

ACADEMIE DE MONTPELLIER
UNIVERSITE MONTPELLIER II
- Sciences et Techniques du Languedoc -

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE MONTPELLIER II

Discipline : **Génie Informatique, Automatique et Traitement du signal**

Formation doctorale : **Systèmes Automatiques et Microélectroniques**

Ecole doctorale : **Informatique, Structures, Systèmes**

présentée et soutenue publiquement par

Sébastien DRUON

le 14 Décembre 2004

Titre :

**Modélisation et recalage d'objets et d'environnements
en déformation viscoplastique : Aide au geste pour les
métiers matière**

JURY

Mme Marie-José Aldon	Chargé de recherche CNRS	Examineur
Mme Marie-Paule Cani	Professeur	Rapporteur
M. André Crosnier	Professeur	Directeur de thèse
M. Etienne Dombre	Directeur de recherche CNRS	Examineur
M. Abderrahmane Kheddar	Professeur	Rapporteur
M. Franck Nicoud	Professeur	Examineur

A Stéphanie et toute ma famille,

Remerciements

Notations

a	coefficient de l'équation (3.27)
\mathbf{a}	amortissement éq.(2.27)
a_i	coefficient de l'équation (3.27)
$B_{i,j}$	polynôme de Bernstein bivariant éq.(2.24)
C	concentration en marqueur
c_0	paramètre de l'équation (3.22)
c_1	paramètre de l'équation (3.22)
c_2	paramètre de l'équation (3.22)
D	matrice relative à l'amortissement éq.(2.32)
\mathcal{D}	coefficient de diffusion
$\underline{\underline{D}}$	tenseur des taux de déformation
\underline{d}	dimension caractéristique de l'écoulement
\vec{d}	vecteur direction du ressort éq.(2.27)
$D_{i,j,k}$	divergence du fluide dans le voxel (i,j,k)
E	excès de matière
\mathcal{E}	espace affine
F	force exercée sur le fluide
\mathcal{F}	règle locale de transition d'un automate cellulaire
\vec{g}	force de pesanteur
h	hauteur de liquide
i	coordonnée discrète selon (Ox)
j	coordonnée discrète selon (Oy)
K	constante de raideur du ressort éq.(2.27)
k	coordonnée discrète selon (Oz)
L	dimension de l'espace de travail
\mathcal{L}	élongation du ressort éq.(2.27)
\mathcal{L}_0	élongation initiale du ressort éq.(2.27)
M	matrice relative à la masse éq. (2.32)
m	masse de fluide
N	nombre de voxels d'une arête de l'espace de travail
\mathcal{N}	voisinage d'une cellule d'un automate cellulaire
n	exposant dans l'équation (2.5)
\vec{n}	vecteur normal à la surface

P	pression corrigée
\mathcal{P}	potentiel scalaire de l'équation (3.59)
p	pression
p_0	correction appliquée à la pression éq.(3.4)
$\tilde{P}_{i,j,k}$	points de contrôle d'une courbe 3D
P_{outil}	pression pour l'interaction outil/matière éq.(3.69)
Q	quantité de matière
\mathcal{Q}	ensemble des états que peut prendre une cellule d'un automate cellulaire
q	pourcentage de remplissage d'un voxel
R	taille du voisinage pour un automate cellulaire moyenné
r	vecteur position
Re	nombre de Reynolds
S	silhouette 2D
t	temps
U	composante du vecteur vitesse
\vec{u}	représentant du champ de vitesse
\vec{X}	vecteur coordonnées d'un point
x	grandeur diffusée éq.(3.18)
Z	altitude
z	altitude du point

Lettres grecques

α	viscosité viscoplastique éq.(2.1)
β	viscosité viscoplastique éq.(2.4)
Γ	frontière du domaine Ω
$\underline{\underline{\delta}}$	tenseur identité
δh	pas de discrétisation spatial
δt	pas de discrétisation temporel
ε	déformation
$\dot{\varepsilon}$	vitesse de déformation
λ	constante relative à l'équation (3.12)
μ	viscosité dynamique
ν	viscosité cinématique
ν_D	coefficient de diffusion d'un marqueur
Π	opérateur de projection
ρ	masse volumique
$\underline{\underline{\sigma}}$	tenseur des contraintes
τ	contrainte
Φ	flux
Ψ	objet numérisé
Ω	domaine de l'espace

Indices

c	seuil
$facette$	sur les facettes du voxel éq.(3.34).
ffd	relatif aux déformations "free form"
i, j, k	relatif au voxel (i,j,k)
S	en surface
x	composante selon (Ox)
y	composante selon (Oy)
z	composante selon (Oz)

Exposants à droite

s	en surface
t	au temps t

Exposants à gauche

x	relatif à la composante selon (Ox)
y	relatif à la composante selon (Oy)
z	relatif à la composante selon (Oz)
t	transposition

Table des matières

Introduction	1
1 Aide au geste pour le modelage	3
1.1 Introduction	3
1.2 Les systèmes d'aide au geste	3
1.2.1 Les simulateurs	4
1.2.2 Les systèmes de supervision	6
1.3 Architecture des systèmes pour l'aide au geste	7
1.4 Aide au geste pour le modelage	9
1.5 Description du système étudié	11
2 Modélisation des déformations	13
2.1 Introduction	13
2.2 Physique de la déformation plastique	13
2.2.1 Introduction et définitions	13
2.2.2 Approche rhéologique	14
2.2.3 Approche par la mécanique des fluides	18
2.2.3.1 Les équations de Navier-Stokes	18
2.2.3.2 L'équation de conservation de la masse	19
2.2.3.3 L'équation de la quantité de mouvement	19
2.2.3.4 Les conditions aux limites	21
2.2.3.5 Résolution des équations de Navier-Stokes	22
2.2.4 Conclusion	23
2.3 Les modèles d'objets en déformation	24
2.3.1 Introduction	24
2.3.2 Les modèles géométriques	24
2.3.2.1 Splines et surfaces paramétrables	24
2.3.2.2 Free Form Deformations	27
2.3.2.3 Modélisation par surface implicite	28
2.3.2.4 Conclusion sur les modèles géométriques	30
2.3.3 Les modèles physiques	30
2.3.3.1 Les Systèmes Masses-Ressorts	30
2.3.3.2 Les Modèles à Particules	31
2.3.3.3 Les modèles éléments finis	33
2.3.4 La modélisation des écoulements	35
2.3.4.1 Le modèle Lattice Gaz	35

2.3.4.2	Le modèle "Height Field"	37
2.3.4.3	Le travail de Foster et Metaxas	39
2.4	Conclusion	41
3	Approches volumiques de la déformation du modèle matière	43
3.1	Introduction	43
3.2	Hypothèse des écoulements rampants	44
3.3	Modèles basés sur la conservation de la masse	48
3.3.1	Introduction	48
3.3.2	Approche volumique par Automates Cellulaires	48
3.3.2.1	Le modèle matière	48
3.3.2.2	Discrétisation de l'équation de comportement	49
3.3.2.3	Modélisation de diffusion par automates cellulaires	52
3.3.2.4	Implémentation du modèle	57
3.3.2.5	Interaction Outil-Matière	60
3.3.2.6	Résultats et performances	61
3.3.2.7	Implémentation parallèle	66
3.3.2.8	Conclusions sur ce premier modèle	66
3.3.3	Une approche surfacique de la déformation	67
3.3.3.1	Introduction	67
3.3.3.2	Approche surfacique de la modélisation	67
3.3.3.3	Représentation voxel du corps modélisé	69
3.3.3.4	Choix du champ de vitesse et influence de l'outil	72
3.3.3.5	Résultats et conclusion	73
3.3.4	Conclusion sur ces deux premiers modèles	75
3.4	Modèle complet de déformation viscoplastique	76
3.4.1	Introduction	76
3.4.2	Modélisation du comportement viscoplastique	76
3.4.3	Les conditions aux limites	77
3.4.4	Résolution numérique des équations de Poisson	78
3.4.5	Résultats et Conclusion	80
3.5	Bilan de la modélisation	82
4	Vers un recalage basé voxel du modèle matière	83
4.1	Introduction	83
4.2	Recalage de deux grilles voxel	84
4.2.1	Le mécanisme de recalage	84
4.2.2	Les difficultés d'un recalage de deux grilles voxel	85
4.3	Etude de cas : Capteur shape from silhouette	86
4.3.1	Principe du Shape from Silhouette	86
4.3.2	Dispositif expérimental et implémentation	90
4.3.3	Performances du système d'acquisition	94
4.3.4	Extraction des régions d'intérêt	96
4.4	Conclusion	100
5	Conclusion	101

Bibliographie	104
A Rhéologie de la plasticine	111
B Algorithme des Marching Cubes	113

Table des figures

1.1	Simulateur de chirurgie hépatique par laparoscopie (Epidaure, INRIA) . . .	5
1.2	Apart : Reproduction de sculpture assistée par ordinateur	6
1.3	Approche geste et approche matière	8
1.4	Schéma de principe d'un simulateur virtuel pour le modelage	9
1.5	Schéma de principe d'un système basé sur une approche mixte	10
1.6	Schéma de principe	11
2.1	Les éléments de base de la rhéologie	15
2.2	Rhéogramme d'un corps de Bingham	16
2.3	Le corps de Bingham : système Patin - Amortisseur	16
2.4	Rhéogramme d'un corps de Casson	17
2.5	Rhéogramme d'un liquide newtonien	19
2.6	Cas de la surface libre	22
2.7	Déplacement d'un point de contrôle et effet sur une spline	25
2.8	Construction d'une courbe de Bézier à partir de trois points de contrôle . .	25
2.9	Patch de Bézier et ses points de contrôle	26
2.10	Déformation de l'espace illustrée sur un modèle de girafe ([Deb00])	27
2.11	Description explicite et implicite d'une forme géométrique	29
2.12	Exemple de Modèle Masses-Ressorts en 2D	30
2.13	Intensité de la force de Lennard-Jones en fonction de la distance	32
2.14	Le modèle HPP, basé sur une lattice à symétrie carrée	35
2.15	Le modèle FHP, basé sur une lattice à symétrie hexagonale	36
2.16	Le fluide, limité en haut par la surface et en bas par le fond	37
2.17	Un exemple d'espace de calcul voxelisé	39
2.18	Arrangement des variables pour une cellule typique ([FM96])	40
2.19	Résultats de simulation dans le cas 2D	41
3.1	Histogramme de répartition des vitesses durant un modelage	47
3.2	Discrétisation de l'espace de travail	50
3.3	Discrétisation des valeurs sur le voxel	51
3.4	Voisinage centré sur la cellule cible (cas 2D)	58
3.5	Le voisinage de Margolus (cas 2D)	59
3.6	Le voisinage de Margolus (cas 3D)	59
3.7	Le voisinage de Margolus sur deux pas de temps (cas 2D)	59
3.8	Action de l'outil sur le corps (cas 2D)	60
3.9	Décomposition du mouvement en translations élémentaires	61
3.10	Déformations avec un outil sphérique ($N = 100$, $N_{outil} = 32$)	62

3.11	Un manque de cohésion de la matière	62
3.12	Utilisation d'un outil sphérique ($N = 100$, $N_{outil} = 32$)	63
3.13	Compression par outil parallélépipédique ($N = 100$, $N_{outil} = 64$)	64
3.14	Temps de calcul moyen en fonction de la taille de la lattice	65
3.15	Implémentation multithread de l'automate cellulaire	66
3.16	Surface d'un corps voxelisé (Cas 2D)	70
3.17	Les éléments de surface considérés	71
3.18	Directions des normales imposées par un voxel plein (au centre) à son entourage	71
3.19	Quelques exemples de normales en fonction des voisinages	71
3.20	La distance cartésienne pose des problèmes de cohérence de la déformation	73
3.21	Distance cartésienne et géométrie du corps	74
3.22	Champs de vitesse calculé avec les conditions aux bornes	74
3.23	Compression d'un bloc de matière par deux outils (en vert)	81
4.1	Deux maquettes : l'une réelle et l'autre virtuelle	83
4.2	Correction des erreurs du modèle virtuel	84
4.3	Initialisation du modèle matière en début de modelage	85
4.4	Principe de la stéréovision	87
4.5	Principe d'un capteur "Shape from silhouette"	88
4.6	Les cônes de vision obtenus à partir des silhouettes	89
4.7	Plateforme expérimentale	90
4.8	Modèle pinhole	90
4.9	Photo de la mire	91
4.10	Images de l'objet prises durant la numérisation	92
4.11	Espace des couleurs présentes dans la première image	92
4.12	Images après segmentation	93
4.13	Modèle 3D obtenu par le scanner	94
4.14	Le problème des concavités	95
4.15	Découpage de l'espace voxel en tranches horizontales	96
4.16	Recherche des extrema Nord, Sud, Est et Ouest	97
4.17	Insertion d'un nouveau point dans le polygone	97
4.18	Le nouveau polygone	98
4.19	Le polygone final	98
4.20	Coloration finale des voxels de surface	99
4.21	Modèle réel et résultat de la coloration 3D	99
B.1	Configurations types en 2D	113
B.2	Isosurface 2D calculée à partir de la grille de points	114
B.3	Configurations types en 3D	114

Introduction

Nombreux sont les métiers pour lesquels l'informatique est devenu un outil incontournable : la comptabilité, le dessin technique, le secrétariat en sont autant d'exemples qui viennent immédiatement à l'esprit. Cependant, les apports de cet outil ne se restreignent pas à ces applications "de bureau" : grâce aux développements des techniques de réalité virtuelle, l'utilisateur devient le centre du système. Reposant sur le triptique Immersion, Interaction et Imagination, l'objet d'un système virtuel est de donner l'illusion de la réalité à l'utilisateur en trompant ses sens et en réagissant de manière cohérente à ses actions.

L'un des domaines dans lesquels la réalité virtuelle apporte de nouvelles réponses est l'apprentissage, c'est-à-dire l'acquisition d'un geste et d'un savoir-faire spécifique par un opérateur humain. Les premiers à l'avoir compris sont sans conteste les militaires. Depuis longtemps déjà, les simulateurs virtuels interviennent dans le cursus de formation des apprentis pilotes afin qu'ils puissent s'exercer sans risque à effectuer des manœuvres délicates. L'utilisation de simulateurs virtuels s'est ensuite rapidement étendue au domaine civil, comme par exemple dans le cas de l'apprentissage du geste chirurgical. Grâce à cela, un chirurgien novice peut désormais être formé à des techniques difficiles à maîtriser (comme la laparoscopie) en les pratiquant dans un premier temps sur un patient virtuel.

Les avantages de tels systèmes en terme de formation sont nombreux. Tout d'abord, ils permettent de s'entraîner à des opérations délicates sans toutefois risquer la vie du novice (apprentissage du déminage, pilotage d'avions). Ensuite, ils permettent de former plus facilement des acteurs métier dans des domaines où toute erreur peut être lourde de conséquences (comme dans le domaine médical). Enfin, ils offrent une opportunité d'entraînements réguliers à des situations n'apparaissant qu'exceptionnellement. Ces systèmes nécessitent l'emploi d'un modèle numérique afin de reproduire au mieux les comportements réels de l'objet manipulé (interaction avec l'utilisateur) et permettre la synthèse de retours sensoriels (visuel, haptique, sonore, etc.)

Mais les applications d'un modèle numérique ne se réduisent pas à l'apprentissage du geste : il peut également être employé à assister l'opérateur humain lors de la tâche elle-même. Il peut s'agir d'un guidage mécanique, par exemple pour un axe de perçage ou un plan de découpe. Il peut aussi s'agir de fournir à l'utilisateur des informations complémentaires à celles fournies par ses cinq sens, comme c'est le cas en réalité augmentée. Ces systèmes sont des systèmes d'assistance au geste métier.

Ce manuscrit présente une démarche similaire, mais cette fois-ci adressée aux métiers de l'art, et plus particulièrement au modelage de matériaux tels que la glaise ou la plas-

ticine. L'œuvre de l'esprit humain a besoin de la matière pour être validée : l'artiste ou le designer a donc besoin d'une maquette réelle. Mais pour faire évoluer et enrichir la maquette réelle, il serait nécessaire de disposer d'une maquette virtuelle permettant d'intégrer des contraintes fonctionnelles, ergonomiques, esthétiques, ou de fabrication. Cette approche débouche sur une nouvelle démarche méthodologique et sur un nouvel outil pour la création d'objets de forme complexe par modelage, intégrant à la fois matière réelle et modélisation virtuelle. Grâce à la maquette réelle, la maîtrise dimensionnelle de l'œuvre est conservée. Grâce à la maquette virtuelle, on dispose de tous les outils offerts par la CAO afin d'enrichir la tâche (modèle de référence, positionnement de l'oeuvre dans la matière, informations dimensionnelles, etc.).

Ce travail s'inscrit dans un projet plus vaste qui a pour objectif l'étude et le développement d'un système pour l'aide au geste des métiers matière. Le but recherché est l'intégration du geste en CAO suivant une démarche qui repose sur la mesure du geste et sur la simulation de l'effet de l'outil afin de construire une maquette virtuelle qui soit l'image du geste métier.

Ce mémoire se divise en quatre chapitres. Dans le premier seront présentées les problématiques liées à l'aide au geste dans le cas du modelage. Au travers de l'étude de différents systèmes d'apprentissage et de supervision d'une tâche manuelle, nous illustrerons le besoin d'un modèle de la matière modelée et les contraintes auxquelles ce modèle doit répondre.

Le second chapitre sera donc, quant à lui, consacré à l'étude des techniques qui permettent de simuler des corps en déformation. Deux axes principaux guideront cet exposé. Dans un premier temps, nous présenterons la classe des matériaux "viscoplastiques", à laquelle appartiennent la glaise et la plasticine, ainsi que les lois physiques qui en régissent les déformations. Les différentes techniques utilisées en infographie afin de les modéliser seront abordées ensuite.

Trois nouveaux modèles ont été développés pour simuler de manière réaliste et interactive le comportement d'un matériau semblable à ceux utilisés en modelage. Tous trois se basent sur une représentation volumique du corps (voxels) ainsi que sur une modélisation physique des déformations. Les deux premiers utilisent un modèle très simplifié : sans notion de viscosité, ils garantissent cependant la conservation de la masse et se basent sur l'emploi d'automates cellulaires. Le dernier, quant à lui, intègre les équations de Navier Stokes de manière plus complète.

Dans le dernier chapitre sera enfin évoqué le problème du recalage entre modèle virtuel, issu du calcul, et modèle réel, obtenu par le travail de l'artiste. Quelle que soit la technique utilisée, la modélisation d'un processus physique complexe reste entachée d'erreurs dues aux approximations effectuées. Cette étape permet de recalibrer ponctuellement le modèle virtuel afin qu'il reste cohérent avec le modèle réel. Nous concluons enfin ce travail en effectuant un bilan des contributions proposées dans ce manuscrit ainsi que des résultats obtenus. Des prolongements possibles de cette étude à court et long terme seront alors proposés.

Chapitre 1

Aide au geste pour le modelage

1.1 Introduction

De nombreux métiers reposent sur l’habileté de l’être humain à modifier la matière : la gravure, la sculpture, la chirurgie ou encore l’odontologie¹ en sont des exemples. Toutes ces activités ont en commun le fait de reposer sur **un geste métier** qui ne s’acquiert que par un apprentissage en général assez long. Ce travail s’inscrit dans une démarche qui consiste à proposer un outil de formation dédié aux métiers de l’art, et plus particulièrement aux activités de modelage.

Dans la première partie de ce chapitre seront présentés les enjeux et les problématiques associés aux systèmes d’aide au geste. Ce travail sera effectué au travers de l’étude de deux systèmes : l’un d’apprentissage, l’autre de planification et de supervision.

La seconde partie s’attardera sur la structure des différents systèmes dédiés au geste métier ou à l’apprentissage par simulation virtuelle. En effet, deux alternatives pourraient être envisageables : on peut soit enregistrer le mouvement, soit enregistrer ses effets.

Nous présenterons enfin dans la dernière section les différentes possibilités offertes par un simulateur de modelage, et avec elles la problématique de cette thèse.

1.2 Les systèmes d’aide au geste

Conformément à l’idée couramment admise, les métiers orientés matière (sculpture, modelage, etc.) sont l’un des domaines où l’informatique est relativement absente. Les techniques et les savoir-faire changent peu et se transmettent par l’apprentissage.

En effet, la tâche réalisée est profondément ancrée dans le présent : il est difficile, en ne contemplant que l’objet final, de se représenter l’enchaînement des opérations ayant mené à ce résultat. Ainsi, la transmission du savoir ne peut se faire sans la pratique répétée de l’exercice et l’acquisition de cette dimension temporelle de la tâche.

¹Odontologie : Science consacrée à l’étude des dents et de leurs maladies

Beaucoup de ces activités se caractérisent aussi par un aspect "irréversible" de certaines opérations. Dans le cas de la sculpture par exemple, un retrait excessif de matière ne pourra être annulé : il est impossible de revenir en arrière et corriger une telle erreur.

Un système de CAO peut apporter beaucoup à ces métiers, et ce à plusieurs niveaux. Tout d'abord, il peut procurer à l'acteur humain des informations complémentaires, comme par exemple la dimension temporelle de la tâche qui manquait jusque là, où encore lui permettre de s'exercer. De tels systèmes sont qualifiés de systèmes pour l'aide au geste.

On peut distinguer trois grandes familles de systèmes :

- Les simulateurs, qui permettent l'entraînement de l'acteur métier.
- Les systèmes de planification opératoire, qui interviennent avant le geste et sont destinés à le préparer. Ils s'appuient sur des modélisations du réel (numérisation) auxquelles on adjoint des informations liées à la tâche à accomplir.
- Les systèmes de navigation ou supervision de la tâche, qui assistent l'humain en lui fournissant des indications complémentaires durant le processus.

1.2.1 Les simulateurs

Les simulateurs permettent à un acteur humain de s'entraîner à la réalisation d'une tâche sans toutefois s'exposer aux risques inhérents à celle-ci. L'objet de toute simulation est de tromper l'utilisateur, c'est à dire lui donner la sensation du réel en envoyant à ses sens des informations construites de toutes pièces. Plus ces informations sont réalistes, c'est à dire conformes à ce que percevrait l'utilisateur dans une situation réelle, et plus la simulation est réussie. Le retour sensoriel peut être :

- Haptique ([LCK⁺00], [DAK04], [LKC⁺01])
- Visuel ([LC87a], [KdMD⁺02], [RFC⁺02])
- Auditif ([CTLA03])
- etc.

L'exemple le plus connu de simulateur est bien sur le simulateur de vol, qui permet aux élèves pilotes d'apprendre à diriger un appareil sans quitter le sol et risquer le crash. L'ordinateur leur permet de développer un savoir faire, le pilotage, sans se mettre en danger en cas de fausse manœuvre.

Le danger pour l'utilisateur n'est toutefois pas la seule raison qui puisse rendre un simulateur intéressant. Parfois, comme c'est le cas par exemple dans le domaine médical, il est difficile de permettre à l'acteur d'apprendre le geste sans que cela soit dommageable. Un autre exemple de simulateur permettant l'acquisition d'un savoir faire a été mis au point par l'équipe EPIDAURE, de l'Inria.

Il s'agit là d'un simulateur de chirurgie hépatique par laparoscopie, utilisé pour former des opérateurs à la chirurgie minimalement invasive. Le problème avec la technique de la laparoscopie est qu'elle est assez peu intuitive pour le chirurgien : la notion de profondeur est absente en raison du manque de stéréoscopie, la position de la caméra peut changer, et avec elle les mouvements relatifs de l'outil sur l'écran. On estime à une cinquantaine le nombre d'opérations que doit effectuer un chirurgien novice avant de commencer à maîtriser cette technique.

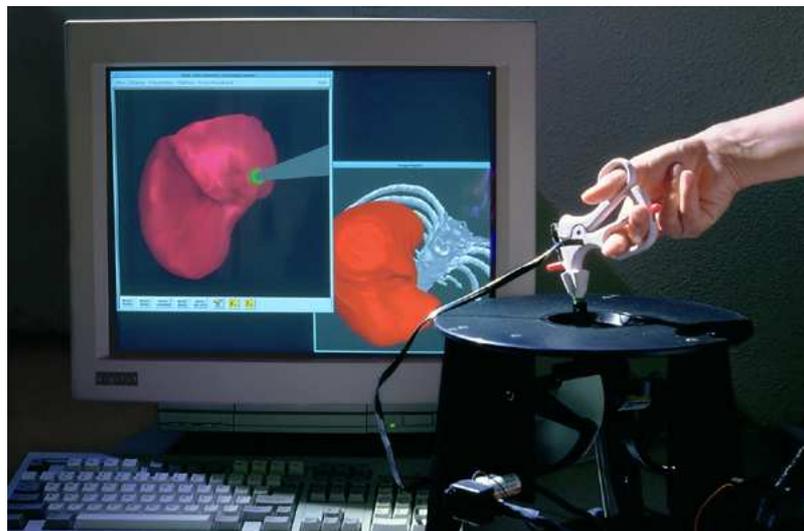


FIG. 1.1 – Simulateur de chirurgie hépatique par laparoscopie (Epidaure, INRIA)

Ce système permet de reproduire de manière réaliste le comportement d'une sonde laparoscopique durant une opération. Le rendu est à la fois visuel, grâce à l'utilisation de modèles déformables reproduisant les réactions du foie à l'action de l'outil, et tactile, par le truchement d'un retour haptique.

Il est important de souligner ici l'importance du modèle de déformation utilisé pour simuler le comportement du foie. Le modèle utilisé ici, développé par Gilles Debunne dans sa thèse ([Deb00]), est un modèle physique basé sur les lois de l'élasticité. En effet, pour que le système puisse remplir son rôle de formation à la chirurgie hépatique, le foie virtuel doit réagir de façon aussi réaliste que possible et bien sûr de manière interactive.

Ce constat peut être étendu à tous les modèles utilisés à des fins de simulation. Le comportement du matériau ou du processus modélisé, pour que la simulation apporte quelque chose dans le domaine métier, doit être le plus réaliste possible. Ceci implique que la simulation soit basée sur la physique sur laquelle repose le phénomène. Malheureusement, une telle approche nécessite de lourds calculs, qui sont pour le moment encore souvent incompatibles avec une exécution en temps réel.

1.2.2 Les systèmes de supervision

Si les simulateurs permettent de s'entraîner, d'autres systèmes ont pour vocation d'aider l'acteur humain lors de la réalisation de la tâche, soit en le guidant soit en lui donnant des indications complémentaires à celles fournies par ses cinq sens.

On peut prendre comme exemple de tels systèmes les travaux réalisés par notre équipe concernant l'aide à la reproduction d'œuvres sculptées : [CD03][Cro03]. Ils ont abouti à la mise au point d'un outil intégrant l'aspect "métier" pour la société Apart.

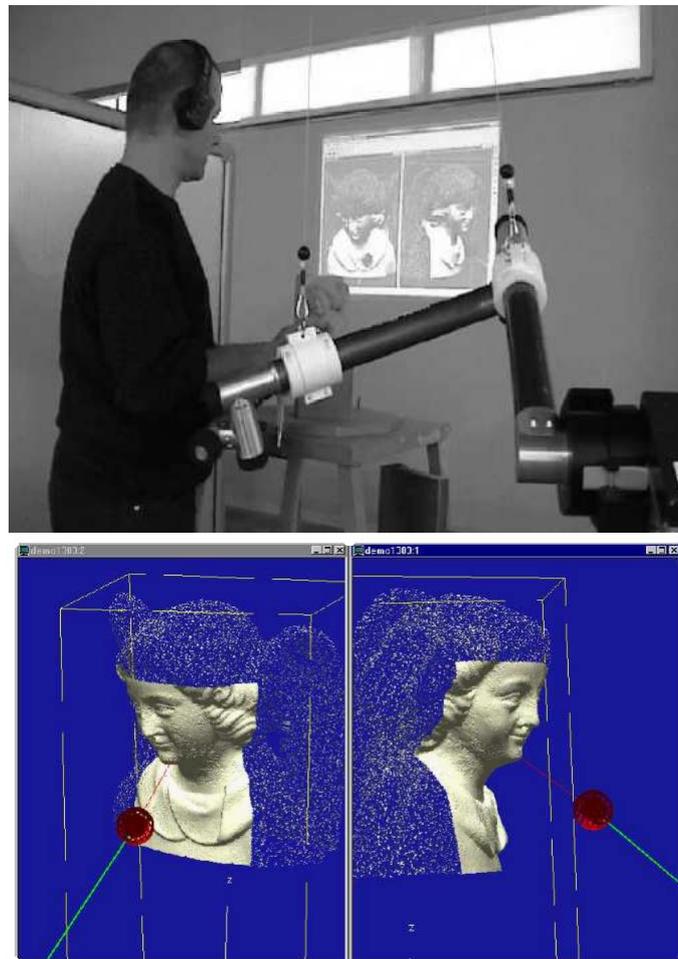


FIG. 1.2 – Apart : Reproduction de sculpture assistée par ordinateur

Les travaux concernent la réalisation d'un système d'aide au geste permettant à un sculpteur de reproduire des originaux en s'appuyant sur un modèle numérique. Le système a été conçu pour le travail de la matière. Pour cela l'opérateur dispose d'un outil standard (meuleuse, disqueuse) et travaille la matière par étapes (ébauche, semi-finition, finition) afin d'aboutir à la forme de l'original. Le geste tout au long du travail est guidé à travers un dispositif qui réalise la synthèse d'un retour sensoriel (visuel, auditif, haptique) calculé à partir d'un modèle numérique (fig 1.2).

La problématique concerne le calcul en temps réel de la position de l'outil par rapport au modèle numérique de la forme à reproduire. Si le modèle géométrique de l'outil est relativement simple, le modèle numérique de la forme à reproduire peut atteindre plusieurs dizaines de millions de polygones pour des originaux de quelques dizaines de centimètres. A partir du mouvement de l'outil, sont déterminés les interactions outil/matière et par la suite le retour sensoriel. L'un des intérêts de ce système en terme de fonctionnalité métier réside dans sa capacité à prévenir une erreur du sculpteur, en lui signalant par un retour sonore, par exemple, le fait qu'il s'approche de la marge de sécurité qu'il s'était fixé durant la préparation de la tâche. Il permet, au delà de cette supervision, d'aider l'acteur métier en lui fournissant un positionnement tridimensionnel de la sculpture dans le bloc de matière brute.

Les enjeux d'un tel système sont doubles. En effet, le travail s'effectue en deux temps :

1. La phase de planification opératoire se déroule avant le processus en lui même. Durant cette phase, l'outil virtuel permet à l'humain de préparer son travail et parfois même de simuler le geste qu'il va devoir effectuer. Dans le cas du système de reproduction de sculpture, c'est durant cette phase que le sculpteur va choisir dans quel sens utiliser son bloc de matière et comment y placer la forme à reproduire. Il peut aussi fixer les valeurs de certains paramètres de la tâche de supervision, comme par exemple la distance minimale en deçà de laquelle il désire déclencher un stimulus sonore, ou encore le facteur d'échelle de la reproduction qu'il désire effectuer.
2. La navigation, quant à elle, correspond à la phase durant laquelle le geste métier est exécuté. Il s'agit là d'un fonctionnement en temps réel du système en réponse aux actions de l'opérateur. Comme on peut le voir sur les photos de la figure 1.2, le modèle est mis à jour grâce à une capture du geste.

Tout comme en simulation, l'un des facteurs clés de la supervision de tâche est donc la capacité du système à fonctionner de manière interactive. Ceci suppose, ici encore, des temps de calcul compatibles avec cette contrainte d'interactivité.

1.3 Architecture des systèmes pour l'aide au geste

Comme on peut aisément le voir, la particularité des métiers matière est la relation très intime existant entre le geste et la matière. Il existe, de ce fait, deux approches différentes afin d'enregistrer le travail d'un acteur métier matière :

- L'approche par le résultat, c'est à dire par l'état de la matière,
- L'approche par le geste, c'est à dire par le mouvement effectué par l'outil.

Ces deux manières d'envisager le problème sont illustrées dans la figure 1.3. Au centre, dans les cellules de couleur foncée, figurent les différentes étapes qui constituent le travail de la matière. L'acteur humain, en effectuant des gestes précis, vient modifier par touches

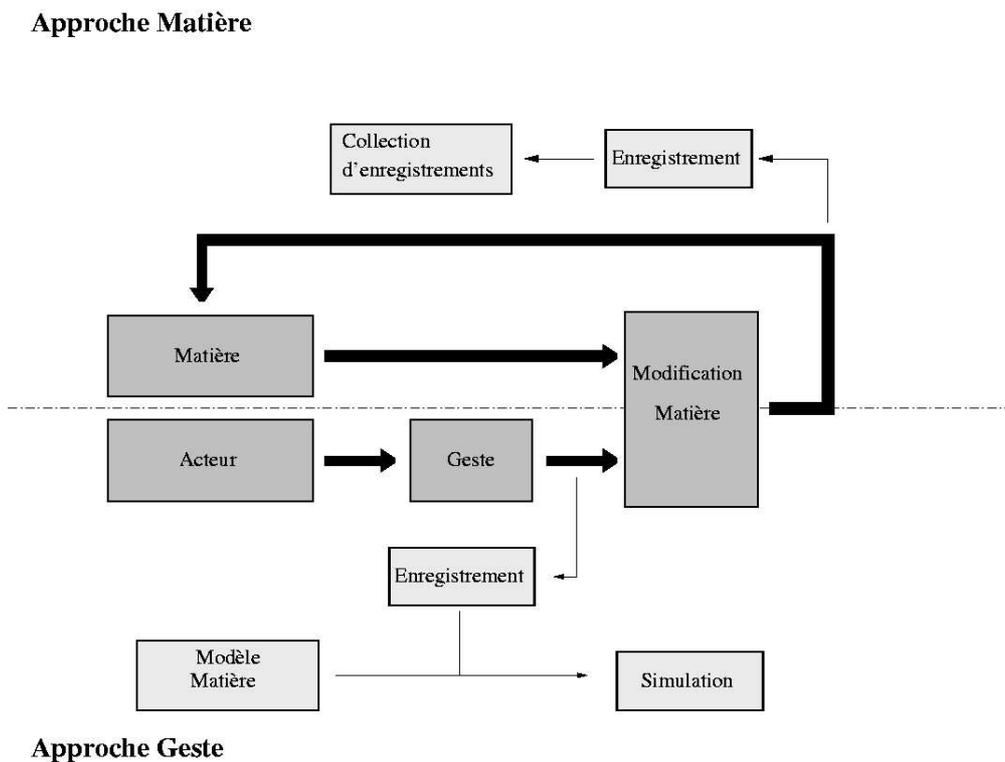


FIG. 1.3 – Approche geste et approche matière

successives le matériau sur lequel il travaille afin de lui donner la forme ou l'état désiré.

L'approche par le résultat, illustrée dans la partie supérieure de la figure, permettrait d'enregistrer régulièrement l'état du matériau travaillé afin de constituer une sorte d'historique des étapes de conception qui ont mené à l'œuvre finale. Le geste qui a mené à ce résultat n'est pas enregistré, il est nécessaire de le reconstituer par imitation et intuition. Ce type d'approche n'est en général pas utilisé dans les systèmes d'aide au geste, car la reproduction du processus ne peut être confié qu'à un acteur métier expérimenté et familier des techniques employées. On lui préfère l'approche par le geste, car il est beaucoup plus aisé de calculer l'état de la matière travaillée à partir d'une séquence de déplacements que le contraire.

Dans l'approche par le geste, le système de CAO est conçu autour de l'être humain. Tout comme pour le système Apart, on enregistre à chaque instant les gestes effectués par l'acteur afin de pouvoir les reproduire. La partie inférieure de la figure 1.3 illustre ce principe. Pour avoir une image complète du processus, il faut à la fois les mouvements et leur effets. On obtient ceci à l'aide d'un modèle virtuel du matériau auquel on applique la séquence de mouvements enregistrée.

Comme nous le verrons par la suite, les modèles utilisés sont en général assez peu complexes. Dans le système Apart, par exemple, la matière est enlevée complètement au passage de l'outil, sans aucune déformation. En raison des difficultés liées à la contrainte d'interactivité, la plupart des travaux utilisent des modèles très simplifiés des phénomènes

physiques concernés : interactions élastiques linéaires, modèles à particules, modèles géométriques, etc.

Ces modèles simples suffisent amplement lorsque seul un réalisme "visuel" est suffisant, mais ils sont inadaptés dès lors que le modèle virtuel doit être rapproché d'un véritable matériau comme dans le système qui nous intéresse.

1.4 Aide au geste pour le modelage

Dans la perspective d'un système d'apprentissage du modelage, deux voies très différentes peuvent être suivies : une approche complètement basée sur la réalité virtuelle, comparable au fonctionnement du projet développé par Epidaure, ou alors une approche mixte, incluant la manipulation d'un véritable matériau.

L'approche qui consisterait à ne travailler que dans un environnement virtuel pourrait être ramenée à un problème d'animation interactive (figure 1.4). A partir d'une mesure des positions de l'outil tenu par l'acteur métier, un modèle d'interaction permettrait de mettre à jour le corps virtuel suivant un modèle de déformation simplifié. La synthèse d'un retour d'effort pourrait être envisagée, basée ce modèle.

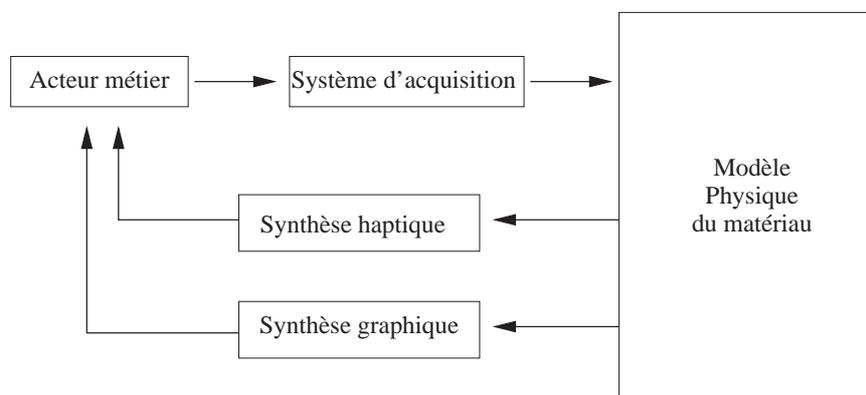


FIG. 1.4 – Schéma de principe d'un simulateur virtuel pour le modelage

L'inconvénient principal d'une approche "tout virtuel" pour le métier est l'absence de maîtrise dimensionnelle durant le modelage. Un autre problème, majeur lui aussi, est l'absence de contact avec la matière. Ce contact apporte de façon inconsciente quantité d'informations à l'artiste par rapport au comportement du matériau ou à la forme qu'il est en train de lui imposer. Les solutions existantes de retour haptique (bras maître, retour de touché) ne sont pas encore assez performantes pour restituer complètement ce contact.

Une approche mixte, à mi chemin entre un simulateur virtuel et un apprentissage classique, permet de surmonter ces difficultés. Comme montré sur la figure 1.5, l'acteur manipule un véritable bloc de matière. Une mesure des gestes est effectuée (position, efforts) et permet de faire évoluer en parallèle un modèle virtuel du matériau modelé. Tout en effectuant un travail dans le sens classique du terme, l'artiste peut ainsi bénéficier des avantages du système informatique : intégration temporelle, capitalisation du geste ou du

savoir-faire, etc.

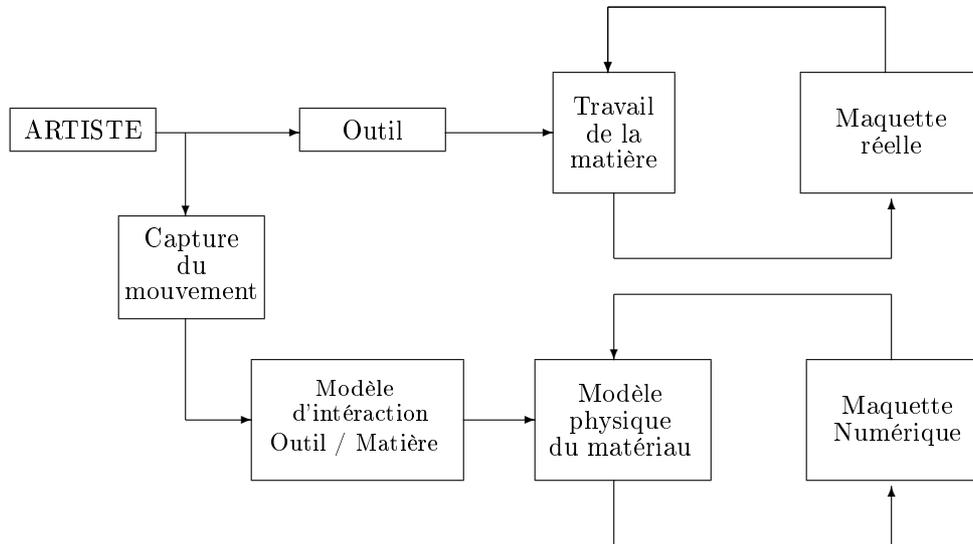


FIG. 1.5 – Schéma de principe d'un système basé sur une approche mixte

Si ce système paraît très proche du précédent, les contraintes qui lui sont associées sont beaucoup plus rigoureuses en raison de la remontée du geste vers le modèle virtuel. A la différence de la première approche, un réalisme apparent ne suffit plus : le modèle de matériau virtuel doit maintenant avoir un comportement aussi proche que possible de celui du matériau réel, puisqu'il pourra lui être comparé à tout moment. Ceci nous contraint à utiliser un modèle de comportement basé sur la physique des déformations, ce qui est plus complexe et bien sûr coûteux en temps de calcul.

Un autre problème associé à l'utilisation de deux modèles, l'un réel et l'autre virtuel, est celui du recalage. Il est en effet impossible de concevoir un modèle traduisant un comportement physique parfait, ce qui veut dire que le comportement du modèle virtuel finira toujours par diverger si on ne le corrige pas régulièrement. C'est le rôle de la phase de recalage. Cette phase intervient à deux moments distincts de la tâche :

- **Lors de l'initialisation du processus**, qui nécessite l'acquisition de la forme initiale de la matière (ainsi qu'un étalonnage géométrique de la cellule).
- **Pendant le processus**, à intervalles réguliers, afin corriger les erreurs éventuelles.

Les problématiques de ce travail sont ainsi au nombre de deux. La première est la conception d'un modèle de matériau qui soit à la fois réaliste, interactif, et basé sur la physique des déformations. L'autre problème est de pouvoir comparer ce modèle virtuel au modèle réel.

1.5 Description du système étudié

Le dispositif envisagé englobe trois fonctionnalités distinctes :

- Une fonction "modelage classique",
- Une fonction "modelage numérique",
- Une fonction "acquisition et recalage".

La fonction "modelage classique" correspond au travail de modelage d'un véritable matériau, qui suppose donc la présence d'un support réel (plasticine, glaise) et des fournitures habituelles (outils, plateau, etc.). Elle correspond à la manière traditionnelle de travailler.

La fonction "modelage numérique", elle, correspond à la superposition d'une couche informatique à la fonction précédente. Composée d'un système d'acquisition du geste (ici grâce à un bras Phantom) et d'un modèle numérique du matériau, elle constitue l'apport principal de ce système à l'acteur métier. On peut envisager d'enrichir ici la tâche à l'aide d'informations complémentaires venant assister l'acteur métier.

Enfin, la fonction "acquisition et recalage" se compose de d'un scanner 3D, permettant la numérisation, ainsi que d'un mécanisme de recalage permettant de corriger le modèle numérique afin de conserver sa cohérence par rapport au modèle réel.

La figure 1.6 illustre le dispositif expérimental :

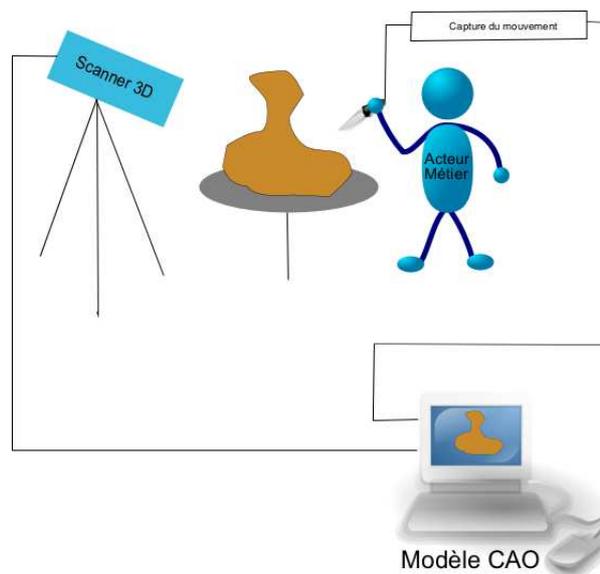


FIG. 1.6 – Schéma de principe

L'acteur métier travaille à l'aide de l'outil (spatule) un matériau brut. Grâce au système de capture, ses gestes sont reportés sur le modèle numérique afin de le faire évoluer de manière parallèle. Le système d'acquisition est formé d'une caméra video et d'un plateau

rotatif : il fonctionne suivant le principe de Shape from Silhouette. A intervalles réguliers, l'acteur métier interrompt son travail afin d'effectuer une renumérisation de la matière modelée et corriger ainsi les imperfections du modèle numérique.

Les impératifs en termes de performance du système sont avant tout liées à l'interactivité : le modèle numérique doit évoluer en temps réel aux interactions de l'utilisateur, ce qui impose un taux de rafraichissement du modèle de l'ordre de la dizaine de Hertz. En ce qui concerne l'acquisition 3D, celle-ci doit se faire de la manière la plus automatisée possible, afin de ne pas imposer un surplus de travail déraisonnable à l'acteur métier.

Chapitre 2

Modélisation des déformations

2.1 Introduction

Comme exposé dans le chapitre précédent, l'une des problématiques liées à ce travail est la modélisation interactive d'un corps subissant des déformations (pâte à modeler, plasticine, glaise, etc.). Le problème est que les principes sur lesquels peut reposer une telle modélisation sont très variés : approche volumique ou surfacique, à voisinage constant ou variant, etc. Ce chapitre est dédié aux modèles de déformations que nous avons pu relever dans différents travaux. Il se décompose en deux parties.

La classe des corps viscoplastiques, à laquelle appartiennent les matériaux que nous désirons modéliser, est tout d'abord présentée d'un point de vue physique. Cette première partie inclut une introduction aux modèles rhéologiques ainsi qu'aux bases de la mécanique des fluides.

La seconde moitié de ce chapitre est, par contre, consacrée aux modèles de corps déformables et aux modèles de fluides utilisés en animation ou en réalité virtuelle. Nous verrons sur quels principes ils reposent, et en quoi ils sont ou ne sont pas adaptés à nos besoins.

2.2 Physique de la déformation plastique

2.2.1 Introduction et définitions

Macroscopiquement, les matériaux montrent une très grande variété de comportements. Les solides, les liquides et les gaz ont des comportements idéaux caractérisés par des propriétés comme la compressibilité, la viscosité, l'élasticité, la solidité, ou la dureté. Mais les matériaux ne se comportent pas toujours comme ces modèles.

Les pâtes, les polymères, et un certain nombre d'autres corps, ne font partie ni des solides, ni des liquides : on ne peut donc pas vraiment les caractériser à l'aide de ce classement. L'étude de ces comportements atypiques amène à penser qu'il n'y a pas en général de distinction claire entre un liquide et un solide. Pour certains un solide doit seulement être défini comme un liquide possédant une très grande viscosité. On fixe arbitrairement

une limite de 10^{13} poises au-dessus de laquelle un matériau est un solide, et en dessous un liquide.

Soumis à de petites contraintes, les solides sont en général élastiques : ils se déforment mais retrouvent leur forme d'origine quand la contrainte est relâchée. Soumis à des contraintes plus importantes, ils peuvent casser ou montrer ce que l'on appelle un comportement plastique en se déformant de manière définitive. **La glaise et la plasticine font partie des corps viscoplastiques** : ce sont des corps qui présentent une contrainte minimale en dessous de laquelle il n'y a pas d'écoulement.

Les matériaux que nous cherchons à modéliser pour notre projet sont des matériaux présentant des déformations régies par des lois complexes, que nous nous proposons d'étudier dans ce chapitre. Cette étude se fera suivant deux approches : dans la première, l'approche rhéologique, nous définirons plus précisément la viscoplasticité et étudierons les modèles physiques généralement utilisés pour les modéliser. La seconde partie, quant à elle, s'attachera à décrire le matériau comme un fluide doté d'une très grande viscosité et obéissant aux lois de la mécanique des fluides.

2.2.2 Approche rhéologique

Étymologiquement, la rhéologie est une discipline qui traite de l'écoulement et des déformations des matériaux sous l'action de contraintes. En réalité, son champ d'étude est beaucoup plus précis. L'élasticité et la mécanique des fluides ne permettant pas de décrire les propriétés de matériaux au comportement intermédiaire entre celui du solide élastique parfait et celui du fluide newtonien, il est apparu nécessaire de développer une science nouvelle. La rhéologie permet de relier les contraintes, les déformations et leur dérivées et caractérise la nature des comportements.

Il serait bien entendu impossible de résumer l'intégralité de la rhéologie dans les quelques pages de ce manuscrit : nous nous contenterons donc ici d'une description de ses principes et applications au cas viscoplastique et invitons le lecteur intéressé à se reporter à [CG00] et [GHP01].

Les réponses des matériaux à quelques essais simples permettent de les ranger dans des classes bien définies. Ces comportements de base, à savoir l'élasticité, la plasticité et la viscosité, peuvent être représentés par les systèmes mécaniques suivants :

- **Le ressort**, qui permet de représenter l'élasticité linéaire parfaite (loi de Hooke). La déformation est ici totalement réversible et il existe une relation biunivoque entre la contrainte τ et la déformation ε .
- **L'amortisseur**, qui schématise la viscosité. Elle est dite pure si la relation entre charge et vitesse de chargement est biunivoque. Le modèle correspond à la loi de Newton si cette relation est linéaire.
- **Le patin**, qui est un limiteur de contrainte. Tant que la contrainte qui lui est ap-

pliée reste inférieure à un certain seuil τ_c , sa déformation est nulle. Au-delà de ce seuil, le patin glisse, la contrainte reste alors constante et égale à la valeur seuil, la déformation pouvant devenir aussi grande que l'on veut. C'est un élément non linéaire.

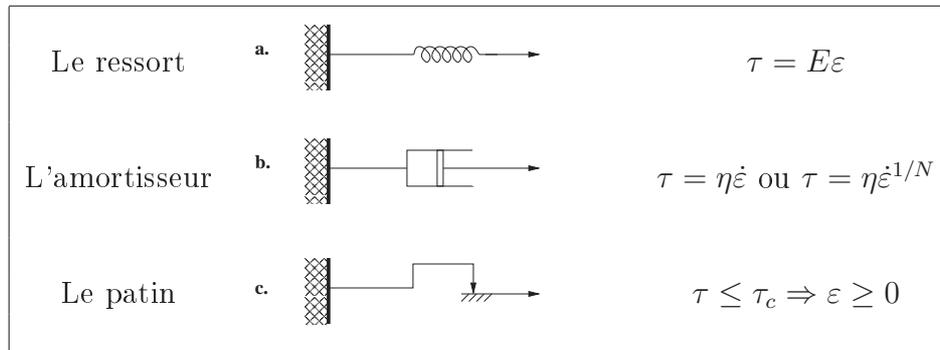


FIG. 2.1 – Les éléments de base de la rhéologie

Un matériau est dit **linéaire** si la fonction $\varepsilon = f(\tau)$ est elle-même linéaire. C'est le cas de la plupart des matériaux soumis à de petites contraintes et la combinaison de ces briques de base, grâce au principe de superposition de Boltzmann, permet de modéliser la quasi-totalité des comportements rencontrés.

En rhéologie, pâte à modeler, glaise et plastiline font partie des liquides dénommés "liquides plastiques". Il s'agit de liquides qui ne s'écoulent qu'à partir d'une certaine valeur de contrainte, notée τ_c et appelée "seuil d'écoulement". L'existence de ce seuil d'écoulement est généralement mise sur le compte de l'existence, au repos, d'une structure tridimensionnelle rigide dans le corps : des particules emboîtées les unes dans les autres, ou floculées. Le seuil correspond à la force nécessaire pour les séparer, c'est à dire vaincre les forces de cohésion qui les relient (forces de Van der Waals) et provoquer l'écoulement. Au delà de ce seuil, la structure rigide est détruite et l'écoulement s'effectue sous l'effet de la contrainte $\tau - \tau_c$.

On distingue deux types de liquides plastiques :

– Les liquides de Bingham

Lorsque la contrainte appliquée dépasse τ_c , les liquides de Bingham adoptent le comportement d'un liquide newtonien : la relation entre τ et $\dot{\varepsilon}$ est linéaire. Leur rhéogramme est représenté sur la figure 2.2.

L'équation rhéologique d'état associée à ce modèle s'écrit :

$$\begin{cases} \tau < \tau_c & \dot{\varepsilon} = 0 \\ \tau \geq \tau_c & \tau = \tau_c + \alpha\dot{\varepsilon} \end{cases} \quad (2.1)$$

où α est un coefficient appelé viscosité plastique.

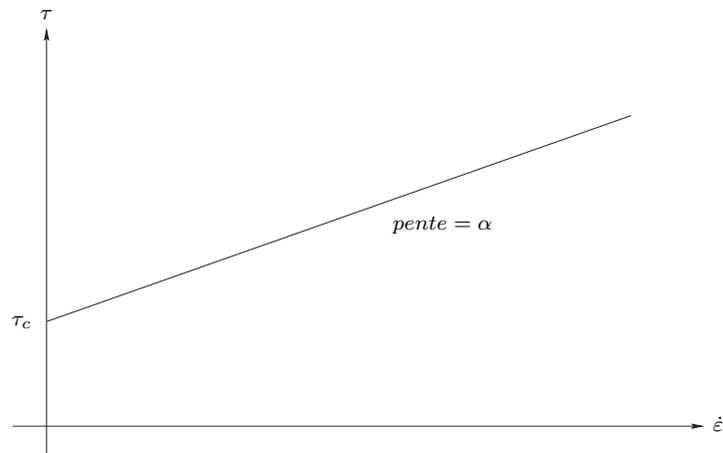


FIG. 2.2 – Rhéogramme d'un corps de Bingham

Un exemple très connu de ce type de corps est celui des peintures à l'huile, étudiées historiquement par Bingham. Même si les huiles utilisées dans ces préparations sont des liquides visqueux newtoniens, ces peintures sont en fait des suspensions. Elles ont la propriété remarquable de s'étaler facilement, sans laisser de traces de pinceau (ce qui suppose une faible viscosité sous contrainte), et en même temps de ne pas couler (ce qui suppose une très grande viscosité au repos).

Il s'agit bien sûr là d'un **comportement idéal**. On peut le modéliser par l'association en parallèle d'un patin et d'un amortisseur :

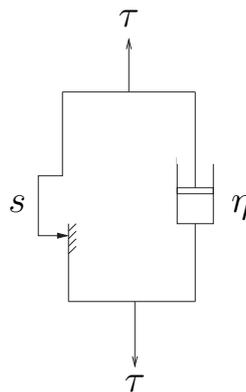


FIG. 2.3 – Le corps de Bingham : système Patin - Amortisseur

L'association de deux éléments en parallèle impose :

$$\begin{cases} \varepsilon = \varepsilon_1 = \varepsilon_2 \\ \tau = \tau_1 + \tau_2 \end{cases} \quad (2.2)$$

Ce qui, appliqué avec un patin et un amortisseur, permet de retrouver les équations du corps de Bingham (2.1) :

$$\begin{cases} \tau < \tau_c & \dot{\epsilon} = 0 & \text{pas de glissement du patin} \\ \tau \geq \tau_c & \tau = \tau_c + \eta \dot{\epsilon} & \text{glissement du patin} \end{cases} \quad (2.3)$$

– Les liquides de Casson

A l'opposé des liquides de Bingham, les corps de Casson montrent un comportement plastique **non idéal** qui se traduit par le fait que leur rhéogramme présente le plus souvent une concavité dirigée vers le bas. Il s'agit donc d'un comportement plastique fluidifiant¹.

La plupart des liquides plastiques sont en fait des liquides de Casson, comme par exemple les boues, pâtes dentifrices, la pâte à pain et certaines suspensions. On peut modéliser mécaniquement le comportement des corps de Casson par l'association en série d'éléments de type Bingham (figure 2.3). Pour décrire quantitativement un tel comportement, on peut utiliser deux équations : l'équation de Casson ou l'équation de Herschel-Bulkley.

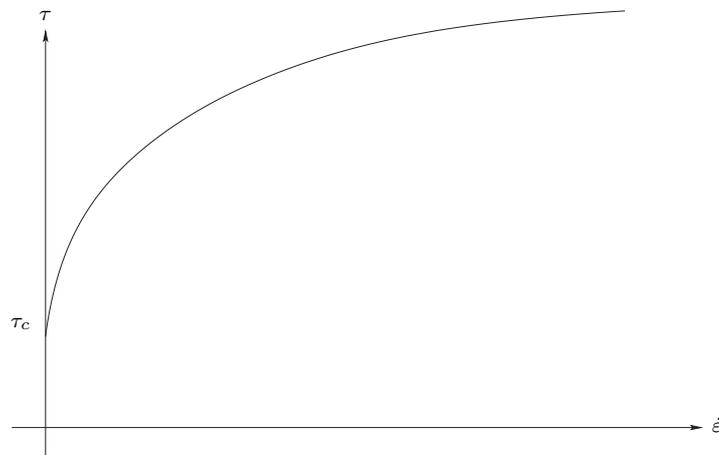


FIG. 2.4 – Rhéogramme d'un corps de Casson

1. Équations de Casson :

$$\begin{cases} \tau < \tau_c & \dot{\epsilon} = 0 \\ \tau \geq \tau_c & \tau^{0.5} = (\tau_c)^{0.5} + (\beta \dot{\epsilon})^{0.5} \end{cases} \quad (2.4)$$

où β est une constante également appelée **viscosité plastique**

¹On peut noter que le comportement plastique épaississant semble excessivement peu fréquent pour ne pas dire inexistant.

2. Équations de Herschel-Bulkley :

$$\begin{cases} \tau < \tau_c & \dot{\epsilon} = 0 \\ \tau \geq \tau_c & \tau = \tau_c + K\dot{\epsilon}^n \end{cases} \quad (2.5)$$

où n est un exposant à identifier en fonction du liquide, dont la valeur est en général inférieure à 1 (comportement plastique fluidifiant).

Ces deux équations permettent, d'après [CG00], d'obtenir de très bonnes corrélations dans la mesure où les vitesses de cisaillement demeurent relativement faibles. En effet, les rhéogrammes tendent asymptotiquement vers une limite rectiligne, région "newtonienne" qui ne peut pas être prise en compte par les équations de Casson ou par celles de Herschel-Bulkley. Dans le cas de vitesses de cisaillement fortes, le modèle de Bingham, plus simple, sera préféré.

Le comportement asymptotique de ces corps, très linéaire, n'est pas sans faire penser au comportement d'un liquide. De là vient l'idée qu'un tel corps peut être aussi envisagé comme un liquide simple présentant une viscosité très importante. Nous allons, dans le paragraphe suivant, développer cette autre approche en introduisant les équations de la mécanique des fluides dans le cas de fluides très visqueux.

2.2.3 Approche par la mécanique des fluides

Comme nous l'avons vu précédemment, l'existence d'une zone newtonienne dans les rhéogrammes observés suggère l'idée que les corps viscoplastiques peuvent être assimilés à des fluides dont la viscosité est extrêmement importante. Nous allons, dans cette section, étudier les équations régissant les écoulements des fluides classiques ainsi que les méthodes de résolution généralement utilisées. Nous nous pencherons, ensuite, sur le cas particulier des fluides visqueux qui correspondent à une forme très particulière de cette équation.

2.2.3.1 Les équations de Navier-Stokes

L'écoulement des fluides est régi par deux équations connues sous le nom d'équations de "Navier-Stokes" : une qui exprime la conservation de la masse de fluide, et une qui, appelée abusivement "équation de quantité de mouvement", est en fait issue du principe fondamental de la dynamique.

Ces deux équations décrivent l'écoulement d'un fluide dit **newtonien**, c'est à dire un matériau dont le rhéogramme est présenté en figure 2.5.

Parmi les fluides newtoniens, on compte entre autres les liquides les plus courants (l'eau par exemple) ainsi que les gaz.

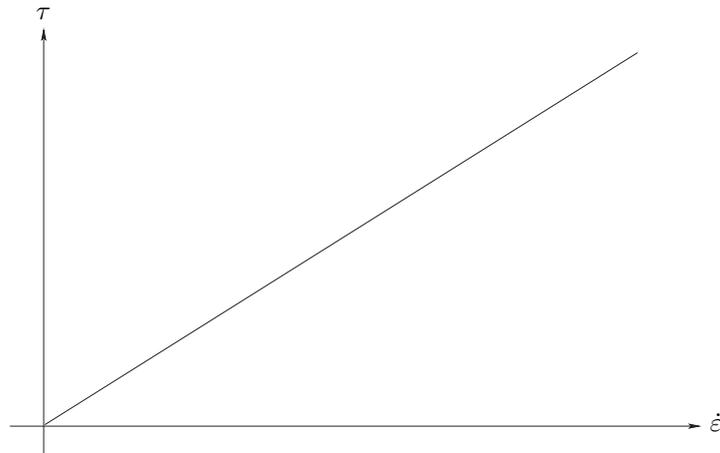


FIG. 2.5 – Rhéogramme d'un liquide newtonien

2.2.3.2 L'équation de conservation de la masse

Soit ρ la masse volumique du fluide et \vec{u} le vecteur vitesse, de composante (u_x, u_y, u_z) , exprimé dans un repère Lagrangien.

La conservation de la masse de fluide contenue dans un domaine Ω s'exprime :

$$\frac{d}{dt} \left(\int_{\Omega} \rho d\Omega \right) = 0 \quad (2.6)$$

Grâce au théorème de Leibnitz, on peut décomposer ce terme en variations observées à l'intérieur du domaine et flux à la frontière Γ :

$$\frac{d}{dt} \left(\int_{\Omega} \rho d\Omega \right) = \int_{\Omega} \frac{\partial \rho}{\partial t} d\Omega + \oint_{\Gamma} \rho \vec{u} \cdot \vec{n} d\Gamma \quad (2.7)$$

Le flux aux frontières $\oint_{\Gamma} \rho \vec{u} \cdot \vec{n} d\Gamma$ est alors ramené à une intégrale de volume $\int_{\Omega} \text{div}(\rho \vec{u}) d\Omega$ par le théorème de Gauss. En reconnaissant que la conservation de la masse doit être assurée pour n'importe quel domaine Ω , on obtient l'expression locale :

$$\boxed{\frac{\partial \rho}{\partial t} + \text{div}(\rho \vec{u}) = 0} \quad (2.8)$$

Ceci est la forme compressible de l'équation de conservation de la masse. Lorsque la masse volumique peut être considérée comme constante (hypothèse d'incompressibilité), elle se simplifie en :

$$\boxed{\text{div}(\vec{u}) = 0} \quad (2.9)$$

2.2.3.3 L'équation de la quantité de mouvement

Il faut maintenant écrire la deuxième loi de la dynamique $\vec{F} = m \frac{d\vec{u}}{dt}$ pour un fluide. Ceci peut être écrit comme la dérivée en temps de la quantité de mouvement contenue dans le domaine Ω . Comme pour la masse, nous pouvons écrire :

$$\int_{\Omega} \frac{\partial(\rho \vec{u})}{\partial t} d\Omega + \int_{\Gamma} \rho \vec{u} \cdot \vec{u} \cdot \vec{n} d\Gamma \quad (2.10)$$

ou encore :

$$\int_{\Omega} \frac{\partial(\rho \vec{u})}{\partial t} d\Omega + \int_{\Omega} \nabla(\rho \vec{u} \otimes \vec{u}) d\Omega \quad (2.11)$$

avec :

- \otimes désignant le produit extérieur entre deux tenseurs d'ordre 1
- ∇ désignant l'opérateur tensoriel "nabla"

Les forces extérieures seront notées $\vec{g} + \vec{F}$, en distinguant l'accélération de pesanteur \vec{g} des autres forces.

Si $d\Gamma$ est un élément de surface du domaine Ω dont la normale extérieure est \vec{n} , l'élément de force appliqué à $d\Gamma$ est :

$$d\vec{F} = \underline{\underline{\sigma}} \vec{n} d\Gamma \quad (2.12)$$

où $\underline{\underline{\sigma}}$ est le tenseur des contraintes.

Les forces appliquées au domaine Ω sont par conséquent :

$$\int_{\Omega} \rho(\vec{g} + \vec{F}) d\Omega + \int_{\Gamma} \underline{\underline{\sigma}} \cdot \vec{n} d\Gamma \quad (2.13)$$

soit encore :

$$\int_{\Omega} \rho(\vec{g} + \vec{F}) d\Omega + \int_{\Omega} \text{div}(\underline{\underline{\sigma}}) d\Omega \quad (2.14)$$

Nous obtenons donc finalement :

$$\boxed{\frac{\partial(\rho \vec{u})}{\partial t} + \nabla(\rho \vec{u} \otimes \vec{u}) = \text{div}(\underline{\underline{\sigma}}) + \rho \vec{g} + \rho \vec{F}} \quad (2.15)$$

Cas newtonien incompressible

Pour les fluides incompressibles newtoniens, cas le plus fréquent, on peut écrire :

$$\begin{cases} \rho = \text{constante} \\ \underline{\underline{\sigma}} = -p \underline{\underline{\delta}} + 2\mu \underline{\underline{D}} \end{cases} \quad (2.16)$$

où p est la pression exprimée en Pascal, $\underline{\underline{\delta}}$ est le tenseur identité, μ le coefficient de viscosité dynamique et $\underline{\underline{D}}$ le tenseur des taux de déformation. Il s'écrit :

$$\underline{\underline{D}} = \frac{1}{2} (\nabla \vec{u} + {}^t \nabla \vec{u}) \quad (2.17)$$

μ est une constante qui dépend du fluide considéré. On peut aussi l'écrire $\rho\nu$ où ν est le coefficient de viscosité cinématique. On obtient donc les équations suivantes en coordonnées cartésiennes :

– **Continuité :**

$$\boxed{\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0} \quad (2.18)$$

– **Quantité de mouvement :**

$$\boxed{\begin{aligned} \frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} + u_z \frac{\partial u_x}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \Delta(u_x) + F_x \\ \frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_y}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \Delta(u_y) + F_y \\ \frac{\partial u_z}{\partial t} + u_x \frac{\partial u_z}{\partial x} + u_y \frac{\partial u_z}{\partial y} + u_z \frac{\partial u_z}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \Delta(u_z) + F_z + g \end{aligned}} \quad (2.19)$$

On peut remarquer que ce système de 4 équations comporte 4 inconnues : u_x , u_y , u_z et p . Sous forme vectorielle, on écrit aussi ces trois dernières équations sous la forme :

$$\boxed{\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \text{grad}) \vec{u} = -\frac{1}{\rho} \text{grad} p + \nu \Delta \vec{u} + \vec{F}} \quad (2.20)$$

2.2.3.4 Les conditions aux limites

Les équations que nous venons d'obtenir sont des équations différentielles partielles. Il convient donc de s'intéresser de manière privilégiée aux conditions aux limites.

- **Interfaces solide-fluide :** Toutes les observations expérimentales et les simulations numériques s'accordent pour affirmer que *la vitesse d'un fluide s'annule au voisinage immédiat d'une paroi solide*. Le fait que la composante normale soit nulle est liée au fait qu'il n'y a pas de pénétration du fluide dans le solide. En revanche, le fait que la composante tangentielle s'annule de même est lié à l'existence de la viscosité : une vitesse discontinue impliquerait une contrainte de cisaillement infinie et donc une divergence de l'énergie dissipée dans l'écoulement.
- **Surfaces libres :** Dans le cas d'une interface fluide-fluide il y a, pour les mêmes raisons que précédemment, continuité de la vitesse tangentielle au travers de l'interface. Si la viscosité de l'un des deux fluides est négligeable devant celle de l'autre, on est alors dans le cas de la surface libre.

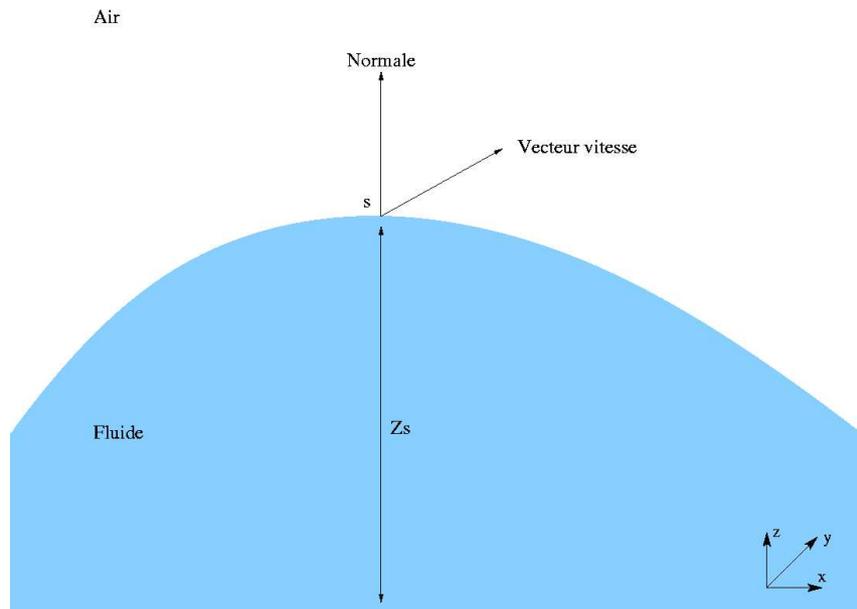


FIG. 2.6 – Cas de la surface libre

La caractéristique des surfaces libres est qu'un point appartenant à la surface reste à la surface du fluide. D'après [Her03], la vitesse d'un point s situé sur une surface libre s'écrit donc :

$$\frac{\partial Z_s}{\partial t} = \vec{u}^s \cdot \vec{n}^s \quad (2.21)$$

2.2.3.5 Résolution des équations de Navier-Stokes

Maintenant que le modèle de comportement des fluides a été établi, nous allons nous intéresser à la résolution des équations de Navier-Stokes. Il existe plusieurs méthodes de résolution. Nous invitons le lecteur qui désirerait approfondir le sujet à se reporter à [Her03] ou [FP02]. Nous nous contenterons, pour la suite, de présenter la méthodologie de résolution et d'approfondir les méthodes de type "différences finies".

Résoudre le problème de Navier-Stokes est rendu difficile par le manque d'équation indépendante pour la pression. Celle-ci intervient dans chacune des trois équations du moment (cf eq 2.19). De plus, il n'y a pas à proprement parler de variable dominante dans l'équation de conservation de la masse (eq 2.18). Cette dernière apparaît comme une contrainte portant sur le champ de vitesse plutôt que comme une équation dynamique.

Une façon de s'affranchir de cette difficulté est de construire un champ de pression qui assure le respect de l'équation de continuité. Cette méthode, qui peut surprendre, est rendue possible par le fait que la valeur de la pression elle-même n'importe pas dans les équations de Navier-Stokes : seul son gradient intervient dans le calcul de l'écoulement.

2.2.3.5.1 L'équation de pression Il est clair que les équations du moment déterminent les composantes du champ de vitesse : leur rôle est donc facilement défini. Ceci ne nous laisse que l'équation de continuité pour calculer le champ de pression, or cette équation ne contient pas la pression. L'astuce généralement utilisée consiste à combiner équation du moment et équation de continuité.

On commence par écrire l'équation (2.20) à l'aide d'une différenciation explicite en temps :

$$\frac{\vec{u}^{t+\delta t} - \vec{u}^t}{\delta t} = -(\vec{u} \cdot \overrightarrow{\text{grad}}) \vec{u} - \frac{1}{\rho} \overrightarrow{\text{grad}}(p) + \nu \Delta \vec{u} \quad (2.22)$$

Grâce à la divergence de cette expression et à l'équation de continuité $\text{div}(\vec{u}) = 0$, on obtient :

$$\boxed{\Delta p = \mu \text{div}(\Delta \vec{u}) - \rho \text{div}((\vec{u} \cdot \overrightarrow{\text{grad}}) \vec{u})} \quad (2.23)$$

Cette nouvelle équation permet alors de calculer explicitement un champ de pression.

2.2.3.5.2 Méthodologie de résolution Maintenant que l'équation de pression est obtenue, la méthodologie de résolution devient plus simple :

- On dispose de \vec{u}^t , champ de vitesse du fluide à l'instant t . \vec{u}^t est supposé de divergence nulle.
- On résout l'équation de pression 2.23 afin d'obtenir un champ de pression compatible avec la contrainte de conservation de la masse.
- On calcule \vec{u}^{t+1} à l'aide des trois équations du moment. Le nouveau champ de vitesse sera de divergence nulle.
- On réitère.

Il est important de souligner le fait que les difficultés numériques ne doivent pas être négligées et une attention toute particulière doit être accordée aux méthodes de différenciation (implicite, explicite, ADI, Cranck-Nicholson ...) car toutes ne convergent pas.

2.2.4 Conclusion

En conclusion, les corps viscoplastiques peuvent être physiquement modélisés suivant deux approches : l'approche fluide ou l'approche solide. Si on les considère comme des solides déformés, alors on suppose qu'une particule de matière donnée conserve son voisinage durant la déformation. La modélisation peut alors se faire grâce à des éléments de liaison comme ceux utilisés en rhéologie. Si au contraire, le matériau est assimilé à un fluide de très grande viscosité, alors la modélisation des déformations se fait par le calcul d'un champ de vitesse grâce aux équations de Navier-Stokes.

2.3 Les modèles d'objets en déformation

2.3.1 Introduction

Alors que l'étude des matériaux vise une modélisation la plus précise possible, et ce quel que soit le temps de calcul, certaines utilisations de ces modèles informatiques répondent à des contraintes assez différentes.

Dans le cas de la simulation virtuelle, par exemple, l'exactitude du modèle physique de déformation importe peu : une simple ressemblance visuelle est souvent suffisante pour tromper l'utilisateur. Le temps de calcul, par contre, est un critère important car l'interactivité du système repose sur lui. En réalité augmentée, le critère temporel reste important, mais la matière modélisée doit se déformer de manière réaliste sans quoi le modèle ne pourra pas être "comparé" au corps réel.

Nous allons, dans les pages suivantes, présenter les travaux existants sur la modélisation des déformations en les classant en deux groupes. Dans la première partie, nous aborderons les méthodes que l'on pourrait qualifier de "visuelles", c'est à dire reposant sur des bases géométriques ou bien sur des algorithmes assez intuitifs. Ces méthodes peuvent donner des résultats visuellement convaincants, mais en de rares cas seulement comparables à une déformation réelle. Dans la seconde partie, par contre, seront présentés des modèles basés sur la physique des déformations, c'est à dire plus réalistes, mais aussi plus complexes et plus difficiles à manipuler.

Nous nous pencherons enfin sur la modélisation des écoulements, et plus particulièrement sur les modèles se basant sur les équations de Navier-Stokes.

2.3.2 Les modèles géométriques

2.3.2.1 Splines et surfaces paramétrables

Cette méthode est la plus ancienne de l'infographie : dans les premiers systèmes de CAO, les designers devaient être en mesure de manipuler simplement la forme d'objets géométriques. De là sont nées les courbes de Bézier et la famille dites des "splines" : courbes d'interpolation, B-splines, splines rationnelles non uniformes (NURBS) dont la caractéristique commune est de définir avec peu de paramètres une courbe lisse en une, deux ou trois dimensions.

Ces courbes (ou surfaces) peuvent être manipulées très simplement grâce à des points de contrôle dont le déplacement permet d'agir sur la courbe (ou la surface) (fig 2.7). Suivant la nature de spline choisie, on peut aussi imposer des contraintes de continuité, dérivabilité, etc.

Un exemple simple : la courbe de Bézier

Soient A, B et C trois points de contrôle de coordonnées supposées connues. La courbe de Bézier simple $[ABC]$ est formée de l'ensemble des points $M(t)$ définis de la manière

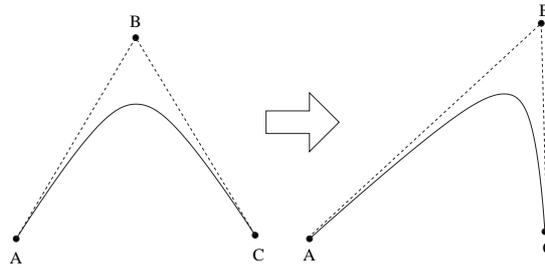


FIG. 2.7 – Déplacement d'un point de contrôle et effet sur une spline

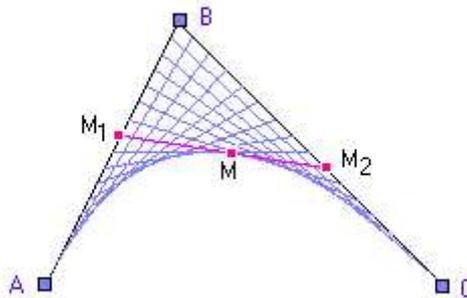


FIG. 2.8 – Construction d'une courbe de Bézier à partir de trois points de contrôle

suivante :

Pour $t \in [0, 1]$:

- Soit $M_1(t)$ le barycentre du système pondéré $\{(A, t); (B, 1 - t)\}$
- Soit $M_2(t)$ le barycentre du système pondéré $\{(B, t); (C, 1 - t)\}$
- $M(t)$ est le barycentre du système pondéré $\{(M_1(t), 1 - t); (M_2(t), t)\}$

En tout point M , la tangente à la courbe est le segment $[M_1 M_2]$. $M(t)$ décrit donc une courbe de Bézier de degré 2, qui, par construction, commence en A et finit en C et a pour tangentes (AB) en A et (BC) en C .

Les propriétés d'association du barycentre nous permettent d'exprimer $M(t)$ plus directement : $M(t)$ est ainsi le barycentre de $\{(A, (1 - t)^2); (B, 2t(1 - t)); (C, t^2)\}$. Les polynômes $(1 - t)^2$, $2t(1 - t)$ et t^2 sont les polynômes de Bernstein de degré 2.

Extension au cas 3D : le patch de Bézier

L'extension de ce cas au 3D porte le nom de patch de Bézier. Le patch est construit à partir d'une matrice $n \times m$ de points de contrôle :

$$\{\tilde{P}_{i,j}, 0 \leq i \leq n, 0 \leq j \leq m\}$$

La surface résultante, maintenant paramétrée par deux variables u et v , est donnée

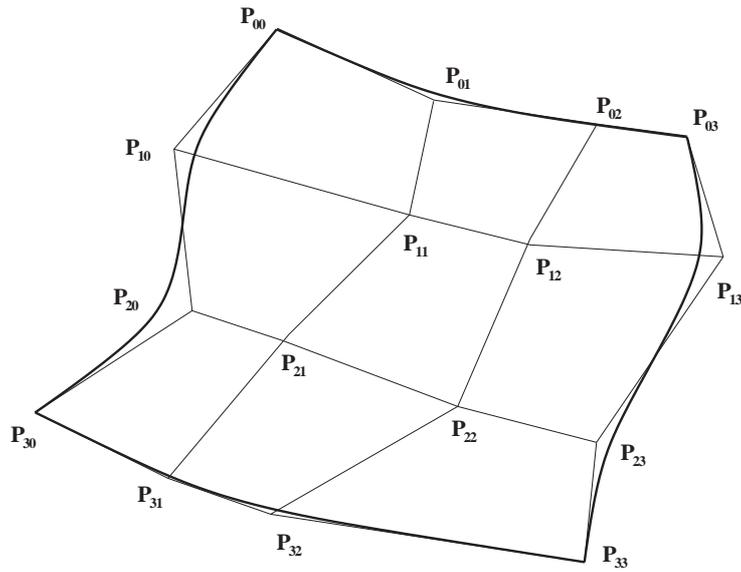


FIG. 2.9 – Patch de Bézier et ses points de contrôle

par l'équation :

$$\tilde{P}(u, v) = \sum_{j=0}^m \sum_{i=0}^n \tilde{P}_{i,j} B_{i,n}(u) B_{j,m}(v) \quad (2.24)$$

où $B_{i,j}(u)$ est un polynôme de Bernstein. Le patch de Bézier est donc une extension très simple du cas 2D. Pour une étude plus complète des splines, le lecteur pourra se reporter à [CRE01] ou [Gal99].

Dans l'étude des déformations d'un corps imposées par un outil, l'action de l'outil sur la matière virtuelle va alors se faire par le déplacement des points de contrôle. Le problème de l'interaction se résume alors un un problème de recherche de collisions entre l'outil et ces points de contrôles.

2.3.2.2 Free Form Deformations

Une autre approche du problème est de ne pas déformer le corps, mais plutôt l'espace dans lequel il se trouve.

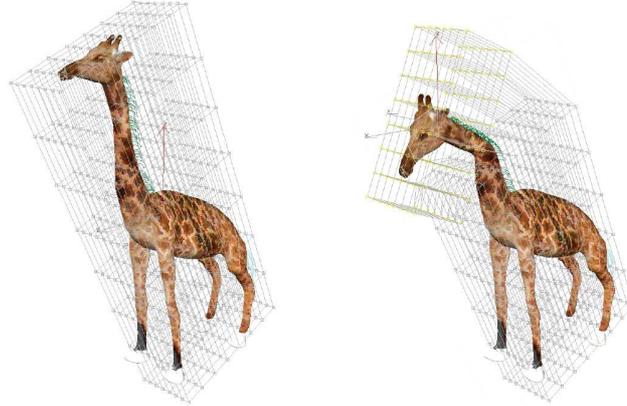


FIG. 2.10 – Déformation de l'espace illustrée sur un modèle de girafe ([Deb00])

Développée à l'origine par Al Barr [Bar84], cette méthode consiste à déformer l'espace à l'aide d'un jeu de fonctions matricielles élémentaires pouvant être combinées (déformations rigides, pliages, etc.)

Sederberg et Parry [SP86] ont ensuite généralisé cette approche par l'utilisation des *Free Form Deformation* (FFD). Un maillage de l'espace est défini, et va pouvoir être déformé. Lors du déplacement de ce maillage, chaque cellule va entraîner avec elle les points qu'elle contient, provoquant ainsi la déformation du corps.

Le procédé se résume en quatre étapes principales :

1. Créer un volume ayant la forme d'un parallélépipède autour de l'objet
2. Imposer des coordonnées locales à chaque point de l'objet à déformer
3. Imposer une grille de points de contrôle sur le parallélépipède
4. Déformer l'objet en bougeant les points de contrôle

Dans la première étape, on définit les coordonnées X de chacun des points du corps en fonction de trois vecteurs $\vec{S}, \vec{T}, \vec{U}$ formant une base de l'espace 3D :

$$\vec{X} = \vec{X}_0 + s \cdot \vec{S} + t \cdot \vec{T} + u \cdot \vec{U} \quad (2.25)$$

où \vec{X}_0 est la coordonnée de l'origine du système paramétrique et (s, t, u) sont les coordonnées paramétriques du point (coordonnées comprises entre 0 et 1).

Une matrice de $(l+1) \times (m+1) \times (n+1)$ points de contrôle $\tilde{P}_{i,j,k}$ est alors définie par la relation :

$$\tilde{P}_{i,j,k} = X_0 + \frac{i}{l} \cdot \vec{S} + \frac{j}{m} \cdot \vec{T} + \frac{k}{n} \cdot \vec{U} \quad (2.26)$$

Les points de contrôle sont alors déplacés par l'utilisateur. La déformation est ensuite appliquée au corps grâce à la relation :

$$X_{ffd} = \sum_{i=0}^l \binom{l}{i} (1-s)^{l-i} s^i \left[\sum_{j=0}^m \binom{m}{j} (1-t)^{m-j} t^j \left[\sum_{k=0}^n \binom{n}{k} (1-u)^{n-k} u^k \tilde{P}_{i,j,k} \right] \right]$$

En 1990, S. Coquillart [Coq90] remarque que les FFD utilisent obligatoirement un parallélépipède comme bloc de contrôle, et propose une extension permettant de spécifier n'importe quelle forme. Ce modèle EFFD (pour Extended Free Form Deformation), repose sur des blocs prismatiques, et les compose pour spécifier une forme quelconque de réseau de déformation.

W.M.Hsu, J.F.Hugues et H. Kaufman, en 1992, proposent quant à eux d'éditer directement la déformation par les points de la surface. Ils utilisent une base B-Spline au lieu des polynômes de Bernstein. Le principal avantage de cette formulation est la localité du contrôle, mais la déformation n'est plus alors forcément incluse dans l'enveloppe convexe des points de contrôle affichés [HHK92].

L'obligation de définir un réseau de déformation (i.e. un ensemble de points et leurs relations) peut sembler assez limitative. L. Moccozet et P. Kalra proposent une autre extension des FFD qu'ils nomment DFFD pour Dirichlet-FFD. L'idée est toujours d'exprimer les coordonnées de chaque point de la surface dans un repère dépendant de points de contrôle, mais cette fois-ci les points de contrôle sont positionnés par l'utilisateur (nombre et position libre) [MMT97].

Plus récemment, B. Crespin s'est intéressé aux aspects de pondération des déformations et à leur influence locale [Cre99]. L'idée est d'associer à chaque primitive déformante un repère et une déformation avec une zone d'influence pondérée par un potentiel scalaire.

En conclusion, les différentes approches des FFD présentent des caractéristiques très semblables aux splines et courbes de Bézier. Ces deux approches sont d'ailleurs assez souvent utilisées de manière conjointe. Les principales caractéristiques en sont donc :

- Un modèle d'interaction outil matière très facile à implémenter, basé le plus souvent sur des points de contrôle,
- Des déformations globales et simples à calculer,
- Un modèle de déformation purement géométrique, sans aucune base physique.

2.3.2.3 Modélisation par surface implicite

La surface d'un objet peut être définie de deux manières fort différentes : **explicitement**, comme nous l'avons fait jusque là, ou bien **implicitement**.

Alors que la description explicite suppose la donnée de la surface à l'aide de polygones ou de courbes, la description implicite repose sur un champ scalaire défini dans tout ou partie de l'espace. La surface est alors le lieu géométrique où cet espace prend une valeur donnée : on l'appelle **surface équipotentielle** ou **isosurface**. L'espace de travail est souvent discrétisé en pixels ou voxels, bien que ce ne soit en aucun cas une obligation.

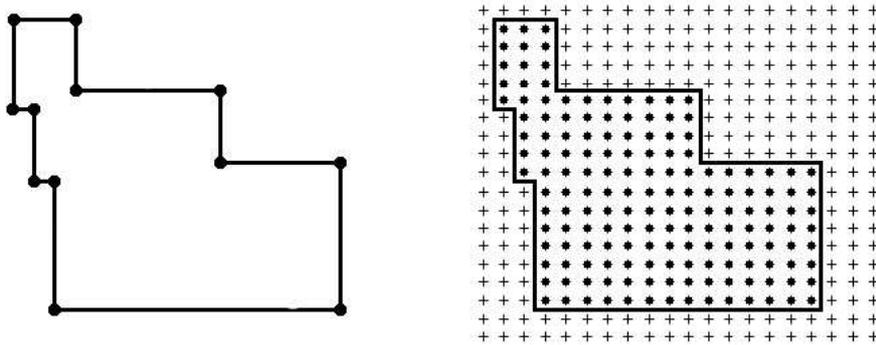


FIG. 2.11 – Description explicite et implicite d'une forme géométrique

Ce type de représentation s'avère beaucoup plus pratique que les représentations explicites dans les cas où l'objet subit de profondes modifications topologiques. Ainsi, par exemple, fusionner deux corps ensemble se fait simplement par une addition des deux champs scalaires.

L'un des apports principaux de la surface implicite est bien sûr la détection de collisions. Alors que les représentations explicites nécessitent des méthodes évoluées telles que les *bounding boxes* ou le *ray casting*, une représentation explicite rend le problème tout à fait trivial. En testant la valeur du champ en un point donné, on peut déterminer instantanément si celui-ci est à l'intérieur ou à l'extérieur du corps.

Le problème des surfaces implicites se situe en général au niveau de la visualisation et plus particulièrement au niveau de la polygonisation. Différents algorithmes existent et permettent de résoudre cette difficulté :

- L'algorithme des *marching cubes* [WMW86], [Blo87], [Blo88], [LC87b], [HC87], [Cra87],
- Les maillages locaux [DTC96],
- Les modèles à particules [WH94].

Le lecteur intéressé par les *marching cubes* pourra se reporter à l'annexe B, consacrée à cet algorithme.

2.3.2.4 Conclusion sur les modèles géométriques

Les modèles géométriques, très prisés en infographie, sont à la fois simples d'utilisation et simples à implémenter. Malheureusement, les déformations qu'ils produisent sont rarement réalistes et n'obéissent en général pas à la contrainte de conservation du volume au cours de la déformation. Nous allons donc maintenant nous intéresser à une seconde classe de modèles : ceux inspirés des lois de la physique.

2.3.3 Les modèles physiques

2.3.3.1 Les Systèmes Masses-Ressorts

Le système Masses-Ressorts est à la fois le modèle le plus simple et le plus intuitif pour représenter des corps mous. De là vient probablement son immense succès dans les logiciels d'infographie et d'animation. Le principe, illustré figure 2.12, est de discrétiser le corps sous la forme d'un nuage de points massiques reliés entre eux par des éléments de type ressort.

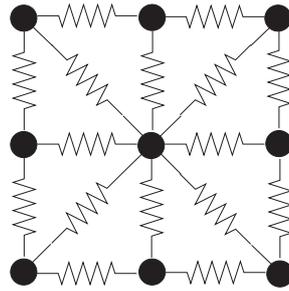


FIG. 2.12 – Exemple de Modèle Masses-Ressorts en 2D

A chaque itération, on commence par calculer les forces appliquées aux masses par chacun des ressorts grâce à la loi de Hooke :

$$\vec{F} = K(L - L_0)\vec{d} \quad (2.27)$$

avec \vec{F} le vecteur force, L l'élongation du ressort, L_0 l'élongation initiale, K sa constante de raideur et \vec{d} son vecteur directeur.

On peut ensuite calculer les nouvelles positions des masses en intégrant l'équation du mouvement pour chacune des masses. Celle-ci est de la forme :

$$m \frac{\partial^2 \vec{x}}{\partial t^2} + K \vec{x} = \vec{f}_{ext} \quad (2.28)$$

avec \vec{f}_{ext} est la somme des forces extérieures et \vec{x} est le vecteur position d'une masse.

Afin de dissiper l'énergie présente dans le système masse ressort, on introduit en général un terme d'amortissement a , qui transforme l'équation du mouvement en :

$$m \frac{\partial^2 \vec{x}}{\partial t^2} + \mathbf{a} \frac{\partial \vec{x}}{\partial t} + K \vec{x} = \vec{f}_{ext} \quad (2.29)$$

Les calculs étant très simples, ce modèle est particulièrement efficace. Son principal défaut est de discrétiser les directions suivant lesquelles s'exercent les efforts au sein du corps modélisé. En effet, les forces ne peuvent contraindre les masses que dans la direction des ressorts, c'est à dire suivant les arêtes du maillage géométrique. L'existence de ces "directions privilégiées" dans les orientations spatiales des forces peut se traduire par une déformation parfois assez peu naturelle. Ainsi, de nombreux travaux ont été effectués afin d'améliorer les systèmes masses ressorts.

Il serait impossible de faire ici la liste de tous les travaux s'intéressant aux modèles masses-ressorts. A titre d'exemple, on pourrait citer les travaux de Y. Chen [CK98] qui introduit de nouveaux types de ressorts (structurels, cisaillement, flexion, etc.) afin de simuler la déformation d'un muscle. Une autre approche, choisie par Hutchinson [HPH96], est de raffiner le réseau de manière adaptative, ce qui permet de réduire le coût en temps de calcul et d'obtenir des résultats plus esthétiques. D'autres encore, comme [Pro95] ou [dCL00] imposent une distance maximale d'élongation des ressorts ou bien encore simulent le déchirement si celle-ci atteint un seuil prédéfini.

Les méthodes d'intégration ont elles aussi fait l'objet de nombreux travaux. Joukhadar, en 1996, propose une méthode d'optimisation du pas d'intégration [Jou96] qui repose sur les variations d'énergie du système. Deux ans plus tard, c'est à Baraff et Witkin de proposer un schéma d'intégration implicite permettant la simulation de ressorts de très forte raideur [BW98], puis à M. Desbrun, avec un algorithme permettant de stabiliser et accélérer le calcul des déformations [DSB99].

En conclusion, les modèles masses-ressorts ont pour principal avantage leur simplicité. D'un point de vue strictement physique, les déformations obtenues peuvent être réalistes dans le cas de matières dont la géométrie impose une ou des directions privilégiées comme c'est le cas pour le tissu (sens des fibres). Il n'y a par contre aucune notion d'écoulement possible : la nécessité d'un maillage spatial du corps les rend inadaptés aux cas où la topologie change fortement au cours de la déformation, comme c'est le cas dans notre problème.

2.3.3.2 Les Modèles à Particules

L'idée de base des systèmes à particules est de reproduire à l'échelle macroscopique les phénomènes régissant la matière à l'échelle atomique. Tout comme pour les systèmes masses-ressorts, les particules sont des masses ponctuelles qui interagissent entre elles.

Cependant, le comportement d'un système à particules est fondamentalement différent de celui d'un système masses-ressorts en ceci que le voisinage d'une particule ne reste pas le même au cours de la déformation. Dans les systèmes masses-ressorts classiques, une masse interagit avec les masses voisines auxquelles elle est reliée : la topologie du système ne change pas. Dans le cas des systèmes à particules, chacune des masses interagit avec toutes les autres et les masses ne sont pas contraintes de conserver un certain voisinage.

Cette absence de restrictions topologiques rend les modèles à particules particulièrement adaptés aux modélisations de fluides : Alan Murta s'est ainsi intéressé à la modélisation de liquides [MM99], et Reeves [Ree83] à la modélisation de fumées et de flammes.

Les lois modélisant les interactions entre particules sont en général inspirées des forces de Lennard-Jones, qui régissent les phénomènes d'attraction / répulsion entre atomes ou molécules. Comme on peut le voir sur la figure 2.13, cette attraction présente une distance d'équilibre d où la force s'annule.

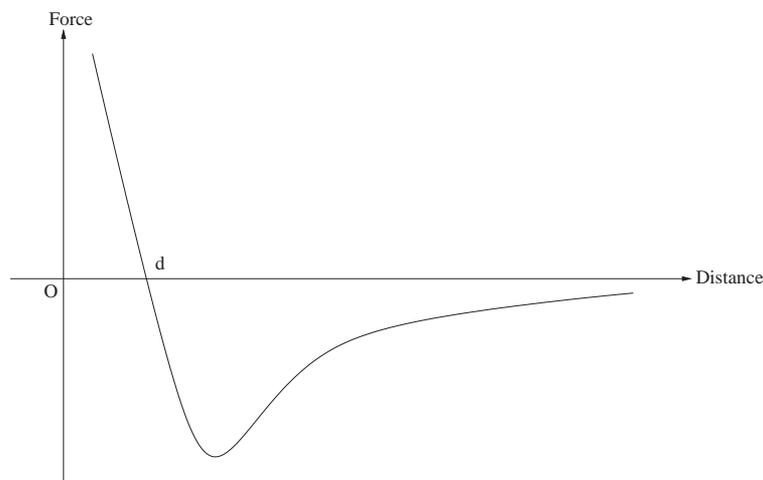


FIG. 2.13 – Intensité de la force de Lennard-Jones en fonction de la distance

On peut aussi noter qu'au delà d'une certaine distance, la force tend à diminuer jusqu'à s'annuler : les deux particules sont alors trop éloignées pour interférer entre elles. Cette particularité peut être employée afin de diminuer la quantité de calculs nécessaire. En effet, si chaque particule interagit avec toutes les autres, la complexité de la méthode varie en fonction du carré du nombre de particules ($O(n^2)$). Marie-Paul Cani et M. Desbrun utilisent ce principe pour limiter le nombre de calculs nécessaires en imposant un rayon d'action maximal aux particules [DC95].

Les lois d'interaction peuvent bien entendu être modifiées à volonté : dans les travaux de Jean-Christophe Lombardo, les particules sont orientées, c'est à dire qu'elles sont constituées d'un couple position, normale [Lom96]. Les interactions entre particules ne sont plus alors limitées à des forces, mais peuvent maintenant aussi être des couples. Les interactions utilisées sont au nombre de cinq :

- **Interaction de cocircularité** : qui tend à positionner les particules sur une même sphère,
- **Interaction de coplanarité** : qui tend à positionner les particules sur un même plan,
- **Interaction de colinéarité** : qui tend à positionner les particules sur une même

ligne,

- **Interaction de frottement** : qui permet d'introduire une atténuation des oscillations,
- **Interaction de cohésion** : assez comparable au potentiel de Lennard-Jones.

Chaque particule évolue donc non seulement en fonction de la position de ses voisines, mais aussi en fonction de l'orientation de celles-ci.

L'affichage et la polygonisation de la surface du corps est plus évident avec les systèmes masses-ressorts, dont la topologie est fixe, qu'avec les systèmes de particules. Bien souvent, ceux-ci sont utilisés conjointement à des surfaces implicites, ce qui donne des résultats très bons, mais coûteux en temps de calcul.

En conclusion, l'intérêt des systèmes à particules est bien sûr leur facilité d'implémentation. Grâce aux phénomènes d'émergences, des lois simples permettent d'obtenir des phénomènes complexes. Cependant, on peut se demander si l'application de lois microscopiques à des entités macroscopiques ne constitue pas une simplification dangereuse. Il est souvent difficile, voire impossible, de prouver la validité physique des modèles obtenus.

2.3.3.3 Les modèles éléments finis

La théorie de l'élasticité, parce qu'elle relie simplement les contraintes aux déformations, permet de modéliser facilement et précisément toute une gamme de corps déformables. Les modèles élastiques et visco-élastiques permettent, par exemple, de décrire avec un certain réalisme les comportements de tissus biologiques et de certains organes : pour n'en citer que quelques-uns, on pourra se reporter à [Deb00] ou [Pic01].

Les éléments finis (ou FEM) sont la méthode de prédilection utilisée en physique ou en ingénierie : la littérature les concernant est abondante, et les modèles développés bien souvent beaucoup trop précis pour les besoins de l'infographie. Les calculs impliqués dans les méthodes FEM sont de plus assez lourds, ce qui les a rendus pendant longtemps peu utilisés dans ce domaine.

Le principe des méthodes FEM repose sur la recherche d'une fonction continue par une approximation discrète. On découpe ainsi le corps simulé en cellules élémentaires, comportant chacune un certain nombre de variables qui décriront l'allure de la fonction à l'intérieur de la cellule. La résolution d'un problème par la méthode des éléments finis se décompose selon les étapes suivantes :

1. **On divise le domaine de calcul Ω en sous-domaines Ω_i** : c'est le maillage. Les sous-domaines doivent vérifier :

$$\bigcup_{i=0}^n \Omega_i = \Omega \quad (2.30)$$

et

$$\Omega_i \cap \Omega_j = \emptyset, \forall i \neq j \quad (2.31)$$

2. **Sur chacun des domaines, on définit des nœuds**, qui sont les points sur lesquels le problème sera évalué. En général, on choisit les sommet de la maille géométrique, mais ce n'est en aucun cas une obligation.
3. **On choisit ensuite les fonctions de formes du sous-domaine**, qui sont les fonctions permettant de définir le champs local à partir des valeurs obtenues aux nœuds. Pour des raisons de complexité, ces fonctions sont en général des polynômes.
4. **On revient ensuite à un problème discret** : Maintenant que les fonctions de formes sont définies, résoudre le problème complet revient à résoudre le problème sur les nœuds choisis.
5. **On résout le problème discret.**
6. **La solution approchée est enfin construite à partir de la solution discrète** : c'est le post-traitement.

Les méthodes FEM présentent deux points faibles. Le premier réside dans l'utilisation d'un maillage : en cas de déformation topologique trop importante, un remaillage du corps est nécessaire, ce qui implique une charge de calcul très lourde. Le second problème est l'importance des calculs mis en jeu à chaque résolution.

Les problèmes d'élasticité se ramènent en effet à la résolution d'un système de la forme :

$$M \frac{\partial^2 U}{\partial t^2} + D \frac{\partial U}{\partial t} + K U = F \quad (2.32)$$

où K est la matrice de rigidité, M et D des matrices liées à la masse et aux coefficients d'amortissement, U le vecteur des déplacements et F le vecteur des forces extérieures.

On considère généralement que les matrices M , D et K sont constantes, ce qui n'est pas tout à fait vrai puisque certains termes peuvent dépendre de la géométrie des éléments. Faire une telle hypothèse revient donc à supposer les déformations petites devant la taille du corps.

Une approche intéressante est développée dans [DDCB01][DC99]. Ces travaux portent sur une méthode FEM à échantillonnage spatial et temporel adaptatif et qui permet d'obtenir la manipulation en temps réel d'objets déformables avec un mesh assez fin.

2.3.4 La modélisation des écoulements

Comme nous l'avons vu dans le paragraphe précédent, les déformations des matériaux viscoplastiques tels que la glaise ou la plasticine sont basées sur des écoulements : la déformation est définitive et les changements de topologie en général assez importants. Nous ne saurions donc conclure cet état de l'art sans nous intéresser de manière privilégiée à la simulation physique de fluides régis par des équations de comportement de type Navier-Stokes.

2.3.4.1 Le modèle Lattice Gaz

Le Modèle HPP

Le principe du modèle Lattice Gaz, dit aussi "Gaz sur réseau", est de simuler un ensemble de particules pouvant entrer en collision et échanger leurs moments. Le premier modèle date de 1973 : il fut introduit par Hardy, de Pazzis et Pomeau [HPdP73], d'où son appellation de "modèle HPP".

Ces systèmes étant, comme nous allons le voir, entièrement discrets, ils ont l'avantage de pouvoir être implémentés de façon parallèle et permettent des simulations avec beaucoup plus de particules que les systèmes plus "réalistes" qui tiennent compte de manière continue (avec une précision finie) des positions, vitesses et potentiels d'interactions.

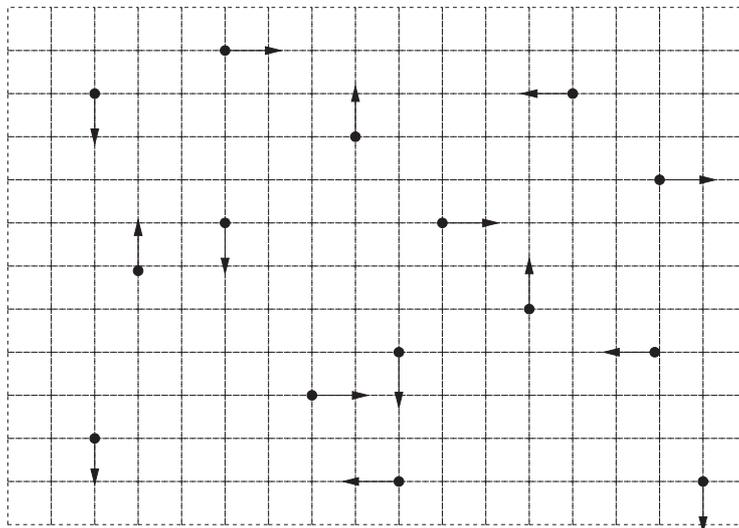


FIG. 2.14 – Le modèle HPP, basé sur une lattice à symétrie carrée

Le modèle HPP est basé sur un réseau à symétrie carrée, dont les nœuds sont les seules positions accessibles aux particules. Les interactions entre particules sont simples : il ne peut y en avoir qu'entre celles se trouvant sur le même nœud et seulement sous forme de collisions locales et instantanées. Les règles qui gouvernent ces collisions sont telles qu'elles conservent le nombre de particules, donc la masse, et l'impulsion.

L'évolution du système d'un pas de temps à l'autre se fait en deux étapes : l'une de propagation, au cours de laquelle les particules se déplacent dans la direction de leur vecteur d'impulsion jusqu'au nœud immédiatement voisin, l'autre d'interaction, pendant laquelle les particules ont la possibilité d'échanger de l'impulsion. Les applications du modèle HPP sont toutefois assez limitées. Le réseau sous-jacent n'ayant que quatre directions, les équations de Navier-Stokes qu'il engendre contiennent des termes dissipatifs anisotropes.

Le Modèle FHP

On peut démontrer qu'un réseau à symétrie hexagonale ne souffre pas de ce manque d'anisotropie. Le modèle FHP, construit par Frisch, Hasslacher et Pomeau [FHP86], exploite le même principe, mais en se basant cette fois sur une lattice offrant six directions.

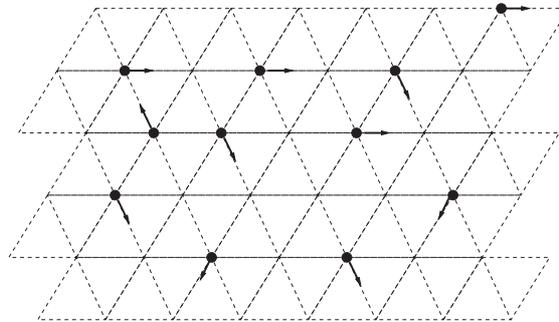


FIG. 2.15 – Le modèle FHP, basé sur une lattice à symétrie hexagonale

Stephen Wolfram, dans [Wol86], démontre que l'équation macroscopique de comportement d'un réseau FHP est similaire à une équation de Navier-Stokes. Les termes convectifs et visqueux sont présents, et ont leur structure habituelle. Il y a, cependant, quelques termes additionnels qui proviennent de la discrétisation spatiale des trajectoires.

Conclusion

Les modèles de type "Lattice Gaz" sont en fait des systèmes à particules, fonctionnant de manière similaire à ceux présentés dans la section précédente, auquel on a adjoint une structure de lattice. Simples à implémenter, les corps modélisés obéissent à des lois de comportement comparables au comportement de fluides newtoniens incompressibles.

Ces modèles, véritables curiosités scientifiques, sont cependant inutilisés en infographie en raison du nombre de particules nécessaires pour obtenir une simulation valable. Ils sont beaucoup plus adaptés à l'étude de certaines propriétés des corps gazeux (étude de l'ergodicité, etc) qu'à la simulation de corps visqueux.

Le problème est avant tout un problème d'échelle : les phénomènes modélisés sont à l'échelle microscopique, ce qui implique que la taille de la particule doit être négligeable devant les longueurs caractéristiques afin que la modélisation ait un sens physique.

2.3.4.2 Le modèle "Height Field"

Utiliser les équations de Navier-Stokes afin de simuler un écoulement n'est pas chose aisée, en partie en raison du couplage champs de vitesse / champ de pression qui implique une résolution assez complexe. Le modèle Height Field permet de résoudre ce problème en simplifiant l'équation de pression grâce à l'utilisation d'une expression hydrostatique.

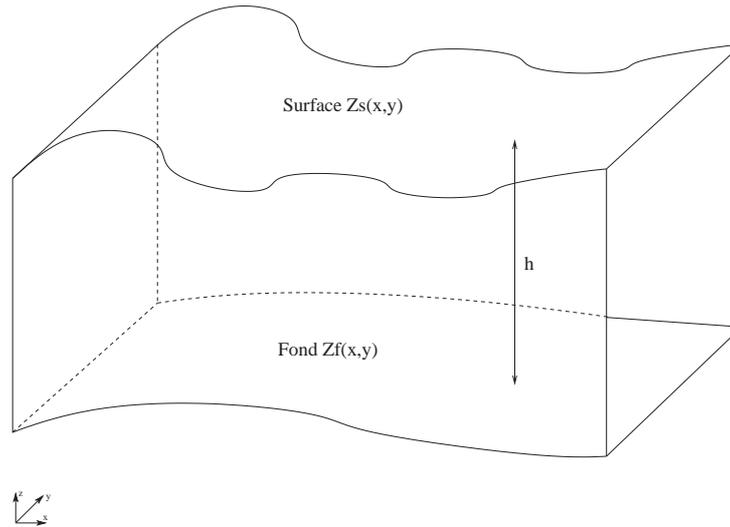


FIG. 2.16 – Le fluide, limité en haut par la surface et en bas par le fond

On considère le système représenté sur la figure 2.16. Le fluide modélisé est délimité en hauteur par la surface de côte Z_S et le fond de côte Z_F .

L'approximation utilisée consiste à considérer que l'accélération due à la pression équilibre la gravité. On dit que la pression est *hydrostatique*. Cette écriture suppose, bien entendu, de faibles mouvements dans le sens vertical. Cela permet d'exprimer la pression en terme de hauteur de liquide.

La pression $p(x, y, z, t)$ n'étant due qu'au poids de la colonne de fluide située au dessus du point de coordonnée (x, y, z) à l'instant t , on a :

$$-\frac{1}{\rho} \frac{\partial p}{\partial z} - g = 0 \quad (2.33)$$

soit :

$$p(x, y, z, t) = -\rho g z(t) + Cste \quad (2.34)$$

La constante est choisie de telle manière qu'on ait $p(x, y, Z_S) = p_{atm}$, c'est à dire que la pression en surface du liquide est la pression atmosphérique. Pour simplifier, on écrit souvent :

$$p(x, y, z) = \rho g (Z_s - z) \quad (2.35)$$

Les équations de Navier-Stokes vont maintenant être moyennées sur la verticale, par intégration depuis le fond jusqu'à la surface. Si h est la hauteur de fluide, on peut ainsi écrire deux nouvelles variables :

$$u = \frac{1}{h} \int_{Z_F}^{Z_S} U_x dz \quad \text{et} \quad v = \frac{1}{h} \int_{Z_F}^{Z_S} U_y dz \quad (2.36)$$

Ces deux moyennes sur la verticale des composantes du vecteur vitesse tridimensionnel seront elles-mêmes les composantes de la vitesses \vec{u} dans la suite. Cette nouvelle vitesse correspond à une restriction en deux dimensions du problème 3D initial.

L'équation de continuité, intégrée entre Z_F et Z_S devient :

$$\int_{Z_F}^{Z_S} \left(\frac{\partial U_x}{\partial x} + \frac{\partial U_y}{\partial y} + \frac{\partial U_z}{\partial z} \right) dz = 0 \quad (2.37)$$

En utilisant la règle de Leibnitz, et compte tenu des conditions d'imperméabilité du fond et de la surface, cette équation devient :

$$\boxed{\frac{\partial h}{\partial t} + \text{div}(h \vec{u}) = 0} \quad (2.38)$$

Le problème de l'écoulement 3D se ramène grâce à cette équation à un problème d'écoulement 2D, la conservation de la masse n'étant plus assurée par la pression mais par la hauteur de liquide définie en tout point du plan $[x, y]$.

Les méthodes utilisant cette écriture sont nombreuses et utilisent des principes très variées. Le projet Telemac, lancé en 1987 par EDF afin de modéliser les écoulements à surfaces libres, en fait partie. Dans ce travail, l'écoulement est modélisé en deux dimensions grâce aux équations de Saint-Venant (équivalent des équations de Navier-Stokes, mais portant sur \vec{u}) et on utilise l'équation (2.38) afin de calculer les hauteurs d'eau en chaque point [Her03].

Rafal Wcislo, quant à lui, utilise une description voxel de l'espace de travail pour modéliser un liquide dans lequel flottent plusieurs objets [WKM98]. La modélisation utilise des automates cellulaires pour calculer l'état de chaque voxel. Deux jeux de règles sont utilisés pour définir l'automate :

- Une règle "Verticale", dont le principe est assez proche de l'équation (2.38), assure la conservation de la quantité de liquide.
- Une règle "Horizontale" est utilisée pour aplanir la surface du liquide.

Avolio, en 2002, présente une approche similaire pour la simulation de coulées pyroclastiques lors d'éruptions volcaniques grâce à un modèle basé lui aussi sur les Automates Cellulaires [CRG+02].

En conclusion, Les modèles de type Height Field sont beaucoup plus aisés à manipuler que les classiques équations de Navier-Stokes : ils autorisent l'utilisation d'outils mathématiques simples et fonctionnent le plus souvent en temps réel. Cependant, cette classe de méthodes ne se prête qu'à la modélisation de corps assimilables à des étendues d'eau, c'est à dire ayant une géométrie essentiellement bidimensionnelle.

2.3.4.3 Le travail de Foster et Metaxas

Nick Foster et Dimitri Metaxas, en 1996, se penchent sur la modélisation de liquides tels que l'eau pour l'infographie [FM96]. Comme le montre la figure 2.17, leur modèle repose sur une description voxelisée de l'espace de calcul.

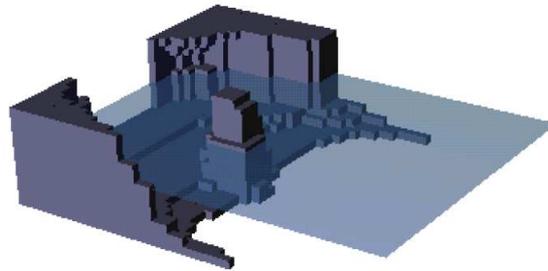


FIG. 2.17 – Un exemple d'espace de calcul voxelisé

L'espace de travail est ainsi divisé en cellules de tailles égales, ou voxels, qui peuvent être dans chacun des quatre états suivants :

- **Plein**, c'est à dire rempli de particules de fluide
- **Vide**,
- **Surface**, pour un voxel situé sur la surface séparant le liquide de l'atmosphère
- **Obstacle**, pour un voxel appartenant à un obstacle rigide (fond, mur, etc.)

Dans tous les cas, un champ de vitesse et un champ de pression sont définis dans chaque cellule. Ces deux champs obéissent aux lois de Navier-Stokes pour un liquide Newtonien incompressible :

$$\begin{aligned}
 \frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} + u_z \frac{\partial u_x}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \Delta(u_x) \\
 \frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_y}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \Delta(u_y) \\
 \frac{\partial u_z}{\partial t} + u_x \frac{\partial u_z}{\partial x} + u_y \frac{\partial u_z}{\partial y} + u_z \frac{\partial u_z}{\partial z} &= -\frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \Delta(u_z) + g
 \end{aligned}
 \tag{2.39}$$

et

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \quad (2.40)$$

Les équations de Navier-Stokes sont résolues sur l'environnement complet. Les obstacles solides et l'atmosphère sont traités de la même manière que le fluide, mais en utilisant des valeurs qui restent contraintes durant la résolution numérique.

Pour chacun des voxels, Foster et Metaxas calculent les composantes du champs de vitesse suivant l'arrangement présenté en figure 2.18.

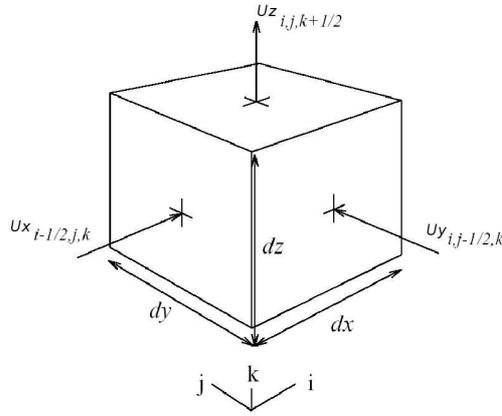


FIG. 2.18 – Arrangement des variables pour une cellule typique ([FM96])

Les pressions, quant à elles, sont calculées au centre de chacun des voxels. La discrétisation de l'équation des moments (2.39) à l'aide d'un schéma explicite donne pour la composante U_x :

$$\begin{aligned} (U_x)_{i+1/2,j,k}^{t+1} = & (U_x)_{i+1/2,j,k} + \delta t \{ (1/\delta x)((U_x)_{i,j,k}^2 - (U_x)_{i+1,j,k}^2) \\ & + (1/\delta y)((U_x \cdot U_y)_{i+1/2,j-1/2,k} - (U_x \cdot U_y)_{i+1/2,j+1/2,k}) \\ & + (1/\delta z)((U_x \cdot U_z)_{i+1/2,j,k-1/2} - (U_x \cdot U_z)_{i+1/2,j,k+1/2}) \\ & + (1/\delta x)(p_{i,j,k} - p_{i+1,j,k}) \\ & + (\nu/\delta x^2)((U_x)_{i+3/2,j,k} - 2(U_x)_{i+1/2,j,k} + (U_x)_{i-1/2,j,k}) \\ & + (\nu/\delta y^2)((U_x)_{i+1/2,j+1,k} - 2(U_x)_{i+1/2,j,k} + (U_x)_{i+1/2,j-1,k}) \\ & + (\nu/\delta z^2)((U_x)_{i+1/2,j,k+1} - 2(U_x)_{i+1/2,j,k} + (U_x)_{i+1/2,j,k-1}) \} \end{aligned} \quad (2.41)$$

On obtient, grâce à cette relation, U_x^{t+1} en fonction de U_x^t (noté U_x) et d'autres termes, tous pris à l'instant t . Le calcul est donc direct.

Pour obtenir la pression, Foster et Metaxas ne se sont pas basés sur la résolution de l'équation de Poisson habituelle, mais sur la quantité $D_{i,j,k}$:

$$\begin{aligned} D_{i,j,k} = & (1/\delta x)((U_x)_{i+1/2,j,k} - (U_x)_{i-1/2,j,k}) \\ & + (1/\delta y)((U_y)_{i,j+1/2,k} - (U_y)_{i,j-1/2,k}) \\ & + (1/\delta z)((U_z)_{i,j,k+1/2} - (U_z)_{i,j,k-1/2}) \end{aligned} \quad (2.42)$$

$D_{i,j,k}$ correspond à la divergence de fluide . Ainsi, un $D_{i,j,k}$ négatif correspond à un influx de fluide donc à une augmentation de pression, et un $D_{i,j,k}$ positif à une perte de fluide, donc à une baisse de pression. Le champ de pression est ainsi mis à jour grâce à la relation :

$$\delta p = -\frac{\beta_0}{2\delta t} \left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2} \right) D \quad (2.43)$$

où β_0 correspond à un coefficient de relaxation. Les résultats obtenus par cette méthode dans le cas liquide sont très intéressants et surtout très réalistes. Les quelques images présentées figure 2.19 donneront au lecteur une idée de la qualité du modèle.

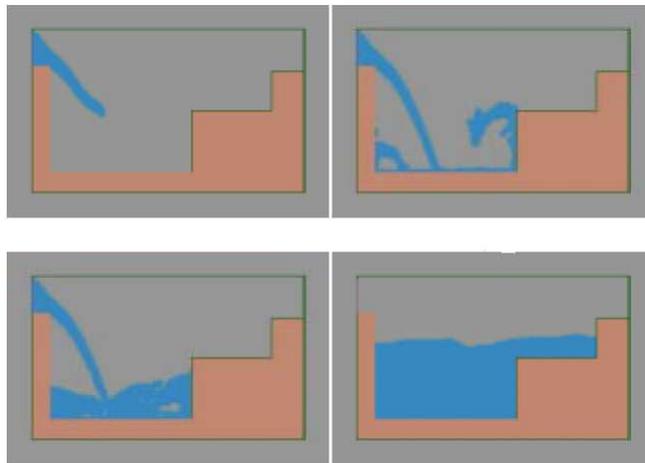


FIG. 2.19 – Résultats de simulation dans le cas 2D

Le modèle utilisé par Foster et Metaxas, cependant, souffre d'une faiblesse : le schéma de différenciation explicite utilisé. En effet, ce schéma peut diverger pour des viscosités trop importantes ou bien encore pour des vitesses de fluide trop élevées.

Différents travaux ont été menés par la suite afin de stabiliser ce calcul : utilisation de différenciation par la méthode de Crank Nicholson, la méthode des pas fractionnaires, ou encore par l'ADI [Sta03]. Cependant, la stabilité de ces méthodes se paie par le temps de calcul nécessaire, celles-ci requérant en général une inversion de matrice assez importante ou bien plusieurs itérations.

2.4 Conclusion

Les différents modèles utilisés afin de reproduire le comportement de solides en déformations sont très variés, mais ne reposent en général pas sur une description du phénomène physique de viscoplasticité. Basé sur des critères géométriques ou sur des équations reposant sur l'élasticité linéaire, ils ne sont pas adaptés à nos besoins.

Les modèles de fluides développés par Foster et Metaxas, par contre, sont totalement basés sur les équations de Navier-Stokes, mais ils souffrent en général de problèmes concernant le temps de calcul. Conçus dans le cadre d'une application cinématographique, ils utilisent une représentation volumique et des méthodes numériques assez lourdes.

Chapitre 3

Approches volumiques de la déformation du modèle matière

3.1 Introduction

Comme nous venons de le voir dans le chapitre 2, les modèles de corps déformables utilisés en infographie sont le plus souvent basés sur les lois de l'élasticité linéaire ou sur des principes géométriques et intuitifs. Ces approches permettent souvent d'obtenir des résultats intéressants, mais souffrent parfois d'un manque de réalisme, notamment dans le cas de déformations impliquant des changements topologiques importants. La plupart de ces modèles, par exemple, ne garantissent pas la conservation du volume durant et après la déformation.

Nous avons présenté, dans le même chapitre, les équations régissant les déformations des corps viscoplastiques tels que la glaise ou la plasticine. Elles sont au nombre de deux : l'équation des moments et l'équation de conservation de la masse. Notre objectif dans ce chapitre est la synthèse d'un modèle permettant d'approcher au mieux ces deux équations, afin d'obtenir un résultat très proche du comportement réel, tout en respectant les contraintes liées à l'interactivité avec l'utilisateur.

Ce chapitre se divise en trois parties. Dans la première, nous verrons que les différents termes de l'équation des moments n'ont pas la même importance suivant le type d'écoulement considéré. Ainsi, une simplification connue sous le nom "d'hypothèse des écoulements rampants" peut être envisagée dans le cas où les viscosités sont très importantes. Cette hypothèse est très utile, aussi nous l'étudierons en détail et vérifierons sa validité dans le cadre de notre travail.

La seconde partie du chapitre est dédiée à une modélisation sans viscosité, intégrant uniquement la conservation de la masse. Deux modèles y sont développés : l'un basé sur une approche volumique et l'utilisation d'automates cellulaires, l'autre sur une approche ne prenant en compte que la surface du corps. Enfin, dans la dernière partie, nous présenterons un dernier modèle complet, intégrant à la fois viscosité et conservation de la masse.

3.2 Hypothèse des écoulements rampants

Introduction

Les équations de Navier-Stokes, présentées au second chapitre et rappelées en 3.1, permettent de modéliser les écoulements de fluides.

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{\text{grad}} p + \nu \Delta \vec{u} - (\vec{u} \cdot \overrightarrow{\text{grad}}) \vec{u} + \vec{F} \\ \text{div}(\vec{u}) = 0 \end{cases} \quad (3.1)$$

où :

- \vec{u} est le champ de vitesse dans le fluide,
- p est la pression,
- ρ est la masse volumique (supposée constante),
- ν la viscosité cinématique.

Ces équations sont valables pour les fluides au sens large du terme : elles permettent de modéliser des corps aux comportements aussi variés que les liquides, les gaz, ou les pâtes. Cette grande variété peut nous amener à penser que tous les termes de l'équation ne jouent pas forcément un rôle majeur dans le comportement macroscopique, et que peut-être certains peuvent être négligés.

Les fluides auxquels nous nous intéressons sont très particuliers : il s'agit de corps dont la viscosité est très élevée et où l'inertie semble négligeable. Une hypothèse très utilisée en mécanique des fluides semble s'adapter parfaitement à notre cas : l'hypothèse des écoulements rampants.

Hypothèse des écoulements rampants

Pour comprendre l'hypothèse des écoulements rampants, il faut tout d'abord comprendre le rôle joué par chacun des termes de l'équation des moments. Comme on peut le voir sur l'équation 3.1, l'évolution du champ de vitesse \vec{u} dépend de quatre termes :

- $\overrightarrow{\text{grad}} p$: il s'agit là du terme de pression. Le champ de vitesse déplace la matière depuis les zones de haute pression vers les zones de basse pression. Comme nous l'avons vu dans le chapitre 2, c'est le terme qui permet de relier les deux équations de Navier-Stokes et d'assurer $\text{div}(\vec{u}) = 0$.
- $\nu \Delta \vec{u}$ représente le frottement entre les différentes couches de fluides : c'est le terme de viscosité. Grâce à ce terme, une particule de fluide en déplacement "entraîne" ses voisines avec elle.

- $(\vec{u} \cdot \overrightarrow{\text{grad}}) \vec{u}$ est un terme convectif. D'un point de vue physique, il correspond à la projection du gradient de vitesse sur la direction locale de l'écoulement : c'est donc un terme inertiel dans cette équation.
- \vec{F} représente l'effet des forces appliquées au corps.

Lorsque la vitesse d'écoulement est très faible ou lorsque le fluide est très visqueux (c'est-à-dire lorsque le nombre de Reynolds est très petit), le terme inertiel de l'équation de Navier-Stokes joue un rôle mineur et peut être négligé. On est dans le cadre de l'hypothèse des écoulements rampants ("Creeping flows"). L'équation des moments devient alors linéaire : c'est l'équation de Stokes.

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{\text{grad}} p + \nu \Delta \vec{u} + \vec{F} \\ \text{div}(\vec{u}) = 0 \end{cases} \quad (3.2)$$

Le terme \vec{F} , qui représente les forces extérieures, est le plus souvent assimilé à une pression hydrostatique constante. On peut l'écrire alors :

$$\vec{F} = -\frac{1}{\rho} \overrightarrow{\text{grad}}(p_0) \quad (3.3)$$

Avec p_0 la correction à apporter au champ de pression p afin d'intégrer ces forces.

L'équation de Stokes devient alors :

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{\text{grad}}(p - p_0) + \nu \Delta \vec{u} \\ \text{div}(\vec{u}) = 0 \end{cases} \quad (3.4)$$

Cette notion de seuil de pression p_0 est à rapprocher de la notion de seuil associée aux fluides de Bingham. Le fluide ne s'écoule que si la contrainte dépasse un certain seuil. Pour la suite, on notera :

$$P = p - p_0 \quad (3.5)$$

et l'expression finale de l'équation de Stokes est :

$$\boxed{\begin{cases} \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{\text{grad}}(P) + \nu \Delta \vec{u} \\ \text{div}(\vec{u}) = 0 \end{cases}} \quad (3.6)$$

L'avantage de cette équation est qu'elle est linéaire : elle est beaucoup plus simple à traiter que l'équation des moments habituelle et représente une base idéale pour nos modélisations. Cependant, avant de pouvoir l'utiliser, nous devons vérifier que nous sommes bien dans le cadre des hypothèses, à savoir faibles vitesses d'écoulement ou très grande viscosité.

Etude de la validité de l'hypothèse

Manipuler de la pâte à modeler ou bien de la plasticine suffit à se convaincre que l'inertie du matériau joue un rôle négligeable dans l'écoulement. En effet, le mouvement de la matière ne se prolonge pas après la fin de l'action de l'outil, et ce quelque soit le mouvement considéré. Cette intuition physique peut être confirmée par l'étude du nombre de Reynolds pour le système considéré. Le nombre de Reynolds Re est une grandeur utilisée en mécanique des fluides afin de comparer les effets inertiels aux effets de viscosité. Si le nombre de Reynolds est petit devant l'unité, alors nous sommes bien dans le cadre de l'hypothèse des écoulements rampants et l'équation de Stokes est valide. Si au contraire il est grand devant 1, alors nous ne pouvons pas négliger le terme inertiel et l'équation des moments doit être utilisée. L'expression du nombre de Reynolds est la suivante :

$$Re = \frac{u d}{\nu} \quad (3.7)$$

Où u représente la vitesse du fluide, ν la viscosité cinématique, et d une longueur caractéristique du système considéré (diamètre ou longueur d'une conduite, taille du corps, etc). Le nombre de Reynolds Re est une grandeur adimensionnelle.

La viscosité n'est pas mesurable facilement sans un appareillage très spécialisé. Nous nous sommes donc reporté sur la littérature (cf Annexe A) afin d'en obtenir une valeur. Les viscosités cinématiques obtenues sont de l'ordre de 10^7 poises pour la plasticine souple et jusqu'à 10^{12} poises pour la plasticine dure. Nous prendrons 10^7 poise pour évaluer le nombre de Reynolds.

Afin d'estimer l'ordre de grandeur des vitesses mises en jeu, nous avons effectué une mesure des vitesses de la main durant une activité de modelage virtuel. Le sujet manipule à l'aide d'un bras haptique à six degrés de liberté un corps virtuel afin de le modeler. Aucun retour haptique n'a été appliqué durant les mesures, le bras ne servant que comme capteur de position. Les mesures de vitesses sont effectuées toutes les 20 millisecondes durant 16 secondes. La figure 3.1 montre sous la forme d'un histogramme de fréquence la répartition de ces vitesses.

Comme on peut le voir, les vitesses mises en jeu sont en moyenne de 8 centimètres par seconde. Le maximum enregistré, non représenté sur cet histogramme pour des raisons de lisibilité, a été de 2 m/s. Il s'agit d'une mesure unique, très éloignée des autres. Nous allons l'utiliser comme valeur de référence afin d'estimer le nombre de Reynolds dans le pire cas.

La distance caractéristique considérée, quant à elle, sera la taille de notre environnement de simulation, soit 15 à 30 cm. Pour ces valeurs, nous obtenons un nombre de Reynolds de :

$$Re = 6.10^{-8} \ll 1 \quad (3.8)$$

L'hypothèse des écoulements rampants est donc tout à fait justifiée dans notre cas

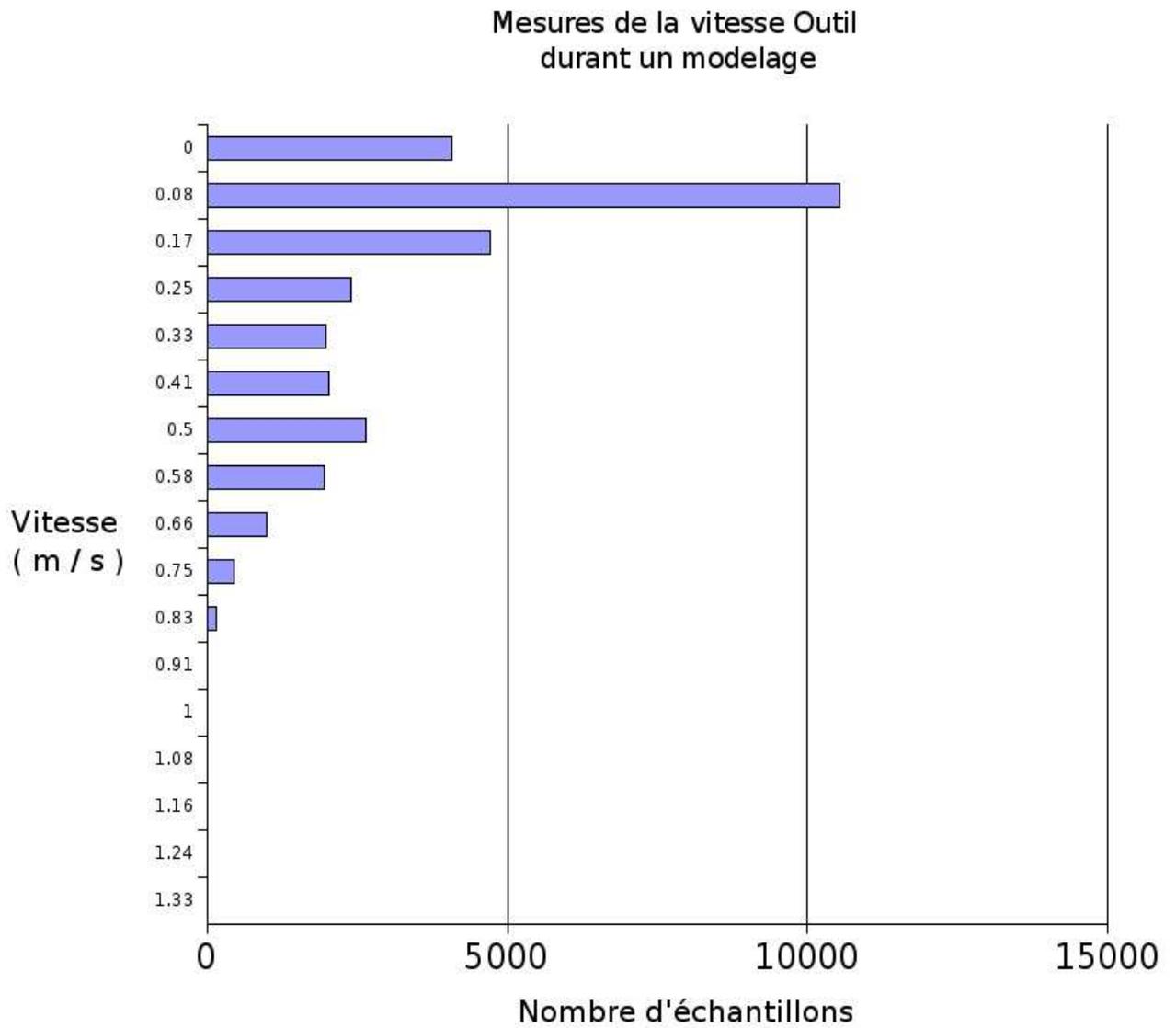


FIG. 3.1 – Histogramme de répartition des vitesses durant un modelage

3.3 Modèles basés sur la conservation de la masse

3.3.1 Introduction

Les écoulements à petit nombre de Reynolds, comme c'est le cas pour notre modèle, peuvent donc se résumer à la simulation de l'équation de Stokes :

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{\text{grad}}(P) + \nu \Delta \vec{u} \\ \text{div}(\vec{u}) = 0 \end{cases} \quad (3.9)$$

On peut observer que deux termes seulement sont à la base de la déformation : un terme de conservation du volume ($\text{grad}(P)$) et un terme de viscosité ($\nu \Delta \vec{u}$).

Nous allons dans un premier temps nous intéresser à la conservation du volume et au rôle que celle-ci joue dans la déformation globale. Pour se faire, nous nous plaçons dans le cadre de la simulation d'un corps "parfait" dont la viscosité est nulle :

$$\boxed{\nu = 0} \quad (3.10)$$

Deux modèles ont été développés afin de simuler ce corps parfait. Le premier, basé sur une relaxation de l'hypothèse d'incompressibilité, nous a permis d'exprimer le comportement sous la forme d'un automate cellulaire. Le second modèle, quant à lui, utilise une approche orientée surface de la déformation : on ne traite plus le volume complet mais uniquement sa périphérie.

3.3.2 Approche volumique par Automates Cellulaires

Les automates cellulaires sont un formalisme intéressant pour atteindre nos objectifs. Très faciles à implémenter, ils permettent de simuler des phénomènes macroscopiques complexes à l'aide de lois locales extrêmement simples. Nous allons voir comment nous pouvons les utiliser afin de mettre au point un premier modèle de déformation basé sur la conservation de la masse.

3.3.2.1 Le modèle matière

Le modèle matière repose sur l'utilisation d'un fluide de viscosité nulle. Etant donné que les vitesses d'écoulement sont supposées très petites, nous pouvons nous placer dans le cadre des écoulements rampants. L'équation des moments devient alors :

$$\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{\text{grad}} P \quad (3.11)$$

Pour cette simulation, nous avons choisi une écriture du champ de pression sous la forme :

$$P = -\lambda \text{div}(\vec{u}) \quad (3.12)$$

Dans cette équation, la pression est directement reliée à l'influx de matière. Si la divergence du champ de vitesse est fortement positive, cela signifie que de la matière est en

train de s'accumuler localement et la pression augmente en conséquence. Si au contraire la divergence est négative, cela veut dire que l'on a une perte locale de matière. La pression négative qui s'en suit crée un "appel" provoquant la compensation de cette perte. Cette écriture est utilisée par Foster et Metaxas pour la modélisation de liquides en synthèse d'image [FM96].

Cette écriture de la pression n'est pas rigoureusement exacte : elle n'assure pas, a priori, que la conservation de la quantité de matière soit rigoureusement respectée. Nous verrons par la suite qu'il s'agit en fait d'un passage à la limite, et que l'automate cellulaire qui en résulte respecte bien la conservation de matière.

En prenant la divergence de l'équation 3.11, on obtient :

$$\operatorname{div}\left(\frac{\partial \vec{u}}{\partial t}\right) = \operatorname{div}\left(-\frac{1}{\rho} \overrightarrow{\operatorname{grad} P}\right) \quad (3.13)$$

Soit encore, grâce au théorème de Schwarz :

$$\frac{\partial \operatorname{div}(\vec{u})}{\partial t} = -\frac{1}{\rho} \Delta P \quad (3.14)$$

En remplaçant $\operatorname{div}(\vec{u})$ grâce à l'équation 3.12, on obtient ainsi :

$$\boxed{\frac{\partial P}{\partial t} = \frac{\lambda}{\rho} \Delta P} \quad (3.15)$$

La forme de l'équation 3.15 correspond à une équation de diffusion. **Le phénomène modélisé peut donc être ramené à un processus de répartition de la surpression au sein du corps** jusqu'à obtenir un équilibre.

Maintenant qu'une loi de comportement est établie pour le matériau, nous devons nous pencher sur la discrétisation de cette loi et sur les aspects numériques du problème.

3.3.2.2 Discrétisation de l'équation de comportement

Le choix de la représentation est un choix crucial pour la modélisation. Nous avons choisi une représentation explicite du corps à simuler, en découpant l'espace de travail en **voxels**. Un voxel est un élément de volume de forme cubique. Il est au volume ce que le pixel est à l'image. L'utilisation des voxels se prête particulièrement bien à des résolutions numériques utilisant des techniques de différences finies. On peut, par exemple, associer à chaque voxel une quantité de matière, une pression, une vitesse, etc.

L'interaction outil-matière s'en trouve, elle-aussi, facilitée : on peut en effet définir instantanément pour chaque voxel son appartenance ou non au corps modélisé. La détection de collisions est ici un problème simple alors qu'il peut devenir relativement complexe dans le cas d'une description surfacique des corps.

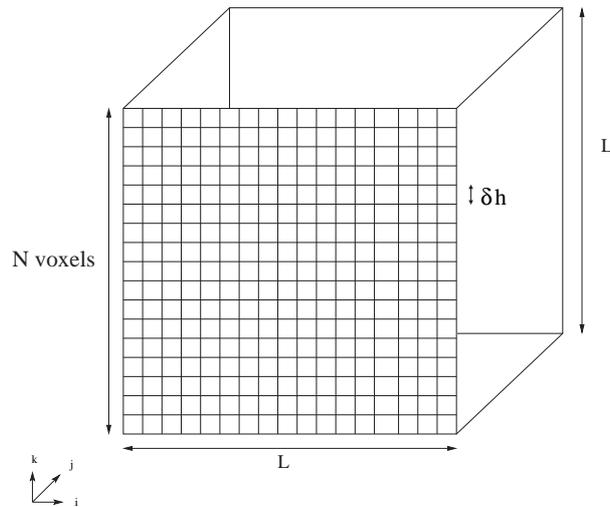


FIG. 3.2 – Discrétisation de l'espace de travail

Comme le montre la figure 3.2, on considère pour la suite un espace de travail de forme cubique et d'arête de longueur L . Chacune de ces arêtes est décomposée en N éléments, ce qui nous donne un pas de discrétisation spatiale défini par :

$$\delta h = \frac{L}{N} \quad (3.16)$$

Connaissant les coordonnées continues x, y, z d'un point, on peut ainsi directement savoir par la relation suivante à quel voxel (i, j, k) il correspond :

$$i = \lfloor \frac{x}{\delta h} \rfloor, j = \lfloor \frac{y}{\delta h} \rfloor, k = \lfloor \frac{z}{\delta h} \rfloor \quad (3.17)$$

avec $\lfloor \cdot \rfloor$ l'arrondissement à la valeur entière inférieure.

Pour la suite, cette structure voxel va servir de support à la discrétisation spatiale des équations de comportement. Ainsi, à chacun des voxels de coordonnées (i, j, k) sera associé à l'instant t :

- $P_{(i,j,k)}^t$ valeur du champ de pression P au centre du voxel
- ${}^x U_{(i,j,k)}^t$ représentation discrète de la composante suivant l'axe x du champ \vec{u}
- ${}^y U_{(i,j,k)}^t$ représentation discrète de la composante suivant l'axe y du champ \vec{u}
- ${}^z U_{(i,j,k)}^t$ représentation discrète de la composante suivant l'axe z du champ \vec{u}
- $Q_{(i,j,k)}^t$ quantité de matière contenue dans le voxel

Le placement spatial de ces grandeurs discrétisées est illustré sur la figure 3.3.

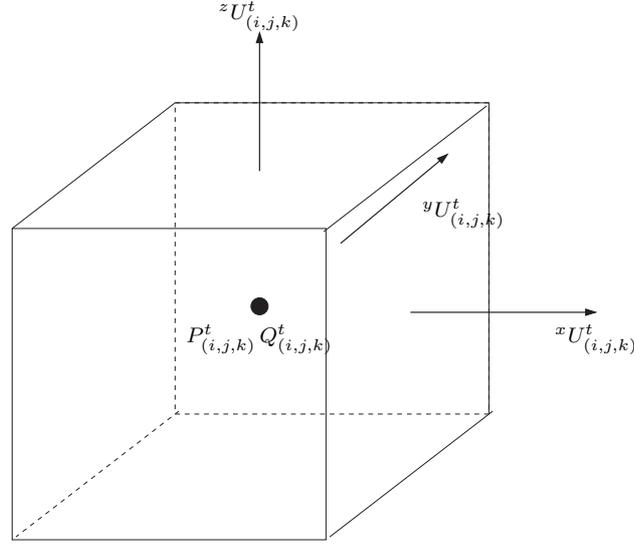


FIG. 3.3 – Discrétisation des valeurs sur le voxel

Ce choix de placement est inspiré de [Sta03]. Les quantités $Q_{(i,j,k)}^t$ et $P_{(i,j,k)}^t$ sont calculées au centre du voxel. Les composantes du champ de vitesses, quant à elles, sont calculées au centre des faces du cube comme indiqué sur la figure 3.3.

Grâce à ces notations, nous pouvons maintenant définir des opérateurs discrets. La divergence discrète sera ainsi définie comme :

$$\tilde{div}(u)_{(i,j,k)}^t = \frac{\left(xU_{(i,j,k)}^t - xU_{(i-1,j,k)}^t + yU_{(i,j,k)}^t - yU_{(i,j-1,k)}^t + zU_{(i,j,k)}^t - zU_{(i,j,k-1)}^t \right)}{\delta h}$$

Et les composantes du gradient discret de pression s'écrivent aussi au centre des faces comme :

$$\tilde{\nabla} P^t = \begin{pmatrix} x \tilde{\nabla} P_{(i,j,k)}^t \\ y \tilde{\nabla} P_{(i,j,k)}^t \\ z \tilde{\nabla} P_{(i,j,k)}^t \end{pmatrix} = \frac{1}{\delta h} \begin{pmatrix} P_{i+1,j,k}^t - P_{i,j,k}^t \\ P_{i,j+1,k}^t - P_{i,j,k}^t \\ P_{i,j,k+1}^t - P_{i,j,k}^t \end{pmatrix}$$

Nous disposons maintenant d'une écriture discrète des principaux opérateurs que nous allons utiliser. Voyons comment, à l'aide de ces opérateurs, un automate cellulaire va permettre de modéliser l'équation de diffusion de pression (3.15) qui est à la base de ce modèle.

3.3.2.3 Modélisation de diffusion par automates cellulaires

Les automates cellulaires sont connus depuis les tout débuts de l'informatique et font de plus en plus l'objet d'études pour la modélisation de phénomènes physiques (modélisation d'avalanches, d'écoulement de lave, etc.). Ils reposent sur le principe d'émergence et permettent, à l'aide de lois très simples exprimées au niveau microscopique, d'obtenir des comportements globaux complexes et réalistes.

Ces automates cellulaires nous intéressent pour deux raisons : tout d'abord, pour la facilité d'implémentation qui leur est associée. Ensuite, parce qu'ils permettent une parallélisation très facile des calculs et donc de tirer le meilleur parti des machines modernes. Après en avoir donnée une définition plus rigoureuse, nous allons nous intéresser à une famille spécifique d'automates cellulaires : les automates moyenneurs. Nous verrons que ces automates permettent de modéliser des comportements diffusifs et verrons comment mettre en place un tel automate dans le cas de l'équation 3.15.

Définition d'un automate cellulaire

Un automate cellulaire est défini par la donnée : d'un réseau régulier, d'un ensemble fini d'états utilisés pour colorier les sommets de ce réseau (les cellules) ainsi que d'une règle décrivant comment une cellule change d'état en fonction des états de ses voisines.

Afin de passer d'une configuration à la suivante, la règle, qui est locale et déterministe, est appliquée uniformément et de manière synchrone à toutes les cellules. Mathématiquement, un automate cellulaire est donc un quadruplet $(\mathcal{L}, \mathcal{N}, \mathcal{Q}, \mathcal{F})$ avec :

- \mathcal{L} la lattice servant de support à l'automate cellulaire. Chacun des nœuds de cette lattice portera un vecteur d'état le décrivant à chaque instant.
- \mathcal{Q} l'ensemble des états que peut prendre une cellule. Dans le cas le plus courant, il s'agit d'un ensemble fini et discret, mais on peut aussi considérer un espace d'états continu et infini.
- \mathcal{N} le voisinage d'une cellule. Ce voisinage est une partie finie ordonnée de \mathcal{L} .
- \mathcal{F} est une application de $\mathcal{Q}^{|\mathcal{N}|}$ dans \mathcal{Q} qui définit la *règle locale de transition*. Cette application permet de calculer le nouveau vecteur d'état d'une cellule en fonction des vecteurs d'états des cellules faisant partie de son voisinage.

Pour la suite de cette étude, nous prendrons comme lattice l'espace voxel servant à décrire l'espace de travail. Chaque voxel sera donc une machine à états finis dont les variables ($Q_{i,j,k}$, $P_{i,j,k}$, etc ...) évolueront grâce à des lois de transition appliquées sur un voisinage.

Automates cellulaires moyennés et équation de diffusion

Une classe particulière d'automates cellulaires est celle des automates moyennés, c'est-à-dire des automates dont chaque cellule effectue une moyenne sur son voisinage afin de déterminer son nouvel état. Ces automates particuliers ont fait l'objet d'études par Jörg Weimar qui a démontré qu'ils permettaient de modéliser des comportements diffusifs [Wei97][Wei02].

Considérons le cas monodimensionnel de l'équation de diffusion :

$$\frac{\partial x(r, t)}{\partial t} = \mathcal{D} \Delta x(r, t) \quad (3.18)$$

avec x la grandeur diffusée, et r la position. Cette équation est en général discrétisée sous la forme :

$$\frac{x(r, t + \delta t) - x(r, t)}{\delta t} = \mathcal{D} \frac{x(r - \delta r, t) - 2x(r, t) + x(r + \delta r, t)}{\delta r^2} \quad (3.19)$$

Ce qui peut être réécrit sous la forme :

$$x(r, t + \delta t) = \mathcal{D} \frac{\delta t}{\delta r^2} x(r - \delta r, t) + \left(1 - 2\mathcal{D} \frac{\delta t}{\delta r^2}\right) x(r, t) + \mathcal{D} \frac{\delta t}{\delta r^2} x(r + \delta r, t) \quad (3.20)$$

On peut généraliser cette discrétisation et utiliser des coefficients a_i plus généraux :

$$x(r, t + \delta t) = \sum_{i=-R}^R a_i x(r + i\delta r, t) \quad (3.21)$$

avec a_i des coefficients de pondération et R la taille du voisinage sur laquelle est effectuée la moyenne.

En utilisant un développement de Taylor en espace des $x(r + i\delta r, t)$, on obtient :

$$x(r, t + \delta t) = c_0 x(r, t) + c_1 \frac{\partial x(r, t)}{\partial r} + \frac{c_2}{2} \frac{\partial^2 x(r, t)}{\partial r^2} + O(\delta r^3) \quad (3.22)$$

avec :

$$c_0 = \sum_{i=-R}^R a_i \quad (3.23)$$

$$c_1 = \sum_{i=-R}^R (i\delta r) a_i \quad (3.24)$$

$$c_2 = \sum_{i=-R}^R (i\delta r)^2 a_i \quad (3.25)$$

On obtient une approximation de l'équation de diffusion à partir de (3.22) en imposant :

$$\begin{aligned} c_0 &= 1 \\ c_1 &= 0 \\ c_2 &= 2\delta t \mathcal{D} \end{aligned} \quad (3.26)$$

Un cas particulier intéressant est celui où tous les a_i sont égaux. On a alors :

$$a = a_i = \frac{1}{2R + 1} \quad (3.27)$$

et le coefficient de diffusion est :

$$\mathcal{D} = \frac{R(R + 1)}{6} \frac{\delta r^2}{\delta t} \quad (3.28)$$

Cette méthode peut aisément se généraliser en deux dimensions ou plus :

$$\boxed{x(r, t + \delta t) = a \sum_{i=-R}^R \sum_{j=-R}^R x(r + (i, j)^T \delta r, t)} \quad (3.29)$$

$$a = \frac{1}{(2R + 1)^2} \quad (3.30)$$

$$D = \frac{R(R + 1)}{6} \frac{\delta r^2}{\delta t} \quad (3.31)$$

où r est maintenant le vecteur position.

En conclusion, nous avons bien établi le fait qu'un automate moyeneur se comporte de manière similaire à une équation de diffusion. Le coefficient de diffusion \mathcal{D} est en relation directe avec la taille R du voisinage considéré dans la moyenne.

Application des automates moyeneurs à la modélisation matière

L'équation de comportement de notre matériau est une équation de diffusion de la pression :

$$\frac{\partial P}{\partial t} = \lambda \Delta P \quad (3.32)$$

où λ est le coefficient de diffusion.

Calculer un champ de pression discret $P_{i,j,k}^t$ peut donc se faire à l'aide d'un automate moyeneur sur un voisinage carré \mathcal{N} :

$$P_{(i,j,k)}^{t+1} = \frac{1}{|\mathcal{N}|} \sum_{(s,t,u) \in \mathcal{N}} P_{(s,t,u)}^t \quad (3.33)$$

Cependant, la variable intéressante pour nous n'est pas la pression, mais plutôt la quantité de matière contenue dans chaque voxel car elle nous donne la position et la forme du corps modelé. Notre objectif est donc de passer d'une loi de comportement exprimée en terme de pression à une loi de comportement exprimée en terme de quantité de matière Q .

La variation de quantité de matière contenue dans le voxel (i, j, k) entre les instants t et $t + 1$ peut être écrite sous forme d'un bilan des entrées et sorties de matière sur chacune des facettes du voxel.

$$Q_{(i,j,k)}^{t+1} = Q_{(i,j,k)}^t + \sum_{\text{Facettes de } (i,j,k)} \delta h^2 \delta t \rho \vec{u}_{\text{facette}} \quad (3.34)$$

ou encore, pour le schéma de discrétisation choisi :

$$Q_{(i,j,k)}^{t+1} = Q_{(i,j,k)}^t - \delta h^2 \delta t \rho \left(\begin{array}{l} xU_{(i,j,k)}^t - xU_{(i-1,j,k)}^t + \\ yU_{(i,j,k)}^t - yU_{(i,j-1,k)}^t + \\ zU_{(i,j,k)}^t - zU_{(i,j,k-1)}^t \end{array} \right) \quad (3.35)$$

on reconnait ici l'écriture d'une divergence discrète :

$$Q_{(i,j,k)}^{t+1} = Q_{(i,j,k)}^t - \delta h^3 \delta t \rho \tilde{div}(u) \quad (3.36)$$

Or, nous avons choisi une expression de $P_{(i,j,k)}^t$ telle que :

$$P_{(i,j,k)}^t = -\lambda \tilde{div}(\vec{u}) \quad (3.37)$$

On obtient alors :

$$Q_{(i,j,k)}^{t+1} = Q_{(i,j,k)}^t + \frac{\delta h^3 \delta t \rho}{\lambda} P_{(i,j,k)}^t \quad (3.38)$$

Cette équation étant valable pour tout temps t , on peut écrire :

$$\begin{aligned} Q_{(i,j,k)}^{t+1} &= Q_{(i,j,k)}^t + \frac{\delta h^3 \delta t \rho}{\lambda} P_{(i,j,k)}^t \\ Q_{(i,j,k)}^t &= Q_{(i,j,k)}^{t-1} + \frac{\delta h^3 \delta t \rho}{\lambda} P_{(i,j,k)}^{t-1} \\ Q_{(i,j,k)}^{t-1} &= Q_{(i,j,k)}^{t-2} + \frac{\delta h^3 \delta t \rho}{\lambda} P_{(i,j,k)}^{t-2} \\ Q_{(i,j,k)}^{t-2} &= \dots \\ Q_{(i,j,k)}^0 &= Q^0 \end{aligned} \quad (3.39)$$

Ce qui donne, par substitutions successives :

$$Q_{(i,j,k)}^{t+1} = Q^0 + \frac{\delta h^3 \delta t \rho}{\lambda} \sum_{\alpha=0}^t P_{(i,j,k)}^\alpha \quad (3.40)$$

Ce qui peut se réécrire :

$$Q_{(i,j,k)}^{t+1} - Q^0 = \frac{\delta h^3 \delta t \rho}{\lambda} \left(\sum_{\alpha=0}^{t-1} P_{(i,j,k)}^{\alpha+1} + P_{(i,j,k)}^0 \right) \quad (3.41)$$

On peut, sans perte de généralité, supposer que le corps n'était soumis à aucun champ de vitesse à l'instant $t = 0$, et donc que la pression était nulle à ce moment. On peut alors simplifier l'expression en :

$$Q_{(i,j,k)}^{t+1} - Q^0 = \frac{\delta h^3 \delta t \rho}{\lambda} \sum_{\alpha=0}^{t-1} P_{(i,j,k)}^{\alpha+1} \quad (3.42)$$

En utilisant l'équation 3.33 dans l'équation 3.42, on obtient :

$$Q_{(i,j,k)}^{t+1} - Q^0 = \frac{\delta h^3 \delta t \rho}{\lambda} \sum_{\alpha=0}^{t-1} \frac{1}{|\mathcal{N}|} \sum_{(s,t,u) \in \mathcal{N}} P_{(s,t,u)}^{\alpha} \quad (3.43)$$

On peut ici inverser les deux sommes sans problème et on obtient :

$$Q_{(i,j,k)}^{t+1} - Q^0 = \frac{\delta h^3 \delta t \rho}{\lambda |\mathcal{N}|} \sum_{(s,t,u) \in \mathcal{N}} \sum_{\alpha=0}^{t-1} P_{(s,t,u)}^{\alpha} \quad (3.44)$$

Si l'on réécrit l'équation 3.40, on peut exprimer la seconde somme en fonction des quantités de matière :

$$\sum_{\alpha=0}^{t-1} P_{(s,t,u)}^{\alpha} = (Q_{(s,t,u)}^t - Q^0) \frac{\lambda}{\delta h^3 \delta t \rho} \quad (3.45)$$

En utilisant cette dernière expression dans l'équation 3.44, on obtient alors l'expression finale :

$$\boxed{\left(Q_{(i,j,k)}^{t+1} - Q^0 \right) = \frac{1}{|\mathcal{N}|} \sum_{(s,t,u) \in \mathcal{N}} \left(Q_{(s,t,u)}^t - Q^0 \right)} \quad (3.46)$$

Les quantités de matière sont donc régies par un automate cellulaire dont la loi d'évolution est similaire à celle de l'automate régissant les pressions. Pour notre modélisation, il suffira de travailler avec les quantités de matière.

L'équation 3.46 porte non pas sur la quantité de matière $Q_{(i,j,k)}^t$, mais sur la grandeur :

$$\left(Q_{(i,j,k)}^t - Q^0 \right)$$

qui représente un excès local de matière par rapport à la distribution initiale Q^0 .

Nous avons affirmé dans les pages précédentes que le choix de l'écriture :

$$P = -\lambda \operatorname{div}(\vec{u})$$

constituait un relachement de la contrainte d'incompressibilité $\operatorname{div}(\vec{u}) = 0$.

On peut en effet constater que d'après l'équation (3.46) une compression locale est admise, le processus contribuant simplement à la distribuer de proche en proche jusqu'à ce qu'elle atteigne des voxels où elle pourra être absorbée.

Si nous n'avions pas choisi cette hypothèse portant sur la forme du champ de pression, la contrainte d'incompressibilité aurait dû être écrite sous la forme :

$$\Delta P = 0$$

Cette équation de Poisson peut être résolue numériquement de plusieurs façons : par transformée de Fourier rapide, par différenciation finie, par méthode multigrille, etc. Une autre alternative aurait pu être l'utilisation d'une méthode de relaxation. Cette méthode repose sur une transformation de l'équation de Poisson en équation de diffusion de la forme :

$$\frac{\partial P}{\partial t} = \Delta P$$

On laisse ensuite évoluer le champ P jusqu'à obtenir convergence, c'est-à-dire :

$$\frac{\partial P}{\partial t} = 0 = \Delta P$$

L'équation de diffusion converge vers une équation de Poisson. Notre méthode basée sur les automates cellulaires repose sur le même principe : c'est en cela que l'on peut dire que nous avons "relâché" la contrainte d'incompressibilité. Nous vérifierons dans la suite, consacrée à la mise en pratique de la méthode, que malgré la relaxation de cette contrainte, nous avons bien conservation de la masse durant la déformation.

3.3.2.4 Implémentation du modèle

Maintenant que les règles de transitions de l'automate cellulaire ont été définies, l'implémentation peut se faire de manière directe. Au début de la simulation, l'espace est initialisé de la façon suivante :

- Les voxels n'appartenant pas au corps contiennent une quantité de matière nulle : $Q_{(i,j,k)}^0 = 0$
- Les voxels appartenant au corps contiennent une quantité de matière Q_0 appelée seuil : $Q_{(i,j,k)}^0 = Q_0$

Cette valeur Q_0 est supposée correspondre à un état où une cellule est en équilibre. Tant que la quantité de matière contenue dans les voxels reste en dessous de ce seuil, le matériau ne s'écoule pas et l'objet conserve sa forme. Lorsque la quantité de matière contenue dans une cellule se trouve au dessus du seuil, l'excès de matière $E_{(i,j,k)}^t = Q_{(i,j,k)}^t - Q_0$ est réparti grâce à un automate moyennneur du type (3.46).

La taille R du voisinage utilisé pour l'automate peut être fixée arbitrairement : nous avons vu qu'elle était liée à la constante de diffusion de la quantité de matière, c'est à dire à la vitesse à laquelle une surpression va se propager dans le corps. Tout ceci provient de la relaxation de la contrainte d'incompressibilité que nous avons faite : plus le voisinage sera grand, et plus la compression locale sera vite dissipée. Nous avons choisi, pour nos tests, une taille $R = 2$ et deux types de voisinage ont été testés : un voisinage centré et un voisinage de Margolus.

1. Voisinage centré :

Le premier type de voisinage est un voisinage classique centré autour de la cellule cible comme le montre la figure 3.4 .

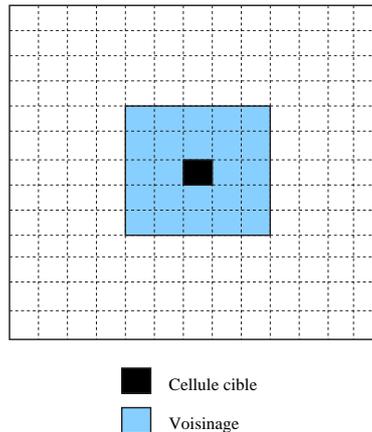


FIG. 3.4 – Voisinage centré sur la cellule cible (cas 2D)

Il correspond au voisinage que nous avons utilisé afin d'établir l'équation 3.46 .

2. Voisinage de Margolus :

Le second voisinage utilisé, présenté en figure 3.5 est un voisinage de Margolus. Dans ce voisinage, les quatre cellules les plus proches forment un "bloc". Les transitions d'états de toutes les cellules appartenant à un même bloc sont réalisées en une seule étape. De plus, les limites des blocs sont changées après chaque pas. Durant les pas pairs, une cellule n'appartient donc pas au même voisinage que durant les pas impairs : elle alterne après chaque transition d'état.

Pour chaque bloc, l'opération est très simple. On calcule tout d'abord la quantité de matière à répartir, en sommant la quantité de matière en excès dans les cellules

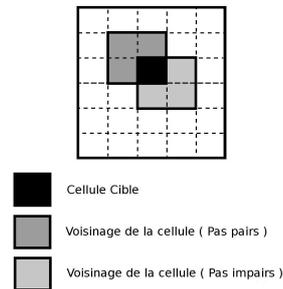


FIG. 3.5 – Le voisinage de Margolus (cas 2D)

du bloc. Cette matière à répartir est ensuite partagée entre les différentes cellules du bloc acceptantes, c'est à dire ne faisant pas partie de l'outil ou d'un obstacle. La généralisation de ce voisinage au cas 3D est triviale. La forme du voisinage de Margolus 3D est montrée figure 3.6 .

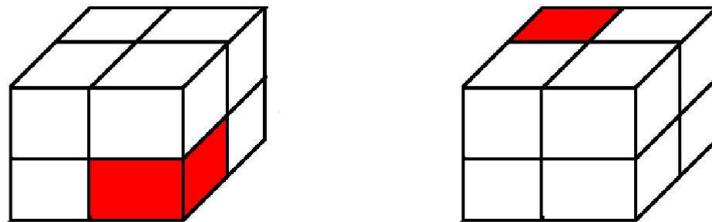


FIG. 3.6 – Le voisinage de Margolus (cas 3D)

Le voisinage de Margolus est en fait très proche du voisinage centré. Ainsi, si on observe son comportement entre les instants t et $t + 2$, on peut s'apercevoir qu'il se ramène à un moyenneur appliqué au voisinage illustré figure 3.7.

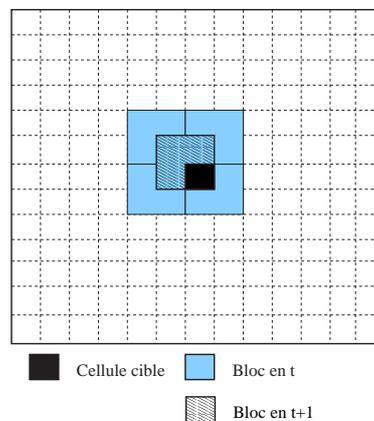


FIG. 3.7 – Le voisinage de Margolus sur deux pas de temps (cas 2D)

Si le voisinage de Margolus présente une certaine dissymétrie par rapport au voisinage centré, il est cependant beaucoup plus intéressant en terme de nombre d'opérations effec-

tuées.

Ainsi, le calcul d'une seule cellule dans le cas du voisinage centré est effectué par une moyenne sur $5 * 5 = 25$ cellules, soit donc :

- 1 division
- 24 additions

Alors que dans le cas du voisinage de Margolus, le calcul d'un bloc de quatre cellules nécessite :

- 5 divisions
- $5 * 3 = 15$ additions

Le voisinage de Margolus est donc beaucoup plus efficace que le voisinage centré en terme de nombre de calculs à effectuer. Il se prête de plus très facilement à la parallélisation.

3.3.2.5 Interaction Outil-Matière

Maintenant que nous disposons d'un modèle de déformation pour le corps modelé, l'action de l'outil devient extrêmement facile à intégrer. Lorsque l'utilisateur applique l'outil sur la matière, il impose à la matière une vitesse au niveau de l'interface outil matériau. Cette vitesse imposée rend (temporairement) la divergence non nulle sur les voxels voisins : l'outil provoque une surpression locale et donc un écoulement.

La modélisation de l'interaction outil-matière repose sur ce principe de surpression locale. Les voxels nouvellement occupés par l'outil virtuel sont vidés, et la matière qu'ils contenaient est transmise aux voisins libres, suivant la direction du mouvement de l'outil : l'outil "pousse" la matière. La figure 3.8 illustre ce principe.

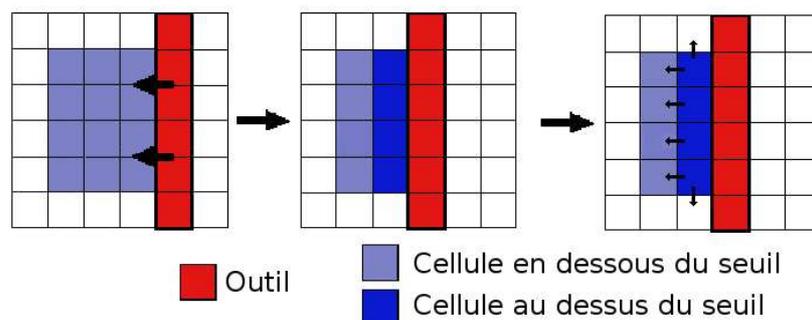


FIG. 3.8 – Action de l'outil sur le corps (cas 2D)

Les voxels qui reçoivent cet excédent se retrouvent au dessus du seuil Q_0 : l'automate cellulaire entame alors le processus de diffusion de la surcharge.

Afin de tester le modèle, nous avons choisi d'utiliser un bras Phantom à six degrés de liberté comme interface d'entrée. L'utilisateur peut ainsi tenir l'organe terminal à la façon d'un stylo et interagir avec l'objet à l'écran. Une boucle permet de récupérer à intervalles réguliers la position et l'orientation de l'organe terminal du Phantom. Pour l'instant, seule la position sera utilisée et nous nous limiterons à des mouvements de translation de l'outil. Le mouvement entre deux positions lues depuis le Phantom est décomposé en mouvements élémentaires qui sont traités successivement.

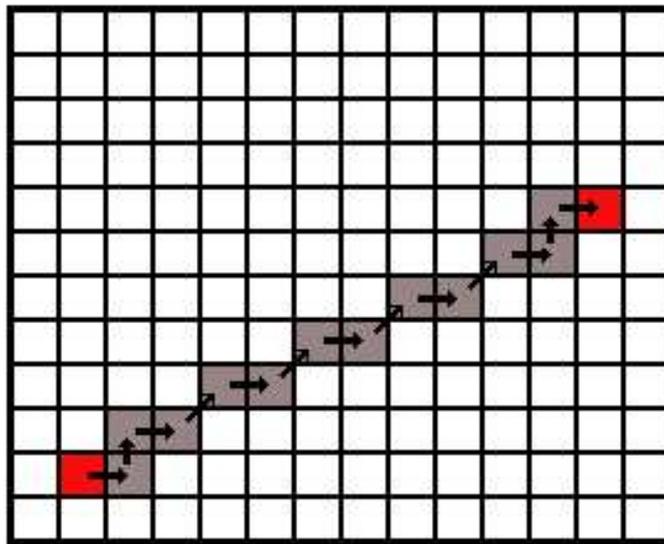


FIG. 3.9 – Décomposition du mouvement en translations élémentaires

3.3.2.6 Résultats et performances

Le système que nous avons décrit a été testé sur PC bi-processeurs Xeon cadencé à 1,7 GHz. La faible complexité des calculs mis en jeu permet une utilisation interactive du modèle pour des espaces de travail allant jusqu'à $N = 100$. Les outils utilisés ici sont de taille $N_{outil} = 32$ voxels et peuvent être de forme quelconque sans que cela influe sur la réactivité du système.

Le rendu graphique est réalisé à l'aide de l'algorithme des Marching Cubes (cf. Annexe B), grâce à un "thread" fonctionnant de manière indépendante. Ce rendu est coûteux en temps de calcul, aussi un processeur lui est totalement affecté, l'autre servant à l'exécution du modèle matière.

La figure 3.10 montre un exemple de déformation obtenue à l'aide du modèle développé. Un cube de matière (en jaune sur ce schéma) est déformé à l'aide d'un outil en

forme de boule (en gris). On observe la formation d'un bourrelet provoqué par le mouvement de l'outil et la déformation est assez réaliste.

Etant donnée l'absence de viscosité dans le modèle, les couches de matière en mouvement n'entraînent pas les couches voisines, ce qui donne un bord d'attaque assez franc à l'endroit où l'outil est passé. Cette absence de viscosité fait que la matière semble aussi manquer de cohérence : ceci peut se traduire par un "émiettement" de la matière modelée lorsqu'on la manipule pendant une longue durée (figure 3.11)

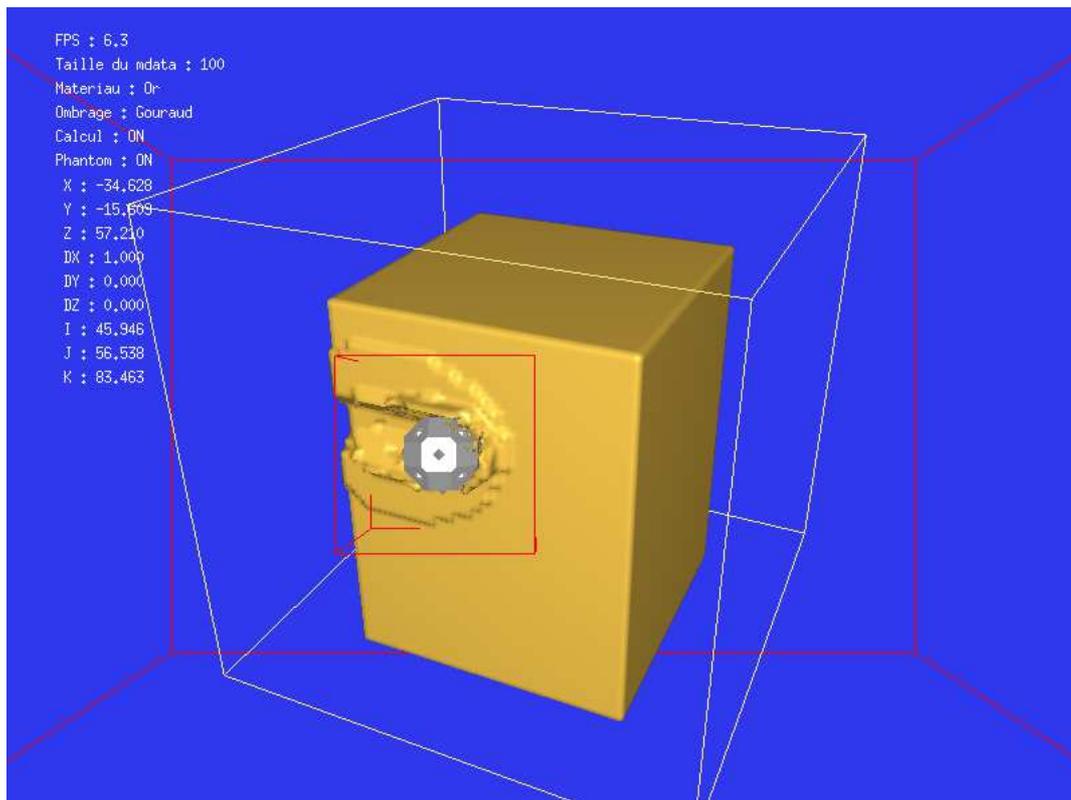


FIG. 3.10 – Déformations avec un outil sphérique ($N = 100$, $N_{outil} = 32$)

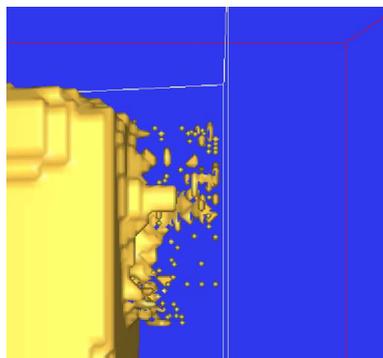
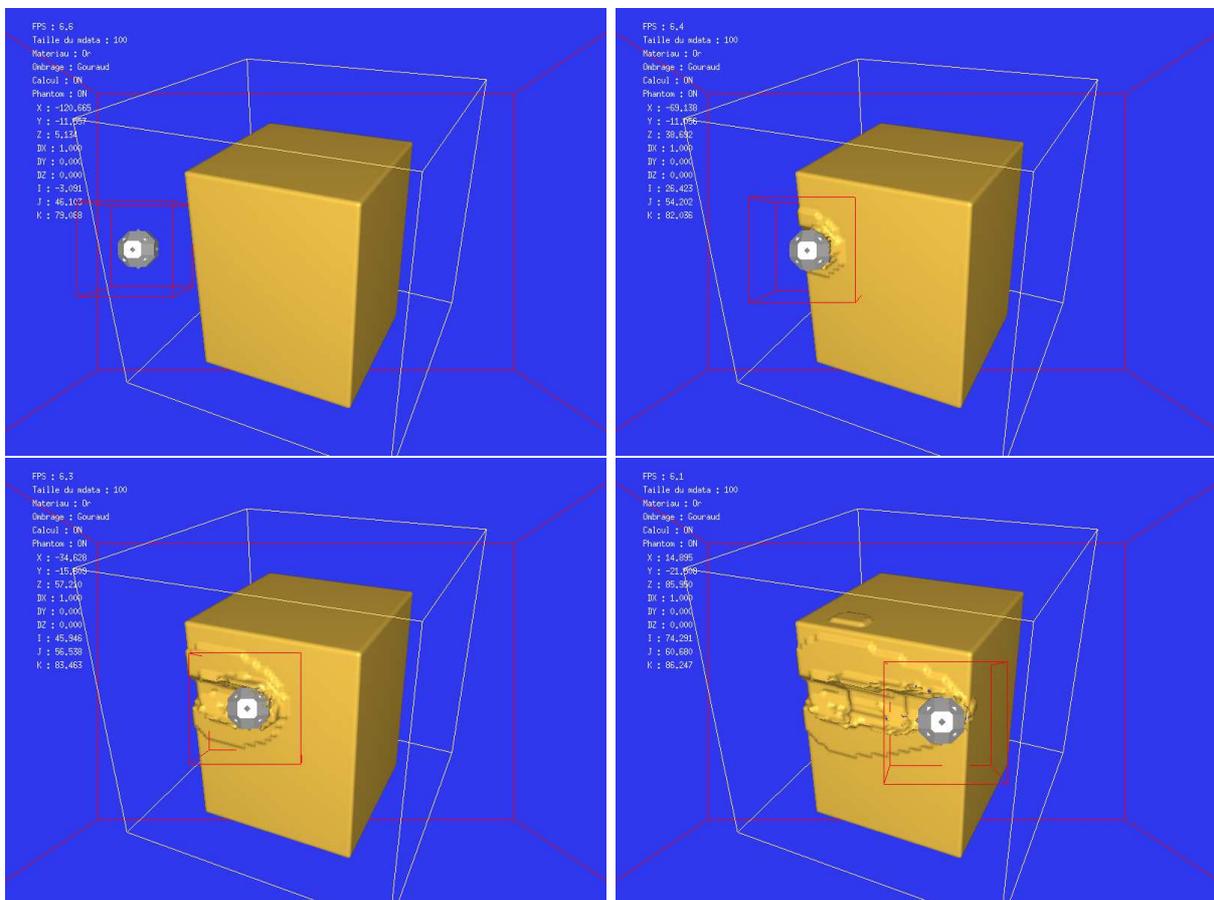


FIG. 3.11 – Un manque de cohésion de la matière

FIG. 3.12 – Utilisation d'un outil sphérique ($N = 100$, $N_{outil} = 32$)

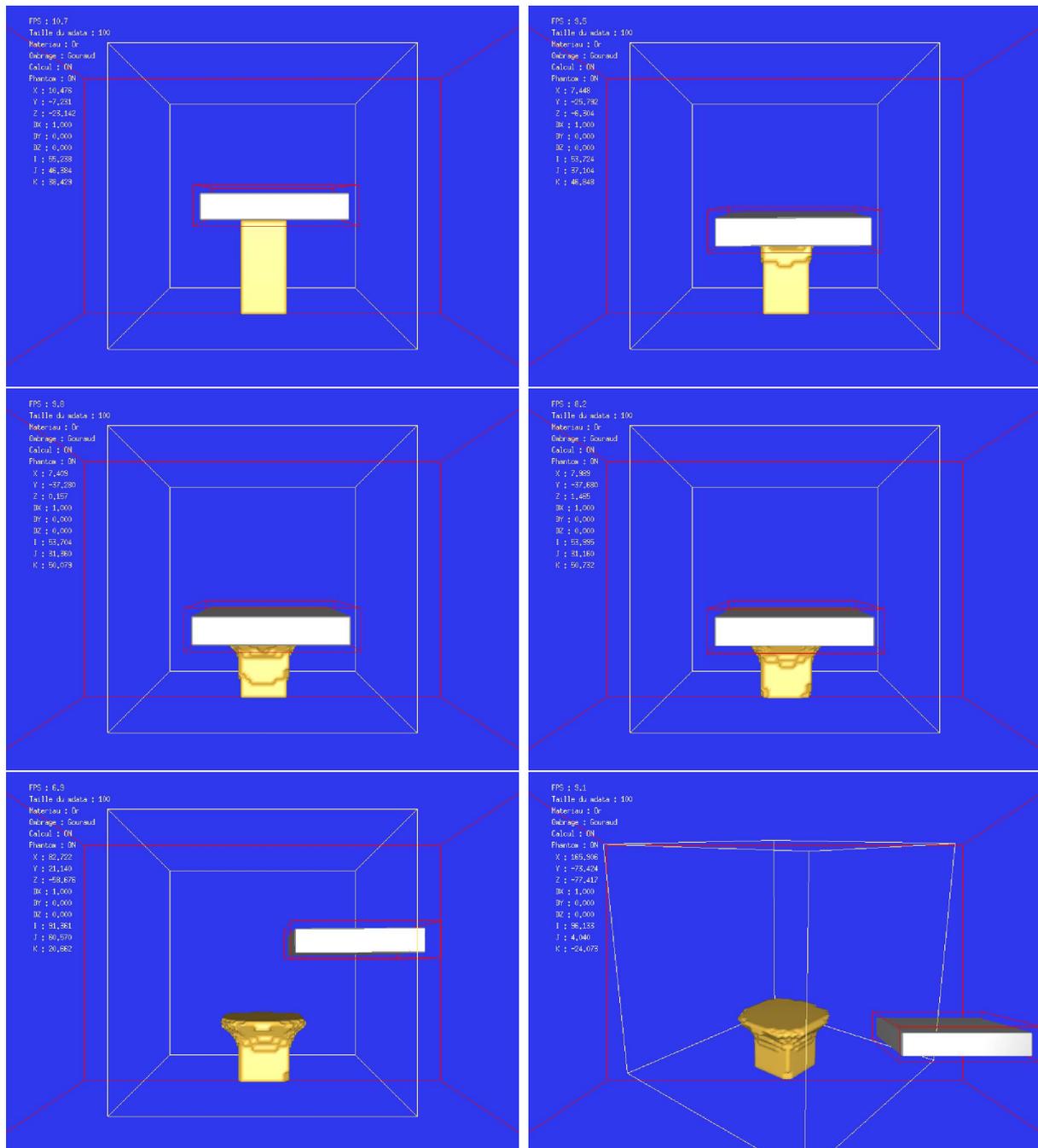


FIG. 3.13 – Compression par outil parallélépipédique ($N = 100$, $N_{outil} = 64$)

Complexité du modèle

La logique voudrait que le temps de calcul soit directement relié au nombre de voxels utilisés dans le modèle. Afin de vérifier cette hypothèse, nous avons mesuré pour différentes discrétisations de l'espace de mises à jour de l'espace voxel effectués par seconde. Les résultats sont présentés dans le tableau 3.1.

N	200	150	100	90	80	70	60	50
Nombre de calculs par seconde	3	6	20	30	42	49	90	130

TAB. 3.1 – Evolution du temps de calcul en fonction de N

Comme on aurait pu le deviner, le temps de calcul est d'autant plus important que le nombre de voxels à traiter est grand. Ces mêmes résultats ont été portés sur le graphique de la figure 3.14 .

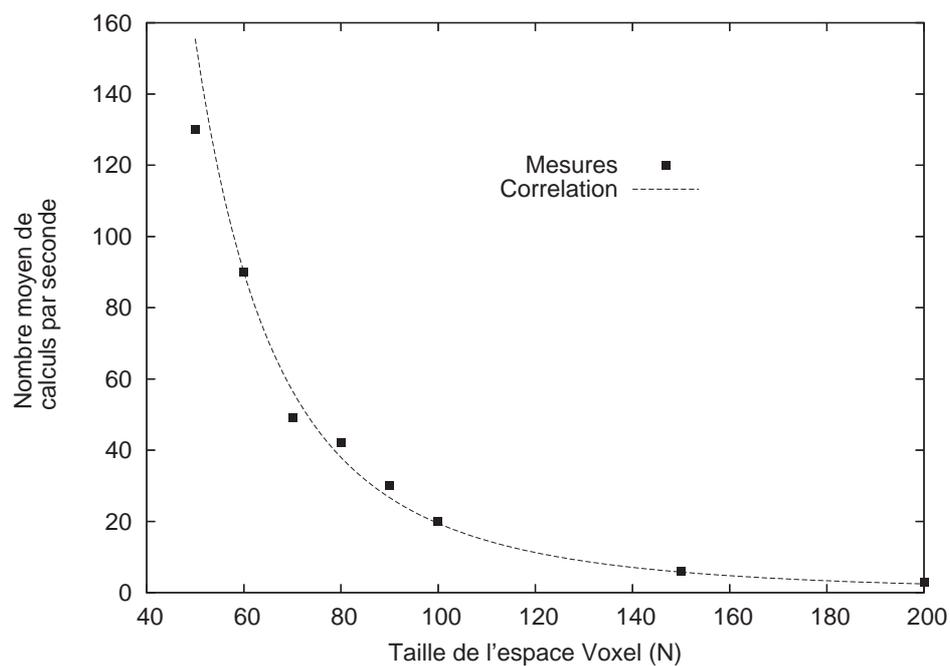


FIG. 3.14 – Temps de calcul moyen en fonction de la taille de la lattice

La courbe représentée en pointillés sur la figure représente une fonction de la forme $y = a.x^3$. **On peut donc vérifier expérimentalement que le temps nécessaire au calcul se comporte bien suivant une loi en N^3** , ce qui n'est pas un avantage pour une utilisation interactive.

3.3.2.7 Implémentation parallèle

L'un des avantages d'une méthode basée sur les automates cellulaires est souvent la facilité avec laquelle l'algorithme peut être porté sur des machines multiprocesseurs. Dans notre cas, on peut remarquer que le calcul de chaque élément (bloc ou cellule suivant le voisinage choisi) se fait de manière indépendante des autres. Ce constat nous a permis de modifier notre prototype afin d'en faire une version multithread.

Le principe, illustré en figure 3.15, est basé sur une file contenant les éléments à traiter. Chacun des threads peut extraire un item de la file, le traiter de manière autonome, stocker le résultat, puis en extraire un autre et ainsi de suite ...

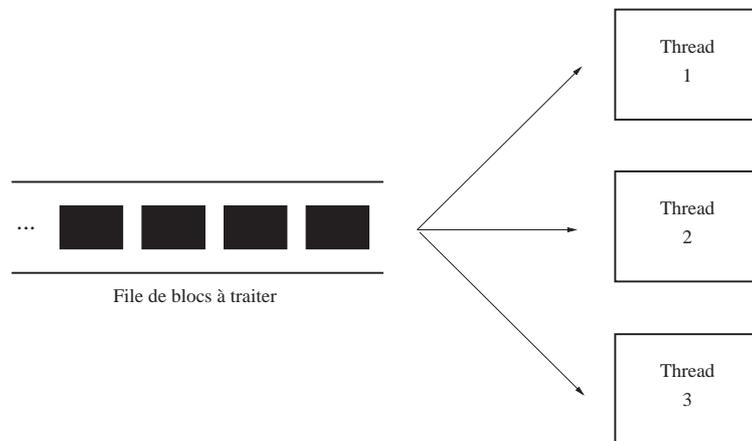


FIG. 3.15 – Implémentation multithread de l'automate cellulaire

Le nombre de threads utilisés peut être choisi librement sans changer quoi que ce soit au programme. Nous avons effectué des mesures pour un espace de travail de taille $N = 100$ en faisant varier le nombre de threads de calcul. Les résultats obtenus sont donnés dans le tableau suivant :

Nbre de threads utilisés	1	2	3	4	5
Nombre de Calculs par seconde	20	35	40	42	41

TAB. 3.2 – Evolution du temps de calcul en fonction du nombre de threads

La parallélisation permet donc d'obtenir des performances intéressantes. Cependant, on peut constater qu'au delà d'un certain nombre de threads, le nombre de calculs effectués par seconde n'évolue plus. Cette limite nous laisse à penser que nous avons atteint les limites de la machine sur laquelle les simulations ont été effectuées.

3.3.2.8 Conclusions sur ce premier modèle

Cette première approche du problème nous a donc mené à un modèle dont les résultats semblent intéressants en terme de réalisme visuel de la déformation. L'utilisation des

automates cellulaires rend cette solution viable en termes de temps de calcul et d'interactivité, et apporte de plus une très grande simplicité dans l'implémentation.

Le principal inconvénient de cette méthode est le fait qu'elle se base sur une description volumique de l'espace de simulation, ce qui se traduit par une complexité cubique. Cependant, les possibilités de parallélisation associées aux automates cellulaires et l'utilisation d'un voisinage de Margolus permettent une réduction des temps de calcul.

3.3.3 Une approche surfacique de la déformation

3.3.3.1 Introduction

La méthode précédente, bien que se prêtant très facilement à une implémentation massivement parallèle, possède deux points faibles : l'absence de viscosité, qui limite le réalisme du comportement obtenu, et le fait qu'elle agit sur le volume, ce qui se traduit par un coût important en terme de temps de calcul.

En observant le comportement du modèle précédent, on peut observer que l'effet de la suppression locale a toujours lieu à la surface du corps, l'intérieur ne servant finalement que de médium à la transmission d'une onde. Cette constatation effectuée, nous nous sommes alors demandé s'il n'était pas possible de ne traiter que cette surface, et non pas tout le volume. Ceci nous permettrait d'économiser le traitement de nombreux voxels et donc d'accélérer de manière intéressante le calcul.

Le modèle que nous allons maintenant aborder est basé sur ce principe. Grâce à un suivi de la matière à l'aide de l'équation des marqueurs, nous allons exprimer sous la forme d'une quantité scalaire Φ quels voxels de la surface vont avoir tendance à gagner de la matière et quels voxels vont avoir tendance à en perdre.

3.3.3.2 Approche surfacique de la modélisation

Cette nouvelle modélisation repose sur l'incompressibilité du matériau, c'est à dire sur le respect strict de la contrainte :

$$\operatorname{div}(\vec{u}) = 0 \quad (3.47)$$

en tout point du matériau modelé.

On suppose que des marqueurs sont placés de manière uniforme dans le corps à l'instant initial. Ceux-ci vont être entraînés par la matière et nous permettre de la suivre. L'équation d'évolution de la concentration de marqueurs $C(x, y, z, t)$ nous est donnée par la mécanique des fluides :

$$\rho \frac{\partial C}{\partial t} + \rho \operatorname{div} \left(C \vec{u} - \nu_D \overrightarrow{\operatorname{grad}}(C) \right) = J_{source} \quad (3.48)$$

où J_{source} est un taux de création de marqueurs, ν_D est le coefficient de diffusion.

Cette équation est en général appliquée en mécanique des fluides à des études de transport de contaminant. Physiquement, l'évolution en temps de la quantité C dépend :

- du transport du traceur par le fluide (convection),
- de la diffusion du traceur,
- de termes sources (création de marqueurs) ou puits (consommation de marqueurs).

Dans notre cas, il n'y a ni source ni puits de marqueurs. On peut donc poser :

$$J_{source} = 0 \quad (3.49)$$

De plus, on suppose que les marqueurs ne se déplacent pas lorsque le fluide modélisé est au repos. La diffusion est donc nulle :

$$\nu_D = 0 \quad (3.50)$$

Tout ceci nous ramène donc à une forme simplifiée de l'équation des traceurs 3.48 :

$$\frac{\partial C}{\partial t} = - \operatorname{div}(C \vec{u}) \quad (3.51)$$

ce qui s'écrit encore :

$$\frac{\partial C}{\partial t} = - C \operatorname{div}(\vec{u}) - \vec{u} \cdot \overrightarrow{\operatorname{grad}}(C) \quad (3.52)$$

En prenant en compte la contrainte d'incompressibilité $\operatorname{div}(\vec{u}) = 0$, on arrive finalement à l'équation finale :

$$\boxed{\frac{\partial C}{\partial t} = - \vec{u} \cdot \overrightarrow{\operatorname{grad}}(C)} \quad (3.53)$$

Cette équation est en fait la relation que nous cherchions car elle ne porte que sur la surface du corps. En effet, le gradient de quantité de marqueurs est nul à l'intérieur du corps puisque nous avons fait une hypothèse d'incompressibilité et une hypothèse d'équi-répartition initiale des marqueurs. Il est, par contre, non nul pour les points appartenant à la frontière du corps, c'est à dire à la surface, et est orienté suivant la normale de celle-ci.

De là découle l'idée fondamentale de ce modèle. Au lieu de parcourir tout le volume pour traiter la déformation, nous n'allons nous intéresser qu'aux voxels formant la surface du corps et déterminer à l'aide de la relation 3.53 les voxels qui perdent de la matière, et les voxels qui en gagnent. Pour ce faire, on définit sur la surface de notre matériau une quantité Φ de la forme :

$$\Phi = \vec{u} \cdot \vec{n} \quad (3.54)$$

Avec :

- \vec{u} la vitesse du fluide sur l'élément de surface,
- \vec{n} le vecteur normal à l'élément de surface.

Les points correspondant à une valeur de Φ négative sont des points qui perdent de la matière. Les points correspondant à une valeur de Φ positive sont ceux qui au contraire vont en recevoir.

On peut vérifier, par le théorème d'Ostrogradski, que ce modèle de comportement conserve bien la quantité globale de matière. A titre de rappel, ce théorème est le suivant :

Théorème d'Ostrogradski

Considérons un domaine \mathcal{D} de l'espace affine \mathcal{E} , dans lequel le champ de vecteurs \vec{A} est défini en tout point M . Soit une surface fermée S , caractérisée en chacun de ses points par le vecteur unitaire \vec{n} , et délimitant un volume V de \mathcal{D} . On peut alors écrire :

$$\iint_S \vec{A}(P) \cdot \vec{n}(P) dS = \iiint_V \text{div } \vec{A}(M) dV$$

dV est l'élément de volume centré sur le point M de V et dS l'élément de surface entourant le point P de S .

L'application de ce théorème au calcul de Φ sur la surface nous donne donc :

$$\Phi_{\text{Totale}} = \iint_{\text{Surface}} \Phi dS = \iint_{\text{Surface}} \vec{u} \cdot \vec{n} dS = \iiint_V \text{div } \vec{u} dV = 0 \quad (3.55)$$

Si nous faisons varier la quantité de matière sur la surface en fonction de Φ , la conservation de la matière devrait donc être globalement assurée.

3.3.3.3 Représentation voxel du corps modélisé

Comme précédemment, l'espace de travail est représenté à l'aide d'une grille voxel. A chacun des voxels, on associe une grandeur $q_{(i,j,k)}^t$ représentant la matière contenue dans le voxel.

Cette fois, $q_{(i,j,k)}^t$ n'exprime pas une quantité absolue de matière, mais un pourcentage de remplissage du voxel. Etant donné que nous nous sommes donné une contrainte d'incompressibilité stricte, cela revient absolument au même car on peut lier $Q_{(i,j,k)}^t$ et $q_{(i,j,k)}^t$ par la relation :

$$q_{(i,j,k)}^t = \frac{1}{\rho \delta h^3} Q_{(i,j,k)}^t \quad (3.56)$$

et on a $q_{(i,j,k)}^t \in [0, 1]$.

Deux notions ne sont pas sans poser problème pour la discrétisation de l'équation 3.53. Le premier problème est la notion de surface dans un univers où la matière est discrétisée suivant une lattice régulière. Le second problème, quant à lui, porte sur le calcul des normales à cette surface.

Définition de la surface dans un espace de travail discret

Puisque nous désirons établir un modèle basé sur des modifications surfaciques, un concept sur lequel il est intéressant de s'attarder est celui de surface. Dans un univers de simulation constitué de voxels, il n'est pas toujours aisé de définir quels voxels forment une surface permettant de délimiter le corps.

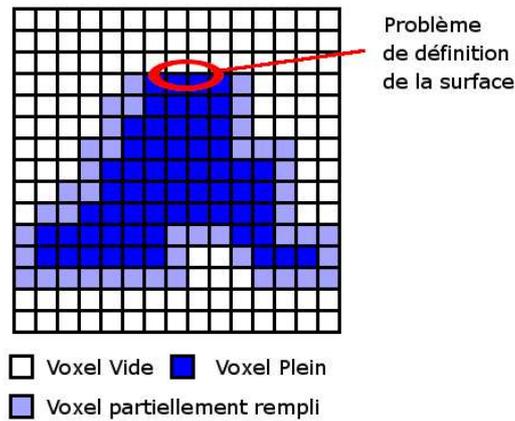


FIG. 3.16 – Surface d'un corps voxelisé (Cas 2D)

Si un voxel est partiellement rempli, alors nous pouvons dire qu'il appartient à la surface du corps. Un cas qui pose problème est le cas où l'on passe directement d'un voxel plein à un voxel vide, comme illustré sur la figure 3.16. Dans ce cas, la surface est exactement à l'interface entre le voxel vide et le voxel plein. On ne peut pas vraiment dire que l'un ou l'autre la contienne.

Ce problème a été résolu en introduisant un concept différent de celui de voxel pour la surface. Nous avons appelé l'entité sur laquelle porte nos calculs un "élément de surface". Cette élément de surface peut être :

- soit formé d'un seul voxel non saturé $q < 1$,
- soit formé de deux voxels voisins, l'un saturé et l'autre vide.

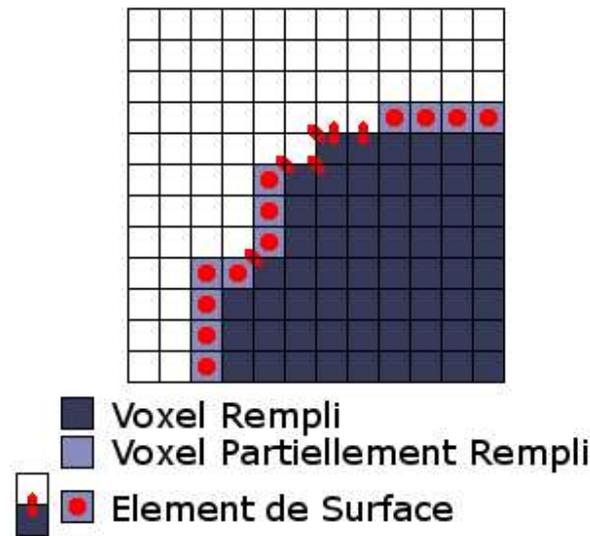


FIG. 3.17 – Les éléments de surface considérés

Il reste maintenant à définir un vecteur normal pour chacun des éléments de surface. Nous avons choisi de précalculer ces normales en fonction de la configuration des voxels voisins saturés. Chacun de ces voisins impose à son entourage une direction de normale, comme montré sur la figure 3.18.

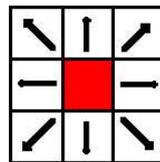


FIG. 3.18 – Directions des normales imposées par un voxel plein (au centre) à son entourage

Par application d'un principe de superposition, on construit alors une Look-Up-Table dans laquelle toutes les configurations de voisinages sont présentes. Dans le cas d'un élément de surface double, le voxel central dont on étudie le voisinage est celui qui est vide. La figure 3.19 illustre quelques cas de normales avec les voisinages associés.

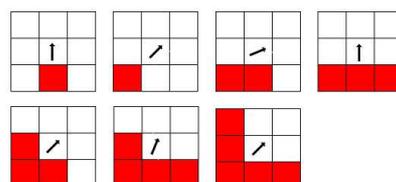


FIG. 3.19 – Quelques exemples de normales en fonction des voisinages

On suppose, pour le moment, qu'un champ de vitesse a été calculé et qu'il respecte

la condition de divergence nulle. On suppose, de même, que l'on dispose des éléments de surface stockés dans une liste, chacun possédant un vecteur normal \vec{n} .

Si la valeur de Φ de l'élément de surface, calculée grâce à la formule 3.54, est négative, celui-ci perd de la matière. Si au contraire elle est positive, l'élément de surface gagne de la matière. Cette évolution est obtenue pratiquement grâce à l'expression :

$$q_{(i,j,k)}^{t+1} = q_{(i,j,k)}^t + b \Phi_{(i,j,k)}^t \quad (3.57)$$

où le coefficient b est une constante imposée par l'utilisateur. Ici encore, il nous faut préciser le cas des éléments de surface doubles. Si Φ est positif, le voxel auquel on applique cette formule est le voxel vide qui va recevoir la matière puisque son adjoint est déjà saturé. Si Φ est négatif, c'est bien sûr le voxel plein qui perd de la matière.

Même si nous avons vu que Φ_{Totale} était nulle dans le cas continu, ce n'est pas tout à fait vrai dans le cas discret en raison de notre discrétisation de la surface du corps. La matière manquante n'est cependant en général pas très importante par rapport à la matière totale : nous avons choisi de la partager et la distribuer à chacun des éléments de surface.

3.3.3.4 Choix du champ de vitesse et influence de l'outil

La méthode de répartition de la matière que nous venons d'étudier est très intéressante, car peu gourmande en calcul. Elle nécessite cependant de disposer d'une expression de champ de vitesse qui soit à la fois :

- de divergence nulle,
- définie et calculable aisément en tout point de l'espace.

Une idée simple pour construire un tel champ de vitesse \vec{u} est de faire l'hypothèse que le champ dérive d'un potentiel scalaire \mathcal{P} :

$$\vec{u} = \overrightarrow{grad}(\mathcal{P}) \quad (3.58)$$

Etant donné que la divergence de \vec{u} est nulle, \mathcal{P} doit respecter :

$$\Delta \mathcal{P} = 0 \quad (3.59)$$

En utilisant la linéarité de cette équation de Poisson, nous avons choisi l'écriture suivante :

$$\mathcal{P}_{(x,y,z)} = \sum_n \frac{\mathcal{P}_n}{(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2} \quad (3.60)$$

où le potentiel est calculé en tout point comme une somme de potentiels créés par des sources placées en (x_n, y_n, z_n) et de valeurs \mathcal{P}_n .

Lorsque l'outil entre en contact avec la matière, il crée un champ de vitesse, ce qui revient à dire qu'il impose un potentiel en un ou plusieurs points de la surface du corps. Ces sources de potentiel permettent ainsi d'agir sur le corps et de provoquer l'écoulement suivant des directions radiales au point de contact outil/matière.

3.3.3.5 Résultats et conclusion

Les quelques images ci-après permettent de voir les limitations de ce modèle. Etant donné que l'écriture du champ de vitesse ne tient absolument pas compte de la géométrie du corps, la deuxième branche du U se déforme alors qu'elle n'est pas affectée par l'outil.

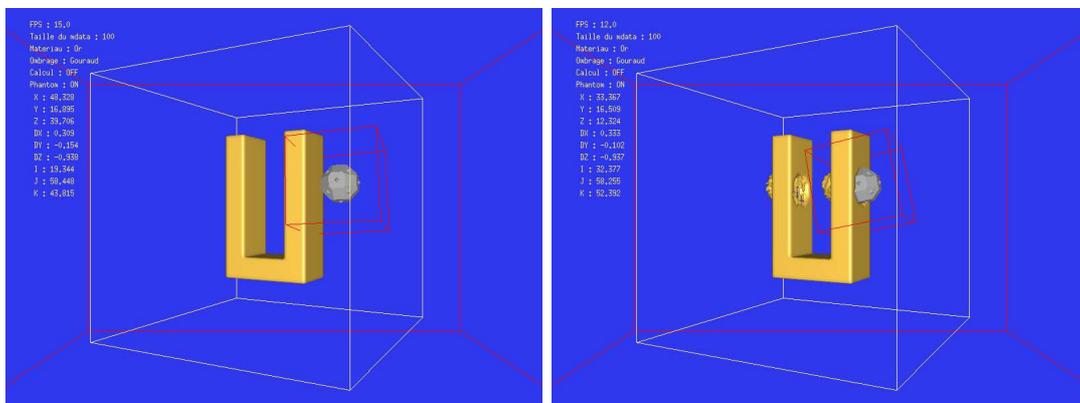


FIG. 3.20 – La distance cartésienne pose des problèmes de cohérence de la déformation

Le champ de vitesse choisi ne tient pas compte de la topologie du matériau, ce qui pose problème dans le cas de corps où les concavités sont importantes. A la place de la distance cartésienne utilisée dans l'expression du potentiel, il nous faudrait concevoir une distance "au sens du corps", c'est à dire exprimant le parcours le plus court entre la source de potentiel et le point considéré sans toutefois sortir de la matière comme illustré en figure 3.21.

Notre conclusion à propos de ce modèle est que la contrainte d'incompressibilité peut permettre d'obtenir à elle seule des comportements très intéressants, mais il manque une information majeure qui est la géométrie du corps puisque celui-ci est le médium grâce auquel la vitesse se propage. Cette information est présente de manière implicite dans le premier modèle, mais pas dans celui-ci.

Il serait envisageable de résoudre cette difficulté en résolvant l'équation de Poisson 3.59 sur le corps uniquement, et d'obtenir ainsi un potentiel intégrant les aspects géométriques. Ce serait en fait revenir à notre premier modèle puisque l'équation de diffusion utilisée converge vers une équation de Poisson. Cette expérience a été effectuée, et elle donne les résultats de la figure 3.22 qui confirment bien cette affirmation.

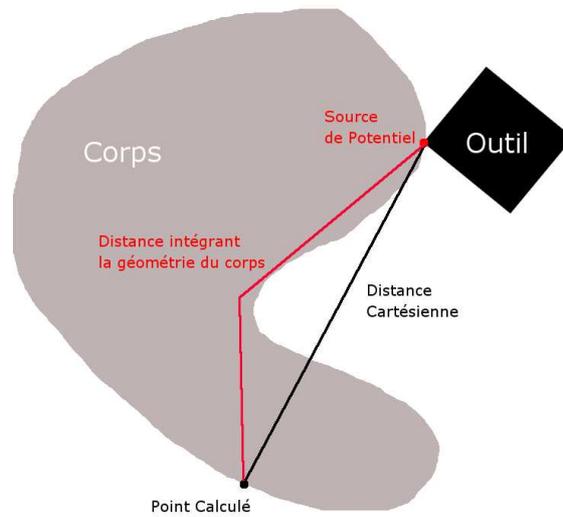


FIG. 3.21 – Distance cartésienne et géométrie du corps

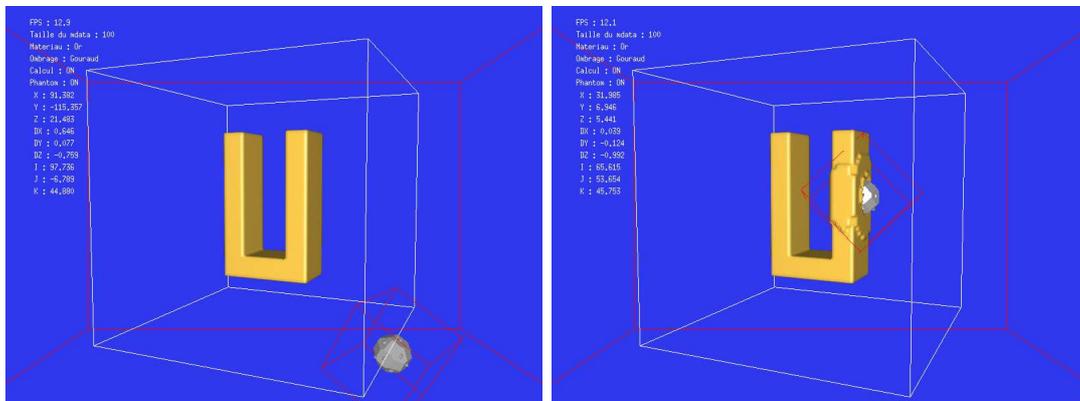


FIG. 3.22 – Champs de vitesse calculé avec les conditions aux bornes

3.3.4 Conclusion sur ces deux premiers modèles

Sur les phénomènes qui sont à la base d'une déformation viscoplastique, à savoir viscosité et conservation de la masse, nos deux modèles ne prennent en compte que le dernier. Ces deux modèles sont conceptuellement très différents, malgré le fait qu'ils ont été conçus en partant de la même idée.

Dans le premier, l'accent est mis sur une approche volumique. Les grandeurs calculées sont évaluées sur tous les voxels, ce qui permet d'intégrer la géométrie du corps dans le calcul du champ de vitesse. La conséquence d'une telle approche est que les performances sont très fortement liées à la taille de l'espace de travail et évoluent suivant une loi cubique. Plusieurs méthodes nous ont permis de rendre toutefois ce modèle utilisable de manière interactive : l'utilisation d'un formalisme local par l'emploi des automates cellulaires, apporte une simplicité d'implémentation et des possibilités de parallélisation appréciables.

Le second modèle, quant à lui, est toujours basé sur la conservation de la matière, mais cette fois à l'aide d'une approche surfacique. Cette dernière nous intéresse dans le sens où elle permet de réduire de manière importante les temps de calcul en diminuant le nombre de voxels à traiter. La difficulté est alors de construire un champ de vitesse qui soit cohérent avec la forme géométrique de l'objet sans quoi le réalisme du comportement est fortement affecté. Il ne semble pas possible de construire un tel champ sans revenir à un modèle comparable au premier.

Le bilan de cette première partie consacrée à la modélisation de la viscoplasticité est donc qu'une modélisation, même grossière, doit intégrer la notion de distance "au sein du corps", et donc une approche du champ de vitesse basée sur le volume. Plus rigoureusement, on peut dire que les conditions aux limites imposées par les surfaces libres ne sont pas négligeables et doivent donc entrer en compte dans la résolution de l'équation de Poisson.

Enfin, il semble que la contrainte d'incompressibilité du matériau permette, à elle seule, d'obtenir un comportement macroscopique visuellement très correct. Cependant, le corps manque de cohésion, ce qui se traduit par des bords d'attaque très nets dans les déformations.

3.4 Modèle complet de déformation viscoplastique

3.4.1 Introduction

Jusque là, nous nous sommes intéressés à la modélisation de corps assimilés à des corps "parfaits" de viscosité nulle. Les modèles que nous avons développés souffrent d'un manque de cohésion de la matière, ce que nous attribuons à cette hypothèse de départ. Le terme de viscosité, en effet, permet de prendre en compte les frottements internes, c'est à dire le fait que de la matière en mouvement entraîne avec elle la matière qui l'entoure.

En ce qui concerne le calcul du champ de vitesse, la forme géométrique du corps joue un rôle prépondérant dans le réalisme de la déformation. Ceci provient en fait des conditions aux limites, qui ne sont pas imposées de manière réaliste dans le dernier modèle que nous avons présenté.

Pour ce nouveau modèle, nous allons intégrer les deux équations complètes (équation des moments et équation de conservation de la matière) dans le but d'obtenir une modélisation la plus exacte possible. Une attention toute particulière sera accordée aux conditions aux limites à imposer sur les champs de pression et vitesse ainsi qu'aux algorithmes numériques utilisés.

3.4.2 Modélisation du comportement viscoplastique

Ce nouveau modèle se base sur les équations de Stokes présentées au début de ce chapitre. Ces équations sont rappelées ci-dessous :

$$\begin{cases} \frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \overrightarrow{grad}(P) + \nu \Delta \vec{u} \\ div(\vec{u}) = 0 \end{cases} \quad (3.61)$$

La première de ces deux équations, c'est à dire l'équation des moments, est une écriture du principe fondamental de la dynamique avec ρ remplaçant la masse. On pourrait la réécrire sous la forme :

$$\rho \frac{\partial \vec{u}}{\partial t} = \eta \Delta \vec{u} - \overrightarrow{grad}(P) \quad (3.62)$$

Les deux premiers termes de cette équation constituent une équation de diffusion, dont le coefficient est la viscosité dynamique η . On peut ainsi assimiler ce facteur η à la vitesse de propagation des valeurs du champ de vitesse dans le corps.

L'ordre de grandeur de η est tel que les mesures de l'objet que nous modélisons sont négligeables devant la vitesse de propagation du champ de vitesse (cf. Annexe A). Nous adopterons donc, pour la suite, une hypothèse selon laquelle le champ de vitesse s'établit instantanément dans le corps, et le coté dynamique de la modélisation peut être négligé.

Nous pouvons nous ramener ainsi à un modèle de la forme :

$$\boxed{\begin{cases} \Delta \vec{u} = \frac{1}{\eta} \overrightarrow{\text{grad}}(P) \\ \Delta P = 0 \end{cases}} \quad (3.63)$$

Ce système est très intéressant car formé de deux équations de Poisson, ce qui signifie qu'il est totalement linéaire et que le principe de superposition peut être utilisé.

Parce que le temps n'intervient plus dans l'équation des moments, une partie des instabilités liées au traitement numérique des équations différentielles partielles ne nous posera pas problème. De plus, pression et vitesse sont régies par le même type d'équation, ce qui nous permettra d'utiliser les mêmes méthodes numériques pour les deux problèmes.

Les équations de Poisson font partie de ce que l'on appelle les "Boundary value problems", c'est à dire les problèmes de conditions au limites. Nous allons donc voir quelles valeurs nous devons imposer aux bornes de l'objet afin de déterminer le champ de vitesse.

3.4.3 Les conditions aux limites

La résolution des équations de Poisson 3.63 nécessite de définir les conditions aux limites des champs de pression et de vitesse. Il existe ici trois types de conditions aux limites, chacun associé à un type d'interface.

1. Interfaces Matériau / Obstacle rigide

Les obstacles rigides sont caractérisés par leur vitesse nulle et par le fait qu'ils empêchent le matériau de s'écouler dans les voxels qu'ils occupent. Les limites du volume de travail, par exemple, en font partie afin d'éviter que la matière ne puisse sortir de l'espace de calcul.

Etant donné que l'obstacle est inamovible, et ce quelle que soit la force qu'on lui applique, on peut utiliser le principe d'action-réaction : les pressions aux bords et dans l'obstacle sont les mêmes. On a ainsi une condition de type Neumann :

$$\frac{\partial P}{\partial \vec{n}} = 0 \quad (3.64)$$

où \vec{n} est le vecteur normal à la surface de l'obstacle.

La condition portant sur la vitesse provient d'une condition de non-pénétration : le fluide ne peut pas entrer dans l'obstacle. On a donc :

$$\vec{u} \cdot \vec{n} = 0 \quad (3.65)$$

2. Surfaces libres

La surface libre, comme son nom l'indique, est la surface du matériau qui n'est pas soumise à une action de l'outil ou de tout autre corps solide. Il s'agit en fait d'une interface Air / Matériau.

La pression est en équilibre avec la pression atmosphérique. On a donc une condition aux bornes de type Dirichlet :

$$P = 0 \quad (3.66)$$

La dynamique de l'air étant considérée comme négligeable devant celle du matériau, on n'impose pas de condition particulière sur le champ de vitesse.

3. Interfaces Matériau / Outil

C'est au travers de ce type de condition aux limites que va se faire l'interaction outil-matière. Le matériau, en déplacement, entraîne avec lui la matière à la même vitesse. On a donc :

$$\vec{u} = \vec{u}_{Outil} \quad (3.67)$$

En ce qui concerne la pression appliquée par l'outil sur le corps, nous pourrions la calculer comme étant le rapport de la force appliquée sur l'aire de contact. Nous ne disposons cependant pas de la mesure de la force. Nous avons donc décidé que la pression serait assimilée à une constante P_{Outil} choisie par l'utilisateur.

$$P = P_{Outil} \quad (3.68)$$

Maintenant que le modèle matière et les conditions aux limites sont énoncés, le modèle proprement dit est complet. La discrétisation sous forme de voxels se fait suivant le placement présenté pour le premier modèle (cf. figure 3.3). Nous allons maintenant nous pencher sur la méthode utilisée pour la résolution numérique des équations de Poisson.

3.4.4 Résolution numérique des équations de Poisson

Si de nombreuses méthodes existent pour la résolution des équations de Poisson, dont certaines sont même très efficaces, très peu sont réellement adaptées à nos besoins. En effet, les caractéristiques de notre problème sont les suivantes :

- **la forme du domaine de résolution est quelconque**, puisque c'est celle de l'objet modelé,
- **les conditions aux limites sont hétérogènes** : conditions de Neuman et de Dirichlet,

- la résolution doit être la plus rapide possible.

Parmi les méthodes de résolution rapides, on peut distinguer deux grandes familles : les méthodes multigrilles et les méthodes basées sur les transformées de Fourier rapides.

Le succès des méthodes multigrilles repose sur l'utilisation de plusieurs niveaux de discrétisation sur chacun desquels on construit un schéma discret. Ainsi, pour traiter un problème de taille N , on se ramène à un problème de taille $N/2$ et ainsi de suite récursivement. Des opérateurs de transfert permettent ensuite de faire remonter la solution d'une grille grossière vers une grille plus fine en se basant sur des opérations de lissage. Ces méthodes sont très efficaces en terme de complexité, mais malheureusement inadaptées à des problèmes dont la topologie est quelconque comme c'est notre cas. Les méthodes basées sur transformées de Fourier rapide, souffrent de la même faiblesse.

Nous nous sommes donc orientés vers des méthodes plus simples, basées sur les différences finies. Ces méthodes se prêtent en effet beaucoup mieux aux conditions aux limites imposées par la modélisation physique. Même si elles ne sont pas très rapides, elles sont faciles à implémenter et à comprendre. Celle que nous avons choisi est la méthode de Jacobi.

La méthode de Jacobi fait partie des méthodes dites "de relaxation". Elle permet de résoudre une équation elliptique du type :

$$\mathcal{L}u = \tilde{\rho} \quad (3.69)$$

où \mathcal{L} est un opérateur elliptique, u le champ que l'on désire déterminer et $\tilde{\rho}$ le terme source.

L'idée de base est de réécrire ce problème comme une équation de diffusion :

$$\frac{\partial u}{\partial t} = \tilde{\rho} - \mathcal{L}u \quad (3.70)$$

En laissant évoluer temporellement le champ u , celui-ci converge vers la solution qui nous intéresse lorsque $t \rightarrow \infty$.

Cette démarche, dans le cas d'une équation de Poisson 2D, se base sur un schéma de différentiation de type FTCS (Forward Time Centered Space), qui amène à :

$$u_{i,j}^{t+1} = u_{i,j}^t + \frac{\delta t}{\delta h} (u_{i+1,j}^t + u_{i-1,j}^t + u_{i,j+1}^t + u_{i,j-1}^t - 4u_{i,j}^t) - \delta t \tilde{\rho}_{i,j} \quad (3.71)$$

Ce schéma n'étant stable que si la condition de Courant est remplie :

$$\frac{\delta t}{\delta h} \leq \frac{1}{4} \quad (3.72)$$

Pour obtenir une convergence la plus rapide possible, sans instabilité, on se place alors dans le cas :

$$\delta t = \frac{\delta h}{4} \quad (3.73)$$

Le schéma numérique devient alors :

$$u_{i,j}^{t+1} = u_{i,j}^t + \frac{1}{4} (u_{i+1,j}^t + u_{i-1,j}^t + u_{i,j+1}^t + u_{i,j-1}^t) - \frac{\delta h}{4} \tilde{\rho}_{i,j} \quad (3.74)$$

Ce qui nous ramène à une résolution locale simple où il est facile d'intégrer les conditions aux limites.

3.4.5 Résultats et Conclusion

La méthode n'a pour le moment été testée que dans le cas 2D pour lequel elle donne des résultats assez corrects (figure 3.23). L'implémentation dans le cas 3D est en cours.

D'ores et déjà, nous pouvons espérer que la déformation gagne en réalisme : les bords d'attaque au niveau de l'outil sont plus doux et la matière semble avoir gagné une certaine cohésion grâce à la simulation de la viscosité.

L'approche volumique (ou surfacique dans le cas 2D) semble donc tout a fait justifiée : même si elle est couteuse en temps de calcul, elle est nécessaire afin de conserver le réalisme physique de la déformation en intégrant par les conditions aux bornes la géométrie du corps modelé.

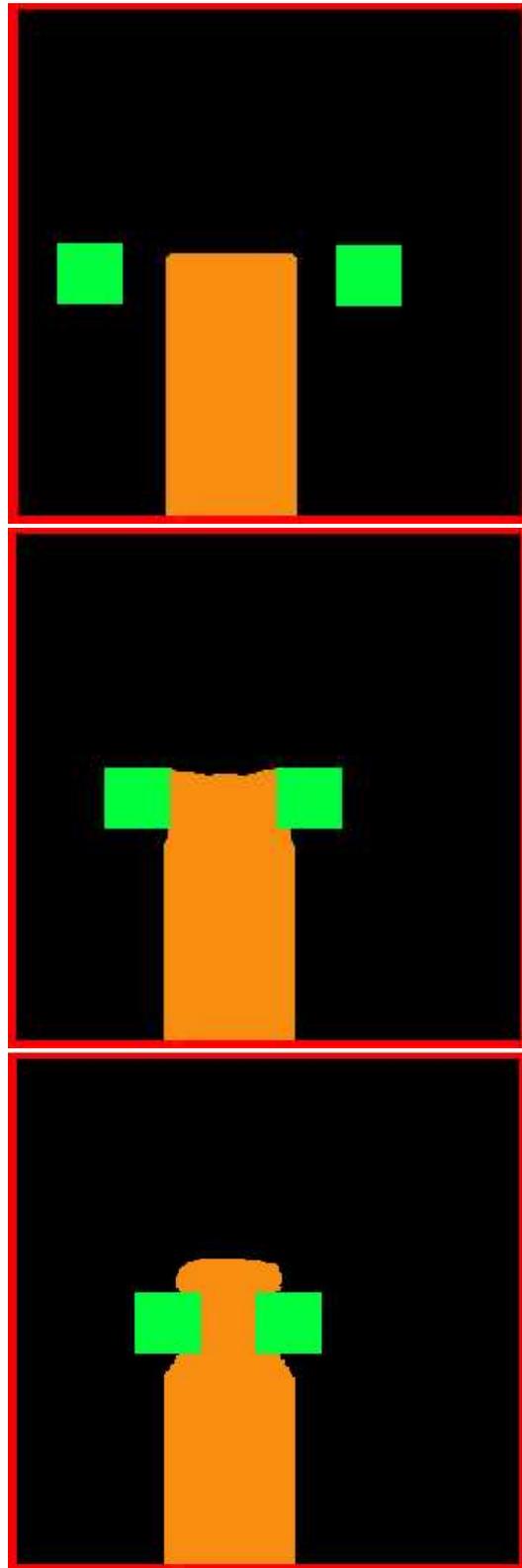


FIG. 3.23 – Compression d'un bloc de matière par deux outils (en vert)

3.5 Bilan de la modélisation

Partant des équations régissant le comportement des fluides, et en se basant sur une hypothèse d'écoulement rampant, nous avons établi dans ce chapitre trois modèles matière : les deux premiers n'intégrant que le coté incompressible du matériau, et le dernier intégrant à la fois viscosité et incompressibilité.

L'étude des deux premiers modèles nous a permis de démontrer qu'une approche volumique est absolument nécessaire à un modèle basé sur la physique des déformations viscoplastiques. Si les conditions aux bornes imposées par les différentes interfaces du corps ne sont pas prises en compte, la déformation n'est pas réaliste car elle ne tient pas compte de la géométrie de ce corps.

Cette approche volumique se paie bien sûr par des algorithmes dont les complexités sont fortement reliées à la taille de l'espace de simulation. Elles sont dans le cas général d'ordre 3. Ceci est un inconvénient majeur puisque notre objectif est d'obtenir une simulation interactive.

Cette complexité cubique est fort heureusement compensée par le fait que les procédures de résolution des équations de comportement s'expriment sous des formes locales et permettent de mettre en place des solutions admettant une très forte parallélisation. Les différentes méthodes étudiées pour résoudre ces équations elliptiques (Automates cellulaires, méthode de Jacobi) se prêtent sans aucun problème à des implémentations sur machines multiprocesseurs.

Chapitre 4

Vers un recalage basé voxel du modèle matière

4.1 Introduction

L'axe directeur de ce travail est, comme nous l'avons dit dans le chapitre 1, la conception d'une machine d'apprentissage où l'artiste modèle à la fois un véritable bloc de matière et un bloc de matière virtuelle. La figure 4.1 illustre ce principe.

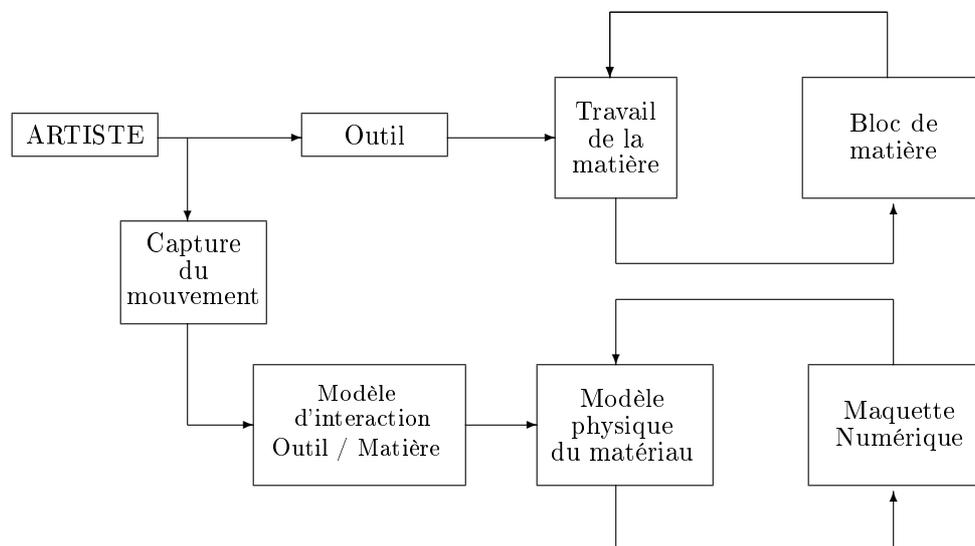


FIG. 4.1 – Deux maquettes : l'une réelle et l'autre virtuelle

On peut distinguer sur cette figure les deux chaînes d'action : en haut, la chaîne d'action concernant le modèle réel, en bas celle concernant le modèle numérique. Grâce à un procédé de capture du mouvement (position, vitesse, force), le modèle virtuel est censé subir les mêmes actions que le modèle réel. On ne peut cependant pas garantir, en utilisant ce schéma, que le modèle virtuel reste à tout instant en correspondance avec le modèle réel en raison des imperfections qui existent dans la modélisation.

Certes, les modèles numériques de déformation qui ont été présentés jusque là sont basés sur la physique des corps viscoplastiques, ce qui leur confère un comportement cohérent avec celui que l'on peut observer sur un véritable matériau. Cependant, ils ne traduisent pas parfaitement tout ce qui se passe à l'intérieur de ce dernier. L'un des phénomènes ignorés dans la modélisation est par exemple l'influence de la température. Celle-ci influe de manière importante sur la valeur de la viscosité et donc sur la déformation globale. Toute personne ayant utilisée de la pâte à modeler sait qu'il faut la pétrir quelques temps avant qu'elle ne devienne plus souple : les changements de température et d'hygrométrie affectent la structure microscopique du matériau et donc son comportement.

Même dans l'hypothèse où un modèle physique absolument exhaustif serait à notre disposition, on ne dispose pas de la valeur exacte de la viscosité ou de la densité, mais seulement d'un ordre de grandeur. Quoi que l'on fasse, le comportement du modèle virtuel finira toujours par diverger de celui du modèle réel en raison de ces inconnues.

4.2 Recalage de deux grilles voxel

4.2.1 Le mécanisme de recalage

Puisqu'il est apparemment impossible d'obtenir un modèle virtuel qui évolue exactement comme le modèle réel, il est important de prévoir un mécanisme de rattrapage d'erreur afin de corriger le modèle virtuel. Un moyen envisageable pour effectuer cette correction est d'utiliser le modèle réel comme référence :

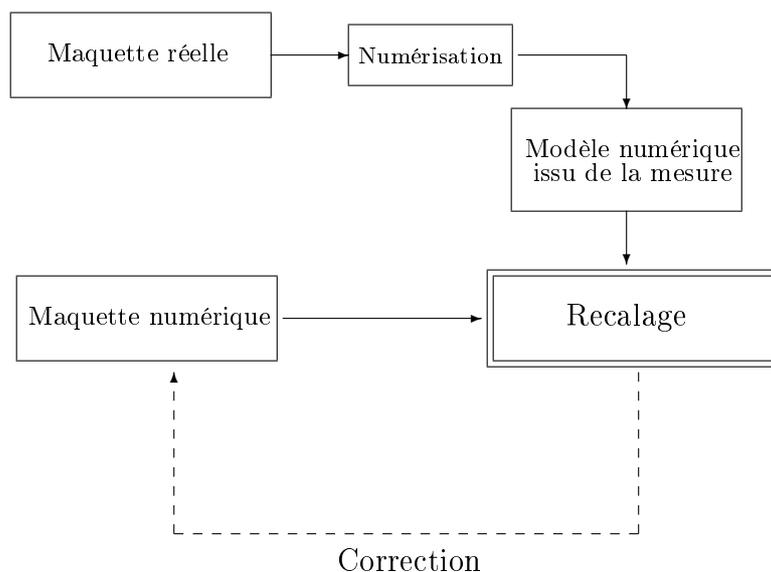


FIG. 4.2 – Correction des erreurs du modèle virtuel

La figure 4.2 illustre ce mécanisme. A intervalles réguliers, un dispositif d'acquisition (scanner 3D) permet de numériser automatiquement la maquette réelle afin d'en extraire

un modèle géométrique. Ce second modèle numérique est ensuite utilisé afin de corriger le premier : c'est cette opération de correction que nous désignons sous le terme de "recalage".

Ce dispositif d'acquisition permet aussi d'initialiser le modèle matière au début du processus. Une numérisation du bloc de matière brute est effectuée avant de commencer le modelage comme illustré sur la figure 4.3

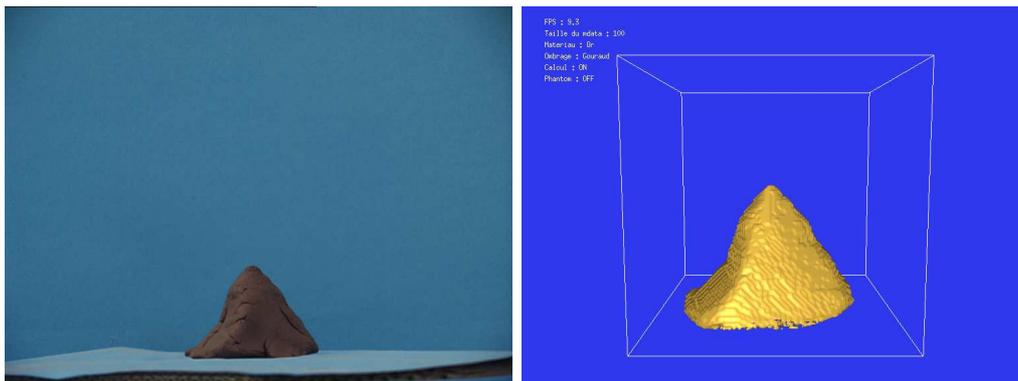


FIG. 4.3 – Initialisation du modèle matière en début de modelage

4.2.2 Les difficultés d'un recalage de deux grilles voxel

Une question légitime que l'on peut se poser est celle de la nécessité réelle d'une correction. En effet, si l'on dispose d'un modèle mesuré, pourquoi s'encombrer d'une correction et ne pas prendre comme nouveau modèle virtuel celui donné par la numérisation ?

En pratique, les modèles issus de la numérisation ne sont pas des modèles parfaits. On distingue plusieurs technologies pouvant servir à l'acquisition 3D, chacune ayant ses avantages et ses inconvénients :

- **Les capteurs vidéos** sont les moins chers, mais aussi les moins précis en terme de mesure. Les principes utilisés pour la reconstruction 3D à partir d'images reposent sur des modèles numériques où la caméra est assimilée à un opérateur projectif depuis l'espace de travail 3D vers l'image 2D. Les zones très concaves de l'objet qui n'apparaissent pas en raison d'occultations ne peuvent pas être traitées par ces scanners.
- **Les capteurs laser** sont plus coûteux, mais aussi beaucoup plus précis que les capteurs vidéos. Là encore, certaines concavités ne sont pas accessibles à la mesure, surtout lorsque celle-ci est effectuée de manière automatique.
- **Les capteurs mécaniques**, très utilisés en métrologie (palpage) sont une solution qui nécessite un travail beaucoup trop long pour l'opérateur. En effet, il faut en général mesurer manuellement un grand nombre de points afin d'obtenir un modèle

géométrique utilisable, ce qui les exclut des solutions envisageables.

- **Les capteurs volumiques**, très utilisés en médecine (tomographie, échographie 3D, IRM), sont capables de donner une vue parfaite de l'ensemble du corps, et ceci avec une précision très grande. En raison d'un volume de travail très restreint, il nécessitent cependant une machinerie encombrante afin de positionner précisément le capteur.

En conclusion, les méthodes accessibles sont les méthodes basées sur des capteurs vidéos ou laser, qui fournissent des modèles hétérogènes en terme de qualité. **Dans les zones convexes du corps, elles donnent de très bon résultats. Dans les zones concaves, par contre, le modèle issu de la numérisation ne sera pas une référence fiable en raison des occultations.**

Un autre problème qui se pose au niveau du recalage est celui de **l'hétérogénéité des représentations entre maquette issue du modèle physique et maquette issue de la numérisation**. Alors que le modèle simulé est un modèle basé sur une grille voxel, le modèle mesuré peut être un nuage de points surfaciques ou un modèle basé lui aussi sur une grille voxel. Dans le premier cas, plusieurs algorithmes existants permettent d'en extraire une représentation sous forme de voxels (segmentation volumique), ce qui nous ramène dans tous les cas à un recalage de deux grilles voxels.

La modélisation par une grille de voxels étant une représentation implicite du corps, elle est fortement liée à la structure, à la position et à l'orientation de la lattice qui lui sert de support. Un même corps voxélisé suivant deux lattices non alignées donnera donc deux modèles différents. Ce **problème d'alignement des lattices** est à prendre en compte dans le recalage.

Enfin, un dernier problème est celui de la **résolution du capteur utilisé**. Comment cette imprécision se traduit-elle lors du passage à une représentation voxel ?

Nous avons apporté un début de réponse à ces problèmes en étudiant et en mettant en œuvre un capteur particulier : un scanner vidéo basé sur le principe du "Shape from Silhouette".

4.3 Etude de cas : Capteur shape from silhouette

4.3.1 Principe du Shape from Silhouette

La plupart des scanners, qu'ils soient laser ou vidéo, sont basés sur le principe de la stéréovision. Il s'agit d'une reproduction du système visuel humain qui consiste à voir une même scène sous au moins deux points de vue différents. Comme l'illustre la figure 4.4, la différence entre ces incidences permet alors d'établir la profondeur de la scène.

Le principe de ces capteurs est celui de la triangulation : la projection 2D d'un point 3D est sur le rayon formé par le centre optique (i.e. C_1 ou C_2) de la caméra et son image

sur l'écran m_1 ou m_2 . Si l'on a deux images de différents points de vue du même point M , celui-ci se trouve à l'intersection des deux rayons.

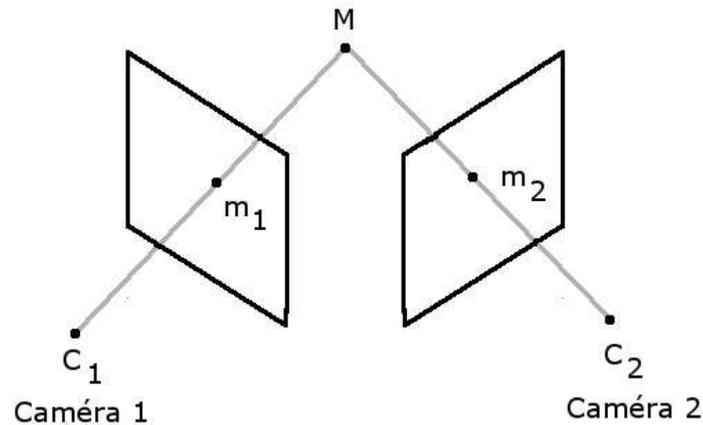


FIG. 4.4 – Principe de la stéréovision

Quatre étapes sont nécessaires pour réaliser une reconstruction de surfaces à l'aide de cette approche :

- **Le calibrage des systèmes d'acquisition**, c'est-à-dire l'obtention des paramètres physiques permettant d'identifier les caméras à des opérateurs mathématiques de projection.
- **L'acquisition des images par les caméras.**
- **La mise en correspondance des images.** Cette étape sert à retrouver, dans chaque paire d'images, les points m_1 et m_2 correspondant à un même point M de l'espace 3D.
- **La reconstruction.** A partir des points d'appariement des images stéréographiques et des paramètres physiques du système d'acquisition, le principe de triangulation peut être appliqué. Un nuage de points tridimensionnels de la scène reconstituée est obtenu.

Deux problématiques sont liées à la reconstruction 3D de scènes avec un capteur stéréoscopique. La première est la quantité de données à traiter. La seconde est la complexité et l'incertitude de l'étape d'appariement. En effet, connaissant la position du point m_1 , il faut être en mesure de lui faire correspondre automatiquement le point m_2 (cf. fig 4.4). Les algorithmes de « mise en correspondance » développés sont généralement très sensibles aux bruits, aux déformations introduites par les optiques, ainsi qu'à l'éclairage.

La technique que nous allons utiliser pour effectuer l'acquisition 3D est en général considérée comme plus simple à mettre en place et plus robuste, car basée sur l'emploi

d'une seule caméra (Elle est également l'une des plus populaires si l'on considère l'ensemble des articles ayant trait à cette approche). En terme de matériel, elle ne nécessite qu'une caméra numérique, ainsi qu'un plateau rotatif sur lequel est placé l'objet à numériser.

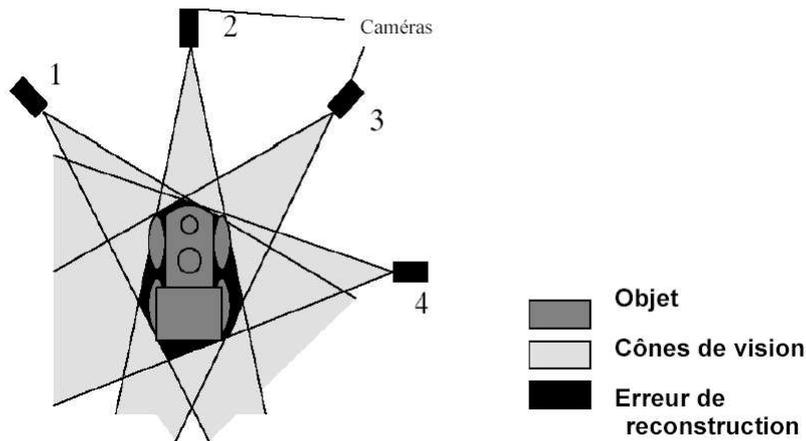


FIG. 4.5 – Principe d'un capteur "Shape from silhouette"

Comme illustré par la figure 4.5, l'idée de base de l'algorithme du « Shape from Silhouette » est extrêmement simple. Supposons que nous disposions de plusieurs clichés, notés c , de l'objet. Pour chacun de ces clichés c nous pouvons extraire la silhouette, notée S_c , correspondant aux pixels de l'image appartenant à l'objet. Le modèle 3D est alors obtenu à partir des silhouettes S_c de l'objet par calcul de l'intersection volumique des cônes de vision, notée Ψ_{max} , ayant pour sommets les centres optiques des caméras et pour génératrices les silhouettes S_c .

Supposons que nous ayons K images d'un objet Ψ et considérons l'ensemble

$$\{S_k, k = 1, \dots, K\}$$

des silhouettes de l'objet Ψ figurant sur ces K images.

Admettons provisoirement que chacune des caméras ayant servi à prendre une photographie de l'objet Ψ réalise une transformation de l'espace 3D sur les images 2D et que nous connaissions avec exactitude les caractéristiques de ces K transformations.

Notons $\Pi_k, k = 1, \dots, K$ ces transformations avec

$$\begin{aligned} \Pi_k : \mathbb{R}^3 &\rightarrow \mathbb{R}^2 \\ P &\mapsto \Pi_k(P) \end{aligned}$$

où $\Pi_k(P)$ désigne les coordonnées 2D de l'image du point P dans la photographie obtenue par la $k^{ième}$ caméra.

Avec ces notations remarquons que :

$$\forall P \in \Psi, \forall k \in [1, K], \Pi_k(P) \in S_k \quad (4.1)$$

Ce qui est équivalent à :

$$\forall k \in [1, K], \Psi \subseteq \Pi_k^{-1}(S_k) \quad (4.2)$$

ou encore :

$$\Psi \subseteq \bigcap_{k=1}^K \Pi_k^{-1}(S_k) \quad (4.3)$$

Cette intersection est une portion de l'espace 3D, notée Ψ_{max} , qui se définit comme l'intersection volumique des cônes de vision de sommets les positions des K caméras et de génératrice les silhouettes S_k . En particulier, tout objet Ψ est inclus dans le Ψ_{max} qu'il génère.

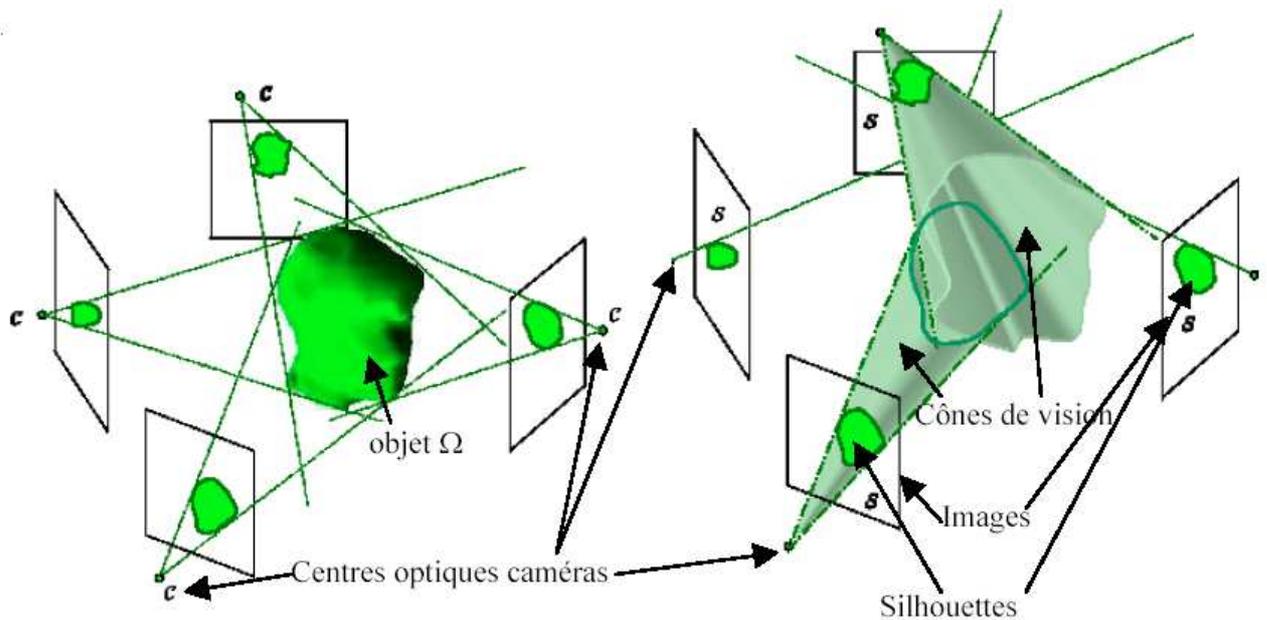


FIG. 4.6 – Les cônes de vision obtenus à partir des silhouettes

4.3.2 Dispositif expérimental et implémentation

Le dispositif utilisé afin d'effectuer la numérisation 3D est le suivant :

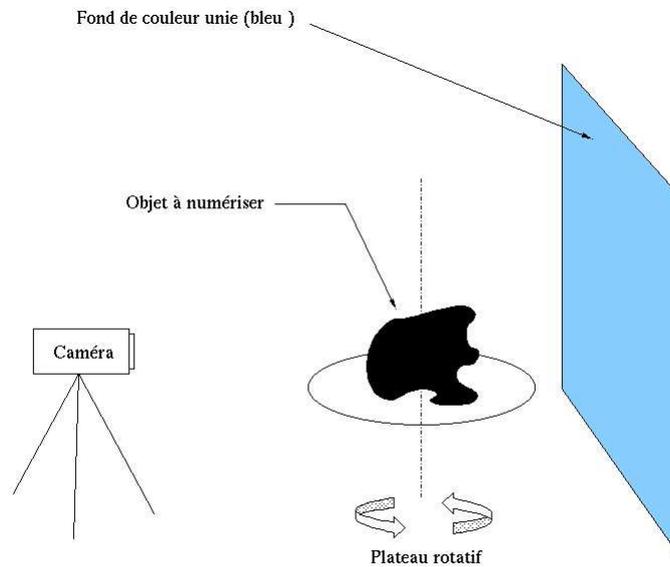


FIG. 4.7 – Plateforme expérimentale

L'objet est placé sur un support rotatif dont le positionnement peut être contrôlé de manière très précise. Un fond bleu est utilisé afin de faciliter l'extraction des silhouettes.

La première étape de l'acquisition est l'identification de l'opérateur de projection $\Pi(P)$ sur lequel est basée la construction du modèle tridimensionnel. On désigne en général cette identification sous le nom de "calibration de la caméra". Nous avons choisi comme modèle d'opérateur projectif un modèle sténopé, aussi connu sous le nom de modèle "pinhole". Il est illustré en figure 4.8.

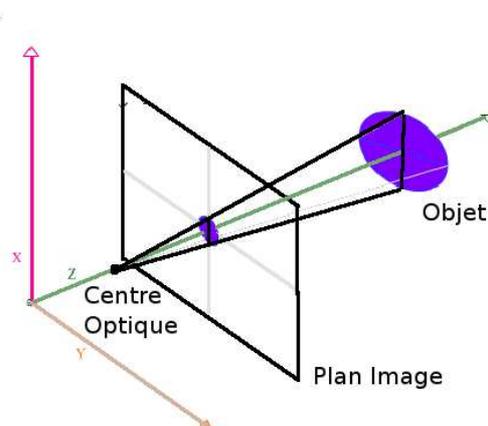


FIG. 4.8 – Modèle pinhole

Calibration de la caméra

Le mécanisme de formation de l'image est simple. A chaque point P de la scène 3D est associé un point image p , obtenu par une projection centrale sur un plan (le "plan image" ou "rétine"). Le centre de cette projection est le point O , que l'on appelle "centre optique" de la caméra.

Différents paramètres sont propres à chaque caméra : on les appelle paramètres "intrinsèques" du modèle. Il s'agit de :

- **La distance focale f** , c'est-à-dire la distance euclidienne du centre optique au plan image.
- **La position du point principal**, c'est-à-dire de l'origine du repère 2D lié au plan image. Ces deux facteurs permettent de rendre compte du fait que la matrice sensible de la caméra n'est en général pas parfaitement alignée avec le système optique.
- **Deux facteurs de déformation linéaires**, exprimant le fait que l'élément d'image, le pixel, n'est pas carré mais rectangulaire. Ceci se traduit par deux facteurs d'échelle si l'on désire se ramener à un système métrique.
- **Des facteurs de déformation** (déformation en coussinet, etc.) qui permettent d'intégrer les non-linéarités dues à la sphéricité des optiques.

Enfin, il faut aussi identifier les paramètres extrinsèques de la caméra, c'est à dire les paramètres définissant position et orientation du repère caméra dans le repère de l'espace de travail.

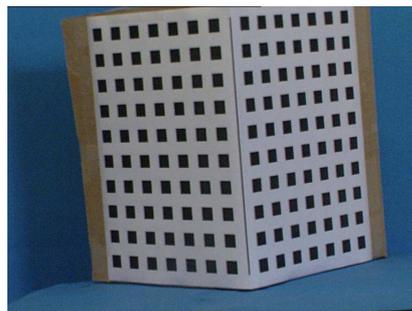


FIG. 4.9 – Photo de la mire

Afin d'identifier ces différents paramètres, on place sur le plateau de numérisation une mire géométrique. Pour chacun des points que l'on extrait de l'image de cette mire (sur la figure 4.9 les coins des carrés), on connaît à la fois la position dans le repère image et dans le repère 3D. Grâce à ces données, des méthodes d'estimation des paramètres ([Fau93], [AAK71]) permettent d'obtenir une expression numérique du modèle de la caméra. Le système est alors prêt pour effectuer l'acquisition.

Extraction des silhouettes

Après chaque incrément de rotation du plateau, le système prend une photo de la scène.

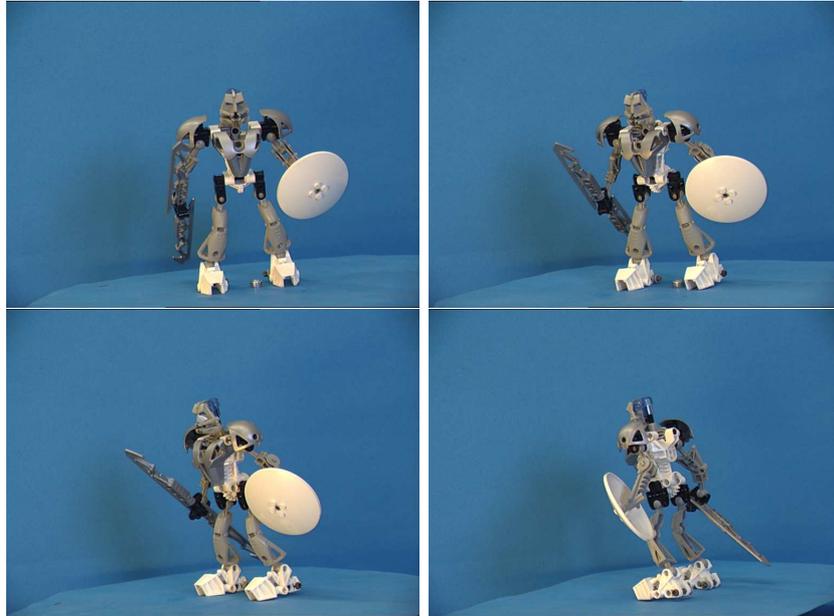


FIG. 4.10 – Images de l'objet prises durant la numérisation

Grâce à l'utilisation du fond de couleur bleu unie, il est relativement aisé de séparer les pixels appartenant à l'objet des pixels du fond. Cette segmentation utilise un codage IHS (Intensity, Hue, Saturation) qui présente l'avantage par rapport au codage classique RGB (Red, Green, Blue) de séparer la teinte (composantes H et S) de l'intensité (composante I).

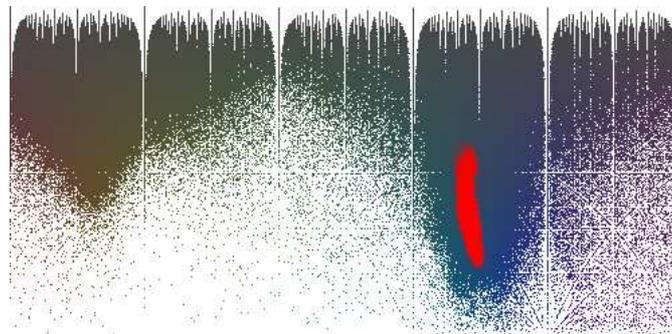


FIG. 4.11 – Espace des couleurs présentes dans la première image

La figure 4.11 montre dans un espace Hue x Saturation les couleurs de la première image. La zone colorée en rouge correspond aux pixels appartenant au fond. L'utilisateur sélectionne au début de l'acquisition quelques zones appartenant au fond afin d'avoir un modèle de ces teintes pour permettre ensuite la reconnaissance du fond dans l'image.

L'obtention des silhouettes peut alors se faire de manière automatique : pour chaque pixel d'une image, on calcule les composantes H et S de la couleur. Si le point appartient à la zone correspondant au fond, alors il est coloré en noir. Dans le cas contraire, il est coloré en blanc. On obtient ainsi les images suivantes :

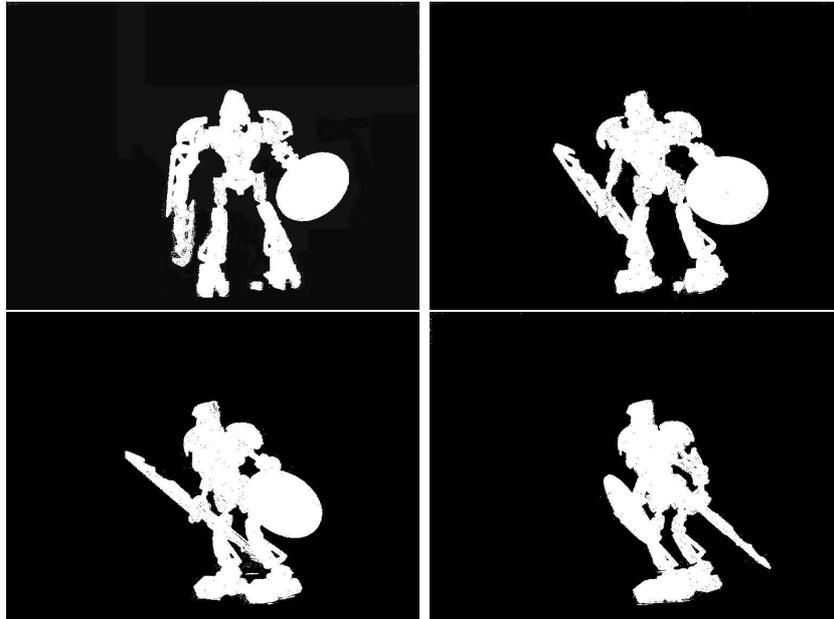


FIG. 4.12 – Images après segmentation

Une fois ces deux étapes effectuées, les silhouettes sont disponibles et il ne reste plus qu'à calculer le volume 3D correspondant.

Élévation du modèle 3D

Un espace voxel est alors défini en utilisant la même lattice que celle supportant le modèle matière. Chacun des voxels est initialisé à 1, ce qui signifie qu'il est rempli.

Le modèle 3D de l'objet numérisé est obtenu en projetant sur chacune des images les centres de chaque voxel grâce à l'opérateur Π_k obtenu par calibration de la caméra.

Si le point image obtenu est un point appartenant à la silhouette, alors ce voxel ne change pas de valeur. Dans le cas contraire, le voxel ne respecte pas la contrainte imposée par l'équation 4.3 : il est noté comme n'appartenant pas au corps par la valeur 0.

Une analogie simple consisterait à dire que l'on "coupe" tout ce qui dépasse des silhouettes pour chacune des photos.

Le "shape from silhouette" étant une méthode largement documentée, nous invitons le lecteur intéressé par cette méthode à se reporter à [Zhe94] ou [NW97] pour une description plus détaillée des algorithmes.

4.3.3 Performances du système d'acquisition

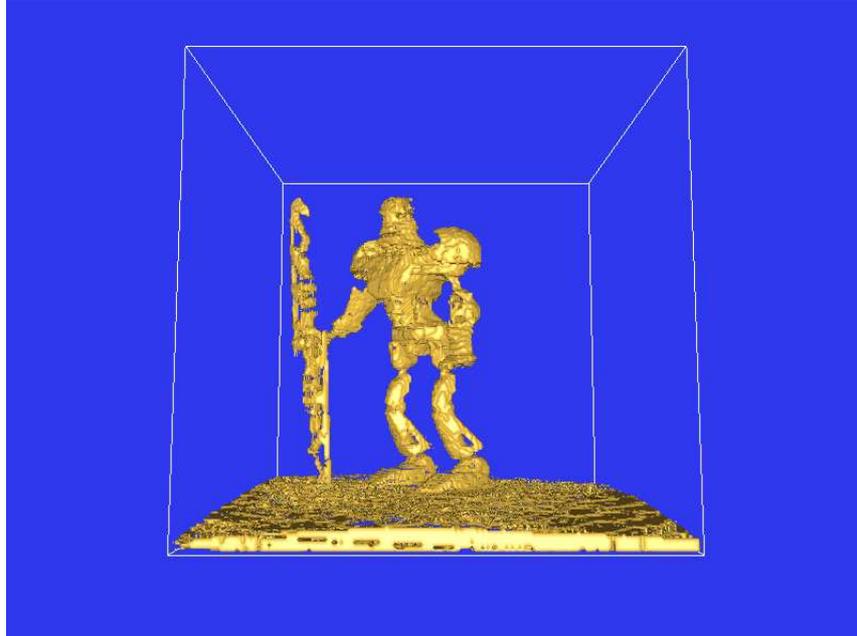


FIG. 4.13 – Modèle 3D obtenu par le scanner

La figure 4.13 montre le résultat de la numérisation du robot illustré figure 4.10 avec une lattice de taille $\mathcal{N} = 200$. La précision obtenue est de l'ordre du millimètre, ce qui est cohérent avec les besoins auxquels le système doit répondre.

Actuellement, le temps nécessaire à une numérisation est de l'ordre de quelques minutes pour une acquisition composée de 360 photos (une par degré). Ce délai assez long est principalement dû au matériel utilisé pour le plateau rotatif, qui tourne assez lentement. Une adaptation de ce plateau devrait permettre de réduire ce temps à environ une minute.

Cet outil de numérisation nous est utile durant deux étapes du travail :

- Durant la phase d'initialisation du modèle matière, car il permet de numériser le bloc de matière brute avant le modelage.
- Régulièrement durant le modelage, afin de corriger le modèle matière en fonction du bloc de matériau réellement modelé.

Plusieurs caractéristiques de ce capteur en font un système particulièrement intéressant dans le cadre de notre application :

1. **L'utilisation d'une représentation volumique** : Ce scanner est, tout comme les modèles physiques, basé sur une représentation volumique de l'objet numérisé. Ceci nous dispense de l'étape de passage d'une représentation surfacique à une représentation volumique qui pourrait se traduire par l'introduction de sources d'imprécisions

supplémentaires.

2. **L'utilisation d'une rétroprojection** : évite les "trous" qui pourraient apparaître dans le cas d'une méthode basée sur le lancer de rayons. Si la silhouette est de bonne qualité (pas de parasites), le modèle volumique est bien construit.
3. **Le positionnement de la lattice** sur laquelle se base la représentation voxel peut être imposée par l'utilisateur.

Le modèle résultant de l'acquisition n'est cependant pas une image exacte du corps numérisé, mais seulement une enveloppe convexe de sa forme. C'est là que réside la principale faiblesse de ce capteur car **il n'apporte aucune information sur les zones concaves de la forme**. La figure 4.14 montre cette limitation dans le cas 2D.

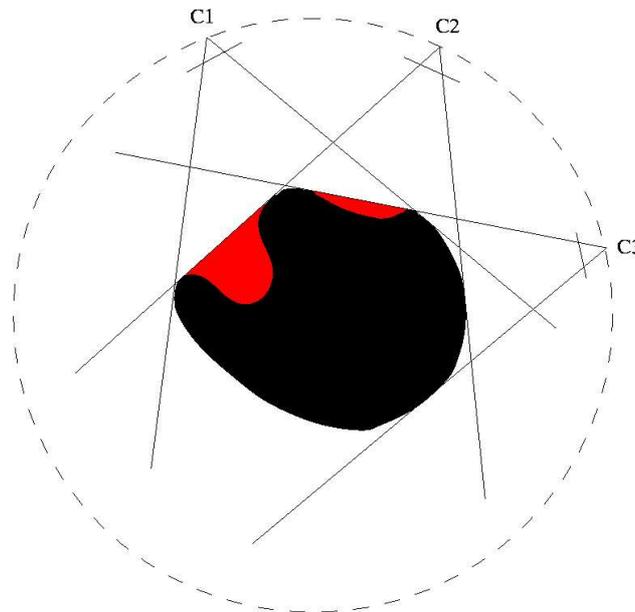


FIG. 4.14 – Le problème des concavités

Si le corps (ici en noir) présente des zones concaves, celles-ci n'influent pas sur le contour de la silhouette. Or c'est précisément sur ce contour qu'est basée la forme géométrique du corps reconstruit.

Dans ce cas précis, il n'est pas possible d'utiliser le modèle issu de la numérisation comme nouveau corps de départ pour la modélisation physique car il est incomplet. Le modèle mesuré est censé être plus proche de la forme de la maquette réelle dans les zones convexes, mais la seule information dont nous disposons concernant les zones concaves est celle apportée par le modèle physique. Le recalage devra incorporer cette contrainte.

4.3.4 Extraction des régions d'intérêt

L'idée du recalage est de permettre de corriger le modèle matière en fonction de la forme du bloc modelé. Le problème inhérent à un capteur basé sur le Shape From Silhouette est que la précision de la numérisation n'est pas homogène : comme nous l'avons expliqué ci-dessus, les zones concaves ne peuvent pas être obtenues avec celui-ci.

Les régions d'intérêt sont donc différentes selon qu'il s'agisse du modèle obtenu par simulation ou du modèle obtenu par acquisition. L'approche développée dans cette section a pour objet de définir une coloration des voxels appartenant à la surface du corps, c'est à dire pouvoir déterminer pour chaque voxel de surface s'il fait partie d'une zone concave ou convexe au sens du scanner.

L'algorithme de coloration concave/convexe se base sur une approche 2D/3D du problème, par un découpage du modèle en "sections" horizontales comme illustré sur la figure suivante :

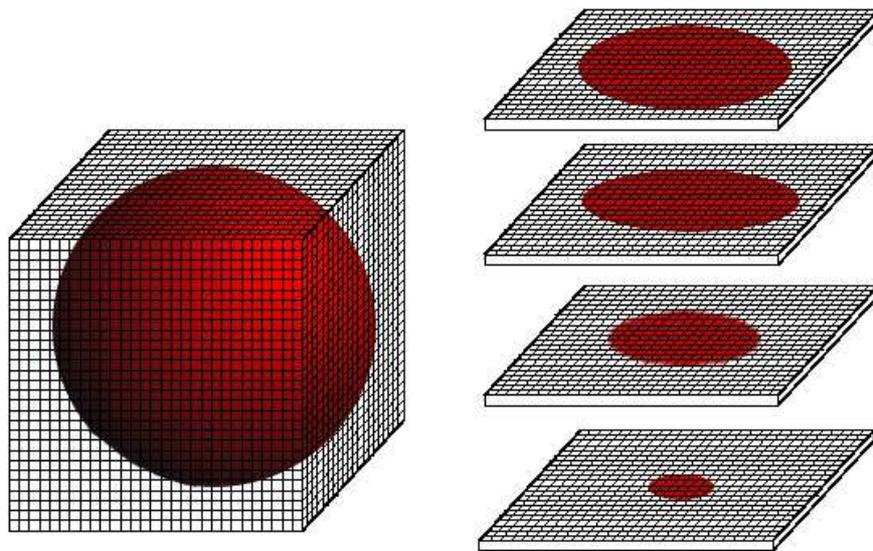


FIG. 4.15 – Découpage de l'espace voxel en tranches horizontales

Chacune de ces coupes est ensuite traitée de façon indépendante comme une image 2D. Le problème de coloration suivant la concavité ou convexité devient alors un problème portant non plus sur des voxels, mais maintenant sur des pixels. Une fois la coloration obtenue pour chacune des coupes, elles seront recomposées afin de reformer une forme 3D.

Détection des zones concaves en 2D

L'algorithme de segmentation 2D consiste à faire évoluer un polygone afin de le faire tendre vers l'enveloppe convexe Ψ_{Max} . Ce polygone peut être assimilé à un "élastique" que l'on tendrait autour de la forme 2D dont on veut connaître les concavités. Là où l'élastique adhère à la surface, le corps est convexe. Là où l'élastique s'en sépare, le corps est concave.

L'algorithme est initialisé avec un polygone formé des quatre extrema de la coupe 2D. Ces quatre points sont appelés points Nord (coordonnée maximale en x), Sud (coordonnée minimale en x), Est (Coordonnée maximale en y) et Ouest (coordonnée minimale en y). Chacun d'entre eux est doté d'un vecteur correspondant à la normale au contour au point considéré. (voir fig 4.16).

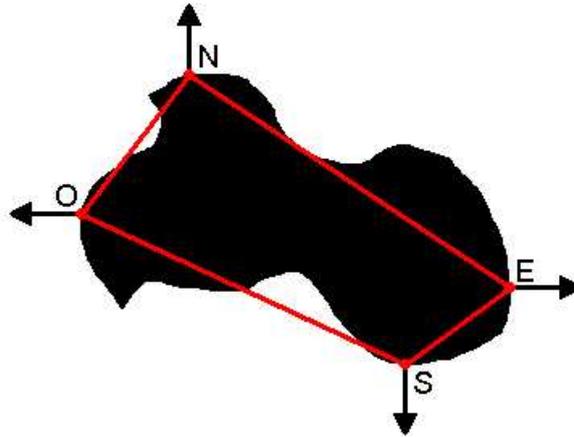


FIG. 4.16 – Recherche des extrema Nord, Sud, Est et Ouest

L'algorithme de segmentation cherche à déformer ce polygone initial jusqu'à obtenir l'enveloppe convexe de la forme. Chaque arête est décomposée et on y insère de nouveaux points jusqu'à ce que le polygone la contienne complètement. Afin d'expliquer le processus de déformation du polygone, on va considérer l'arête définie par le point N et le point E.

Un vecteur V est calculé par la somme des vecteurs portés par les deux extrémités de l'arête. Ce vecteur donne la direction de recherche. Pour trouver un nouveau point à insérer dans le polygone, on parcourt l'image en partant de l'arête [NE] et en suivant la direction donnée par ce vecteur V .

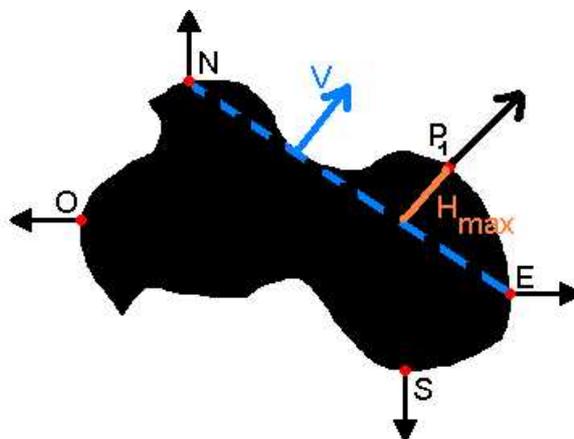


FIG. 4.17 – Insertion d'un nouveau point dans le polygone

Le nouveau point à introduire dans le polygone est le point, appartenant à la forme, dont la distance H à l'arête $[NE]$ est la plus grande. Sur la figure 4.17, ce point est noté P_1 . Le vecteur V devient alors la normale issue du point P_1 et celui-ci est ajouté au polygone.

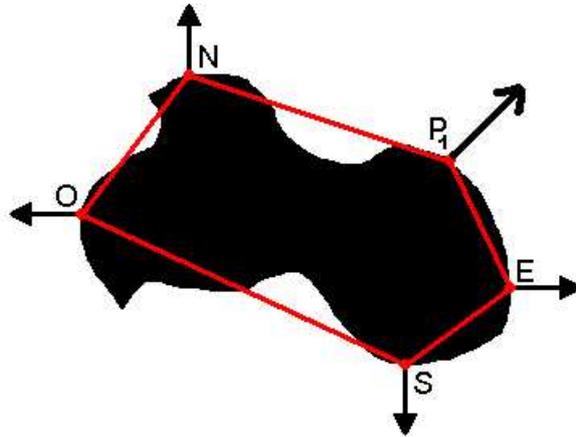


FIG. 4.18 – Le nouveau polygone

On réitère ensuite ce procédé de manière récursive pour chacune des arêtes jusqu'à ce qu'il ne soit plus possible d'insérer de nouveau point ou jusqu'à ce qu'une arête soit composée de deux points voisins. La situation finale est illustrée par la figure 4.19

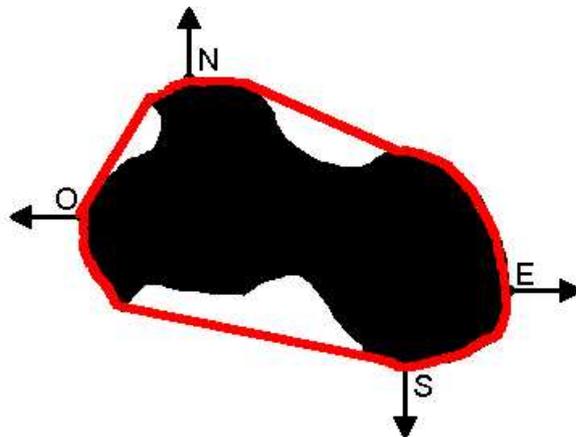


FIG. 4.19 – Le polygone final

Dans cette situation, on peut distinguer deux classes de pixels de surface :

- **Les pixels de surface appartenant au polygone**, qui font partie des zones dites "convexes" et pour lesquels le modèle issu de la mesure est fiable.
- **Les pixels de surface n'appartenant pas au polygone**, qui font parties des zones dites "concaves" et pour lesquels on ne dispose pas de données autres que

celles issues du modèles physique.

Ces deux classes sont finalement illustrées sur la figure 4.20

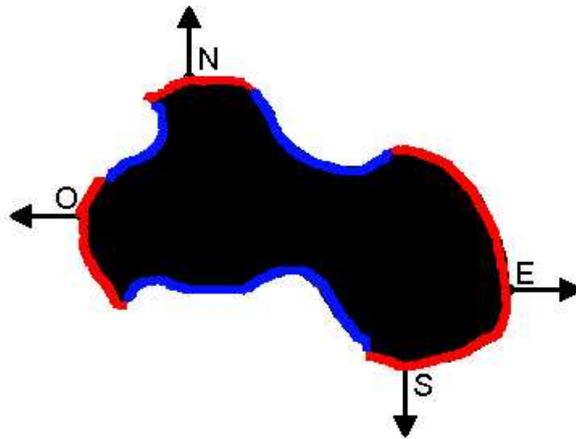


FIG. 4.20 – Coloration finale des voxels de surface

Une fois ce traitement effectué sur chacune des tranches horizontales, on les rassemble afin de recomposer l'espace voxel (cf figure 4.21).

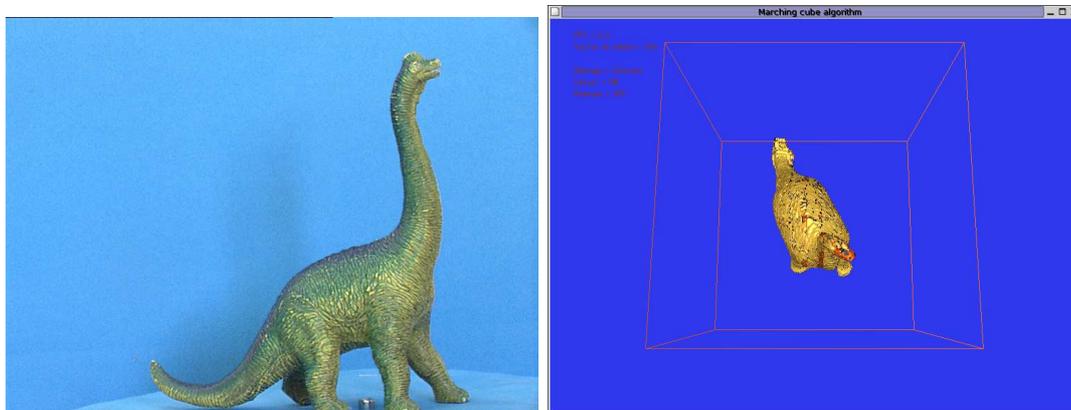


FIG. 4.21 – Modèle réel et résultat de la coloration 3D

4.4 Conclusion

Le choix d'utiliser à la fois un matériau réel et un modèle matière dans le système d'aide au modelage impose l'utilisation d'un procédé de numérisation, et ceci pour deux raisons. La première est l'initialisation du modèle matière. La numérisation nous permet de disposer de la forme du matériau brut en début de modelage. La seconde est la possibilité de "corriger" les dérives du modèle matière en effectuant une procédure de recalage.

Le système d'acquisition présenté est basé sur le principe du "Shape from Silhouette", qui permet une numérisation rapide et automatisée. La précision obtenue, de l'ordre du millimètre, est suffisante pour notre application. L'inconvénient de ce capteur est son manque de précision dans les zones concaves, qui n'apparaissent pas sur la silhouette et ne peuvent donc pas apparaître sur le modèle 3D final.

La conséquence de cette observation est que chacun des deux modèles, à savoir le modèle matière et le modèle obtenu par numérisation, comportent des informations intéressantes. Alors que le modèle issu de l'acquisition est très précis sur les zones convexes, les informations éventuelles concernant les concavités ne peuvent être obtenues que grâce à la simulation. Ce constat nous a amené à nous intéresser à un algorithme de coloration en zones concaves et convexes, afin de pouvoir se fixer un critère indiquant quel modèle suivre durant le recalage.

Le recalage en lui-même n'a pu être abordé dans ce travail en raison du temps imparti. Il s'agit d'un problème de recalage non rigide complexe, auquel la méthode des Snakes peut peut-être apporter un début de réponse. Les modalités de sa mise en pratique restent cependant à définir.

Chapitre 5

Conclusion

Les systèmes d'aide au geste dédiés au modelage artistique ne peuvent pour le moment s'affranchir d'un véritable matériau. Il existe en effet une très forte relation entre le modelleur et la matière travaillée : tout d'abord parce que cette matière sert de support à l'oeuvre de l'esprit, ensuite parce que les informations sensorielles qu'en retire l'acteur métier sont nécessaires à sa tâche. Les systèmes actuels permettant la synthèse haptique, que ce soient les bras maîtres ou les systèmes de toucher, sont encore insuffisants pour permettre au modelleur un véritable contrôle de la déformation qu'il fait subir au matériau. De plus, priver l'acteur métier de ce support physique revient à lui ôter la maîtrise dimensionnelle qu'il a durant le processus.

Ce travail s'est donc inscrit dans une démarche qui consiste à allier modelage classique et utilisation de l'outil informatique. L'acteur métier, qui manipule un véritable bloc de matière, dispose ainsi des avantages d'un modèle numérique pour l'aider dans sa tâche. Le principe repose sur une capture du geste qui permet la mise à jour d'un modèle matière virtuel. Ceci suppose donc de disposer à la fois d'un modèle physiquement réaliste du matériau utilisé (glaise, plasticine, etc.), et d'un système permettant la remontée des informations depuis le modèle réel vers le modèle simulé. Ainsi, régulièrement, l'acteur métier doit interrompre son travail afin de lancer une étape de "recalage" du modèle virtuel sur le modèle réel afin de les faire correspondre.

Les enjeux scientifiques sont donc doubles : réussir à traduire le comportement physique du matériau dans un modèle numérique de déformation utilisable dans des contraintes d'interactivité, et concevoir un système d'acquisition et de recalage permettant le passage de l'espace de travail réel à l'espace de simulation.

Les matériaux utilisés en modelage, qui sont des composés à base de liquides (eau, huile) et de particules en suspension, ont un comportement à mi-parcours entre celui d'un solide et celui d'un liquide : ils sont qualifiés de "viscoplastiques". On peut les assimiler à des fluides à seuil, c'est à dire des fluides qui ne s'écoulent que lorsque les contraintes qu'on leur applique dépassent un certain seuil critique. Ils sont de plus caractérisés par leur très forte viscosité. Un état de l'art concernant les modèles de déformation utilisés habituellement en infographie nous a montré qu'ils reposent le plus souvent sur la théorie de l'élasticité linéaire ou sur une approche géométrique et intuitive. Ces modèles ne per-

mettent pas de représenter le comportement réel de tels matériaux et ne peuvent donc pas être utilisés ici. La modélisation de fluides très peu visqueux a par contre déjà fait l'objet de travaux basés sur des approches intégrant la mécanique des fluides. Dans le cas d'une très grande viscosité, les équations de Navier-Stokes se simplifient grâce à l'hypothèse des écoulements rampants. Cette hypothèse a permis le développement de trois modèles matière basés sur une approche volumique (espace voxel) de la déformation.

Le premier de ces trois modèles, n'intégrant qu'une notion d'incompressibilité du matériau, se base sur des automates cellulaires qui présentent l'avantage de se prêter très facilement à une implémentation parallèle. Par cette approche, le modèle bénéficie à la fois de la simplicité inhérente aux automates cellulaires et de temps de calcul très intéressants sur des machines multiprocesseurs, une implémentation multithread permettant de "compenser" le temps de calcul très important induit par l'approche volumique.

Le second modèle, quant à lui, représente une tentative pour "sauter" directement au résultat, sans passer par les étapes intermédiaires de calcul. Puisque la déformation a toujours lieu en surface du corps, l'idée était de court-circuiter le traitement volumique afin de se ramener à un traitement surfacique du corps modelé. Cette voie, qui semblait prometteuse en terme de temps de calcul, a abouti cependant à un modèle peu réaliste, puisque n'intégrant pas la géométrie du corps.

Le troisième modèle, enfin, intègre à la fois viscosité et incompressibilité, ce qui se traduit par un meilleur réalisme de la déformation, mais aussi par un algorithme plus coûteux en temps. Basé sur une méthode de Jacobi, il permet de conserver une approche locale de la déformation et donc des possibilités d'implémentation parallèle.

Il semble qu'un calcul volumique est pour le moment incontournable, cependant les travaux effectués sur le second modèle laissent espérer qu'une partie au moins du calcul (à savoir la répartition de matière) peut être rendue beaucoup plus rapide grâce à l'utilisation du théorème d'Ostrogradski. Si ces trois modèles apportent des résultats assez prometteurs, la modélisation physique en temps réel d'un corps viscoplastique reste encore un problème ouvert.

Quel que soit le modèle matière choisi, il finira par diverger du comportement de la matière réelle, c'est pourquoi il est nécessaire de renumériser le corps modelé à intervalles réguliers. Actuellement, la numérisation du corps est effectuée grâce à un modèle basé sur la méthode du Shape From Silhouette. Ce capteur présente l'avantage d'effectuer une numérisation automatique et rapide. Il fournit des données d'une précision suffisante dans les zones convexes, mais celles-ci sont généralement incorrectes dans les zones concaves. Le recalage des données virtuelles sur les données acquises par numérisation devra donc tenir compte de cette notion. La méthode de recherche des zones d'intérêt que nous avons développée n'est qu'une première étape dans la mise au point d'un processus de recalage, mais beaucoup de travail reste à faire.

Toutes les problématiques liées à ce travail ne sont donc pas résolues. Les perspectives à court terme sont bien sûr le développement pratique de ce recalage et la mise en

place d'un système complet intégrant toutes les différentes étapes que nous avons évoquées dans ce manuscrit. A plus long terme, le problème de la modélisation des corps pour des applications de réalité virtuelle ou augmentée peut encore faire l'objet de nombreux développements tant au niveau de la modélisation physique que de la résolution numérique des équations associées.

Bibliographie

- [AAK71] Y. I. ABDEL-AZIZ et H. M. KARARA : Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. Proceedings of the Symposium on Close Range Photogrammetry, 1971.
- [ACD⁺97] N. AYACHE, S. COTIN, H. DELINGETTE, J.M. CLEMENT, J. MARESCAUX et M. NORD : Simulation of endoscopic surgery. *Journal of Minimally Invasive Therapy and Allied Technologies (MITAT)*, 1997.
- [Bar84] A. BARR : Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–29, 1984.
- [Blo87] J. BLOOMENTHAL : Polygonization of implicit surfaces. Xerox Technical report CSL-87-2, Xerox, Mai 1987.
- [Blo88] J. BLOOMENTHAL : Polygonization of implicit surfaces. *In Computer Aided Geometric Design*, volume 5, pages 341–355, 1988.
- [BW98] D. BARRAF et A. WITKIN : Large steps in cloth simulation. *In SIGGRAPH 98 Conference Proceedings*, pages 43–54, 1998.
- [CD97] M-P. CANI et M. DESBRUN : Animation of deformable models using implicit surfaces. *IEEE Trans. on Visualisation and Computer Graphics*, 3(1), 1997.
- [CD03] A. CROSNIER et S. DRUON : Human centered system for computer aided replication of sculptures. *In IROS'03 : IEEE International Conference on Intelligent Robots and Systems*, pages 3751–3756, Las Vegas (USA), 2003.
- [CDA95] S. COTIN, H. DELINGETTE et N. AYACHE : New perspectives for realistic simulations of hepatic surgery. *In Actes du Congrès Mondial de Télémedecine*, Toulouse, France, décembre 1995.
- [CDA96] S. COTIN, H. DELINGETTE et N. AYACHE : Real time volumetric deformable models for surgery simulation. *In Visualization in Biomedical Computing, Proceedings*, volume 1131 de *Lecture Notes in Computer Science*. Springer Verlag Heidelberg-New York, septembre 1996.
- [CDA97] S. COTIN, H. DELINGETTE et N. AYACHE : Surgery simulation with visual and haptic feedback. *In Proceedings of the 8th International Symposium of Robotics Research*, Tokyo, Japan, octobre 1997.
- [CDC⁺96a] S. COTIN, H. DELINGETTE, J.-M. CLEMENT, V. TASSETTI, J. MARESCAUX et N. AYACHE : Volumetric deformable models for simulation of laparoscopic surgery. *In Proceedings of the International Symposium on Computer and*

Communication Systems for Image Guided Diagnosis and Therapy, Computer Assisted Radiology (CAR'96), volume 1124 de *International Congress Series*. Elsevier, juin 1996.

- [CDC⁺96b] S. COTIN, H. DELINGETTE, J-M. CLÉMENT, V. TASSETTI, J. MARESCAUX et N. AYACHE : Geometric and Physical Representations for a Simulator of Hepatic Surgery. *In Medecine Meets Virtual Reality IV*, Interactive Technology and the New Paradigm for Healthcare, pages 139–151. IOS Press, janvier 1996.
- [CG00] G. COUARRAZE et J.L. GROSSIORD : *Initiation à la rhéologie*. Editions TEC & DOC, 3 édition, 2000.
- [CK98] CHEN et KAUFMAN : Physically based animation of volumetric objects. *Computer Animation 98*, pages 154–160, 1998.
- [Coq90] S. COQUILLARD : Extended free-form deformation : a sculpturing tool for 3d geometric modeling. Rapport technique 1250, INRIA Rocquencourt, Juin 1990.
- [Cra87] C. CRAWFORD : "minimization of directed points generated in three-dimensional dividing cubes method". United States Patent Number : 4,885,688, Nov 1987.
- [Cre99] B. CRESPIN : Implicit free-form deformations. *In Proceedings of implicit surface*, 1999.
- [CRE01] E. COHEN, R. RIESENFELD et G. ELBER : *Geometric Modeling with Splines : An Introduction*. AK Peters, Ltd., 2001. ISBN 1-56881-137-3.
- [CRG⁺02] G.M. CRISCI, R. De ROSA, S. Di GREGORIO, R. RONGO et W. SPATARO : Pyroclastic flow simulation by a cellular automatic model. *In International Symposium on Interaction between Volcanoes and their Basement and Related Geological Hazards*, 2002.
- [Cro03] A. CROSNIER : Skill gesture assisted by computer for rapid prototyping. application to the replication of sculptures. Rapport technique, LIRMM, 2003.
- [CTLA03] C. COTTEREAU, I. THOUVENIN, C. LENAY et D. AUBERT : User perception analysis of 3d worlds throught the case of virtual reality for blind people. *In Virtual Concept 2003*, 2003.
- [DAK04] C. DURIEZ, C. ANDRIOT et A. KHEDDAR : A multi-threaded approach for deformable/rigid contacts with haptic feedback. *In HAPTICS 2004*, pages 272–279, 2004.
- [DC95] M. DESBRUN et M.P. CANI : Animating soft substances with implicit surfaces. *In SIGGRAPH 95*, 1995.
- [DC99] G. DEBUNNE et M.P. CANI : Animation multirésolution interactive d'objets déformable. *In Journées AFIG*, Nov 1999.
- [DC03] S. DRUON et A. CROSNIER : Virtual sculpture : A model of virtual clay. *In VSMM'03 : 9th International Conference on Virtual Systems and Multimedia*, pages 604–609, Octobre 2003.

- [DC04] S. DRUON et A. CROSNIER : Physics based models for viscoplastic bodies applied to free form modelling with virtual clay. *In Workshop VIA 2004 5Virtual Concept*, Compiègne (France), 2004.
- [DCB03] S. DRUON, A. CROSNIER et L. BRIGANDAT : Efficient cellular automata for 2d/3d free-form modeling. *In Journal of WSCG (Winter School of Computer Graphics)*, volume 11, pages 102–108, 2003.
- [dCL00] F. Boux de CASSON et C. LAUGIER : Simulating 2d tearing phenomena for interactive medical surgery simulators. *In Proceedings of Computer Animation*, pages 113–123, 2000.
- [DDCB01] G. DEBUNNE, M. DESBRUN, M.P. CANI et A. BARR : Dynamic real-time deformations using space and time adaptive sampling. *In Computer Graphics Proceedings, Annual Conference Series*. ACM Press / ACM SIGGRAPH, Aug 2001. Proceedings of SIGGRAPH'01.
- [Deb00] G. DEBUNNE : *Animation multirésolution d'objets déformables en temps-réel. Application à la simulation chirurgicale*. Thèse de doctorat, INPG, 2000.
- [Del98] H. DELINGETTE : Towards realistic soft tissue modeling in medical simulation. *Proceedings of the IEEE : Special Issue on Surgery Simulation*, 1998.
- [DSB99] M. DESBRUN, P. SCHRODER et A. BARR : Interactive animation of structured deformable objects. *In Graphics Interface 99*, 1999.
- [DSCP94a] H. DELINGETTE, G. Subsol S., COTIN et J. PIGNON : A Craniofacial Surgery Simulation Testbed. *In Visualization in Biomedical Computing (VBC'94)*, octobre 1994.
- [DSCP94b] H. DELINGETTE, G. SUBSOL, S. COTIN et J. PIGNON : A craniofacial surgery testbed. Rapport technique 2199, I.N.R.I.A., Sophia-Antipolis, France, février 1994.
- [DTC96] M. DESBRUN, N. TSINGOS et M.P. CANI : Adaptive sampling of implicit surfaces for interactive modelling and animation. *Computer Graphics Forum*, 15(5):319–327, 1996.
- [Fau93] O. FAUGERAS : *Three-Dimensional Computer Vision*. MIT Press, 1993.
- [FHP86] U. FRISCH, B. HASSLACHER et Y. POMEAU : Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, 56:1505–1508, 1986.
- [FM96] N. FOSTER et D. METAXAS : Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5), September 1996.
- [FP02] J.H. FERMIGIER et M. PERIC : *Computational methods for fluid Dynamics*. Springer, 3 édition, 2002. ISBN 3-540-42074-6.
- [Gal99] J. GALLIER : *Curves and Surfaces in Geometric Modeling : Theory and Algorithms*. Morgan Kaufmann, 1999. ISBN 1558605991.
- [GHP01] E. GUYON, J.P. HULIN et L. PETIT : *Hydrodynamique physique*. CNRS Editions - EDP Sciences, 2001.
- [HC87] W. LORENSEN H. CLINE : "system and method for the display of surface structures contained within the interior region of a solid body". United States Patent Number : 4,710,876, Dec 1987.

- [Her03] J.M. HERVOUET : *Hydrodynamique des écoulements à surface libre - Modélisation numérique avec la méthode des éléments finis*. Presse de l'Ecole Nationale des Ponts et Chaussées, 2003. ISBN 2-85978-379-2.
- [HHK92] W. M. HSU, J. F. HUGUES et H. KAUFMAN : Direct manipulation of free-form deformations. *In SIGGRAPH 92*, pages 177–184, 1992.
- [HPdP73] J. HARDY, Y. POMEAU et O. de PAZZIS : Time evolution of two-dimensional model systems in invariant states and time correlation functions. *J. Math. Phys.*, 14:1746–1759, 1973.
- [HPH96] D. HUTCHINSON, M. PRESTON et T. HEWITT : Adaptive refinement for mass/spring simulation. *In 7th Eurographics Workshop on Animation and Simulation*, pages 31–45, Poitiers, France, 1996.
- [JL96] A. JOUKHADAR et C. LAUGIER : Dynamic simulation : Model, basic algorithms, and optimization. *In In Proc. of the Workshop on Algorithmic Foundation on Robotics*, Toulouse, juillet 1996.
- [Jou96] A. JOUKHADAR : Adaptive time step for fast converging dynamic simulation system. *In Proc. of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 418–424, Novembre 1996.
- [KdMD⁺02] A. KRUPA, M. de MATHELIN, C. DOIGNON, J. GANGLOFF, G. MOREL, L. SOLER, J. LEROY et J. MARESCAUX : Automatic 3-d positioning of surgical instruments during robotized laparoscopic surgery using automatic visual feedback. *In T. DOHI et R. KIKINIS, éditeurs : Int. Conf. on Medical Image Computing and Computer-Assisted Intervention, MICCAI'02*, volume 2488 de *Lecture Notes in Computer Science*, pages 9–16, Tokyo, Japon, September 2002.
- [KT96] A. KAISS et P. Le TALLEC : Modélisation numérique du contact oeil-trépan. *Revue Européenne des Eléments Finis*, 5(3):375–408, 1996.
- [LC87a] W. LORENSEN et H. CLINE : Marching cubes : A high resolution 3d surface construction algorithm. *In Computer Graphics (Proceedings of SIGGRAPH '87)*, volume 21-4, pages 163–169, 1987.
- [LC87b] W. LORENSEN et H. CLINE : Marching cubes : A high resolution 3d surface construction algorithm. *In Computer Graphics (Proceedings of SIGGRAPH 87)*, volume 21-4, pages 163–169, 1987.
- [LCK⁺00] A. LECUYER, S. COQUILLART, A. KHEDDAR, P. RICHARD et P. COIFFET : Pseudo-haptic feedback : Can isometric input devices simulate force feedback? *In IEEE Int. Conf. on Virtual Reality*, pages 83–90, 2000.
- [LKC⁺01] A. LECUYER, A. KHEDDAR, S. COQUILLART, L. GRAUX et P. COIFFET : A haptic prototype for the simulations of aeronautics mounting/unmounting operations. *In IEEE ROMAN'2001*, pages 18–21, Bordeaux-Paris, France, 2001.
- [Lom96] J.C. LOMBARDO : *Modélisation d'objets déformables avec un système de particules orientées*. Thèse de doctorat, IMAGIS / IMAG, Janvier 1996.
- [McC76] K. McCLAY : The rheology of plasticine. *Tectonophysics*, 33:T7–T15, 1976.

- [MM99] A. MURTA et J. MILLER : Modelling and rendering liquids in motion. *In WSCG 99*, pages 194–201, 1999. ISBN 80-7082-490-5.
- [MMT97] L. MOCCOZET et N. MAGNENAT-THALMANN : Dirichlet free-form deformations and their application to hand simulation. *In Proceedings of computer animation*, pages 93–102, 1997.
- [NW97] W. NIEM et J. WINGBERMSUHLE : Automatic reconstruction of 3d objects using a mobile monoscopic camera. *In International Conference on Recent Advances in 3D Imaging and Modeling*, pages 173–180, May 1997.
- [Pic01] G. PICINBONO : *Modèles géométriques et physiques pour la simulation d'interventions chirurgicales*. Thèse de doctorat, INRIA Sophia Antipolis, Février 2001.
- [Pro95] X. PROVOT : Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface*, pages 147–154, 1995.
- [QDC⁺95] G. QUATREHOMME, H. DELINGETTE, S. COTIN, G. SUBSOL, M. FIDRICH, P. BAILET, G. GREVIN et A. OLLIER : A fully tridimensional method for facial reconstruction based on deformable models. *In International Association for Craniofacial Identification*, novembre 1995.
- [Ree83] W. REEVES : Particle systems - a technique for modeling a class of fuzzy objects. *Computer Graphics*, 17(3):359–376, 1983.
- [RFC⁺02] J. REHG, M. FLAGG, T. CHAM, R. SUKTHANKAR et G. SUKTHANKAR : Projected light displays using visual feedback. *In Proceedings of the International Conference on Control, Automation, Robotics, and Vision*, December 2002.
- [RFCG04] R. BLANCH, E. FERLEY, M.P. CANI et J.D. GASCUEL : Non-realistic haptic feedback for virtual sculpture. Rapport technique RR-5090, INRIA, U.R. Rhone-Alpes, January 2004. Projets ARTIS et EVASION, theme 3.
- [SP86] T. SEDERBEG et S. PARRY : Free-form deformations of solid geometric models. *Computer Graphics*, 20(4):151–160, 1986.
- [Sta03] J. STAM : Real-time fluid dynamics for games, 2003.
- [Wei97] J. WEIMAR : Cellular automata for reaction–diffusion systems. *Parallel Computing*, 23(11):1699–1715, 1997.
- [Wei02] J. WEIMAR : Three-dimensional cellular automata for reaction-diffusion systems. *Foundations of Computer Science*, 2002.
- [WH94] A. WITKIN et P. HECKBERT : Using particles to sample and control implicit surfaces. *In Proceedings of SIGGRAPH 94*, pages 269–278, 1994.
- [WKM98] R. WCISLO, J. KITOWSKI et J. MO'SCI'NSKI : Cellular automaton as a fast tool for animation of liquid in multi-object scenes. *In 6th International Conference in Central Europe on Computer Graphics and Visualization (WSCG98)*, pages 417–423, 1998.
- [WMW86] G. WYVILL, C. MCPHEETERS et B. WYVILL : Data structure for soft objects. *The Visual Computer*, 2(4):1986, 1986.
- [Wol86] S. WOLFRAM : Cellular automaton fluids : Basic theory. *Journal of Statistical Physics*, 45:471–526, November 1986.

- [Zhe94] J. Y. ZHENG : Acquiring 3-d models from a sequence of contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):163–178, February 1994.

Annexe A

Rhéologie de la plasticine

Les données expérimentales concernant la glaise ou la pâte à modeler sont, à ma connaissance, quasi inexistantes. La plasticine, quant à elle, est un matériau bien connu des géologues car elle est couramment utilisée pour les simulations de mouvements tectoniques. Paradoxalement, les études rhéologiques de la plasticine sont elles aussi difficiles à obtenir : les données suivantes sont issues de [McC76], unique source de données que nous ayons pu trouver. L'auteur a étudié la rhéologie de trois types de plasticine :

- Une plasticine "tendre" (special soft)
- Une plasticine "dure" (special hard)
- Une pasticine "blanche" (standard white)

Pour chaque type de plasticine, des essais à température constante (25°C ou 45°C) et vitesse de déformation constante ont été réalisées dans des conditions de déformation unidimensionnelle (compression ou relaxation).

La composition exacte du matériau est inconnue car protégée par brevet. L'auteur indique qu'il s'agit d'un mélange d'huile, de cires, de poussières et d'une base minérale composée de calcite, de dolomite et de micro-fossiles dont la taille de grain est comprise entre $0.5 \mu m$ et $10 \mu m$.

Les éprouvettes de plasticine utilisées sont des blocs de $15 \times 15 \times 5$ cm et $5 \times 5 \times 5$ cm dans le cas de la plasticine dure, en raison de sa grande résistance. Les résultats des expériences montrent une bonne reproductibilité (erreur de 5%).

Les tests ont montré que pour tous les types de plasticine, il existe une loi non linéaire entre la contrainte et la vitesse de déformation. Le comportement peut être décrit comme caractéristique d'un fluide de Reiner-Rivlin.

La loi de déformation est du type :

$$\dot{\epsilon} = K e^{-\frac{Q}{RT}} \tau^N$$

où :

- $\dot{\epsilon}$ est la vitesse de déformation
- Q est une énergie d'activation
- R est la constante des gaz parfaits
- τ est la contrainte appliquée.

A température constante, cette loi devient :

$$\dot{\epsilon} = C\tau^N$$

Les résultats des tests de compression donnent des valeurs de N entre 6 et 9.

On peut également déduire des expériences des valeurs de viscosité de la plasticine pour des contraintes de 10% et des vitesses de déformation allant de $1.025 \cdot 10^{-3}$ à $8.21 \cdot 10^{-6} \text{ m.s}^{-1}$:

- Plasticine tendre : $4.4 \cdot 10^8$ à $2.7 \cdot 10^{10}$ Po
- Plasticine dure : $1.7 \cdot 10^{10}$ à $1.27 \cdot 10^{12}$ Po

Annexe B

Algorithme des Marching Cubes

Nous avons choisi de modéliser les corps sous la forme d'une représentation implicite de type voxel.

L'algorithme des Marching Cubes permet de calculer une isosurface d'un corps représenté sous la forme d'un champ scalaire. Nous présentons ici le principe de cet algorithme dans le cas 2D, en notant qu'il se généralise en 3D.

Principe en 2D

Approche tout ou rien

Nous considérons une grille régulière de points. Ces points représentent les centres des voxels, et contiennent ou non de la matière. Si l'on considère 4 points (formant un carré), plusieurs configurations ressortent. Elles sont au nombre de $2^4 = 16$, mais se réduisent par rotation ou symétrie à seulement 6 (figure B.1).

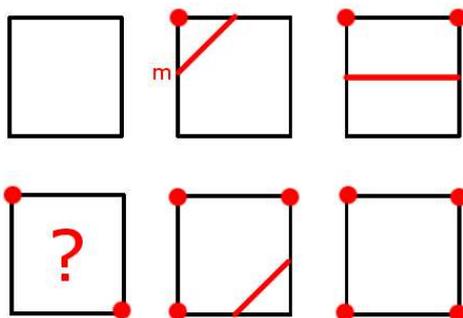


FIG. B.1 – Configurations types en 2D

Il est possible de calculer l'isosurface associée à chacune des configurations comme le montre la figure B.2. La position m de l'extrémité du segment de l'isosurface est ici arbitrairement fixé au milieu des deux voxels considérés.

Certains cas sont ambiguës et demandent un traitement spécifique que nous n'aborderons pas ici.

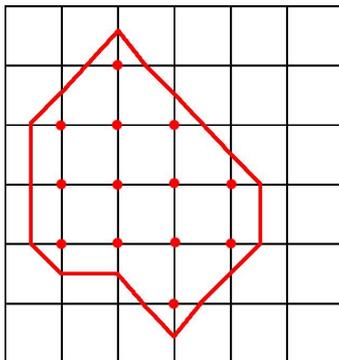


FIG. B.2 – Isosurface 2D calculée à partir de la grille de points

Approche continue

Le modèle tout ou rien fournit des résultats corrects mais il oblige à fixer un seuil arbitraire en-dessous duquel le voxel est considéré comme vide. Il est néanmoins possible d'affiner la position de l'isosurface en plaçant le point m au barycentre de masse des deux voxels considérés s'ils contiennent tous les deux de la matière.

Généralisation au cas 3D

Le même algorithme peut être appliqué à un champ scalaire à trois dimensions en considérant non pas un carré de points mais un cube (8 points). Evidemment le nombre de configuration est ici bien plus important ($2^8 = 256$), et se réduit à 15 configurations types par symétrie et rotation (voir figure B.3).

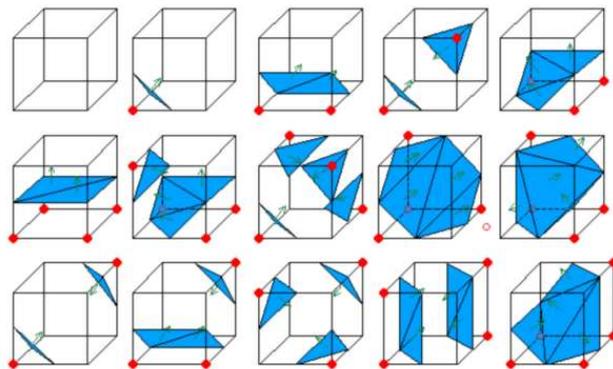


FIG. B.3 – Configurations types en 3D