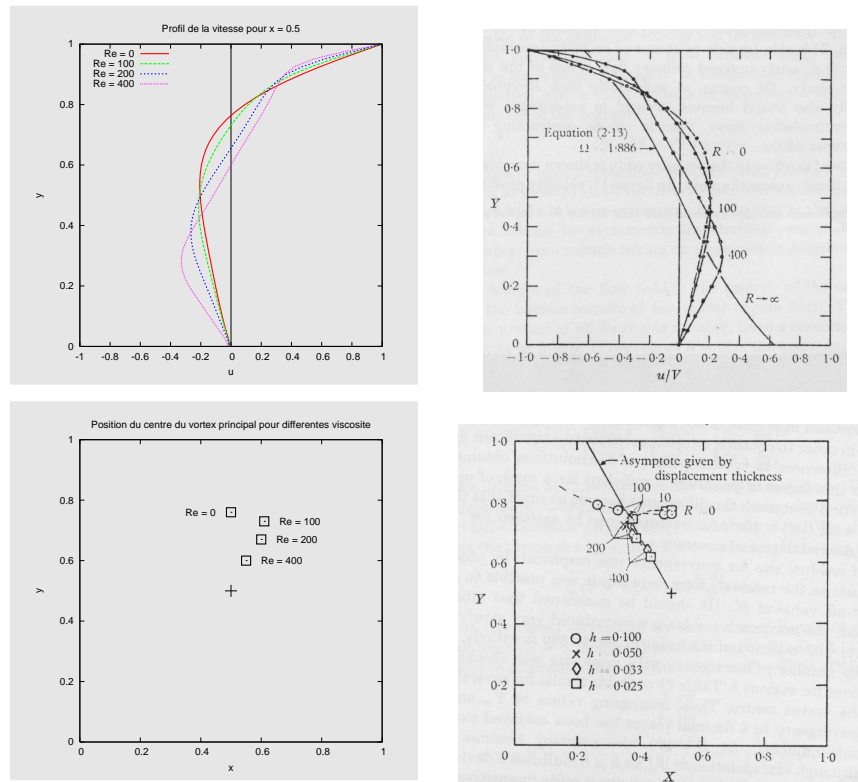
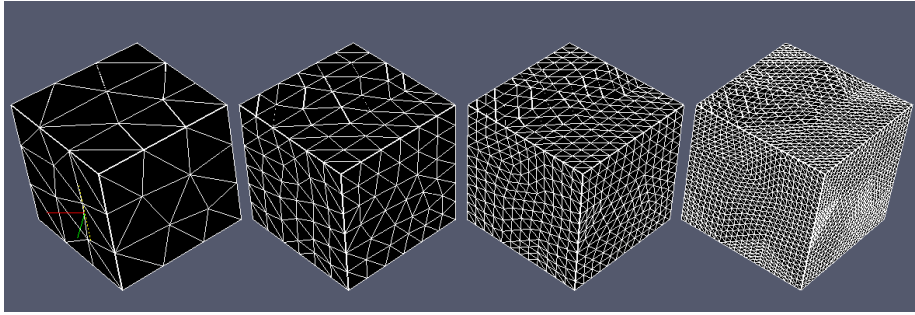


Figure 5.22: Comparaisons pour le problème de la cavité entraînée. À gauche sont figurées les données issues du simulateur que nous avons implémenté, et à droite les données de référence extraites de Burggraf [13]. Les deux graphiques du haut comparent le profil de la vitesse horizontale pour la ligne verticale médiane du domaine, et ce, pour différentes viscosités. Les graphiques du bas comparent l'influence de la viscosité sur la position du centre du vortex principal. On ne considère pour ce graphique que la comparaison pour le maillage le plus fin, représenté par des carrés dans les données de référence. Dans les deux cas, on peut noter un bon accord de notre simulateur avec les données de référence, modulo la symétrie due à l'inversion du cas-test.



\mathcal{T}_h	Nb. elts	Nb. ddl u	Nb. ddl p	Nb. nz A	Nb. nz B	Coût (Mo)
Élément mini						
3	1 60	666	62	18 414	3 738	0.26
6	1 280	4 842	334	139 392	27 264	1.92
12	10 240	37 173	2 151	1 086 507	208 569	14.97
24	81 920	291 831	15 357	8 583 453	1 632 351	118.08
Élément de Taylor-Hood						
3	1 60	1 002	62	66 402	6 789	0.84
6	1 280	6 453	334	485 109	47 682	6.12
12	10 240	43 071	2 151	3 704 265	356 589	46.66
24	81 920	347 691	15 357	28 943 145	2 756 487	364.16

Figure 5.23: Évolution de la consommation mémoire. Les statistiques concernent la série de maillages tétraédriques non-structurés du cube unité raffinés successivement, référencés dans la colonne \mathcal{T}_h par le nombre de divisions des arêtes. Sont ensuite présentés, le nombre d'éléments du maillage, le nombre de degrés de liberté en vitesse et en pression du problème, le nombre d'entrées non-nulles des matrices A et B , et le coût mémoire approximatif d'une telle paire de matrices, en comptant 12 octets par élément non-nul (un double et un entier).

Performances Au vu des remarques que l'on a pu faire précédemment sur la faible convergence de la méthode de résolution, l'analyse des performances en termes de vitesse de simulation n'a guère de sens pour porter un jugement définitif sur une résolution par éléments finis d'un problème de dynamique des fluides. La durée du calcul varie en effet largement en fonction du problème posé, des tolérances de convergence choisies, ou bien de la taille du maillage.

À titre d'exemple, les résultats pour le problème bidimensionnel « de la cavité entraîné » décrit plus haut, et dont la non-linéarité est gérée par point-fixe, ont nécessité approximativement 70 minutes de calcul sur un Pentium 4 cadencé à 2.6 GHz pour $Re = 400$. Le même problème, pour $Re = 100$, est résolu en approximativement 7 minutes et 30 secondes.

Toutefois, on peut affirmer que les temps de calcul sont longs, voire très longs, ce qui, en conjonction avec la faible robustesse de la méthode, nous a empêché de produire des animations. La quasi-totalité du temps de calcul est consommée par l'algorithme de résolution, le calcul de la discrétisation elle-même ne représentant qu'une part très restreinte du temps total requis par la méthode.

En revanche, le simulateur déjà écrit permet tout à fait d'appréhender les ressources nécessaires en termes de mémoire pour employer une discrétisation

par éléments finis. Pour illustrer ceci, la figure 5.23 présente l'évolution du coût mémoire pour une série de maillages tétraédriques non-structurés du cube unité, obtenus par raffinement successifs. Comme on peut le voir dans ces statistiques, si le coût en mémoire reste linéaire par rapport au nombre d'éléments, un grand nombre d'éléments est rapidement requis, et seules des grilles de taille relativement modeste peuvent être gérées. On notera cependant que le nombre de points de discrétisation d'un maillage non-structuré n'est pas directement comparable à celui d'une grille Cartésienne régulière, puisque le maillage non-structuré peut profiter d'éléments de tailles variables et s'adapter plus naturellement à la frontière.

Ces statistiques prouvent également l'intérêt que l'on peut trouver à employer les éléments mini plutôt que les éléments de Taylor-Hood, plus grand consommateurs de ressources, eux, abstraction faite d'éventuels problèmes de convergence.

5.7 Bilan

On a présenté une méthode de résolution d'un problème de Stokes généralisé par la méthode des éléments finis, dans le but de résoudre les équations de Navier-Stokes. Si l'on est parvenu à implémenter le simulateur des équations de Stokes correspondant, et à obtenir quelques résultats y compris dans le cas des équations de Navier-Stokes stationnaires, on n'est toutefois pas en mesure de résoudre les équations de Navier-Stokes non-stationnaires, et *a fortiori* de produire des animations du fluide.

La difficulté à produire des animations vient du problème de la très faible convergence, voire de la non-convergence, des méthodes itératives de résolution employées. On conjecture, au vu des tests effectués, que cette mauvaise convergence est inhérente aux problèmes résolus et ne découle pas d'une implémentation défaillante, sans que l'on puisse toutefois totalement exclure cette hypothèse. Le manque de performance et la robustesse limitée de la résolution font ainsi obstacles à la résolution de problèmes non-stationnaires et rendent difficile le traitement de problèmes d'une taille significative.

Cette problématique de convergence de la résolution numérique est en fait centrale à la simulation elle-même. En effet, c'est durant cette phase qu'est réellement résolu le problème posé, et non pendant les calculs de discrétisation. La résolution ardue traduit donc la nature complexe du problème sous-jacent. Or, la simulation de fluide n'est pas connue pour offrir des problèmes simples à résoudre.

Il est en particulier malaisé, dans ces conditions, de négocier le passage de simulations bidimensionnelles à des simulations tridimensionnelles, passage qui se traduit par une augmentation drastique de la taille des problèmes à résoudre. La convergence faible, et éventuellement erratique, rend également délicat le réglage des critères de convergence des méthodes itératives : si les critères sont trop restrictifs, la méthode se poursuit sans jamais arriver à converger, et s'ils sont trop lâches, les résultats ne sont pas ceux attendus, même s'ils peuvent s'en approcher.

S'il est indéniablement regrettable de ne pas être parvenu à développer un simulateur capable de produire des animations utilisables en infographie, les travaux réalisés donnent tout de même satisfaction sur différents points. Comme

l'ont confirmé les résultats présentés, la méthode des éléments finis parvient à gérer parfaitement des domaines de simulation de frontières arbitraires. Pour ces domaines de frontières quelconques, on arrive également à imposer une imperméabilité de la solution qui est totalement respectée par la méthode.

À cette performance s'ajoute la totale absence d'ambiguïté concernant la gestion du maillage et la prise en compte des conditions aux limites, ce qui permet une résolution absolument nette des problèmes posés. On notera que l'on n'oppose pas cette caractéristique de clarté de résolution aux méthodes MAC dans leur généralité, mais plutôt à l'approche de résolution parfois un peu floue pratiquée dans le contexte de l'infographie.

Un dernier point notable de satisfaction concerne non pas la méthode de résolution elle-même, mais l'approche que nous avons suivie pour en valider l'implémentation. Il est clair qu'une phase de validation rigoureuse est nécessaire, à l'image de celle que l'on peut trouver pour des solveurs dédiés à l'ingénierie. Seule une telle phase permet d'infirmer ou de confirmer une implémentation, de découvrir des erreurs, et de mettre en lumière des problèmes ou des difficultés intrinsèques à la méthode comme on a pu le constater lors des développements réalisés. En fait, cette phase s'impose naturellement dès lors que l'on considère la problématique de simulation non plus comme un algorithme de simulation produisant un résultat, mais comme le solveur dédié à la résolution d'une équation.

Dans ce cas, la validation numérique prend tout son sens, et offre une base concrète d'évaluation de la méthode mise en œuvre. Au niveau de complexité des méthodes appliquées, juger approximativement de manière visuelle des résultats obtenus ne suffit plus, en effet, à constituer un mode de validation crédible. On notera bien que cette affirmation ne vient absolument pas en contradiction avec la finalité visuelle de la simulation en infographie car on parle ici de la mise au point des méthodes. Il reste évident que, du point de vue de l'infographie, la validation finale résulte du jugement subjectif des spectateurs, alors que la validation de l'intérêt de la méthode dépend du verdict des animateurs.

Si l'on revient justement sur la méthode elle-même et sur son adéquation avec l'informatique graphique, on peut conjecturer, avec l'expérience acquise, qu'elle ne constituera sans doute pas le mode de simulation des fluides dominant. En effet, il apparaît assez clairement que cette méthode, si elle présente un certain nombre d'avantages, ne peut rivaliser avec des solveurs à grille régulière, ni en vitesse, ni en taille des problèmes qu'il est possible de résoudre. Même en admettant une marge de progression significative, il est peu probable que le traitement par éléments finis devienne la règle dans le cas de simulations simples, les plus nombreuses. En revanche, les qualités intrinsèques de la méthode, qui nous ont poussé vers son étude et que l'on a évoquées précédemment, restent indéniables, et ce mode de simulation offre un intérêt pour la gestion de simulations complexes, comme celles requérant une interaction précise avec des solides, par exemple. Néanmoins, cette conjecture ne peut être vérifiée qu'avec l'obtention de solveurs plus performants et correctement validés.

Les résultats obtenus par la voie des éléments finis appellent donc à poursuivre l'étude de la méthode. On n'a employé ici qu'une méthode très basique, et il existe une multitude de travaux, dans les communautés de physique et de mathématiques, destinées à produire des simulateurs bien plus performants.

Dans cette optique d'amélioration, on peut distinguer des perspectives selon deux échelles de temps. À court terme, pour rendre la méthode utilisable et

pouvoir juger de son intérêt avec plus de certitude, il est nécessaire de s'appuyer sur une procédure de résolution numérique plus robuste et beaucoup plus performante [10, 131]. Il est également possible de varier le type d'éléments, par exemple en employant un élément mini optimisé [67] ou d'autres types d'éléments économiques. Si elle est bien évaluée et bien contrôlée, la réduction de la précision des solutions engendrée par ces éléments peut être exempte de tout effet dommageable dans le contexte de l'infographie.

À plus long terme, et une fois la simulation mise au point, l'on devrait pouvoir tirer réellement parti des avantages qu'offre la méthode des éléments finis, par exemple en explorant l'usage de maillages bien adaptés à des domaines arbitraires, ou pour améliorer les performances en variant la taille des éléments au sein des maillages. Une fois bien maîtrisée, cette simulation peut également offrir une solution pour représenter directement la surface d'un liquide par le maillage lui-même, dans une simulation à surface libre, en y intégrant éventuellement des effets avancés comme la tension de surface [121], ou dans la formulation d'un problème d'interaction du fluide avec des solides [89].

Néanmoins, il faut garder à l'esprit que cette méthode reste complexe à appréhender, et pas nécessairement simple à mettre en œuvre, notamment en raison de la largeur du spectre des points à traiter dans son application : aspects théoriques des espaces d'approximation à choisir, intégration numérique pratique, résolution itérative de systèmes linéaires ou encore gestion d'un maillage et de masses de données importantes. Avant même de pouvoir prétendre à la performance, l'implémentation d'un solveur fonctionnel reste déjà une entreprise ardue, compte tenu de la complexité propre de chacune des composantes du problème.

Chapitre 6

Visualisation interactive de la réfraction

À l'examen d'un objet transparent, tel un liquide, le phénomène visuel le plus frappant est certainement celui engendré par la réfraction de la lumière. Cet effet est d'autant plus remarquable que le point de vue se déplace par rapport à l'objet, la déformation de l'arrière-plan s'animant alors. Si la représentation de ce phénomène constitue un grand classique de l'informatique graphique avec des rendus par lancé de rayons, il a été assez peu simulé dans le cadre d'un rendu interactif.

Ce chapitre présente une technique que nous avons mise au point afin de permettre une visualisation interactive de ce phénomène de réfraction, sans poser de contrainte sur la géométrie des objets réfringents. La visualisation n'est toutefois pas sans contrainte, et reste limitée à des objets statiques.

Elle repose sur un principe similaire à celui proposé par Heidrich *et al.* [69], dans lequel la distorsion engendrée par la présence de l'objet est capturée durant un précalcul à l'aide d'un échantillonnage dense. Cette information est ensuite employée durant le rendu pour synthétiser la distorsion induite par la réfraction correspondant au point de vue courant. Le problème que l'on rencontre pour appliquer une telle technique est de représenter efficacement cette très grande masse de données.

Plutôt que de nous appuyer sur une représentation par un *light field* paramétré par deux plans, nous proposons l'emploi d'une représentation surfacique de l'information, similaire dans l'esprit aux *surface light fields* [19], mais avec une gestion de la variation directionnelle de l'information à l'aide d'harmoniques sphériques. L'usage interactif d'un tel mode de compression repose alors sur le support matériel intégral de la méthode, une opportunité offerte par la programmabilité du matériel graphique.

Si le recours à un précalcul de la distorsion induite par la réfraction ne constitue pas une nouveauté en soi, c'est dans la gestion de l'information que réside l'innovation.

La méthode proposée est décrite dans les sections 6.3 à 6.5. Son cadre général, ainsi qu'une description des données manipulées seront exposés les premiers, avant de passer aux aspects relatifs à la représentation compressée de ces données à l'aide d'harmoniques sphériques. L'emploi de ces données pour le rendu

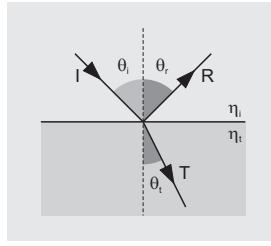


Figure 6.1: Intersection d'un rayon de lumière avec un dioptr. La figure se place dans le plan d'incidence défini par la direction du rayon d'incidence et la normale au dioptr au point d'intersection.

interactif sera ensuite détaillé, puis des résultats seront présentés dans la section 6.6, avant de tirer un bilan de la méthode. Finalement, on esquissera dans la section 6.8 les contours d'une autre utilisation possible d'information directionnelle, pour représenter des *surface light fields*.

Avant de poser plus précisément le modèle des situations que la méthode est en mesure de traiter, un très bref rappel d'optique, ainsi qu'un tour d'horizon des autres techniques dédiées à figurer la réfraction en infographie feront l'objet des deux sections suivantes.

6.1 Rappels d'optique

Le modèle que nous choisissons d'associer à la propagation de la lumière est celui communément employé en infographie, qui découle de l'optique géométrique. La lumière y est idéalisée pour pouvoir être modélisée à l'aide de rayons, se déplaçant en lignes droites dans des milieux matériels homogènes. Par ailleurs, il n'est pas fait de distinction entre les longueurs d'ondes des différentes radiations lumineuses : toutes sont supposées voyager à la même vitesse dans tous les milieux.

On considère que la scène est partitionnée en un ensemble de milieux homogènes, séparés par des interfaces clairement délimitées, des dioptr. À chaque milieu est associé un matériau transparent ou semi-transparent, d'indice de réfraction connu. Ces milieux sont par ailleurs supposés être non-conducteurs, ce qui entraîne que leurs indices de réfraction sont réels et non complexes. L'absence de prise en compte des longueurs d'ondes implique que les matériaux sont caractérisés par un seul et unique indice de réfraction.

Les phénomènes optiques que l'on cherche à modéliser se situent essentiellement à l'interface entre matériaux. Les lois de l'optique prédisent en effet qu'avec les hypothèses admises, un rayon incident donne naissance à deux rayons, l'un réfléchi et l'autre réfracté, lorsqu'il rencontre une interface entre deux milieux. Le rayon réfléchi va poursuivre son trajet dans le milieu d'incidence, au contraire du rayon réfracté qui va poursuivre sa course dans le milieu opposé. Une telle situation est représentée dans la figure 6.1. Les deux rayons ont pour origine le point d'intersection du rayon incident avec l'interface, et se déplacent dans le plan défini par la normale à l'interface au point d'incidence, et par le rayon incident.

6.1.1 Géométrie des rayons

Pour caractériser la géométrie d'une telle situation, on pose, en considérant le plan dans lequel tous les rayons évoluent :

- θ_i l'angle d'incidence, compris entre le rayon incident I et la normale à l'interface
- θ_r l'angle de réflexion, compris entre le rayon réfléchi R et la normale à l'interface
- θ_t l'angle de réfraction, compris entre le rayon réfracté T et l'opposé de la normale à l'interface

Les deux milieux, incident et opposé, ont pour indices de réfraction respectifs η_i et η_t .

La direction du rayon réfléchi est alors décrite par la loi :

$$\theta_i = \theta_r$$

et la direction du rayon réfracté est donnée par la loi de Descartes :

$$\eta_i \sin \theta_i = \eta_t \sin \theta_t$$

Dans le cas où $\frac{\eta_i}{\eta_t} \sin \theta_i \notin [-1, 1]$, l'angle θ_t n'est pas défini, et la situation correspond au cas de la réfraction interne totale : le rayon incident est totalement réfléchi, sans réfraction, à l'exception d'une onde évanescente, ici négligeable.

6.1.2 Énergie portée par les rayons

Il s'agit également de caractériser la répartition entre les rayons réfléchis et réfractés de l'énergie portée par le rayon incident. Il faut pour cela se référer à la nature ondulatoire de la lumière. Dans ce cadre ondulatoire, la lumière est représentée par une onde électromagnétique, dont la polarisation peut être décomposée en deux composantes perpendiculaires à la direction de propagation.

Pour l'étude de la situation considérée, on choisit de décomposer l'onde selon les deux composantes parallèle et perpendiculaire au plan d'incidence dans lequel évoluent les rayons. Les équations de Fresnel expriment alors le rapport des amplitudes des ondes lumineuses réémises par rapport à celles de l'onde incidente, pour chacune des deux composantes :

$$\begin{cases} r_{\perp} = -\frac{\sin(\theta_i - \theta_t)}{\sin(\theta_i + \theta_t)} \\ r_{\parallel} = \frac{\tan(\theta_i - \theta_t)}{\tan(\theta_i + \theta_t)} \end{cases} \quad \begin{cases} t_{\perp} = \frac{2 \sin \theta_t \cos \theta_i}{\sin(\theta_i + \theta_t)} \\ t_{\parallel} = \frac{2 \sin \theta_t \cos \theta_i}{\sin(\theta_i + \theta_t) \cos(\theta_i - \theta_t)} \end{cases}$$

où r_{\perp} et r_{\parallel} correspondent aux rapports pour l'onde réfléchie, et t_{\perp} et t_{\parallel} aux rapports pour l'onde réfractée. Pour la synthèse d'image, ce ne sont toutefois pas les amplitudes des ondes qui présentent un intérêt, mais les énergies qu'elles portent, calculables par $|E| = (E_{\perp}^2 + E_{\parallel}^2)^{\frac{1}{2}}$ pour une onde E .

Le rapport \bar{R} entre énergie réfléchie $|R|$ et énergie incidente $|I|$ est alors calculable comme :

$$\bar{R} = \frac{|R|^2}{|I|^2} = \frac{r_{\perp}^2 I_{\perp}^2 + r_{\parallel}^2 I_{\parallel}^2}{I_{\perp}^2 + I_{\parallel}^2}$$

Dans le cas d'une lumière naturelle, non-polarisée, les deux composantes de polarisation sont en moyenne identiques $\langle I_{\perp} \rangle = \langle I_{\parallel} \rangle$, ce qui implique que :

$$\begin{aligned}\bar{R} &= \frac{1}{2} [r_{\perp}^2 + r_{\parallel}^2] \\ &= \frac{1}{2} \left[\frac{\sin^2(\theta_i - \theta_t)}{\sin^2(\theta_i + \theta_t)} + \frac{\tan^2(\theta_i - \theta_t)}{\tan^2(\theta_i + \theta_t)} \right]\end{aligned}$$

Le rapport \bar{T} entre énergie réfractée $|T|$ et énergie incidente $|I|$ correspond, lui, à :

$$\bar{T} = 1 - \bar{R}$$

Un cas particulier survient pour une incidence normale, c'est-à-dire quand $\theta_i = 0$, puisqu'il n'est plus possible de dégager un plan d'incidence particulier. Dans ce cas, le rapport pour l'énergie réfléchie est défini comme :

$$\bar{R} = \frac{(\eta_i - \eta_t)^2}{(\eta_i + \eta_t)^2}$$

et le rapport pour l'énergie réfractée reste toujours exprimé par :

$$\bar{T} = 1 - \bar{R}$$

Ces formules sont établies à partir de la nature ondulatoire de la lumière et des équations de Maxwell. Leur dérivation peut être retrouvée dans les ouvrages de référence consacrés à l'optique [59].

6.1.3 Absorption

Si un matériau n'est pas parfaitement transparent, il interagit avec la lumière qui le traverse et il est alors considéré comme un milieu participatif. Dans le cas particulier le plus simple qui sera considéré ici, il est possible de supposer que le matériau se contente d'absorber une partie de l'énergie lumineuse qui le traverse, sans jamais la réémettre dans aucune direction.

La variation d'énergie due à l'absorption dI sur une distance infinitésimale dx du média, est alors proportionnelle à l'énergie I_0 à l'entrée du média, ainsi qu'à un coefficient d'absorption κ propre au matériau. L'atténuation est donc de la forme :

$$dI = -\kappa I_0 dx$$

Par conséquent, l'intensité de la lumière varie suivant une loi exponentielle, en fonction de la distance x parcourue dans le média, du coefficient d'absorption κ , et de l'intensité I_0 à l'entrée du milieu semi-transparent :

$$I(x) = I_0 e^{-\kappa x}$$

6.2 Travaux existants

Nous présentons dans cette section les principales approches de l'informatique graphique grâce auxquelles il est possible d'inclure des objets réfringents dans des images de synthèse. Celles-ci ne sont pas très nombreuses, bien que la problématique soit étudiée depuis plus d'une vingtaine d'années.

L'aisance de l'inclusion du phénomène de réfraction dépend très largement de la nature du rendu employé pour synthétiser l'image. On rappelle que le phénomène décrit précédemment, localement, pour un seul rayon, prend place lorsque n'importe quel rayon rencontre n'importe quel dioptre. Ceci engendre d'une part une explosion du nombre de rayons nécessaires à la composition d'une scène, et possède d'autre part le potentiel d'annuler toute structure commune entre les calculs requis par pixels proches.

6.2.1 Approches apparentées au lancé de rayons

La méthode la plus simple et la plus populaire pour représenter la réfraction est sans nul doute le lancé de rayons, comme déjà illustré dans les travaux de Whitted [179]. Avec ce mode de rendu, dans lequel les rayons de lumière sont explicitement simulés, et générés à la demande, la prise en compte de la séparation du rayon incident en rayon réfléchi et rayon réfracté sur un dioptre se réalise naturellement : il suffit de suivre le modèle optique précédemment présenté qui s'applique parfaitement.

Toutefois, même parfaitement simulée dans un cadre de lancé de rayons, la réfraction est très fortement dépendante de l'orientation locale des surfaces, et particulièrement de la continuité de variation de la normale sur la surface des objets. Si la définition d'une telle normale est implicite pour les objets paramétriques, ce n'est pas le cas pour des objets volumiques définis par échantillonnage, pour lesquels la reconstruction de la normale doit être effectuée avec attention pour obtenir un résultat de bonne qualité, comme le proposent Li et Mueller [90].

La facilité de traitement par le lancé de rayons est due au fait que des rayons infinitésimaux sont simulés, analogues au modèle proposé par l'optique géométrique. En revanche, si la lumière n'est plus directement simulée sous la forme de rayons, mais de faisceaux, comme le proposent Heckbert et Hanrahan [68], le modèle ne peut plus être appliqué directement, même si la lumière peut être aisément suivie dans la scène. Dans le cadre d'un algorithme récursif de lancé de faisceaux, le résultat de la réfraction d'un faisceau incident par un dioptre n'est en effet pas exactement représentable par un autre faisceau. L'approximation proposée par Heckbert et Hanrahan consiste alors à ignorer l'erreur de représentation qui peut exister et à tout de même considérer un faisceau réfracté, approximatif.

Du point de vue qualitatif, l'approche par lancé de rayons est la meilleure, outre qu'elle se trouve être la plus simple à mettre en œuvre, car elle peut gérer l'irrégularité des trajets de la lumière sans aucun problème. Néanmoins, son coût se révèle trop élevé pour un usage à grande échelle en conjonction avec une implémentation logicielle. Une implémentation employant un matériel graphique programmable est possible, comme le proposent Purcell *et al.* [125], mais se révèle complexe à mettre en œuvre, et uniquement adaptée à des scènes réduites.

6.2.2 Approches par projection de primitives

Pour autoriser un rendu plus rapide, il est nécessaire de se tourner vers les approches de rendu plus structurées, maintenant pleinement accélérées par le matériel graphique, qui projettent des primitives à l'écran et les y remplissent.

Cependant, ces approches ne sont effectives que s'il existe une forte cohérence des calculs requis entre les pixels couverts par la, ou les, primitives projetées à l'écran. Les différences au sein des approches par projection de primitives résultent de différents compromis entre précision de l'approximation et régularité des calculs à effectuer, et donc vitesse de rendu.

Une première simplification souvent effectuée, est qu'il est possible de découpler l'objet réfringent de son environnement, supposé infiniment distant. Cette hypothèse a pour conséquence d'autoriser la représentation de l'environnement par une *environment-map* [56], et de pouvoir se concentrer sur la réfraction engendrée par l'objet.

On peut distinguer, dans le contexte du rendu par projection de primitives, deux classes d'algorithmes : ceux généraux, applicables à une large classe d'objets, et ceux spécialisés dans le traitement d'un type de scène ou d'objets spécifiques.

Pour traiter le cas d'objets généraux, mais minces, Kay et Greenberg [79] proposent d'assimiler le résultat de la réfraction à un champ de faibles distorsions, et d'appliquer la déformation calculée à une image de fond. Le champ de distorsion est calculé en considérant la géométrie des objets fins transparents à visualiser, et une réfraction très simplifiée. Cette approche pionnière est bien évidemment très limitée, et ne peut admettre que de faibles distorsions, les rayons étant supposés voyager globalement dans la même direction.

Une autre approche, proposée par Wyman [182], est bien plus récente et autorise à traiter des objets d'épaisseur arbitraire et variable. Pour cela, il est fait usage d'une approximation qui suppose que la réfraction est uniquement double dans l'objet, *i.e.* la lumière y pénètre en un premier point de la surface avant des objets, puis en ressort ensuite en un autre point sur la face arrière, sans interaction intermédiaire. La méthode procède en estimant pour chaque pixel à rendre — c'est-à-dire pour chaque point d'entrée dans l'objet — la position de sortie de l'objet ainsi que la normale associée, grâce à une estimation de la distance parcourue dans l'objet et grâce à un rendu préalable de la surface arrière de l'objet.

Cette technique, qui autorise des effets très réalistes dans la majorité des cas, fournit des résultats interactifs grâce à un support matériel intégral. Elle se trouve tout de même limitée par l'hypothèse restrictive sur la nature du trajet des rayons de lumière dans l'objet, et ne peut simuler qu'une paire de déviations des rayons. Dans le cas d'un objet auto-occlusif, cette limitation se traduit par la disparition pure et simple dans le rendu de la partie occultée de l'objet.

D'autres approches existent, mais elles sont spécialisées dans le traitement d'objets ou de scènes d'une nature particulière et s'appuient sur ces particularités.

Ts'o et Barsky [168] proposent ainsi un traitement pour le cas d'une large surface d'eau maillée, comme une surface d'océan, vue depuis un point en altitude. La surface pouvant être approximativement supposée plane à cette échelle, tout comme le fond de l'étendue d'eau, tous les rayons adoptent un comportement similaire depuis un point de vue en altitude : une seule déviation de la lumière, due à la surface, existe entre le fond et le point de vue. Il est alors proposé de calculer la déviation des rayons en chaque sommet du maillage de la surface, et d'indexer en conséquence une texture figurant le fond de l'eau.

Une autre situation très spécifique incluant une réfraction importante est celle traitée par Guy et Soler [60], qui consiste en la visualisation de pierres

précieuses. L'objectif est d'offrir un rendu de très grande qualité dans un but de prototypage pour la joaillerie. Le modèle proposé, qui inclut un luxe de détails optiques bien supérieur à celui normalement employé pour la synthèse d'images, s'apparente fortement à un rendu par lancé de faisceaux supporté par le matériel graphique programmable. La gestion de la réfraction y est donc traitée en employant l'approximation proposée par Heckbert et Hanrahan [68], qui permet de conserver un caractère récursif à la propagation de la lumière. Si le modèle très complet et l'implémentation matérielle réalisée permettent d'obtenir des résultats visuellement plus que convaincants à une vitesse interactive, des limitations subsistent néanmoins : la technique, très coûteuse, ne peut s'appliquer qu'à des objets comportant au plus quelques centaines de faces, celles-ci devant nécessairement être planes.

Toujours en employant la même approximation sur les faisceaux, Diefenbach et Badler [30] proposent un rendu de l'image en plusieurs passes, chacune matérialisant l'effet de la réfraction à travers une face plane réfringente de la scène. Elles sont appliquées récursivement et l'image finale produite peut comporter des zones où la ligne de vue a été déviée plusieurs fois. Il est à noter que cette approche ne fait pas usage d'un environnement placé à l'infini, mais de rendus effectifs, et tient donc correctement compte des effets de parallaxe. Elle ne considère cependant que des objets plans.

Si des effets de parallaxe sont recherchés en conjonction avec la réfraction, c'est-à-dire si l'environnement ne peut être supposé infiniment distant, il est nécessaire de recourir à une représentation de ce dernier plus complexe qu'une *environment map*. La solution proposée par Heidrich *et al.* [69] est alors de représenter l'environnement comme un *light field*, et de pouvoir ainsi l'échantillonner en fonction de la direction et de la position de sortie des rayons de l'objet. L'inconvénient d'une telle approche tient évidemment aux ressources qu'elle requiert pour représenter l'environnement.

6.2.3 Approches hybrides

Le problème des approches par projection est qu'elles imposent une grande régularité dans les calculs effectués de pixel en pixel. Cette régularité nécessaire allant à l'opposé du caractère peu structuré de la réfraction, mais pouvant être facilement gérée par lancé de rayons, des approches hybrides ont été développées. Pour représenter un objet réfringent maillé, elles agissent essentiellement en évaluant la déviation induite par la réfraction pour les sommets du maillage par lancé de rayons, puis emploient une procédure de remplissage des pixels couverts à l'écran par les triangles interpolant les déviations.

Afin d'atteindre des vitesses de rendu interactives, Ohbuchi [117] propose d'employer un matériel spécialement développé prenant en charge la phase de lancé de rayons. Hakura et Snyder [63] proposent plutôt d'employer une subdivision adaptative du maillage, afin de limiter au maximum le nombre d'évaluations de trajets de la lumière, tout en conservant une qualité visuelle satisfaisante.

Dans cette dernière approche, il est également proposé de faire usage d'une forme de représentation avancée de l'environnement, propre à représenter partiellement les effets de parallaxe. Cette représentation s'appuie sur le stockage de l'environnement sous différents points de vue, c'est-à-dire en considérant des environnements structurés en couches concentriques successives, pour tenir compte des disparités de distance qui y sont présentes.

6.2.4 Approche avec précalcul

Plutôt que de chercher à simuler complètement la déviation induite par la présence d'objets réfringents de manière interactive, une autre approche est proposée par Heidrich *et al.* [69]. Elle consiste à capturer la distorsion induite par l'objet durant une phase de précalcul, puis à employer les données précalculées pour synthétiser interactivement la distorsion correspondant au point de vue courant, sans nécessiter des calculs importants.

La difficulté d'une telle approche consiste en la manipulation de cette distorsion, qui est fonction, en chaque point de l'objet, de la direction d'observation, et qui constitue donc une fonction à quatre dimensions. Pour stocker l'échantillonnage d'une telle fonction, il est proposé d'employer le formalisme des *light fields* paramétrés par deux plans. En raison d'une forme de compression limitée, une large quantité de mémoire est nécessaire à la réalisation de cette technique. L'échantillonnage offrant par conséquent une résolution relativement faible, la qualité du résultat s'en ressent. En outre, cette forme de paramétrisation n'est pas satisfaisante de par la liberté de point de vue restreinte qu'elle offre.

L'approche que nous proposons est dans son principe semblable à celle de Heidrich *et al.* [69], mais le mode de stockage des données diffère radicalement comme on pourra en juger.

6.2.5 Acquisition d'une distorsion réelle

À la périphérie de la problématique de synthèse complète d'objets réfringents, se situe la problématique de reconstitution d'objets réels dans des environnements arbitraires, le *matting*. La problématique est alors de capturer une distorsion produite par un objet réel, pour un point de vue donné, et de pouvoir replaquer ultérieurement l'objet sur un fond arbitraire, en conservant le même point de vue virtuel. L'objet réel apparaît alors accompagné du nouveau fond, distordu de manière adéquate. Pour réaliser l'acquisition, Zongker *et al.* [190] emploient un certain nombre de clichés de l'objet réel devant des motifs connus assimilables à des codes binaires, puis, par un processus d'optimisation non-linéaire, déterminent pour chaque pixel, la zone de l'environnement qui est visible à travers lui.

6.2.6 Bilan

Si la synthèse du phénomène de réfraction est bien maîtrisée en employant un processus de lancé de rayons, sa simulation reste très délicate dans un contexte interactif. En effet, la faible cohérence qui apparaît entre pixels successifs empêche la bonne intégration de la réfraction, dans le contexte d'un rendu par projection puis remplissage de primitives. Bien que l'introduction de la programmabilité du matériel graphique contribue à pouvoir faire varier légèrement les calculs entre pixels, la réfraction est trop imprévisible, et nécessite une description trop globale de l'objet, pour pouvoir être facilement simulée en chaque pixel.

La démarche de toutes les techniques interactives pour intégrer la réfraction, est alors de la régulariser au maximum, pour retrouver la régularité de calculs recherchée par le rendu par projection. Cette régularisation peut s'opérer soit par des approximations justifiées par le domaine d'application, soit par une ap-

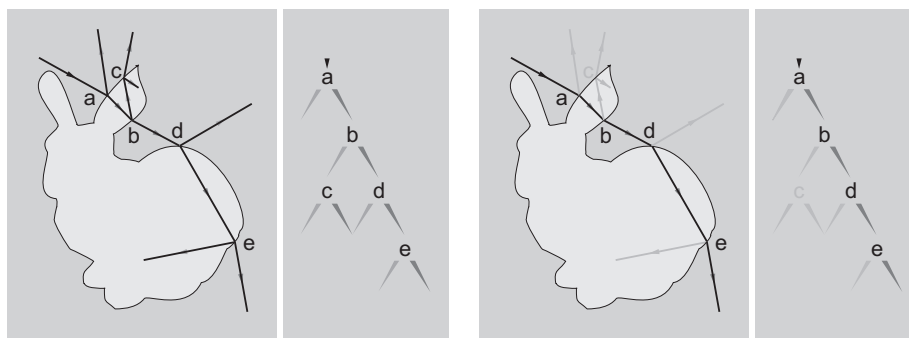


Figure 6.2: Propagation d'un rayon de lumière dans un objet. À gauche : un arbre binaire de rayons se développe à partir d'un unique rayon incident, car n'importe quel rayon lumineux rencontrant n'importe quel dioptre se trouve divisé. À droite : élagage de l'arbre en privilégiant un unique trajet de la lumière à chaque intersection avec un dioptre.

proche hybride pour laquelle le rendu par projection se limite à interpoler une distorsion calculée hors de l'affichage lui-même, soit encore par une capture préalable de la distorsion limitant le rendu à une synthèse de la distorsion cohérente avec le point de vue courant.

La méthode que nous proposons suit cette dernière approche. Le recours à un précalcul permet alors de s'abstraire de la géométrie exacte des objets durant le rendu, ce qui autorise à traiter des géométries arbitraires.

6.3 Principe de la méthode

L'algorithme que nous proposons, ainsi que les données indispensables à sa mise en place vont maintenant être détaillés, en commençant par l'exposition du modèle choisi pour capturer l'influence d'un objet réfringent.

6.3.1 Modèle de réfraction adopté

La description de la propagation de la lumière fait usage du principe de retour inverse de la lumière. Le trajet réel de la lumière est donc rebroussé par les rayons virtuels que l'on va manipuler, depuis les pixels de l'image à former.

On s'intéresse à modéliser l'influence de la présence d'un objet réfringent de surface O dans un environnement et l'on étudie par conséquent le comportement d'un rayon de lumière issu d'un pixel de l'image, lorsqu'il est dévié par un tel objet.

En considérant que n'importe quel rayon incident à n'importe quel dioptre rencontré se trouve divisé, tel que décrit dans la section 6.1, il est aisé de constater qu'à partir d'un rayon initial, un arbre binaire de rayons se développe, comme illustré dans la figure 6.2.

Une simulation exacte et complète de la réfraction requerrait la simulation de tous les rayons présents dans cet arbre. En raison de l'irrégularité et de l'imprévisibilité des rayons qui y apparaissent, seuls les algorithmes de lancé de rayons ont cette capacité — dans une certaine mesure du moins, puisque même

ces algorithmes doivent élaguer des branches pour rendre l'algorithme praticable

La gestion de la totalité de l'arbre n'étant pas possible interactivement, on choisit de procéder à une approximation drastique : qui consiste à n'en conserver qu'une seule branche. Par conséquent, un rayon initial se propage dans la scène en étant éventuellement dévié, mais ne se divise jamais.

Pour choisir la branche privilégiée dans l'arbre, le trajet le plus représentatif est extrait en employant l'heuristique proposée par Hakura et Snyder [63]. À chaque interface, un algorithme glouton choisit la direction de poursuite du trajet entre la direction réfléchie et la direction réfractée. Le choix est déterminé en s'appuyant sur les coefficients d'atténuation des rayons réfléchis et réfractés, déduits des formules de Fresnel. La direction correspondant au rayon le moins atténué est celle retenue. Avec cette approximation, un rayon pénétrant dans un objet réfringent se trouve donc dévié un certain nombre de fois, peut également en sortir temporairement et y rentrer à nouveau, puis finit par le quitter définitivement comme le présente la figure 6.2.

En considérant cette première approximation, et puisque la lumière est alors considérée comme suivant un trajet unique, la déviation de la lumière par un objet peut être représentée par la fonction :

$$\begin{aligned} R_{sp} : O \times S^2 &\longrightarrow O \times S^2 \\ (o_i, \omega_i) &\longmapsto R_{sp}(o_i, \omega_i) = (o_o, \omega_o) \end{aligned}$$

Pour un rayon rencontrant initialement la surface de l'objet au point o_i selon la direction ω_i , la fonction R_{sp} restitue le point de sortie final o_o de l'objet ainsi que la direction de sortie ω_o associée au trajet de la lumière.

La deuxième approximation qui est faite, classique pour les méthodes de rendu interactif, est de supposer que l'objet réfringent peut être complètement découplé de son environnement. Cette hypothèse a deux implications : une fois qu'un rayon quitte l'objet pour rejoindre l'environnement, il le fait définitivement. Ensuite, l'hypothèse autorise à supposer l'environnement de l'objet comme distant, et donc autorise à pouvoir le représenter comme une *environment-map* cubique.

Du point de vue du trajet des rayons lumineux, cette approximation signifie également que la position de sortie de l'objet des différents rayons n'a plus d'importance dans le rendu, seule comptant la direction finale des rayons en sortie des objets réfringents. En tenant compte de cette absence des effets de parallaxe, la déviation de la lumière induite par un objet peut alors être représentée par la fonction :

$$\begin{aligned} R : O \times S^2 &\longrightarrow S^2 \\ (o_i, \omega_i) &\longmapsto R(o_i, \omega_i) = \omega_o \end{aligned}$$

Pour un rayon rencontrant initialement la surface de l'objet au point o_i selon la direction ω_i , la fonction R restitue la direction de sortie ω_o de l'objet associée au trajet de la lumière.

Suivant la même hypothèse de propagation unique des rayons, et pour tenir compte d'une éventuelle absorption de la lumière par un objet semi-transparent, il est également possible de caractériser la distance parcourue dans l'objet par les rayons lumineux. Ceci équivaut donc à la donnée de la fonction

$$\begin{aligned} D_d : O_s \times S^2 &\longrightarrow \mathbb{R} \\ (o_i, \omega_i) &\longmapsto D_d(o_i, \omega_i) = d \end{aligned}$$

qui décrit la distance totale d , parcourue dans l'objet réfringent, par le rayon rencontrant initialement la surface de l'objet au point o_i selon la direction ω_i .

6.3.2 Principe du rendu

Selon ces hypothèses, la réfraction induite par un objet peut être complètement caractérisée comme le champ de distorsion R des trajets lumineux, variable selon la direction d'observation, et défini en chaque point de la surface O .

La stratégie que nous décidons d'employer suit les traces de celle proposée par Heidrich *et al.* [69], et consiste à capturer cette information de distorsion durant une phase de précalcul, préalable au rendu. Les données sont ensuite employées, en conjonction avec le point de vue courant, pour synthétiser interactivement la distorsion induite par l'objet réfringent. Une atténuation de la lumière dépendante de la distance effectivement parcourue par les rayons dans les objets, peut également être appliquée selon le même principe.

Si une telle démarche permet de s'abstraire, durant la restitution interactive de la réfraction, de la géométrie réelle des objets et des perturbations qu'elle induit, elle implique également que la configuration géométrique des objets réfringents est statique.

Pour assurer un rendu interactif, la totalité du rendu est pris en charge par le matériel graphique programmable. Certains choix de conception de la méthode, en particulier ceux liés à l'interpolation spatiale de la distorsion sur la surface de l'objet, découlent de contraintes engendrées par le matériel graphique.

Gestion des données de distorsion

L'emploi d'une telle stratégie implique de mettre au point une représentation des données capturant la distorsion, c'est-à-dire de la fonction R , suffisamment efficace pour autoriser un rendu interactif.

La représentation proposée pour capturer la fonction R , qui est définie sur le domaine à quatre dimensions $O \times S^2$, fait usage d'un découplage analogue à celui proposé par Chen *et al.* [19] dans le contexte du *light field mapping*.

Un ensemble de points-échantillons $O_d = \{o_i\}_i \in O$ est choisi sur la surface O de l'objet, et à chaque échantillon est associée une fonction d'interpolation spatiale $S_{o_i}(p)$, définie sur la surface de l'objet.

Une séparation des variables de la fonction de distorsion est alors employée pour découpler les dimensions spatiales des dimensions directionnelles. La fonction se trouve donc approximée par une somme de produits de fonctions bidimensionnelles :

$$R(p, \omega) \approx \sum_{o_i \in O_d} S_{o_i}(p) \cdot R_{o_i}(\omega)$$

En chaque échantillon $o_i \in O_d$, la fonction R_{o_i} décrit la distorsion de tous les rayons incidents à la surface au point o_i :

$$\begin{aligned} R_{o_i} : S^2 &\longrightarrow S^2 \\ \omega &\longmapsto R(o_i, \omega) \end{aligned}$$

Cette formulation revient donc à interpoler spatialement la distorsion, sur la surface O , la distorsion variable en fonction du point de vue étant connue indépendamment en chaque échantillon o_i .

<p>Précalcul</p> <ul style="list-style-type: none"> P1. Choix de l'échantillonnage de la surface $O_d = \{o_i\}_i$ P2. Échantillonnage directionnel de la fonction de distorsion R_{O_i} en chaque échantillon $o_i \in O_d$, par lancé de rayons P3. Compression de chaque fonction R_{o_i} sous une forme fréquentielle à l'aide d'harmoniques sphériques <p>Rendu interactif</p> <ul style="list-style-type: none"> R1. Calcul de l'environnement dans le cas d'un environnement dynamique R2. Synthèse de la distorsion en accord avec le point de vue courant q, pour tous les échantillons $\{o_i\}_i$: construction de la <i>refraction map</i> $\mathcal{R}_{\rho,q}$ R3. Rendu en employant la distorsion stockée dans la <i>refraction map</i> $\mathcal{R}_{\rho,q}$
--

Figure 6.3: Algorithme global

Dans la mesure où l'objectif est d'aboutir à un rendu interactif, il est naturel de chercher à associer l'interpolation des fonctions $\{S_{o_i}\}_i$ à l'interpolation des textures assurée par le matériel graphique. La section 6.5 discutera le choix des échantillons composant O_d en rapport avec cette exigence. Le choix des fonctions d'interpolation spatiale n'est donc pas libre, mais découle implicitement de la paramétrisation des échantillons dans les textures utilisées par le rendu.

Une représentation fréquentielle, tirant parti des harmoniques sphériques, est employée pour la gestion de l'aspect directionnel de ces informations, c'est-à-dire pour la représentation des fonctions $\{R_{o_i}\}_i$. Dans ce cas, la représentation disponible, intrinsèquement continue, n'oblige pas à recourir à une quelconque interpolation directionnelle durant le rendu. La représentation fréquentielle, qui assure également une forme de compression, est détaillée dans la section 6.4. Il est à noter que ce mode de représentation, en utilisant une base fréquentielle, fournit un mode de compression destructif : la réfraction ne peut donc être restituée parfaitement.

Algorithme général

En adoptant la démarche précédemment décrite, la gestion de la réfraction se décompose en deux phases, toutes deux résumées dans la figure 6.3.

Durant la phase de précalcul, on commence par choisir la position des échantillons composant O_d . Ensuite, pour chacun des échantillons o_i , la distorsion des rayons incidents R_{o_i} est évaluée, pour un échantillonnage complet de l'hémisphère de directions d'incidence. On utilise pour cela un algorithme de lancé de rayons modifié pour suivre l'algorithme glouton de propagation des rayons précédemment décrit. Chaque fonction de distorsion directionnelle ainsi échantillonnée est ensuite projetée sur la base d'harmoniques sphériques, pour en déduire la représentation fréquentielle compressée. On procède de même pour l'information de distance parcourue dans l'objet, dans le but d'inclure une atténuation.

Le rendu interactif opère ensuite en deux phases. La première consiste à synthétiser la distorsion de la lumière à partir des données compressées, en fonction du point de vue courant q et pour tous les points échantillons de la surface de

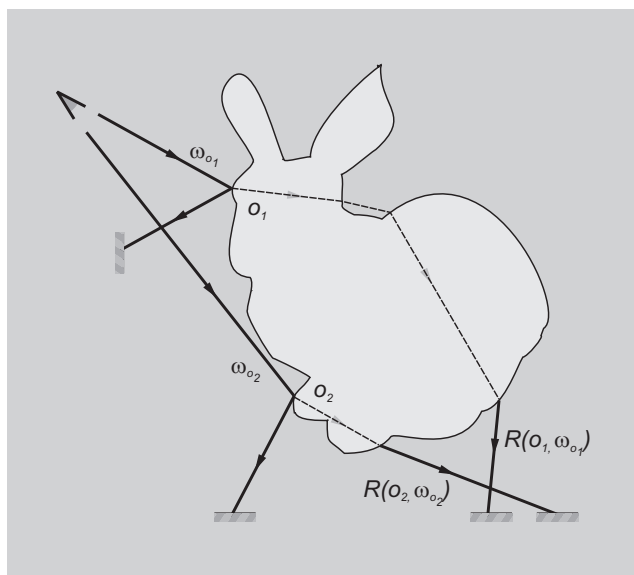


Figure 6.4: Modèle de la situation simulée. Sont combinés un échantillonnage de l'environnement suivant la direction primaire réfléchie, et un échantillonnage de l'environnement suivant la direction finale réfractée. Cette dernière contribution est atténuée suivant la distance effectivement parcourue dans l'objet. Les informations directionnelles de direction finale réfractée et de distance parcourue sont extraites des données compressées représentées par les harmoniques sphériques.

l'objet. Cette étape donne naissance à la *refraction map* $\mathcal{R}_{\rho,q}$, une texture qui stocke ces informations, en usant de la paramétrisation ρ qui fixe le placement des échantillons $\{o_i\}_i$. Comme on le verra ultérieurement, cette paramétrisation est choisie pour faire correspondre l'interpolation spatiale des échantillons sur la surface — les fonctions $\{S_{o_i}\}_i$ — avec l'interpolation matérielle des textures fournie par le matériel graphique. Formellement, la *refraction map* peut être définie comme

$$\mathcal{R}_{\rho,q}(\rho(o_i)) = R_{o_i}(\omega_q) \quad \forall o_i \in O_d$$

La décompression s'accompagne de traitements qui seront détaillés ultérieurement, nécessaires à un rendu de bonne qualité. Cette première phase se déroule sans affichage à l'écran.

Ensuite, la phase de rendu effectif à l'écran est exécutée, dans laquelle l'objet est affiché à l'image. Elle consiste principalement, pour chaque pixel, à échantillonner l'environnement représenté sous forme de *cube-map*, en accord avec la *refraction map* $\mathcal{R}_{\rho,q}$. Une éventuelle atténuation, fonction de la distance de parcours des rayons dans l'objet, peut également être appliquée. Comme illustré dans la figure 6.4, il est également possible de tenir compte du premier rayon réfléchi vers l'environnement, issu de l'intersection du rayon initial avec la surface O , pour plus de réalisme. Cette phase finale du rendu sera détaillée dans la section 6.5.3.

Avant la phase de rendu, la représentation sous forme de *cube map* de l'environnement autour de l'objet réfringent, *via* une série de projections, peut être réalisée si nécessaire. Cette étape n'est requise que si l'environnement est dy-

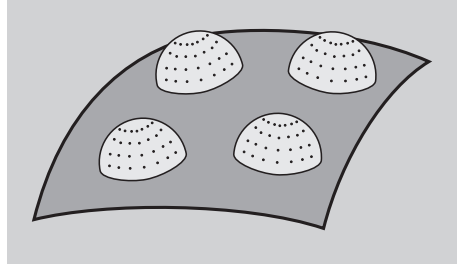


Figure 6.5: Échantillonnage de la fonction de distorsion. L'échantillonnage est double, spatial et directionnel : en chaque échantillon spatial, $o_i \in O_d \subset O$, la fonction est échantillonnée pour un ensemble de directions $\{\omega_{o_i}^j\}_j = S_d^2 \subset S^2$ couvrant l'hémisphère au dessus de la surface.

namique, nécessitant d'être recalculé d'image en d'image. Dans le cas contraire, l'*environment map*, statique, est supposée connue, et cette étape optionnelle disparaît.

Mise en œuvre de l'échantillonnage durant le précalcul

Pour chaque échantillon spatial $o_i \in O_d$, et comme le montre la figure 6.5, un échantillonnage dense de la distorsion des rayons incidents à la surface en ce point est réalisé durant le précalcul. Une fois le rayon incident réfracté à l'intérieur de l'objet, il est propagé selon la stratégie gloutonne de propagation de la lumière décrite lors de la simplification du modèle optique, jusqu'à sortir définitivement de l'objet. La propagation est réalisée grâce à un algorithme de lancé de rayons.

L'échantillonnage de l'hémisphère des directions incidentes emploie un échantillonnage dense et régulier, mais surtout identique pour tous les échantillons de surface $o_i \in O_d$. La constance du motif d'échantillonnage découle de la nécessité d'une phase de compression performante sur la base des harmoniques sphériques, comme on le verra dans la section suivante.

Caractéristiques de l'approche

En résumé, une telle approche de la gestion de la réfraction présente les caractéristiques suivantes :

- L'environnement est supposé infiniment distant de l'objet à représenter, ce qui permet de le représenter par une *environment-map*.
- À l'intérieur de l'objet réfringent, la lumière suit un unique trajet privilégié, trajet déterminé par un algorithme glouton qui choisit à chaque interaction avec un dioptré, le chemin le moins atténué.
- L'atténuation de la lumière traversant l'objet réfringent est dépendante de la distance parcourue dans l'objet.
- La notion d'objet retenue est très large : aucune restriction n'a été posée sur la géométrie de la surface, ni sur le nombre de matériaux de l'objet à représenter, qui peut d'ailleurs présenter plusieurs composantes connexes. Un maillage triangulaire de la surface est supposé connu.

- La géométrie de l'intégralité de l'objet réfringent est statique dans son espace de définition mais peut subir des transformations rigides globales.

Le modèle de la réfraction qui a été posé présente également des limitations, mais dont le préjudice est limité :

- Aucun effet de parallaxe n'est pris en compte, l'environnement étant représenté sous la forme d'un *environment-map*.
- La restriction de la propagation de la lumière à un unique trajet limite la richesse de la restitution des objets.
- Le stockage des informations directionnelles sous une forme fréquentielle fortement compressée altère la réfraction : elle n'est donc restituée qu'approximativement, dans ses grandes lignes et sous une version lissée. En particulier, les discontinuités franches ne peuvent être capturées.
- Si la direction de sortie des rayons lumineux des objets est connue, la position sur la surface de l'objet ne l'est pas : la surface arrière des objets est pour ainsi dire perdue, ce qui y empêche toute application d'un modèle d'illumination local durant le rendu final.

Une dernière limitation, plus conséquente, ne découle pas du modèle adopté pour la réfraction, mais de la stratégie de précalcul mise en œuvre par la méthode : la géométrie des objets et les matériaux qui les composent doivent être statiques. Compte tenu de la quantité de données manipulées, même en ayant recours à une compression, la méthode proposée ne peut gérer que des objets rigides et n'est pas adaptée à l'animation.

6.4 Harmoniques sphériques et fonctions directionnelles

Comme on l'a déjà mentionné, chacune des fonctions directionnelles $\{R_{o_i}\}_i$ associée à un échantillon spatial $o_i \in O_d$ est représentée à l'aide de la base des fonctions harmoniques sphériques. Cette section décrit plus en détail cette base, ainsi que le passage de, et vers, la représentation fréquentielle compressée.

Le choix de cette représentation pour les données directionnelles est dicté par plusieurs facteurs : tout d'abord, par la nature intrinsèquement sphérique de la représentation, ensuite par sa capacité de compression, et même de compression progressive, et enfin, par la parfaite régularité de l'algorithme de décompression qui découle de son utilisation, et qui en autorise une entière prise en charge par le matériel graphique.

On souligne que cet usage des harmoniques sphériques, pour représenter directement une fonction et décompresser des valeurs ponctuelles, ne doit pas être confondu avec celui, plus commun dans le cadre du rendu interactif, qu'en font les techniques relatives au *Precomputed Radiance Transfer* [146]. Celles-ci, qui cherchent à réaliser la convolution d'un environnement lumineux avec un noyau de transfert sphérique emploient avantageusement les représentations par harmoniques sphériques de l'environnement et du noyau pour réaliser la convolution dans le domaine fréquentiel. Grâce aux propriétés d'orthogonalité de la base de fonctions, la convolution peut alors être réduite à un produit scalaire de vecteurs réels, donc être effectuée avec un coût extrêmement réduit.

Dans le cas de la réfraction que nous traitons, aucune convolution n'est mise en œuvre, et il est nécessaire de procéder à une reconstruction explicite des

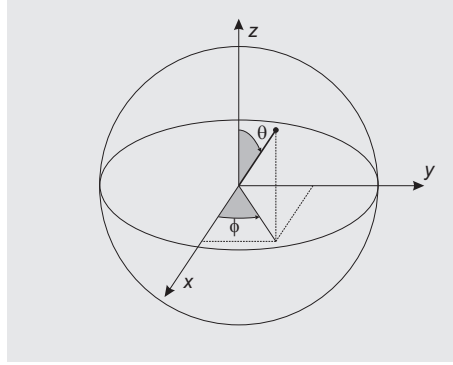


Figure 6.6: Système de coordonnées sphériques employé pour définir les fonctions harmoniques sphériques Y_l^m .

valeurs stockées sous forme fréquentielle. Chaque fonction R_{σ_i} est représentée par elle-même, isolément des autres. Cependant la régularité parfaite de la représentation implique que toutes les fonctions peuvent être traitées en parallèle lors de la décompression. Ceci ouvre la voie à un algorithme interactif lorsque la décompression est réalisée par un matériel graphique programmable.

6.4.1 Définition et propriétés des harmoniques sphériques

Les harmoniques sphériques consistent en une famille, infinie, de fonctions polynomiales de fréquences croissantes, définies sur la sphère S^2 . Ces fonctions, généralement notées Y_l^m , constituent les solutions de la partie angulaire de l'équation de Laplace exprimée en coordonnées sphériques [97].

Elles ont la particularité de former une famille complète de fonctions, et donc de pouvoir servir de support à la décomposition de toute fonction $\varphi \in L^2(S^2)$:

$$\begin{cases} \varphi = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} \alpha_l^m \cdot Y_l^m \\ \alpha_l^m = \int_{S^2} \varphi \cdot Y_l^m ds \end{cases} \quad \forall \varphi \in L^2(S^2)$$

D'une manière analogue à une série de Fourier, pour les fonctions périodiques de \mathbb{R} , toute fonction peut donc être décomposée comme combinaison linéaire des fonctions de base Y_l^m .

Les fonctions Y_l^m , à valeurs complexes, sont doublement indexées par les indices l et m , avec $-l \leq m \leq l$. Comme on peut le remarquer dans la figure 6.7, l'indice l correspond à la fréquence associée à chaque fonction, d'une manière analogue à la fréquence des différences sinusoïdes d'une décomposition de Fourier.

Les fonctions harmoniques sphériques présentent également des propriétés d'orthogonalité :

$$\int_{S^2} Y_{l_1}^{m_1} \cdot Y_{l_2}^{m_2} ds = \delta_{l_1}^{l_2} \cdot \delta_{m_1}^{m_2}$$

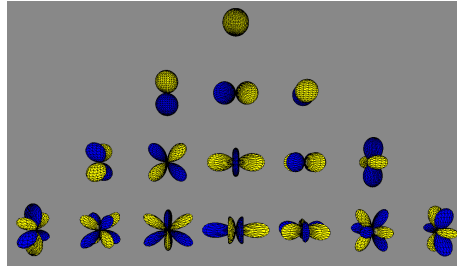


Figure 6.7: Représentation des premières fonctions harmoniques sphériques. Les fonctions pour les fréquences $l = 0, \dots, 3$ sont illustrées, la fréquence allant croissant vers le base de la figure. Les fonctions sont figurées comme des sphères dont le rayon est déformé par la valeur absolue de la fonction à représenter $|Y_l^m|$. Les deux couleurs illustrent la différence de polarité.

En se référant au système de coordonnées sphérique représenté dans la figure 6.6, les fonctions harmoniques sphériques peuvent être définies analytiquement à partir des fonctions trigonométriques usuelles :

$$\begin{cases} Y_l^m = Y_l^m(\theta, \phi) = K_l^m P_l^{|m|}(\cos \theta) e^{im\phi} \\ K_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} \end{cases}$$

avec K_l^m une constante de normalisation. On notera qu'une variante de définition existe, avec l'introduction d'un terme de phase, $(-1)^m$, ici omis. Une définition récursive existe également.

La définition présentée fait usage des polynômes de Legendre associés P_l^m , qui peuvent être définis par les relations de récurrence suivantes [124] :

$$P_l^m : \begin{cases} (l-m)P_l^m(x) = x(2l-1)P_{l-1}^m - (l+m-1)P_{l-2}^m \\ P_m^m(x) = (-1)^m (2m-1)!! (1-x^2)^{\frac{m}{2}} \\ P_{m+1}^m(x) = x(2m+1)P_m^m \end{cases} \quad \begin{cases} 0 \leq m \leq l \\ -1 \leq x \leq 1 \end{cases}$$

où $n!!$ représente le produit de tous les entiers naturels impairs inférieurs ou égaux à n .

Dans la mesure où l'on cherche à produire une implémentation matérielle de la décompression, *i.e.* de l'évaluation des fonctions de base Y_l^m , ces expressions ne sont pas satisfaisantes car elles imposent l'usage de quantités angulaires et de fonctions trigonométriques peu adaptées dans ce contexte. En considérant les points de la sphère, non par leurs coordonnées angulaires mais sous une représentation Cartésienne :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin \theta \cdot \cos \phi \\ \sin \theta \cdot \sin \phi \\ \cos \theta \end{bmatrix} \quad \begin{cases} \theta \in [0, \pi] \\ \phi \in [0, 2\pi[\end{cases}$$

il est possible d'obtenir une expression Cartésienne des fonctions de base :

$$Y_l^m = Y_l^m(x, y, z)$$

$$= K_l^m \sum_{p,q,s} \frac{1}{p!q!s!} \left[-\frac{x+iy}{2} \right]^p \left[\frac{x-iy}{2} \right]^q z^s \quad \begin{cases} x^2 + y^2 + z^2 = 1 \\ p + q + s = l \\ p - q = m \end{cases}$$

Sous cette représentation, les fonctions de base Y_l^m apparaissent sous une forme beaucoup plus simple à calculer avec un matériel graphique programmable : des polynômes de degré l , fonctions des trois variables représentant une direction en coordonnées Cartésiennes.

Puisque seule nous intéresse la représentation de fonctions à valeurs réelles, et non à valeurs complexes, il est possible de définir des fonctions de base purement réelles y_l^m , contreparties des fonctions complexes Y_l^m , par :

$$y_l^m = \begin{cases} \sqrt{2} \operatorname{Re}(Y_l^m) & m > 0 \\ \sqrt{2} \operatorname{Im}(Y_l^m) & m < 0 \\ Y_l^0 & m = 0 \end{cases}$$

Comme précédemment, toute fonction à valeur réelle peut être décomposée en une combinaison linéaire infinie de ces fonctions y_l^m .

Représentation tronquée Dans la mesure où la représentation en série fréquentielle est nécessairement tronquée lors de son utilisation, seule une approximation basse-fréquence $\hat{\varphi}$ de la fonction φ est accessible, les plus hautes fréquences de la fonction codée étant perdues lors de cette troncature :

$$\varphi \approx \hat{\varphi} = \sum_{l=0}^L \sum_{m=-l}^{m=l} w_l^m \cdot Y_l^m \quad w_l^m = \int_{S^2} \varphi \cdot Y_l^m ds$$

En modulant le nombre de fréquences conservées, la qualité de l'approximation $\hat{\varphi}$ varie depuis une fonction constante, si seul le premier coefficient correspondant à $l = 0$ est conservé, pour s'approcher de manière incrémentale de φ au fur et à mesure de l'ajout de fréquences supplémentaires. Les harmoniques sphériques autorisent donc une forme de compression progressive. Il faut toutefois souligner que l'ajout d'une fréquence correspond à l'ajout, non d'une seule fonction, mais de l'ensemble $\{y_l^m\}_{-l \leq m \leq l}$. De fait, le cardinal de la base de fonctions est de l'ordre de $o(l^2)$.

Dans le cas général, la représentation par une série tronquée ne fournit qu'une approximation de la fonction initialement considérée. Cette approximation est continue puisque constituée d'une somme de fonctions continues : il n'est donc pas possible d'encoder de discontinuités.

La troncature de la série provoque en outre des oscillations du signal reconstruit à proximité des zones de discontinuité de la fonction codée, oscillations connues sous le nom de phénomène de Gibbs. Un moyen d'atténuer ce problème, par un double lissage, fréquentiel et spatial, sera présenté ultérieurement.

Invariance rotationnelle Une propriété intéressante, dans l'objectif de représenter une fonction par sa projection sur la base des fonctions harmoniques

sphériques, est l'invariance rotationnelle assurée par cette famille de fonctions. Même si sa représentation en série est tronquée, une fonction sera représentée avec la même exactitude quelle que soit la rotation que l'on peut lui faire subir par rapport au système de coordonnées. En particulier, ceci signifie que l'on est libre de choisir l'orientation du système de coordonnées, sans altérer aucunement le pouvoir de représentation des harmoniques sphériques.

6.4.2 Compression

Dans le cadre visé par l'application, on souhaite convertir la représentation échantillonnée de chaque fonction R_{o_i} en sa représentation fréquentielle, ce qui, d'après la description précédente, revient à calculer les différents coefficients w_l^m , chacun étant défini comme une intégrale sur la sphère.

Une technique standard pour évaluer ces intégrales est de recourir à une intégration de Monte-Carlo. L'inconvénient est que la recherche de résultats précis avec une telle oblige à des intégrations très coûteuses. Dans l'application visée, réaliser une telle compression pour chaque échantillon $o_i \in O_d$ n'est pas envisageable, du fait du nombre de fonctions à compresser.

Nous proposons donc une technique de compression plus performante, sous réserve d'utiliser un échantillonnage de l'hémisphère d'incidence identique pour tous les échantillons $o_i \in O_d$. Soit un ensemble fini de directions d'échantillonnage $S_d^2 \subset S^2$. La valeur de la fonction R_{o_i} est connue pour chacune de ces directions $\omega_j \in S_d^2$, et supposée interpolée entre les échantillons à l'aide de fonctions d'interpolation B_j . Par conséquent, la fonction continue R_{o_i} , connue au travers de ses échantillons $\{R_{o_i}(\omega_j)\}_{\omega_j \in S_d^2}$, s'écrit sous la forme :

$$\begin{aligned}\widehat{R}_{o_i}(\omega) &= \widehat{R}_{o_i}(\theta, \phi) = \sum_j r_j \cdot B_j(\theta, \phi) \\ r_j &= R_{o_i}(\omega_j)\end{aligned}$$

En introduisant cette décomposition dans la définition des coefficients des harmoniques sphériques, il vient, par linéarité :

$$w_l^m = \int_{S^2} \left[\sum_j r_j \cdot B_j \right] \cdot y_l^m ds = \sum_j r_j \cdot \int_{S^2} B_j \cdot y_l^m ds$$

On pose maintenant une indexation linéaire des différents termes liés aux harmoniques, par la relation suivante :

$$i = l(l+1) + m \quad \begin{cases} i, l, m \in \mathbb{N} \\ -l \leq m \leq l \end{cases}$$

et on définit, pour une représentation jusqu'à l'ordre l et un échantillonnage directionnel de n points, les deux vecteurs W et R contenant respectivement tous les coefficients recherchés de la décomposition fréquentielle, et tous les échantillons :

$$\begin{aligned}W &= [w_i]_i = [w_0^0, w_1^{-1}, w_1^0, w_1^1, \dots, w_l^{-l}, \dots, w_l^l]^T \\ R &= [r_j]_j = [r_0, \dots, r_n]^T\end{aligned}$$

La projection complète d'une fonction s'exprime alors sous la forme du produit matrice-vecteur :

$$W = M \cdot R \quad \text{avec } M = [M]_{ij} = \int_{S^2} y_i \cdot B_j \, ds$$

L'intérêt d'une telle écriture est de séparer les données à compresser, le vecteur R , du schéma d'échantillonnage qui se trouve entièrement localisé dans la matrice M . À condition d'utiliser le même échantillonnage pour toutes les fonctions à compresser, c'est-à-dire pour toutes les fonctions $\{R_{o_i}\}_i$, la même matrice M peut être employée pour toutes les compressions. Elle peut par conséquent être calculée précisément, une seule fois pour toutes les compressions à effectuer. La compression effective de n'importe quelle fonction échantillonnée se réduit ensuite à une simple multiplication matrice-vecteur.

Pour calculer cette matrice, nous avons employé une intégration de Monte-Carlo très précise. Le motif d'échantillonnage de l'hémisphère des directions incidentes choisi est régulier, dérivé de la paramétrisation par deux angles de l'hémisphère. Les fonctions d'interpolation associées correspondent aux fonctions d'interpolation bilinéaires sur les quadrangles sphériques engendrés par une telle discrétisation.

N'importe quel échantillonnage peut cependant être employé : dans ce cas, on propose de calculer une triangulation de l'hémisphère à partir des échantillons et de dériver les fonctions d'interpolation linéaires sur chaque triangle sphérique ainsi défini. La triangulation sphérique de l'hémisphère est obtenue par un calcul d'enveloppe convexe.

Pour employer le même schéma d'échantillonnage de l'hémisphère, quel que soit l'échantillon $o_i \in O_d$, l'échantillonnage directionnel S_d^2 est réalisé en considérant le repère local tangent à chaque échantillon spatial $o_i \in O_d$. Chaque fonction échantillonnée l'est donc dans son propre repère local, ce dont il faudra tenir compte lors de la décompression interactive. En raison de l'invariance rotationnelle d'une représentation par harmoniques sphériques, le choix de ce repère n'engendre aucune différence dans la compression par rapport au repère canonique.

Le choix de conserver ce repère, et de ne pas réaligner chaque fonction compressée avec le repère canonique, est dû à la difficulté de calculer les matrices appliquant des rotations aux coefficients d'harmoniques sphériques. Sous réserve de pouvoir les calculer, il est évidemment possible, et même avantageux, de procéder au réalignement de toutes les fonctions compressées dans un même repère.

On notera qu'il existe une corrélation entre nombre d'échantillons de la fonction à compresser, et fréquence de la représentation, de manière analogue à la relation décrite par théorème de Nyquist. Dans tous les cas, et comme le montre la figure 6.8, il est nécessaire d'employer un nombre conséquent d'échantillons afin de capturer correctement la fonction à représenter.

Une dernière remarque quant à la compression concerne le fait que seules sont échantillonnées des données sur l'hémisphère d'incidence extérieur à l'objet. On choisit de compléter l'hémisphère interne par une symétrie des données par rapport au plan équatorial de la sphère. Bien que d'autres stratégies soient envisageables, Sloan *et al.* [145] ont montré qu'il s'agit de la complétion simple la plus intéressante du point de vue de la précision.

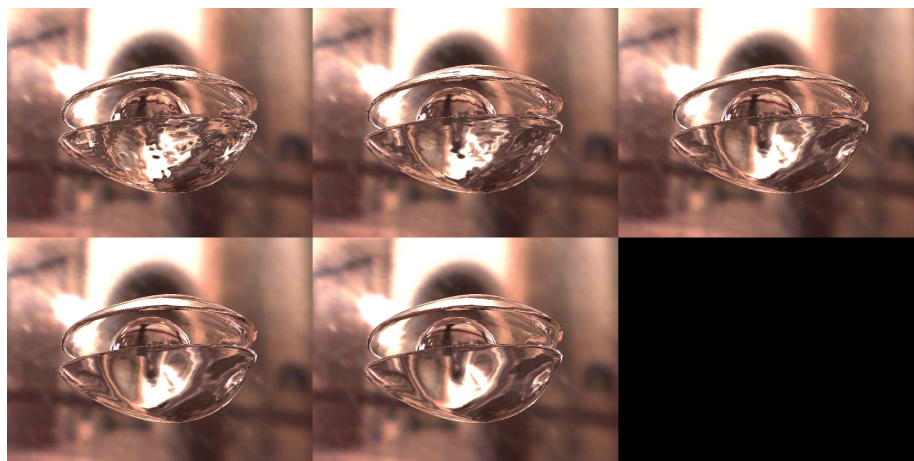


Figure 6.8: Influence de l'échantillonnage directionnel. Le même objet est compressé en faisant varier la densité de l'échantillonnage directionnel S_d^2 , toujours généré par une subdivision angulaire régulière de l'hémisphère d'observation. Les subdivisions présentées correspondent à (8, 16), (16, 32), (32, 64), (64, 128), et (128, 256) subdivisions des angles (θ, ϕ) . La compression est réalisée jusqu'à l'ordre $l = 8$. Plus le nombre d'échantillons est élevé, meilleur est le résultat, mais l'amélioration n'est quasiment plus perceptible au-delà de la subdivision (32, 64).

Résultat de la compression Pour chaque point d'échantillonnage spatial $o_i \in O_d$, la fonction de distorsion $R_{o_i} : S^2 \rightarrow \mathbb{R}^3$, représentée en coordonnées Cartésiennes : $[x(\omega), y(\omega), z(\omega)]$, est compressée selon ce principe, chaque coordonnée l'étant indépendamment. La représentation Cartésienne est préférée à une représentation angulaire plus compacte, car elle ne présente pas de problème de définition à proximité des pôles de la sphère. La fonction scalaire de distance parcourue dans l'objet, $D_{o_i} : S^2 \rightarrow \mathbb{R}$ est également compressée selon le même processus.

Compressées, ces deux fonctions sont donc représentées par 4 vecteurs de coefficients d'harmoniques sphériques. Du point de vue de l'implémentation, toutes ces données sont stockées dans différentes textures qui seront employées lors de la décompression par le matériel graphique. Les quatre coefficients associés à un même point-échantillon $o_i \in O_d$ sont groupés dans les différentes composantes RGBA des pixels, la distorsion $[x, y, z]$ dans les composantes RGB, et l'atténuation dans la composante A.

Comme l'illustre la figure 6.9, une texture différente est associée à chaque fonction de base y_i , et regroupe tous les coefficients associés à cette fonction, pour tous les échantillons $o_i \in O_d$. Le placement des données dans toutes ces textures suit une paramétrisation unique μ , dont la définition sera discutée dans la suite. Le placement exact n'a pas d'importance durant la décompression : le jeu de textures utilisé est simplement vu comme un tableau parallèle des différents coefficients de la décomposition fréquentielle, une texture étant associée à une fonction de base donnée. Le résultat de la procédure de décompression sera également formaté selon la même paramétrisation μ .

Bien que des formats de stockage en nombres flottants des textures puissent être utilisés, nous n'avons pas constaté d'altération des résultats si une quantifi-

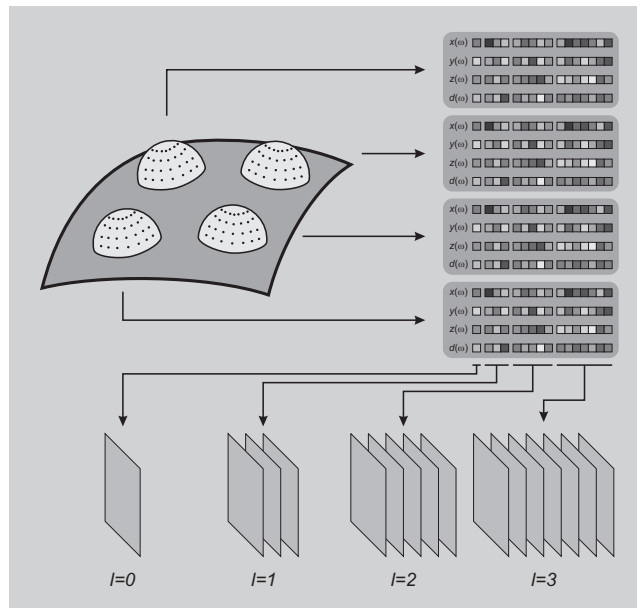


Figure 6.9: Données compressées. Chaque échantillon $o_i \in O_d$ engendre quatre séries de coefficients d'harmoniques $\{w_j^{o_i}\}_j$, codant la direction de sortie de la lumière et la distance parcourue dans l'objet. Tous les coefficients $\{w_j^{o_i}\}_{o_i \in O_d}$ correspondant à une fonction de base y_i sont groupés dans une texture, les coefficients des quatre séries étant associés aux quatre composantes de couleur.



Figure 6.10: Influence de la quantification des coefficients d'harmoniques. L'image de gauche correspond à un stockage des coefficients avec un format en virgule flottante comportant 32 bits par composante, l'image de droite à un stockage des coefficients avec une quantification des coefficients sur 8 bits. Les résultats sont parfaitement similaires.

cation des données sur 8 bits par composante est employée, comme en atteste la figure 6.10. La quantification est séparée pour chaque texture, *i.e.* pour chaque ensemble de coefficients $\{w_j^{o_i}\}_{o_i \in O_d}$, et réalisée en fonction de la plage des valeurs à stocker. C'est donc un format entier quantifié qui sera préféré au stockage en virgule flottante en raison du gain d'espace-mémoire qui en résulte. En raison de contraintes qu'ils induisent sur les dimensions des textures, les formats compressés de textures n'ont pas été testés. En résultat de la compression des données de distorsion et d'atténuation jusqu'à l'ordre l , $(l + 1)^2$ textures sont donc générées. La taille des textures est fonction du nombre d'échantillons choisis sur la surface de l'objet, c'est-à-dire du cardinal de O_d .

6.4.3 Décompression assistée par le matériel

Les données compressées ainsi définies sont employées durant le rendu, pour synthétiser la distorsion correspondant au point de vue courant. Cette section se focalise sur le problème de la reconstruction de cette distorsion, en dehors des autres traitements effectués. On reviendra ultérieurement sur ces autres traitements, et sur l'insertion de la décompression dans le mécanisme global du rendu.

Le problème qui est détaillé ici est donc la construction de la *distortion map* $\mathcal{D}_{\mu,q}$, correspondant au point de vue q , stockée selon la paramétrisation μ , paramétrisation identique à celle des données. La *refraction map* $\mathcal{R}_{\rho,q}$ utilisée pour le rendu final en découle, par le biais des traitements décrits ultérieurement.

L'utilisation d'un arrangement μ des échantillons $o_i \in O_d$ commun au résultat de la décompression, aux données, ainsi qu'à tous les tampons intermédiaires, autorise à s'abstraire de sa définition. Dans cette phase, au sein de laquelle les rendus sont effectués dans des textures et ne sont pas affichés à l'écran, le matériel graphique programmable est employé comme une machine SIMD à large bande passante.

Reconstruire la distorsion revient à calculer la somme pondérée des fonctions harmoniques, pour chaque échantillon $o_i \in O_d$, avec ses propres coefficients $\{w_j^{o_i}\}_j$, et pour sa direction d'observation locale $\omega_{o_i}(q)$:

$$\widehat{R}_{o_i}(\omega_q) = \sum_k w_k^{o_i} \cdot y_k(\omega_{o_i}(q)) \quad \omega_{o_i}(q) = \frac{d}{|d|}, \quad d = T_{o_i} \cdot q - o_i$$

La transformation T_{o_i} , rigide, correspond à la transformation associée au repère tangent à la surface en o_i , repère dans lequel ont été définies les différentes directions d'échantillonnage de la fonction de distorsion.

Comme on peut le remarquer, le calcul est rigoureusement identique pour tous les échantillons, seules varient les valeurs de pondération des fonctions de base et les transformations. La tâche principale consiste en l'évaluation des fonctions harmoniques sphériques $\{y_k\}_k$, avec des variables différentes. Pour cela, la forme Cartésienne de ces fonctions est employée.

L'évaluation de cette somme pondérée est réalisée grâce au matériel graphique programmable, simultanément pour tous les échantillons $o_i \in O_d$, en réalisant les calculs dans un *pixel shader*, et en rendant les résultats dans une texture. Dans la mesure où cette somme comporte un grand nombre de fonctions à évaluer et accède à un grand nombre de textures, et en raison des limitations techniques du matériel graphique, il n'est pas possible de réaliser le calcul en une

```

Si première passe
  s = 0
Sinon
  s = extraction de la somme partielle d'une texture tampon intermédiaire

[x, y, z] = calcul de la direction d'observation locale  $\omega_{o_i}(q)$ 

Génération des puissances  $[x^j, y^j, z^j]$  nécessaires aux polynômes

Pour chaque polynôme  $y_i$  assigné à la passe
   $p_i$  = évaluation de  $y_i(x, y, z)$ 
   $w_i$  = extraction du coefficient associé à  $y_i$  depuis une texture et déquantification
   $s = s + w_i \cdot p_i$ 

Si dernière passe
  Quantification de la direction de sortie
  Transformation de la distance décompressée en atténuation

Emission de  $s$  comme sortie

```

Figure 6.11: Structure des *shaders* de décompression. Ces *shaders* sont générés dynamiquement, les fonctions de base étant distribuées de manière à minimiser le nombre de passes. La distribution est assurée par un algorithme glouton. Les instructions figurées en italique correspondent aux méta-instructions de génération, les autres aux instructions incluses dans les *shaders*.

seule passe de rendu. Il est alors scindé en plusieurs passes, en partitionnant la somme à calculer en autant de morceaux que nécessaire pour que chacun d'eux puisse être traité, en fonction des limites techniques imposées par le matériel.

Afin d'assurer un rendu le plus efficace possible, en limitant le nombre de passes générées, le code de calcul de ces passes de calcul n'est pas généré manuellement mais est produit automatiquement, par un algorithme de génération de *shaders*. Un algorithme glouton y distribue les différentes fonctions de base à évaluer entre les passes, en tentant de les remplir au maximum. La sélection des fonctions de base se fait de manière décroissante sur les indices des fonctions de bases polynomiales : plus grand est l'indice de la fonction de base, plus élevé est, *a priori*, le degré du polynôme qu'elle représente.

L'anatomie des *shaders* générés, qui reste simple, est présentée dans la figure 6.11. Pour générer le code d'une passe, les polynômes de base $\{y_i\}_i$ sont d'abord calculés symboliquement, puis le générateur de code produit le code d'évaluation des polynômes comme une somme de monômes. Pour permettre une évaluation plus efficace des fonctions de base, les différentes puissances des variables sont calculées une unique fois en préambule au calcul des monômes.

Pour rester compatible avec une approche multi-passe, deux textures intermédiaires servent de tampons entre les passes. L'une stocke la somme partielle intermédiaire résultant des calculs précédents, et permet le passage de la somme intermédiaire de la passe $n-1$ à la passe n . L'autre représente la somme partielle de destination, utile pour fournir le résultat de la passe n à la passe $n+1$. Ces deux textures sont échangées à chaque passe. L'impossibilité de lire et d'écrire

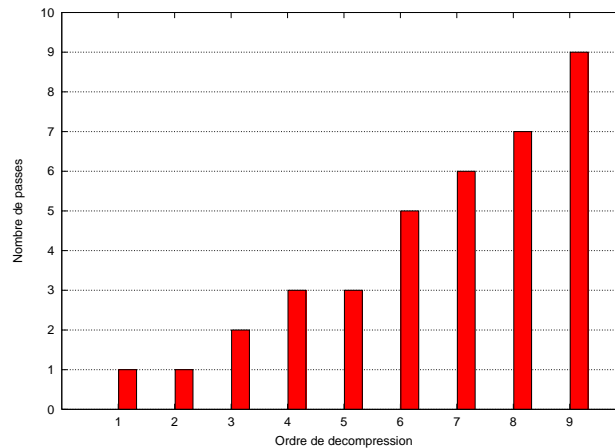


Figure 6.12: Nombre de passes requises par la décompression. Le graphique présente le nombre de passes nécessaires à la décompression, en fonction de l'ordre de décompression, pour une architecture-cible de type DirectX 2.0b. Selon la nature exacte de matériel graphique, il est possible d'employer une décompression jusqu'à l'ordre $l = 8$, ou $l = 9$.

simultanément dans une même texture justifie leur raison d'être. Afin de ne pas altérer les résultats, et pour autoriser autant de passes que nécessaire, elles emploient un format de données en virgule flottante.

La figure 6.12 présente le nombre de passes nécessaires à la décompression en fonction de l'ordre de décompression demandé. Pour un ordre l de représentation, $(l + 1)^2$ fonctions de base doivent être évaluées. La décompression maximale que l'on parvient à atteindre correspond à $l = 8$ ou $l = 9$, selon les limitations du matériel graphique employé, pour une architecture-cible de type DirectX 2.0b.

Dans la mesure où les mêmes fonctions de base sont employées pour toutes les fonctions, et en raison de la nature vectorielle des calculs, la décompression des quatre fonctions stockées dans les différentes composantes de couleur n'est pas plus onéreuse que la reconstruction d'une seule fonction.

Pour rendre des pixels, c'est-à-dire pour déclencher l'exécution de calculs par les *pixel shaders*, deux cas de figure existent, qui sont illustrés dans la figure 6.13. Le choix de l'un ou de l'autre dépend de l'approche choisie pour le rendu, approches qui seront détaillées dans la section 6.5. Dans le premier cas, qui correspond à une approche garantissant la continuité, la paramétrisation des échantillons est telle que la texture est remplie d'échantillons : le rendu s'opère alors sur la totalité des textures, mettant à jour tous les texels puisque tous, ou presque, correspondent à un échantillon $o_i \in O_d$ pour lequel un calcul est à réaliser. Dans le deuxième cas, la position des échantillons dans les textures découle du dépliage d'un maillage, et tous sont situés sous l'empreinte du maillage dans la texture. Dans ce cas, afin d'éviter de consommer des ressources dans les zones inutilisées de la texture, les calculs ne sont effectués que dans l'empreinte, en rendant le maillage dans la texture, avec pour position des sommets leurs coordonnées de texture.

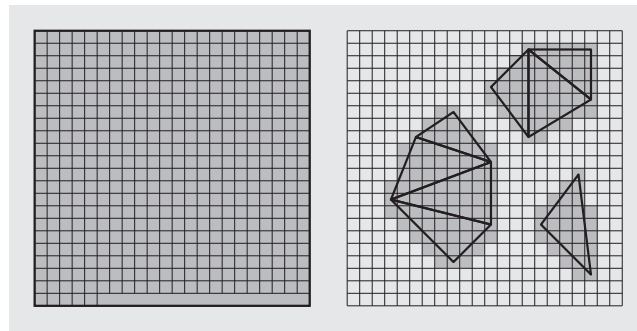


Figure 6.13: Supports des calculs. À gauche est représenté le cas d'un stockage destructuré des échantillons : les calculs sont exécutés sur l'intégralité de la texture, par le rendu d'un carré la couvrant en totalité. La zone non utilisée de la texture est très faible car elle comprend au plus une ligne. À droite est représenté le cas où les échantillons restent structurés et groupés sous l'empreinte du maillage de surface déplié. Le rendu, effectué en s'appuyant sur le maillage, ne touche alors que les zones pertinentes de la texture et en ignore les zones inutilisées.



Figure 6.14: Effets du lissage. Le même objet est présenté, de gauche à droite, sans aucun lissage, avec un lissage fréquentiel, et avec un lissage fréquentiel combiné à un lissage spatial à trois points. La disparition de l'apparence bosselée de la surface est déjà notable avec le lissage fréquentiel, d'un coût nul. L'application supplémentaire d'un lissage spatial autorise une réfraction d'aspect légèrement plus lisse.

6.4.4 Lissage

Comme les données de distorsion décompressées déterminent les directions d'échantillonnage de l'environnement, leur régularité conditionne directement l'aspect, lisse ou non, du rendu. La compression de la distorsion étant réalisée par une technique destructive, un rendu direct à partir des données brutes décompressées peut adopter une apparence légèrement bosselée non souhaitée, comme illustré dans la figure 6.14. Pour conférer aux résultats visuels l'aspect lisse attendu, on propose d'appliquer deux filtres de lissage différents.

Lissage fréquentiel

Comme on l'a déjà mentionné, un des défauts d'une compression fréquentielle est l'apparition du phénomène de Gibbs. Ce phénomène, dû à la troncature de la représentation en série et à la perte des hautes fréquences, se traduit par

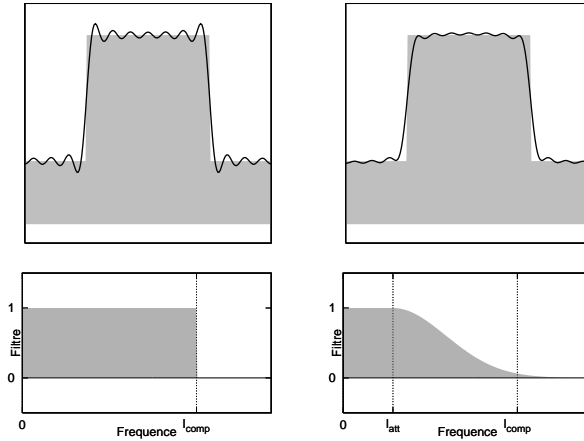


Figure 6.15: Phénomène de Gibbs. Signal original et reconstruit, selon deux filtres, l'un en créneau, l'autre avec une atténuation semi-Gaussienne. Des oscillations parasites apparaissent à proximité des discontinuités du signal, lors de la troncature de la représentation fréquentielle. Les oscillations sont de moindre amplitude avec le filtre semi-Gaussien.

des oscillations parasites du signal reconstruit à proximité des zones de fortes variations du signal, comme le montre la figure 6.15.

Pour diminuer ces parasites, il est possible d'atténuer progressivement les coefficients de la décomposition de la série fréquentielle plutôt que de les tronquer abruptement. Cette opération revient à appliquer dans le domaine fréquentiel un filtre semi-Gaussien passe-bas au signal, plutôt qu'un filtre en créneau.

Le filtre appliqué aux coefficients de la décomposition w_l^m possède alors la forme suivante, en fonction de l'indice de fréquence l :

$$F_{l_{att},\sigma}(l) : \begin{cases} 1 & \text{si } l < l_{att} \\ e^{-(l-l_{att})^2/\sigma^2} & \text{si } l_{att} \leq l \leq l_{comp} \\ 0 & \text{si } l_{comp} < l \end{cases}$$

L'atténuation est nulle pour les fréquences inférieures à l_{att} , puis prend la forme d'une atténuation semi-Gaussienne. L'atténuation est implicitement totale pour les fréquences tronquées de la série, supérieures à l_{comp} . L'effet d'un tel lissage est illustré dans la figure 6.15. Les paramètres l_{att} et σ qui contrôlent la forme exacte de l'atténuation sont laissés à la discrétisation de l'utilisateur, pour qu'il puisse équilibrer la réduction du phénomène de Gibbs par rapport à un lissage excessif, comme le proposent Westin *et al.* [178].

Un des principaux intérêts d'un tel lissage est qu'il se révèle d'un coût complètement nul, même si on désire conserver la possibilité de l'appliquer interactivement lors de la décompression. En effet, l'application du filtre aux coefficients w_l^m non filtrés de la série peut être combinée aux facteurs de déquantification des données, en les modifiant de manière adéquate.

Lissage spatial

Une seconde forme de lissage peut également être appliquée, mais cette fois sous la forme d'un lissage spatial des données décompressées. Au contraire du filtrage fréquentiel précédent, ce filtrage spatial possède un coût d'application non-nul, qui diffère selon la paramétrisation des échantillons $o_i \in O_d$ dans les textures.

Dans le cas d'une paramétrisation structurée dans l'espace de texture, correspondant à l'approche que l'on qualifiera ultérieurement de directe, la structure de voisinage entre échantillons est conservée lors du passage de l'espace de la surface de l'objet à l'espace des textures. Le filtre de lissage est, dans ce cas, appliqué par une convolution locale des données décompressées avec un filtre de lissage Gaussien, directement dans l'espace des textures, en considérant comme échantillons voisins sur la surface les échantillons voisins dans la texture.

Dans le cas d'une paramétrisation déstructurée, correspondant à l'approche choisie pour obtenir une garantie de continuité, la structure de voisinage des échantillons n'est pas préservée lors du passage de la surface de l'objet à l'espace des textures. Dans ce cas, pour chaque échantillon, ses voisins sont identifiés ainsi que leurs distances, durant la phase de précalcul. Ces informations de voisinage sont ensuite codées sous la forme de textures d'indirections, qui permettent de repérer les voisins de chaque échantillon dans l'espace des textures. Le filtre de lissage Gaussien procède alors à une opération de *gather* explicite en employant les textures d'indirection précalculées, qui lui sont fournies. Une texture d'indirection étant nécessaire pour chacun des échantillons appartenant au support du filtre, un filtre de lissage à n points requiert n textures.

On note que ce lissage spatial pourrait être réalisé à coût nul, contrairement à l'implémentation que nous avons décrite. En effet, ce lissage se réduit à des combinaisons linéaires des données décompressées. Or, la reconstruction à partir des coefficients d'harmoniques w_l^m est elle-même linéaire, ce qui signifie qu'il serait possible de substituer au lissage spatial des données décompressées effectué *a posteriori*, un lissage spatial *a priori* des coefficients w_l^m . Ceci induit toutefois deux contraintes : d'une part les coefficients originaux étant modifiés, le filtrage est définitif et ne plus être modifié interactivement, et d'autre part toutes les fonctions harmoniques doivent être définies dans le même repère, pour que les coefficients correspondant aux différents échantillons spatiaux $o_i \in O_d$ soient compatibles entre eux.

6.5 Algorithme de rendu

L'usage des données directionnelles regroupées dans la *distortion map* $\mathcal{D}_{\mu,q}$, dont on a présenté la construction dans la section précédente, ne peut être direct. En effet, il est compliqué par les limitations dues à la rigidité de la programmation du matériel graphique, et en particulier la rigidité de l'interpolation des textures.

Le problème particulier qu'il reste à régler, pour obtenir des rendus de bonne qualité, est de parvenir à plaquer l'interpolation spatiale de la distorsion entre les échantillons $o_i \in O_d$ sur l'interpolation des textures disponibles dans le matériel graphique. Cette étape est incontournable pour obtenir un rendu aux performances satisfaisantes.

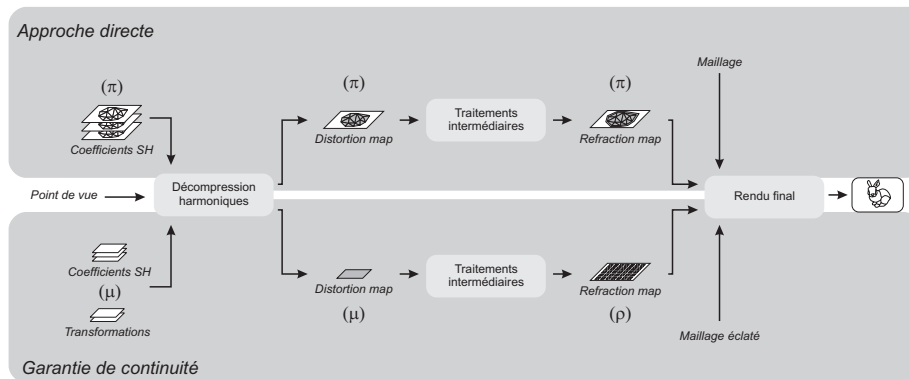


Figure 6.16: Architecture du rendu. Quelle que soit l'approche choisie, la décompression de la *distortion map* prend place initialement. Des traitements dépendants de l'approche choisie s'appliquent ensuite, pour donner naissance à la *refraction map*, qui est employée lors du rendu final. La paramétrisation des textures est indiquée entre parenthèses.

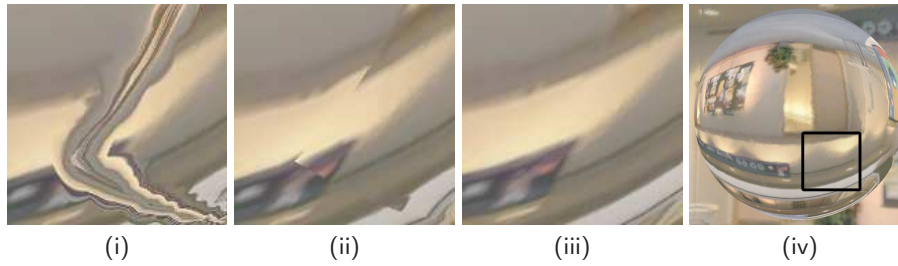


Figure 6.17: Comparaison de la continuité de l'interpolation. Les deux images de gauche (i) et (ii) illustrent respectivement les résultats produits par l'approche directe, sans, et avec, une gouttière dynamique dans la texture. L'image (iii) illustre le résultat qu'il est possible d'obtenir grâce à la technique avec garantie de continuité. L'image (iv) situe la position des trois autres images par rapport à l'objet employé pour le test, une sphère.

La difficulté est alors de parvenir à un rendu offrant le moins de discontinuités possibles dans la réfraction, ou mieux encore sans discontinuité aucune. Ces ruptures visuelles éventuelles sont les résultats de paramétrisations elles-mêmes discontinues de la surface de l'objet dans l'espace de la texture.

Nous proposons deux approches pour faire usage de la distorsion lors du rendu, toutes deux illustrées dans la figure 6.16, dont l'objectif est de pallier ces problèmes de discontinuité visuelle.

La première, d'un emploi plus simple et plus souple, permet l'exploitation des données décompressées en limitant fortement l'apparition de discontinuités, mais sans pouvoir les supprimer totalement. Elle sera présentée la première dans la section 6.5.1. La seconde, plus restrictive, mais capable de garantir une parfaite continuité dans l'interpolation, sera décrite dans la section 6.5.2. L'illustration de résultats obtenus par ces deux méthodes est disponible dans la figure 6.17.

Toutes deux ont pour rôle de traiter les données décompressées brutes de la *distortion map* $\mathcal{D}_{\mu,q}$, pour fournir la *refraction map* $\mathcal{R}_{\rho,q}$ utilisée dans le rendu

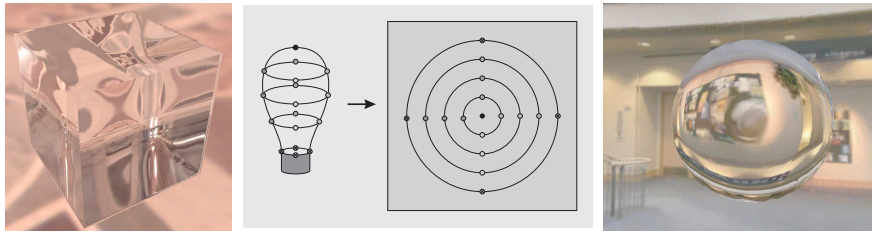


Figure 6.18: Discontinuités acceptables dans la réfraction. Trois cas où la technique directe peut être appliquée, et où les discontinuités qui surviennent ne sont pas gênantes : en confondant discontinuités parasites et discontinuités naturelles, en utilisant une paramétrisation adaptée, pour le cas de surfaces à bords, ou en jugeant acceptables les discontinuités faibles.

final. Ce rendu final, qui réalise l’affichage de l’objet, sera présenté ensuite dans la section 6.5.3.

Tous les traitements appliqués par l’une ou l’autre approche sont entièrement accélérés par le matériel graphique. Ils agissent sur les données décompressées associées aux différents échantillons et correspondent à des traitements dans l’espace des textures, sans affichage à l’écran.

C’est dans les deux sections suivantes que l’on discutera également l’arrangement des échantillons $o_i \in O_d$ sur la surface de l’objet, en liaison avec les paramétrisations μ et ρ .

6.5.1 Approche directe

La première approche proposée est applicable si seule une continuité imparfaite de l’interpolation de la réfraction est requise sur la surface de l’objet. Ce cas de figure n’est pas utopique et peut survenir dans les cas illustrés dans la figure 6.18, à savoir quand une discontinuité naturelle existe, induite par la géométrie de l’objet comme pour les arêtes du cube ; quand les discontinuités de paramétrisation de la texture peuvent être masquées, par un choix de paramétrisation judicieux réalisé par un artiste, comme dans le cas d’une ampoule ; ou tout simplement quand quelques imperfections visuelles sont jugées tolérables.

Une caractérisation plus précise correspond aux objets dont chaque partie de la surface, sur laquelle la réfraction doit être continue, peut être paramétrée en 2D par une paramétrisation telle qu’aucun sommet ne soit dupliqué — hors cas particulier d’une éventuelle périodicité acceptable si l’échantillonnage des paires d’arêtes dupliquées est rigoureusement identique — La continuité de la réfraction sur les arêtes par lesquelles passe le découpage de la surface n’est alors pas garantie. Pour une telle paramétrisation, la topologie de chaque partie du maillage est préservée lors de sa projection dans l’espace des textures.

Dans le cas d’une telle paramétrisation π , les positions des échantillons de surfaces $o_i \in O_d$ sont choisies comme les centres des texels engendrés par π . L’échantillonnage de la surface de l’objet est alors totalement contrôlé par π , et peut être aussi bien uniforme que non-uniforme. On notera également que les relations d’adjacence entre échantillons sont préservées lors du passage de l’espace de la surface O à l’espace de la texture $\pi(O)$, sauf sur les bords des différents morceaux de surface.

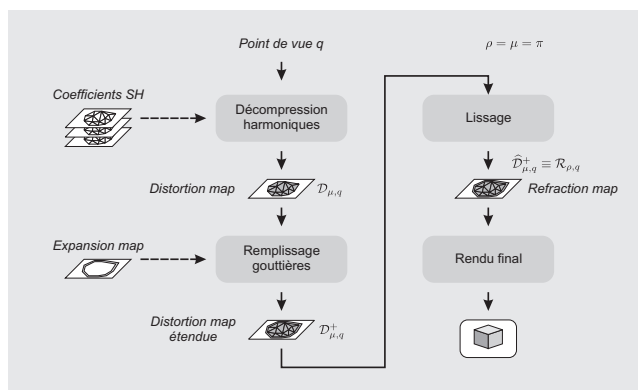


Figure 6.19: Étapes du rendu pour l'approche directe.

Dans ce cas de figure, la paramétrisation π , fournie par l'utilisateur, peut être également choisie pour ρ et μ : $\rho = \mu = \pi$. La même disposition est alors utilisée de bout en bout du rendu, depuis le stockage des données compressées w_l^m , jusqu'à la *refraction map* finale, résultant en une approche simple.

La conservation des topologies du maillage et des échantillons, lors du passage de la surface de l'objet aux textures présentent plusieurs avantages.

Tout d'abord, les échantillons sont groupés sous l'empreinte du maillage dans l'espace de la texture, et leur mise à jour est possible par un rendu des triangles du maillage original, en utilisant pour positions des sommets leurs coordonnées de texture. Seules les zones des textures contenant des échantillons valides sont alors touchées, et aucune ressource de calcul n'est perdue en exécutant un *shader* dans les zones non-employées de la texture.

Ensuite, il est possible de reporter le calcul des directions d'observation $\omega_{o_i}(q)$, du *pixel shader* au *vertex shader*, lors du calcul de décompression. La direction est alors calculée aux sommets du maillage, et interpolée sur la surface des triangles, c'est-à-dire au niveau des échantillons. Cette factorisation des calculs est toujours profitable pour accroître la performance. Bien évidemment, dans ce cas, les repères tangents associés aux échantillons $o_i \in O_d$ doivent être définis en conséquence.

Finalement, et comme on a déjà pu le mentionner, le lissage *a posteriori* des données décompressées peut être réalisé directement dans l'espace des textures, puisque les relations d'adjacence entre échantillons y sont préservées. Le filtre de lissage appliqué accède alors implicitement aux échantillons voisins sur la surface de l'objet, en s'appuyant sur le voisinage dans la texture.

Algorithme de rendu La séquence des différentes étapes du rendu dans ce cas est résumée dans la figure 6.19. Toutes les étapes sont prises en charge par le matériel graphique, mais seul le rendu final donne lieu à un affichage à l'écran.

Dans la mesure où l'on a supposé qu'aucune continuité stricte n'était requise aux bords des surfaces, on propose d'employer des zones-tampons dans la texture bordant l'empreinte du maillage — des gouttières — pour éviter de polluer les données calculées par des données non définies de la texture, lors de l'interpolation des textures réalisée par le matériel graphique. Bien évidemment,

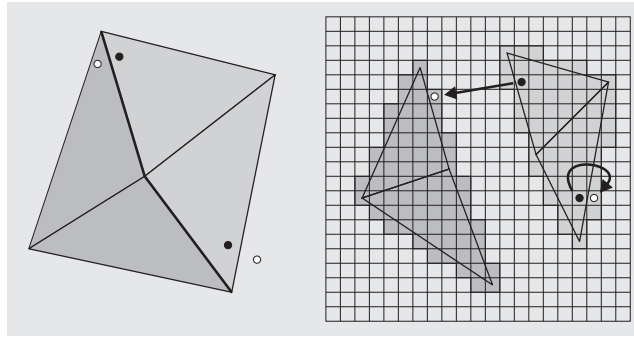


Figure 6.20: Stratégie de remplissage des gouttières. À gauche est figuré un maillage composé de quatre triangles. À droite est présentée sa projection dans une texture. Deux texels de gouttière, représentés en blanc, sont remplis en considérant deux texels valides, en noir. Si le texel de gouttière correspond à un débordement du triangle qui, sur la surface, recouvre un autre triangle, on emploie pour la copie un échantillon issu de ce triangle, comme représenté en haut de la figure. Dans le cas contraire, le texel de gouttière est proche d'un bord naturel du maillage, et il est rempli par extension de la valeur la plus proche.

dans la mesure où les données stockées dans les textures sont ici calculées dynamiquement, les gouttières doivent, elles aussi, être mises à jour dynamiquement.

Le mode de mise à jour que l'on emploie est une simple copie de valeurs déjà existantes, et non un calcul plus complexe. Plutôt que d'étendre systématiquement la valeur de bordure la plus proche dans la gouttière, on recherche la valeur d'un texel du triangle opposé scindé par la partition de la surface, s'il existe, comme l'illustre la figure 6.20. Ce texel opposé existe si le texel examiné est proche d'une arête issue d'un découpage de la surface, et non de son bord naturel.

Les gouttières sont remplies à l'aide d'une passe de rendu usant d'un *pixel shader* adéquat, qui recopie selon l'indirection choisie les données de la *distortion map* $\mathcal{D}_{\mu,q}$ dans la *distortion map* étendue $\mathcal{D}_{\mu,q}^+$.

Après cette expansion, le filtrage spatial peut être appliqué si l'utilisateur le souhaite, pour aboutir à la *distortion map* étendue et lissée $\widehat{\mathcal{D}}_{\mu,q}^+$. Dans le cas d'un lissage de rayon important, il est important de s'assurer de disposer de gouttières d'une taille suffisante pour éviter toute pollution des données interpolées.

La *refraction map* finale employée pour le rendu final $\mathcal{R}_{\rho,q}$ correspond à cette texture lissée et étendue : $\mathcal{R}_{\rho,q} = \widehat{\mathcal{D}}_{\mu,q}^+$.

6.5.2 Approche avec garantie de continuité

Pour certains types d'objets, par exemple la plupart des objets munis d'une surface sans bord, il n'est pas possible d'employer la méthode précédente sans provoquer de discontinuités jugées indésirables. Pour pallier ce problème, on propose d'adopter la stratégie présentée dans cette section, à même de garantir une continuité de l'interpolation de la distorsion sur la totalité de la surface triangulée.

L'apparition de discontinuités parasites est localisée aux arêtes par lesquelles passe le découpage de la surface originale, découpage réalisé pour en permettre

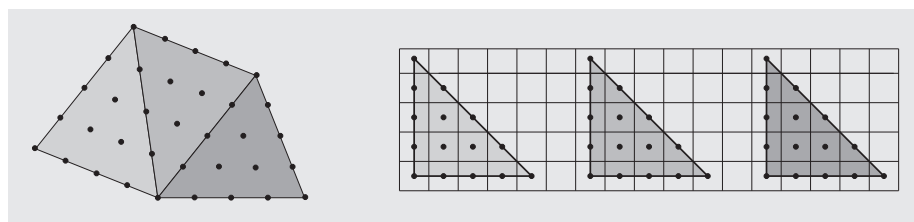


Figure 6.21: Échantillonnage uniforme dérivé de la triangulation. Les échantillons sont représentés sur la surface de l'objet dans la partie gauche de la figure, et dans la texture dans la partie droite.

la paramétrisation dans la texture. Une même arête physique du maillage se trouve alors dupliquée à deux endroits différents dans la texture. Durant le rendu de deux triangles scindés par la partition de la surface, à proximité de l'arête commune, l'interpolation bilinéaire effectuée par le matériel graphique emploie les texels les plus proches dans la texture, qui ne correspondent pas aux mêmes échantillons. L'interpolation ne peut donc fournir une valeur continue à travers l'arête.

L'idée intuitive d'employer des gouttières dynamiques, dont les valeurs sont copiées depuis les échantillons symétriques par rapport à l'arête, n'est pas suffisante pour garantir la continuité, comme on peut le constater au travers de la technique précédente, et comme présenté dans la figure 6.17. Même s'il est envisageable d'appliquer des algorithmes de remplissage des gouttières plus performants qu'une simple copie, une dissimilarité dans les taux d'échantillonnage des deux arêtes dupliquées ne peut être compensée par ce biais.

Le problème est que l'interpolation réalisée par le matériel graphique n'est pas programmable, mais toujours bilinéaire, et qu'elle accède donc systématiquement aux quatre échantillons de la texture bordant le point considéré.

Paramétrisation choisie Puisque l'algorithme d'interpolation du matériel graphique ne peut être modifié, la seule latitude que l'on a est d'effectuer un prétraitement des données stockées, avant interpolation, ainsi que de choisir la paramétrisation pour aboutir à une interpolation correcte.

La procédure proposée suit exactement ces deux voies : elle détermine en premier lieu une paramétrisation particulière de la surface du maillage, puis elle emploie ensuite un algorithme simple de correction des données à interpoler. La conjonction de ces aspects garantit alors la continuité de l'interpolation sur la totalité de la surface.

La paramétrisation ρ choisie correspond à celle produisant un échantillonnage régulier de chaque triangle, comme le présente la figure 6.21, les centres des texels définissant les positions des échantillons $o_i \in O_d$. Chaque triangle est géré indépendamment de tous les autres dans le maillage. Le taux d'échantillonnage, d'au moins un échantillon par sommet du maillage, est choisi par l'utilisateur. L'absence de liberté dans le schéma d'échantillonnage est la contrepartie de la garantie de continuité dans l'interpolation.

Un tel échantillonnage est choisi pour trois raisons : il est calculable de manière triviale, il a la propriété que toutes les arêtes de tous les triangles sont compatibles entre elles en termes de position des échantillons *a priori*, et il

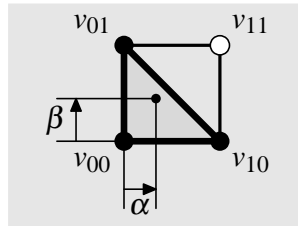


Figure 6.22: Interpolation dans la texture. Représentation d'une cellule normalisée de la texture, comprise entre quatre échantillons.

est possible de créer une procédure de correction de l'interpolation qui lui est associée. Les différents triangles peuvent alors être stockés dans une texture en étant alignés avec les axes de la texture, les échantillons communs à plusieurs triangles étant dupliqués. Aucun problème de placement des triangles ne se pose dans la texture, tous étant identiques du point de vue de l'échantillonnage quelle que soit l'orientation choisie.

La duplication de certains échantillons peut être importante avec cette paramétrisation ρ , surtout si le taux d'échantillonnage des triangles est faible. Afin d'éviter une duplication des calculs, en particulier de ceux de décompression, la paramétrisation ρ n'est pas celle utilisée pour la majorité des calculs. Elle est uniquement employée pour formater les échantillons en conjonction avec la procédure de correction de l'interpolation, que l'on détaillera dans la suite, et qui produit la *refraction map* finale $\mathcal{R}_{\rho,q}$. Tous les calculs sont réalisés sur des textures formatées par la paramétrisation μ , qui est définie en ne conservant qu'une seule et unique copie de chacun des échantillons $o_i \in O_d$ induits par ρ , et en les réarrangeant dans une texture de taille minimale. La paramétrisation μ des échantillons dans l'espace des textures est donc complètement déstructurée, et les relations de voisinage entre échantillons sur la surface totalement perdues dans l'espace des textures.

Correction de l'interpolation Pour obtenir une interpolation correcte des échantillons sur la surface des différents triangles, il est nécessaire que seuls des échantillons situés à l'intérieur du triangle lui-même soient employés. Or le filtrage bilinéaire proposé par le matériel graphique interpole de manière systématique quatre échantillons dans la texture.

En considérant une cellule carrée normalisée de la texture, entre quatre points-échantillons dans la texture, comme le présente la figure 6.22, la valeur interpolée est calculée selon le schéma :

$$I_{bil}(\alpha, \beta) = (1 - \alpha) \cdot (1 - \beta) \cdot v_{00} + \alpha \cdot (1 - \beta) \cdot v_{10} \\ + \alpha \cdot \beta \cdot v_{11} + (1 - \alpha) \cdot \beta \cdot v_{01}$$

Avec ce schéma linéaire, les valeurs produites le long des arêtes sont interpolées linéairement entre les deux valeurs situées aux deux sommets de l'arête. Au contraire, les valeurs strictement internes à la cellule sont affectées par les valeurs des quatre échantillons.

Si l'on considère le cas d'un triangle muni d'un échantillon à chaque sommet, placé comme décrit dans la figure 6.22, l'interpolation souhaitée n'est pas

bilinéaire, mais barycentrique, sur la surface occupée par le triangle. Cette interpolation, de la forme

$$I_{bar}(\alpha, \beta) = \alpha \cdot v_{10} + \beta \cdot v_{01} + (1 - \alpha - \beta) \cdot v_{00}$$

ne peut être implémentée explicitement puisque l'algorithme d'interpolation est fixe. En revanche, elle prend naturellement place si la valeur du quatrième échantillon v_{11} , externe au triangle, satisfait à la relation suivante :

$$v_{11} = v_{01} + v_{10} - v_{00}$$

alors qu'une interpolation bilinéaire I_{bil} est appliquée.

Dans ce cas, la valeur interpolée le long de l'arête diagonale est bien interpolée linéairement entre les valeurs associées aux échantillons, et le recollement d'interpolations séparées sur différents triangles devient continu.

En calculant dynamiquement la valeur v_{11} , il est donc possible d'interpoler de manière adéquate une quantité sur une cellule triangulaire, à condition que son arrangement suive celui que l'on vient de décrire.

Il est possible d'étendre cette interpolation correcte à des triangles échantillonnés d'une manière plus dense, mais toujours régulière. Les triangles étant stockés de manière fixe dans la texture, alignés avec les axes, deux zones se dessinent sur leur surface, comme le montre la figure 6.23 : celle composée des cellules d'interpolation dont les quatre échantillons sont internes au triangle, et celles pour lesquelles un échantillon est externe.

Pour le premier groupe de cellules, l'interpolation bilinéaire naturelle est tout à fait satisfaisante. Pour le deuxième, une interpolation barycentrique des trois échantillons valides est requise : elle est implémentée de manière indirecte, en calculant dynamiquement les valeurs des échantillons externes, comme on l'a présenté.

Avec un tel schéma, tous les segments d'arêtes entre deux échantillons sont interpolés linéairement entre les valeurs portées par ces échantillons. Comme toutes les arêtes sont discrétisées d'une manière équivalente, l'interpolation sur la totalité de la surface de l'objet est garantie continue, sous réserve que les valeurs portées par des échantillons dupliqués soient identiques.

Le calcul de ρ et de la position des échantillons est donc très simple : la triangulation de la surface de l'objet est complètement éclatée, triangle par triangle, et les triangles sont alignés dans la texture, un par un, leur taille étant fonction du taux d'échantillonnage spécifié par l'utilisateur. Tous les triangles étant identiques, aucune contrainte ne s'applique quant à leur placement ou leur sélection. Les échantillons $o_i \in O_d$ sont par suite définis comme les centres des texels valides de la texture, en ignorant les duplicata éventuels. La paramétrisation ρ est définie comme celle qui transforme O_d en ce placement.

En fait, les triangles ne sont pas placés alignés, mais tête-bêche par groupes de deux, comme illustré dans la figure 6.24 pour faire meilleur usage de l'espace de texture. La procédure de correction de l'interpolation opère de manière analogue pour la configuration symétrique.

Algorithme de rendu La succession des différentes étapes du rendu avec garantie de continuité est présentée dans la figure 6.25.

Comme dans la procédure directe, la *distortion map* $\mathcal{D}_{\mu,q}$ est tout d'abord calculée, les données y étant formatées selon la paramétrisation μ des échantillons, la plus efficace, chaque échantillon n'apparaissant qu'une et une seule

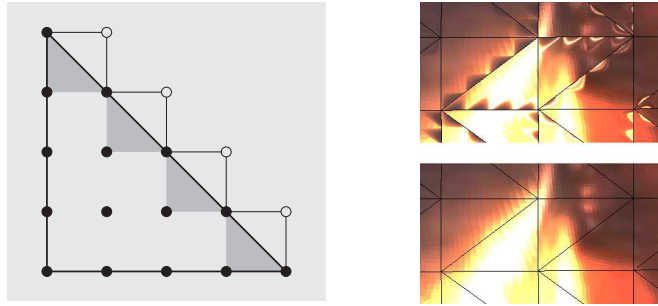


Figure 6.23: Zones d'interpolation. À gauche, illustration des deux zones d'interpolation des triangles : celle figurée en couleur claire, pour laquelle l'interpolation bilinéaire produit le résultat attendu, et celle figurée en couleur sombre, pour laquelle il est nécessaire d'émuler une interpolation barycentrique. Pour cela, les valeurs externes au triangle, figurées par des points blancs, sont calculées dynamiquement. À droite, on peut distinguer ces zones lors du rendu si aucune correction n'est appliquée, mais elle disparaissent dès lors que la correction est appliquée.

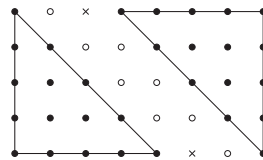


Figure 6.24: Stockage des triangles. Les points noirs représentent les texels naturellement valides, les points blancs ceux dont il est nécessaire de calculer les valeurs dynamiquement pour corriger l'interpolation, et les croix les texels inutilisés

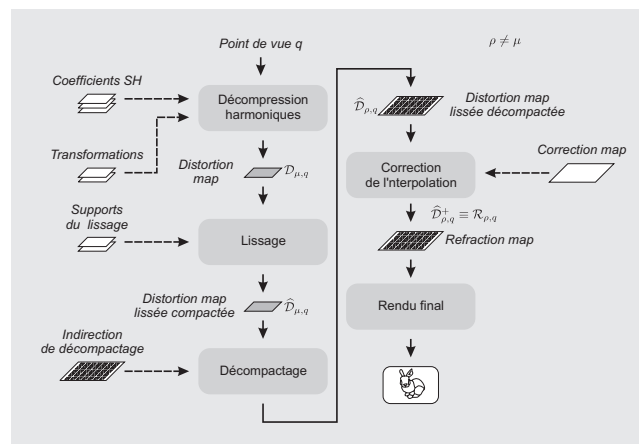


Figure 6.25: Étapes du rendu pour l'approche avec garantie de continuité

fois. Puisque la topologie des échantillons n'est alors pas préservée dans l'espace des textures, le calcul de la direction d'observation $\omega_{o_i}(q)$ doit être réalisé explicitement pour chaque échantillon $o_i \in O_d$. Les données nécessaires à ce calcul, *i.e.* les matrices de transformation T_{o_i} doivent alors être stockées dans des textures supplémentaires fournies à l'algorithme de décompression.

Une fois décompressées, les données peuvent être filtrées spatialement si nécessaire. Ce filtrage est également appliqué dans l'espace paramétré par μ pour éviter tout calcul inutile. Comme on l'a évoqué dans la section consacrée au lissage spatial, le filtre requiert une connaissance explicite des échantillons composant les supports des filtres, pour chaque échantillon spatial $o_i \in O_d$. Ces supports sont fournis à la procédure de lissage sous forme de textures d'indirection par rapport à la paramétrisation μ . La *distortion map* filtrée, $\widehat{D}_{\mu,q}$ est le résultat de ce filtrage. On notera que pour une telle procédure de filtrage, aucune gestion particulière des bords de l'objet n'est à réaliser durant le rendu, puisque les échantillons sont choisis durant le précalcul et stockés explicitement.

Finalement, les données lissées sont reformatées selon la paramétrisation ρ , pour aboutir aux données $\widehat{D}_{\rho,q}$ correctement disposées pour la procédure de correction d'interpolation. Cette étape de reformatage correspond à un simple décompactage, une copie avec une indirection. Finalement, l'algorithme de correction de l'interpolation décrit ci-dessus emploie ces données pour produire la *distortion map* lissée et corrigée, $\widehat{D}_{\mu,q}^+$. Cette dernière n'est rien d'autre que la *refraction map* finale $\mathcal{R}_{\rho,q}$, prête à fournir les données pour une interpolation continue sur la totalité du maillage de surface lors du rendu final.

Bien qu'un peu plus complexes que la version directe, les différents algorithmes de cette chaîne de traitement restent suffisamment simples pour que chacun puisse être implémenté par une unique passe de rendu entre deux textures, exception faite de la compression elle-même qui requiert généralement plus d'une seule passe. L'ensemble de la décompression peut donc être réalisé par le matériel graphique, ce qui autorise un rendu interactif.

6.5.3 Rendu final

La phase de rendu final est très simple, une fois la *refraction map* finale préparée, puisque cette dernière a été calculée pour être directement interpolée sur les différents triangles de la surface.

Le rendu effectué est résumé dans la figure 6.26. Il consiste essentiellement à échantillonner l'environnement pour le rayon réfléchi primaire, et pour le rayon réfracté dont la direction est interpolée depuis la *refraction map*. Les deux contributions sont ensuite pondérées en fonction des facteurs d'atténuation de Fresnel qui doivent être également calculés, ainsi que de l'éventuelle atténuation subie par le rayon réfracté lors de son parcours dans l'objet réfringent.

Dans la mesure où la distance parcourue est non-bornée, contrairement aux coordonnées du vecteur de distorsion, il n'est pas possible de la stocker de manière fiable dans une *refraction map* dont le format des valeurs est borné. Or le format de texture entier, borné, est une nécessité dans l'architecture-cible, dans la mesure où l'on désire interpoler cette texture ultérieurement. Pour éviter cet écueil, l'atténuation, fonction de la distance est calculée dans la dernière passe de décompression. L'atténuation étant bornée, elle peut être stockée de manière fiable sous forme quantifiée dans la *refraction map*. En revanche, une

1. Calcul éventuel de l'*environment map* si celle-ci n'est pas statique
2. Renormalisation de la normale n interpolée des valeurs aux sommets
3. Renormalisation de la direction d'observation $\omega(q)$ interpolée des valeurs aux sommets
4. Extraction des facteurs de Fresnel \mathcal{R} et \mathcal{T} depuis une texture 1D en fonction de l'angle d'incidence calculé à partir de n et de q
5. Calcul de la direction réfléchie r
6. Échantillonnage de l'*environment map* cubique selon la direction r pour calculer l'intensité lumineuse réfléchie R
7. Extraction de la direction réfractée t et de la l'atténuation a à partir de la *refraction map* interpolée automatiquement
8. Modification de t pour la prise en compte d'une transformation rigide de l'objet
9. Interpolation de la direction t avec la direction d'observation $\omega(q)$ non perturbée
10. Échantillonnage de l'*environment map* cubique selon la direction t pour calculer l'intensité lumineuse réfractée T
11. Combinaison pour former le résultat $res = \mathcal{R} \cdot R + a \cdot \mathcal{T} \cdot T$

Figure 6.26: Algorithme de rendu final

seule valeur d'atténuation est ainsi rendue disponible, ce qui oblige à colorer *a posteriori* l'atténuation durant le rendu final.

On propose également d'appliquer, durant l'affichage, un traitement simple pour simuler une variation dans les matériaux des objets. La déviation des rayons réfractés a été précalculée pour un jeu de matériaux connu à l'avance. Or, on sait que si les indices de réfraction de l'objet et de l'environnement sont identiques, les rayons ne sont pas déviés. On propose donc d'interpoler la direction distordue entre la direction non perturbée, associée à un rapport unitaire d'indices de réfraction, et la distorsion calculée, associée à un rapport d'indices connu. Il est clair que cette technique est absolument empirique et dénuée de toute base physique, mais elle peut se révéler utile pour animer un changement de matériaux, et présente un coût quasi-nul.

6.6 Résultats

La figure 6.27 présente divers objets réfringents, affichés grâce à la technique présentée, et recense également quelques statistiques générales sur ces objets. Comme on peut le constater, la vitesse d'affichage reste interactive, même pour des objets de taille non-négligeable. Ces mesures ont été réalisées sur une machine équipée d'un processeur Athlon XP 1800+ cadencé à 1.53 GHz et d'une carte graphique Radeon 9700 Pro embarquant 128 Mo de mémoire graphique, supportant l'architecture de *Shaders 2.0b* de DirectX.

Les temps sont relatifs à la décompression la plus précise réalisable sur ce matériel, avec des harmoniques d'ordre 8, soit une compression employant 81 fonctions de base différentes — comme pour toutes les autres images présentées, sauf mention contraire —

Les images présentent une résolution d'environ 1350×1100 pixels. Le calcul



Objet	$ O_d $	Échantillons distorsion	Nb. triangles	Texture SH	Mémoire totale	Affichage (Hz)
Coquillage	34 k	70 M	4288	186^2	17 Mo	52
Lapin	35 k	72 M	69451	187^2	35 Mo	40
Tasse	42 k	86 M	9184	205^2	24 Mo	43
Théière	41 k	84 M	2256	203^2	23 Mo	45
Lucy	119 k	244 M	238734	346^2	121 Mo	7

Figure 6.27: Résultats. Les images aussi bien que les statistiques ont trait à des compressions d'ordre $l = 8$, et emploient donc 81 fonctions de base. Dans les statistiques, $|O_d|$ représente le nombre d'échantillons choisis sur la surface des objets. Le nombre d'échantillons de distorsion correspond aux nombre de trajets de la lumière évalués lors du précalcul. La colonne nommée Texture SH correspond à la taille des textures stockant les coefficients des fonctions compressées.



Figure 6.28: Matériaux multiples. Un objet composite, formé de deux milieux différents, air et eau, est représentable par la méthode sans changement. Durant le rendu, la méthode s'abstrait complètement de la nature des objets à visualiser.

complet de la distorsion étant réalisé dans l'espace des textures, l'évaluation de la distorsion reste complètement indépendante de la taille de l'image. Le rendu final, qui, lui, inclut l'affichage, reste lié à la taille de la projection des objets à l'image.

Matériaux évolués Un des intérêts de la technique proposée est de déporter toute la complexité du calcul de réfraction sur l'algorithme de lancé de rayon exécuté en précalcul. Il est par conséquent tout à fait possible d'incorporer des objets mettant en jeu plusieurs matériaux aussi bien qu'un seul, comme le présente la figure 6.28. En effet, qu'un seul ou plusieurs matériaux soient présents, le résultat du précalcul reste un champ de distorsion, utilisable lors de la phase de rendu.

La figure 6.29 illustre l'application d'une atténuation variable selon la distance effectivement parcourue par les rayons dans l'objet, compressée et employée de la même manière que les données de distorsion. Puisque la distance est stockée, et non l'atténuation elle-même, il est possible de la faire varier interactivement l'absorption associée au matériau.

La figure 6.30 illustre pour sa part l'effet de l'interpolation de la direction distordue, interpolation qui permet d'émuler une variation interactive des matériaux de la scène. Si les résultats ne prétendent pas s'approcher de résultats proprement calculés pour des matériaux différents, ils peuvent en fournir une approximation parfois convaincante. Cette interpolation constitue donc un moyen simple d'assouplir à peu de frais la rigidité de la méthode liée au précalcul.

Comparaison avec d'autres méthodes Les résultats produits par la méthode que nous proposons peuvent être visuellement comparés avec ceux produits par d'autres méthodes dans la figure 6.31. La comparaison est effectuée sur un modèle de coquillage contenant une perle, un objet fortement auto-occlusif.

Les différents résultats font l'objet d'une comparaison avec une image de référence, obtenue par lancé de rayons. Comme on peut le constater, la technique classique de réfraction interactive, qui consiste à tenir compte uniquement de la déviation initiale du rayon réfracté, produit des résultats qui n'ont rien

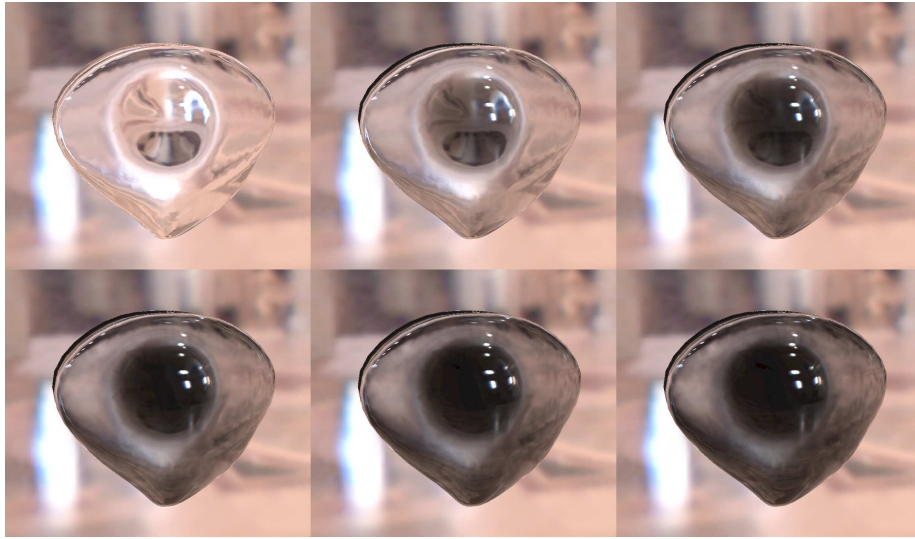


Figure 6.29: Atténuation variable. Une atténuation variable, dépendant de la distance parcourue dans l'objet par chaque rayon, peut être appliquée. Les résultats présentés vont d'un objet complètement transparent à un objet quasiment opaque. La perle ressort bien du coquillage car elle est beaucoup plus épaisse que la coquille, et engendre donc une atténuation plus importante.

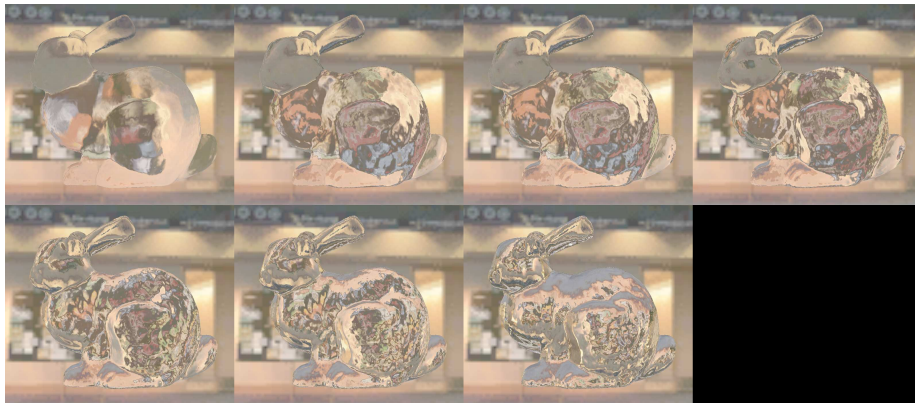


Figure 6.30: Émulation d'un changement de matériaux. L'intensité de la distorsion est modulée entre la valeur décompressée et une valeur nulle. La dernière image de la série correspond à la distorsion décompressée non modifiée.

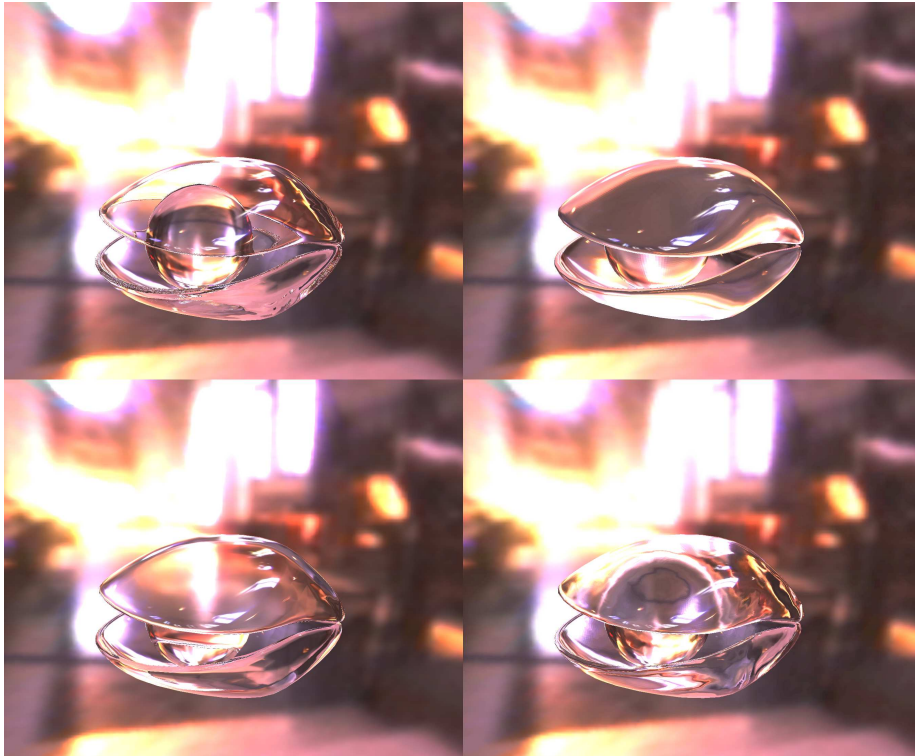


Figure 6.31: Comparaison avec d'autres méthodes. La première image correspond à une image de référence obtenue par lancé de rayons. Les images suivantes correspondent respectivement à une réfraction ne considérant que la première déflexion, considérant une paire de déflexions [182], et à la technique que nous proposons. La technique que nous proposons parvient à tenir compte de l'influence de la perle, même si celle-ci est occluee.

à voir avec l'image de référence. Elle se contente essentiellement de distordre l'environnement, mais ne permet pas d'appréhender la forme de l'objet.

La technique proposée par Wyman [182], qui permet de tenir compte d'une paire de réfractions, est déjà sensiblement plus proche de la référence. Cependant, le trajet plus complexe de la lumière qui rentre et sort plusieurs fois de l'objet n'est pas capturé correctement, puisque seules les surfaces avant et arrière de l'objet sont considérées par la méthode. Le résultat de cette approximation est que la perle disparaît complètement lorsqu'elle est occultée.

Finalement, on peut constater que la technique que nous proposons est à même de capturer ces déviations multiples de la lumière. Bien que capturée imparfaitement, la perle ne disparaît pas complètement malgré son occlusion par la coquille, et continue de déformer l'environnement. L'objet est ainsi clairement plus réaliste, même si la réfraction apparaît lissée, et moins franche que dans le résultat calculé par lancé de rayons.

Capture et compression de la réfraction La figure 6.32 illustre la variation de la distorsion générée, en fonction de la qualité de compression. La première série d'images, de la théière, permet de constater que lorsque davantage de fonctions de base sont employées, l'approximation par harmoniques s'approche effectivement de la réfraction de référence, calculée par lancé de rayons.

La seconde série d'images, figurant le modèle géométrique complexe du *Stanford bunny*, illustre un résultat assez paradoxal : plus l'objet est complexe, moins le nombre d'harmoniques pour faire apparaître un résultat plausible est élevé. En effet, sans référence visuelle pour établir une comparaison, même des approximations très lâches apparaissent à un observateur comme étant parfaitement crédibles. La réfraction est tellement complexe qu'il est impossible de prédire mentalement le résultat pour des géométries complexes. On notera cependant que l'emploi d'un plus grand nombre de fonctions de base tend à mieux faire ressortir le volume de l'objet.

La dernière série d'images illustre plus clairement la différence liée au niveau de compression, dans le cas d'un objet plus simple, le coquillage contenant une perle. Pour cet objet muni d'une géométrie plus simple, la réfraction peut être mieux appréhendée, même sans référence visuelle explicite. On constate dans cette série d'images, que la perle se dessine avec une netteté croissante à mesure que le nombre de fonctions employées pour la compression augmente. Une représentation crédible d'un tel objet demande donc l'emploi d'une compression de bonne qualité.

Comme seul un nombre restreint de fréquences peut être stocké, seule une capture basse-fréquence de la distorsion est possible. Hors problèmes éventuels de capture liés à l'échantillonnage, la distorsion synthétisée estompe donc naturellement les détails des perturbations. Dans la mesure où la fréquence des distorsions engendrées par un objet varie selon son éloignement à la surface où elle est capturée, la configuration géométrique des objets joue un grand rôle dans la précision de la restitution. La figure 6.33 illustre ce phénomène, en présentant une étoile vue derrière une plaque plane transparente, avec trois espacements possibles entre les deux objets. Comme attendu, plus l'étoile est éloignée de la plaque, plus la fréquence de la distorsion augmente et moins bonne est sa capture. Cette figure illustre également l'effet de lissage produit par le filtre fréquentiel.

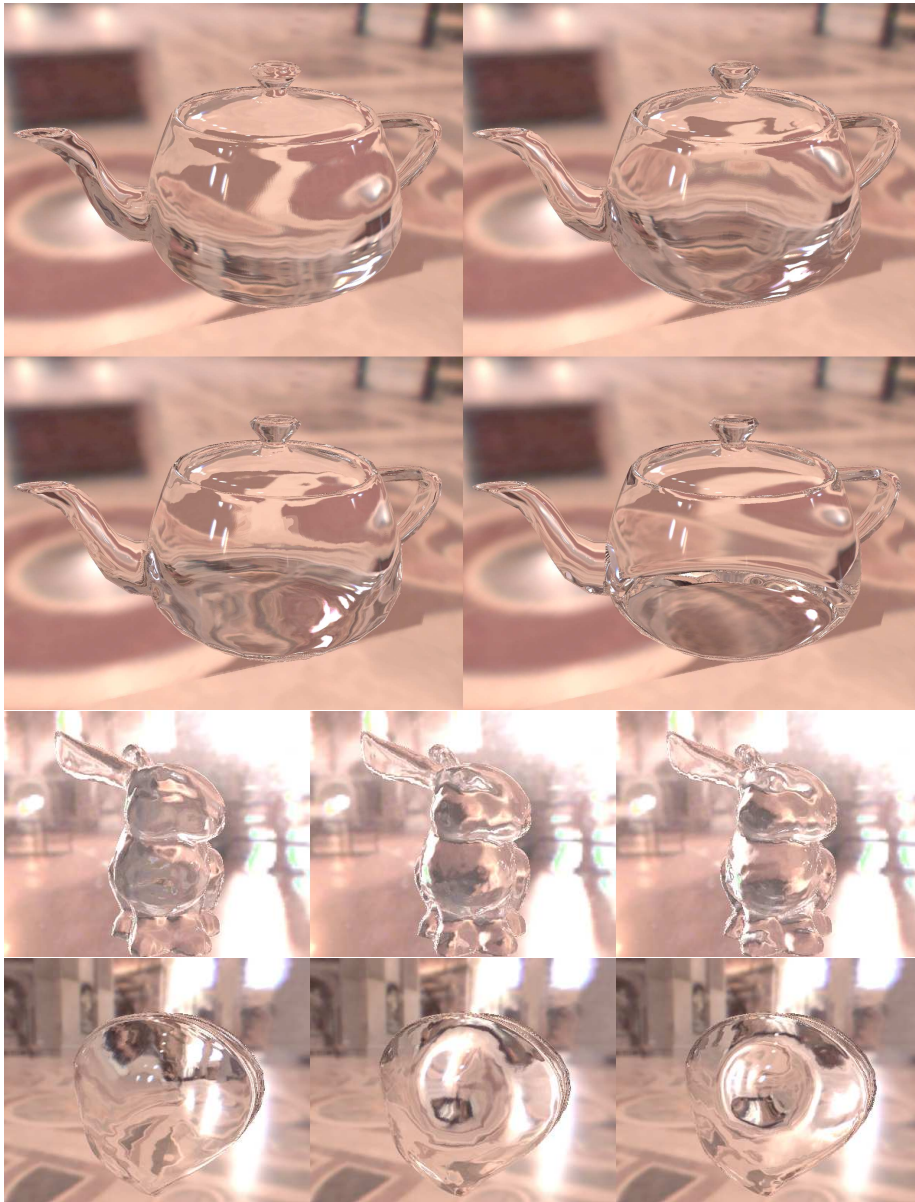


Figure 6.32: Variation de la compression. Trois séries d'images sont présentées, pour trois objets différents représentés avec les ordres de compression $l = 1$, $l = 4$, et $l = 8$. La quatrième image de la première série figure une image de référence obtenue par lancé de rayons.

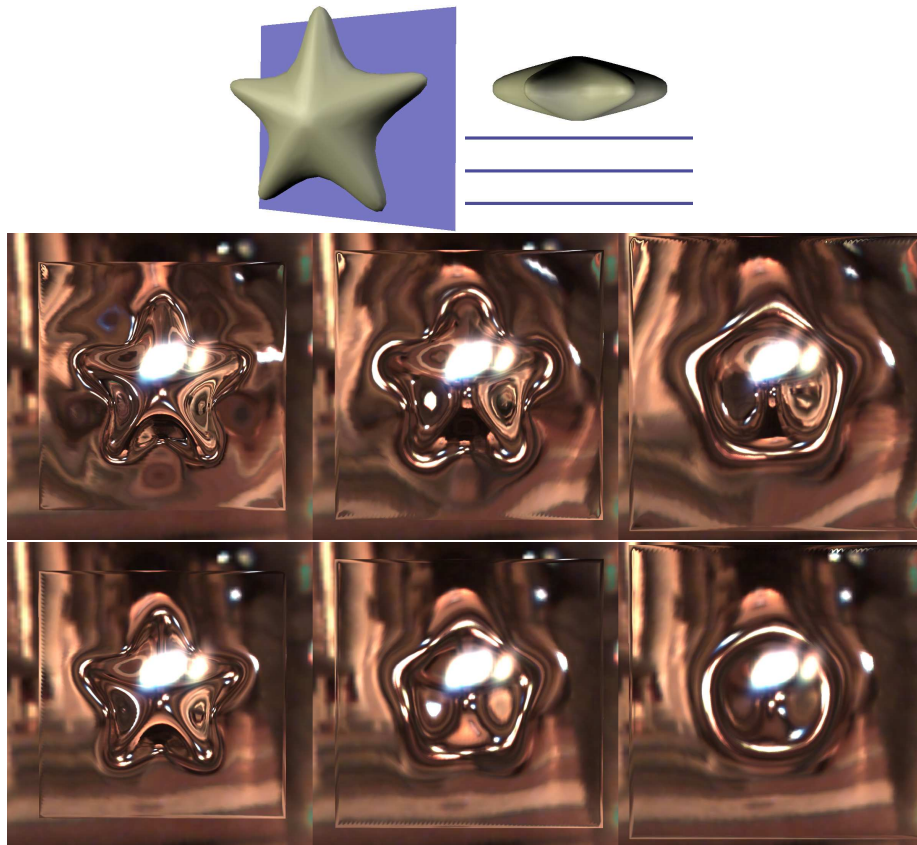


Figure 6.33: Capture progressive. Un objet étoilé est vu à travers une plaque transparente, à trois distances différentes. Les deux objets et les trois configurations de distance sont présentés en haut de la figure. Les deux séries d'images suivantes correspondent aux résultats obtenus pour une décompression de qualité maximale, $l = 8$, pour les trois distances. La série du bas inclut un lissage fréquentiel, au contraire de la série du haut.

On remarquera dans cette situation une qualité de la méthode : bien que la capture soit de moins en moins précise, elle s'estompe très progressivement, et jamais de manière abrupte. Une distorsion restera toujours visible, dans certaines limites, ce qui est largement préférable du point de vue du réalisme à une disparition complète de certaines structures : l'objet reste ainsi mieux appréhendé, même si ce n'est pas forcément avec une très grande précision.

Vitesse de rendu et coût mémoire Les figures 6.34 et 6.35 présentent la vitesse de rendu pour trois objets de taille croissante, ainsi que les ressources mémoires requises. Ces statistiques se focalisent sur la technique la plus intéressante, celle garantissant une parfaite continuité dans l'interpolation.

On constate qu'une large part des ressources n'est pas employée pour stocker les données elles-mêmes. Les différentes données auxiliaires requises sont en effet nombreuses. La duplication de la géométrie est également importante, puisque chaque triangle est géré indépendamment. Précisons que l'implémentation réalisée n'est pas économe en mémoire, et que ces besoins auxiliaires pourraient certainement être réduits en jouant notamment sur le format de stockage de certaines données auxiliaires, mais au prix d'une implémentation moins souple.

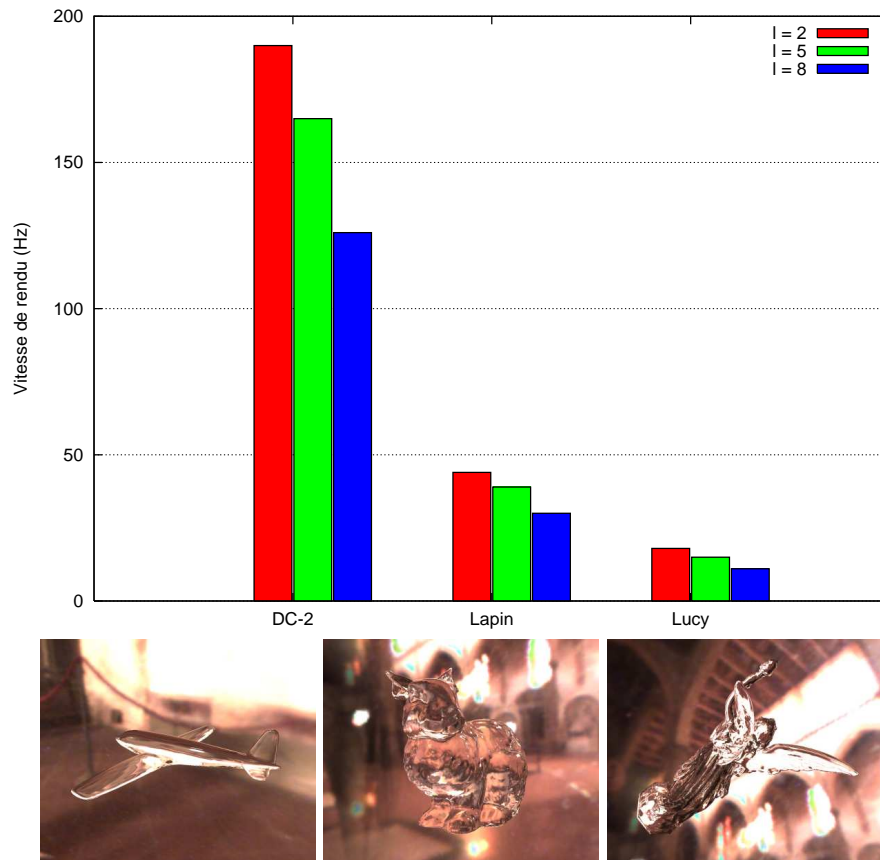
Cette figure 6.35 illustre également, à titre de comparaison, les besoins en mémoire de la technique sans garantie de continuité. On remarque alors que cette technique directe est plus économe du point de vue du coût-mémoire par échantillon, en limitant l'usage de données auxiliaires et la duplication de la géométrie.

La vitesse d'affichage dépend bien évidemment de manière directe du nombre de fonctions de base à décompresser ainsi que du nombre d'échantillons à traiter. Les statistiques fournies sont relatives à une décompression complète, à chaque image, de tous les échantillons, jusqu'à l'ordre indiqué. On note que le fait de découpler la préparation de la *refraction map* de son utilisation, permet également de distribuer, si nécessaire, son calcul sur plusieurs images sans difficulté. En outre, le caractère progressif de la compression autorise à moduler dynamiquement la décompression, en fonction d'un caractère de visibilité, ou de distance à l'objet.

Pour évaluer plus complètement les performances de la méthode, il est nécessaire de tenir compte des temps de précalcul. Celui-ci est en majorité consacré à l'échantillonnage de la fonction de distorsion. Le précalcul dépend alors du nombre de trajets de la lumière dans l'objet à évaluer. À titre d'exemple, le lapin présenté est échantillonné avec 2048 directions pour chacun des 35 000 échantillons de surface, ce qui représente approximativement 71.5 millions de trajets à évaluer. Avec l'implémentation utilisée du lancé de rayons, accélérée par l'utilisation d'une grille régulière, cette tâche a requis approximativement 26 minutes, produisant 1.1 Go de données. La compression de ces données a ensuite été exécutée en 5 minutes et 20 secondes. S'il n'est certes pas rapide, le précalcul reste tout à fait acceptable en termes de temps de calcul.

Bruit Un défaut de la méthode, qu'il n'est malheureusement pas possible d'illustrer par des images fixes, est qu'elle est, sous certaines conditions, relativement sensible à un bruit dans la réfraction, visible lors des mouvements de l'objet.

Les causes en sont multiples. Il est principalement dû au fait que le rendu



Objet	Nb. échantillons	Nb. triangles
DC-2	7 k	3.5 k
Lapin	35 k	70.0 k
Lucy	120 k	239.0 k

Figure 6.34: Vitesse de rendu. La vitesse de rendu est illustrée pour trois objets d'un échantillonnage croissant, pour les trois niveaux de compression $l = 2$, $l = 5$, et $l = 8$. La technique employée est celle avec garantie de continuité. Les performances sont mesurées sur une carte graphique nVidia GeForce 6800GT, pour des images d'approximativement 1350×1100 pixels, illustrées dans le bas de la figure. La vitesse dépend logiquement du nombre d'échantillons à manipuler et de l'ordre de décompression.

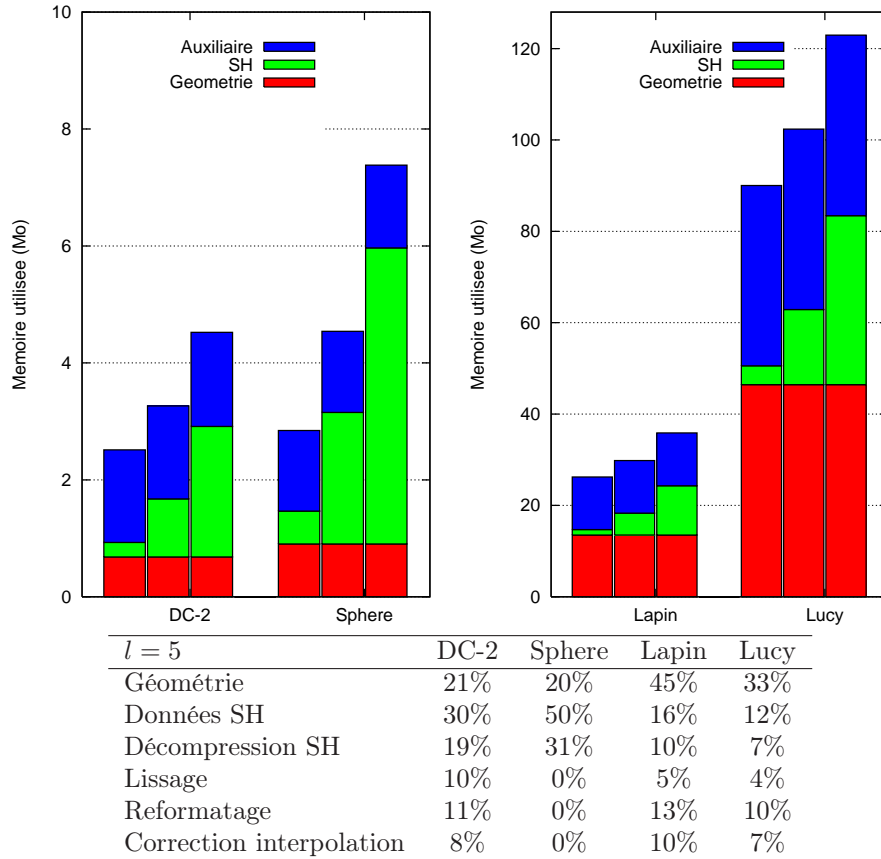


Figure 6.35: Utilisation mémoire. L'utilisation mémoire pour les trois objets de la figure 6.34 est également présentée pour les trois niveaux de compression, en séparant la géométrie, les données compressées, et les données auxiliaires du rendu. Ces dernières comprennent les données nécessaires à un lissage spatial à cinq points. Les statistiques concernant la sphère fournissent un point de comparaison avec la méthode directe, moins intéressante car ne produisant pas des résultats parfaitement continus. La sphère est échantillonnée par environ 30 000 points sur sa surface, et ceux-ci occupent 42% de l'espace qui leur est alloué dans les textures. Elle est donc comparable au lapin, qui, lui, emploie environ 35 000 échantillons. Le tableau détaillant davantage la répartition de la mémoire correspond au cas $l = 5$. Une variation de l'ordre de décompression n'affecte que la quantité de mémoire associée aux données, les autres postes restants fixes.

est en partie réalisé dans des textures, et non intégralement à l'écran. Puisque le taux d'échantillonnage de la surface est choisi de manière fixe lors du précalcul, il ne peut s'adapter dynamiquement à celui de l'image courante. S'il est supérieur à celui de l'image, un bruit lié à de l'*aliasing* peut alors survenir. Dans la mesure où les échantillons sont déstructurés dans la texture pour la méthode garantissant la continuité, il n'est pas possible de recourir à un filtrage de la texture de type *mip-mapping*, plus performant que le simple filtrage linéaire, pour combattre cet effet, sans polluer les interpolations avec des données parasites.

Le problème est amplifié par le fait que la distorsion est employée comme une indirection, pour échantillonner l'environnement avec un échantillonnage ponctuel. Les parties de surfaces présentant de fortes courbures tendent alors à produire d'importantes distorsions, résultant en la projection de larges pans de l'environnement sur de petites surfaces à l'écran. Dans ces zones, un bruit temporel apparaît alors facilement, même lors de petits changements dans la direction d'observation.

Le choix du taux d'échantillonnage de la surface doit, par conséquent, se faire avec attention, ce qui n'empêche pas que, dans le cas d'objets présentant une projection avec une forte variation de distance, il peut se révéler difficile de déterminer un échantillonnage acceptable.

6.7 Bilan

Comme le montrent les résultats que l'on a présentés, la méthode proposée permet de gérer la réfraction de manière interactive, et ce, sans poser aucune contrainte sur la géométrie des objets réfringents ou sur les matériaux qui les composent.

La méthode s'appuie sur la capture de la distorsion produite par la présence de l'objet durant une phase de calcul. Cette capture, qui nécessite d'évaluer le trajet suivi par la lumière dans l'objet, est réalisée simplement par un processus de lancé de rayons. Les données produites, très volumineuses, sont encodées à l'aide d'une représentation fréquentielle à base d'harmoniques sphériques. Durant le rendu interactif, cette forme compressée des données est exploitée pour synthétiser la distorsion correspondant au point de vue courant, dans le but d'échantillonner un environnement en conséquence. Grâce à la programmabilité du matériel graphique, l'exploitation de ces données compressées peut être réalisée interactivement.

Cette approche par capture permet de ramener le calcul de la réfraction, phénomène particulièrement irrégulier, à une information parfaitement régulière de distorsion directionnelle définie sur la surface de l'objet. Le matériel graphique tire alors profit de cette grande régularité pour offrir un rendu interactif. On tire également parti des avantages du lancé de rayons pour gérer des géométries arbitraires.

Deux alternatives de gestion des données ont été proposées. La première, intuitive et directe, souffre de défauts de continuité à proximité des discontinuités de paramétrisation de la surface, mais autorise cependant des résultats plus que satisfaisants. Elle conserve en outre l'avantage de pouvoir employer une paramétrisation des objets fournie par l'utilisateur, ce qui permet de contrôler la densité de l'échantillonnage spatial.

La seconde permet de garantir une continuité parfaite de l'interpolation,

et autorise donc à traiter tous les objets d'une manière automatique et sans défauts. Cette représentation plus correcte se paye cependant par le fait que la paramétrisation des échantillons adoptée n'est plus contrôlable de manière externe : elle est choisie automatiquement par l'algorithme. Le seul contrôle laissé à l'utilisateur se limite au taux d'échantillonnage des triangles. Un tel placement des échantillons n'est de loin pas optimal, puisqu'il est uniquement lié à la topologie de la triangulation, et non au plongement géométrique de la surface. Dans les faits, cette limitation ne se révèle pas trop gênante, sous réserve que la triangulation de la surface soit relativement homogène.

Globalement, la technique que nous avons proposée reste toutefois limitée, et ce sur trois points principaux. Tout d'abord, elle consomme des ressources-mémoires importantes. Pour une part significative, celles-ci ne sont ainsi pas attribuées aux données utiles compressées, mais à des données auxiliaires. C'est seulement aux niveaux de qualité les plus importants que se rétablit l'équilibre entre données utiles et données auxiliaires. Même s'il semble possible de supprimer à terme une partie de ces données non-utiles, l'approche paraît, à l'heure actuelle, trop coûteuse pour une utilisation à grande échelle.

Le deuxième point, concerne la forme de compression employée : en raison de la croissance quadratique du nombre des fonctions harmoniques sphériques, ce mode de compression semble limité, et passe mal à l'échelle, d'autant plus que chaque nouvelle fréquence stockée ne produit qu'une amélioration de plus en plus faible des résultats. Par conséquent, il est clair qu'il n'est pas possible de capturer les phénomènes correspondant à de hautes fréquences, car beaucoup trop de coefficients seraient requis pour les représenter. La technique proposée ne peut donc pas capturer avec exactitude l'intégralité de la réfraction, et doit se contenter d'approximations lissées.

Finalement, cette méthode, tout du moins dans son implémentation actuelle, pose des problèmes de qualité en termes de bruit, certes limités et qui peuvent être jugulés en portant attention à l'échantillonnage spatial employé.

Une limitation supplémentaire importante, stratégique plutôt que tactique, découle du recours à un précalcul, qui empêche l'utilisation de la méthode sur des objets animés.

Malgré ses limites, une telle méthode présente tout de même l'avantage d'autoriser une inclusion convaincante d'objets réfringents. S'ils ne sont pas toujours parfaitement représentés, les distorsions fines étant estompées, la méthode présente cependant l'avantage de convertir la dégradation en un lissage progressif, et sans coupure brusque des résultats, plutôt plaisant sur le plan visuel. En outre, la plupart des distorsions majeures de la réfraction restent capturées. Le coût mémoire reste également du domaine du raisonnable pour des objets qui ne requièrent pas trop d'échantillons. De plus, la vitesse d'affichage peut être variée de manière assez souple, en modulant la décompression.

Plusieurs perspectives concrètes existent pour améliorer cette méthode, à court terme. Tout d'abord, il est possible très concrètement d'offrir une méthode de rendu avec garantie de continuité beaucoup plus souple que celle testée ici. La méthode proposée peut en effet être étendue à des découpages par morceaux de surfaces, et non par triangles isolés. La seule contrainte, relativement légère, qui subsisterait alors sur la paramétrisation, serait que les arêtes dupliquées présentent un taux d'échantillonnage identique. Cette version beaucoup plus souple du rendu n'ayant pu être testée, par manque d'une méthode de paramétrisation contraignant cette garantie sur l'échantillonnage des arêtes, on ne l'a pas

présentée ici. Sa mise en œuvre semble prometteuse, en offrant notamment l'opportunité de supprimer certains des traitements intermédiaires, et surtout en supprimant les ressources mémoires requises à leur implémentation.

Une deuxième perspective concrète se situe dans la possibilité, déjà évoquée, de gérer toutes les fonctions de distorsion avec une seule base commune de fonctions harmoniques sphériques. Pour arriver à ce résultat tout en conservant une compression efficace, dépendante de repères locaux aux échantillons, il est nécessaire de pouvoir effectuer de manière efficace des rotations sur les versions compressées des fonctions. L'usage d'une base de fonctions unique offre alors des avantages pratiques très nets, comme la capacité d'effectuer un filtrage spatial à coût nul, de supprimer quelques calculs, ou d'autoriser un *mip-mapping* qui permettrait de pallier en partie les problèmes de bruit.

Ces deux optimisations, conceptuellement simples sinon facilement réalisables, autoriseraient conjointement une technique plus efficace en mémoire utilisée et en temps de calcul.

D'autres pistes d'amélioration existent également, liées à la programmabilité croissante des matériels graphiques. En particulier, la décompression actuellement effectuée à l'aide de plusieurs passes pourrait probablement être réalisée beaucoup plus efficacement en une seule passe. Outre la suppression de lectures et d'écritures intermédiaires inutiles, il semble envisageable d'évaluer les fonctions harmoniques par les relations de récurrence qui existent entre elles, plutôt que par une évaluation polynomiale. Ce mode d'évaluation des fonctions de base offre l'avantage d'être nettement moins coûteux.

À plus long terme s'ouvrent également des perspectives pour tenter de rendre plus utilisable cette stratégie de rendu par précalcul. L'objectif, dans ce cas, est de parvenir à un mode de compression plus évolué que celui offert par les seules harmoniques sphériques. En effet, il apparaît assez clairement que l'on touche avec cette méthode les limites de la représentation par harmoniques sphériques, dans un contexte interactif : employées brutes, elles ne permettent de capter que des signaux de trop basses fréquences. De surcroît, il semble indispensable, pour rendre le stockage de données moins volumineux, d'introduire une forme de compression spatiale des données. Devant ce double objectif, il semble prometteur de se pencher sur les formes de compression plus évoluées proposées dans le contexte des *surface light fields*.

En particulier, il serait intéressant de parvenir à capturer plus nettement les discontinuités franches qui surviennent dans la réfraction. Avec sa forme de compression actuelle, la méthode proposée ne peut en effet se départir d'un lissage excessif.

Toutefois, la mise au point d'une compression plus performante n'est pas tâche aisée, les données de distorsion restant très peu prévisibles. C'est pourtant à cette seule condition qu'il est envisageable de ramener l'utilisation mémoire d'une telle technique par précalcul dans des limites plus raisonnables que les limites actuelles.

6.8 Extension aux *surface light fields*

La technique de visualisation de la réfraction que nous avons présentée repose sur la capacité à représenter et à manipuler une information directionnelle sur la surface des objets.



Figure 6.36: Textures directionnelles. Les données sont assimilables à un ensemble de textures, stockant la radiance émise pour un échantillonnage des directions d'observation. La radiance émise par le sol de la scène illustrée dans la dernière image est ici figurée, pour quatre directions. Il est possible de constater la variation de direction des reflets, ainsi que le déplacement du pic spéculaire dû à la lampe, selon la direction d'observation.

Nous avons décidé d'appliquer cette capacité de restitution de données directionnelles à une autre problématique, celle des *surface light fields*, et de simuler des textures de couleur directionnelles. Au lieu de stocker une information géométrique de distorsion, comme dans le cas de la réfraction, une information photométrique est alors encodée, et restituée interactivement. La visée d'une telle technique est l'affichage interactif des résultats de calculs d'illumination globale directionnelle, avec des scènes pouvant inclure des matériaux non diffus, légèrement spéculaires.

6.8.1 Calcul des données de radiance

La donnée finale produite par l'algorithme d'illumination globale directionnelle est une fonction $L(x, \omega)$, qui décrit la radiance émise par le point x d'une surface de la scène dans la direction ω . Cette fonction est connue en un certain nombre d'échantillons régulièrement disposés sur les différentes surfaces de la scène, ce qui revient à connaître un ensemble de textures codant la radiance émise pour différentes directions, comme l'illustre la figure 6.36. On remarquera la parenté immédiate de ces données avec les données de distorsion de la réfraction.

Nous n'aborderons pas ici en détail la problématique du calcul de la solution directionnelle au problème de l'illumination globale, dans la mesure où le calcul de cette solution ne fait pas partie des travaux que nous avons personnellement menés.

Dans les grandes lignes, la technique employée pour calculer cette solution fait usage à sa base d'une stratégie de *photon-mapping*, commençant par propager des photons dans la scène pour gérer l'illumination indirecte. En chaque échantillon spatial, l'irradiance est évaluée, en fonction de tous les transferts d'énergie vers l'échantillon dans la scène. Cette procédure est accélérée par une décomposition hiérarchique de la scène, elle-même guidée par des critères de visibilité et des critères de transfert énergétique. Ainsi sont constituées des entités suffisamment homogènes pour que leur transfert d'énergie vers l'échantillon puisse être évalué avec rapidité et précision. Selon le degré de décomposition de la scène, depuis un amas d'objets jusqu'à une face élémentaire, les procédures de calcul du transfert d'énergie varient. Elles incluent notamment l'usage des photons initialement distribués. Diverses optimisations sont mises à profit pour réduire le coût de ces procédures.

L'irradiance complète d'un échantillon étant calculée, la radiance directionnelle $L(\omega)$ en est extraite en considérant une convolution avec la *brdf* locale. Pour atteindre les meilleures performances, les échantillons sont traités de manière structurée, ce qui induit un bon usage des optimisations introduites. En particulier, les échantillons, structurés régulièrement sur les surfaces, emploient un mécanisme à deux niveaux de calcul de visibilité avec la décomposition de la scène, qui permet de réduire le nombre de tests de visibilité. Malgré ces optimisations, l'obtention de résultats directionnels au problème de l'illumination globale reste extrêmement coûteuse. Le volume de données générées est également très important, puisqu'il s'agit de l'échantillonnage d'une fonction à quatre dimensions.

6.8.2 Visualisation des données

Une fois obtenues, ces données peuvent être compressées par projection sur la base des fonctions harmoniques sphériques, comme on l'a décrit pour les données de distorsion de la réfraction.

L'usage de ces données est ensuite plus simple que pour la réfraction, puisqu'il est direct. Dans la mesure où les divers échantillons sont structurés comme des textures, et où le résultat de la décompression correspond directement à la radiance émise vers l'observateur, l'affichage des données décompressées est immédiat. Il ne requiert aucun traitement supplémentaire, contrairement au cas de la réfraction. La décompression peut donc être regardée comme la reconstruction d'une texture chromatique usuelle, dépendante du point de vue.

Une petite différence subsiste dans les données par rapport à la réfraction : les données de radiance calculées distinguent en effet *a priori* les composantes de l'illumination diffuse et directionnelle. Cette information supplémentaire permet de séparer les deux signaux, et de n'encoder que la partie directionnelle de l'illumination. Comme on la sait nécessairement positive, les valeurs reconstruites négatives peuvent être écrêtées, puis la partie diffuse ajoutée au signal. Un tel traitement permet d'éviter que le phénomène de Gibbs ne rogne la composante diffuse, comme on peut le voir dans la figure 6.37. En raison des forts reflets spéculaires qui peuvent apparaître dans les données de radiance, les oscillations parasites sont en effet rapidement gênantes, et requièrent un filtrage important.

6.8.3 Résultats

On peut juger des résultats offerts par la méthode dans la figure 6.38, pour trois scènes vues sous différents points de vue. Les résultats sont rendus en employant une compression d'ordre $l = 8$, pour un total de 81 fonctions de base, avec des images d'approximativement 1350×1100 pixels. Les performances sont mesurées sur une carte graphique Radeon 9700 Pro.

La première scène représente un environnement fermé, éclairé par une source lumineuse positionnée au plafond. Le plafond et trois des murs sont diffus et colorés, le sol et le mur à l'arrière-plan étant munis d'une *brdf* légèrement spéculaire. On peut constater dans les images le déplacement de la position du pic spéculaire produit par la source lumineuse en fonction de l'angle de vue sur le sol. On peut également noter les variations de mur à l'arrière-plan, notamment les mouvements du reflet du vase. La seconde scène présente un environnement

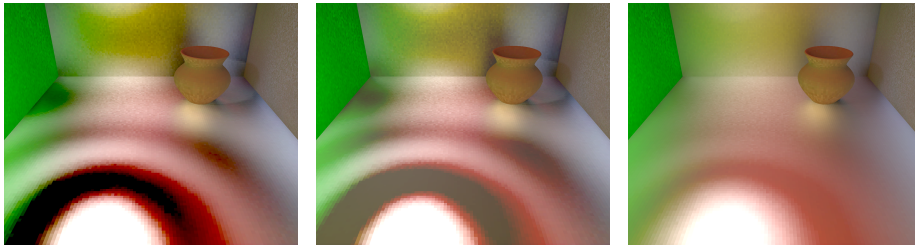


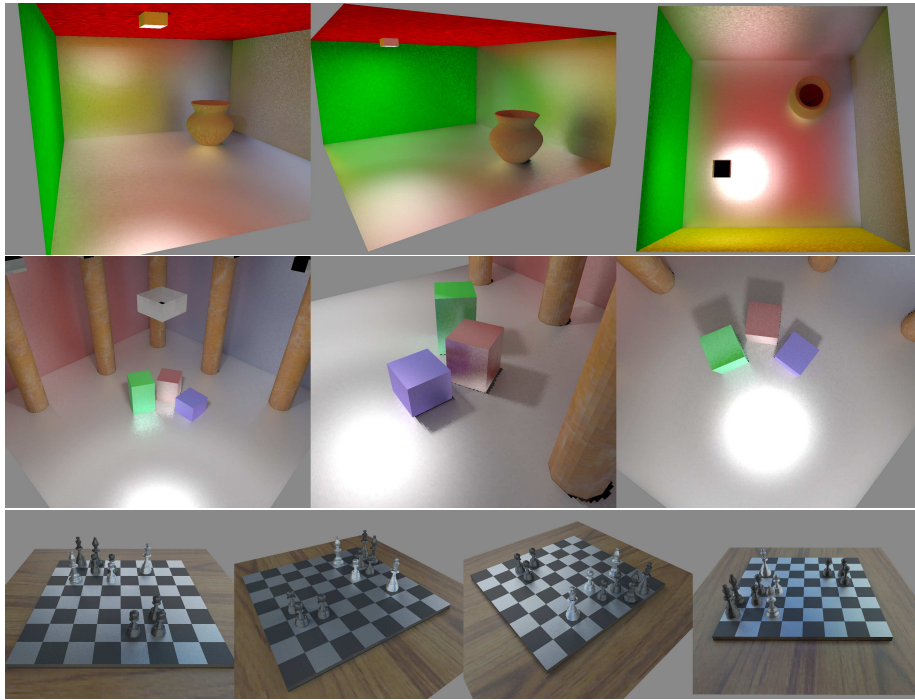
Figure 6.37: Phénomène de Gibbs. Les oscillations parasites apparaissent autour des reflets spéculaires qui sont des variations très importantes et très localisées du signal. À gauche, la radiance totale, combinant composante diffuse et composante directionnelle, est compressée. Au milieu, le signal directionnel est compressé seul, et ses valeurs reconstruites négatives sont écrêtées. À droite, le signal est séparé et un filtre fréquentiel est appliqué.

fermé similaire, mais contenant plus d'objets. Le sol et les trois cubes sont munis de brdf spéculaires. Le côté directionnel de l'illumination indirecte transparait essentiellement dans les reflets des cubes sur le sol et des cubes les uns sur les autres. Finalement, la dernière scène est constituée d'un échiquier éclairé par tout un environnement lumineux. Toutes les cases de l'échiquier ainsi que les pièces qui sont posées dessus sont munies de brdf spéculaires. L'illumination incidente de l'environnement étant moins directionnelle, on ne perçoit pas de pics francs dans l'éclairage, mais on peut tout de même distinguer une variation dans l'éclairage en fonction de la position d'observation, notamment dans sa couleur. Quelques reflets des pièces sur l'échiquier peuvent également être distingués.

Finalement, la figure 6.39 donne, à l'aide d'un test plus synthétique, un aperçu des résultats qu'il est possible d'obtenir à l'aide d'une compression par harmoniques sphériques de la radiance émise. Un quadrilatère coloré est posé sur un plan spéculaire, et toute la scène est éclairée par une source lumineuse située en hauteur, non visible dans les figures. On peut constater dans ces images une augmentation de la focalisation des reflets à mesure que le nombre de fréquences employées pour la compression augmente. On notera également que l'accroissement des fréquences s'accompagne d'une augmentation des oscillations parasites liées au phénomène de Gibbs, dans la couleur reconstruite. Des reflets fantômes apparaissent alors, incluant des couleurs non-présentes dans la scène, ce qui en fait des défauts assez visibles.

6.8.4 Bilan

Les résultats obtenus en représentant l'information directionnelle de radiance émise à l'aide d'harmoniques sphériques, restent mitigés. Si la technique permet indéniablement de représenter les variations directionnelles, elle pêche toutefois par deux aspects majeurs. Tout d'abord, la représentation découlant des fonctions harmoniques sphériques reste trop limitée en termes de précision : si les objets n'apparaissent plus diffus, seules les variations directionnelles les plus grossières de la radiance parviennent à être restituées. Ensuite, aucune compression spatiale des données n'est présente, ce qui n'autorise ni échantillonnage dense, ni application à grande échelle de la représentation sur des scènes importantes.



Scene	Nb. échantillons	Mémoire utilisée (Mo)	Vitesse (Hz)
Vase	25 k	10	60
Cubes	146 k	52	16
Jeu d'échecs	174 k	65	11

Figure 6.38: Exemples de résultats. Sont également présentées l'utilisation mémoire et la vitesse de rendu des différentes scènes dans le tableau. Les performances sont mesurées sur une carte graphique Radeon 9700 Pro. Dans le cas des deux premiers environnements, fermés, certains des murs sont omis de la visualisation.

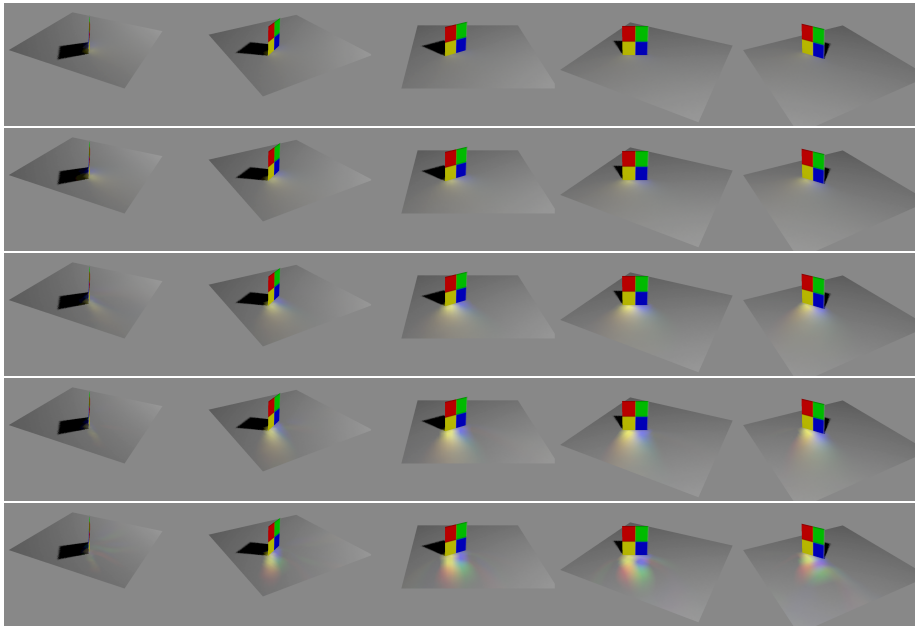


Figure 6.39: Variation de la compression. De haut en bas, $l = 0$ qui correspond à une illumination purement diffuse, $l = 1$, $l = 3$, $l = 5$, et $l = 9$. Différents points de vue sont présentés.

Par ailleurs, en termes de qualité de restitution, les oscillations parasites liées à une représentation en fréquence sont très promptes à se manifester, les données d'illumination ayant naturellement tendance à contenir de fortes variations dues aux reflets spéculaires. Les variations d'amplitude dans les données sont en effet beaucoup plus marquées que dans le cas de la réfraction, cas dans lequel on s'intéresse à coder les variations de direction d'un vecteur unitaire.

Comparée aux techniques de représentation des *surface light fields* en pointe, l'utilisation brute des harmoniques sphériques telle qu'appliquée ici, ne semble donc pas offrir d'avantage, puisqu'elle est loin de présenter les mêmes taux de compression et la même qualité, et qu'elle ne bénéficie pas non plus d'une exécution rapide. Le fait que les harmoniques sphériques correspondent à une base de représentation statique, alors que les techniques en pointe dans la représentation des *light fields* s'adaptent à la forme des données à compresser, explique en partie que l'on n'obtienne pas des performances plus satisfaisantes.

Conclusion

Comme on l'aura compris, les travaux qui ont fait l'objet de cette thèse relèvent à la fois du domaine de la simulation et de celui de la visualisation. Étant donné l'importance prépondérante de la simulation dans cette étude, nous y reviendrons après avoir d'abord discuté de la visualisation.

Visualisation Les travaux que nous avons effectués à des fins de visualisation, même s'ils ont tous trait aux liquides, ont concerné des points précis et ne gravitent pas autour d'une problématique commune. Nous examinerons donc séparément les travaux liés à la méthode MAC, et ceux touchant à la réfraction.

Visualisation de simulations MAC Du point de vue de la visualisation, les efforts déployés pour exploiter sans reconstruction de surface les données issues des simulateurs MAC, c'est-à-dire des nuages volumiques de points, ont abouti à des résultats n'offrant qu'une satisfaction partielle. Si la visualisation rapide de ces données par une technique de rendu volumique, assistée par le matériel, est intéressante pour juger des animations, ses utilisations restent limitées, et sa qualité n'est pas suffisante pour être étendue au champ d'une visualisation réaliste. Par ailleurs, l'intégration de ces nuages volumiques au sein d'une visualisation réaliste par lancé de rayons s'est révélée décevante. L'exploitation des nuages de points volumiques semble de prime abord attrayante, mais le manque de densité des nuages, à l'échelle des détails de la surface, n'autorise finalement pas une visualisation directe de bonne qualité.

Perspectives pour la visualisation de simulations MAC Dans la mesure où l'utilisation de représentations du liquide effectuées uniquement à base de marqueurs tend à disparaître dans la perspective d'une simulation de grande qualité, l'attrait de l'intégration des nuages de points au sein des algorithmes de lancé de rayons a fortement diminué. Le type de représentation qui est d'actualité dans ces simulations est en effet plutôt constitué d'une représentation hybride, combinant surfaces implicites et marqueurs. L'intérêt de poursuivre les travaux que nous avons menés pour représenter directement un liquide par un nuage de points ne semble donc pas justifié, au vu de l'importance des lacunes à combler. Par exemple, chercher à traiter les nuages de points résultant des simulations, pour en améliorer ultérieurement la visualisation en conservant l'approche que nous avons testée, ne présente probablement pas d'avantage significatif sur le fait de reconstruire une surface.

Par contre, la problématique de la combinaison d'une simulation de liquides à surface libre avec sa visualisation, toutes deux interactives, semble offrir un

objectif prometteur. Le recours à un simulateur MAC, avec ses marqueurs, n'est pas à négliger dans ce cas en raison de sa simplicité d'utilisation et de ses bonnes performances. La recherche d'un mode de visualisation de bonne qualité adapté aux données discrètes du simulateur, paraît dans le contexte interactif une voie dont l'exploration semble encourageante. Cet objectif reste également valable dans le cas des simulations SPH, non Eulériennes, qui ne produisent pas de *levelset*.

Dans ce contexte, il semble pertinent de recourir à des rendus par projection à l'image des points, de type *splatting*. L'emploi d'une combinaison performante des primitives à l'image permettrait en effet d'éviter l'écueil de la faible densité des points auquel s'est heurté le rendu par lancé de rayon que nous avons testé. Un tel mode de rendu demande alors d'explorer, dans un cadre interactif, l'adjonction d'effets comme la réfraction ou l'atténuation due à un liquide trouble, pour obtenir une visualisation pertinente. La nature particulière, volumique, des nuages de points issus des simulations offre alors une différence notable à étudier, par rapport à la structuration surfacique des données plus communément employée par ces algorithmes.

Les représentations implicites par *levelset* étant populaires, il serait également intéressant de pouvoir visualiser ces fonctions implicites de manière interactive. Par leur caractère de fonction de distance, elles peuvent autoriser des optimisations spécifiques, de stockage ou d'accélération de la vitesse de rendu, dans le contexte d'un rendu volumique interactif.

Représentation interactive de la réfraction La visualisation interactive des phénomènes de réfraction, tirant parti du matériel graphique, se révèle plus satisfaisante au niveau des résultats obtenus, comme en témoigne sa publication [52]. Bien qu'elle reste encore perfectible au niveau du volume de stockage des données, déjà fortement compressées, elle autorise la visualisation d'effets de réfraction complexes sur des objets de géométrie arbitraire. L'approche suivie rencontre cependant des difficultés de passage à l'échelle d'une capture spatiale à grande résolution. Elle se trouve également limitée en fréquence, ce qui fait obstacle à la capture des détails fins et des discontinuités de la réfraction. Le dépassement de ces limitations exigerait la mise au point d'un mode de compression nettement plus performant, tâche délicate au vu de la nature très aléatoire et très variable des déviations engendrées par la réfraction.

Perspectives dans le traitement de la réfraction Des perspectives existent pour prolonger l'approche de gestion de la réfraction que nous avons proposée.

À court terme, il est en effet possible de réaliser un certain nombre de modifications de l'algorithme proposé, pour le rendre plus souple et plus performant. Deux axes sont pour cela à finaliser : celui qui permet d'employer une paramétrisation très peu contrainte de la surface, et celui qui consiste à n'employer qu'une seule base de fonctions pour représenter les informations directionnelles. L'action conjointe de ces deux modifications devrait alors permettre de supprimer une partie des traitements requis par l'approche et d'en diminuer le coût mémoire. Une autre piste à court terme est de profiter des avancées du matériel graphique pour optimiser largement la décompression, en autorisant une évaluation récursive beaucoup plus performante.

Néanmoins, si ces améliorations à court terme sont susceptibles d'augmenter sensiblement les performances, ou de diminuer quelque peu l'utilisation mémoire, elles ne peuvent, à elles seules, rendre la méthode utilisable à grande échelle. Pour pallier le défaut majeur de la méthode, qui est une consommation excessive de mémoire, il est nécessaire de parvenir à un mode de compression des données bien plus performant.

À plus long terme, dans cette perspective, et en se référant aux techniques plus performantes du domaine des *surface light fields*, on peut dégager deux axes pour améliorer la compression. Tout d'abord, une forme de compression spatiale doit nécessairement être introduite pour espérer obtenir un échantillonnage de la surface des objets à la fois dense et peu coûteux. Ensuite, il paraît nécessaire de se tourner vers un schéma de compression capable de s'adapter aux données à représenter, à la manière des techniques de réduction de dimension, et au contraire des harmoniques sphériques qui possèdent un caractère trop rigide pour être très efficaces.

Simulation des fluides Les travaux que nous avons menés dans le cadre de la simulation de fluides à destination de l'infographie, contrairement, à ceux qui ont trait au rendu, convergent tous vers un seul et même objectif : parvenir à simuler une interaction automatique entre liquide et solides.

Démarche suivie et résultats obtenus pour la simulation de fluides

Pour ce qui concerne la simulation, nous avons développé et publié [51] un mode d'interaction entre solides déformables et fluides simulés par la méthode MAC, mode d'interaction qui, s'il produit des résultats de qualité visuelle probante, n'épargne pas à son animateur une mise en œuvre ardue, en raison de son caractère empirique.

Il nous a semblé intéressant, pour pallier les défauts liés à la démarche empirique, de pousser le réalisme plus avant en nous appuyant sur une formulation physique stricte d'un problème réel d'interaction entre deux objets continus. Pour pouvoir poser un tel problème, il est nécessaire, pour le simulateur de fluides, de s'accommoder de domaines de simulations arbitraires, ce dont les simulateurs s'appuyant sur des grilles Cartésiennes sont incapables. C'est pourquoi nous nous sommes tournés vers un mode de simulation des équations de Navier-Stokes *via* une méthode par éléments finis capable de gérer naturellement de tels domaines. Cette forme de simulation présente en outre l'avantage de pouvoir s'abstraire de tout problème de grille, et lève toute ambiguïté quant à la gestion des frontières du domaine.

Nous avons donc réalisé un solveur fondé sur cette méthode. En raison de la complexité de la méthode, et du handicap d'une convergence parfois erratique, ce solveur ne peut déjà être employé pour une production d'images. Il nous a toutefois permis de constater les avantages de la méthodes des éléments finis dans la gestion de domaines complexes, à côté de défauts évidents, comme une consommation très importante en ressources de calcul. Pour réaliser ce solveur, nous avons quitté le domaine de l'infographie pure, pour nous tourner vers des techniques apparentées à celles de la physique et des mathématiques appliquées.

Perspectives liées la simulation de fluides Dans le cadre de la simulation des fluides en infographie, et de leur interaction avec des solides, on peut

distinguer plusieurs axes d'amélioration.

La raison principale qui nous a poussés à nous tourner vers la simulation par éléments finis est la possibilité offerte de gérer des domaines arbitraires. L'apparition récente de méthodes par différences finies compatibles avec des domaines arbitraires, ouvre la voie à la recherche de simulations d'interactions entre solides et fluide, en employant des méthodes de ce type.

Nous pensons toutefois que la simulation par éléments finis peut rester une technique intéressante dans le contexte de l'infographie, et pas uniquement dans un objectif d'interaction. Prouver cet intérêt pour l'infographie constitue une perspective à court terme, et passe par l'obtention d'un simulateur réellement opérant, capable de produire des résultats de manière robuste. Sur ce point, les mathématiques appliquées recèlent un vaste arsenal de techniques, qu'il se révélera sans nul doute fructueux d'explorer. Il est cependant clair que ces techniques restent souvent complexes, ce qui nous a pour le moment empêchés de les mettre en œuvre. Elles incluent par exemple l'emploi de préconditionneurs adaptés, mais dont l'application nécessite elle-même la résolution de problèmes conséquents.

À moyen terme, l'on pourra évaluer plus justement l'adéquation de la technique des éléments finis avec les contraintes de l'infographie, en particulier son utilisation dans un contexte de liquide à surface libre, ou dans un contexte d'interaction avec des objets déformables, notre motivation initiale.

Une autre application concrète où la simulation de liquides par éléments finis peut être très pertinente est la simulation de gouttes et de ruissellements sur des objets. Une telle simulation requiert en effet de connaître précisément la surface du liquide simulé, de pouvoir y appliquer des effets de tension de surface, tout en simulant des liquides peu volumineux mais pouvant se déplacer sur des distances importantes : ces différents points sont susceptibles de trouver une réponse grâce aux points forts de la méthode de résolution par éléments finis, méthode qui n'a pas à se préoccuper d'un maillage fixe de l'environnement, et peut explicitement représenter la géométrie de la surface.

Quelles que soient les optimisations ou les variantes de la méthode employées, on peut néanmoins prévoir que la simulation par éléments finis de problèmes liés aux fluides restera coûteuse.

En mesure d'introduire un plus grand réalisme dans les scènes, les simulations par éléments finis, qui emploient un maillage non structuré, pourraient également offrir un avantage dans un objectif de contrôlabilité des animations. En effet, combiné à un cadre de déformation hiérarchique, pour proposer un outil local aussi bien que global, le maillage de simulation pourrait constituer un support valable du contrôle de la simulation, relativement intuitif pour un animateur.

Dans un cadre plus large, et sans rester limité à la problématique de l'interaction ou à un type de simulation particulier, il apparaît que d'autres approches existent en simulation de fluide, non encore testées en infographie. Il semble notamment que les approches multi-grilles, opérant une résolution hiérarchique des équations, aient marqué un net tournant au niveau des performances dans la résolution des simulations numériques. Qu'elles soient couplées avec des méthodes par différences finies ou par éléments finis, ces méthodes apparaissent d'un grand intérêt.

Bien que cette affirmation relève d'un lieu commun, il paraît clair que les perspectives liées à une accélération des algorithmes de résolution par l'utili-

sation du matériel graphique sont prometteuses. La convergence actuelle du matériel graphique avec des unités de calculs généralistes mérite en effet d'être soulignée, estompant les contraintes algorithmiques drastiques qui rendaient ce matériel peu utilisable. Leur utilisation dans le cadre de simulations réalistes, et pas seulement dans le cadre de simulations interactives assez approximatives, pourrait parvenir à un point intéressant.

Discussion Pour conclure, nous proposons de poser la question de la pertinence de recherche du réalisme dans l'infographie : s'il est évident que l'on recherche des résultats d'apparence visuelle toujours plus réaliste, se pose la question du prix que l'on est prêt à payer pour obtenir ces mêmes résultats voulus plus réalistes.

L'explosion des travaux dans le domaine de la simulation de fluides en infographie, et notamment la reprise de travaux des communautés de physique et de mathématiques, au cours des cinq ou dix dernières années, a en effet permis de développer des simulateurs dont le réalisme est déjà plus que satisfaisant, et qui sont largement utilisables. Employées de manière judicieuse, avec leurs résultats intégrés à une scène globale, les animations résultantes sont largement en mesure de leurrer bon nombre d'observateurs : les mouvements paraissent plausibles, et les images qui en sont tirées sont photo-réalistes. Sans affirmer que les méthodes actuelles apportent une réponse exhaustive à la thématique, il n'est pas interdit de poser le problème du bénéfice visuel que peuvent apporter des améliorations, bénéfice nécessairement faible par rapport à des techniques déjà éprouvées et performantes.

Au regard de méthodes déjà complexes et généralement probantes, l'investissement pour obtenir des techniques dites plus réalistes risque de voir croître fortement son coût en ressources et en complexité d'appréhension et d'implantation, à l'exemple de l'emploi des méthodes par éléments finis. En particulier, et comme d'ailleurs illustré par cette méthode, il est nécessaire de recourir à des techniques peu courantes en informatique graphique, très souvent associées au calcul scientifique et aux mathématiques appliquées. Déterminer si de telles techniques valent cet investissement demande de distinguer entre réalisme physique et propriétés visuelles attendues de la simulation. Loin de restreindre le champ des développements à réaliser, cette question conduit à explorer pour l'infographie des problématiques de recherche de compromis, avec les notions de simplifications des modèles, ou d'approximations, avec pour finalité des simulations plus performantes, voire interactives.

Dans tous les cas, et que l'approche choisie prenne appui sur un modèle physique des plus exacts, ou qu'elle adopte une méthode de résolution perfectionnée et efficace, il paraît néanmoins nécessaire de mieux tenir compte du fait que la simulation est fondée sur la résolution d'équations, et de se rapprocher d'une vision plus mathématique du problème. Parvenir à produire de meilleurs résultats demande en effet d'abandonner quelque peu la vision communément admise en infographie, de considérer les simulateurs comme de simples mécaniques dédiées à produire des animations à partir de la description d'une scène, pour adopter la vision plus mathématique et plus stricte, dans laquelle la même finalité est appréciée sous l'angle de la résolution d'un problème mis en équations. Si l'adoption du point de vue commun de l'infographie et l'emploi de procédures simplifiées est, bien sûr, possible et même nécessaire, une fois la simulation par-

faitement maîtrisée — l'objectif n'étant pas l'emploi de simulateurs dédiés à l'ingénierie — un développement des simulations en adoptant le point de vue mathématique, plus rigoureux, semble incontournable pour progresser.

Annexe A

Notations et rappels mathématiques

Sont rappelées ci-dessous quelques notations concernant des champs définis dans l'espace Euclidien $E = \mathbb{R}^3$, ayant plus particulièrement trait aux opérateurs différentiels classiques agissant sur eux. On donne également le développement de quelques identités utilisées dans l'établissement des équations de la mécanique des milieux continus et des diverses équations gouvernant les fluides.

A.1 Définition des opérateurs différentiels

Les champs manipulés sont supposés être suffisamment différentiables pour qu'un sens puisse être donné aux quantités définies. Ils sont exprimés dans un système de coordonnées Cartésien $(O, x, y, z) = (O, x_1, x_2, x_3)$. Il convient de rappeler que les définitions données ici ne sont d'ailleurs valables que dans le cas de coordonnées Cartésiennes : dans le cas de systèmes de coordonnées plus complexes, tel le système curviligne, les définitions des opérateurs sont plus compliquées.

Gradient Étant donné un champ scalaire $f : E \rightarrow \mathbb{R}$, l'opérateur gradient ∇ est défini comme :

$$\nabla f = [\nabla f]_{i=1,2,3} = \left[\frac{\partial f}{\partial x_i} \right]_{i=1,2,3} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}$$

Le gradient d'un champ scalaire est donc un champ vectoriel.

Étant donné un champ vectoriel $u = [u_i]_{i=1,2,3} = [u_x, u_y, u_z]^T : E \rightarrow \mathbb{R}^3$, on définit également un opérateur de gradient comme le gradient de chaque

composante :

$$\nabla u = [\nabla u]_{i,j=1,2,3} = \left[\frac{\partial u_j}{\partial x_i} \right]_{i,j=1,2,3} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_y}{\partial x} & \frac{\partial u_z}{\partial x} \\ \frac{\partial u_x}{\partial y} & \frac{\partial u_y}{\partial y} & \frac{\partial u_z}{\partial y} \\ \frac{\partial u_x}{\partial z} & \frac{\partial u_y}{\partial z} & \frac{\partial u_z}{\partial z} \end{bmatrix}$$

Le gradient d'un champ de vecteurs est donc défini comme un champ de tenseurs du second ordre.

On définit également l'opérateur ∇^T par :

$$\nabla^T u = [\nabla u]^T = \left[\frac{\partial u_i}{\partial x_j} \right]_{i,j=1,2,3}$$

Divergence Étant donné un champ vectoriel $u = [u_x, u_y, u_z]^T : E \rightarrow \mathbb{R}^3$, l'opérateur de divergence $\nabla \cdot$ est défini comme :

$$\nabla \cdot u = \sum_{i=1}^3 \frac{\partial u_i}{\partial x_i}$$

La divergence d'un champ vectoriel est donc un champ scalaire. Un champ u dont la divergence est nulle $\nabla \cdot u = 0$ est appelé un champ solénoïdal.

Étant donné un champ de tenseurs du second ordre $\tau = [\tau_{ij}]_{i,j=1,2,3} : E \rightarrow \mathbb{R}^3 \otimes \mathbb{R}^3$, l'opérateur de divergence $\nabla \cdot$ est défini comme :

$$\nabla \cdot \tau = \left[\sum_i \frac{\partial \tau_{ij}}{\partial x_j} \right]_{j=1,2,3} = \begin{bmatrix} \nabla \cdot [\tau_{i1}]_{i=1,2,3} \\ \nabla \cdot [\tau_{i2}]_{i=1,2,3} \\ \nabla \cdot [\tau_{i3}]_{i=1,2,3} \end{bmatrix}$$

La divergence d'un champ de tenseurs du second ordre est donc un champ de vecteurs.

Rotationnel Étant donné un champ vectoriel $u = [u_x, u_y, u_z]^T : E \rightarrow \mathbb{R}^3$, il est également possible de définir l'opérateur rotationnel $\nabla \times$ comme :

$$\nabla \times u = \begin{bmatrix} 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial x} \\ -\frac{\partial}{\partial z} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$$

Le rotationnel d'un champ vectoriel u est donc un autre champ vectoriel. Si le rotationnel est partout nul, le champ u est dit irrotationnel.

Laplacien Pour finir, l'opérateur Laplacien Δ d'un champ scalaire $f : E \rightarrow \mathbb{R}$ est défini comme :

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^3 \frac{\partial^2 f}{\partial x_i^2}$$

De même, il est possible de définir un Laplacien pour un champ vectoriel $u = [u_x, u_y, u_z]^T : E \rightarrow \mathbb{R}^3$ comme :

$$\Delta u = \begin{bmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{bmatrix}$$

A.2 Interprétation des opérateurs

On va maintenant chercher à donner une interprétation physique, plus intuitive, des différents opérateurs précédemment définis.

Pour un champ scalaire f , le gradient ∇f représente le champ des variations de f . La direction engendrée par ∇f est celle de plus grande variation du champ, toujours perpendiculaire aux isocontours de f , la norme $|\nabla f|$ indiquant l'intensité de cette variation.

Pour un champ vectoriel u , la divergence $\nabla \cdot u$ est une mesure de l'expansion ou de la contraction locale du champ. La divergence peut en effet être également définie comme l'intégrale du flux traversant la frontière d'un volume infinitésimal Ω :

$$\nabla \cdot u = \lim_{\Omega \rightarrow 0} \frac{1}{|\Omega|} \oint_{\partial\Omega} u \cdot n \, ds$$

De ces deux interprétations, il est possible d'appréhender le Laplacien Δf d'un champ scalaire f comme un indicateur de la variation du gradient ∇f , sans tenir compte de son orientation. Cette liaison est matérialisée dans une formule exprimant la courbure κ d'une isosurface de f :

$$\kappa = \nabla \cdot \left(\frac{\nabla f}{|\nabla f|} \right)$$

Dans le cas où f représente une fonction de distance, la courbure est donc égale au Laplacien.

Finalement, $\nabla \times u$, le rotationnel d'un champ vectoriel u , est une mesure de la circulation locale du champ, c'est-à-dire de sa rotation sur lui-même. La norme $|\nabla \times u|$ indique l'intensité de cette circulation, la direction définie par $\nabla \times u$ indiquant la perpendiculaire au plan de rotation. Ces propriétés sont plus clairement visibles lorsque le rotationnel est défini à partir de la circulation du champ le long d'une courbe :

$$(\nabla \times u) \cdot n = \lim_{\Omega \rightarrow 0} \frac{1}{|\Omega|} \oint_{\partial\Omega} u \cdot dl$$

où $\partial\Omega$ est la courbe frontière bordant une surface infinitésimale Ω de normale unitaire n .

A.3 Dérivée directionnelle

Si la variation d'un champ f , selon les axes du système de coordonnées dans lequel il est exprimé, est directement déductible des composantes de ∇f , on peut chercher à exprimer cette variation selon une direction n unitaire quelconque, comme illustré dans la figure A.1

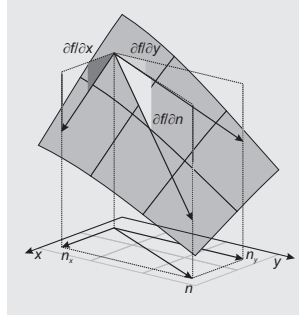


Figure A.1: Calcul de la dérivée directionnelle. La situation illustrée correspond au cas d'un champ bidimensionnel.

Par définition :

$$\frac{\partial f}{\partial n} = \lim_{h \rightarrow 0} \frac{f(x + hn) - f(x)}{h}$$

ce qui, en posant $F(h) = f(x + hn)$ se réduit à

$$\begin{aligned} \frac{\partial f}{\partial n} &= \lim_{h \rightarrow 0} \frac{F(h) - F(0)}{h} = \frac{dF}{dh} \\ &= \frac{d}{dh} f(x + n_x h, y + n_y h, z + n_z h) \\ &= \frac{\partial f}{\partial x} \cdot \frac{\partial(x + n_x h)}{\partial h} + \frac{\partial f}{\partial y} \cdot \frac{\partial(y + n_y h)}{\partial h} + \frac{\partial f}{\partial z} \cdot \frac{\partial(z + n_z h)}{\partial h} \\ &= \frac{\partial f}{\partial x} \cdot n_x + \frac{\partial f}{\partial y} \cdot n_y + \frac{\partial f}{\partial z} \cdot n_z \\ &= n \cdot \nabla f \end{aligned}$$

Par conséquent, $n \cdot \nabla f$ représente la dérivée de f dans la direction n , que l'on trouve parfois notée $\frac{\partial f}{\partial n}$. Le gradient est ainsi une description complète de la tangence au champ dans n'importe quelle direction.

A.4 Identités et propriétés utiles

On pose $\partial_i = \partial_{x_i} = \frac{\partial}{\partial x_i}$ et $\partial_{ii} = \partial_i \partial_i$.

Les deux propriétés suivantes :

$$\begin{aligned} \nabla \times \nabla f &= 0 & \forall f \in E \rightarrow \mathbb{R} \\ \nabla \cdot \nabla \times u &= 0 & \forall u \in E \rightarrow \mathbb{R}^3 \end{aligned}$$

se démontrent très simplement :

$$\begin{aligned} \nabla \times \nabla f &= \begin{bmatrix} \partial_y \partial_z f - \partial_z \partial_y f \\ \partial_z \partial_x f - \partial_x \partial_z f \\ \partial_x \partial_y f - \partial_y \partial_x f \end{bmatrix} \\ &= 0 \end{aligned}$$

$$\begin{aligned}
\nabla \cdot \nabla \times u &= \partial_x [\partial_y u_z - \partial_z u_y] + \partial_y [\partial_z u_x - \partial_x u_z] + \partial_z [\partial_x u_y - \partial_y u_x] \\
&= [\partial_{yz} - \partial_{zy}]u_x + [\partial_{xz} - \partial_{zx}]u_y + [\partial_{xy} - \partial_{yx}]u_z \\
&= 0
\end{aligned}$$

On présente maintenant quelques identités employées dans la dérivation des équations de la mécanique continue, pour un système de coordonnées Cartésien.

Soit u un champ vectoriel, p et ρ des champs scalaires, I le tenseur identité de rang deux, δ le symbole de Kronecker, et \otimes le produit tensoriel.

$$\begin{aligned}
\nabla \cdot (\rho u) &= \sum_i \frac{\partial \rho u_i}{\partial x_i} \\
&= \sum_i \left[\rho \frac{\partial u_i}{\partial x_i} + \frac{\partial \rho}{\partial x_i} u_i \right] \\
&= \rho \sum_i \frac{\partial u_i}{\partial x_i} + \sum_i \frac{\partial \rho}{\partial x_i} u_i \\
&= \rho \nabla \cdot u + (u \cdot \nabla) \rho
\end{aligned}$$

$$\begin{aligned}
\nabla \cdot \nabla^T u &= \left[\sum_i \frac{\partial}{\partial x_i} \left[\frac{\partial u_i}{\partial x_j} \right] \right]_{j=1,2,3} \\
&= \left[\frac{\partial}{\partial x_j} \sum_i \frac{\partial u_i}{\partial x_i} \right]_{j=1,2,3} \\
&= \left[\frac{\partial}{\partial x_j} \nabla \cdot u \right]_{j=1,2,3} \\
&= \nabla \nabla \cdot u
\end{aligned}$$

$$\begin{aligned}
\nabla \cdot (\rho u \otimes u) &= \nabla \cdot [\rho u_i u_j] \\
&= \left[\sum_i \frac{\partial}{\partial x_i} ((\rho u_i) u_j) \right]_{j=1,2,3} \\
&= \left[\sum_i \frac{\partial}{\partial x_i} (\rho u_i) u_j + (\rho u_i) \frac{\partial u_j}{\partial x_i} \right]_{j=1,2,3} \\
&= \left[u_j \sum_i \frac{\partial}{\partial x_i} (\rho u_i) + \rho \sum_i u_i \frac{\partial u_j}{\partial x_i} \right]_{j=1,2,3} \\
&= [u_j \nabla \cdot (\rho u) + \rho (u \cdot \nabla) u_j]_{j=1,2,3} \\
&= u \nabla \cdot (\rho u) + \rho (u \cdot \nabla) u
\end{aligned}$$

$$\begin{aligned}
\nabla \cdot (pI) &= \left[\sum_i \frac{\partial [pI]}{\partial x_i} \right]_{j=1,2,3} \\
&= \left[\sum_i \frac{\partial p \delta_{ij}}{\partial x_i} \right]_{j=1,2,3} \\
&= \left[\frac{\partial p \delta_{jj}}{\partial x_j} + \sum_{j \neq i} \frac{\partial p \delta_{ij}}{\partial x_i} \right]_{j=1,2,3} \\
&= \left[\frac{\partial p}{\partial x_j} \right]_{j=1,2,3} \\
&= \nabla p
\end{aligned}$$

$$\begin{aligned}
\nabla \times \Delta u &= \nabla \times \begin{bmatrix} \Delta u_x \\ \Delta u_y \\ \Delta u_z \end{bmatrix} \\
&= \begin{bmatrix} \partial_y \Delta u_z - \partial_z \Delta u_y \\ \partial_z \Delta u_x - \partial_x \Delta u_z \\ \partial_x \Delta u_y - \partial_y \Delta u_x \end{bmatrix} \\
&= \begin{bmatrix} \partial_y \sum_i \partial_{ii} u_z - \partial_z \sum_i \partial_{ii} u_y \\ \partial_z \sum_i \partial_{ii} u_x - \partial_x \sum_i \partial_{ii} u_z \\ \partial_x \sum_i \partial_{ii} u_y - \partial_y \sum_i \partial_{ii} u_x \end{bmatrix} \\
&= \begin{bmatrix} \sum_i \partial_{ii} (\partial_y u_z - \partial_z u_y) \\ \sum_i \partial_{ii} (\partial_z u_x - \partial_x u_z) \\ \sum_i \partial_{ii} (\partial_x u_y - \partial_y u_x) \end{bmatrix} \\
&= \Delta \nabla \times u
\end{aligned}$$

Rotationnel d'un terme advectif pour un fluide incompressible On rappelle que par définition $\omega = \nabla \times u$ et que $\nabla \cdot u = 0$ car le fluide est incompressible.

Pour deux champs vectoriels a et b , on suppose par ailleurs connues les deux identités suivantes :

$$\begin{aligned}
\nabla(a \cdot b) &= (a \cdot \nabla)b + (b \cdot \nabla)a + a \times (\nabla \times b) + b \times (\nabla \times a) \\
\nabla \times (a \times b) &= a \nabla \cdot b - b \nabla \cdot a + (b \cdot \nabla)a - (a \cdot \nabla)b
\end{aligned}$$

On considère la première identité avec $a = b = u$:

$$\nabla(u \cdot u) = (u \cdot \nabla)u + (u \cdot \nabla)u + u \times (\nabla \times u) + u \times (\nabla \times u)$$

donc :

$$(u \cdot \nabla)u = \frac{1}{2} \nabla(u \cdot u) + (\nabla \times u) \times u = \frac{1}{2} \nabla(u \cdot u) + \omega \times u$$

Par conséquent

$$\begin{aligned}
\nabla \times [(u \cdot \nabla)u] &= \frac{1}{2} \nabla \times \nabla(u \cdot u) + \nabla \times (\omega \times u) \\
&= \omega \nabla \cdot u - u \nabla \cdot \omega + (u \cdot \nabla)\omega - (\omega \cdot \nabla)u
\end{aligned}$$

Comme $\nabla \cdot u = 0$ et $\nabla \cdot \omega = \nabla \cdot (\nabla \times u) = 0$, on obtient finalement :

$$\nabla \times [(u \cdot \nabla)u] = (u \cdot \nabla)\omega - (\omega \cdot \nabla)u$$

Bibliographie

- [1] Pierre ALLIEZ, David COHEN-STEINER, Mariette YVINEC et Mathieu DESBRUN : Variational tetrahedral meshing. *In SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 617–625, New York, NY, USA, 2005. ACM Press.
- [2] Alexis ANGELIDIS et Fabrice NEYRET : Simulation of Smoke based on Vortex Filament Primitives. *In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, pages 87–96, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [3] Adam W. BARGTEIL, Tolga G. GOKTEKIN, James F. O'BRIEN et John A. STRAIN : A semi-lagrangian contouring method for fluid simulation. *ACM Transactions on Graphics*, 25(1):19–38, 2006. ISSN 0730-0301.
- [4] Daniel BARRERO, Mathias PAULIN et René CAUBET : A physics based multi-resolution model for the simulation of turbulent gases and combustion. *In N. MAGNENAT-THALMANN et D. THALMANN, éditeurs : Computer Animation and Simulation '99*, SpringerComputerScience, pages 177–186. Springer-Verlag Wien New York, 1999. Proceedings of the Eurographics Workshop in Milano, Italy, September 7–8, 1999.
- [5] Daniel BARRERO, Mathias PAULIN et René CAUBET : Simulating turbulent combustion. *In Graphicon 2000, Moscou*, 2000.
- [6] Richard BARRETT, Michael BERRY, Tony F. CHAN, James DEMMEL, June M. DONATO, Jack DONGARRA, Viktor EIJKHOUT, Roldan POZO, Charles ROMINE et Henk Van der VORST : *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.
- [7] Byron BASHFORTH et Yee-Hong YANG : Physics-based explosion modeling. *Graphical Models*, 63(1):21–44, janvier 2001.
- [8] Kenneth E. BATCHER : Sorting networks and their applications. *In Proceedings of the AFIPS Spring Joint Conference*, volume 32, pages 307–314, 1968.
- [9] William BAXTER, Yuanxin LIU et Ming C. LIN : A viscous paint model for interactive applications. *Computer Animation and Virtual Worlds*, 15 (3–4):433–441, 2004.

- [10] Michele BENZI, Gene H. GOLUB et Jörg LIESEN : Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, may 2005.
- [11] Michel BERCOVIER et Michael ENGELMAN : A finite element for the numerical solution of viscous incompressible flows. *Journal of Computational Physics*, 30(2):181–201, février 1979.
- [12] Jeff BOLZ, Ian FARMER, Eitan GRINSPUN et Peter SCHRÖDER : Sparse matrix solvers on the gpu : Conjugate gradients and multigrid. *ACM Transactions on Graphics*, 22(3):917–924, 2003. ISSN 0730-0301.
- [13] Odus R. BURGGRAF : Analytical and numerical studies of the structure of steady separated flows. *Journal of Fluid Mechanics*, 24(1):113–151, Jan 1966.
- [14] Brian CABRAL, Nancy CAM et Jim FORAN : Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VVS '94 : Proceedings of the 1994 symposium on Volume visualization*, pages 91–98, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-741-3.
- [15] Mark CARLSON, Peter J. MUCHA, III R. BROOKS VAN HORN et Greg TURK : Melting and flowing. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 167–174, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-573-4.
- [16] Mark CARLSON, Peter J. MUCHA et Greg TURK : Rigid fluid : Animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics*, 23(3):377–384, 2004. ISSN 0730-0301.
- [17] Jim X. CHEN et Niels da VITORIA LOBO : Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations. *Graphical Models and Image Processing*, 57(2):107–116, mars 1995.
- [18] Jim X. CHEN, Niels da VITORIA LOBO, Charles E. HUGHUES et J. Michael MOSHELL : Real-time fluid simulation in a dynamic virtual environment. *Computer Graphics and Applications*, 17(3):52–61, may–jun 1997.
- [19] Wei-Chao CHEN, Jean-Yves BOUGUET, Michael H. CHU et Radek GRZESZCZUK : Light field mapping : efficient representation and hardware rendering of surface light fields. *ACM Transactions on Graphics*, 21(3):447–456, 2002.
- [20] Stephen CHENNEY : Flow tiles. In *SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 233–242, New York, NY, USA, 2004. ACM Press. ISBN 3-905673-14-2.
- [21] Nuttapon CHENTANEZ, Tolga G. GOKTEKIN, Bryan E. FELDMAN et James F. O'BRIEN : Simultaneous coupling of fluids and deformable bodies. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 83–89, 2006.

- [22] Norishige CHIBA, Kazunobu MURAOKA, Hiromichi IAKAHASHI et Mamoru MIURA : Two-dimensional visual simulation of flames, smoke and the spread of fire. *Journal of Visualization and Computer Animation*, 5:37–53, 1994.
- [23] Norishige CHIBA, Shinji SANAKANISHI, Kenichiro YOKOYAMA, Isao OTAWARA, Kazunobu MURAOKA et Nobuji SAITO : Visual simulation of water currents using a particle-based behavioural model. *Journal of Visualization and Computer Animation*, 6(3):155–171, 1995.
- [24] Alexandre Joel CHORIN : Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104):745–762, octobre 1968.
- [25] Philippe G. CIARLET : *Basic error estimates for elliptic problems*, in *Handbook of Numerical Analysis*, volume 2, pages 17–351. North-Holland, Amsterdam, 1991.
- [26] Simon CLAVET, Philippe BEAUDOIN et Pierre POULIN : Particle-based Viscoelastic Fluid Simulation. In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : *ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, pages 219–228, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [27] Mathieu DESBRUN et Marie-Paule CANI : Animating soft substances with implicit surfaces. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 287–290, New York, NY, USA, 1995. ACM Press. ISBN 0-89791-701-4.
- [28] Mathieu DESBRUN et Marie-Paule CANI : Smoothed particles : A new paradigm for animating highly deformable bodies. In R. BOULIC et G. HEGRON, éditeurs : *Computer Animation and Simulation '96*, Eurographics, pages 61–76. Springer-Verlag Wien New York, 1996. Proc. Eurographics Workshop, Poitiers, France, August 31–September 18, 1996.
- [29] Mathieu DESBRUN et Marie-Paule CANI : Active implicit surface for animation. In *Graphics Interface*, pages 143–150, June 1998.
- [30] Paul J. DIEFENBACH et Norman I. BADLER : Multi-pass pipeline rendering : Realism for dynamic environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 59–70. ACM, 1997.
- [31] Yoshinori DOBASHI, Tomoyuki NISHITA et Tsuyoshi OKITA : Animation of clouds using cellular automaton. In *Computer Graphics and Imaging '99*, 1999.
- [32] Todd F. DUPONT et Yingjie LIU : Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *Journal of Computational Physics*, 190(1):311–324, septembre 2003.
- [33] David S. EBERT, Wayne E. CARLSON et Richard E. PARENT : Solid spaces and inverse particle systems for controlling the animation of gases and fluids. *The Visual Computer*, 10(4):179–190, 1994.

- [34] Sharif ELCOTT, Yiyong TONG, Eva KANSO, Peter SCHRÖDER et Mathieu DESBRUN : Stable, circulation-preserving, simplicial fluids. *ACM Transaction on Graphics*, 26(1):4, January 2007. ISSN 0730-0301.
- [35] Klaus ENGEL, Martin KRAUS et Thomas ERTL : High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. *In HWWS '01 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-407-X.
- [36] Douglas ENRIGHT, Stephen MARSCHNER et Ronald FEDKIW : Animation and rendering of complex water surfaces. *In SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 736–744, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-521-1.
- [37] Zhe FAN, Ye ZHAO, Arie KAUFMAN et Ying HE : Adapted Unstructured LBM for Flow Simulation on Curved Surfaces. *In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, pages 245–254, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [38] Raanan FATTAL et Dani LISCHINSKI : Target-driven smoke animation. *ACM Transactions on Graphics*, 23(3):441–448, 2004. ISSN 0730-0301.
- [39] Ronald FEDKIW, Jos STAM et Henrik Wann JENSEN : Visual simulation of smoke. *In SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 15–22, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-374-X.
- [40] Ronald P. FEDKIW, Tariq ASLAM, Barry MERRIMAN et Stanley OSHER : A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152(2):457–492, Jul 1999.
- [41] Bryan E. FELDMAN, James F. O'BRIEN et Okan ARIKAN : Animating suspended particle explosions. *ACM Transactions on Graphics*, 22(3):708–715, 2003. ISSN 0730-0301.
- [42] Bryan E. FELDMAN, James F. O'BRIEN et Bryan M. KLINGNER : Animating gases with hybrid meshes. *ACM Transactions on Graphics*, 24(3):904–909, 2005. ISSN 0730-0301.
- [43] Bryan E. FELDMAN, James F. O'BRIEN, Bryan M. KLINGNER et Tolga G. GOKTEKIN : Fluids in Deforming Meshes. *In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, pages 255–260, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [44] Nick FOSTER et Ronald FEDKIW : Practical animation of liquids. *In SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-374-X.

- [45] Nick FOSTER et Dimitri METAXAS : Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, septembre 1996.
- [46] Nick FOSTER et Dimitris METAXAS : Controlling fluid animation. *In Proceedings of Computer Graphics International 1997*, pages 178–188, Los Alamitos, CA, USA, juin 1997. IEEE Computer Society. ISBN 0-8186-7825-9.
- [47] Nick FOSTER et Dimitris METAXAS : Modeling the motion of a hot, turbulent gas. *In SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 181–188, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. ISBN 0-89791-896-7.
- [48] Alain FOURNIER et William T. REEVES : A simple model of ocean waves. *In SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 75–84, New York, NY, USA, 1986. ACM Press. ISBN 0-89791-196-2.
- [49] Patrick FOURNIER, Arash HABIBI et Pierre POULIN : Simulating the flow of liquid droplets. *In Graphics Interface*, pages 133–142, June 1998.
- [50] Manuel Noronha GAMITO, Pedro Faria LOPES et Mário Rui GOMES : Two-dimensional simulation of gaseous phenomena using vortex particles. *In Dimitri TERZOPOULOS et Daniel THALMANN, éditeurs : Computer Animation and Simulation '95*, Eurographics, pages 3–15. Springer-Verlag Wien New York, 1995. Proceedings of the Eurographics Workshop in Maastricht, Netherlands, September 2–3, 1995.
- [51] Olivier GÉNEVAUX, Arash HABIBI et Jean-Michel DISCHLER : Simulating fluid-solid interaction. *In Graphics Interface*, pages 31–38. CIPS, Canadian Human-Computer Communication Society, A K Peters, June 2003. ISBN 1-56881-207-8, ISSN 0713-5424.
- [52] Olivier GÉNEVAUX, Frédéric LARUE et Jean-Michel DISCHLER : Interactive refraction on complex static geometry using spherical harmonics. *In SI3D '06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 145–152, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-295-X.
- [53] Robert A. GINGOLD et Joe J. MONAGHAN : Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, novembre 1977.
- [54] Tolga G. GOKTEKIN, Adam W. BARGTEIL et James F. O'BRIEN : A method for animating viscoelastic fluids. *ACM Transactions on Graphics*, 23(3):463–468, 2004. ISSN 0730-0301.
- [55] Nolan GOODNIGHT, Cliff WOOLLEY, Gregory LEWIN, David LUEBKE et Greg HUMPHREYS : A multigrid solver for boundary value problems using programmable graphics hardware. *In M. DOGGETT, W. HEIDRICH, W. MARK et A. SCHILLING, éditeurs : SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 102–111, San Diego, California, 2003. Eurographics Association.

- [56] Ned GREENE : Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11), novembre 1986.
- [57] Shannon T. GREENWOOD et Donald H. HOUSE : Better with Bubbles : Enhancing the Visual Realism of Simulated Fluid. In Dinesh K. PAI et Ronan BOULIC, éditeurs : *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 287–296, Grenoble, France, 2004. Eurographics Association. ISBN 3-905673-14-2.
- [58] Eran GUENDELMAN, Andrew SELLE, Frank LOSASSO et Ronald FEDKIW : Coupling water and smoke to thin deformable and rigid shells. *ACM Transactions on Graphics*, 24(3):973–981, 2005. ISSN 0730-0301.
- [59] Robert D. GUENTHER : *Modern Optics*. John Wiley & Sons, 1990.
- [60] Stéphane GUY et Cyril SOLER : Graphics gems revisited. *ACM Transactions on Graphics*, 23(3):231–238, 2004.
- [61] Arash HABIBI, Annie LUCIANI et Alexis VAPILLON : A physically-based model for the simulation of reactive turbulent objects. In *Winter School of Computer Graphics 1996*, pages 113–122, février 1996. held at University of West Bohemia, Plzen, Czech Republic, 12-16 February 1996.
- [62] Sunil HADAP et Nadia MAGNENAT-THALMANN : Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.
- [63] Ziyad S. HAKURA et John M. SNYDER : Realistic reflections and refractions on graphics hardware with hybrid rendering and layered environment maps. In *12th Eurographics Workshop on Rendering*, pages 289–300. Eurographics, 2001.
- [64] Francis Harvey HARLOW et James Eddie WELCH : Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.
- [65] Mark J. HARRIS, William V. Baxter III, Thorsten SCHEUERMANN et Anselmo LASTRA : Simulation of cloud dynamics on graphics hardware. In M. DOGGETT, W. HEIDRICH, W. MARK et A. SCHILLING, éditeurs : *SIGGRAPH / Eurographics Workshop on Graphics Hardware*, pages 92–101. Eurographics Association, 2003.
- [66] David R. HAUMANN et Richard E. PARENT : The behavioral test-bed : obtaining complex behavior from simple rules. *The Visual Computer*, 4(6):332–347, décembre 1988.
- [67] Frédéric HECHT et Carlos PARÉS : Nsp1b3 : un logiciel pour résoudre les équations de navier stokes incompressible 3d. rapport de recherche rr-1449. Rapport technique, INRIA, jun 1991.
- [68] Paul S. HECKBERT et Pat HANRAHAN : Beam tracing polygonal objects. *Computer Graphics*, 18(3):119–127, 1984.

- [69] Wolfgang HEIDRICH, Hendrik LENSCH, Michael F. COHEN et Hans-Peter SEIDEL : Light field techniques for reflexions and refractions. *In Rendering Techniques '99*, pages 187–196. Eurographics, 1999.
- [70] Nathan HOLMBERG et Burkhard C. WÜNSCHE : Efficient modeling and rendering of turbulent water over natural terrain. *In GRAPHITE '04 : Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 15–22, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-883-0.
- [71] Thorsten HOLTKÄMPER : Real-time gaseous phenomena : a phenomenological approach to interactive smoke and steam. *In AFRIGRAPH '03 : Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa*, pages 25–30, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-643-9.
- [72] Jeong-Mo HONG et Chang-Hun KIM : Animation of bubbles in liquid. *Computer Graphics Forum*, 22(3):253–253, 2003.
- [73] Jeong-mo HONG et Chang-hun KIM : Controlling fluid animation with geometric potential. *Computer Animation and Virtual Worlds*, 15:147–157, 2004.
- [74] Jeong-Mo HONG et Chang-Hun KIM : Discontinuous fluids. *ACM Transactions on Graphics*, 24(3):915–920, 2005. ISSN 0730-0301.
- [75] Insung IHM, Byungkwon KANG et Deukhyun CHA : Animation of Reactive Gaseous Fluids through Chemical Kinetics. *In Dinesh K. PAI et Ronan BOULIC, éditeurs : Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 203–212, Grenoble, France, 2004. Eurographics Association. ISBN 3-905673-14-2.
- [76] James T. KAJIYA et Brian P. von HERZEN : Ray tracing volume densities. *In SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 165–174, New York, NY, USA, 1984. ACM Press. ISBN 0-89791-138-5.
- [77] Kazufumi KANEDA, Shinya IKEDA et Hideo YAMASHITA : Animation of water droplets moving down a surface. *The Journal of Visualization and Computer Animation*, 10(1):15–26, 1999.
- [78] Michael KASS et Gavin MILLER : Rapid, stable fluid dynamics for computer graphics. *In SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 49–57, New York, NY, USA, 1990. ACM Press. ISBN 0-201-50933-4.
- [79] Douglas S. KAY et Donald P. GREENBERG : Transparency for computer synthesized images. *Computer Graphics*, 13(2):158–164, 1979.
- [80] Michael KECKEISEN, Stefan KIMMERLE, Bernhard THOMASZEWSKI et Markus WACKER : Modelling effects of wind fields in cloth animations. *Journal of WSCG*, Vol. 12(1–3):205–212, 2004.

- [81] Richard KEISER, Bart ADAMS, Dominique GASSER, Paolo BAZZI, Philip DUTRÉ et Markus GROSS : A Unified Lagrangian Approach to Solid-Fluid Animation. *In Symposium on Point-Based Graphics*, pages 125–133. Eurographics Association, 2005.
- [82] ByungMoon KIM, Yingjie LIU, Ignacio LLAMAS et Jarek ROSSIGNAC : Flowfixer : Using bfec for fluid simulation. *In Pierre POULIN et Eric GALIN, éditeurs : Proceedings of the Eurographics Workshop on Natural Phenomena*, pages 51–56, Dublin, Ireland, 2005. Eurographics Association. ISBN 3-905673-29-0.
- [83] Janghee KIM, Deukhyun CHA, Byungjoon CHANG, Bonki KOO et Insung IHM : Practical animation of turbulent splashing water. *In SCA '06 : Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 335–344, 2006.
- [84] Peter KIPFER, Mark SEGAL et Rüdiger WESTERMANN : UberFlow : A GPU-Based Particle Engine. *In Mark HARRIS et David LUEBKE, éditeurs : Graphics Hardware*, pages 115–122, Grenoble, France, 2004. Eurographics Association. ISBN 3-905673-15-0.
- [85] Thomas KLEIN, Mike EISSELE, Daniel WEISKOPF et Thomas ERTL : Simulation, modelling and rendering of incompressible fluids in real time. *In Proceedings of the Vision, Modeling, and Visualization vConference 2003*, pages 365–373, nov 2003.
- [86] Bryan M. KLINGNER, Bryan E. FELDMAN, Nuttapong CHENTANEZ et James F. O'BRIEN : Fluid animation with dynamic meshes. *In SIGGRAPH '06 : ACM SIGGRAPH 2006 Papers*, pages 820–825, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-364-6.
- [87] Jens KRÜGER et Rüdiger WESTERMANN : Linear algebra operators for gpu implementation of numerical algorithms. *ACM Transactions on Graphics*, 22(3):908–916, 2003. ISSN 0730-0301.
- [88] A. KUNIMATSU, Y. WATANABE, H. FUJII, T. SAITO, K. HIWADA, T. TAKAHASHI et H. UEKI : Fast simulation and rendering techniques for fluid objects. *Computer Graphics Forum*, 20(3):57–67, 2001. ISSN 1067-7055.
- [89] Patrick LE TALLEC et Jean MOURO : Structures en grands déplacements couplées à des fluides en mouvement. rapport de recherche rr-2961. Rapport technique, INRIA, 1996.
- [90] Shengying LI et Klaus MUELLER : Accelerated, high-quality refraction computations for volume graphics. *In International Workshop on Volume Graphics 2005*, pages 73–81, 2005.
- [91] Wei LI, Xiaoming WEI et Arie KAUFMAN : Implementing lattice boltzmann computation on graphics hardware. *The Visual Computer*, 19(7–8):444–456, 2003.
- [92] Li LING, Murali DAMODARAN et Robert K. L. GAY : Aerodynamic force models for animating cloth motion in air flow. *The Visual Computer*, 12(2):84–104, feb 1996.

- [93] William E. LORENSEN et Harvey E. CLINE : Marching cubes : A high resolution 3d surface construction algorithm. *In SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press. ISBN 0-89791-227-6.
- [94] Frank LOSASSO, Frédéric GIBOU et Ron FEDKIW : Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics*, 23(3):457–462, 2004. ISSN 0730-0301.
- [95] Annie LUCIANI, Arash HABIBI, Alexis VAPILLON et Yves DUROC : A physical model of turbulent fluids. *In Dimitri TERZOLPOULOS et Daniel THALMANN, éditeurs : Computer Animation and Simulation '95*, Eurographics, pages 16–29. Springer-Verlag Wien New York, 1995. Proc. Eurographics Workshop, Maastricht, Netherlands, September 2–3, 1995.
- [96] Leon B. LUCY : A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, décembre 1977.
- [97] Thomas Murray MACROBERT : *Spherical harmonics : an elementary treatise on harmonic functions with applications*. Pergamon Press, 1967.
- [98] Hugh MALLINDER : The modelling of large waterfalls using string texture. *Journal of Visualization and Computer Animation*, 6:3–10, 1995.
- [99] Antoine MCNAMARA, Adrien TREUILLE, Zoran POPOVIĆ et Jos STAM : Fluid control using the adjoint method. *ACM Transactions on Graphics*, 23(3):449–456, 2004. ISSN 0730-0301.
- [100] Guy R MCNAMARA et Gianluigi ZANETTI : Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20):2332–2335, Nov 1988.
- [101] Viorel MIHALEF, Dimitris METAXAS et Mark SUSSMAN : Animation and Control of Breaking Waves. *In Dinesh K. PAI et Ronan BOULIC, éditeurs : Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 315–324, Grenoble, France, 2004. Eurographics Association. ISBN 3-905673-14-2.
- [102] Viorel MIHALEF, Betul UNLUSU, Dimitris METAXAS, Mark SUSSMAN et Youssuf HUSSAINI : Physics based boiling simulation. *In SCA '06 : Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 317–324, 2006.
- [103] Gavin MILLER et Andrew PEARCE : Globular dynamics : A connected particle system for animating viscous fluids. *Computers & Graphics*, 13(3):305–309, 1989.
- [104] Ryo MIYAZAKI, Satoru YOSHIDA, Yoshinori DOBASHI et Tomoyuki NISHITA : A method for modeling clouds based on atmospheric fluid dynamics. *In Proceedings of the Ninth Pacific Conference on Computer Graphics and Applications*, pages 363–372, 2001.

- [105] Matthias MÜLLER, David CHARYPAR et Markus GROSS : Particle-Based Fluid Simulation for Interactive Applications. In D. BREEN et M. LIN, éditeurs : *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 154–159, San Diego, California, 2003. Eurographics Association.
- [106] Matthias MÜLLER, Simon SCHIRM, Matthias TESCHNER, Bruno HEIDELBERGER et Markus GROSS : Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds*, 15(3–4):159–171, 2004.
- [107] Matthias MÜLLER, Barbara SOLENTHALER, Richard KEISER et Markus GROSS : Particle-Based Fluid-Fluid Interaction. In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : *ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, pages 237–244, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [108] David MOULD et Yee-Hong YANG : Modeling water for computer graphics. *Computers & Graphics*, 21(6):801–814, nov–dec 1997.
- [109] Alan MURTA et James MILLER : Modelling and rendering liquids in motion. In *WSCG'99 Conference Proceedings*, 1999.
- [110] Fabrice NEYRET : Qualitative simulation of convective cloud formation and evolution. In D. THALMANN et M. van de PANNE, éditeurs : *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '97*, Eurographics, pages 113–124. Springer-Verlag Wien New York, 1997.
- [111] Fabrice NEYRET : Advected Textures. In D. BREEN et M. LIN, éditeurs : *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 147–153, San Diego, California, 2003. Eurographics Association.
- [112] Fabrice NEYRET et Nathalie PRAIZELIN : Phenomenological simulation of brooks. In Nadia MAGNENAT-THALMANN et Daniel THALMANN, éditeurs : *Computer Animation and Simulation '01*, Eurographics, pages 53–64. Springer-Verlag Wien New York, 2001. Proc. Eurographics Workshop, Manchester UK, September 2–3, 2001.
- [113] Duc Quang NGUYEN, Ronald FEDKIW et Henrik Wann JENSEN : Physically based modeling and animation of fire. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 721–728, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-521-1.
- [114] Daniel NIXON et Richard LOBB : A fluid-based soft-object model. *IEEE Computer Graphics and Applications*, 22(4):68–75, jul–aug 2002.
- [115] James F. O'BRIEN et Jessica K. HODGINS : Dynamic simulation of splashing fluids. In *Proceedings of Computer Animation '95*, pages 195–205, avril 1995.
- [116] James F. O'BRIEN et Jessica K. HODGINS : Graphical modeling and animation of brittle fracture. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 137–146, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5.

- [117] Eisaku OHBUCHI : A real-time refraction renderer for volume objects using a polygon-rendering scheme. *In Computer Graphics International*, pages 190–195, 2003.
- [118] Sang Il PARK et Myoung Jun KIM : Vortex Fluid for Gaseous Phenomena. *In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : ACM SIGGRAPH /Eurographics Symposium on Computer Animation*, pages 261–270, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [119] Darwyn R. PEACHEY : Modeling waves and surf. *In SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 65–74, New York, NY, USA, 1986. ACM Press. ISBN 0-89791-196-2.
- [120] Frank PERBET et Marie-Paule CANI : Animating prairies in real-time. *In SI3D '01 : Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 103–110, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-292-1.
- [121] Richard C. PETERSON, Peter K. JIMACK et Mark A. KELMANSON : The solution of two-dimensional free-surface problems using automatic mesh generation. *International Journal for Numerical Methods in Fluids*, 31 (6):937–960, 1999.
- [122] Frédéric PIGHIN, Jonathan M. COHEN et Maurya SHAH : Modeling and Editing Flows Using Advected Radial Basis Functions. *In Dinesh K. PAI et Ronan BOULIC, éditeurs : Eurographics / SIGGRAPH Symposium on Computer Animation*, pages 223–232, Grenoble, France, 2004. Eurographics Association. ISBN 3-905673-14-2.
- [123] Simon PREMOŽE, Tolga TASDIZEN, James BIGLER, Aaron LEFOHN et Ross T. WHITAKER : Particle-based simulation of fluids. *Computer Graphics Forum*, 22(3):401–401, 2003.
- [124] William H. PRESS, Saul A. TEUKOLSKY, William T. VETTERLING et Brian P. FLANNERY : *Numerical Recipes in C, The art of Scientific Computing, Second Edition*. Cambridge University Press, 1992.
- [125] Timothy J. PURCELL, Ian BUCK, William R. MARK et Pat HANRAHAN : Ray tracing on programmable graphics hardware. *In SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 703–712, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-521-1.
- [126] Nick RASMUSSEN, Doug ENRIGHT, Duc NGUYEN, Sebastian MARINO, Nigel SUMNER, Willi GEIGER, Samir HOON et Ron FEDKIW : Directable photorealistic liquids. *In SCA '04 : Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 193–202, New York, NY, USA, 2004. ACM Press. ISBN 3-905673-14-2.
- [127] Nick RASMUSSEN, Duc Quang NGUYEN, Willi GEIGER et Ronald FEDKIW : Smoke simulation for large scale phenomena. *ACM Transactions on Graphics*, 22(3):703–707, 2003. ISSN 0730-0301.

- [128] John N. REDDY : *An introduction to the finite Element Method, Second Edition*. McGraw-Hill, 1993.
- [129] William T. REEVES : Particle systems – a technique for modeling a class of fuzzy objects. *ACM Transactions On Graphics*, 2(2):91–108, 1983. ISSN 0730-0301.
- [130] Marcin J. RUDOLF et Jacek RACZKOWSKI : Modeling the motion of dense smoke in the wind field. *Computer Graphics Forum*, 19(3), August 2000. ISSN 1067-7055.
- [131] Yousef SAAD : *Iterative Methods for Sparse Linear Systems, Second Edition*. SIAM, 2003.
- [132] Georgios SAKAS : Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer*, 9(4):200–212, 1993. ISSN 0178-2789.
- [133] Gernot SCHAUFLER et Henrik Wann JENSEN : Ray tracing point sampled geometry. In *Rendering Techniques 2000*, pages 319–328, 2000.
- [134] Carlos E. SCHEIDEGGER, João L. D. COMBA et Rudnei D. da CUNHA : Practical CFD Simulations on Programmable Graphics Hardware using SMAC. *Computer Graphics Forum*, 24(4):715–728, 2005.
- [135] Joshua SCHPOK, William DWYER et David S. EBERT : Modeling and animating gases with simulation features. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 97–105, New York, NY, USA, 2005. ACM Press. ISBN 1-7695-2270-X.
- [136] Andrew SELLE, Nick RASMUSSEN et Ronald FEDKIW : A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.*, 24(3):910–914, 2005. ISSN 0730-0301.
- [137] James A. SETHIAN : *Level Set Methods and Fast Marching Methods — Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.
- [138] Maurya SHAH, Jonathan M. COHEN, Sanjit PATEL, Penne LEE et Frédéric PIGHIN : Extended Galilean Invariance for Adaptive Fluid Simulation. In Dinesh K. PAI et Ronan BOULIC, éditeurs : *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 213–221, Grenoble, France, 2004. Eurographics Association. ISBN 3-905673-14-2.
- [139] Lin SHI et Yizhou YU : Inviscid and incompressible fluid simulation on triangle meshes. *Computer Animation and Virtual Worlds*, 15(3–4):173–181, 2004.
- [140] Lin SHI et Yizhou YU : Controllable smoke animation with guiding objects. *ACM Transactions on Graphics*, 24(1):140–164, 2005. ISSN 0730-0301.

- [141] Lin SHI et Yizhou YU : Taming Liquids for Rapidly Changing Targets. In Demetri TERZOPOULOS et Victor ZORDAN, éditeurs : *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 229–236, Los Angeles, California, 2005. Eurographics Association. ISBN 1-59593-198-8.
- [142] Mikio SHINYA et Alain FOURNIER : Stochastic motion-motion under the influence of wind. *Computer Graphics Forum*, 11(3):119–128, 1992.
- [143] Karl SIMS : Particle animation and rendering using data parallel computation. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 405–413, New York, NY, USA, 1990. ACM Press. ISBN 0-201-50933-4.
- [144] Karl SIMS : Choreographed image flow. *The Journal of Visualization and Computer Animation*, 3:31–43, 1992.
- [145] Peter-Pike SLOAN, Jesse HALL, John HART et John SNYDER : Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics*, 22(3):382–391, 2003.
- [146] Peter-Pike SLOAN, Jan KAUTZ et John SNYDER : Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, 21(3):527–536, 2002.
- [147] Oh-Young SONG, Hyuncheol SHIN et Hyeong-Seok KO : Stable but nondissipative water. *ACM Transactions on Graphics*, 24(1):81–97, 2005. ISSN 0730-0301.
- [148] Jos STAM : Stochastic dynamics : Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum*, 16(3):159–164, August 1997. ISSN 1067-7055.
- [149] Jos STAM : Stable fluids. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5.
- [150] Jos STAM : Interacting with smoke and fire in real time. *Communications of the ACM*, 43(7):76–83, 2000. ISSN 0001-0782.
- [151] Jos STAM : A simple fluid solver based on the fft. *Journal of Graphics Tools*, 6(2):43–52, 2001.
- [152] Jos STAM : Flows on surfaces of arbitrary topology. *ACM Transactions on Graphics*, 22(3):724–731, 2003. ISSN 0730-0301.
- [153] Jos STAM : Real-time fluid dynamics for games. In *Proceedings of the Game Developer Conference*, mars 2003.
- [154] Jos STAM et Eugene FIUME : Turbulent wind fields for gaseous phenomena. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 369–376, New York, NY, USA, 1993. ACM Press. ISBN 0-89791-601-8.

- [155] Jos STAM et Eugene FIUME : Depicting fire and other gaseous phenomena using diffusion processes. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 129–136, New York, NY, USA, 1995. ACM Press. ISBN 0-89791-701-4.
- [156] Kevin STEELE, David CLINE, Parris K. EGBERT et Jonathan DINERSTEIN : Modeling and rendering viscous liquids. *Computer Animation and Virtual Worlds*, 15(3-4):183–192, 2004.
- [157] John STEINHOFF et David UNDERHILL : Modification of the euler equations for “vorticity confinement” : Application to the computation of interacting vortex rings. *Physics of Fluids*, 6(8):2738–2744, août 1994.
- [158] Dan STORA, Pierre-Olivier AGLIATI, Marie-Paule CANI, Fabrice NEYRET et Jean-Dominique GASCUEL : Animating lava flows. In *Graphics Interface*, pages 203–210, June 1999.
- [159] Tsunemi TAKAHASHI, Hiroko FUJII, Atsushi KUNIMATSU, Kazuhiro HIWADA, Takahiro SAITO, Ken TANAKA et Heihachi UEKI : Realistic animation of fluid with splash and foam. *Computer Graphics Forum*, 22(3):391–400, 2003.
- [160] Daiki TAKESHITA, Shin OTA, Macjiko TAMURA, Tadahiro FUJIMOTO, Kazunobu MURAOKA et Norishige CHIBA : Particle-based visual simulation of explosive flames. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, pages 482–486, 2003.
- [161] Demetri TERZOPOULOS et Kurt FLEISCHER : Deformable models. *The Visual Computer*, 4(6):306–331, 1988.
- [162] Demetri TERZOPOULOS, John PLATT et Kurt FLEISCHER : Heating and melting deformable models (from goop to glop). In *Graphics Interface '89*, pages 219–226, June 1989.
- [163] Daniel THALMANN : A “lifegame” approach to surface modeling and rendering. *The Visual Computer*, 2(6):384–390, décembre 1986.
- [164] Nils THÜREY, Ulrich RÜDE et Marc STAMMINGER : Animation of open water phenomena with coupled shallow water and free surface simulations. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 157–165, 2006.
- [165] Ruofeng TONG, Kazufumi KANEDA et Hideo YAMASHITA : A volume-preserving approach for modeling and animating water flows generated by metaballs. *The Visual Computer*, 18(8):469–480, décembre 2002.
- [166] David TONNESEN : Modeling liquids and solids using thermal particles. In *Graphics Interface '91*, pages 255–262, June 1991.
- [167] Adrien TREUILLE, Antoine MCNAMARA, Zoran POPOVIĆ et Jos STAM : Keyframe control of smoke simulations. *ACM Transactions on Graphics*, 22(3):716–723, 2003. ISSN 0730-0301.

- [168] Pauline Y. Ts'o et Brian A. BARSKY : Modeling and rendering waves : Wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Transactions on Graphics*, 6(3):191–214, 1987.
- [169] Matthias UNBESCHIEDEN et Andrzej TREMBILSKI : Cloud simulation in virtual environments. In *VRAIS '98 : Proceedings of the Virtual Reality Annual International Symposium*, pages 98–104, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 0-8186-8362-7.
- [170] Huamin WANG, Peter J. MUCHA et Greg TURK : Water drops on surfaces. *ACM Transactions on Graphics*, 24(3):921–929, 2005. ISSN 0730-0301.
- [171] Xiaoming WEI, Wei LI et Arie KAUFMAN : Melting and flowing of viscous volume. In *16th International Conference on Computer Animation and Social Agents*, pages 54–59, mai 2003.
- [172] Xiaoming WEI, Wei LI, Klaus MUELLER et Arie KAUFMAN : The lattice-boltzmann method for simulating gaseous phenomena. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):164–176, mar–apr 2004.
- [173] Xiaoming WEI, Ye ZHAO, Zhe FAN, Wei LI, Feng QIU, Suzanne YOAKUM-STOVER et Arie KAUFMAN : Lattice-based flow field modeling. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):719–729, nov–dec 2004.
- [174] Xiaoming WEI, Ye ZHAO, Zhe FAN, Wei LI, Suzanne YOAKUM-STOVER et Arie KAUFMAN : Blowing in the Wind. In D. BREEN et M. LIN, éditeurs : *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 75–85. Eurographics Association, 2003.
- [175] Henrik WEIMER et Joe WARREN : Subdivision schemes for fluid flow. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 111–120, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5.
- [176] Jakub WEJCHERT et David HAUMANN : Animation aerodynamics. In *SIGGRAPH '91 : Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 19–22, New York, NY, USA, 1991. ACM Press. ISBN 0-89791-436-8.
- [177] James Eddie WELCH, Francis Harvey HARLOW, John P. SHANNON et Bart J. DALY : The mac method. a computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces. Rapport technique LA-3425, Los Alamos Scientific Laboratory, nov 1965.
- [178] Stephen H. WESTIN, James R. ARVO et Kenneth E. TORRANCE : Predicting reflectance functions from complex surfaces. *Computer Graphics*, 26(2):255–264, 1992.
- [179] Turner WHITTED : An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, juin 1980.

- [180] Patrick WITTING : Computational fluid dynamics in a traditional animation environment. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 129–136, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5.
- [181] Enhua WU, Youquan LIU et Xuehui LIU : An improved study of real-time fluid simulation on gpu. *Computer Animation and Virtual Worlds*, 15(3–4):139–146, 2004.
- [182] Chris WYMAN : An approximate image-space approach for interactive refraction. *ACM Transactions on Graphics*, 24(3):1050–1053, 2005.
- [183] Takashi YABE, Feng XIAO et Takayuki UTSUMI : The constrained interpolation profile method for multiphase analysis. *Journal of Computational Physics*, 169(2):556–593, mai 2001.
- [184] Larry YAEGER, Craig UPSON et Robert MYERS : Combining physical and visual simulation - creation of the planet jupiter for the film "2010". In *SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 85–93, New York, NY, USA, 1986. ACM Press. ISBN 0-89791-196-2.
- [185] Gary D. YNGVE, James F. O'BRIEN et Jessica K. HODGINS : Animating explosions. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 29–36, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5.
- [186] Han Young YOON, Seiichi KOSHIZUKA et Yoshiaki OKA : A particle-gridless hybrid method for incompressible flows. *International Journal for Numerical Methods in Fluids*, 30(4):407–424, 1999.
- [187] Satoru YOSHIDA et Tomoyuki NISHITA : Modelling of smoke flow taking obstacles into account. In *Proceedings of Pacific Graphics 2000*, octobre 2000.
- [188] Wen ZHENG, Jun-Hai YONG et Jean-Claude PAUL : Simulation of bubbles. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH / Eurographics symposium on Computer animation*, pages 325–333, 2006.
- [189] Yongning ZHU et Robert BRIDSON : Animating sand as a fluid. *ACM Transactions on Graphics*, 24(3):965–972, 2005. ISSN 0730-0301.
- [190] Douglas E. ZONGKER, Dawn M. WERNER, Brian CURLLESS et David H. SALESIN : Environment matting and compositing. In *Proceedings of ACM SIGGRAPH 99*, pages 205–214. ACM, 1999.
- [191] Matthias ZWICKER, Hanspeter PFISTER, Jeroen van BAAR et Markus GROSS : Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, Jul–Sep 2002.