

THÈSE

PRÉSENTÉE À
L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE
L'INGÉNIEUR

par Damien ANDRÉ

POUR OBTENIR LE GRADE DE DOCTEUR

Spécialité : **Mécanique et ingénierie**

MODÉLISATION PAR ÉLÉMENTS DISCRETS DES PHASES D'ÉBAUCHAGE ET DE DOUCISSAGE DE LA SILICE

Thèse dirigée par Ivan IORDANOFF

Co-dirigée par Jean-luc CHARLES et Jérôme NÉAUPORT

Soutenance prévue le 15 Mars 2012

Devant la commission d'examen formée de :

M. Frédéric DONZÉ, Professeur, 3S-R, Université de Grenoble	Rapporteur
M. Jérôme FORTIN, Professeur, INSSET, Université de Picardie	Rapporteur
M. Laurent BAILLET, Professeur, ISTerre, Université de Grenoble	Examineur
M. Olivier CAHUC, Professeur, I2M, Université de Bordeaux 1	Examineur
M. Jean-Luc CHARLES, Maître de Conférence, I2M, Arts et Métiers Paristech	Examineur
M. Ivan IORDANOFF, Professeur, I2M, Arts et Métiers Paristech	Examineur
M. Philippe LORONG, Professeur, PIMM, Arts et Métiers Paristech	Examineur
M. Jérôme NÉAUPORT, Docteur HDR, CEA-CESTA	Examineur

Remerciements

Ce projet a été réalisé au sein du laboratoire I2M dans le cadre du projet EFESO dirigé par le CEA-CESTA et soutenu par la région Aquitaine. Ainsi, je tiens à remercier mon directeur de thèse, Ivan Iordanoff, ainsi que mes co-directeurs, Jean-luc Charles et Jérôme Néauport, pour leurs confiances et leurs grandes implications. Je remercie également Cédric Maunier pour ses conseils et son aide.

Je remercie également les membres du Jury Philippe Lorong, Donzé Frédéric, Jérôme Fortin, Laurent Baillet et Olivier Cahuc d'avoir accepté d'évaluer mes travaux.

Je tiens également à remercier ma femme, Belen André, qui m'a soutenu et épaulé tout au long de ce parcours. Je pense aussi à mes enfants, Jehann et Éline, qui m'ont toujours donné le courage de me battre lors de moments difficiles.

Je remercie également mes collègues et amis de bureau, Lucie, Sébastien, Inigo, Younes et Nicolas pour leurs bonnes humeurs et leurs camaraderies.

Je remercie finalement l'ensemble des hommes et femmes du laboratoire, stagiaires, doctorants, post-doctorants, maîtres de conférence, professeurs, personnels techniques et administratifs qui participent à la vie du laboratoire et l'ont rendue agréable et conviviale.

Table des matières

Table des matières	iii
1 Contexte	1
1.1 Le laser Mégajoule	2
1.1.1 La fusion nucléaire par confinement inertiel	2
1.1.2 Les chaînes lasers	3
1.1.3 Les endommagements lasers	5
1.2 Les opérations de surfacage	7
1.3 Les micro fractures sous surfacique	10
1.4 Techniques expérimentales de mesure des SSD	10
1.4.1 Mesures destructives	10
1.4.2 Mesures non destructives	11
1.4.3 Mesure indirecte	11
1.5 Études expérimentales « procédés / SSD »	12
1.6 Apport de l'outil numérique	13
1.7 La silice, un matériau fragile ?	14
1.8 Cadre théorique, hypothèses	16
1.9 Méthodes continues	17
1.10 Méthodes discrètes	17
1.11 Choix de la méthode numérique	20
1.12 Les différentes méthodes discrètes	20
1.13 Mise en œuvre de la méthode DEM	21
1.14 Choix du code de calcul DEM	23
1.15 Synthèse, énoncé du plan	24
2 Plate-forme de calcul DEM	27
2.1 Algorithme de résolution dynamique explicite	28
2.1.1 Calcul des positions	28
2.1.2 Calcul des orientations	28
2.1.3 Vue synthétique de l'algorithme de résolution	29
2.2 L'approche orientée objet	29
2.2.1 Les notions d'objet et de classe : un modèle d'abstraction	30
2.2.2 Les relations	31
2.2.3 L'héritage	32
2.2.4 L'encapsulation	33
2.3 Choix du langage de programmation	34
2.4 Architecture logicielle de la plate-forme « GranOO »	35
2.4.1 La libGeometrical	36
Les entités géométriques et opérations courantes	36

	Les formes	37
	Performance	38
	Robustesse	39
	Accessibilité	40
	Synthèse	41
2.4.2	La bibliothèque libDEM	42
	Vue générale	42
	Les conteneurs intelligents de la <i>libDEM</i> : les <code>SetOf</code>	43
	Une gestion aisée des traitements : les greffons	45
	Enregistrement et chargement de domaine discret : les fichiers <i>gdd</i>	46
	Synthèse	47
2.5	Environnement de développement	48
2.5.1	Plate-forme et outils de développement	48
2.5.2	Hébergement du projet	48
2.6	Environnement d'exploitation	50
2.6.1	Téléchargement et déploiement	50
2.6.2	Quelques améliorations	50
2.6.3	Les étapes clés	51
2.7	Les fichiers nécessaires à l'élaboration d'une expérience DEM de traction	52
	Le fichier <code>Main.cpp</code>	52
	Les fichiers d'aides à la compilation : <code>Makefile</code> et <code>Makefile.EditMe</code>	54
	Le fichier de description d'un domaine discret : <code>Cylinder.gdd</code>	54
	Le fichier d'entrée : <code>Tension.inp</code>	54
2.7.1	Synthèse	55
2.8	Conclusion	56
3	Méthodologie de calibration	59
3.1	Modèle DEM hybride « lattice - particulaire »	61
3.2	Modèle de liaison cohésive <i>poutre</i>	62
3.3	Construction et validation du domaine compact initial	64
3.3.1	Méthodologie de compaction	64
3.3.2	Homogénéité	65
3.3.3	Isotropie	65
3.3.4	Raffinement	68
3.3.5	Synthèse	68
3.4	Calibration des paramètres élastiques	69
3.4.1	Présentation de l'essai numérique de traction	69
	Caractéristiques géométriques de l'éprouvette discrète	69
	Chargement quasi statique	70
3.4.2	Calibration du coefficient de Poisson microscopique ν_μ	72
3.4.3	Calibration du module de Young microscopique E_μ et du rayon adimensionné $\tilde{r}_\mu$	73
3.4.4	Validation	74
3.4.5	Étude de convergence	76
3.5	Calibration des paramètres dynamiques	78
3.5.1	Le problème de l'inertie	78
3.5.2	Validation sur des domaines discrets massifs	80

3.6	Calibration de la rupture	81
3.6.1	Critère microscopique de rupture	82
3.6.2	Présentation de l'essai de traction quasi statique à rupture	83
3.6.3	Calibration du paramètre microscopique de rupture σ_{μ_f}	85
3.6.4	Validation	86
3.6.5	Étude de convergence	87
3.6.6	Synthèse	88
4	Simulation du doucissage	91
4.1	Dispositif expérimental de référence	93
4.2	Résultats expérimentaux de référence	96
4.3	Description du modèle de doucissage DEM	96
4.3.1	Domaine d'étude et description générale	97
4.3.2	Modélisation des différents corps	97
	La silice	99
	Les particules d'usure	100
	Les abrasifs	101
	L'outil	101
	La solution aqueuse	101
4.3.3	Modélisation des interactions	102
	Le contact continu	102
	Le contact de Hertz	104
	Amortissement	105
	Interaction avec le fluide, contact visqueux	106
4.3.4	Conditions limites et chargements	107
	Fenêtre temporelle de simulation	108
	Modélisation de la précharge	108
	Modélisation de l'avance de l'outil	109
4.3.5	Méthodologies de quantification des paramètres du modèle	110
	Quantification de la précharge	111
	Quantification de la vitesse d'avance de l'outil	111
	Quantification de la taille des particules abrasives	112
	Quantification de la concentration en particules abrasives du <i>slurry</i>	112
4.4	Résultats numériques et discussions	113
4.4.1	Variables d'intérêt et méthodologie de post-traitement	114
	Variable d'intérêt	114
	Zones temporelles d'intérêt	114
	Méthodologie de mesure	115
4.4.2	Étude de l'expérience de référence	118
	Évolution de l'altitude de la paroi supérieure	118
	Évolution de l'effort normal appliquée à la paroi supérieure	120
4.4.3	Influence de la pression (précharge)	121
4.4.4	Influence de la vitesse de l'outil	124
4.4.5	Influence de la taille des abrasifs	126
4.4.6	Influence de la concentration en abrasif	129
4.4.7	Analyses des rugosités	132
4.5	Synthèse et critiques des résultats	136
4.6	Conclusion	137

5 Conclusion générale et perspective	139
Appendices	143
A De l'usage des quaternions en mécanique du solide	145
A.1 Introduction	145
A.2 Changement de norme : nombre scalaire	146
A.3 Changement de direction : versor	146
A.4 Définition d'un quaternion	146
A.5 Propriétés mathématiques	147
A.5.1 Addition	147
A.5.2 Multiplication	147
A.5.3 Forme conjuguée	148
A.5.4 Norme	148
A.5.5 Inverse	148
A.5.6 Quelques opérations remarquables	148
A.5.7 Quaternions unitaires et forme polaire	148
A.6 Quaternion et rotation	149
A.6.1 Démonstration	149
A.6.2 Relation entre quaternions et matrice de passage	150
A.7 Quaternion et vitesse de rotation	150
A.8 Application en dynamique du solide	152
A.8.1 Quaternion et angles d'Euler	152
A.8.2 Singularité des angles d'Euler	153
A.9 Synthèse	153
B Description du fichier d'entrée : Tension.inp	155
B.1 Paramètres généraux de calcul	155
B.2 Lecture du domaine discret	155
B.3 Les greffons de <i>pre-processing</i>	156
B.4 Les greffons de <i>processing</i>	156
B.5 Les greffons de <i>post-processing</i>	157
B.6 Définition de chargements usuels	157
B.7 Définition de conditions limites usuelles	157
B.8 Vue de l'ensemble du fichier Tension.inp	158
B.9 Synthèse	159
C Modélisation et implémentation du modèle de liaison cohésive poutre	161
C.1 Modélisation	161
C.1.1 Paramétrage du problème	161
C.1.2 Résolution	163
C.2 Sollicitation de traction-compression	164
C.3 Sollicitation de torsion	164
C.4 Sollicitation de flexion	164
C.4.1 Bilan	166
C.5 Implémentation dans la plate-forme de simulation DEM « GranOO »	167
C.5.1 Calcul du torseur des efforts transmis, méthode UpdateLoad()	167
C.5.2 Expression des vecteurs rotation $\vec{\theta}_1$ et $\vec{\theta}_2$	168
C.5.3 Calcul du moment	168
C.5.4 Calcul de la résultante	169

C.5.5	Affectation des moments et des résultantes	169
C.6	Validation de l'implémentation	170
C.6.1	Pré-étude, Invariabilité de la raideur	171
C.6.2	Étude de convergence	172
C.7	Conclusion	172
D	Calcul des énergies : description des méthodes employées	175
D.1	Travail des efforts externes	175
D.2	Énergie interne de déformation	175
D.3	Énergie cinétique	176
E	Essai de torsion 1D, démonstration de l'erreur de 10% de calibration de l'inertie	179
F	Description de l'essai de flexion à rupture	183
G	Présentation des essais de torsion à rupture	187
H	Résultats expérimentaux de référence	189
I	Algorithme de reconnaissance d'agrégats	193
I.1	Fonction de détection d'un agrégat : DetectCluster()	193
I.2	Détection de tous les agrégats	194
J	Condition de périodicité	197
J.1	Réaction des liaisons cohésives et conditions périodiques	198
J.2	Détection des contacts et conditions périodiques	199
J.3	Prise en charge de la double périodicité	200
K	Algorithme de reconnaissance des surfaces externes	205
	Table des figures	209
	Liste des tableaux	213
	Bibliographie	215

Glossaire

bibliothèque ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. 27, 34, 35, 38, 39

bilatérale Une interaction ou liaison de contact est dite bilatérale si elle est capable de transmettre des actions mécaniques suivant les deux sens de la direction normale au contact. 18

design rationale est le terme employé pour désigner une documentation expliquant les raisons de choix techniques et du design général d’une application logicielle.. 51

fonction récursive Les fonctions récursives sont des fonctions dont le calcul nécessite d’invoquer la fonction elle-même.. 193

fraction volumique Rapport entre volumes effectifs occupés par les éléments discrets sur le volume englobant. 65, 68

greffon Un greffon, ou plugin en anglais, est un module informatique qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités.. 27, 35, 42, 43, 48

nombre de coordination Nombre moyen de contacts par élément discret. 65, 68

patron de conception En informatique, un patron de conception, ou design pattern en anglais, est un concept de génie logiciel destiné à résoudre les problèmes récurrents suivant le paradigme objet.. 35, 45

singleton Le singleton est un patron de conception permettant d’assurer qu’une seule et unique instance d’une classe peut être créée.. 44, 45

Acronyms

ASCII *American Standard Code for Information Interchange*. 46, 52

CEA Commissariat à l'Énergie Atomique et aux énergies alternatives. 1, 2, 92, 93, 96

CESTA Centre d'Étude Scientifiques et Techniques d'Aquitaine. 2, 92, 93, 96

CNEM *Constrained Natural Element Method*. 49

DAM Direction des Applications Militaires. 2

DEM *Discrete Element Method*. 17, 18, 20–25, 27–29, 31, 35–37, 39, 41, 42, 45–48, 50, 51, 55–57, 59–61, 80, 89, 91–93, 97, 99, 108–113, 137, 155, 156, 159, 197

DLP Département des Lasers de Puissance. 92, 93, 96

EFESO Étude et Formation en Surfaçage Optique. 1

FAQ *Frequently Asked Question*. 51

FEM *Finite Element Method*. 17

gdd *granoo discrete domain file*. 47, 52, 156, 157

GranOO *Granular Object Oriented workbench*. 29, 35, 48, 50, 51, 54–56, 155, 159

HiPER *High Power laser Energy Research*. 3

ITER *International Thermonuclear Experimental Reactor*. 3

LIL Ligne d'Intégration Laser. 2

LMJ Laser MégaJoule. 2, 3, 5, 7

M2P *Mechanics, Materials and Processes*. 49, 56

MRR *Material Removal Rate*. 12

NIF *National Ignition Facility*. 2, 3, 5

OO Orienté Objet. 30–33, 35–37, 46, 56, 159

PDF *Portable Document Format*. 50

PFD Principe Fondamental de la Dynamique. 28, 39

POO Programmation Orientée Objet. 23, 30, 48, 56

RCP *Random Close Packing*. 65, 68

Rms *Rough root mean square.* 7, 93

SSD *SubSurface Damage.* 10–12, 16, 20, 21, 24, 60, 92, 93, 96, 97, 114, 116–118, 121, 122, 124, 126, 127, 129, 132, 136, 137

STL *Standard Template Library.* 43, 44, 47

SVN *Subversion.* 49, 50

TIRM *Total Internal Reflection Microscopy.* 11

UML *Unified Modeling Language.* 31

X-FEM *eXtented Finite Element Method.* 17

XML *Extensible Markup Language.* 27, 35, 45, 46, 50, 52, 54, 155, 159

CHAPITRE 1

Contexte

*« Ce qui se conçoit bien s'énonce
clairement - Et les mots pour le dire
arrivent aisément. »*

« L'art poétique », N. Boileau

Résumé

La thèse *Modélisation par éléments discrets des phases d'ébauchages et de doucissages de la silice* s'inscrit dans le cadre du projet Étude et Formation en Surfaçage Optique (EFESO) dirigé par le Commissariat à l'Énergie Atomique et aux énergies alternatives (CEA) en collaboration avec des laboratoires de recherche en mécanique, en physico-chimie, ainsi que des acteurs industriels impliqués dans la fabrication de composants optiques.

Le projet EFESO est une des composantes de recherches et d'innovations initiées par le CEA visant à la compréhension et la minimisation des endommagements lasers des composants optiques en Silice.

Il est admis que la présence de microfractures en sous surface, induites par les procédés d'abrasion des composants optiques, joue un rôle clé dans l'initiation des dommages lasers [31]. Cette thèse propose de simuler différents procédés (doucissage et ébauchage) par la méthode des éléments discrets afin de caractériser la densité et la répartition des microfissures. Ces résultats devront être mis en relation avec les paramètres d'usinage dans le but de minimiser la population de microfissures générée par ces procédés d'obtention des composants optiques.



FIGURE 1.1 – Vue aérienne du LMJ, Août 2009 (<http://www-lmj.cea.fr/>)

1.1 Le laser Mégajoule

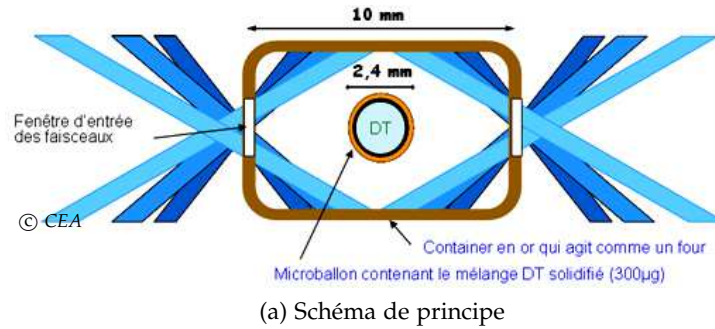
La stratégie de défense nationale française repose sur la dissuasion nucléaire. Dans ce cadre, la Direction des Applications Militaires (DAM), pôle de défense et sécurité du CEA, conçoit, fabrique, maintient en condition opérationnelle, puis démantèle les têtes nucléaires des forces océaniques et aéroportées.

Implanté sur la commune du Barp, à 30 kilomètres au sud de Bordeaux, le Centre d'Étude Scientifiques et Techniques d'Aquitaine (CESTA) est un établissement de la DAM. Créé en 1965, le centre occupe aujourd'hui une superficie de 700 hectares. Le CESTA collabore pleinement au programme de simulation mis en œuvre au sein de la DAM pour garantir la fiabilité et la sûreté des armes, du fait de l'absence d'essais nucléaires auxquels la France a renoncé. C'est sur ce centre qu'a débuté la construction du Laser Mégajoule (LMJ), dont le prototype, la Ligne d'Intégration Laser (LIL) a permis, en 2003, de valider les choix technologiques du futur LMJ. Prévu pour entrer en service fin 2014, le LMJ permettra la réalisation des expériences scientifiques à hautes températures et hautes pressions, indispensables à la validation des modèles physiques utilisés dans la simulation du fonctionnement des armes nucléaires.

Pour cela, une réaction de fusion nucléaire est déclenchée à petite échelle dans une enceinte confinée et instrumentée. La technique employée pour déclencher la réaction atomique est dite de *fusion nucléaire par confinement inertiel*. Cette même technique est utilisée par l'homologue américain du LMJ, le *National Ignition Facility* (NIF).

1.1.1 La fusion nucléaire par confinement inertiel

La fusion nucléaire est une réaction consistant à obtenir un noyau atomique lourd à partir de deux noyaux atomiques légers. Cette réaction dégage une grande

FIGURE 1.2 – Illustration de la cible (<http://www-lmj.cea.fr/>)

quantité d'énergie. Elle existe à l'état naturel dans le soleil, mais peut être également déclenchée artificiellement.

Outre la dissuasion nucléaire, la fusion nucléaire pourrait devenir une source d'énergie propre et inépuisable. La réussite des programmes de fusion par confinement inertiel comme le LMJ et le NIF pourrait alors ouvrir la voie à ce mode de production. Cette alternative est d'ailleurs étudiée par le programme européen *High Power laser Energy Research* (HiPER). La faisabilité d'une autre technique, dite du *confinement magnétique*, est également étudiée dans le cadre du projet *International Thermonuclear Experimental Reactor* (ITER). La technique du *confinement inertiel*, mise en œuvre par le LMJ consiste, en un temps très bref, à comprimer et à réchauffer un mélange d'atomes légers de la famille de l'hydrogène (*deutérium* et *tritium*). Ce mélange est confiné dans un « microballon » de quelques millimètres de diamètre, lui-même contenu dans un conteneur en or d'une longueur avoisinant la dizaine de millimètres (voir figures 1.2).

La réaction est réalisée au sein d'une chambre d'expérience formée d'une sphère de 10 mètres de diamètre et constituée d'une paroi d'aluminium de 10 centimètres d'épaisseur puis d'une couche de béton de 40 centimètres d'épaisseur (voir figure 1.3).

L'énergie nécessaire au déclenchement de la fusion est transmise par l'intermédiaire d'un maximum de 240 faisceaux lasers, réparti en 30 *chaînes lasers*, pouvant délivrer durant 10 à 20 nanosecondes, une énergie totale de plus de 1,5 mégajoules [86].

1.1.2 Les chaînes lasers

Très schématiquement, chaque faisceau laser se compose :

1. d'un pilote qui crée l'impulsion laser d'une longueur d'onde de 1053 nanomètres (1ω) et d'une énergie de quelques nanojoules. Cette source est



FIGURE 1.3 – Photo de la chambre d’expérience (<http://www-lmj.cea.fr/>)

ensuite pré-amplifiée de façon à augmenter l’énergie du faisceau du nano-joule au joule.

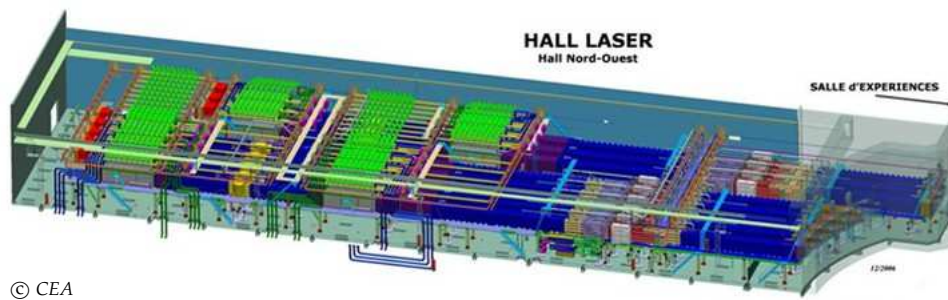
2. d’un système d’amplification utilisant du verre phosphore dopé néodyme permettant d’accroître l’énergie du faisceau laser à un niveau de 15 à 20 kilojoules. Au cours de l’amplification, la dimension du faisceau est agrandie pour réduire l’énergie surfacique transportée et ne pas endommager les composants traversés par celui-ci.
3. d’un système de conversion et de focalisation permettant de convertir la longueur d’onde du faisceau de l’infrarouge, soit, 1053 nanomètres (1ω) à l’ultraviolet, soit, 351 nanomètres (3ω) puis de focaliser afin de réduire la tache du faisceau laser à une dimension de 6 millimètres au centre de la chambre d’expérience. .
4. d’un *nez de chambre* composé d’un hublot de chambre au delà duquel le faisceau se propage dans le vide, d’une lame de phase dont la fonction est de mettre en forme spatialement le faisceau, puis d’une lame anti-déchets assurant la protection des biens et des personnes.

La mise en œuvre de ces chaînes lasers exploite plus de 10 000 grands composants optiques de très haute qualité de finition, dont les sections utiles sont d’environ 400×400 millimètres [70].

Les caractéristiques des faisceaux lasers dans le réseau dit 3ω (en fin de chaîne) sont :

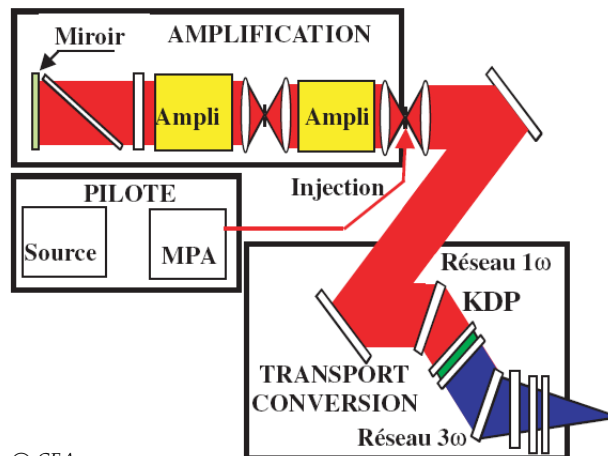
- une forte densité d’énergie (fluence) culminant à 14 J/cm^2 au centre de la chambre d’expérience ;
- une durée d’impulsion de l’ordre 3 ns ;
- une longueur d’onde de 351 nm (ultraviolet).

Ces conditions, très sévères, affectent les durées de vie des composants optiques, notamment, par l’apparition d’*endommagements lasers*.



© CEA

(a) Schéma 3D



© CEA

(b) Schéma de principe

FIGURE 1.4 – Schémas d'une chaîne laser

1.1.3 Les endommagements lasers

Les premières mises en service d'installations pouvant être qualifiées de *lasers de puissance* se situent au début des années 1970. Dès lors est apparue la problématique de *tenue aux flux lasers* des optiques traversés par ces flux lasers [42]. En effet, les composants optiques, soumis à ces régimes de flux lasers, sont particulièrement sujets à des endommagements. Les dommages sont généralement situés en face arrière de l'optique. Leurs tailles caractéristiques peuvent aller de quelques micromètres à quelques centimètres (voir figure 1.5).

La figure 1.6a montre l'évolution des puissances mises en jeu par les différentes installations lasers dans le monde au cours des dernières décennies. Le graphe montre que les conditions de fonctionnement sont de plus en plus sévères. Pour faire face à ce défi technologique, la recherche scientifique, portant sur la tenue au flux laser, s'intensifie comme en témoigne la figure 1.6b. Ce dernier graphe montre également que ce sujet reste encore ouvert. En effet, les dernières installations en date comme le LMJ ou le NIF repoussent encore plus loin les exigences et besoins en terme de tenue aux flux lasers des composants optiques.

Les dommages lasers, au-delà d'une certaine taille caractéristique, dégradent la qualité du faisceau laser. Cette dégradation peut même être contagieuse en provoquant des surintensités sur les optiques en aval et propager ainsi de nou-

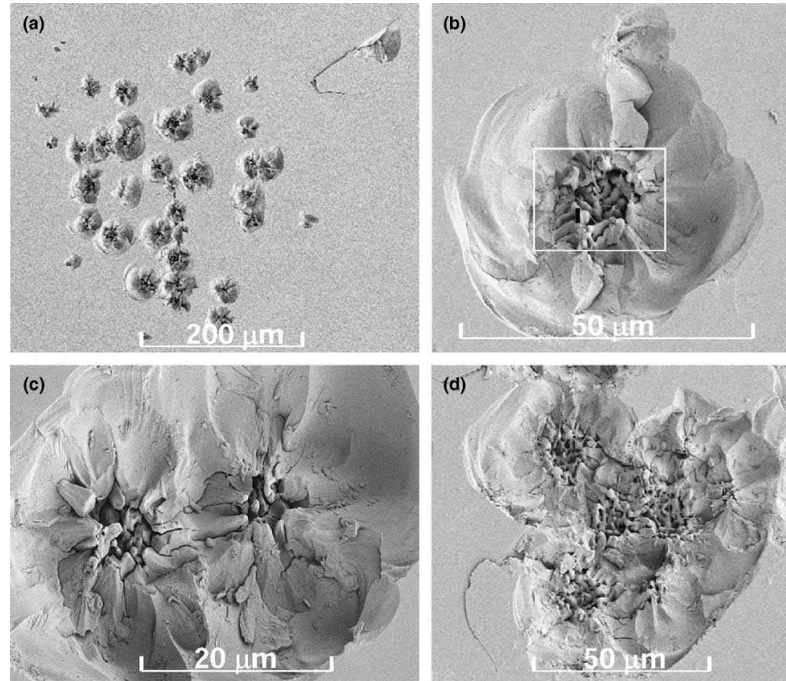


FIGURE 1.5 – Dommages laser - image extraite de [135] -

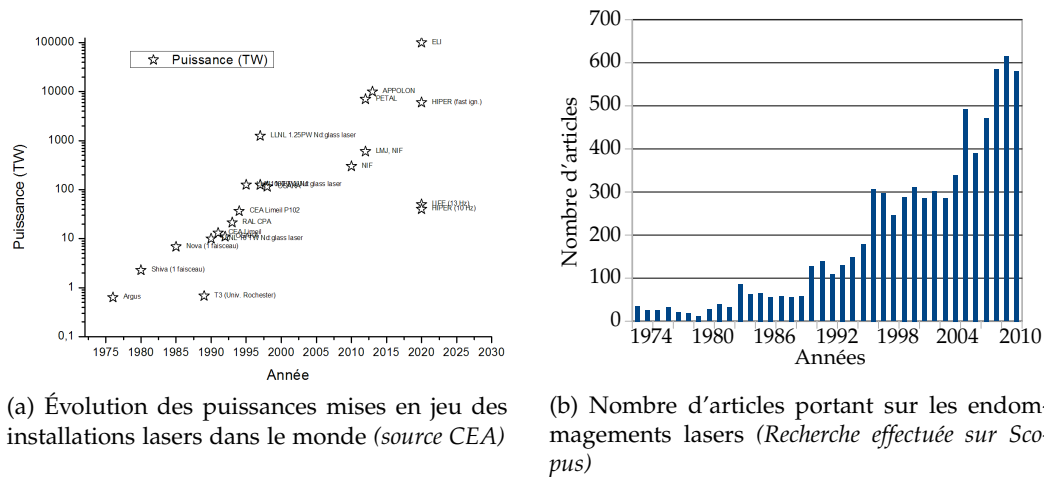


FIGURE 1.6 – Relation entre puissance des installations lasers et travaux portant sur les endommagements lasers

veaux dommages.

De plus, il est observé un phénomène de croissance exponentielle des dommages lors de nouveaux tirs lasers [30]. Cela entraîne de lourdes conséquences d'un point de vue opérationnel; car même si un dommage paraît de premier abord minime, celui-ci, en augmentant de taille au cours de tirs successifs, peut se révéler ultérieurement néfaste.

En conséquence, il est nécessaire de procéder à des opérations de maintenance visant à remplacer les composants optiques défectueux. Ces opérations entraînent des mises en arrêt intempestives ainsi que des coûts non négligeables d'achat de nouveaux composants [7].

Les mécanismes physiques inhérents à l'endommagement laser sont toujours grandement discutés dans la communauté scientifique. Ce sujet fait l'objet depuis une vingtaine d'années d'un cycle de conférences annuelles internationales : *SPIE Laser Damage*.

Il est reconnu et admis, toutefois, que les dommages s'amorcent sur des singularités (défauts) présentes dans le composant optique. Ces défauts absorbent une partie de l'énergie du faisceau laser et entraînent localement une élévation de température pouvant aller jusqu'à la température de sublimation de la silice. La pression exercée par le gaz est alors si forte que des fissures sont générées dans la matière [8].

Des efforts particuliers ont été faits depuis de nombreuses années pour réduire le nombre de défauts générés dans le cycle de vie de la pièce : fabrication, stockage, manipulation, transport... et améliorer ainsi leur tenue au flux laser [52, 129, 83, 95, 94].

Malgré ces progrès, la qualité des composants optiques reste encore à améliorer. En effet, la présence d'un réseau de *micro fractures sous surfacique* dans le substrat de silice [6] est identifiée, aujourd'hui, comme source d'initiation des dommages lasers [11, 18, 113, 41, 31, 8, 85].

Cette problématique est donc au coeur de ce mémoire. Le travail de thèse se focalisera plus particulièrement sur les phénomènes mécaniques, mis en jeu au cours de certaines opérations de fabrication des optiques appelées *surfaçage*, qui induisent l'apparition de ces défauts.

1.2 Les opérations de surfaçage

Le terme générique « surfaçage » comprend plusieurs opérations d'usinage par abrasion. Elles consistent par passes successives, à l'aide d'abrasifs de plus en plus fins, d'atteindre de façon progressive une planéité et une rugosité de surface voulues. Ces spécifications, dans le cadre du LMJ, peuvent être qualifiées de très contraignantes en imposant au fabricant un défaut de planéité de surface de l'ordre de quelques centaines de nanomètres ainsi qu'une rugosité *Rough root mean square* (Rms)¹ de l'ordre du nanomètre sur la surface utile.

Ces procédés, de dénominations différentes selon leurs précisions respectives, sont [63] :

L'ébauchage Cette opération consiste, à partir d'un bloc brut de silice à obtenir une pièce dont les dimensions latérales seront celles de la pièce finale tout en laissant une surcote au niveau de l'épaisseur. Elle est réalisée par sciage puis rectification. L'enlèvement de matière est réalisé à l'aide de meules diamantées. L'abrasif utilisé est formé de particules de diamant enchâssées dans une matrice résinoïde ou métallique. Ces abrasifs sont qualifiés de *fixes*.

Le doucissage Cette opération consiste, à partir d'une pièce ébauchée, à éroder mécaniquement l'épaisseur de la pièce afin de diminuer la rugosité et d'améliorer la planéité de surface. Elle est réalisée à l'aide de particules abrasives d'une dizaine à une centaine de micromètres en suspension dans une solution aqueuse. Les particules abrasives sont généralement de l'alumine, du carbure de silicium ou du carbure de bore. Ce type d'abrasif est

1. écart type de la mesure de rugosité.

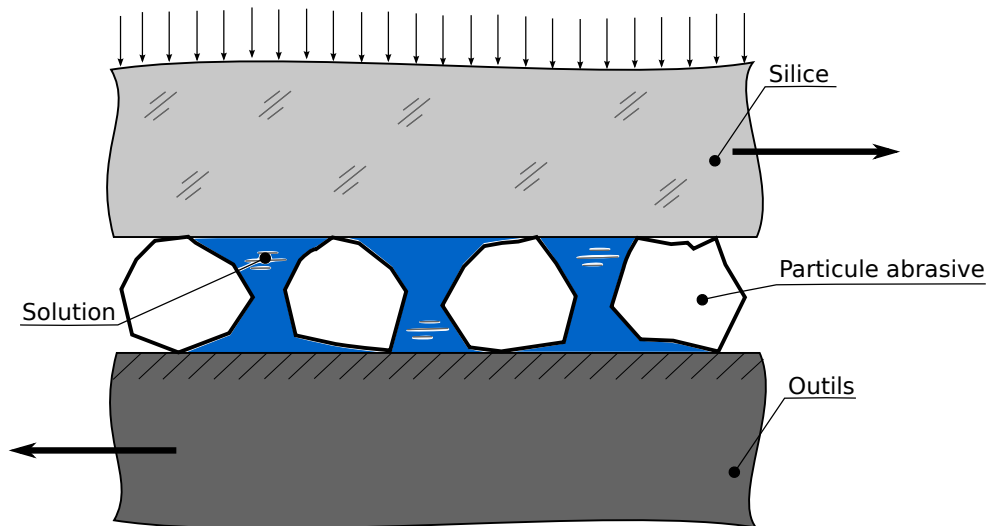


FIGURE 1.7 – Schéma de principe du doucissage

qualifié de *libre*. L'ensemble formé par la solution aqueuse et les grains abrasifs en suspension est dénommé *slurry*. La figure 1.7 décrit schématiquement l'opération de doucissage. La solution abrasive se situe à l'interface entre la pièce de silice et un outil en bronze, fonte, laiton ou céramique. Une pression est exercée sur la pièce afin d'assurer des contacts multiples entre les particules abrasives sur la silice. Un mouvement relatif entre la pièce et l'outil permet de déplacer ces contacts au cours du processus d'abrasion et d'assurer un enlèvement de matière homogène.

Le polissage La technique employée au cours de l'opération de polissage est la même que pour l'opération de doucissage. Toutefois, les grains abrasifs ont une taille de l'ordre du micron et les matériaux utilisés sont généralement de l'oxyde de terre rare (zirconium, cérium, ...). L'outil peut être en poix ou en matériau synthétique. C'est à cette étape que la pièce prend sa transparence. La qualité de la pièce à l'issue de cette phase est quasiment celle de la pièce finale.

Au cours des opérations d'ébauchage et de doucissage, il est admis que l'enlèvement de matière est principalement dû à des causes mécaniques [63, §5.3.1, §5.4.1] contrairement à l'opération de polissage où les mécanismes sont physico-chimiques [40, 21] et [63, §5.4.1]. *Les interactions chimiques ne seront pas traitées au cours de ce mémoire, seuls les effets mécaniques seront pris en compte. En conséquence, l'opération de polissage ne sera pas étudiée.*

L'enlèvement de matière au cours des opérations d'ébauchage et de doucissage est donc considéré comme essentiellement d'origine mécanique. Sous l'effet de la pression de contact des particules abrasives, des fissures sont générées dans le substrat de silice. C'est finalement la répétition de ces actions mécaniques qui permet l'arrachement de fragments de matière.

Le modèle « classique » explique ce phénomène en développant une analogie entre procédé de surfacage et essais d'indentation [13]. Les essais d'indentation consistent à appliquer une charge sur la surface d'un matériau. L'outil utilisé, *indenteur* de formes normalisées [39] (conique, pyramidale, sphéro-conique, hémisphérique...) , doit être d'une dureté supérieure à celle du matériau indenté.

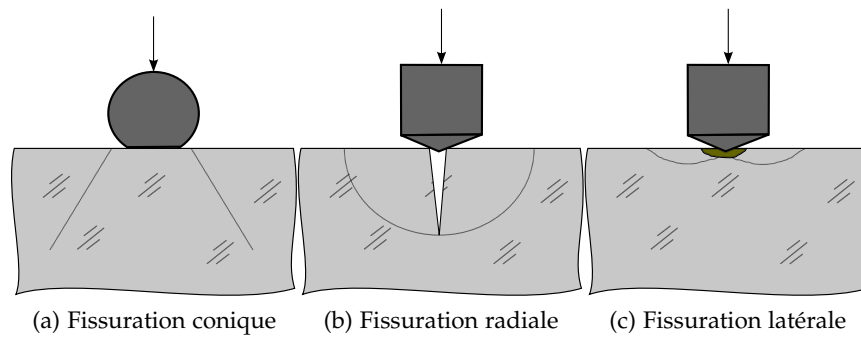


FIGURE 1.8 – Illustration de différentes géométries de fissuration créées par indentation [124]

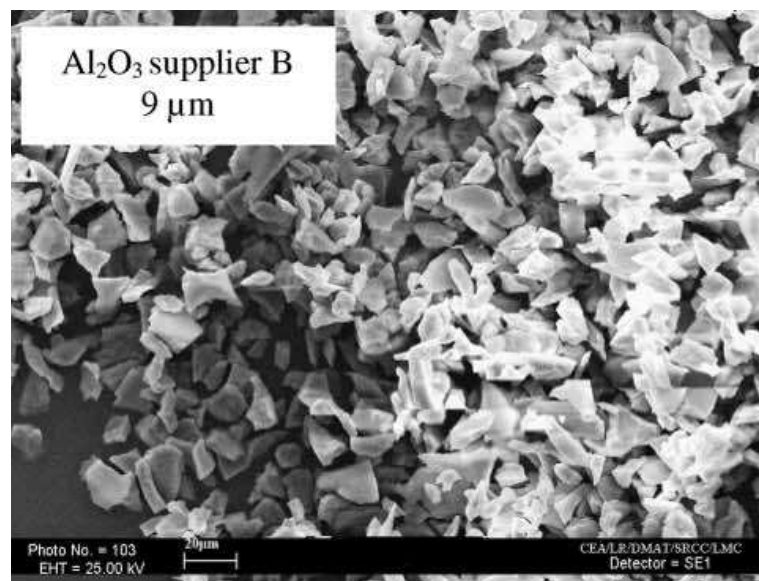


FIGURE 1.9 – Image issue de microscopie électronique à balayage de poudres abrasives - images extraites de [93] -

Ces essais peuvent permettre, par exemple, de qualifier la dureté d'un matériau en mesurant l'empreinte résiduelle de l'outil sur la surface. Dans le cas de matériaux fragiles comme la silice, ces essais peuvent s'accompagner de fissurations de géométries différentes suivant la forme et la pression exercée par l'indenteur [72], comme l'illustre les figures 1.8.

Le modèle classique considère finalement le procédé de surfacage comme étant la répétition de multiples indentations. Dans ce cas, les indenteurs sont les particules abrasives. Or, les poudres d'abrasifs fournis dans le commerce présentent un caractère aléatoire que ce soit au niveau des formes des particules abrasives ou bien de leurs dimensions comme l'illustrent la figure 1.9. En conséquence, tous les profils de fissuration peuvent potentiellement être présents. Dans ce cas, les fractures débouchantes latérales ou radiales (figures 1.8b et 1.8c) participent à l'enlèvement de matière. À l'opposé, les fractures non débouchantes de type conique (figure 1.8a), dont les fissures plongent en profondeur dans la matière contribuent, quant à elles, à la création d'un réseau de micro fractures en

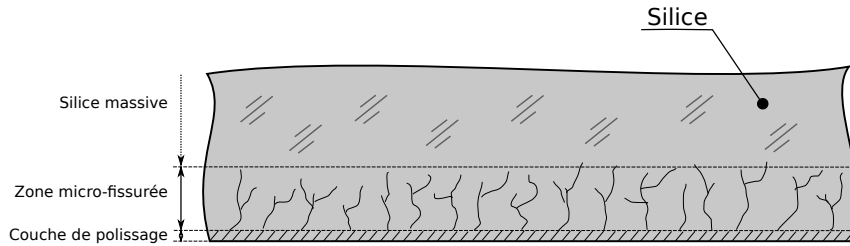


FIGURE 1.10 – Structure de l’interface de silice après polissage [120]

sous surface.

1.3 Les micro fractures sous surfacique

Les micro fractures sous surfaciques ou, *SubSurface Damage* (SSD) dans la littérature anglophone, sont des fissures non débouchantes créées par les opérations de surfaçage de la silice. La figure 1.10 décrit l’interface de la silice après polissage.

Le substrat de silice se compose en premier lieu d’une *zone de polissage* d’une dizaine de nanomètres d’épaisseur. En second lieu vient une *zone micro-fissurée* dont l’épaisseur est comprise entre quelques micromètres et une dizaine de micromètres générée principalement par les opérations d’ébauchage et de doucissage.

Les techniques expérimentales permettant une observation et une qualification de cette zone micro-fissurée constituent à elles seules une véritable problématique scientifique. La section suivante propose une description *non exhaustive* du panel des différentes approches expérimentales liées à l’étude des SSD.

1.4 Techniques expérimentales de mesure des SSD

Dans la référence [133], les auteurs font une revue détaillée des techniques de mesure des SSD. Celles-ci sont classées suivant deux catégories : les mesures destructives et les mesures non destructives. Les sous-sections suivantes présentent brièvement les techniques les plus majoritairement employées.

1.4.1 Mesures destructives

Acidage Cette technique consiste à plonger la pièce de silice dans un bain d’acide. Sous l’action chimique de l’acide, la pièce est érodée sans créer de SSD supplémentaire. De plus, l’érosion chimique est considérée comme *isotrope*, c’est à dire, un enlèvement de matière constant dans toutes les directions de l’espace. L’érosion permet alors d’ouvrir puis de révéler les fissures. Cet élargissement permet alors le passage d’un rugosimètre mécanique. L’érosion étant isotrope, il est alors possible d’établir la profondeur maximale de SSD grâce à la mesure de la rugosité *peak to valley* établie par le rugosimètre.

Polissage local La pièce est polie localement de façon à créer une érosion dont l’empreinte (voir figure 1.11) est de géométrie connue *a priori*. Le procédé de polissage doit garantir la non-formation de SSD. Pour cela différentes techniques sont employées : le *ball-dimple method* [138], le *taper polishing method*

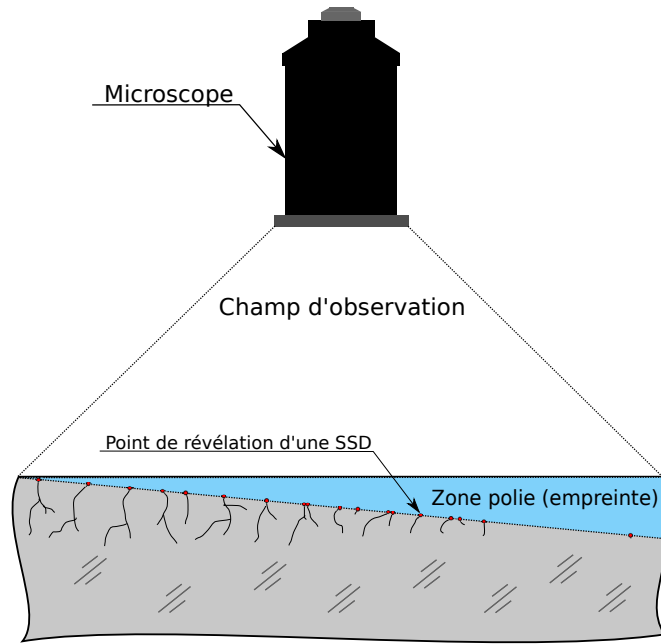


FIGURE 1.11 – Procédé de quantification des SSD par polissage local (*Taper polishing*)

[48], ou plus récemment, le *MRF taper* [124]. Il s'ensuit une phase d'observation de l'empreinte réalisée avec un microscope, où les SSD, révélées par le super-polissage puis éventuellement par acidage léger, peuvent être dénombrées comme l'illustre la figure 1.11.

1.4.2 Mesures non destructives

Les mesures non destructives reposent principalement sur des techniques d'imagerie comme la microscopie confocale [92], la *Total Internal Reflection Microscopy* (TIRM) [68], la tomographie [46], ou bien l'interférométrie [122]. Ces techniques permettent d'extraire des images ou des cartes en deux ou trois dimensions de la zone étudiée. La mise en œuvre de ces techniques est plutôt lourde, peu applicable à l'industrie. De plus, l'investigation est locale. Les zones étudiées sont petites, de l'ordre de quelques micromètres cube. À cette échelle l'occurrence des dommages peut être faible : environ 10^{-6} dommage par μm^3 pour du doucissage [124].

1.4.3 Mesure indirecte

Ce qui est qualifié ici de mesure indirecte peut-être également classé dans la catégorie des mesures non destructives. Cette technique repose sur une loi empirique sans réel fondement théorique. Cette loi suppose une relation de proportionnalité [91] entre rugosité de surface, généralement notée R_t , et profondeur maximale des SSD notée c_{max} . La loi s'exprime donc simplement par :

$$c_{max} < kR_t \quad (1.1)$$

Auteurs	Relations	Préparation de l'échantillon
Hed [48]	$c_{max} < 4R_t$	doucissage
Randi [108]	$c_{max} < 1,4R_t$	ébauchage
Suratwala [124]	$c_{max} < 9,1R_t$	ébauchage et doucissage
Néauport [93]	$c_{max} < 3,3R_t$	ébauchage

TABLE 1.1 – Tableau synthétique des différentes relations $c_{max} = f(R_t)$

De cette façon, connaissant le facteur k , la seule mesure de rugosité pic à pic de l'échantillon permet une estimation de la profondeur maximale de SSD. Le tableau 1.1 synthétise les différentes valeurs du facteur k établies dans la littérature. La variabilité de k s'explique par une forte corrélation de ce coefficient suivant la méthode de préparation de l'échantillon et de la méthode de mesure [93].

1.5 Études expérimentales « procédés / SSD »

Le modèle « classique » présenté dans la section 1.2 basé sur l'analogie entre abrasion et indentation permet une explication théorique du processus d'abrasion et de la formation du réseau de fissures sous surfaciques. Toutefois, il semble quelque peu réducteur.

Par exemple, dans la référence [124], les auteurs observent que la profondeur maximale de SSD expérimentale est d'environ 10^4 fois supérieure à la profondeur prédite par le modèle théorique. Pour justifier cet écart, les auteurs avancent l'hypothèse de présence de particules abrasives, appelées *rogue particles*, en faible nombre² dont les dimensions seraient 10 fois supérieures à la moyenne. Ces particules, en raison de leurs dimensions, seraient en « première ligne » lors du procédé, et joueraient donc un rôle prépondérant lors de la génération des SSD.

Une autre étude, plus complète, a été réalisée en 2010 par Néauport [93]. Les paramètres machines (vitesse d'avance, pression) ainsi que les caractéristiques du *slurry* (concentration, et dimension moyenne des grains abrasifs) ont été systématiquement testés en regard du taux d'enlèvement matière, noté *Material Removal Rate* (MRR), et de la profondeur maximale de SSD générée c_{max} . Ces paramètres sont considérés par les auteurs comme étant les plus influents. La tableau 1.2 synthétise les tendances observées. Sur ce tableau, les flèches orientées vers le haut et vers le bas symbolisent une tendance respectivement croissante et décroissante. Bien entendu, ces résultats ne sont valables que pour les gammes mesurées ; il est difficile d'affirmer si ces tendances sont monotones. Toutefois, certaines d'entre elles sont déroutantes et semblent quelque peu en contradiction avec le modèle classique présenté en section 1.2.

Pour des vitesses et des pressions croissantes, le taux d'enlèvement matière augmente quasi linéairement. Ce phénomène est d'ailleurs bien connu des polisseurs et fut observé par Preston en 1927 [107]. Ce qui est plus surprenant est que la profondeur de SSD décroît. Il paraîtrait logique, pourtant, que les efforts de contacts abrasifs / silice augmentent avec la vitesse d'avance et la pression exercée sur l'échantillon. Dans ce cas, les fissures non débouchantes prédites par le modèle classique devrait être plus profondes. Encore une fois, ce modèle semble être remis en question. À moins que l'intuition immédiate du mécanicien suppo-

2. Entre une pour dix milles et une pour dix millions.

Types	Paramètres	Unités	c_{max}	MRR
Paramètres machine	Vitesse d'avance ↗	m/s	↘	↗
	Pression ↗	Pa	↘	↗
Caractéristiques du slurry	Concentration ↗	%	↗	↗
	Dimension moyenne ↗	m	↗	↗

TABLE 1.2 – Synthèse de l'influence des paramètres du procédé sur la génération de SSD et du taux d'enlèvement de matière pour des opérations de surfacage par abrasifs libres. Résultats extraits de [93].

sant un accroissement des efforts de contacts avec ceux des paramètres machines soit erronée.

En effet, les phénomènes se produisant à l'interface silice / outils sont encore mal connus. Cette méconnaissance est liée à la difficulté d'accéder expérimentalement à l'interface de surfacage. Les échelles étudiées, de quelques dizaines de micromètres, ainsi que la cinématique de l'expérience rendent très difficiles des mesures *in situ*.

1.6 Apport de l'outil numérique

Dans ce cadre, l'outil numérique s'avère une alternative intéressante. Les codes de calcul modernes permettent une investigation fine des phénomènes physiques mis en jeu au cours d'une simulation grâce à l'emploi de *capteurs numériques*. Il serait alors possible d'extraire des données numériques *in situ*, dont l'accessibilité est, aujourd'hui, impossible par les moyens de mesure expérimentaux. L'enjeu final est d'apporter un éclairage nouveau sur un phénomène peu accessible expérimentalement, en s'appuyant sur l'outil numérique et les simulations. Il serait alors possible de tester la validité des scénarios de création de SSD issus de la littérature, dont, notamment, le modèle classique. Également, cet outil peut être mis à profit pour tester certaines hypothèses et aussi, proposer des scénarios plus pertinents qui permettent de mieux comprendre la physique du surfacage et le comportement de la silice lors de cette opération.

Toutefois, l'apport que présente l'outil numérique est conditionné *in fine* par la pertinence du modèle en regard des phénomènes à simuler. La validité du modèle est donc une condition impérative à la réussite de ce projet. C'est ici, une des idées directrices de ce travail de thèse : proposer un modèle numérique suffisamment éprouvé et quantitatif pour le considérer comme *prédictif*.

Le modèle du surfacage décrit plus haut s'approche fortement d'un problème *tribologique*. La tribologie est la science qui étudie d'un point de vue local ou macroscopique les différents types de contact : adhérent, frottant, lubrifié, etc., ainsi que leurs conséquences, dont l'usure des parties frottantes. Pour le tribologue, l'enlèvement de matière du surfacage peut être vu comme de l'usure, et l'interface silice/outil comme un troisième corps complexe [43] composé de la solution aqueuse, de particules abrasives et de particules d'usure comme l'illustre la figure 1.12. En conséquence, une attention particulière sera donnée aux méthodes numériques éprouvées pour aborder les problèmes de tribologies, et notamment, les problèmes d'usure.

Avant de décrire les méthodes numériques possibles liées à la problématique

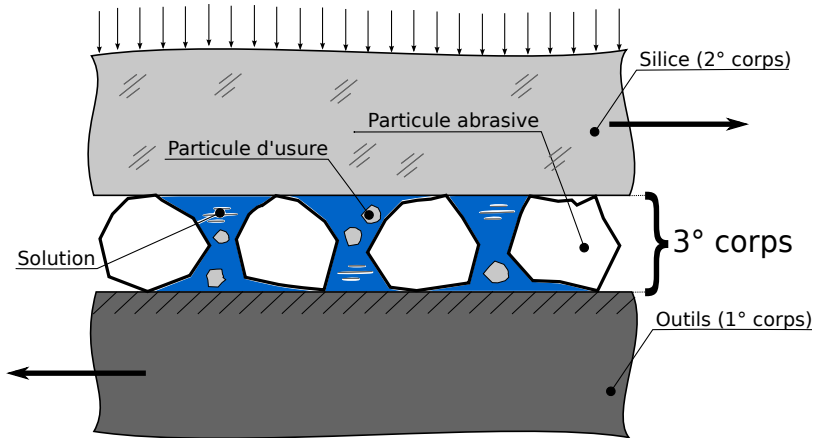


FIGURE 1.12 – Le surfacage vue comme un problème tribologique

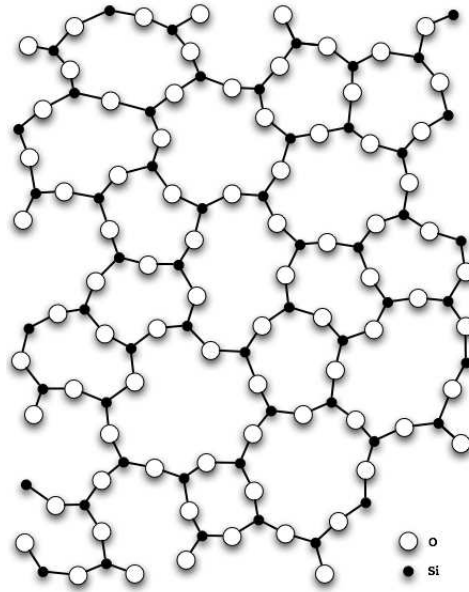
du surfacage et de la génération des SSD, il est proposé tout d'abord, d'établir un cadre théorique afin de restreindre le champ d'investigation des méthodes possibles.

1.7 La silice, un matériau fragile ?

La silice de composition chimique SiO_2 est un matériau amorphe. Sa structure atomique, contrairement aux matériaux cristallins, est dépourvue d'ordre à longue distance ($>20 \text{ \AA}$), mais présente des périodicités à courtes et moyennes distances ($<20 \text{ \AA}$). Une illustration de cette structure pseudoaléatoire est donnée figure 1.13. Il est alors difficile d'établir pour ces matériaux amorphes des modèles de changement d'échelles micro-macro [14, § 4.1.1], ceci incluant le domaine d'étude des propriétés mécaniques du verre. Toutefois, ces comportements macroscopiques peuvent être identifiés par des essais mécaniques, même si les phénomènes au niveau moléculaire sont difficiles à appréhender.

D'un point de vue macroscopique, la silice, dans un domaine de température suffisamment éloigné de la zone de transition vitreuse, se comporte comme un matériau homogène, isotrope, élastique et fragile. Elle se caractérise alors par un module de Young E , un coefficient de poisson ν , et une limite à rupture en traction σ_t [14, § 13]. En pratique, les essais montrent une grande dispersion de la limite à la rupture, sa quantification induit la réalisation d'un grand nombre de tests afin d'obtenir une valeur fiable. Cette dispersion a été expliquée en premier par Griffith [45] qui supposa que les échantillons de verre comporte des microfissures qui agissent comme concentrateurs de contrainte et fragilisent sa structure. De plus, il démontra que la limite à la rupture pratique est bien inférieure à la limite de cohésion moléculaire σ_{th} . Il est admis que pour le verre, le facteur σ_{th}/σ_t est d'environ 10 [72, § 1.5]. Ces résistances théoriques s'approchent de valeurs mesurées expérimentalement pour des échantillons préparés dans des conditions permettant de minimiser le nombre de fissures présentes dans l'éprouvette [137], ceci étayant la thèse soutenue par Griffith.

Malgré cet aspect-quasi idéal de solide élastique fragile, la silice révèle des phénomènes de déformations irréversibles : densification [3] et plasticité [127]. Ces transitions fragiles / ductiles interviennent à de très faibles échelles dimen-

FIGURE 1.13 – Structure de la silice SiO_2

sionnelles [5]. La densification apparaît lorsque la silice est soumise à un fort chargement de type compression hydrostatique. Sous cette sollicitation, la masse volumique de la silice s'accroît localement. D'un point de vue moléculaire, cette densification s'accompagne d'une diminution de l'angle Si-O-Si [14, § 13.2.2.1]. La plasticité des matériaux amorphes ne peut s'expliquer par les mécanismes de dislocation propres aux matériaux cristallins. Le scénario le plus probable est que la plasticité est le fruit de succession de ré-arrangements à l'échelle atomique sous l'effet de sollicitations mécaniques [4]. La plastification est utilisée pour réaliser des opérations locales de surfaçage sur des matériaux fragiles tels que silices ou céramiques et permet d'obtenir un état de surface quasi parfait et une quasi absence de SSD [10, Chap. 5 : Ductile-Regime Grinding of Brittle Materials]. Toutefois, ce procédé est encore émergent pour la silice et peu adapté à une problématique industrielle.

Au travers des différentes publications sur le surfaçage conventionnel du verre, les phénomènes de déformations irréversibles ne sont pas mentionnés. Il est donc probable que l'influence de ces phénomènes soit marginale durant les phases d'ébauchage et de doucissage de la silice. La silice sera donc supposée au cours de cette thèse comme un matériau idéal : homogène, isotrope, élastique et fragile dont les caractéristiques sont résumées par le tableau 1.3. Sur ce tableau, le seuil de rupture en flexion (σ_f) est supérieur au seuil de rupture en traction (σ_t). Ce résultat étonnant semble confirmer que la rupture de la silice s'initie sur des défauts locaux. La probabilité d'activer ces défauts est beaucoup plus grande sur des essais de traction que sur des essais de flexion. Les figures 1.14 illustrent ce phénomène, la zone des contraintes maximales, où la rupture est susceptible de s'initier sur un défaut, représente, théoriquement, la totalité du volume de l'éprouvette pour un essai de traction, alors qu'elle ne représente qu'une zone restreinte de la peau de l'éprouvette pour la flexion.

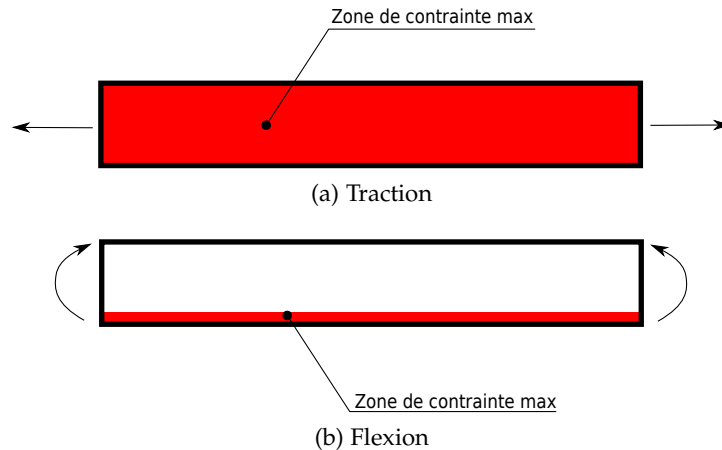


FIGURE 1.14 – Zone des contraintes maximales sous sollicitations de traction et de flexion

Paramètres	Symboles	Valeurs	Unités
Masse volumique	ρ	2 201	$kg.m^{-3}$
Module de Young	E	72,5	GPa
Coefficient de Poisson	ν	0,17	.
Seuil de rupture en traction	σ_t	50	MPa
Seuil de rupture en flexion	σ_f	67	MPa
Seuil de rupture en compression	σ_c	-50	GPa

TABLE 1.3 – Caractéristiques mécaniques de la silice S312 à 20 °C

1.8 Cadre théorique, hypothèses

Il a été vu, au cours des sections précédentes que les phénomènes mis en jeu lors d’une opération de surfaçage sont complexes. Toutefois, il sera considéré que le phénomène prépondérant agissant sur l’enlèvement de matière et la génération de SSD est avant tout mécanique. Dans ce cadre, la prise en compte des actions de contact des particules abrasives sur la pièce est requise comme source de sollicitations mécaniques.

D’autre part, la silice sera considérée comme un matériau purement élastique fragile [14, § 13]. Les phénomènes non linéaires, intervenant à des échelles dimensionnelles réduites comme la *densification*, la *plastification* ou bien le *fluage* seront supposés comme très peu influents et ne seront pas pris en compte. En résumé, il est proposé une vue simplifiée du surfaçage où :

- seules les actions mécaniques sont prises en compte notamment les actions de contact abrasif / silice ;
- la génération de SSD ainsi que l’enlèvement de matières sont les conséquences de ruptures fragiles dans le substrat de silice ;
- l’écoulement de particules d’usures et des particules abrasives à l’interface silice / outil devra être pris en compte ;
- les déformations engendrées sont purement élastiques.

À partir du modèle simplifié du surfaçage décrit plus haut, il est possible d’établir un cahier des charges minimal des différents phénomènes clés à simuler. Ainsi,

la méthode numérique choisie devra être capable de modéliser des phénomènes de **déformations élastiques**, de **multi-fissurations fragiles**, d'**arrachement** et de **contacts multi-corps**. Deux familles de méthodes numériques s'avèrent pertinentes : les méthodes basées sur la mécanique des milieux continus et celles basées sur la mécanique des milieux discrets.

1.9 Méthodes continues

Les approches basées sur la mécanique des milieux continus sont les plus répandues. Par exemple, la méthode des éléments finis ou, *Finite Element Method* (FEM) dans la littérature anglaise, est massivement utilisée en bureau d'étude pour le dimensionnement de structures ou bien comme outil de recherche. Il s'agit de résoudre par divers biais et algorithmes de calcul les équations de la mécanique des milieux continus. Les solutions du problème sont calculées en des points de références, les noeuds, issus de la phase initiale de discrétisation géométrique. Pour d'autres points de l'espace, les solutions sont calculées par interpolation à partir des points de référence « voisins » de la zone étudiée.

Ces méthodes sont tout à fait adaptées à la simulation des déformations élastiques. Par contre, la modélisation des phénomènes de ruptures provoquant des discontinuités au sein du matériau est beaucoup plus problématique. Dans le cas des méthodes avec maillage, la méthode aujourd'hui la plus évoluée est la *eXtended Finite Element Method* (X-FEM) proposée par Moes en 1999 [88]. L'originalité de cette méthode est la possibilité d'introduire une discontinuité en déplacement indépendante du maillage [99]. Cette méthode semble très bien adaptée pour modéliser l'évolution d'une fissure sous des sollicitations quasi-statiques et, plus récemment, sous des sollicitations dynamiques [112], mais des limitations apparaissent pour la simulation de milieux multi-fissurés.

Une autre méthode très répandue est la méthode des zones cohésives [96, 97, 109, 100]. Elle introduit une discontinuité potentielle sur des interfaces pré-définies. Lorsque cette interface n'est pas connue à l'avance, cette méthode devient très coûteuse en temps de calcul en faisant intervenir des procédures de remaillage automatique et perd de son intérêt. De plus, elle introduit une raideur d'interface qui influe sur le comportement dynamique du matériau simulé.

1.10 Méthodes discrètes

Les méthodes discrètes, ou *Discrete Element Method* (DEM) dans la littérature anglophone, permettent de simuler un ensemble de solides rigides (ou éventuellement pseudo-déformables) en interaction par contacts. Le principe fondamental de la dynamique est appliqué de façon à déterminer les déplacements et rotations de chacun des solides [139, pages 245-246]. Dans un modèle DEM, les solides sont appelés *élément discret*, et, l'ensemble de ceux-ci forme un *domaine discret*.

Cette méthode fût tout d'abord appliquée de façon assez naturelle en mécanique des milieux granulaires par Cundall en 1979 [23] pour une classe de problèmes ne pouvant être résolus par voies analytiques ou méthodes continues. Un exemple académique de ce type de problème est la *compaction* illustrée sur la figure 1.15 : il s'agit de déterminer le volume minimal que peut occuper un ensemble de sphères de différentes tailles (milieu discret), dont la loi de distribution est connue, dans un contenant de dimension fini. Dans ce cas, l'emploi de la mé-

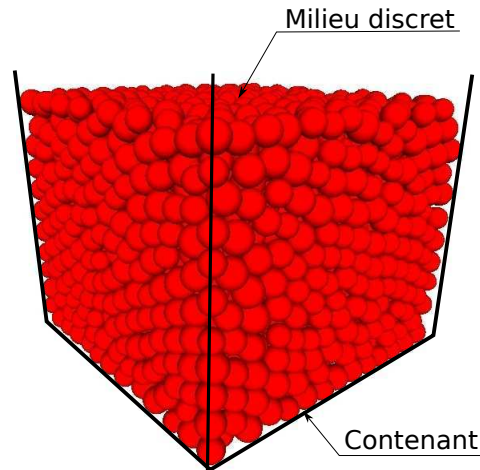


FIGURE 1.15 – Illustration d'un problème de compaction

thode DEM est naturel. Les sphères sont autant de solides rigides interagissant entre eux par contacts mécaniques. Sous l'effet de sollicitations externes comme des vibrations et la gravité, la simulation permet d'atteindre une solution stabilisée où le volume occupé par l'ensemble des solides est minimal. Ce problème requiert la présence d'un grand nombre de solides afin de garantir la reproductibilité de la solution. Ces considérations statistiques ont une grande importance et seront discutées en détail plus loin dans ce mémoire.

Plus récemment, des auteurs ont mis à profit cette méthode pour l'étude de l'endommagement de solides hétérogènes comme les bétons [50], les roches [12], ou de matériaux homogènes comme les céramiques [125]. Dans ces études, le matériau étudié est représenté par un agglomérat d'éléments discrets. La cohésion du milieu est assurée par l'intermédiaire d'interactions bilatérales entre éléments discrets. De par leurs effets, ces liaisons seront nommées *liaisons cohésives*. Les liaisons cohésives peuvent être de différentes natures suivant les propriétés mécaniques à modéliser. Les figures 1.16 illustrent la cohésion et l'élasticité d'un matériau au travers d'une modélisation DEM d'un problème de flexion de poutre cylindrique. Les images 1.16c et 1.16d montrent le réseau de liaisons cohésives.

Pour simuler la fissuration du milieu, un critère de rupture, généralement en déformation ou force, est introduit au niveau des liaisons cohésives. Les liaisons rompues sont exclues du modèle et n'agissent plus sur le système. Les figures 1.17 illustrent l'application de cette méthode pour un problème de rupture de poutre cylindrique sous sollicitation de torsion. La figure 1.17b montre le réseau de liaisons cohésives où les liaisons rompues sont jaunes clairs.

Les différents comportements illustrés par les figures 1.16 et 1.17 sont le résultat à l'échelle de l'éprouvette d'une multitude d'interactions élémentaires entre éléments discrets. Le comportement de l'ensemble des éléments discrets est difficilement prédictible et peut être qualifié de propriété émergente³ [90, préface]. Ainsi, les capacités de fléchir ou bien de rompre à 45°(cf exemples précédents) sont des propriétés émergentes. Ce point de vue est également retenu par Zienkiewicz [139, § 9.1] : au cours de ce paragraphe, l'auteur souligne la mise en jeu de deux échelles :

3. Pour en savoir plus sur la théorie de l'émergence, le lecteur est invité à consulter le numéro 143 de l'année 2005 du journal Sciences et Avenir (Hors-série) consacré à ce thème.

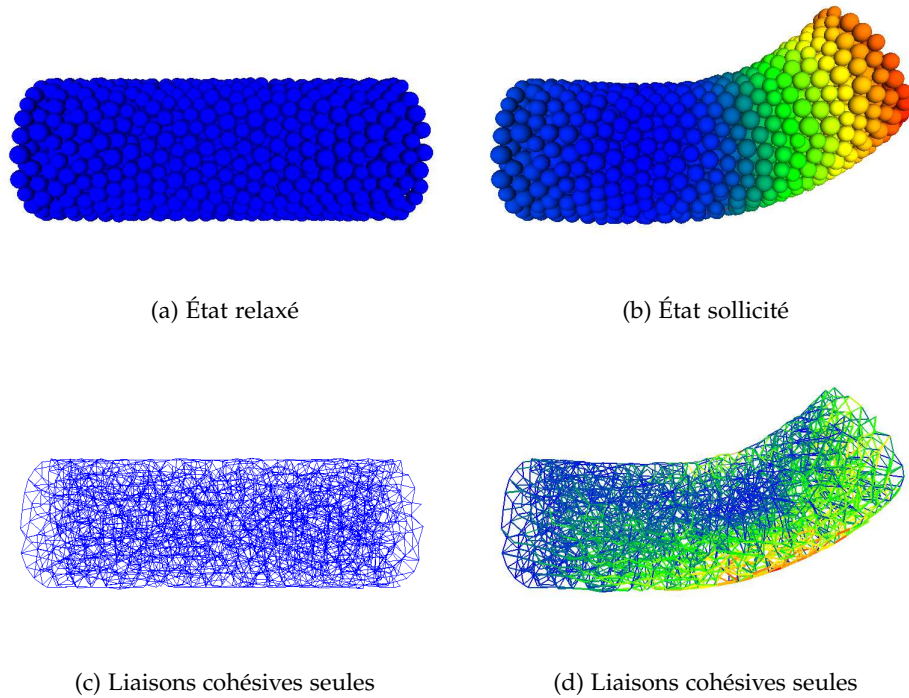


FIGURE 1.16 – Modélisation DEM d'un problème de flexion de poutre cylindrique

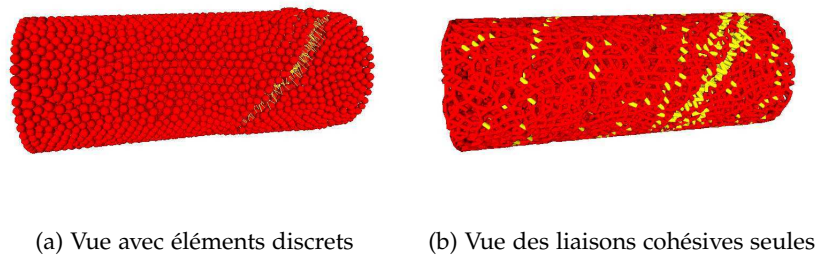


FIGURE 1.17 – Modélisation DEM d'un problème de rupture de poutre cylindrique sous sollicitation de torsion

L'échelle microscopique représentative de l'élément discret et de ses possibles interactions avec ses voisins. C'est à cette échelle que sont introduits les paramètres et comportements du système. Les lois modélisées sont généralement simples.

L'échelle macroscopique représentative d'une structure composée par un ensemble d'éléments discrets en interactions. Le comportement, à cette échelle, découle des interactions multiples à l'échelle microscopique. L'étude et l'analyse du comportement macroscopique est généralement la finalité d'une simulation DEM.

1.11 Choix de la méthode numérique

Le surfaçage est caractérisé, entre autre, par la présence de multiples discontinuités dans le matériau abrasé. Aussi, Zhang dans la référence [136, page 888] observe que les méthodes continues, dans leur ensemble, ont des difficultés à appréhender mathématiquement les problèmes de multi-fissurations alors que les méthodes discrètes souffrent de limitations quant aux problèmes de propagation de fissure.

Dans le cadre de la simulation du surfaçage, la présence et la génération de SSD placent le problème dans un contexte multi-fissuré. En outre, une modélisation fine des cinétiques de fissuration, bien que souhaitable, n'est pas un prérequis fondamental quant au choix de la méthode.

De plus, il a été vu que le surfaçage s'approche fortement d'un problème tribologique (voir section 1.6). Les travaux de Godet sur les contacts frottants [43] ont permis d'établir le socle de la tribologie actuelle. Ces travaux ont permis de révéler, à l'interface du contact, l'existence d'un écoulement. Cet écoulement est formé de fines particules provenant des deux parties en contact et éventuellement de lubrifiant : c'est le *troisième corps*. Toutefois, comme pour le surfaçage, il est très difficile de caractériser expérimentalement ce troisième corps. Aussi, durant la dernière décennie, la DEM a été intensément mise à profit pour étudier et mieux comprendre cette interface [57, 56, 55, 32, 33, 110, 34].

En conclusion, la DEM, comparativement aux méthodes continues, répond de façon plus satisfaisante au cahier des charges établi à la section 1.8. Elle permet naturellement de gérer le contact multi-corps, ne souffre pas de limitation concernant les problèmes de multi-fissurations, et finalement, est capable de modéliser un milieu cohésif et élastique. Enfin, les aptitudes de la DEM ont été éprouvées par l'étude de problèmes tribologiques proches du surfaçage. C'est donc une approche numérique DEM qui sera mise en œuvre dans cette étude.

Aussi, les méthodes discrètes se déclinent en plusieurs catégories. Il est proposé dans la section suivante, une description de ces approches, afin de sélectionner la plus pertinente d'entre elles au regard du problème à traiter.

1.12 Les différentes méthodes discrètes

Parmi les méthodes discrètes existantes se distinguent trois grandes catégories :

Les lattice model sont constituées d'un réseau de liaisons cohésives reliant des points matériels [116, 117]. Les algorithmes de résolution sont basés sur

l'obtention de la matrice de rigidité globale. L'hypothèse de petites déformations est généralement employée pour linéariser et simplifier le problème. L'absence de volume pose des problèmes en phase de post-rupture lors de refermetures de fissures. Des solutions sont proposées en associant aux points un volume équivalent grâce à une décomposition spatiale de Voronoï [53].

Les modèles particuliers sont les plus proches du modèle proposé initialement par Cundall. Les éléments discrets sont généralement de formes sphériques afin de privilégier la rapidité des calculs [87, 79]. Les algorithmes de résolutions sont de type *dynamique explicite*. Cela impose l'utilisation de lois d'interactions mécaniques *régulières* basées sur des relations simples entre effort et déplacement. Des lois d'interactions linéaires, comparables au modèle mécanique du ressort, sont principalement utilisées. Un panel des applications de cette méthode est présenté dans la référence [36].

La dynamique des contacts initiée par Jean et Moreau [89, 17], la dynamique des contacts introduit la notion de contact *non-régulier* ou *non smooth contact* et fait toujours l'objet de développements [37, 38, 71]. Le contact est rigoureusement respecté en interdisant toute interpénétration. Les algorithmes de résolution utilisés sont de type *dynamique implicite*.

Le modèle de dynamiques des contacts, plutôt adapté aux problèmes quasi-statiques ou de dynamiques peu rapides, se révèle tout à fait pertinent pour l'étude de matériaux granulaires. Dans le cadre du surfaçage, il s'agit d'abord de modéliser par un agrégat d'éléments discrets un matériau élastique fragile. Les lois d'interactions mécaniques *régulières* seront alors préférées, car elles permettent d'introduire naturellement de l'élasticité à un ensemble d'éléments discrets.

Aussi, dans le cadre de la thèse, un modèle « hybride », permettant de conjuguer les avantages des modèles particuliers et *lattice*, sera développé en empruntant à ce dernier la notion de liaison cohésive et en associant aux points matériels un volume représentatif de forme sphérique afin de pouvoir traiter le contact unilatéral lorsqu'une liaison bilatérale est rompue. Le schéma de résolution numérique *dynamique explicite* de la méthode particulière sera mis en œuvre de façon à permettre un traitement aisé de lois d'interactions mécaniques régulières. En cela, le choix de la méthode DEM est cohérente avec les études numériques de phénomènes tribologiques d'usure précédemment citées [57, 56, 55, 32, 33, 110, 34].

1.13 Mise en œuvre de la méthode DEM

Ces travaux de thèse se situent dans la continuité de précédentes études menées par Iordanoff et Néauport, où la méthode DEM a été appliquée avec succès pour simuler le surfaçage du verre [54]. Ces travaux montrent une bonne adéquation avec les observations expérimentales : décroissance logarithmique des SSD dans l'épaisseur de silice, et mise en exergue de l'effet des dimensions des particules abrasives sur la taille maximale des SSD. Ces résultats sont synthétisés sur la figure 1.18.

Le lecteur attentif remarquera que le graphique précédent est *sans unité*. En effet, les paramètres, dans le modèle DEM proposé, sont adimensionnés. La description du surfaçage est donc, ici, uniquement *qualitative*.

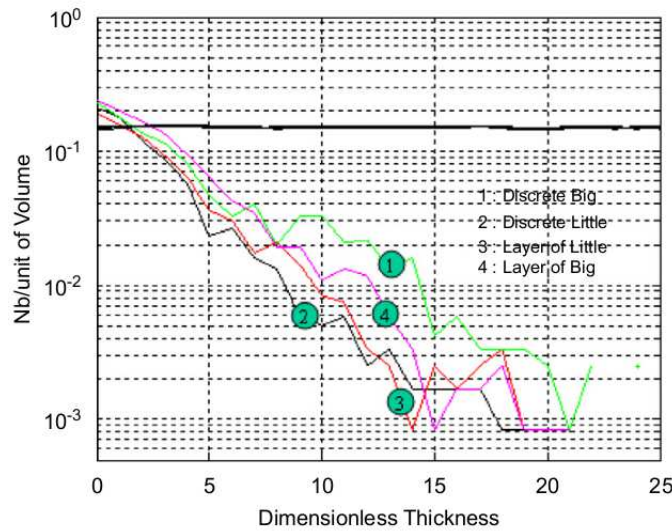


FIGURE 1.18 – Graphique montrant l'évolution de la densité de SSD dans l'épaisseur de silice issue de simulation DEM. Résultats extraits de [54].

L'enjeu est, en s'appuyant sur ces précédents travaux, de proposer un modèle DEM du surfacage permettant une description *quantitative* des phénomènes. La difficulté réside dans le fait que les propriétés mécaniques d'un matériau sont exprimées en terme de grandeurs issues de la mécanique des milieux continus. Pour des matériaux isotropes élastiques fragiles, il s'agit principalement du module de Young, du coefficient de Poisson, et des limites à rupture en traction et en compression. Comment transcrire alors ces grandeurs à un milieu discret, où les notions fondamentales de la mécanique des milieux continus comme contrainte et déformation sont absentes ?

La seule alternative, pour un modèle DEM, est d'introduire indirectement ces grandeurs mécaniques *via* les lois d'interactions à l'échelle microscopique. Puis, seule l'analyse du comportement macroscopique du domaine discret permet de vérifier, *a posteriori*, la bonne concordance du modèle avec les grandeurs recherchées.

Ce problème a été discuté en détail par Ostoja-Starzewski [98]. L'auteur propose des lois de passage entre échelle microscopique et échelle macroscopique pour différentes configurations ordonnées de modèles *lattice*. Dans le dernier paragraphe de cet article, dédié aux réseaux désordonnés, l'auteur, ne pouvant résoudre analytiquement le problème, propose l'emploi d'essais de calibration. Ce dernier point est également, celui retenu par Potyondy et Cundall [106, §3.1] : « *For continuum models, the input properties (such as modulus and strength) can be derived directly from measurements performed on laboratory specimens. For the BPM⁴ (...) the input properties of the components usually are not known. (...) For the general case of arbitrary packing of arbitrarily sized particles, the relation is found by means of a calibration process (...)* ».

La quantification des lois d'interactions microscopiques ou, *calibration*, est une étape requise et essentielle à l'élaboration d'un modèle DEM quantitatif. Ce sujet fera l'objet, plus loin dans ce mémoire, d'un chapitre complet où sera détaillée la

4. "Bonded Particle Model", modèle DEM utilisé par Potyondy and Cundall

démarche mise en place. Cette phase induit la mise en œuvre de nombreuses simulations DEM de natures différentes : essais quasi-statiques, dynamiques, avec et sans rupture dont des tests de traction, flexion, torsion, chocs, compaction, etc.

La réalisation d'un tel plan d'expérience numérique implique de nombreux essais faisant varier les conditions limites, les chargements, les géométries, les algorithmes de résolution, et les lois d'interactions. Cela n'est alors envisageable qu'à l'aide d'outils numériques et informatiques adaptés. La question du choix du code de calcul DEM et, plus particulièrement, de code de calcul de DEM *dynamique explicite* est donc posé.

1.14 Choix du code de calcul DEM

La sélection du code de calcul DEM est une question délicate, d'autant plus que les contraintes sont fortes et les choix restreints. Le code de calcul doit garantir à *minima* :

1. la justesse des calculs ;
2. des performances satisfaisantes ;
3. une mise en œuvre aisée de différentes configurations de simulations : conditions limites, chargements, géométries, etc ;
4. la possibilité d'élaborer et d'intégrer de nouveaux modèles et algorithmes ;
5. la possibilité de simuler différents cas de chargements et de conditions limites, mais aussi et surtout, la possibilité de simuler des phénomènes d'usures.

Les points 1, 2, et 3 sont les minimums requis pour tous codes de calculs. Le point 4 est certainement le plus contraignant. Le code doit être « orienté recherche », en proposant des mécanismes d'élaboration et d'intégration de nouveaux modules de calculs. Des codes *libres* ou *open-source* seront alors privilégiés de façon à garantir l'intégration de nouveaux algorithmes. Enfin, les codes écrits dans des langages modernes supportant le paradigme de la Programmation Orientée Objet (POO) seront également privilégiés.

Ces critères permettent de dégager deux environnements de calcul : *ESyS-Particle*⁵ [134] et *Yade*⁶. Pour avoir un aperçu du rayonnement scientifique de ces deux environnements de calcul le moteur de recherche dédié aux publications scientifiques *google scholar* a été mis à profit. Pour les articles parus avant 2009, date à laquelle cette question devait être résolue, seulement 4 articles contiennent les mots clés « ESyS-Particle discrete element » alors que 37 occurrences sont trouvées pour les mots clés « Yade discrete element ». Le code de calcul *ESyS-Particle* peu populaire a été donc écarté. Le code de calcul *Yade*, plus particulièrement axé sur des problématiques d'endommagement des bétons [15, 50, 49, 67, 121] et de simulation des matériaux granulaires [61, 60, 118], est assez éloigné de préoccupations tribologiques. L'investissement en terme de développement informatique pour adapter le code n'est donc pas négligeable.

Aussi la décision a été prise de refonder le code de calcul interne DEM *Grand3D*. *Grand3D*, intensément utilisé au laboratoire ces cinq dernières années, est plus spécifiquement axé sur des problèmes tribologiques. Ce code, basé sur

5. <https://launchpad.net/esys-particle>

6. <https://yade-dem.org/wiki/Yade>

un style de programmation *impératif* et *procédural* rendant le programme peu souple d'un point de vue opérationnel, a été complétement revu afin de s'ouvrir à une gamme plus large de simulation. La proximité du code avec ses développeurs est également un dernier élément ayant motivé ce choix. Ce dernier point permet une réactivité très forte des personnes impliquées et donc une économie de temps.

Ce travail de développement informatique, représentant une part importante des tâches effectuées durant la thèse, dont l'aboutissement présente un intérêt pour la communauté scientifique confrontée aux problèmes de tribologies, de dynamiques rapides, de multi-fracturation, ou d'études de certains procédés de fabrication, fera l'objet d'un chapitre complet.

1.15 Synthèse, énoncé du plan

Ce chapitre a permis d'introduire tout d'abord la problématique de tenue au flux laser des optiques de silice. Il a été ensuite montré, que, la présence d'un réseau de micro-fractures sous surfaciques dans le substrat de silice est une source d'initiation des dommages lasers qui limite la durée de vie des composants. Le modèle théorique *classique*, expliquant l'apparition de SSD au cours de ces opérations, a été introduit dans la section 1.3. Comme le témoigne la bibliographie, ce sujet fait aujourd'hui l'objet de recherches intensives. Certaines études expérimentales, décrites dans la section 1.5, mettent en exergue des incohérences entre observations et fondements théoriques. Dans ce cadre, l'apport de l'outil numérique (voir section 1.6) permettrait alors une investigation fine de l'interface de surfacage, très difficilement réalisable expérimentalement.

Après une présentation des différentes méthodes numériques possibles, la méthode des éléments discrets (DEM) a finalement été retenue (voir section 1.10) en raison de sa capacité à simuler les milieux multi-fissurés et les problèmes tribologiques. Plus précisément, il a été décidé de re-concevoir le code de calcul DEM dynamique explicite *Gran3D*, orienté vers des problèmes tribologiques. À l'aide d'une approche moderne de conception et de la programmation orientée objet, le nouveau code de calcul devra permettre une mise en œuvre aisée de l'ensemble des campagnes d'essais numériques. Les objectifs sont :

- de caractériser, dans un premier temps, le comportement élastique de la silice « numérique » ;
- de caractériser, dans un second temps, le comportement à la rupture de la silice « numérique » ;
- d'appliquer finalement ces comportements à la simulation du surfacage.

Toutefois, la mise en œuvre de la DEM pose un certain nombre de difficultés quant à l'obtention de résultats quantitativement valables. C'est ici la problématique centrale de la thèse. La démarche présentée ci-après doit permettre la réalisation de cet objectif.

Le chapitre 2 de ce mémoire sera consacré à la présentation du travail de conception et de développement du nouvel environnement de calcul DEM dynamique explicite baptisé « *GranOO* ». L'intérêt et le besoin de ce nouvel outil ont été introduits au cours de la section 1.14 .

Le chapitre 3 sera dédié à l'élaboration des modèles micro-mécaniques ainsi qu'à la démarche de calibration introduite à la section 1.13. Cette étape devra permettre la mise en place et la validation d'un modèle DEM prédictif et quanti-

tatif.

Le dernier chapitre présentera le modèle DEM du doucissage du verre. Les différents résultats et tendances obtenus seront alors discutés et analysés en regard des observations expérimentales issues de la bibliographie.

Le mémoire sera conclu finalement par une synthèse de la démarche générale. Une analyse critique sera aussi entreprise afin de dégager les différentes problématiques non résolues par ce travail de thèse et, ainsi, proposer des améliorations pour de futurs travaux de recherches scientifiques.

CHAPITRE 2

Plate-forme de calcul DEM

« *Never use brute force in fighting
an exponential.* »

Andrei Alexandrescu

Résumé

La plate-forme de calcul par éléments discrets, Granular Object Oriented workbench (GranOO) doit permettre une mise en œuvre d'une large gamme d'expériences numériques, orientée vers des simulations DEM dynamiques explicites, tout en respectant des contraintes fortes en terme d'évolutivité, de sécurité et de performance. Pour répondre à ces besoins, le choix du langage de programmation C++ [123], normalisé, performant et multiparadigme, s'est imposé. Conçue selon l'approche Orientée Objet (OO), l'architecture de GranOO repose sur une organisation en bibliothèques logicielles, permettant ainsi une évolutivité accrue. La sécurité est assurée par l'application systématique du concept de *programmation par contrat* [84]. Enfin, les performances ont été testées en comparaison de bibliothèques qui font référence en la matière. En dernier lieu, une architecture d'exploitation basée sur l'emploi de *greffons (plugins)* dont l'ordonnancement est fait au travers de fichiers modifiables au format Extensible Markup Language (XML), permet une grande souplesse d'utilisation.

2.1 Algorithme de résolution dynamique explicite

Toute modélisation DEM conduit à calculer l'accélération des éléments discrets à chaque pas de la boucle temporelle. Le calcul de la nouvelle position des éléments discrets nécessite d'intégrer l'accélération pour fournir position et vitesse au pas de temps considéré.

Dans le cadre des méthodes particulières, les schémas d'intégration numérique, de type *dynamique explicite*, les plus souvent rencontrés sont [28, §13] : *Verlet vitesse*, *Runge-Kutta*, *leapfrog*, ou encore *Gear*. Dans la référence [111], les auteurs comparent ces différents algorithmes. En fin de compte, il apparaît qu'aucun d'entre eux ne se distingue en terme de stabilité ou de rapidité. Aussi, pour sa simplicité, la méthode d'intégration, nommée *Verlet vitesse*, sera préférée.

Le schéma d'intégration *Verlet vitesse* se décompose en deux étapes : calcul des nouvelles positions, puis calcul des nouvelles vitesses à partir des accélérations courantes. Les accélérations courantes (linéaires et angulaires) sont déduites des forces et moments agissant sur les éléments discrets par l'application du Principe Fondamental de la Dynamique (PFD).

2.1.1 Calcul des positions

Pour chaque élément discret de vecteur position $\vec{p}$, l'application du PFD donne :

$$\ddot{\vec{p}} = \frac{\vec{f}}{m} \quad (2.1)$$

Où :

- $\ddot{\vec{p}}$ est l'accélération de l'élément discret ;
- $\vec{f}$ représente le bilan des forces agissant au centre de gravité de l'élément discret ;
- m est la masse de l'élément discret.

Il est alors possible d'appliquer, pour chaque élément discret l'algorithme *Verlet vitesse* :

$$\dot{\vec{p}}(t + dt) = \dot{\vec{p}}(t) + \frac{dt}{2} (\ddot{\vec{p}}(t) + \ddot{\vec{p}}(t + dt)) \quad (2.2)$$

$$\vec{p}(t + dt) = \vec{p}(t) + dt \dot{\vec{p}}(t) + \frac{dt^2}{2} \ddot{\vec{p}}(t) \quad (2.3)$$

Où :

- t et dt sont respectivement le temps et le pas d'intégration ;
- $\vec{p}$, $\dot{\vec{p}}$ et $\ddot{\vec{p}}$ sont respectivement les vecteurs position, vitesse et accélération de l'élément discret.

2.1.2 Calcul des orientations

Afin de privilégier la rapidité de calcul, les orientations des éléments discrets sont représentées par des quaternions¹ [105, §2.5]. Le schéma est strictement similaire à celui des positions. L'application du PFD permet d'obtenir l'accélération angulaire. Ainsi, pour chaque élément discret, l'accélération angulaire est donnée

1. Les quaternions sont décrits en détail en annexe A

par (voir annexe A, équation A.27) :

$$\ddot{q} = \dot{q} \bullet \bar{q} \bullet \dot{q} + \frac{1}{2} \dot{q} \bullet \left[I^{-1} (\vec{\tau} - 4 (\bar{q} \bullet \dot{q}) \wedge I. (\bar{q} \bullet \dot{q})) \right] \quad (2.4)$$

Où :

- l'opérateur ' $\bullet$ ' représente la multiplication entre deux quaternions ;
- l'opérateur ' $\cdot$ ' représente le produit matrice-vecteur ;
- q , $\dot{q}$, $\ddot{q}$ représentent respectivement les orientations, vitesses, accélérations angulaires d'un élément discret ;
- $\bar{q}$ est le conjugué de q , et $\bar{q} \bullet \dot{q}$ est assimilé au vecteur $\frac{1}{2} \vec{\omega}$;
- $\vec{\tau}$ représente le bilan des moments agissant sur l'élément discret ;
- I est le tenseur d'inertie de l'élément discret.

L'application du schéma d'intégration *Verlet vitesse* donne :

$$\dot{q}(t + dt) = \dot{q}(t) + \frac{dt}{2} (\ddot{q}(t) + \ddot{q}(t + dt)) \quad (2.5)$$

$$q(t + dt) = q(t) + dt \dot{q}(t) + \frac{dt^2}{2} \ddot{q}(t) \quad (2.6)$$

Où :

- t et dt sont respectivement le temps et le pas d'intégration ;
- q , $\dot{q}$, et $\ddot{q}$ représentent respectivement les position, vitesse et accélération angulaires.

Enfin, pour éviter une dérive des résultats, pouvant conduire à une divergence du schéma numérique, il est nécessaire de procéder, à chaque pas de temps, à une renormalisation du quaternion $q(t)$ [105, §2.5].

2.1.3 Vue synthétique de l'algorithme de résolution

La synthèse des étapes introduites précédemment peut être mise sous la forme de l'algorithme 1. Son apparente simplicité masque une véritable difficulté de mise en œuvre : l'élaboration d'une expérience numérique, dont les chargements, les conditions aux limites, les modèles micromécaniques ou bien les géométries mises en jeu peuvent être de natures différentes.

C'est ici l'un des enjeux principaux de la plate-forme *Granular Object Oriented workbench* (GranOO) : fournir des outils et mécanismes, dans un environnement d'utilisation cohérent, d'aide à l'élaboration d'expériences numériques DEM.

2.2 L'approche orientée objet

Depuis l'avènement de l'informatique et des premiers processeurs, la complexité des programmes n'a cessé d'augmenter. De nouveaux langages informatiques sont alors apparus, proposant une syntaxe et une grammaire de plus en plus abstraites de façon à se rapprocher de « *notre mode cognitif d'organisation des connaissances* » [9, Avant-propos]. Ce fut l'avènement des premiers langages de programmation dit *procéduraux*. Cette complexité croissante demanda également des efforts d'organisation du code source. La *programmation modulaire* fit alors son apparition où le code est composé de fonctions dont chacune est chargée de

Algorithm 1 Algorithme de résolution *dynamique explicite*

Require: $\vec{p}(0)$ $\dot{\vec{p}}(0)$ $\vec{p}(0)$ $q(0)$ $\dot{q}(0)$ $\ddot{q}(0)$
 $t \leftarrow 0$
for all itération n **do**
 for all élément discret i **do**
 $\vec{p}_i(t + dt) \leftarrow$ Application de *verlet vitesse* (eq. 2.3)
 $\vec{f}_i(t + dt) \leftarrow$ Bilan des forces agissant sur l'élément discret i
 $\ddot{\vec{p}}_i(t + dt) \leftarrow$ Principe fondamental de la dynamique (eq. 2.1)
 $\vec{p}_i(t + dt) \leftarrow$ Application de *verlet vitesse* (eq. 2.2)

 $q_i(t + dt) \leftarrow$ Application de *verlet vitesse* (eq. 2.6)
 $q_i(t + dt) \leftarrow$ Renormalisation
 $\vec{\tau}_i(t + dt) \leftarrow$ Bilan des moments agissant sur l'élément discret i
 $\ddot{q}_i(t + dt) \leftarrow$ Principe fondamental de la dynamique (eq. ??)
 $\dot{q}_i(t + dt) \leftarrow$ Application de *verlet vitesse* (eq. 2.5)
 end for
 $t \leftarrow t + dt$
end for

traiter un aspect élémentaire du problème à résoudre. Dans l'approche procédurale et modulaire, les données sont « hermétiquement » séparées des traitements, en créant un fichier dédié exclusivement aux données (comme par exemple les fichiers *common* du langage *Fortran* ou les variables globales du langage C). Le cloisonnement des données et des traitements peut entraîner une perte de cohérence du code source. Il en découle un ensemble de pratiques potentiellement nocives comme les commentaires sur le sens et l'utilisation des variables.

La POO, élaborée par Kay dans les années 1970 au travers du langage *Small-talk* [64], propose une organisation différente où traitements et données sont regroupés au sein de mêmes entités : *les objets*.

La littérature foisonne d'ouvrages portant sur l'approche orientée objet. Par exemple, le site « Amazon.com », entreprise de commerce en ligne et distributeur de livres, recense plus de 2300 ouvrages dont le titre possède les mots clés « object oriented ». Il ne sera pas fait ici une description exhaustive des techniques de l'Orienté Objet (OO), celles-ci pouvant être le sujet d'un ouvrage complet, mais plutôt une brève introduction aux concepts fondamentaux nécessaires à la compréhension du document.

2.2.1 Les notions d'objet et de classe : un modèle d'abstraction

Un objet, dans l'approche OO, est une *abstraction*. C'est la représentation d'un concept, par le biais d'informations que le développeur juge pertinentes pour la finalité du programme. Un objet possède des données qui lui sont propres, appelées *attributs*. Un objet est « responsable » de ces attributs. Ceux-ci sont manipulés par l'objet lui-même, au travers de traitements, appelés *méthodes*.

Par exemple, un élément discret sphérique pourra être décrit par des attributs « masse volumique », « rayon » et « position dans l'espace ». Des méthodes pourront également être associées à l'objet élément discret comme la possibilité de calculer sa propre masse en fonction des valeurs données à ses attributs.

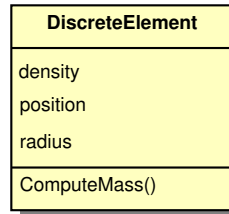


FIGURE 2.1 – Exemple de représentation d’une classe dans le formalisme UML. Vue (simplifiée) de la classe `DiscreteElement`.

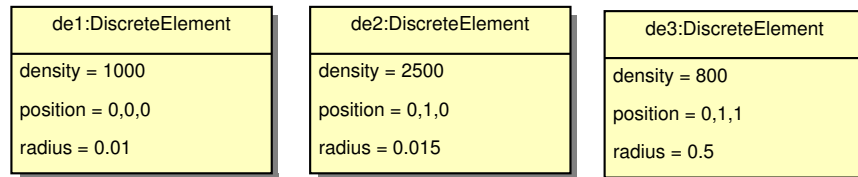


FIGURE 2.2 – Exemple de *diagramme d’objets* issu du formalisme UML. Vue de trois instances `de1`, `de2` et `de3` de la classe `DiscreteElement`.

Les objets sont décrits par des modèles : les *classes*. Une classe est un modèle qui décrit toutes les propriétés et les comportements d’une même famille d’objets. Ainsi, le concept d’élément discret peut être décrit au travers d’une unique classe qui définit des propriétés et comportements communs à tous les éléments discrets. Les objets sont alors la concrétisation de cette classe, ou plus précisément, une *instance* de cette classe. Par exemple une simulation *DEM* peut contenir des milliers d’objets « éléments discret », dont les valeurs des attributs diffèrent (dimension, position, etc.), mais qui sont tous décrits par un seul modèle : la classe « élément discret ».

Dans le cadre de développements OO, l’*Unified Modeling Language* (UML) [101] est un outil de description incontournable. Ce langage graphique normalisé² à base de pictogrammes sera utilisé tout au long de ce chapitre afin d’apporter une vue synthétique et didactique. Issues du formalisme UML, les figures 2.1 et 2.2 décrivent classe et objets « élément discret ». La figure 2.1 montre une classe, avec les attributs et méthodes de la classe `DiscreteElement`. La figure 2.2 est un *diagramme d’objets* où sont représentées les différentes instances de la classe `DiscreteElement` nommées `de1`, `de2` et `de3`. Dans cette dernière vue apparaissent les valeurs des attributs `density`, `position` et `radius` assignées aux différents objets. Il est dit qu’un objet est une instantiation (ou une instance) de la classe qui le modélise.

2.2.2 Les relations

Les objets doivent être capables de communiquer entre eux par l’envoi de messages. Par exemple, dans une simulation *DEM*, les objets `DiscreteElement` peuvent interagir par l’intermédiaire de contacts ou de liaisons. Suivant la norme UML, ces relations peuvent prendre plusieurs formes :

L’association constitue la relation la plus simple entre deux classes, c’est une simple connexion qui permet à deux classes de se connaître.

2. Cf. Object Management Group : <http://www.omg.org/spec/UML/2.0/>

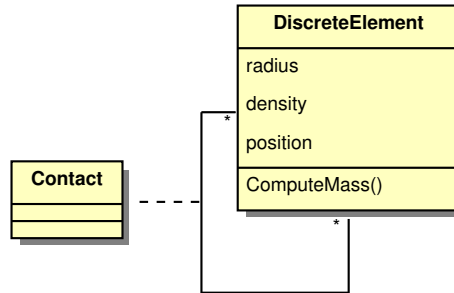


FIGURE 2.3 – Classe portée par une relation. La classe contact est portée par l’association entre deux objets de la classe DiscreteElement.

L’agrégation est une association non symétrique qui exprime un couplage fort et une relation de subordination. Par exemple, cette relation peut être celle de la possession : un élément discret *possède* une position dans l’espace. Dans ce cas, une relation d’agrégation sera établie entre la classe « élément discret » et la classe « position ». Une agrégation peut être traduite par la phrase clé : *A possède un B*.

La composition est une forme d’agrégation plus forte. C’est une relation exclusive où les cycles de vie des agrégats sont liés. Par exemple, une éprouvette cylindrique peut être composée d’un ensemble d’éléments discrets. Un élément discret appartenant à cette éprouvette ne pourra appartenir à une autre éprouvette. En outre, si cette éprouvette est détruite, il faudra également détruire les éléments discrets la composant. Une composition peut être traduite par la phrase clé : *A est fait d’un ensemble de B*.

Les relations peuvent amener à la création de nouvelles instances de classe. Par exemple, une interférence géométrique, ou interpénétration, détectée entre deux éléments discrets, peut induire la création d’un contact. Ce contact peut alors être modélisé par une classe, qui est dite « portée » par la relation entre deux éléments discrets. Le diagramme de classe, représenté sur la figure 2.3, illustre ce modèle.

2.2.3 L’héritage

L’héritage est le mécanisme au cœur de l’approche OO. L’héritage fournit un niveau d’abstraction supplémentaire. Il est possible de généraliser des propriétés communes à différentes classes par la construction d’une hiérarchie de classe.

Par exemple, la section précédente a introduit la classe Contact portée par la relation entre deux éléments discrets. Il est alors judicieux d’utiliser les mécanismes d’héritage de façon à pouvoir définir plusieurs types de contact, comme, le contact élastique, le contact visqueux, ou, pourquoi pas, le contact thermique. Ces différents types de contact peuvent alors tous dériver d’une même classe de base où sont factorisées leurs propriétés communes. La figure 2.4 montre la mise en application de ce point de vue au travers d’un diagramme de classes. Toutes les propriétés définies dans la classe Contact seront alors communes aux trois classes dérivées ElasticContact, ViscousContact et ThermalContact. La phrase clé traduisant une relation d’héritage entre une classe A et une classe B est : *A est un B*. Ainsi, il est possible de dire qu’un ViscousContact *est un* Contact.

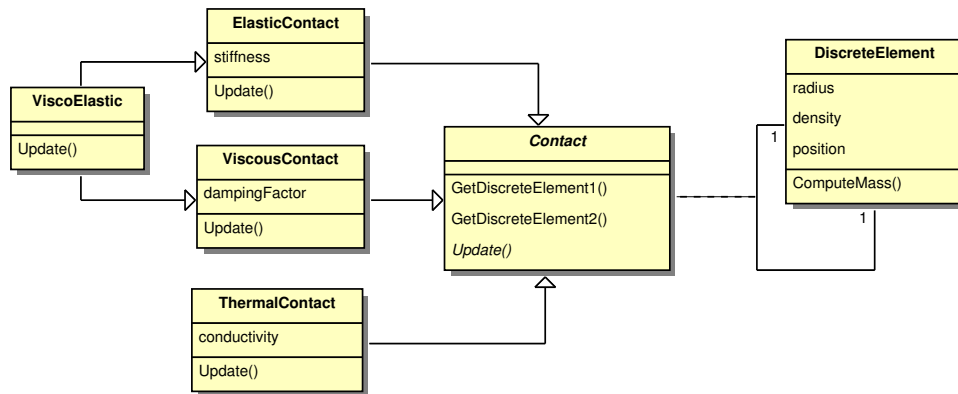


FIGURE 2.4 – *Diagramme de classes*. La classe `Contact` est dérivée en classes `ElasticContact`, `ViscousContact` et `ThermalContact`.

Outre la capitalisation des propriétés, l'héritage permet le *polymorphisme*. Le polymorphisme correspond à la possible redéfinition de certaines méthodes héritées de la classe mère par les classes filles.

Par exemple, les contacts ont des comportements différents suivant qu'ils sont élastiques, visqueux ou thermiques. Sur la figure 2.4 la méthode `Update()` de la classe `Contact` est en italique. Cela signifie que cette méthode est *virtuelle* et doit être redéfinie par les classes dérivant de la classe `Contact`. En conséquence, le développeur pourra redéfinir les méthodes `Update` de chacune des classes filles en fonction de leurs spécificités.

Le polymorphisme permet aussi d'invoquer les méthodes *virtuelles* d'objets sans connaître leurs véritables types. Ainsi, un élément discret peut envoyer un message à l'un de ses contacts pour lui demander de se mettre à jour, sans se préoccuper de sa réelle nature : élastique, visqueux ou thermique.

Un héritage peut également être *multiple*. Par exemple si l'on souhaite définir un contact viscoélastique, il suffit d'introduire une nouvelle classe dérivant du contact élastique et du contact visqueux (voir figure 2.4).

2.2.4 L'encapsulation

Il a été vu, que les objets sont capables de communiquer entre eux. L'approche OO permet au développeur de définir une interface de communication de l'objet avec son environnement extérieur. Cette interface peut être comparée à un filtre, où certains attributs et méthodes sont rendus inaccessibles : c'est le principe d'*encapsulation*.

Le principe d'encapsulation est classiquement expliqué par l'*allégorie de l'iceberg* (voir figure 2.5). La classe est alors symbolisée par un iceberg. L'interface (ou *partie publique* sur la figure) est représentée par la partie immergée de l'iceberg. L'environnement extérieur à la classe est contraint de dialoguer *via* cette interface : c'est la *vue externe* de la classe. La partie submergée invisible pour l'environnement de la classe incarne la *vue interne* (ou *partie privée* sur la figure), dont, la classe est seule responsable.

L'encapsulation permet le principe de responsabilisation et, par extension, de sécurisation des attributs. Il est possible d'introduire des mécanismes permettant à la classe d'assurer l'intégrité de ses données. Le développeur consciencieux in-

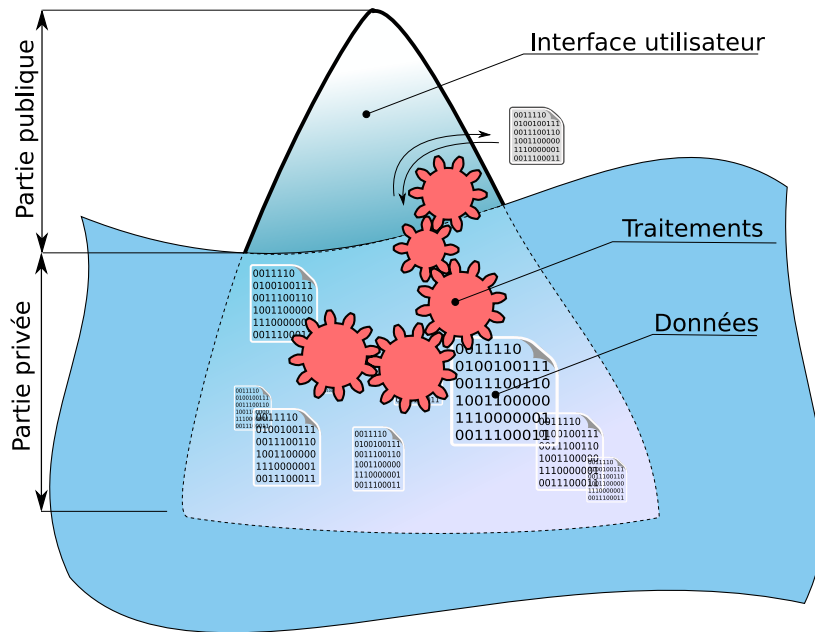


FIGURE 2.5 – Illustration du principe d’encapsulation par l’allégorie de l’iceberg

terdira alors systématiquement un accès direct en écriture des différents attributs composant une classe. Seul un accès contrôlé sera autorisé grâce à l’introduction de méthodes d’accès appropriées. Par exemple, ce mécanisme peut être utilisé pour vérifier qu’un utilisateur distrait ou qu’une erreur de calcul n’affecte pas une masse volumique négative à un élément discret.

En outre, le principe d’encapsulation facilite l’évolution du code. Il est possible de modifier et d’améliorer les mécanismes internes à une classe tout en garantissant la pérennité de son interface. La vue externe n’étant pas modifiée, les modifications n’impactent pas la façon dont l’environnement dialogue avec les classes.

2.3 Choix du langage de programmation

Le C++ [123] supporte la quasi-totalité des paradigmes de programmation : programmation procédurale, programmation modulaire, l’abstraction de données, programmation par objets, programmation générique, etc. Le C++ compte, aujourd’hui, parmi les trois langages les plus utilisés dans le monde³. Cette popularité a permis le développement d’un grand nombre de bibliothèques logicielles, dont certaines d’entre elles sont disponibles sous licences libres, et, par conséquent, accessibles à tous. De plus, le C++ est un langage compilé, dont les performances sont devenues comparables à celles du langage C ou du langage *Fortran*, considérés jusqu’à la dernière décennie comme les plus rapides [131].

3. voir par exemple :

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

<http://www.langpop.com/>

2.4 Architecture logicielle de la plate-forme « GranOO »

L'architecture de la plate-forme de calcul par éléments discrets « GranOO » découle d'une approche OO où les classes sont regroupées par bibliothèques logicielles thématiques. Ainsi, la plate-forme GranOO est composée de trois bibliothèques logicielles écrites en langage C++, qui peuvent être comparées à des « boîtes à outils » (voir figure 2.6).

libUtil est dédiée plus particulièrement à des problématiques informatiques : lecture et écriture de fichiers XML, gestion de greffons, compression de données, capteurs numériques, etc. La *libUtil* intègre également divers outils mathématiques : fonctions mathématiques usuelles et calculs statistiques.

libGeometrical est dédiée aux calculs de géométrie euclidienne dans un espace 3D ou 2D. La *libGeometrical* fournit des classes modélisant les concepts de point, de vecteur, de quaternion, de matrice d'inertie, de repère et également de formes géométriques. Fondamentale pour les calculs DEM, cette bibliothèque fera l'objet d'une section complète.

libDEM est dédiée plus particulièrement aux concepts liés à la DEM. Les concepts d'élément discret, de contact, de liaison cohésive, de chargement, de condition limite, d'intégration temporelle, de domaine discret, d'ensemble d'éléments discrets, etc. sont fournis par la *libDEM*. Cette bibliothèque s'appuie sur les deux précédentes. Elle fera également l'objet d'une description détaillée.

Le développement d'une bibliothèque informatique demande un travail conséquent. Aussi, est-il indispensable de prospector les solutions informatiques déjà existantes. Ce travail, évoqué au cours de la section 1.14 du chapitre 1, justifie la conception de la bibliothèque *libDEM*. La *libUtil* est une bibliothèque hétérogène élaborée en fonction des besoins spécifiques rencontrés. Elle offre des interfaces simplifiées à des bibliothèques externes, des *patrons de conception* et bien d'autres classes utilitaires et ne peut donc être remplacée par une autre bibliothèque. Par contre, la décision a été plus difficile concernant la *libGeometrical*.

En effet, il existe plusieurs librairies C++ de calculs géométriques : *cgal*⁴, *boost.geometry*⁵ ou bien *wykoby*⁶. La plus aboutie d'entre elles est certainement *cgal*. Toutefois, celles-ci n'offrent pas de support pour des calculs de dynamique du solide et souffrent de l'absence du concept de *quaternion*. En conséquence, il a été décidé d'élaborer une bibliothèque de géométrie euclidienne répondant directement aux besoins d'un calcul DEM en intégrant, notamment, les quaternions.

Aujourd'hui, les bibliothèques *libUtil*, *libGeometrical* et *libDEM* comptent respectivement environ 2600, 4100, et 10000 lignes de code utiles. Ce travail de conception et de mise en œuvre au travers du langage C++ est présenté dans les sections qui suivent.

4. <http://www.cgal.org/>

5. <http://trac.osgeo.org/ggl/>

6. <http://www.wykobi.com/>

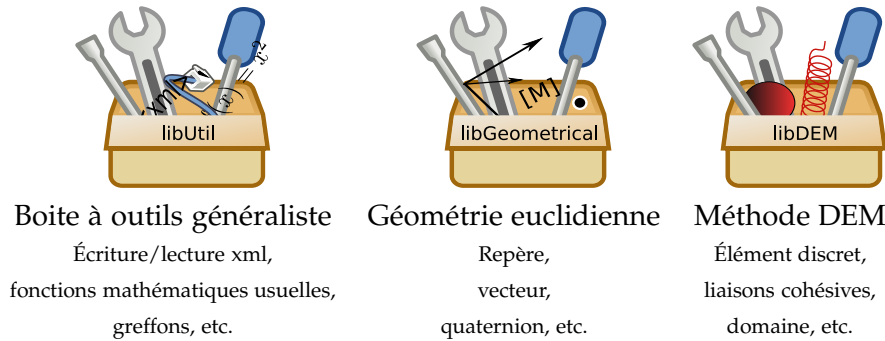


FIGURE 2.6 – Bibliothèques composant la plate-forme de calcul GranOO

2.4.1 La libGeometrical

libGeometrical est une bibliothèque dédiée aux calculs de géométrie dans l'espace. Ainsi, la *libGeometrical* propose :

- des entités de base de calcul en géométrie euclidienne : vecteur, point, repère, etc. ;
- des formes géométriques : sphère, cône, rectangle, etc. ;
- des opérations courantes utiles aux calculs 3D : changement de repère, projection, rotation, norme, etc.

Les entités géométriques et opérations courantes

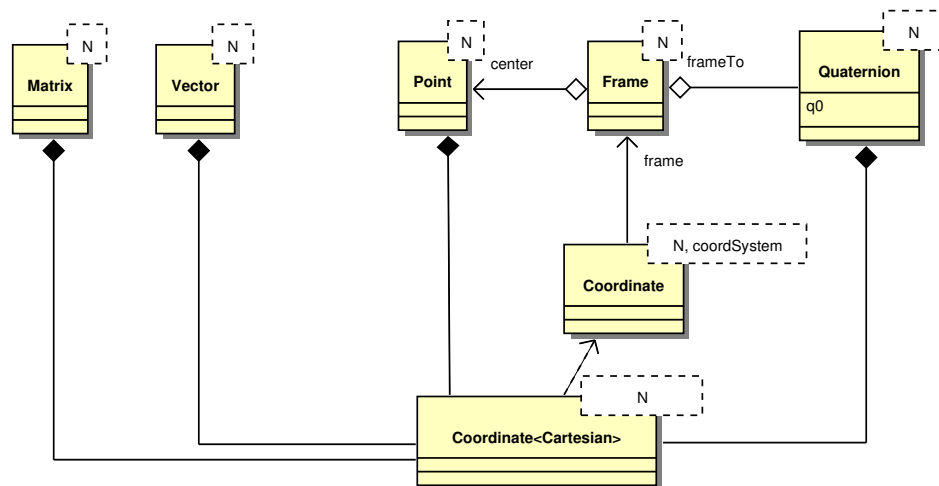
Une modélisation DEM requiert l'usage intensif de calculs de géométrie euclidienne. Il s'agit de positionner et d'orienter les éléments discrets dans l'espace, de décrire leurs cinématiques, d'appliquer les équations de la dynamique, etc.

Les outils de calcul géométrique sont donc fondamentaux. Aussi, un effort particulier de modélisation a été entrepris. Les différents mécanismes de l'OO, décrits dans la section 2.2, ont été utilisés afin de permettre une description au plus proche des concepts de la géométrie euclidienne.

Ainsi, il est considéré que toute entité géométrique (point, vecteur, matrice d'inertie, quaternion) est composée de coordonnées. Ces coordonnées sont exprimées dans un système de coordonnées (cartésien, polaire, etc.) par rapport à un repère et un espace donné. La figure 2.7 montre le diagramme de classes associé à ce modèle.

L'utilisation de la *généricité* [123, chapitre 8] a permis de prendre en compte la dimension de l'espace. Sur le diagramme, le paramètre N , contenu dans les rectangles en pointillés, symbolise le caractère générique des classes par rapport à la dimension de l'espace. Cette technique permet de *spécialiser* certaines fonctionnalités suivant le contexte : 2D ou 3D. Contrairement à l'héritage et au polymorphisme dynamique (voir section 2.2.3), la spécialisation n'entraîne pas de perte de performance, car celle-ci est traitée lors de la compilation [130] et, non pas, à l'exécution du programme [27]. Dans le cadre de simulations DEM en trois dimensions, le contexte 3D a été développé et est aujourd'hui finalisé. La *généricité* a également été mise à contribution pour prendre en compte le contexte de système de coordonnées. Actuellement, le système cartésien a été implémenté.

Au coeur de ce modèle se trouve la classe *Coordinate*, et plus particulièrement la spécialisation de celle-ci pour un système cartésien : *Coordinate<Cartesian>*.

FIGURE 2.7 – Les entités géométriques au cœur de la *libGeometrical*

L'expression de coordonnées par rapport à un repère est symbolisée sur le diagramme par une relation d'association unilatérale entre la classe *Coordinate* et la classe *Frame*.

Des relations de composition (voir section 2.2.2) sont utilisées entre la classe *Coordinate* et les classes *Matrix*, *Vector*, *Point*, et *Quaternion*. Ainsi, il est possible de dire que, par exemple, un vecteur est composé de coordonnées.

Enfin, le concept du repère modélisé par la classe *Frame* est composé, de façon naturelle, d'un point, représentant son origine, et d'un quaternion⁷, représentant l'orientation de sa base par rapport à la base du repère global.

Le modèle présenté ici a été conçu, dans l'intention d'approcher au mieux les concepts mathématiques de la géométrie euclidienne. Ainsi, le vecteur ou le point ne se réduisent pas simplement à une collection de nombres flottants. L'approche OO permet d'attribuer des propriétés et des comportements qui donnent un sens, une « intelligence », aux entités informatiques modélisées.

Comme il a été vu à l'introduction de cette section, un calcul DEM repose sur l'usage intensif des entités développées dans la *libGeometrical*. Par exemple, des centaines de milliers d'opérations vectorielles peuvent intervenir à chaque itération de calcul. En conséquence, la *libGeometrical*, au cœur du calcul DEM, a bénéficié d'attentions particulières en terme de *performance*, mais aussi de *robustesse* et d'*accessibilité* qui seront abordées dans les sections suivantes.

Les formes

La modélisation OO associée aux formes géométriques est illustrée par le diagramme de classes 2.8. Le mécanisme de l'héritage a été mis à profit. Ainsi, la classe de base *Shape* est composée d'un repère qui permet de positionner et d'orienter la forme géométrique dans l'espace. Les classes intermédiaires *VolumeShape* et *SurfaceShape* héritant toutes deux de la classe *Shape* permettent de distinguer deux catégories de forme : les volumes et les surfaces. Ainsi les classes :

- *Sphere*, *Cone*, *Cylinder* et *Box* sont assimilées à des volumes ;
- *Rectangle* et *Disk* sont assimilées à des surfaces.

7. Voir annexe, section A

et écriture des différents attributs les composant. Aussi les classes de la *libGeometrical* sont toutes déclarées amies entre elles de façon à permettre un accès extrêmement rapide aux données.

Les quaternions sont des entités mathématiques, composées d'un nombre réel et d'un triplet de nombres imaginaires, qui permettent de représenter des rotations dans un espace 3D (voir annexe A). Contrairement aux angles d'Euler et aux matrices de passages, les quaternions peuvent être manipulés sans recours systématique aux fonctions trigonométriques comme *sinus* et *cosinus*. Le gain de temps est alors considérable. De plus, les quaternions ne souffrent pas du *gimbal lock*, singularité inhérente aux angles d'Euler qui peut entraîner des calculs erronés [105, §2.5].

Pour vérifier finalement le bon usage de ces techniques, les performances de la *libGeometrical* ont été validées par des *benchmarks*. Ces *benchmarks* consistent à répéter de nombreuses fois une des opérations fournies par la *libGeometrical*. Cette opération est ensuite réalisée le même nombre de fois à l'aide d'une bibliothèque de référence. Les temps de calcul, évalués en nombre de sauts d'horloge du processeur, sont ensuite comparés. La référence choisie est la bibliothèque d'algèbre linéaire *Blitz++*⁸, reconnue pour ses performances en terme de rapidité. Le concepteur, Veldhuizen, est l'auteur d'articles présentant des méthodes d'optimisation des performances de calcul d'objets mathématiques de faible taille [131] par l'utilisation de la méta-programmation [132]. Ces objets mathématiques de faible taille étant comparables à ceux de la *libGeometrical*, cette bibliothèque a donc été choisie comme référence. La figure 2.9 montre le résultat de cette étude comparative pour une compilation en mode optimisée.

Pour l'opération n°1 (multiplication d'un vecteur par une matrice avec affectation du résultat) la librairie *Blitz++* offre des performances trois fois supérieures à la *libGeometrical*. Toutefois cette opération dans un calcul DEM est assez marginale. Elle n'apparaît qu'une seule fois par itération et par élément discret lors de l'application du PFD pour l'accélération angulaire (voir équation ??).

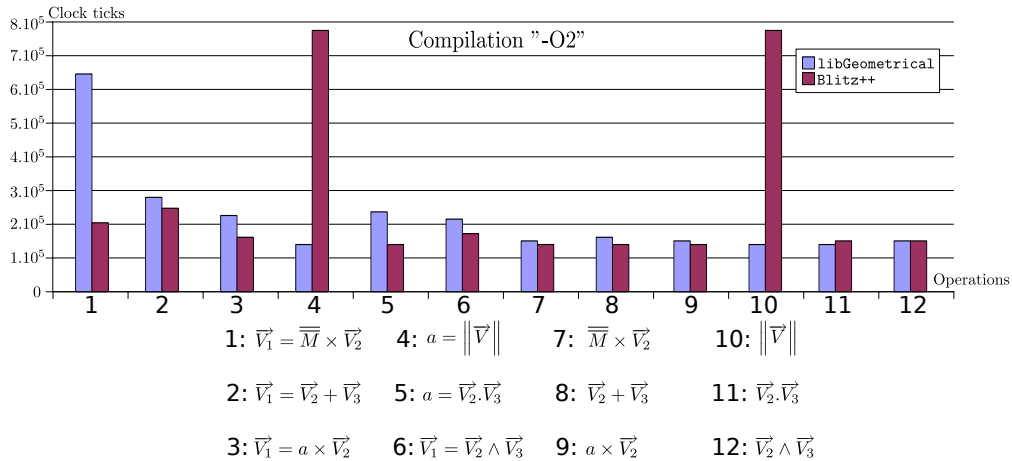
En outre, les opérations n°4 et n°10, de normalisation avec et sans affectation, sont 4 à 5 fois plus performantes lorsqu'elles sont exécutées par la *libGeometrical*. Aussi ces opérations, contrairement à la multiplication vecteur matrice, sont extrêmement courantes, notamment lors des phases de détection des contacts.

Enfin, les autres opérations montrant des performances globalement similaires, il est possible d'en conclure que l'implémentation de la *libGeometrical* permet des performances très satisfaisantes particulièrement adaptées à la résolution de problèmes DEM.

Robustesse

Bien que développée pour des calculs DEM, la *libGeometrical* est une bibliothèque généraliste. Celle-ci peut alors être utilisée par plusieurs projets, dont le code de calcul et de modélisation de pyromécanismes développé dans le cadre de la thèse de Lucie Chorel [20]. Ainsi, la *libGeometrical*, dont les premières versions datent de septembre 2009, a pu bénéficier d'une longue période de *débogage* en intégrant les rapports de bogue, ou les demandes de nouvelles fonctionnalités des différents utilisateurs. Aujourd'hui, la *libGeometrical* peut être considérée comme une bibliothèque dont la fiabilité a été mise à l'épreuve.

8. <http://www.oonumerics.org/blitz/>

FIGURE 2.9 – Comparatif des performances de la *libGeometrical* et de *Blitz++*

De plus, la bibliothèque prévoit des mécanismes de vérification durant l'exécution du programme afin de prévenir les éventuelles manipulations hasardeuses. Pour cela, le paradigme de la programmation par contrat [84] a été mis à profit. Il s'agit de vérifier lors d'un appel de fonction que les données passées à celle-ci ne sont pas erronées, puis, que le résultat renvoyé ne l'est pas également. Par exemple, il sera interdit d'additionner des vecteurs exprimés dans des repères différents ou bien d'accéder à la quatrième composante d'un vecteur 3D. La détection de ces erreurs est réalisée à l'aide de routines de vérification dont le non-respect provoque un arrêt de l'exécution du programme et l'affichage d'un message d'erreur explicite. Ces mécanismes, les *assertions*, sont fournis par la librairie standard du langage C [66]. En cas de contrainte forte en terme de performance, ces vérifications peuvent être désactivées à l'aide d'une procédure de compilation spéciale.

En conclusion, un débogage minutieux combiné à des techniques de programmation par contrat permettent à l'utilisateur d'avoir une confiance accrue quant aux résultats obtenus à l'aide de cette bibliothèque.

Accessibilité

Le terme d'accessibilité peut être traduit par *confort d'utilisation*. Il s'agit de fournir à l'utilisateur des mécanismes permettant une facilité de lecture et d'écriture. Dans ce cadre, plusieurs techniques ont été mises à profit.

La surcharge d'opérateur permet au développeur C++ de redéfinir différents opérateurs mathématiques suivant le contexte. Cette approche permet à l'utilisateur une écriture de haut niveau, intuitive, proche du langage mathématique, dont voici quelques exemples :

```
//Addition vectorielle
Vector<_3D> v3 = v1 + v2;
//Produit vectoriel
Vector<_3D> v4 = v1 ^ v2;
//Produit vecteur - scalaire
Vector<_3D> v5 = 3. * v1;
//Produit scalaire
double      d1 = v1 * v2;
```



```
//etc.
```

La définition des méthodes permet de qualifier les comportements d’une classe. Ces méthodes sont mises à profit pour fournir un panel d’opérations usuelles de calcul, dont voici quelques exemples :

```
//Calcul de la norme
double b = v1.GetNorm();
//Calcul d'un vecteur normal
Vector<_3D> v3 = v1.GetOrthogonalVector();
//Obtention du vecteur position d'un point
Vector<_3D> v4 = p1.GetPositionVector();
//etc.
```

La surcharge de constructeur permet de donner à l’utilisateur différentes méthodes pour instancier un objet. À la charge de l’utilisateur de choisir la plus judicieuse suivant le contexte. L’extrait de code ci-dessous illustre cette utilisation :

```
//Affectation des coordonnees d'un vecteur a la construction
Vector<_3D> v1(0.,0.,0.);
//Constructeur par default, les coordonnees sont alors nulles
Vector<_3D> v2;
//Construction d'un vecteur a partir de deux points
Vector<_3D> v3(p1, p2);
//etc.
```

Les changements de repère sont implémentés de façon à rendre cette opération aisée pour l’utilisateur. Cette fonctionnalité, très utile, est possible grâce à l’architecture orientée objet présentée précédemment. Inédite dans le domaine des bibliothèques géométriques, cette fonctionnalité est illustrée dans l’extrait de code suivant :

```
//Recuperation du repere global construit par default
const Frame<_3D>& f0 = Frame<_3D>::Global;
//Construction d'un nouveau repere
Point<_3D> p1(1., 1., 1.);
Vector<_3D> v0(1., 1., 1.);
//Rotation de pi/2 d'axe v0 representee par le quaternion q1
Quaternion<_3D> q1(v0, 3.14/2.);
//Nouveau repere f1 d'origine p1 et de base q1
Frame<_3D> f1(p1, q1);
//Construction d'entites exprimees dans f1
Vector<_3D> v1(0.,0.,0., f1);
Point<_3D> p2(0.,0.,0., f1);
//Changement de repere
Vector<_3D> v2(v1, f0);
Point<_3D> p3(p2, f0);
//etc.
```

Synthèse

Il a été présenté au cours de cette section, l’architecture de la *libGeometrical*. Intensément utilisée par les calculs DEM, la *libGeometrical* est soumise à des contraintes fortes en terme de *performance*, mais aussi de *sécurité* et d’*accessibilité*. Au cours de cette section, il a été montré que les performances de la *libGeometrical*

peuvent rivaliser avec des bibliothèques existantes reconnues pour leurs performances. La sécurité est assurée par l'application de la programmation par contrat [84] permettant de prévenir les manipulations hasardeuses. Enfin, la *libGeometrical* permet une écriture de haut niveau, proche du langage et des concepts mathématiques géométriques facilitant ainsi l'accessibilité de cette bibliothèque. La *libGeometrical* offre donc des fondations solides sur lesquelles va s'appuyer la bibliothèque *libDEM* pour offrir les modèles de la méthode des éléments discrets.

2.4.2 La bibliothèque libDEM

libDEM est une bibliothèque dédiée plus particulièrement à la mise en œuvre de simulations DEM. Les concepts de base comme les éléments discrets, les liaisons cohésives et les contacts sont modélisés par des classes. La *libDEM* offre également un panel de fonctionnalités utiles à la mise en œuvre d'expériences DEM : conteneurs, expressions aisées de conditions limites et de chargements usuels, parois « réfléchissantes », intégration de *greffons* de traitements, notions de formes englobantes, etc.

Une description détaillée de chacune de ces fonctionnalités étant quelque peu fastidieuse, il ne sera présenté, au cours de cette section, que les concepts et outils fondamentaux proposés par la *libDEM*. Ainsi, une première partie introduira la *libDEM* par une vue générale du « noyau » de son architecture, une deuxième partie sera ensuite dédiée aux conteneurs, une troisième partie introduira la notion de *greffon* et enfin, il sera décrit les mécanismes de sauvegarde et de chargement des domaines discrets.

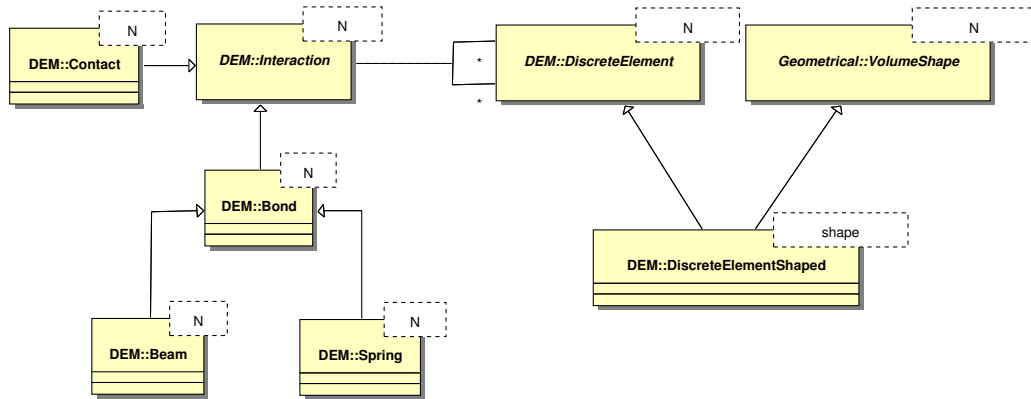
Vue générale

La figure 2.10 montre une vue générale de la *libDEM*. Comme pour la *libGeometrical*, les différentes classes sont génériques de la dimension de l'espace.

Au cœur du modèle se trouve la classe `DiscreteElement`. La classe `Interaction` portée par la relation (voir section 2.2.2) entre deux éléments discrets, modélise tout type d'interaction physique pouvant intervenir entre éléments discrets. Cette classe est ensuite dérivée en `Contact` et `Bond` (liaison cohésive). Les classes `Contact` et `Bond` peuvent elles mêmes être dérivées de façon à modéliser différents types d'interactions, comme illustré par les classes `Beam` ou `Spring`.

Bien que dédiée aux éléments discrets de forme sphérique, l'architecture retenue supporte d'autres formes. Pour cela les techniques de l'héritage multiple (voir section 2.2.3) et de la généricité ont été mises à profit. La classe `DiscreteElement` ne possède pas de forme intrinsèque. Celle-ci est introduite par héritage *via* sa classe enfant `DiscreteElementShaped`. Cette dernière classe est une classe générique paramétrée par le type `shape` qui représente une forme géométrique. Des techniques de vérification statique, proposées par la librairie *Boost* au travers de classes de *traits* [75], permettent de vérifier à la compilation que le paramètre générique choisi est une classe se situant dans la hiérarchie de la classe `VolumeShape` de la *libGeometrical*. Enfin, ce paramètre générique est utilisé pour définir l'autre classe parente de `DiscreteElementShaped`. La combinaison de ces techniques permet de créer n'importe quelle forme d'élément discret. La seule contrainte est que la forme soit modélisée par une classe héritant de `VolumeShape`.

Pour élaborer une expérience DEM, l'utilisateur a sans cesse besoin de pouvoir extraire des ensembles d'entités (éléments discrets, contacts, liaisons, etc.) et

FIGURE 2.10 – Vue simplifiée de l'architecture générale de la *libDEM*

de leurs appliquer des traitements spécifiques. Par exemple le traitement d'une condition limite en déplacement consiste à :

1. identifier l'ensemble des éléments discrets auxquels s'applique la condition limite ;
2. effectuer des traitements sur chacun des éléments discrets de cet ensemble de façon à appliquer la condition limite.

En conséquence, la *libDEM* fournit des outils facilitant ce type d'action. L'identification d'ensemble est réalisée *via* des conteneurs spécialisés et les traitements sont facilités par l'emploi de *greffons*.

Les conteneurs intelligents de la *libDEM* : les *SetOf*

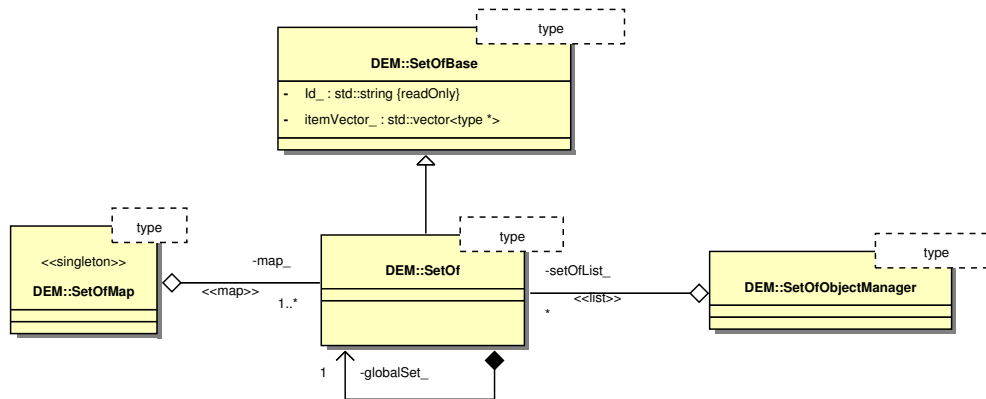
Les *SetOf* composent l'un des mécanismes fondamentaux de GranOO. Ils permettent de référencer des objets de différents types, *DiscreteElement*, *Bond* ou *Contact*, pour différents traitements : application de chargement, de condition limite, ou encore, extraction de données pour un capteur numérique. Aussi, ces conteneurs doivent garantir :

1. un accès rapide en écriture et lecture ;
2. une accessibilité aisée des données ;
3. l'intégrité des données.

Les conteneurs génériques de la *Standard Template Library* (STL), la bibliothèque standard du C++, sont réputés pour leur robustesse et rapidité. Ces derniers ont donc été mis à profit par la classe *SetOf* de façon à garantir les performances. Par contre, ces conteneurs sont délicats à manipuler, surtout en terme de sécurité des données. En conséquence, un panel de fonctionnalités et de mécanismes a été incorporé afin d'assurer un meilleur respect des points 2 et 3.

D'un point de vue informatique, ces conteneurs peuvent être décrits comme des tableaux *homogènes* de tailles *variables*. *Homogène* signifie que le tableau ne pourra contenir que des variables de même type. *Variable* signifie que la taille du tableau n'est pas connue à l'avance et que celle-ci peut varier durant l'exécution du programme. Le diagramme de classes 2.11 présente une vue de la conception de ces conteneurs dans la *libDEM*.

Au centre du modèle décrit par la figure 2.11, se trouve la classe *SetOf*. Cette classe hérite de *SetOfBase* qui possède comme attribut un conteneur gé-

FIGURE 2.11 – Vue détaillée de l’architecture générale des conteneurs de la *libDEM*

nérique de la STL (`std::vector`), ainsi qu’une chaîne de caractère d’identification (`std::string`). Ces classes sont fournies par la STL, la bibliothèque standard du C++.

La chaîne de caractère `Id_` permet d’affecter un identifiant unique à un objet `SetOf`. Ainsi, il est possible de repérer un ensemble d’objets par un identifiant unique. Cet identifiant permet de référencer chaque instance de `SetOf` par l’intermédiaire de la classe `SetOfMap` qui agit comme un *dictionnaire* de `SetOf`. Cette dernière classe est un *singleton* [2, chapitre 6] qui garantit que, seule, une unique instance de cette classe peut être créée durant l’exécution du programme. En définitive, cela permet à l’utilisateur de rechercher un objet `SetOf` grâce à son identifiant unique, par l’intermédiaire de l’objet `SetOfMap`.

Les `SetOf` proposent également des mécanismes d’enregistrement et de désenregistrement automatique. Ces mécanismes sont rendus possibles par une classe interface : `SetOfObjectManager`. Le mécanisme d’héritage est alors mis à contribution. Ainsi, les classes filles de `SetOfObjectManager`, dont les classes `DiscreteElement`, `Bond` et `Contact`, bénéficient des comportements implémentés dans la classe mère.

La classe `SetOfObjectManager` maintient à jour une liste de `SetOf` dans lequel un objet (élément discret, liaison, etc.) est répertorié. Cette liste permet, lors de la destruction de l’objet, de le désenregistrer automatiquement des `SetOf` où il est répertorié, grâce à l’implémentation de son *destructeur*. Ce mécanisme prévient alors des erreurs d’accès en mémoire : les *erreurs de segmentation* [103].

Enfin, le constructeur de la classe `SetOfObjectManager` est également implémenté de façon à référencer automatiquement un objet nouvellement construit par une instance particulière de `SetOf` : le `globalSet_`. De cette façon, tous les objets d’une simulation sont référencés. Cette fonctionnalité, utile pour l’utilisateur, permet également d’éviter les *fuites de mémoire* [73].

En résumé, les instances de la classe `SetOf` peuvent être définies comme des conteneurs intelligents, permettant l’accessibilité aisée et la robustesse des données, en offrant un panel de fonctionnalités plus étendu que les conteneurs de la STL :

- Ils sont accessibles facilement grâce à un identifiant unique ;
- Le désenregistrement des objets référencés est automatique lors de leurs destructions ;
- Un accès inversé est également possible : un objet maintient à jour une liste

- de `SetOf` dans lesquels il est référencé ;
- L'instance unique `globalSet_`, dont la gestion est automatique, référence la totalité des objets d'une simulation.

Une gestion aisée des traitements : les greffons

Un environnement de calcul orienté recherche doit permettre à un utilisateur, de développer et d'intégrer ses propres traitements : modèles physiques, schémas d'intégration numérique, extraction de données, etc. Ainsi, la plate-forme *GranOO* doit être capable :

- d'assurer un développement aisé de traitements spécifiques par des utilisateurs scientifiques dont les connaissances en développement C++ peuvent être faibles ;
- d'assurer l'intégration et l'ordonnancement de ces traitements au cours du calcul DEM.

Aujourd'hui, le premier point est assuré au moyen du langage C++ et repose sur l'utilisation intensive des outils proposés par la *libDEM*, *libGeometrical* et la *libUtil*. Le second point repose sur l'utilisation du concept informatique de *greffon*. Ce concept, permettant une extension de fonctionnalités, est traduit par la classe `PlugIn` illustrée par le diagramme de classes 2.12. Sur ce diagramme, les classes `UserPlugIn_1`, `UserPlugIn_2` et `UserPlugIn_3` représentent des greffons utilisateurs. L'intégration et l'ordonnancement des greffons dans un calcul DEM est réalisé par l'intermédiaire d'un fichier d'entrée XML, paramétré par l'utilisateur.

Pour permettre ce mécanisme, le patron de conception de la *fabrique* [2, chapitre 8] a été utilisé. Ce patron de conception permet d'instancier automatiquement des objets, dont la classe dérive d'une classe abstraite, à partir d'identifiant comme une chaîne de caractère.

Cette *fabrique* est implémentée au travers de la classe *singleton* `PlugInManager`. L'objet `PlugInManager` a alors la charge de balayer le fichier d'entrée et d'instancier les greffons. Ceux-ci sont ensuite appelés au cours de l'exécution du programme, par l'objet de la classe *singleton* `ComputeProblem`, suivant la séquence prévue par l'utilisateur au travers du fichier d'entrée. Cet appel déclenche la méthode virtuelle `Run()` d'un objet de la classe de base *PlugIn* qui exécute la méthode `Run()` de la classe enfant par le mécanisme de *polymorphisme dynamique*. Ainsi, l'utilisateur a la possibilité d'exécuter ses propres algorithmes en redéfinissant la méthode `Run()` de son greffon.

Enfin, pour plus de souplesse, le patron de conception de la *fabrique* a été amélioré. La redéfinition de la méthode virtuelle `ReadXmlFile()` permet à l'utilisateur d'affecter des valeurs aux attributs de son greffon *via* le fichier d'entrée. L'utilisateur peut donc paramétrer le comportement de son greffon sans avoir recours à la re-compilation de celui-ci.

En conclusion, le concept de *greffon*, mis en œuvre au travers du *patron de conception* de la *fabrique* et implémenté au travers des classes visibles sur la figure 2.12, permet à l'utilisateur de concevoir, puis d'intégrer et d'ordonner, ses propres algorithmes au travers d'un fichier de description unique. La *libDEM* fournit également nombre de greffons pour des traitements usuels : application de la gravité, intégration temporelle *verlet vitesse*, algorithme de détection de contact optimisé, etc.

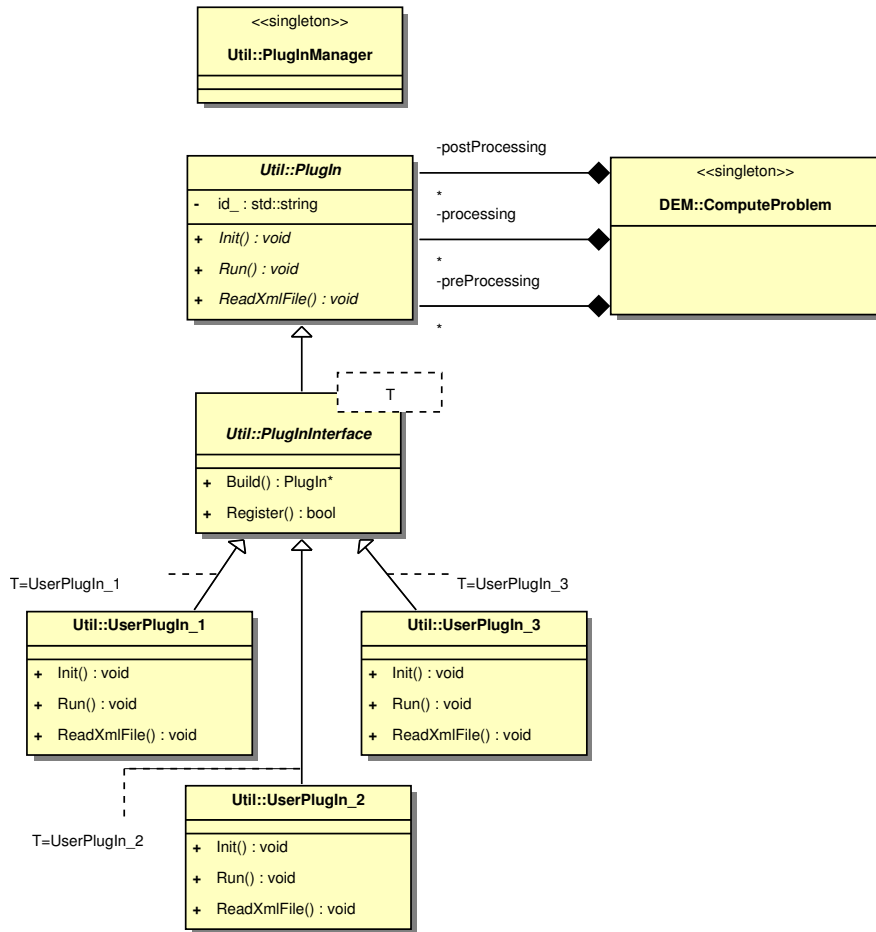


FIGURE 2.12 – Architecture du système de greffon

Enregistrement et chargement de domaine discret : les fichiers *gdd*

L'élaboration d'une simulation DEM nécessite au préalable la construction d'un échantillon numérique. Cet échantillon peut être lui-même issu d'une simulation DEM, comme par exemple, une expérience de *compaction*. Il est donc nécessaire de concevoir des mécanismes de sauvegarde et de chargement de domaines discrets permettant, entre autres, de construire une expérience DEM à partir de résultats issus d'autres expériences DEM. Cette problématique de lecture/écriture de données, dans le domaine de l'informatique, est connue sous le nom de *sérialisation*.

Dans le cadre d'une architecture OO comme le propose *GranOO*, chaque objet doit être responsable de sa propre écriture et lecture en mémoire. Aussi, les mécanismes de *sérialisation* OO proposés par la librairie *boost* ont été mis à profit⁹. En effet, la *sérialisation boost* propose des fonctionnalités avancées, dont :

divers formats de sauvegarde XML, ASCII ou binaire. Le format binaire ne garantit pas l'interopérabilité entre différents systèmes d'exploitation. Le format XML, bien que séduisant, n'est pas adapté aux volumes importants de données. Aussi, le format *American Standard Code for Information Interchange* (ASCII) a été préféré. Ce format présente la particularité de n'écrire que les

9. www.boost.org/doc/libs/release/libs/serialization

strictes informations nécessaires. En cela, ce format est adapté aux données volumineuses. De plus, ce format supporte la compression de données et permet ainsi de réduire encore le volume stocké.

une approche orientée objet qui permet une intégration simple et efficace de la sérialisation au sein des classes concernées. La *sérialisation boost* supporte l'héritage et le polymorphisme dynamique.

la prise en charge par défaut de nombreux types dont les types *natifs* du C++ (entier, flottant, etc.) mais aussi des classes et structures de la STL comme les conteneurs génériques.

la prise en charge des pointeurs et des références dont la mise en œuvre par des techniques classiques est une tâche très difficile.

la gestion de version qui permet l'évolution du code tout en garantissant la *rétro-compatibilité* des formats de sauvegarde.

Grâce à la *sérialisation boost* les objets de la *libDEM* sont donc en mesure de s'écrire ou de se lire à partir d'un flux de données. Du point de vue utilisateur, la mise en œuvre de la sérialisation est rendue aisée grâce à la classe *singleton* *DiscreteDomain*, répertoriant l'ensemble des objets d'une simulation, comme le montre l'exemple suivant :

```
// Recuperation de l'objet singleton DiscreteDomain
DiscreteDomain<_3D>& dd = DiscreteDomain<_3D>::Get();
// Ecriture dans un fichier
dd.Save("experience.gdd");
// Effacement du domaine discret, tous les E.D sont detruits
dd.Clear();
// Lecture du fichier, tous les E.D sont reconstruits
dd.Load("experience.gdd");
// ...
```

Un fichier *granoo discrete domain file* (gdd) peut être considéré comme une « image » complète d'un domaine discret. Ainsi, nombre de caractéristiques sont sauvegardées : formes, positions, vitesses et accélérations des éléments discrets, caractéristiques mécaniques des liaisons cohésives, etc. L'écriture de fichiers peut facilement être automatisée au cours d'une simulation DEM grâce à l'utilisation d'un greffon standard : le *plugin* *SaveDomain*.

Enfin, le programme *gddViewer* permet de visualiser le contenu d'un fichier gdd en 3D interactive. L'utilisateur peut alors s'appuyer sur cet outil pour vérifier le déroulement d'une expérience numérique ou pour faciliter sa description à des tiers. Ce visualisateur, qui ne sera pas décrit en détail dans ce mémoire, offre également un lot de fonctionnalités utiles : créations d'animations, aperçu de données suivant une échelle de couleur, exploration des propriétés d'un objet, etc.

Synthèse

Cette section a permis de décrire l'architecture, mais aussi, les solutions logicielles mises en application dans *GranOO*. L'architecture globale repose sur une organisation modulaire par ensembles fonctionnels où chaque module est concrétisé par une bibliothèque logicielle thématique.

L'objectif de cette organisation est d'offrir des fonctionnalités avancées pour permettre, à un utilisateur scientifique non développeur, la mise en place d'une

expérience DEM dans un environnement de développement orienté recherche performant et cohérent.

Dans ce cadre, le concept informatique de *greffon* a été mis à profit pour permettre l'intégration de traitements spécifiques. L'intégration et l'ordonnancement de ces *greffons* sont réalisés par l'intermédiaire d'un fichier d'entrée unique.

Également, l'utilisation de la POO a permis de développer des outils de haut niveau, proches des concepts du scientifique, qui facilitent le développement de traitements spécifiques. Aussi, le concept de la programmation par contrat et l'utilisation de conteneurs évolués garantit une robustesse accrue des bibliothèques développées.

Enfin, diverses techniques de programmation ont permis de garantir des performances satisfaisantes en terme de rapidité de calcul.

2.5 Environnement de développement

2.5.1 Plate-forme et outils de développement

La plate-forme de développement utilisée est de type *GNU/Linux*. Le choix a été d'opter systématiquement pour des solutions libres et des standards non propriétaires, garantissant la disponibilité des programmes développés.

Le système d'exploitation libre *GNU/Linux* est disponible pour toutes les architectures matérielles. Ce système est reconnu pour ses performances, sa sécurité et sa souplesse. Il peut être employé alors pour diverses applications : serveurs de calculs, ordinateurs de bureau ou portables, systèmes embarqués ou bien encore téléphones portables. Le système d'exploitation *GNU/Linux* se décline en plusieurs *distributions*. La distribution *Ubuntu*¹⁰ est aujourd'hui l'une des plus populaires. Disponible gratuitement, celle-ci a été choisie comme architecture de développement et d'exploitation. Le système *GNU/Linux* offre également un large panel d'outils de développement informatique distribué et maintenu par le projet *GNU*¹¹ [77] :

le compilateur g++ permet de générer un fichier exécutable ou une bibliothèque à partir de sources C++ suivant quatre étapes : pré-traitement, compilation, assemblage et édition des liens.

le débogueur gdb est un puissant outil d'aide au débogage des programmes. Il permet de lancer l'exécution d'un programme puis de le stopper, de l'exécuter *pas à pas*, d'examiner le contenu de variables et de les modifier.

le gestionnaire de projet make permet, grâce à un système de gestion de dépendances, de gérer et d'automatiser la construction de projets. Dans le cas de GranOO, il s'agit principalement d'automatiser et de faciliter les tâches de compilation des nombreux fichiers sources C++ qui constituent les trois bibliothèques.

2.5.2 Hébergement du projet

Le développement de *GranOO* repose sur une architecture standard dans le domaine des environnements de travail collaboratif. Le projet *GranOO* est hé-

10. <http://www.ubuntu.com/>

11. <http://www.gnu.org/>



FIGURE 2.13 – Pages d’accueil de la plateforme M2P

bergé par la plate-forme numérique de travail collaboratif¹² du département Mechanics, Materials and Processes (M2P) de l’école Arts et Métier ParisTech (voir figure 2.13). Développée par l’équipe de Philippe LORONG, cette plate-forme numérique permet d’offrir aux développements informatiques, un environnement de travail à distance de qualité. Elle permet également d’améliorer la visibilité, de valoriser et de promouvoir des synergies inter-projets. Par exemple, des travaux, actuellement en cours, portent sur le couplage entre méthodes discrètes et méthodes continues [59]. La partie applicative de ces travaux propose de faire dialoguer deux environnements de calcul hébergés par la plate-forme M2P : *GranOO*¹³ et le code de calcul Constrained Natural Element Method (CNEM)¹⁴.

En plus d’outils de communications et de promotions, la plate-forme numérique M2P met à disposition un environnement informatique de type *forge*. Les forges sont des outils destinés aux développements collaboratifs. Elles permettent :

d’identifier les différents contributeurs d’un projet et de leurs attacher différents droits et permissions en fonction de leurs prérogatives respectives.

la gestion de version, assurée par le logiciel Subversion (SVN). Le code source du projet est alors hébergé sur un répertoire distant accessible par l’Internet. Les développeurs travaillent sur des copies locales de ce répertoire. Lorsque le développeur juge sa contribution stable et pertinente, il peut alors intégrer ses modifications au répertoire distant. SVN offre alors des fonctionnalités avancées, dont la fusion de versions ou la gestion de conflits.

la distribution du code source qui permet, aux utilisateurs désireux de profiter du logiciel, de télécharger l’intégralité du code source.

diverses fonctions dont : le suivi de bogue, l’élaboration de statistiques diverses, la mise en ligne de canaux de communication, la gestion de documentations, etc.

12. <http://plateformesn-m2p.ensam.eu>

13. <http://www.granoo.org>

14. http://plateformesn-m2p.ensam.eu/CNEM_Pages

Ainsi, cette architecture permet de distinguer les *utilisateurs* et les *développeurs*. SVN permet à un utilisateur de télécharger une copie locale du projet puis de la mettre à jour régulièrement. Par contre, il n'aura pas les droits permettant de modifier le répertoire distant, cette possibilité étant réservée aux développeurs identifiés comme tels.

Un utilisateur peut toutefois soumettre aux développeurs ses contributions grâce à divers canaux de communication. Si cette contribution présente un intérêt général, le code utilisateur, généralement écrit sous forme de greffons (voir section 2.4.2), est alors vérifié et éventuellement réécrit par l'équipe de développeurs, pour finalement être intégré au répertoire distant. Cette procédure permet de *capitaliser* les diverses contributions tout en garantissant leurs fiabilités et leurs performances.

2.6 Environnement d'exploitation

Cette section propose de se placer du point de vue de l'utilisateur. Les différentes étapes, nécessaires à l'élaboration d'une expérience numérique DEM, y seront décrites. L'exemple choisi est un essai de traction sur éprouvette cylindrique.

2.6.1 Téléchargement et déploiement

Le premier point d'entrée, pour un utilisateur de GranOO, est le portail Internet <http://www.granoo.org> (voir figure 2.14). Ce site met en ligne la documentation « *InstallGranOO.pdf* » où sont décrites les procédures de téléchargement et d'installation. À l'issue de ces opérations, l'utilisateur dispose d'une copie locale de GranOO où sont disponibles :

- l'ensemble du code source ;
- les bibliothèques *libUtil*, *libGeometrical* et *libDEM* compilées sur sa machine ;
- des exemples de simulation DEM ;
- la documentation au format Portable Document Format (PDF).

2.6.2 Quelques améliorations

Aujourd'hui, seule la plate-forme *GNU/Linux* est supportée officiellement. Toutefois, aucune restriction majeure n'interdit le portage de GranOO sous d'autres plate-formes : *MacOSX* ou *Windows*. En effet, le code source de GranOO n'utilise que des composants standards et portables. Par contre, certains outils de développement ont été configurés pour utiliser toute la puissance de l'environnement de développement disponible sous *GNU/Linux* : c'est le cas des fichiers *Makefile* utilisés par le gestionnaire de projet standard *make*, qui nécessitent un travail de portage pour être exploités sous *MacOSX* ou *Windows*.

La documentation de GranOO est en cours d'élaboration. Elle comporte actuellement une cinquantaine de pages rédigées en anglais où sont présentées :

- la procédure d'installation ainsi qu'un aperçu rapide des possibilités de GranOO ;
- la liste des greffons standards ainsi que leurs descriptifs ;
- la liste et le descriptif des balises XML supportées par le fichier d'entrée.

Toutefois, cette documentation pourra largement être complétée pour assurer à l'utilisateur une bonne compréhension de l'environnement de développement

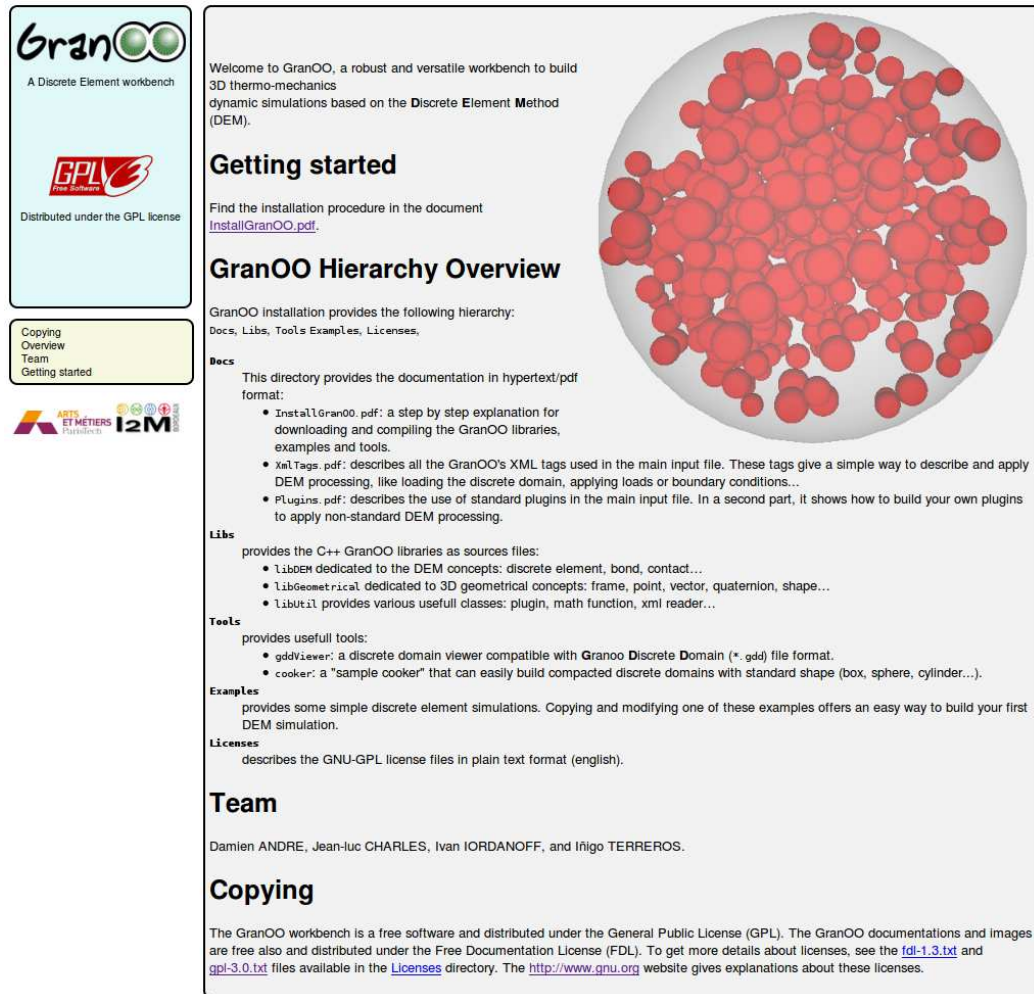


FIGURE 2.14 – Page d'accueil du site granoo.org

nécessaire pour assurer son autonomie. Dans ce but, il est envisagé l'ajout d'une section *Frequently Asked Question (FAQ)*, d'une section *design rationale*, ainsi que d'un tutoriel. Pour compléter ce dispositif, il sera également mis en ligne un descriptif détaillé et exhaustif des bibliothèques *libUtil*, *libGeometrical*, et *libDEM* basée sur l'outil de génération de documentation *doxygen*¹⁵.

2.6.3 Les étapes clés

Le coeur de GranOO est composé des bibliothèques *libUtil*, *libGeometrical* et *libDEM*. GranOO n'est donc pas un logiciel. Pour élaborer une simulation DEM, l'utilisateur doit dialoguer avec ces bibliothèques *via* le langage C++ et le système de greffons présenté à la section 2.4.2 puis générer un exécutable. Lors de l'exécution du programme, GranOO fournit plusieurs mécanismes permettant la sauvegarde de données au cours de la simulation. Les étapes clés nécessaires à l'élaboration et à l'exploitation d'une simulation DEM sont synthétisées sur la figure 2.15. Celles-ci se composent de :

15. <http://www.doxygen.org>

- 1- **l'écriture des sources C++** qui permet de définir des greffons et de traitements utilisateurs.
- 2- **la compilation des sources C++** dont la mise en œuvre est assistée par l'outil *gnu make*. Le fichier `Makefile.EditMe` offre une interface très simplifiée à la configuration de cet outil. L'utilisateur produit alors un fichier *exécutable*.
- 3- **l'exécution** de la simulation. Le paramétrage de celle-ci est réalisé par l'intermédiaire d'un fichier d'entrée XML présentant une syntaxe claire et accessible. Au cours de la simulation, l'exécutable peut produire différents fichiers résultats, dont :
 - un ensemble de fichiers `gdd` permettant de conserver les résultats de la simulation pour des pas de temps donnés ;
 - un fichier ASCII tabulé `Sensors.txt` dont le contenu est déterminé par l'utilisateur *via* l'utilisation de *capteurs numériques*. Les capteurs numériques, qui ne seront pas détaillés dans ce mémoire, permettent une mesure fine de différents phénomènes et un suivi de leur évolution au cours de la simulation.
- 4- **l'exploitation** des fichiers résultats.
 - Les fichiers `gdd` peuvent être lus et visualisés par le programme *gddViewer* ;
 - Le fichier `Sensors.txt` peut être traité par divers outils : par exemple, à l'aide de la bibliothèque *matplotlib*¹⁶ facilitant la mise en œuvre de graphiques paramétrables à l'aide de scripts *Python*¹⁷.

2.7 Les fichiers nécessaires à l'élaboration d'une expérience DEM de traction

La liste des fichiers contenus dans un répertoire dédié à la construction d'un essai de traction est :

```
-rw-r--r--  1 andre 84      1262 2011-04-28 21:44 Main.cpp
-rwxr-xr-x  1 andre 84     6667 2011-07-01 15:25 Makefile
-rw-r--r--  1 andre 84       313 2011-04-20 16:41 Makefile.EditMe
-rw-r--r--  1 andre 84    1290872 2011-04-20 16:41 cylinder.gdd
-rw-r--r--  1 andre 84     3441 2011-07-04 19:11 Tension.inp
```

Le fichier `Main.cpp`

Le fichier `Main.cpp` contient la fonction `main()` qui constitue le point d'entrée pour tout exécutable C++. Dans la majorité des cas, l'implémentation de la fonction `main()` se résume à l'appel de la méthode `Run()` de la classe `ComputeProblem`.

```
1 #include "CommonLibsNameSpace.hpp"
2 #include "libDEM/SRC/ComputeProblem.hpp"
3
4 int main ( int argc, char * argv[] )
5 {
```

16. <http://matplotlib.sourceforge.net/>

17. <http://www.python.org/>

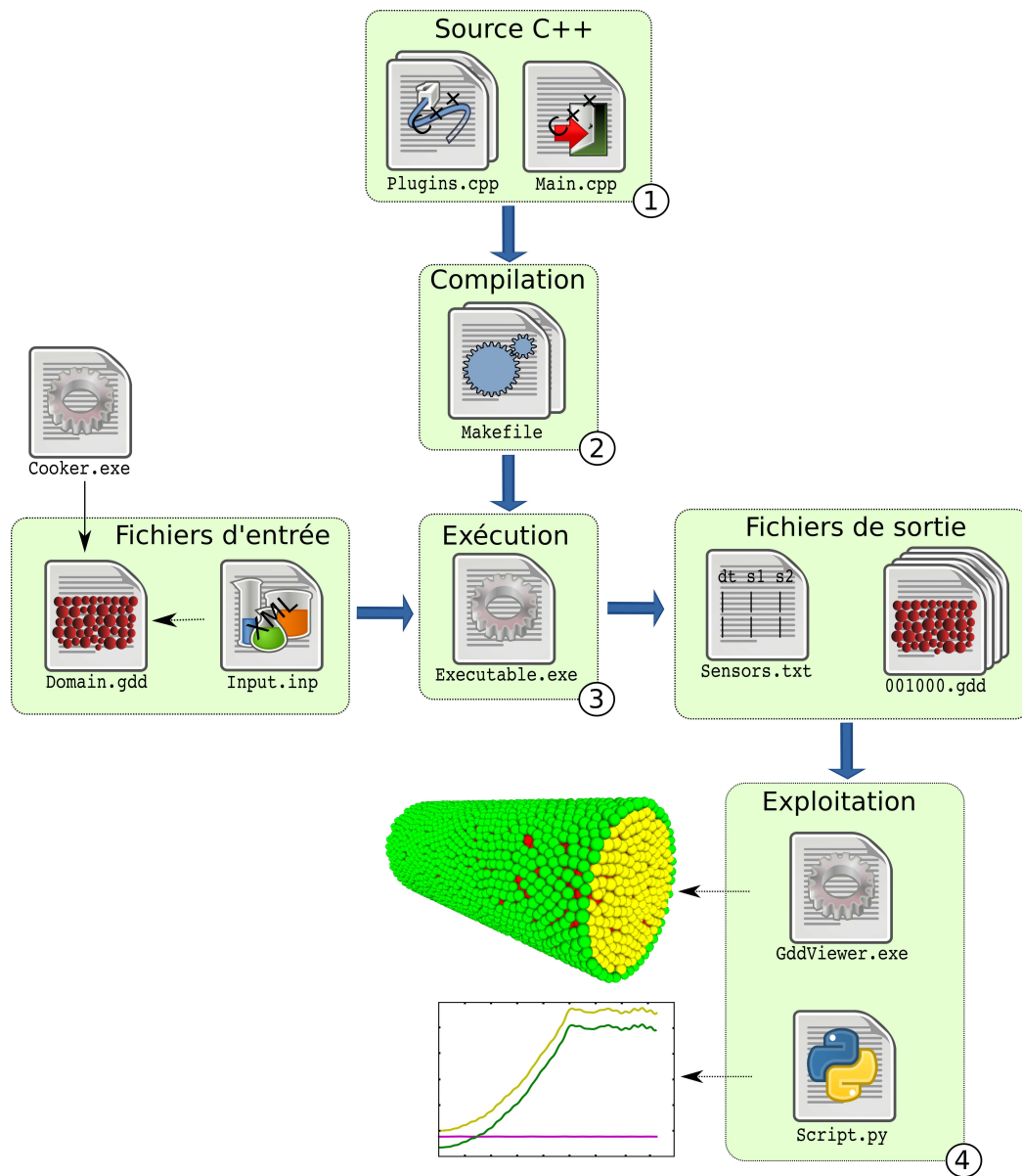


FIGURE 2.15 – Vue synthétique de l'architecture d'exploitation

```

6  return DEM::ComputeProblem<Geom::_3D>::Run(argc, argv);
7  }

```

La lecture des arguments d'exécution `argc` et `argv` est alors déléguée à la classe `ComputeProblem`. Cela permet, par exemple, de spécifier le fichier d'entrée à l'exécution du programme.

Les fichiers d'aides à la compilation : `Makefile` et `Makefile.EditMe`

L'utilitaire *gnu make* permet d'automatiser et de faciliter les tâches de compilation et d'édition des liens permettant de produire un exécutable. Ce programme est mis en œuvre au travers de fichiers de configuration où sont décrites les règles de compilation. Aussi, pour faciliter l'usage de *gnu make*, les règles de compilation sont décrites par deux fichiers : `Makefile` et `Makefile.EditMe`. Le premier, le fichier `Makefile`, est quelque peu complexe pour un utilisateur non familiarisé avec la syntaxe des *makefile*. Il est livré avec la distribution GranOO et contient l'ensemble des règles génériques et ne doit généralement pas être modifié. Le second, le fichier `Makefile.EditMe`, offre une interface très simplifiée de l'utilitaire *gnu make* :

```

# 1) give the executable name (EXE_NAME)
# 2) list the plugins to be compiled and linked (PLUGINS_CPP)
# 3) list the source files to be compiled and linked (LIST_CPP)

EXE_NAME = granoo
PLUGINS_CPP =
LIST_CPP = Main.cpp

```

Il permet à l'utilisateur de spécifier simplement :

- le nom du fichier exécutable qu'il souhaite générer ;
- la liste des greffons personnalisés à compiler (cette liste peut être vide si il est utilisé que des greffons livrés avec GranOO) ;
- la liste des fichiers sources (hors greffons) à compiler, comme le fichier `Main.cpp`.

Le fichier de description d'un domaine discret : `Cylinder.gdd`

Ce fichier correspond au domaine discret qui sera chargé lors de la simulation de traction. Il représente l'éprouvette qui sera sollicitée au cours de l'expérience numérique. Ce fichier est créé par un autre programme, *cooker*, fourni par la distribution GranOO, qui permet de construire des domaines discrets compactés de différentes formes. La figure 2.16 représente une visualisation de ce fichier à l'aide d'un autre programme fourni par GranOO : *gddViewer*.

Le fichier d'entrée : `Tension.inp`

Le fichier d'entrée `Tension.inp`, utilisant la syntaxe XML, permet à l'utilisateur d'organiser son expérience de façon claire, synthétique et paramétrable. Ainsi, un fichier d'entrée permet de définir :

1. les paramètres généraux de calcul : itérations, répertoire de stockage des fichiers résultats, pas d'intégration, etc. ;

```
<ComputeProblem TotIteration="10000" OutDir="Results" TimeStep="1e-10"/>
```

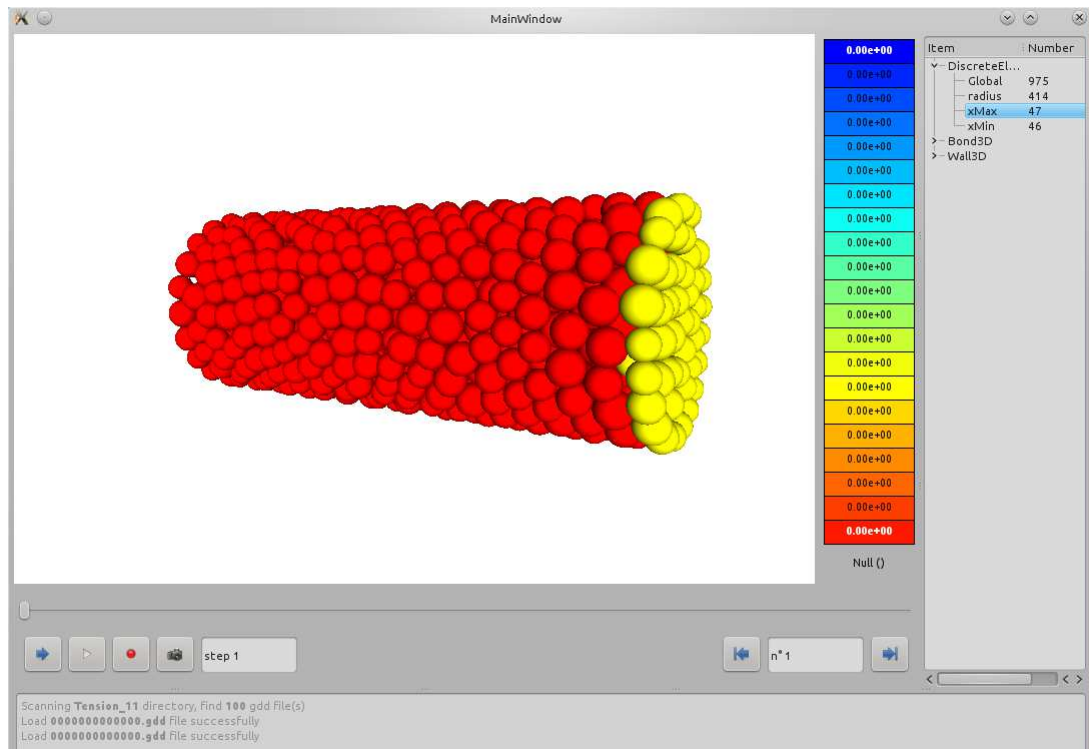



FIGURE 2.16 – Visualisation du fichier Cylinder .gdd avec le programme gddViewer

2. le domaine discret à charger ;

```
<SampleFile File="cylinder.gdd" />
```

- la séquence de greffons à exécuter en phase de *preprocessing* (avant d'entrer dans la boucle temporelle) ;
- la séquence de greffons à exécuter en phase de *processing* (à chacune des itérations de la boucle temporelle) ;
- la séquence de greffons à exécuter en phase de *postprocessing* (après la boucle temporelle) ;
- la description de conditions limites et de chargements usuels.

L'annexe B où est détaillé le fichier d'entrée décrivant une expérience de traction.

2.7.1 Synthèse

Cette section a permis d'introduire la mise en œuvre d'une expérience numérique DEM à l'aide de la plate-forme GranOO. Celle-ci se décline alors en plusieurs étapes : écritures des *greffons* C++, compilation, paramétrage de la simulation à l'aide du fichier d'entrée, exécution, puis exploitation des résultats.

GranOO permet, par l'architecture d'exploitation décrite dans cette section, de concilier deux besoins antagonistes : offrir un cadre rigoureux permettant une certaine standardisation des simulations DEM, tout en garantissant suffisamment de souplesse d'utilisation.

2.8 Conclusion

Différents aspects de la plate-forme de calcul DEM ont été introduits au cours de ce chapitre. Tout d’abord, il a été décrit de façon synthétique l’algorithme général de résolution *dynamique explicite*.

Ensuite, il a été introduit l’approche Orientée Objet (OO). Par rapport à une conception procédurale, l’approche OO, basée sur la notion d’objets où sont regroupés données et traitements, permet une modélisation fine et rigoureuse de différents concepts que doit traiter un programme informatique.

Au cours de la section 2.3, le choix du langage C++ a été justifié. Outre le support complet de la POO, le C++ accepte un grand nombre de paradigmes de programmation tout en garantissant de très bonnes performances.

La section 2.4 détaille l’architecture logicielle de la plate-forme GranOO composée de bibliothèques thématiques OO écrites en C++. Différentes solutions techniques ont permis de répondre à des contraintes fortes en terme de performance, de robustesse, d’accessibilité et d’évolutivité.

Il a été décrit ensuite, au cours de la section 2.5, l’architecture de développement retenue. Celle-ci permet une gestion efficace de développements collaboratifs. En outre, la mise en ligne au travers du portail M2P des *Arts et Métiers Paritech* permet la valorisation du travail effectué ainsi qu’une diffusion aisée. Enfin, cette organisation permet la capitalisation des différents développements utilisateurs à l’issue d’une phase de validation effectuée par l’équipe en charge du projet.

Ce chapitre s’est conclu par une mise en application de la plate-forme GranOO au travers d’un exemple simple. Cet exemple a pu montrer la convivialité de l’environnement de développement ainsi que la cohérence des démarches et des outils mis en place.

En résumé la plate-forme de calcul DEM dynamique explicite GranOO permet :

- de prototyper rapidement et facilement une simulation DEM ;
- de supporter des calculs lourds ;
- de capitaliser les développements faits par les utilisateurs ;
- d’évoluer sur la durée ;
- d’être accessible, distribuable et maintenable par l’Internet.

Dans la suite de ce mémoire, la plate-forme GranOO sera fortement mise à contribution pour réaliser diverses expériences numériques. Le chapitre suivant, consacré à la méthodologie de calibration des paramètres microscopiques, repose sur l’utilisation intensive d’essais de calibration et de validation. Ainsi, diverses éprouvettes numériques seront élaborées, puis, soumises et des sollicitations de traction, torsion et flexion, avec et sans rupture, en quasi statique et en dynamique, etc. Le dernier chapitre, consacré à l’expérience de surfacage numérique, exploite également la plate-forme GranOO. L’expérience numérique du surfacage sera donc construite. Différentes conditions d’usinage seront alors prises en compte pour permettre des comparaisons entre résultats expérimentaux et numériques.

Enfin, l’architecture d’exploitation de GranOO, reposant sur un canevas bien défini, facilite la réutilisation d’expériences numériques déjà élaborées. Concernant le surfacage, la modélisation effectuée durant la thèse peut servir alors de base de travail à des développements et des améliorations ultérieures, et de façon

plus générale, à toute modélisation DEM de type *dynamique explicite*.

CHAPITRE 3

Méthodologie de calibration

*« J'aime quand un plan se déroule
sans accroc. »*

John Hannibal Smith

Résumé

Le comportement mécanique des matériaux est généralement simulé par des approches issues de la mécanique des milieux continus. Cependant, lorsqu'il s'agit de simuler des phénomènes de multi fissurations voir de multi fracturations, les modèles de la mécanique discrète s'avèrent mieux adaptés, car ils prennent en compte naturellement les discontinuités générées par les interfaces. La difficulté est alors de s'assurer qu'une approche par éléments discrets (DEM) permet bien de retrouver le comportement mécanique au sens de la mécanique des milieux continus. Ce chapitre propose une méthodologie permettant, à partir des données connues du matériau à simuler (module de Young, coefficient de Poisson, célérité de propagation des ondes), de quantifier les paramètres microscopiques du modèle DEM.

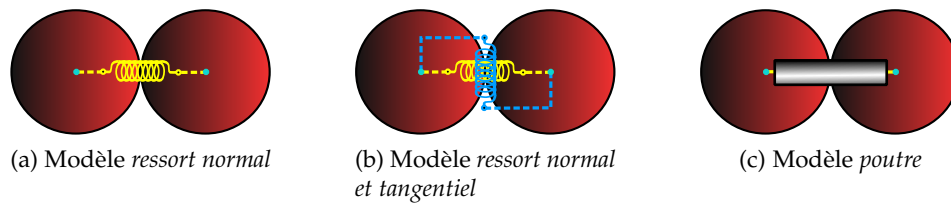


FIGURE 3.1 – Illustration de différents modèles de liaisons cohésives

Le comportement mécanique des matériaux est généralement simulé par des approches issues de la mécanique des milieux continus. Cependant, lorsqu'il s'agit de simuler des phénomènes de multi fissurations voir de multi fracturations, les modèles de la mécanique discrète s'avèrent mieux adaptés, car ils prennent en compte naturellement les discontinuités générées par les interfaces. Dans le cas du surfacage, le matériau possède une zone continue (volume qui n'est pas affecté par les SSD), une zone multi-fissurée (zone des SSD) et une zone discontinue (le troisième corps composé de particules abrasives et de silice). Contrairement aux méthodes continues, l'une des principales difficultés des méthodes DEM est la modélisation de la zone continue. Aussi, l'élaboration d'un modèle DEM quantitatif nécessite au préalable de déterminer puis de quantifier les lois d'interactions à l'échelle microscopique (voir section 1.13).

Schlangen et Garboczi [115, §3] ont comparé les influences de différents modèles de liaisons cohésives sur la géométrie de fissuration générée pour un matériau fragile. L'expérience utilisée, pour cette étude comparative, est un test de cisaillement à rupture d'éprouvette entaillée, décrite sur la figure 3.2. Les résultats diffèrent considérablement suivant le modèle choisi :

- le modèle de liaison cohésive *ressort normal* (voir figure 3.1a) ne fait apparaître qu'une seule fissure longiligne. La direction de propagation est perpendiculaire à l'éprouvette suivant l'effort de cisaillement ;
- le modèle de liaison cohésive *ressort normal et tangentiel* (voir figure 3.1b) engendre un angle d'amorçage de fissure à 45° en accord avec les observations expérimentales. Toutefois, la fissure conserve ensuite sa direction et se propage de façon rectiligne ;
- enfin, le modèle de liaison cohésive *poutre* (voir figure 3.1c) permet une géométrie de fissuration proche des observations expérimentales tel que le montre la figure 3.2.

Selon les auteurs, pour s'approcher au mieux du comportement d'un matériau continu, le modèle de liaison cohésive doit être capable d'exprimer des rigidités dans toutes les directions de l'espace en translation et en rotation. Ainsi, le modèle poutre permet de mieux modéliser le comportement d'un matériau continu.

D'autre part, la configuration géométrique du domaine discret initial influence également les propriétés mécaniques émergentes. Dans une autre référence, Schlangen et Garboczi [114, §3] ont montré que le modèle de liaison cohésive « poutre » associé à une organisation géométrique aléatoire produit des géométries de fissuration plus réalistes que pour des géométries ordonnées.

Dans le cas de domaines discrets désordonnés, le problème de la calibration des propriétés microscopiques reste ouvert (voir section 1.13). De plus, ce problème a été surtout étudié pour des applications en mécanique des milieux

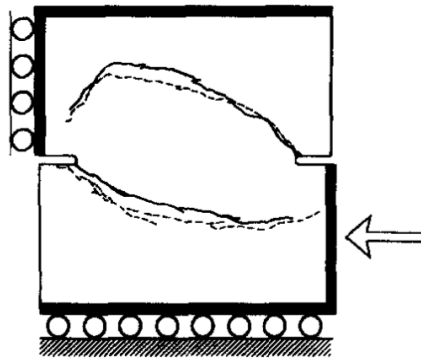


FIGURE 3.2 – Expérience de cisaillement à rupture utilisée par Schlangen et Garboczi. Image extraite de [115].

granulaires où les éléments discrets sont en interaction par contact. Par exemple, Hentz [49] propose l'utilisation d'essais d'identification permettant ensuite l'application du modèle *best-fit* établi par Liao [74]. Fakhimi et al. [29] mettent en œuvre des analyses dimensionnelles associées à des essais de calibration. Finalement, Tamarez et Plesha [126] proposent des formulations analytiques de volumes représentatifs.

Le modèle de liaison cohésive *poutre* est plus marginal. Ce modèle fut introduit en 1988 par Hermann [51]. Il a été utilisé pour des modèles *lattice* 2D ordonnés [116, 117] puis désordonnés [114, 69, 24, 53, 25]. Dans la référence [114], les auteurs considèrent que les propriétés mécaniques microscopiques doivent être égales aux propriétés mécaniques du matériau à simuler. La géométrie des poutres est déterminée de façon à satisfaire un critère d'élasticité uniforme. Les autres références ne décrivent que très partiellement la méthodologie de calibration.

En conséquence, une méthodologie de calibration du modèle de liaison cohésive *poutre*, adaptée aux méthodes particulières *dynamiques explicites*, qui permet de retrouver le comportement mécanique du matériau doit être proposé.

La méthode d'obtention de l'échantillon numérique puis la calibrations des liaisons cohésives sont les deux étapes essentielles pour l'obtention d'un modèle DEM quantitatif. La non-dépendance de la calibration vis-à-vis de la taille des éléments discrets sera également un critère déterminant.

3.1 Modèle DEM hybride « lattice - particulaire »

Le modèle DEM proposé par Schlangen et Garboczi est un modèle *lattice* (voir section 1.12). Lorsqu'une liaison cohésive est détruite, l'absence de volume ne permet pas de gérer correctement la phase de re-fermeture de fissure. Une solution, proposée par Ibrahimbegovic [53], est d'associer aux points matériels du modèle *lattice* un volume obtenu par décomposition spatiale de *Voronoi*. Toutefois, cette méthodologie est très coûteuse en temps de calcul, surtout pour des simulations en trois dimensions.

Le modèle développé dans le cadre de cette thèse est assez similaire à celui utilisé dans [19]. Le modèle *particulaire* est associé au modèle *lattice*. Le but est alors de conjuguer les avantages des deux méthodes, soit :

- de permettre, grâce au modèle *lattice*, une modélisation fine de la fissuration ;
- de permettre, grâce au modèle *particulaire*, une grande rapidité de calcul et de traitement permise par la géométrie sphérique des éléments discrets.

Les figures 3.3 illustrent la mise en œuvre de ce modèle :

la **figure 3.3a** illustre un domaine discret dont les éléments discrets sont reliés par des liaisons cohésives symbolisées par des ressorts. Ce domaine représente alors une éprouvette « de matériau continu ». L'éprouvette est en position initiale. Une action mécanique est exercée de façon à solliciter le domaine en flexion.

la **figure 3.3b** représente le domaine dans une position déformée. Pour simuler la fissuration, un critère de rupture des liaisons cohésives en déformation ou en force est introduit. Si l'une d'entre elles dépasse ce seuil, celle-ci est alors détruite définitivement. Dans l'exemple, illustré par cette image, une liaison cohésive proche de l'encastrement est détruite.

la **figure 3.3c** représente le domaine revenu dans un état proche de la configuration initiale. La liaison cohésive rompue est alors suppléée par un contact unilatéral entre les deux éléments discrets auxquels la liaison était en relation. Cela permet de prendre en compte la refermeture de fissure et d'éviter que la matière ne puisse s'interpénétrer.

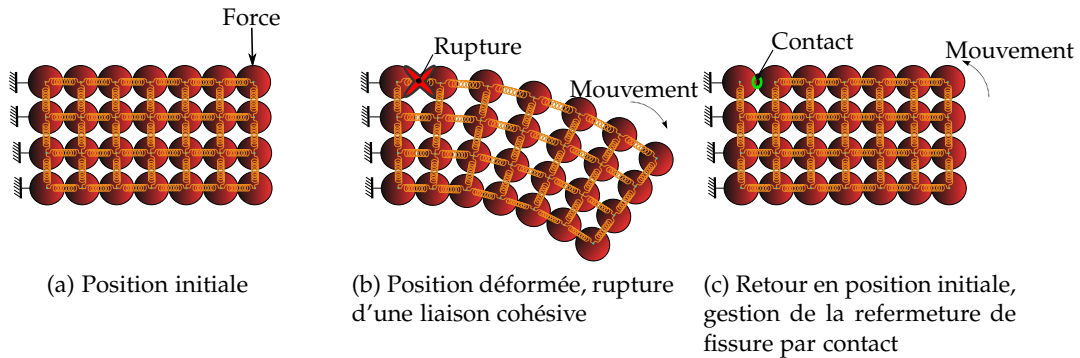


FIGURE 3.3 – Illustration du modèle hybride « lattice - particulaire »

Dans ce chapitre, il ne sera pas abordé la problématique de refermeture de fissure. Celle-ci sera introduite au cours du chapitre suivant. Seules les problématiques de calibration des liaisons cohésives et de génération du domaine initial seront traitées ici.

3.2 Modèle de liaison cohésive *poutre*

La figure 3.4a illustre la liaison cohésive de type poutre au sein d'un domaine discret. La forme choisie est de type cylindrique. Elle permet de décrire sa géométrie avec seulement deux paramètres : longueur L_μ et rayon r_μ . Deux paramètres mécaniques sont également associés : module de Young E_μ et coefficient de Poisson ν_μ .

La figure 3.4b illustre une liaison cohésive poutre dans un état déformé. Les extrémités de la poutre sont « encastées » au centre des éléments discrets O_1 et O_2 . La poutre est considérée comme relaxée lorsque l'axe $(O_1, \vec{X}_1)$ appartenant à

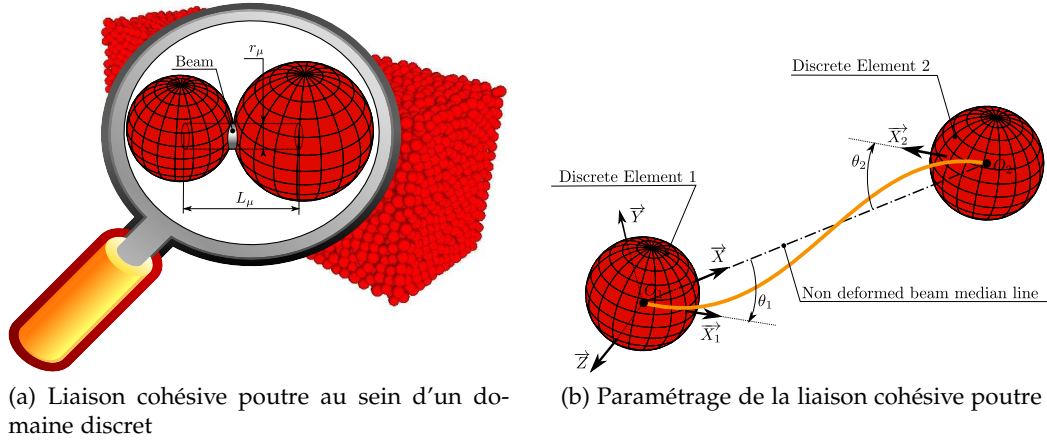


FIGURE 3.4 – Illustration de la liaison cohésive poutre

l'élément discret 1 et l'axe $(O_2, \vec{X}_2)$ appartenant à l'élément discret 2 sont directement opposés. Le repère de la poutre est alors défini par $F(O_1, \vec{X}, \vec{Y}, \vec{Z})$. Cette description permet de définir les rotations des sections droites aux extrémités $\vec{\theta}_1$ et $\vec{\theta}_2$ ainsi que l'allongement ΔL . Les déflexions (ou flèches) en O_1 et O_2 sont nulles. D'après les lois de la résistance des matériaux appliquées aux poutres d'Euler-Bernoulli, il est possible d'écrire :

$$\vec{F}_{B/DE1} = +E_\mu S_\mu \frac{\Delta L_\mu}{l_\mu} \vec{X} - \frac{6E_\mu I_\mu}{l_\mu^2} \left((\theta_{2z} + \theta_{1z}) \vec{Y} + (\theta_{2y} + \theta_{1y}) \vec{Z} \right) \quad (3.1)$$

$$\vec{F}_{B/DE2} = -E_\mu S_\mu \frac{\Delta L_\mu}{l_\mu} \vec{X} + \frac{6E_\mu I_\mu}{l_\mu^2} \left((\theta_{2z} + \theta_{1z}) \vec{Y} - (\theta_{2y} + \theta_{1y}) \vec{Z} \right) \quad (3.2)$$

$$\vec{T}_{B/DE1} = +\frac{G_\mu I_{o\mu}}{l_\mu} (\theta_{2x} - \theta_{1x}) \vec{X} - \frac{2E_\mu I_\mu}{l_\mu} \left((\theta_{2y} + 2\theta_{1y}) \vec{Y} - (\theta_{2z} + 2\theta_{1z}) \vec{Z} \right) \quad (3.3)$$

$$\vec{T}_{B/DE2} = -\frac{G_\mu I_{o\mu}}{l_\mu} (\theta_{2x} - \theta_{1x}) \vec{X} - \frac{2E_\mu I_\mu}{l_\mu} \left((2\theta_{2y} + \theta_{1y}) \vec{Y} - (2\theta_{2z} + \theta_{1z}) \vec{Z} \right) \quad (3.4)$$

Où :

- $\vec{F}_{B/DE1}$ est la force de réaction de la poutre sur l'élément discret 1 ;
- $\vec{F}_{B/DE2}$ est la force de réaction de la poutre sur l'élément discret 2 ;
- $\vec{T}_{B/DE1}$ est le moment de réaction de la poutre sur l'élément discret 1 ;
- $\vec{T}_{B/DE2}$ est le moment de réaction de la poutre sur l'élément discret 2 ;
- l_μ est la longueur initiale de la poutre ;
- ΔL_μ est l'allongement de la poutre ;
- $\vec{\theta}_1(\theta_{1x}, \theta_{1y}, \theta_{1z})$ est la rotation de la section droite au point O_1 ;
- $\vec{\theta}_2(\theta_{2x}, \theta_{2y}, \theta_{2z})$ est la rotation de la section droite au point O_2 ;
- S_μ est la section de la poutre ;
- $I_{o\mu}$ est le moment quadratique polaire de la section de la poutre ;
- I_μ est le moment quadratique de la section de la poutre selon $\vec{Y}$ et $\vec{Z}$;

- E_μ est le module de Young de la poutre ;
- G_μ est le module de cisaillement de la poutre.

Les forces et moments de réaction sont exprimés dans le repère local de la poutre F . Les poutres ne possèdent pas de masse. Les propriétés massiques sont reportées sur les éléments discrets.

Par la suite, il sera préféré la notion de rayon adimensionné, permettant d'exprimer le ratio sur le rayon de la liaison cohésive *poutre* sur le rayon moyen des deux éléments discrets :

$$\tilde{r}_\mu = \frac{r_\mu}{(r_1 + r_2)/2} \quad (3.5)$$

Avec :

- $\tilde{r}_\mu$, le rayon adimensionné de la liaison cohésive *poutre* ;
- r_1 , le rayon de l'élément discret d'indice 1 ;
- r_2 , le rayon de l'élément discret d'indice 2 ;
- r_μ , le rayon de la liaison cohésive *poutre*.

Une description détaillée du modèle de liaison cohésive *poutre* est disponible en annexe C. En résumé, le modèle de liaison cohésive *poutre* permet de définir des raideurs en traction, cisaillement, flexion et torsion avec seulement quatre paramètres : le rayon r_μ , la longueur L_μ , le module de Young E_μ et le coefficient de Poisson ν_μ .

3.3 Construction et validation du domaine compact initial

Le domaine initial doit être en accord avec les propriétés structurales du matériau à simuler. Dans le cas de la silice, celles-ci doivent être *homogène* et *isotrope*. La disposition géométrique du domaine initial impacte le comportement mécanique du matériau simulé. Aussi, cette section propose d'étudier et de qualifier le domaine généré suivant deux critères géométriques : l'homogénéité et l'isotropie.

3.3.1 Méthodologie de compaction

Des configurations géométriques ordonnées favorisent l'apparition de comportements mécaniques anisotropes et sont donc à proscrire [114, 19]. De plus, dans le cas de simulation DEM à base d'éléments sphériques, il existe des zones vides entre éléments discrets. Pour réduire ces zones de vides indésirables, une phase de compaction du domaine initial est nécessaire. Différents algorithmes de compaction existent : des méthodes "dynamiques" [78, 82] ou des méthodes géométriques [62].

Dans le cadre de cette étude, une méthode de compaction dynamique est utilisée. Tout d'abord, le domaine à compacter est généré. Les centres des éléments discrets sont placés sur une grille parallélépipédique. Une dispersion uniforme, notée χ , est appliquée aux rayons des éléments discrets.

L'opération de compaction est illustrée sur la figure 3.5b. Elle consiste à compresser le domaine par l'intermédiaire de la paroi supérieure *Pressure Wall* et de générer une instabilité par un mouvement sinusoïdal de la paroi inférieure *Shear Wall*. Les parois répulsives *Repulsive Wall* permettent de confiner le domaine dans un volume donné. La simulation est stoppée lorsque le domaine est stabilisé. Les parois sont finalement détruites pour ne garder que le domaine compacté (voir figure 3.5b).

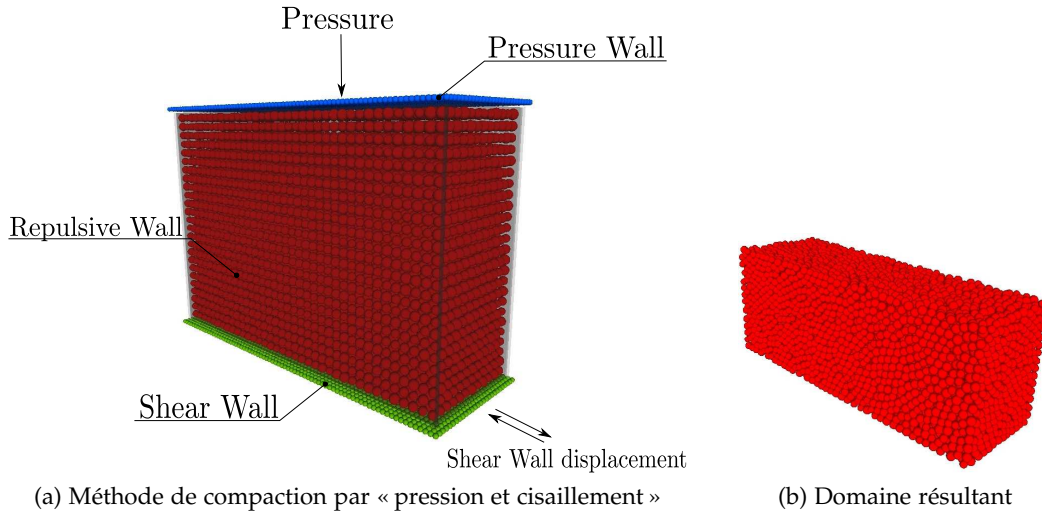


FIGURE 3.5 – Méthode de compaction

3.3.2 Homogénéité

L'application d'une loi de distribution aux rayons des éléments discrets favorise l'apparition de domaines désordonnés. Pour une distribution unimodale des rayons des éléments discrets (domaines monodisperses), le domaine compacté est de type Random Close Packing (*RCP*) [76]. Les caractéristiques inhérentes aux configurations *RCP* sont aujourd'hui bien connues. Aussi, il est possible de vérifier l'homogénéité du domaine compacté d'après les critères issus de la littérature. Ainsi, le nombre de coordination, noté n_b , doit être proche de 6 [44] et la fraction volumique, notée f , proche de 0,63 [35].

3.3.3 Isotropie

Un autre paramètre géométrique du domaine impacte fortement le comportement mécanique : l'anisotropie géométrique. Cambou dans [16, introduction et §3.6] définit ce paramètre comme la répartition des directions des contacts dans l'espace. Si cette répartition est parfaitement homogène, le domaine est considéré comme géométriquement isotrope. Pour « mesurer » l'anisotropie géométrique, les auteurs exploitent un outil mathématique : le *tenseur de fabrique*. Toutefois, cet outil est peu intuitif [65] et ne permet pas de traduire correctement tous les cas de figure. En effet, certaines configurations ordonnées se traduisent par un tenseur de fabrique sphérique, supposé représentatif d'un domaine parfaitement isotrope. Aussi, il a été développé un outil, basé sur une analyse statistique de la configuration géométrique du domaine, permettant de mieux rendre compte et de mesurer l'anisotropie géométrique.

Un solide platonique (géode, voir figure 3.6) comportant 320 facettes de morphologie similaire permet de discrétiser l'espace 3D en *angles solides discrets*. Le volume reliant le centre de la géode à une facette correspond à un angle solide. Si la direction d'un contact est comprise dans l'angle solide, celui-ci est ajouté au groupe correspondant. Ce procédé permet de regrouper les contacts suivant leurs directions privilégiées. Le poids de chaque groupe correspond au nombre de contacts dont la direction est comprise dans l'angle solide de ce groupe. L'anisotropie est alors définie par l'écart type du poids de ces groupes.

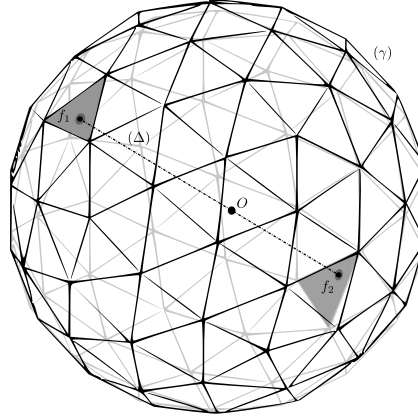


FIGURE 3.6 – Solide platonique (géode) permettant de classifier et de comptabiliser les orientations des contacts

Cette méthode permet alors d'étudier l'influence de la dispersion χ du rayon des éléments discret. Cette dispersion permet de prévenir la formation d'un arrangement géométrique ordonné [76]. Dans cette étude, la loi de dispersion est uniforme. Elle est définie par le paramètre χ :

$$\chi = \frac{R_{max} - R_{min}}{\tilde{R}} \quad (3.6)$$

Où R_{max} , R_{min} et $\tilde{R}$ sont les rayons maximum, minimum et moyen des éléments discrets.

Les figures 3.7e et 3.7f montrent les histogrammes 3D permettant de qualifier l'isotropie de domaines compactés pour :

- une dispersion de 0%, illustrées par les figures 3.7a, 3.7c et 3.7e (colonne de gauche);
- une dispersion de 25%, illustrées par les figures 3.7b, 3.7d et 3.7f (colonne de droite).

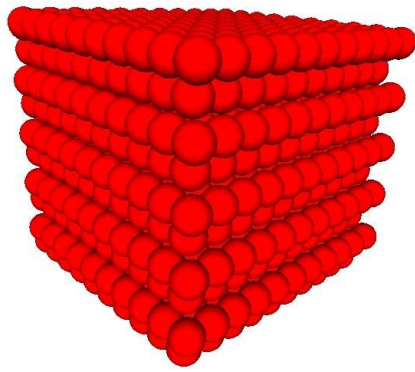
Il est observé que pour une valeur de dispersion radiale $\chi = 0\%$, le domaine compacté est fortement anisotrope. Cette configuration fait apparaître un arrangement géométrique ordonné où les contacts sont orientés suivant des directions privilégiées à 45° . Par contre, la figure 3.7f montre une distribution beaucoup plus homogène des orientations des contacts pour une valeur de $\chi = 25\%$.

Ces figures montrent que la dispersion radiale χ influence fortement l'anisotropie géométrique. Pour quantifier ce niveau d'anisotropie il est proposé de calculer l'écart quadratique moyen des fréquences d'apparition d'un contact dans un groupe directionnel f_i par rapport à une répartition uniforme représentative d'une isotropie parfaite $1/N$:

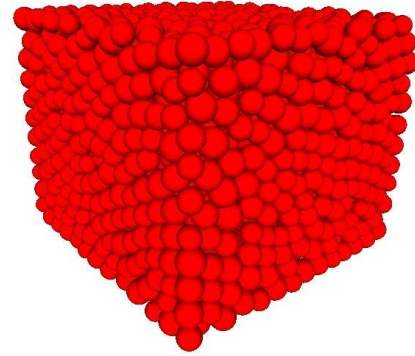
$$e = \sum_{i=1}^N \frac{(f_i - \frac{1}{N})^2}{N} \quad (3.7)$$

Où :

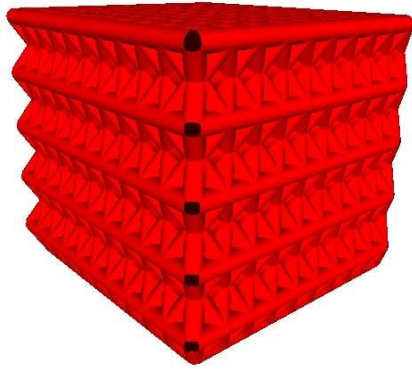
- N est le nombre total de groupe directionnel;
- f_i est la fréquence d'apparition d'un contact dans un groupe directionnel.



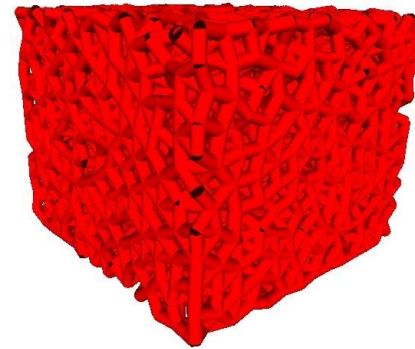
(a) Domaine compacté pour $\chi = 0\%$
(Vue des éléments discrets)



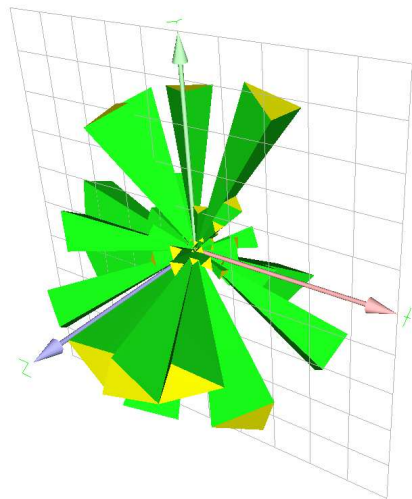
(b) Domaine compacté pour $\chi = 25\%$
(Vue des éléments discrets)



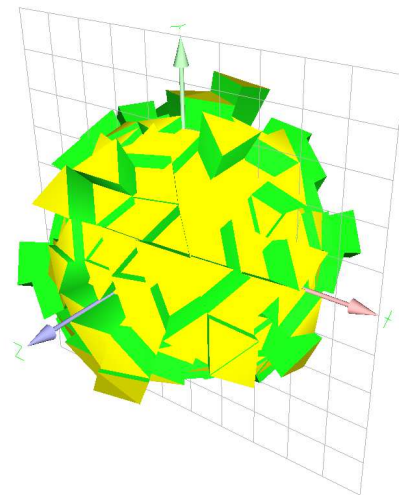
(c) Domaine compacté pour $\chi = 0\%$
(Vue des contacts)



(d) Domaine compacté pour $\chi = 25\%$
(Vue des contacts)



(e) Histogramme 3D pour $\chi = 0\%$



(f) Histogramme 3D pour $\chi = 25\%$

FIGURE 3.7 – Arrangement géométrique de domaines compactés pour différentes valeurs de dispersion radiale.

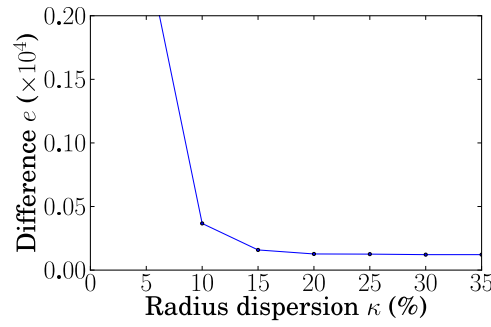


FIGURE 3.8 – Évolution de l'écart quadratique moyen e en fonction de la dispersion χ appliquée au rayon des éléments discrets

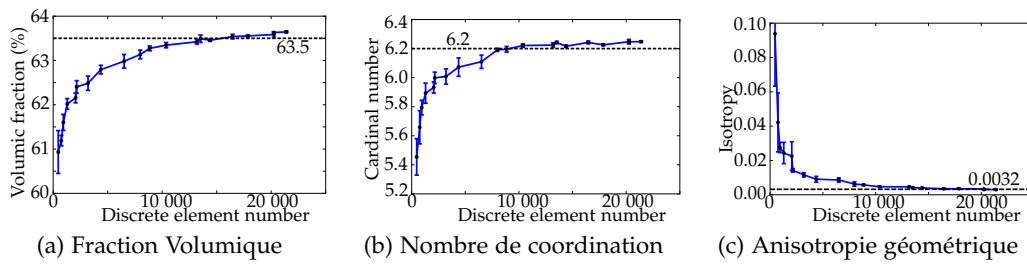


FIGURE 3.9 – Évolution des paramètres géométriques en fonction du nombre d'éléments discrets

L'aspect important de ce critère est le comportement asymptotique (voir figure 3.8). Ces résultats sont en accord avec ceux observés par Luding [104, chapitre 5] :

« *crystallization (...) does not occur for polydisperse packing with $\omega_0 \approx 0.15$* ». Pour le reste de l'étude la valeur de 25 % de dispersion radiale sera appliquée de façon à assurer l'isotropie géométrique du domaine initial.

3.3.4 Raffinement

Une autre interrogation porte sur le nombre minimum d'éléments discrets permettant d'assurer des propriétés géométriques stables. Les figures 3.9a, 3.9b et 3.9c montrent l'évolution de la fraction volumique, du nombre de coordination et de l'anisotropie en fonction du nombre d'éléments discrets pour un volume englobant constant de 1 m^3 . Ces courbes montrent une convergence des résultats pour un nombre d'environ 10 000 éléments. Il est possible alors de définir un critère 1D égal à $10000^{1/3} \approx 21,5$.

3.3.5 Synthèse

Au cours de cette section, des critères de validation du domaine compact initial ont été développés, ainsi :

- basé sur la définition du RCP, le domaine doit posséder un nombre de coordination n_b proche de 6 et une fraction volumique f proche de 0,64 ;

- une dispersion χ de 25% permet d’assurer l’isotropie géométrique. La répartition des directions des contacts est alors homogène ;
- un nombre d’éléments discrets supérieur à 21,5, pour une direction de l’espace, permet d’assurer une stabilité des propriétés géométriques du domaine.

Les liaisons cohésives poutres sont introduites après la phase de compaction du domaine : à chaque contact est associée une liaison cohésive. Dans la suite de l’étude, les contacts ne sont plus pris en compte. Seules les réactions des liaisons cohésives seront calculées.

3.4 Calibration des paramètres élastiques

La liaison cohésive *poutre* possède deux paramètres mécaniques *libres* : le module de Young microscopique, noté E_μ , ainsi que le coefficient de Poisson microscopique, noté ν_μ . Il s’ajoute également un paramètre géométrique *libre* : le rayon adimensionné microscopique $\tilde{r}_\mu$. La longueur l_μ n’est pas un paramètre libre, celle-ci est déterminée à l’issue de la phase de compaction et ne sera pas modifiée.

L’objectif de cette section est de proposer une méthodologie permettant de quantifier ces paramètres afin d’obtenir à l’échelle macroscopique les propriétés élastiques du verre de silice : module de Young de 72,5 GPa et coefficient de poisson de 0,17. Ces caractéristiques macroscopiques seront notées respectivement E_M et ν_M ¹. La méthode de détermination de ces paramètres macroscopiques repose sur la mise en œuvre d’essais numériques de traction quasi statique.

3.4.1 Présentation de l’essai numérique de traction

Les géométries d’éprouvettes numériques utilisées sont cylindriques. Ce domaine (voir figure 3.10) est généré en respectant les critères établis au cours de la section 3.3. La forme de l’éprouvette permet de respecter le critère d’élancement nécessaire à l’application des théories de la résistance des matériaux pour des poutres d’Euler-Bernoulli.

Caractéristiques géométriques de l’éprouvette discrète

Afin de quantifier et de qualifier le comportement macroscopique de l’éprouvette discrète, il est nécessaire d’associer une forme géométrique parfaite au domaine discret.

Il est proposé de définir cette géométrie comme le volume englobant le domaine discret. Ce volume sera considéré comme étant un *domaine continu équivalent* au domaine discret.

Pour définir ses dimensions, les différents éléments discrets composant les surfaces extérieures de l’éprouvette sont repérées : $xMax$, $xMin$ et $radius$ (voir 3.10). Ainsi, le cylindre parfait, associé à l’éprouvette discrète, possède comme

1. Les indices ' M ' et ' μ ' seront utilisés afin de distinguer respectivement les paramètres macroscopiques des paramètres microscopiques.

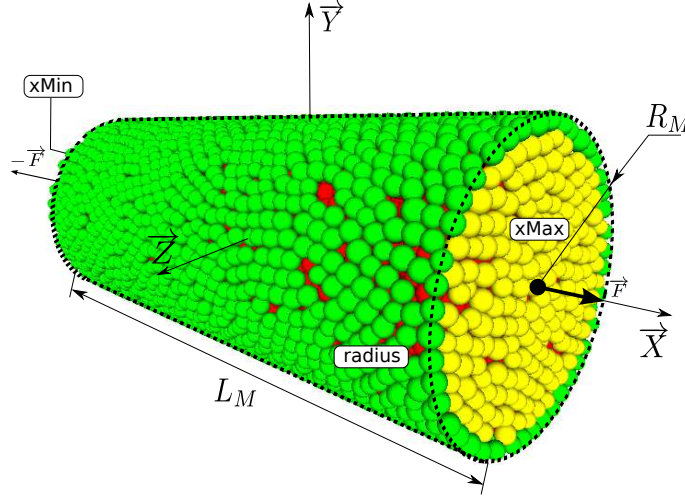


FIGURE 3.10 – Illustration de l'essai numérique de traction

dimensions :

$$L = \frac{1}{n_{xMax}} \sum_{i=1}^{n_{xMax}} \vec{OG}_i \cdot \vec{X} + \frac{1}{n_{xMin}} \sum_{i=1}^{n_{xMin}} \vec{OG}_i \cdot \vec{X} + 2\tilde{r}_{de} \quad (3.8)$$

$$R = \frac{1}{n_{radius}} \sum_{i=1}^{n_{radius}} \sqrt{(\vec{OG}_i \cdot \vec{Y})^2 + (\vec{OG}_i \cdot \vec{Z})^2} + \tilde{r}_{de} \quad (3.9)$$

Où :

- L et R représentent la longueur et le rayon du cylindre englobant ;
- n_{xMax} , n_{xMin} et n_{radius} représentent le nombre d'éléments discrets appartenant aux ensembles $xMax$, $xMin$ et $radius$;
- $\vec{OG}_i$ est le vecteur position de l'élément discret i appartenant à l'ensemble concerné ;
- $\tilde{r}_{de}$ est le rayon moyen des éléments discrets calculé sur l'ensemble du domaine.

Chargement quasi statique

Afin de solliciter l'éprouvette en traction, un chargement est appliqué sur les ensembles d'éléments discrets $xMax$ et $xMin$. La force extérieure, notée $\vec{F}$, équivaut à la somme des efforts d'intensités égales et appliquées sur les éléments discrets appartenant aux faces $xMax$ et $xMin$.

$$\vec{F} = \sum_{i=0}^n \vec{f}_i = n f \vec{X} \quad (3.10)$$

Avec :

- n , le nombre total d'éléments discrets composant le groupe ($xMax$ ou $xMin$) ;
- $\vec{f}_i$, l'effort extérieur appliqué à l'élément discret d'indice i . Dans le cas de la traction, les vecteurs forces $\vec{f}_i$ sont tous égaux et orientés selon l'axe $\vec{X}$.

Les essais de traction menés en quasi statique doivent permettre de retrouver deux caractéristiques mécaniques de l'éprouvette : le module de Young ainsi que

le coefficient de poisson. Pour un essai de traction en quasi statique les valeurs du module de Young macroscopique E_M et du coefficient de Poisson ν_M peuvent être déterminées en utilisant la théorie de la *résistance des matériaux* pour des poutres d'*Euler-Bernouilli*. Ainsi :

$$E_M = \frac{F}{S_0} \times \frac{L_0}{\Delta L} \quad (3.11)$$

$$\nu_M = -\frac{\Delta R}{R_0} \times \frac{L_0}{\Delta L} \quad (3.12)$$

Où :

- F est l'effort extérieur appliqué sur l'éprouvette (voir équation 3.10) ;
- L_0 est la longueur de l'éprouvette au repos ;
- $\Delta L = L - L_0$ est la variation de longueur de l'éprouvette ;
- R est le rayon de l'éprouvette ;
- $\Delta R = R - R_0$ est la variation de rayon de l'éprouvette ;
- $S_0 = \pi R_0^2$ est la section de l'éprouvette au repos.

Afin de s'approcher des conditions statiques, le chargement est appliqué progressivement jusqu'à atteindre une valeur limite (voir figure 3.11). La figure 3.11 montre la répartition des énergies au cours d'un essai de traction. Ce graphe permet d'observer que la contribution de l'énergie cinétique est négligeable devant l'énergie totale². L'essai peut donc être supposé comme quasi statique.

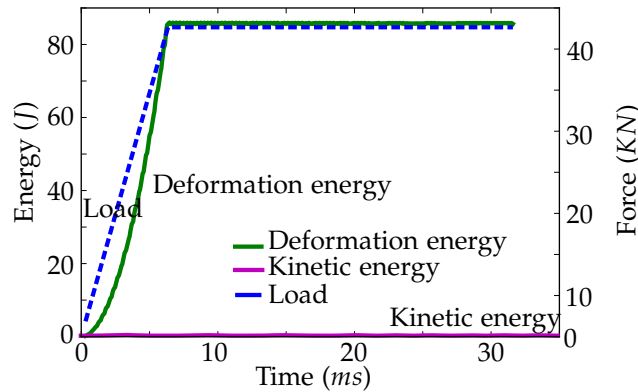


FIGURE 3.11 – Évolution du chargement et répartition des énergies au cours d'un essai de traction quasi statique

De plus, un schéma numérique amortissant est appliqué de façon à établir une convergence vers les solutions statiques. Cette méthode, élaborée par Tchamwa et Wielgosz [128], repose sur un schéma d'intégration explicite dissipatif des hautes fréquences d'oscillation [80].

La figure 3.12 représente les évolutions temporelles des valeurs calculées du module Young macroscopique et du coefficient de Poisson macroscopique donnés par les équations 3.11 et 3.12. Le système converge alors vers les solutions statiques. Ces solutions seront alors supposées être représentatives du module de Young macroscopique E_M et du coefficient de Poisson macroscopique ν_M .

2. Les méthodes de calcul des énergies sont détaillées en annexe D

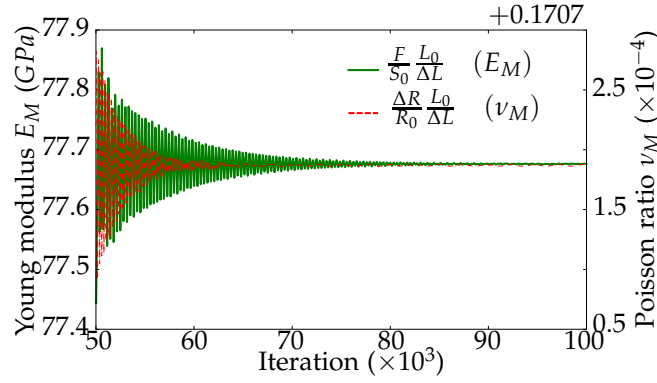


FIGURE 3.12 – Évolution des valeurs calculées de E_M et de ν_M en fonction du temps

3.4.2 Calibration du coefficient de Poisson microscopique ν_μ

La loi de comportement de la liaison cohésive « poutre » montre que le coefficient de Poisson microscopique n'est influent qu'au niveau de la sollicitation de torsion au travers du module de cisaillement (voir équations 3.3 et 3.4). Celui-ci est donné par :

$$G_\mu = \frac{E_\mu}{2(1 + \nu_\mu)} \quad (3.13)$$

Or ce mode de sollicitation joue un rôle limité comme le montre la figure 3.13. En effet, pour un chargement quelconque en quasi statique ou en dynamique, les graphiques montrent que la contribution de l'énergie de déformation de torsion est faible devant les autres énergies. Aussi le coefficient de poisson microscopique sera considéré comme peu influent devant les autres paramètres.

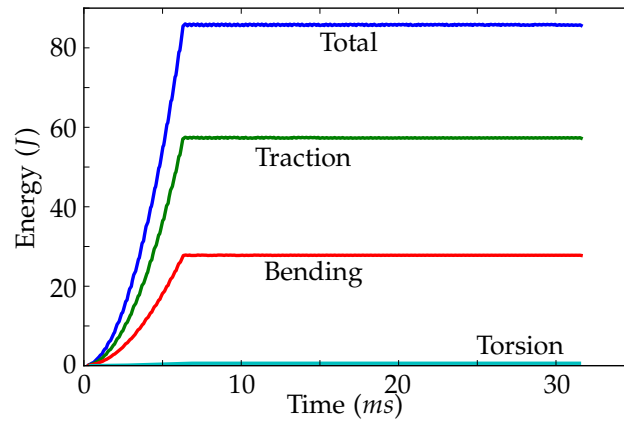
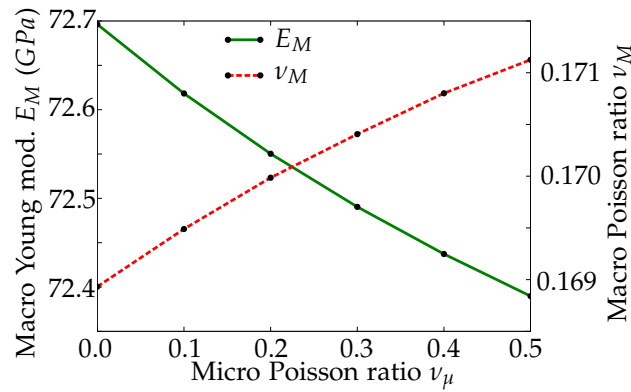


FIGURE 3.13 – Répartition des énergies de déformation pour une éprouvette cylindrique sous sollicitation quasi statique de traction, flexion et torsion

La figure 3.14 confirme cette tendance. Pour une variation du coefficient de Poisson ν_μ comprise entre 0 et 1/2, les paramètres macroscopiques ne fluctuent que de 1,2 % pour ν_M et 0,4 % pour E_M (voir tableau récapitulatif 3.1). Aussi, la valeur de 0,3 du coefficient Poisson microscopique $\nu_\mu = 0,3$ sera adoptée de façon arbitraire pour le reste de l'étude.

FIGURE 3.14 – Évolution de E_M et de ν_M en fonction de ν_μ

ν_μ	0	0,5	Différences
ν_M	0,1690	0,1711	1,2 %
E_M	72,7 GPa	72,4 GPa	0,4 %

TABLE 3.1 – Influence de ν_μ sur les paramètres macroscopiques ν_M et E_M

	E_M	ν_M
E_μ	$E_M = f_1(E_\mu)$ ↗ Croissante d'ordre 1 Figure 3.15a	$\nu_M = f_2(E_\mu)$ → Constante Figure 3.15b
$\tilde{r}_\mu$	$E_M = f_3(\tilde{r}_\mu)$ ↗↗ Croissante d'ordre 2 Figure 3.16a	$\nu_M = f_4(\tilde{r}_\mu)$ ↘↘ Décroissante d'ordre 2 Figure 3.16b

TABLE 3.2 – Tableau synthétique des influences des paramètres microscopiques E_μ et $\tilde{r}_\mu$ sur les paramètres macroscopiques E_M et ν_M

3.4.3 Calibration du module de Young microscopique E_μ et du rayon adimensionné $\tilde{r}_\mu$.

Une étude paramétrique a tout d'abord permis d'étudier l'influence des paramètres microscopiques E_μ et de $\tilde{r}_\mu$ sur le comportement macroscopique. Pour cela, différentes valeurs de E_μ et de $\tilde{r}_\mu$ ont été testées, soit :

- 9 valeurs différentes du module de Young microscopique E_μ comprises entre 20 GPa et 90 GPa avec un incrément de 10 GPa ;
- 8 valeurs différentes du rayon adimensionné $\tilde{r}_\mu$ comprises entre 0,3 et 1 avec un incrément de 0,1.

Ainsi, 72 essais numériques de traction quasi statique ont été réalisés pour mener à bien cette étude.

Les résultats sont synthétisés par le tableau 3.2 où sont présentées les influences des paramètres microscopiques E_μ et $\tilde{r}_\mu$ sur les paramètres macroscopiques E_M et ν_M .

Le tableau 3.2 et la figure 3.15b montrent qu'il existe une indépendance du coefficient de Poisson macroscopique ν_M par rapport au module de Young microscopique E_μ . Il est donc possible de calibrer E_μ sans affecter le coefficient de

Module de Young	Coefficient de Poisson
$E_M = 72.5 \text{ GPa}$	$\nu_M = 0.17$

TABLE 3.3 – Valeurs des paramètres élastiques macroscopiques de la silice

Module de Young	Coefficient de Poisson	Rayon adimensionné
$E_\mu = 269.5 \text{ GPa}$	$\nu_\mu = 0.3$	$\tilde{r}_\mu = 0.68$

TABLE 3.4 – Valeurs, obtenues par calibration de la silice, des paramètres élastiques microscopiques des liaisons cohésives *poutres*

Poisson macroscopique ν_M .

En conséquence, il est possible d'établir une méthode de calibration en trois étapes :

1. Calibration du coefficient de Poisson macroscopique ν_M avec le rayon adimensionné microscopique $\tilde{r}_\mu$. Dans le cas de la silice $\tilde{r}_\mu^{silica} \approx 0.68$ correspond à $\nu_M^{silica} \approx 0.17$ (voir figure 3.17a) ;
2. Mise en œuvre d'essais de traction avec la nouvelle valeur $\tilde{r}_\mu$ pour obtenir l'évolution de E_M en fonction de E_μ ;
3. Calibration du module de Young macroscopique E_M avec le module de Young microscopique E_μ . Dans le cas de la silice $E_\mu^{silica} \approx 269.5 \text{ GPa}$ correspond à $E_M^{silica} \approx 72.5 \text{ GPa}$ (voir figure 3.17b).

3.4.4 Validation

La section précédente a permis de quantifier les paramètres microscopiques en regard des propriétés élastiques de la silice synthétisée sur le tableau 3.3. Le tableau 3.4 récapitule les valeurs microscopiques trouvées. Ces valeurs ont permis d'élaborer des échantillons numériques cylindriques de silice. Pour valider les résultats obtenus, cet échantillon a été soumis à des tests quasi statiques de traction, flexion et torsion. Pour chacun de ces tests, l'une des extrémités est encastree et le chargement est appliqué à l'autre extrémité. Pour assurer l'aspect quasi statique, le protocole défini à la section ?? a été reproduit. Les déplacements et les rotations de l'extrémité libre sont obtenus par moyenne des déplacements et des rotations des éléments discrets appartenant à cette surface.

Les résultats sont synthétisés sur le tableau 3.5. Ils sont comparés aux résultats issus de la théorie de la résistance des matériaux. La comparaison des deux modèles montre des différences inférieures à 7 %.

Une source d'erreur possible peut être due au manque de précision de la mesure du rayon du domaine continu équivalent. Une imprécision de cette mesure entraîne un écart d'ordre 4 sur le calcul du moment quadratique. La figure 3.18 illustre ce problème. D'après la définition de la géométrie du domaine continu établi en section 3.4.1, le rayon du domaine continu est celui noté *Max radius* sur la figure. Or la section effective où travaillent les liaisons cohésives est plus proche de celle notée *Effective section*. En utilisant cette dernière définition l'écart est ramené alors à 1,8 % au lieu de 6.63 %, mais impacte négativement les autres résultats. L'équivalence géométrique entre domaine discret et domaine continu

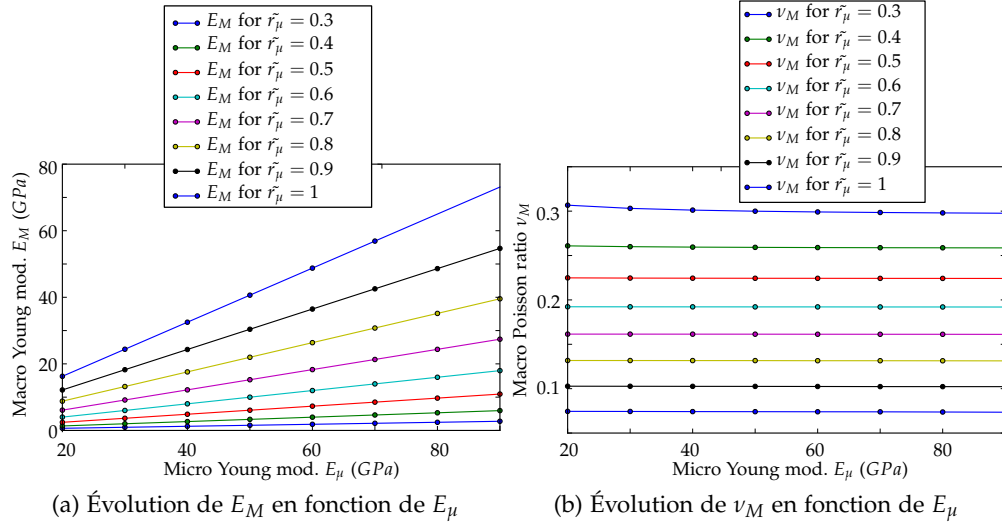


FIGURE 3.15 – Influence du module de Young microscopique E_μ sur les paramètres macroscopiques E_M et ν_M

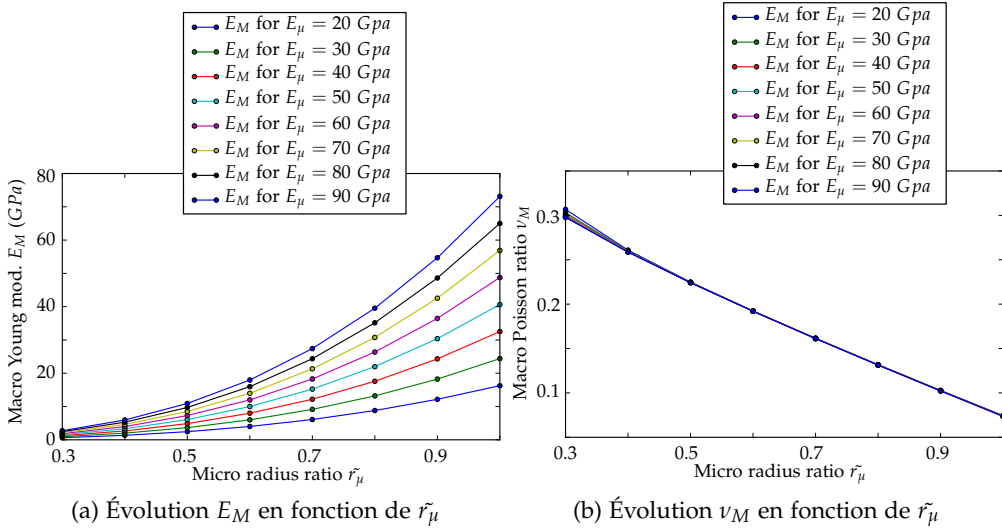


FIGURE 3.16 – Influence du rayon adimensionné microscopique $\tilde{r}_\mu$ sur les paramètres macroscopiques E_M et ν_M

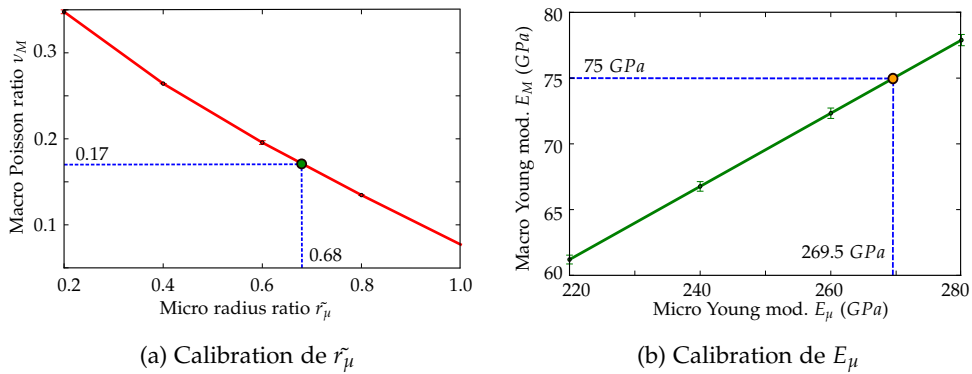


FIGURE 3.17 – Calibration des paramètres microscopiques $\tilde{r}_\mu$ et E_μ du matériau silice

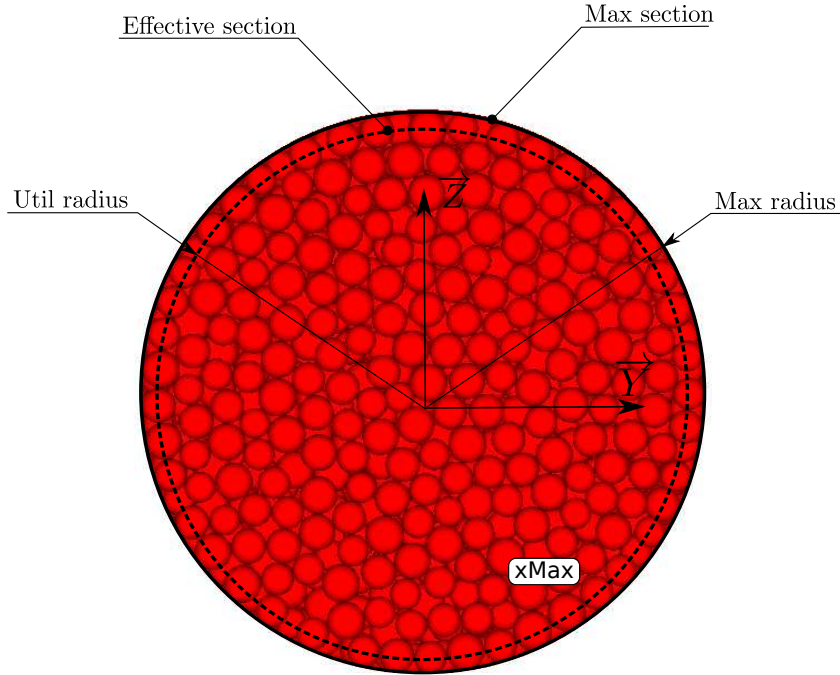


FIGURE 3.18 – Calcul du rayon du domaine continu équivalent

	Traction	Flexion	Torsion
Critère	Déplacement normal de l'extrémité libre	Déplacement tangentiel de l'extrémité libre (flèche)	Rotation de l'extrémité libre
Différence	1.20 %	4.16 %	6.13 %

TABLE 3.5 – Comparaison de résultats numériques et théoriques pour des essais de traction, flexion et torsion

est donc un point délicat qui demande encore des améliorations ultérieures. Toutefois, il sera estimé que, en l'état, les précisions apportées par ce modèle sont suffisantes.

3.4.5 Étude de convergence

Le but de cette étude est d'étudier l'influence du *maillage*³ sur les propriétés élastiques macroscopiques émergentes E_M et ν_M . Pour cela, différentes éprouvettes numériques ont été construites. Celles-ci sont de dimensions identiques, mais réalisées à l'aide de maillages de plus en plus fins (voir figures 3.19). Un aspect statistique est introduit en réalisant également quatre éprouvettes différentes pour une même valeur de finesse du maillage. Pour chacune de ces éprouvettes, les valeurs $\tilde{r}_\mu$, E_μ et ν_μ des liaisons cohésives *poutres* sont identiques.

Les résultats de cette étude sont présentés sur la figure 3.20. À partir de dix milles éléments discrets les valeurs du coefficient de poisson ν_M et du module de Young E_M macroscopique sont comprises autour d'une valeur moyenne à 3% près pour E_M et 2,5% pour ν_M .

3. Le terme *maillage* est employé ici pour désigner la géométrie du réseau de liaison cohésive.

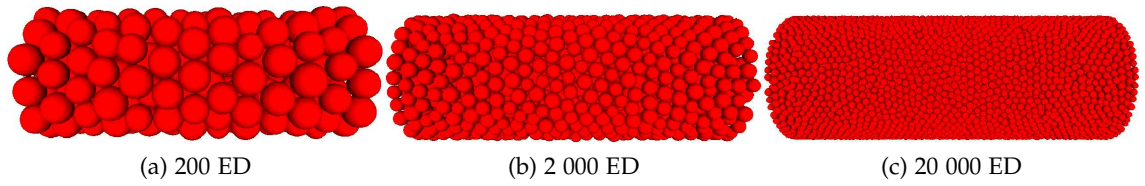


FIGURE 3.19 – Différents échantillons numériques utilisés pour l'étude de convergence

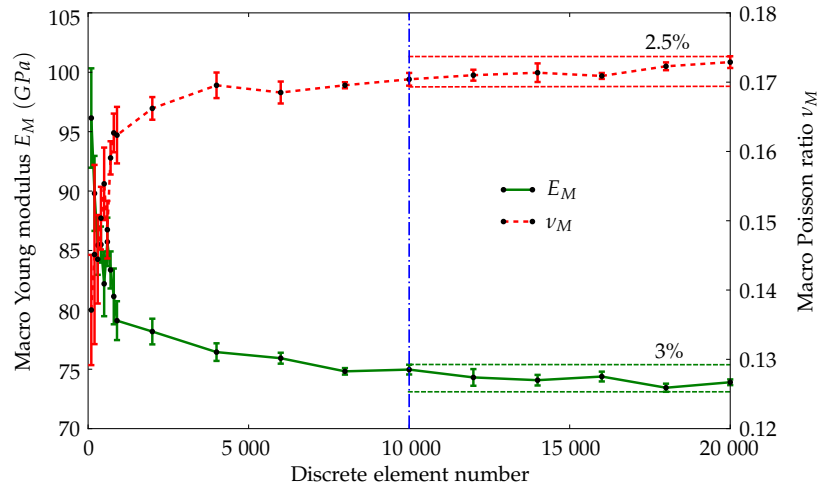


FIGURE 3.20 – Évolution des paramètres macroscopiques en fonction du nombre d'éléments discrets

C'est ici un résultat important. En effet, les méthodes de calibration comme le propose Liao [74] sont moins précises. Hentz [49] montra qu'à l'aide de cette dernière méthode de calibration, la dispersion des propriétés macroscopiques est de 28 %. Aussi, il propose d'introduire un coefficient de correction basé sur un critère énergétique, permettant de ramener la dispersion à un niveau acceptable de 10 %. Toutefois, ce coefficient est dépendant du maillage et doit être recalculé pour chacune des éprouvettes numériques.

Une meilleure précision est ici obtenue sans l'aide de coefficients de correction. Aussi le modèle de liaison cohésive *poutre* est moins sensible au maillage. En conséquence, il sera supposé pour la suite de ce mémoire que la précision de 3% pour E_M et de 2,5% pour ν_M est suffisante et que les propriétés élastiques macroscopiques sont *indépendantes* du maillage.

Par conséquent, un jeu de données de paramètres microscopiques $\tilde{r}_\mu$, E_μ et ν_μ déterminé à l'issue de la phase de calibration sera valable pour n'importe quel domaine discret, à condition que le nombre d'éléments discrets soit suffisant et que le domaine soit compacté de façon à respecter les critères établis dans la section 3.3.

3.5 Calibration des paramètres dynamiques

La section précédente a mis en évidence l'influence des paramètres microscopiques des liaisons cohésives (module de Young, coefficient de Poisson et rayon adimensionné) sur le comportement élastique macroscopique. Il est alors possible de retrouver à l'échelle macroscopique les valeurs du module de Young et du coefficient de Poisson désirées. Bien que nécessaire, cela n'est pas suffisant. Afin de simuler des phénomènes dynamiques tels que les propagations d'ondes mécaniques ou des phénomènes de multi fissurations, il est nécessaire de calibrer également les propriétés massiques du domaine discret.

Les paramètres massiques des éléments discrets (masse et inertie) dépendent du volume et de la masse volumique des éléments discrets. Or, la géométrie des éléments discrets est déterminée lors de la phase de construction du domaine et ne peut être ajustée. En conséquence, seule la masse volumique peut permettre de calibrer les propriétés dynamiques.

La méthode retenue est très simple. Il s'agit de compenser les vides générés par la méthode de compaction en assurant une égalité de masse entre domaine discret et *domaine continu équivalent*. La géométrie du domaine continu équivalent est donnée par le volume englobant décrit dans la section 3.4.1. La masse volumique des éléments discrets est donc ajustée en conséquence :

$$\rho_\mu = \frac{\rho_M V_M}{\sum_{i=1}^N V_{\mu_i}} = \frac{\rho_M}{f} \quad (3.14)$$

Où :

- ρ_μ est la masse volumique des éléments discrets. Tous les éléments discrets possèdent la même masse volumique ;
- V_{μ_i} est le volume de l'élément discret d'indice i ;
- ρ_M est la masse volumique du matériau à simuler ;
- V_M est le volume de la géométrie englobante associée au domaine discret ;
- f est la fraction volumique.

Toutefois, cette approche basée sur une équivalence de masse entre domaine discret et domaine continu ne permet pas d'assurer une égalité des inerties. L'étude suivante met en lumière ce phénomène.

3.5.1 Le problème de l'inertie

Pour étudier l'influence de l'inertie, une série d'études de convergence sur des domaines 1D a été menée. Un domaine discret composé de particules alignées est chargé dynamiquement en traction, flexion puis torsion. L'élément discret de gauche est fixe. Le chargement est appliqué à l'élément discret de droite (voir figure 3.21). Afin d'exciter les modes de vibration propre de la poutre, le chargement appliqué est de type échelon. La période d'oscillation du premier mode sera alors la variable d'étude.

L'étude de convergence porte sur le nombre d'éléments discrets mis en jeu (de 2 à 20). La variable d'étude porte sur les périodes d'oscillation de l'élément discret *libre* (élément discret de droite). Les résultats numériques sont comparés aux solutions analytiques issues de la vibration des milieux continus dans le cas

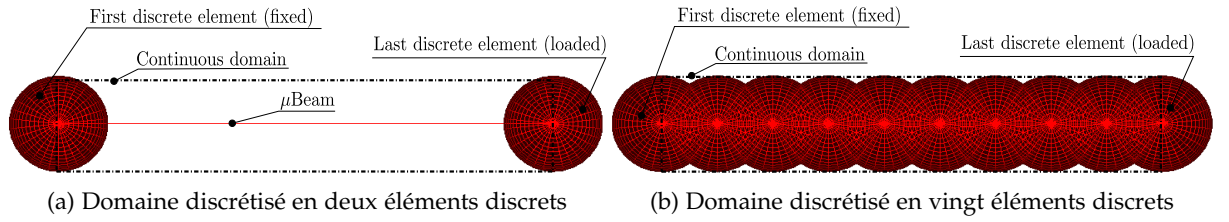


FIGURE 3.21 – Illustration des domaines d'étude de convergence dynamique

de poutres d'Euler-Bernouilli [119, §4.1, §4.2, §4.3] :

$$T_{traction} = 4L\sqrt{\frac{\rho_M}{E}} \quad (3.15)$$

$$T_{flexion} = \frac{2L^2}{\pi}\sqrt{\frac{\rho_M S}{EI}} \quad (3.16)$$

$$T_{torsion} = 4L\sqrt{\frac{\rho_M}{G}} \quad (3.17)$$

Où :

- $T_{traction}$, $T_{flexion}$ et $T_{torsion}$ sont les périodes d'oscillation du premier mode propre d'une poutre encastree-libre pour des sollicitations de traction, flexion et torsion ;
- E , G sont respectivement le module de Young et le module de cisaillement de la poutre ;
- ρ_M est la masse volumique de la poutre ;
- L , S , I sont respectivement la longueur, la section, et le moment quadratique transverse de la poutre.

La masse volumique des éléments discrets est calibrée de façon à assurer une égalité de masse entre domaine discret et domaine continu. Ainsi :

$$\rho_\mu = \frac{3\rho_M L}{4NR} \quad (3.18)$$

Où :

- N est le nombre d'éléments discrets ;
- ρ_μ est la masse volumique des éléments discrets.

Le domaine discret étant longiligne, les paramètres microscopiques des liaisons cohésives *poutre* sont égales aux paramètres macroscopiques :

$$\begin{aligned} E_\mu &= E_M \\ \nu_\mu &= \nu_M \\ r_\mu &= R \end{aligned} \quad (3.19)$$

Les résultats obtenus montrent que pour le cas de l'essai :

de traction (voir figure 3.22a), la solution numérique converge vers la solution analytique.

de flexion (voir figure 3.22b), la solution numérique converge avec une légère erreur vers la solution analytique. Cet écart peut être interprété comme le

résultat de la différence entre inertie du domaine discret et domaine continu équivalent. Toutefois, celle-ci est faible et concorde avec une hypothèse classique de la théorie de vibration des poutres en flexion : *l'énergie cinétique de rotation est négligeable devant l'énergie cinétique de translation*. Par conséquent l'écart de calibration des inerties a un effet minime sur les résultats obtenus.

de torsion (voir figure 3.22c), la solution numérique converge avec un écart d'environ 10% par rapport à la solution théorique. En effet, dans ce cas l'hypothèse précédente n'est pas valable, car seule l'énergie cinétique de rotation est présente.

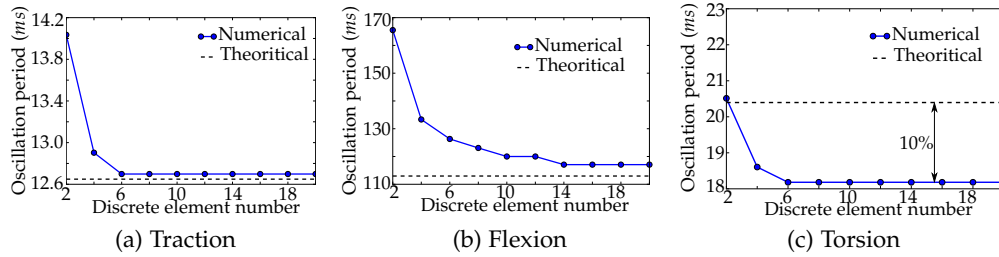


FIGURE 3.22 – Période d'oscillation sous sollicitation dynamique

Cette étude montre une bonne concordance des résultats numériques et théoriques pour des sollicitations de traction et de flexion. Le cas de la torsion est plus problématique. Toutefois, l'écart mesuré de 10% est invariant. La démonstration de ce phénomène est établie en annexe E. De plus, comme le montre les bilans énergétiques illustrés sur les figures 3.13, les modes de sollicitation des liaisons cohésives poutres pour des domaines discrets massifs⁴, sont principalement la traction et la flexion. Aussi il est supposé que l'influence de cet écart est négligeable pour des simulations DEM massives.

3.5.2 Validation sur des domaines discrets massifs

Cette section propose de valider la méthode de calibration des paramètres dynamiques. Des domaines discrets massifs (voir figure 3.10) sont soumis à des essais dynamiques de traction, flexion, torsion et de choc.

L'échantillon numérique correspond au matériau silice dont les paramètres mécaniques et massiques sont :

- un module de Young de $E_M = 72.5 \text{ GPa}$;
- un coefficient de Poisson de $\nu_M = 0.17$;
- une masse volumique de $\rho_M = 2201 \text{ kg/m}^3$.

Les propriétés microscopiques élastiques sont déduites de la méthode de calibration décrite en section 3.4 soit :

- un module de Young microscopique de $E_\mu = 269,5 \text{ GPa}$;
- un coefficient de Poisson de $\nu_\mu = 0,3$;
- un rayon adimensionné de $\tilde{r}_\mu = 0,68$;

Les propriétés massiques sont déduites de l'égalité de masse entre domaine discret et domaine continu à l'aide de l'équation 3.14. Ainsi, la fraction volumique

4. C'est à dire faisant intervenir un nombre important d'éléments discrets (voir par exemple figure 3.10).

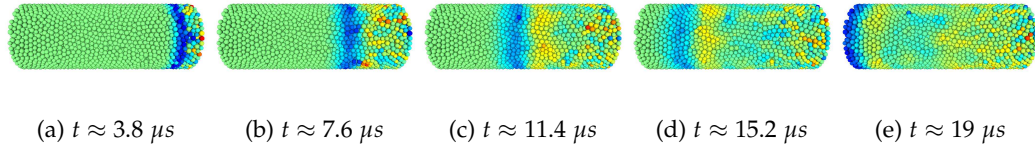


FIGURE 3.23 – Illustration du déplacement du front d’onde mécanique
Le code couleur représente les vitesses des éléments discrets projetées sur l’axe $\vec{x}$

étant égale à environ 63%, la masse volumique microscopique est donc :

$$\rho_\mu \approx 3500 \text{ kg/m}^3$$

Comme pour la section précédente, les résultats numériques des essais dynamiques de traction, flexion, et torsion sont comparés aux périodes d’oscillation, pour une configuration *encastrée-libre*, données par les solutions analytiques (voir équations 3.15, 3.16 et 3.17). Les périodes d’oscillation numériques sont déduites des déplacements et des rotations des éléments discrets appartenant à la face libre, puis traitées par analyse fréquentielle (FFT).

L’essai de choc est illustré par la figure 3.23. Une vitesse initiale est appliquée sur la face $xMax$ pour générer une onde de compression. Pour « mesurer » la vitesse de propagation des ondes mécaniques, la vitesse moyenne des éléments discrets appartenant à la face opposée $xMin$ est enregistrée durant la simulation (voir figure 3.24). Une évolution brusque de celle-ci permet de connaître le moment où le front d’onde est réfléchi. Ainsi, connaissant le temps nécessaire à l’onde pour parcourir l’éprouvette et la longueur de celle-ci, il est possible d’en déduire la célérité de propagation de l’onde mécanique. Cette valeur est alors comparée à la solution théorique donnée par :

$$c = \sqrt{\frac{E}{\rho}} \quad (3.20)$$

Le tableau 3.6 résume les résultats obtenus. Ils montrent une bonne concordance entre modèles théoriques et numériques pour les essais de traction, de torsion et de choc. Le cas de la flexion montre un écart plus grand. Une source d’erreur possible peut être due au manque de précision de la mesure du rayon du domaine continu équivalent détaillée dans la section 3.4.4.

3.6 Calibration de la rupture

Les sections précédentes ont permis de quantifier les propriétés microscopiques en regard des propriétés élastiques et dynamiques de la silice. Cette section propose une méthodologie de calibration des paramètres microscopiques de rupture. La démarche proposée ici est identique à celle de la section précédente : présentation du critère et justification théorique, essais de calibration puis validation des résultats.

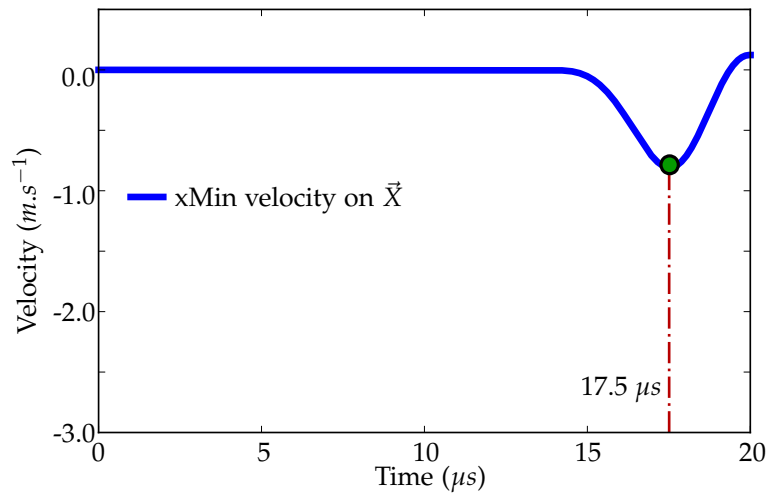


FIGURE 3.24 – Mesure de la vitesse moyenne des éléments discrets projetée sur l'axe $\vec{x}$ de la face $xMin$

	Traction	Flexion	Torsion	Choc
Critère	Période d'oscillation des déplacement normaux de la face libre	Période d'oscillation des déplacement transversaux de la face libre	Période d'oscillation des rotations de la face libre	Célérité des ondes mécaniques
Différences	0.38 %	6.63 %	0.50 %	0.40 %

TABLE 3.6 – Comparaison des modèles numériques et théoriques pour des essais dynamiques de traction, flexion, torsion et choc

3.6.1 Critère microscopique de rupture

Le critère de rupture est introduit au niveau des liaisons cohésives *poutres*. Différents modèles peuvent alors être utilisés : critère de déformation maximale [53], de contrainte maximale [114] ou d'énergie de déformation maximale [58].

Le critère choisi est de type *contrainte maximale* dérivé du critère de *Rankine* adapté aux matériaux fragiles. Ce critère est basé sur le calcul de la contrainte principale maximale [102, §3.2]. En effet, pour les matériaux fragiles le mode de rupture prépondérant est le mode I dit *d'ouverture de fissure* qui correspond à la séparation des interfaces de fissure sous l'action de sollicitations de traction [5, §2.3.1].

Aussi, il sera supposé que la rupture ne peut apparaître que lorsque la liaison cohésive *poutre* est sollicitée en traction et pas en compression. La contrainte principale maximale doit donc être positive. Ce dernier point est d'ailleurs vérifié expérimentalement. Il est observé pour les matériaux fragiles que la contrainte à rupture en compression est très supérieure à la contrainte à rupture en traction. Dans le cas de la silice, ce facteur est de 1000 (voir tableau 1.3 des propriétés mécaniques au chapitre 1).

Aussi, dans cas de poutre d'Euler-Bernouilli, la contrainte principale maximale est donnée par [102, §2.3.2] :

$$\frac{1}{2} \left(\sigma_{max} + \sqrt{\sigma_{max}^2 + 4\tau_{max}^2} \right) \quad (3.21)$$

Où :

- σ_{max} est la contrainte normale maximale ;
- τ_{max} est la contrainte de cisaillement maximale.

La rupture d'une liaison cohésive *poutre* intervient lorsque sa valeur de contrainte principale maximale dépasse un seuil σ_{μ_f} :

$$\sigma_{\mu_f} \geq \frac{1}{2} \left(\sigma_{\mu_{max}} + \sqrt{\sigma_{\mu_{max}}^2 + 4\tau_{\mu_{max}}^2} \right) \quad (3.22)$$

La contrainte normale maximale est calculée au milieu de la liaison cohésive *poutre*. Elle est donnée par :

$$\begin{aligned} \sigma_{\mu_{max}} &= \sigma_{\mu_{max}}^{fl} + \sigma_{\mu_{max}}^{tr} \\ \sigma_{\mu_{max}}^{fl} &= \frac{1}{2} \left\| \overrightarrow{M_{\mu_{fB/DE1}}} + \overrightarrow{M_{\mu_{fB/DE2}}} \right\| \frac{r_{\mu}}{I_{\mu_g}} \\ \sigma_{\mu_{max}}^{tr} &= \frac{N_{\mu}}{S_{\mu}} \end{aligned} \quad (3.23)$$

Où :

- $\overrightarrow{M_{\mu_{fB/DE1}}}$ et $\overrightarrow{M_{\mu_{fB/DE2}}}$ sont les moments de flexion, aux extrémités de la liaison cohésive *poutre*, déduits des équations 3.3 et 3.4 ;
- $\sigma_{\mu_{max}}^{fl}$ est la contrainte normale maximale due aux sollicitations de flexion ;
- $\sigma_{\mu_{max}}^{tr}$ est la contrainte normale maximale due aux sollicitations de traction ;
- r_{μ} est le rayon de la liaison cohésive *poutre* ;
- I_{μ_g} est le moment quadratique de la liaison cohésive *poutre* : $I_g = \pi r_{\mu}^4 / 4$;
- N_{μ} est l'effort normal de la liaison cohésive *poutre* ;
- S_{μ} est la section de la liaison cohésive *poutre* : $S = \pi r_{\mu}^2$.

3.6.2 Présentation de l'essai de traction quasi statique à rupture

La calibration du critère de rupture microscopique σ_{μ_f} s'appuie sur des essais de traction à rupture quasi statique.

Un chargement, dont l'intensité augmente progressivement, est appliqué aux faces *xMax* et *xMin* jusqu'à l'obtention d'une rupture « franche ». Une rupture est considérée comme « franche », lorsqu'une augmentation brutale du nombre de liaisons cohésives rompues apparaît. Comme pour des essais à rupture expérimentaux, un mode de chargement « en déplacement » sera préféré. Ce mode opératoire permet d'obtenir des cinétiques de fissuration plus rapides et de discriminer de façon plus précise une éprouvette rompue d'une éprouvette non fissurée.

Aussi, contrairement aux essais de traction de calibration élastique, (voir section 3.4.1) le test de rupture est piloté en déplacement. Des déplacements $\lambda(t)$ et $-\lambda(t)$ selon l'axe $\vec{X}$ sont appliqués aux extrémités de la poutre macroscopique, *xMax* et *xMin* (voir figure 3.10). Le déplacement imposé $\lambda(t)$ est une fonction li-

néaire croissante par rapport au temps. Cela permet de charger progressivement la structure jusqu'à rupture. Cette condition limite en déplacement se traduit au niveau des éléments discrets par :

$$\vec{d}_i^{xMax} \cdot \vec{X} = \lambda \quad (3.24)$$

$$\vec{d}_i^{xMin} \cdot \vec{X} = -\lambda \quad (3.25)$$

Où $\vec{d}_i^{xMax}$ et $\vec{d}_i^{xMin}$ sont les vecteurs déplacement des éléments discrets i appartenant respectivement aux faces $xMax$ et $xMin$.

La figure 3.25 montre l'évolution du nombre de liaisons cohésives rompues au cours d'un essai de traction. Une évolution brutale du nombre de liaisons rompues permet alors d'identifier l'instant où s'est produite la fissure.

Afin de déterminer la contrainte macroscopique à rupture σ_{M_f} , des capteurs numériques sont introduits aux niveaux des éléments discrets appartenant aux faces $xMax$ et $xMin$. Ils permettent d'enregistrer les efforts $\vec{f}_i$ et moments extérieurs $\vec{m}_i$ appliqués à ces éléments. Ainsi, il est possible de déterminer les efforts normaux macroscopiques, pour chaque itération de la simulation, des faces $xMax$ (N_M^{xMin}) et $xMin$ (N_M^{xMax}) en sommant les efforts $\vec{f}_i$:

$$N_M^{xMin} = \sum_{i=1}^{n_{xMin}} \vec{f}_i \cdot \vec{X}$$

$$N_M^{xMax} = \sum_{i=1}^{n_{xMax}} \vec{f}_i \cdot \vec{X}$$

Les déplacements imposés aux faces $xMax$ et $xMin$ étant d'intensités égales et opposées il sera supposé que l'effort normal macroscopique N_M est constant le long de la poutre macroscopique. Celui-ci sera alors considéré alors comme la moyenne des efforts normaux macroscopiques mesurés aux faces $xMax$ et $xMin$:

$$N_M = \left(\left| N_M^{xMax} \right| + \left| N_M^{xMin} \right| \right) / 2$$

Il est alors possible de déterminer la contrainte normale macroscopique σ_M d'après les lois de la résistance des matériaux :

$$\sigma_M = \frac{N_M}{S_M} = \frac{N_M}{\pi R_M^2} \quad (3.26)$$

La figure 3.25 montre également les contraintes macroscopiques normales mesurées aux deux extrémités de l'échantillon. Celles-ci sont déduites à partir des efforts normaux macroscopiques N_M^{xMin} et N_M^{xMax} :

$$\sigma_M^{xMin} = \frac{N_M^{xMin}}{S_M}$$

$$\sigma_M^{xMax} = \frac{N_M^{xMax}}{S_M}$$

Les évolutions de σ_M^{xMin} et σ_M^{xMax} visibles sur la figure 3.25 montrent :

1. une bonne concordance des valeurs de σ_M^{xMin} et σ_M^{xMax} avant rupture. Les évolutions des contraintes normales après rupture sont les conséquences

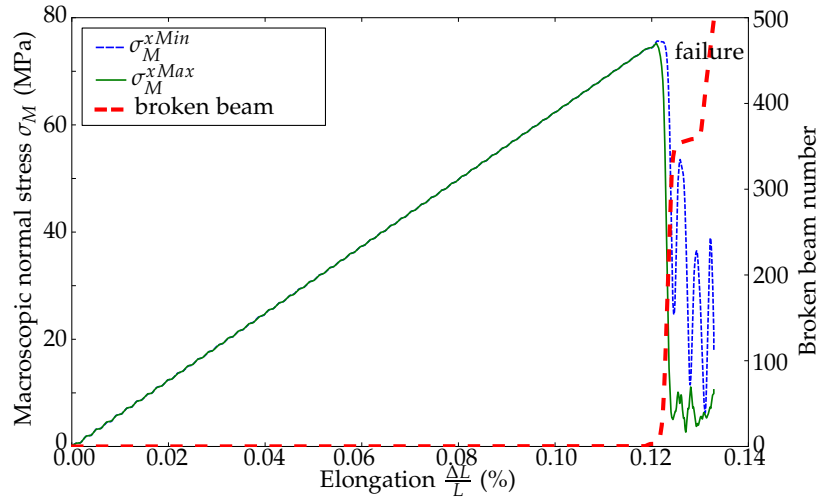


FIGURE 3.25 – Évolutions du nombre de liaisons rompues et des contraintes normales mesurées aux extrémités de la poutre

de l'onde de choc mécanique générée par la rupture : l'énergie élastique stockée dans les liaisons cohésives *poutres* est alors libérée en énergie cinétique. La fissure, ne s'étant pas initiée au centre de l'éprouvette, engendre ensuite les dissymétries observées, des contraintes σ_M^{xMin} et σ_M^{xMax} , après rupture ;

2. une décroissance brutale des contraintes normales après fissuration. Ce changement permet de confirmer la présence d'une rupture.

La contrainte à rupture macroscopique est déduite de la contrainte normale maximale selon le critère de *Rankine* (voir section 3.6.1) adapté aux matériaux fragiles :

$$\sigma_{M_f} = \sigma_M^{max} \quad (3.27)$$

Où σ_M^{max} est la contrainte normale mesurée lors de la rupture de l'échantillon numérique.

3.6.3 Calibration du paramètre microscopique de rupture σ_{μ_f}

La section précédente a permis d'introduire la méthode de détermination de la contrainte à rupture macroscopique σ_{M_f} au travers d'essais de traction quasi statique. Pour quantifier le critère de rupture microscopique σ_{μ_f} , il est proposé de faire varier celui-ci et d'observer l'évolution correspondante de la contrainte macroscopique σ_{M_f} . Afin de prendre en compte l'effet de la dispersion géométrique, quatre échantillons numériques différents sont testés pour chaque valeur du paramètre microscopique de rupture σ_{μ_f} .

La figure 3.26 montre l'évolution de σ_{M_f} en fonction σ_{μ_f} . Cette évolution est linéaire et permet de quantifier la valeur du paramètre microscopique de rupture pour la silice. La valeur de la contrainte à rupture macroscopique $\sigma_{M_f}^{silica} = 50 \text{ MPa}$ (donnée par le tableau 1.3) correspond à une valeur de $\sigma_{\mu_f}^{silica} = 292 \text{ MPa}$ du critère de rupture microscopique.

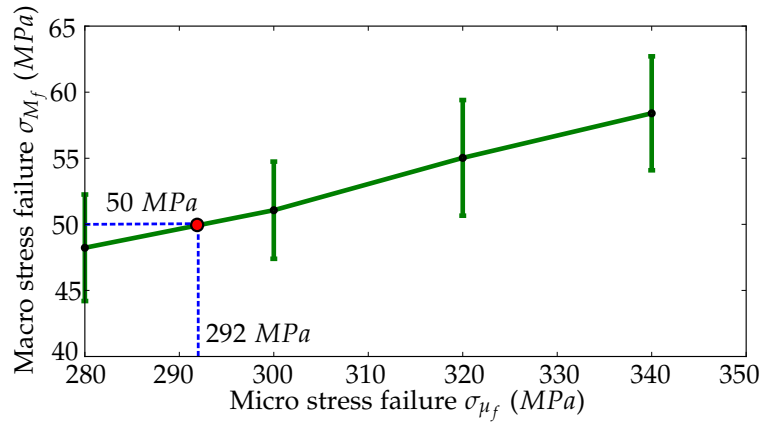


FIGURE 3.26 – Calibration du paramètre microscopique de rupture σ_{μ_f} du matériau silice

3.6.4 Validation

Afin de valider le critère de rupture déterminé à la section précédente, des échantillons numériques de silice sont soumis à des essais de flexion et de torsion. Ces essais sont également pilotés en déplacement suivant le même protocole que pour l'essai de traction à rupture :

1. Des déplacements progressifs sont imposés aux éléments discrets appartenant aux faces $xMax$ et $xMin$ de façon à reproduire les conditions d'un essai de flexion pure ou de torsion ;
2. Des capteurs numériques, mesurant les efforts et moments extérieurs appliqués aux éléments discrets des faces $xMax$ et $xMin$, permettent de déduire les moments de flexion m_{f_M} ou de torsion m_{t_M} macroscopique ;
3. Les lois de la résistance des matériaux sont appliquées de façon à déterminer la contrainte maximale macroscopique maximale σ_{M_f} .

Les détails de mise en œuvre de ces essais sont donnés par les annexes F et G. Le tableau 3.7 synthétise les valeurs des contraintes macroscopiques à rupture pour des essais de traction, flexion et torsion, pour quatre éprouvettes formées d'environ dix mille éléments discrets. Les valeurs des contraintes à rupture macroscopique obtenues pour les essais de traction sont prises comme références. Les résultats montrent une très bonne corrélation entre les résultats des essais de traction et de torsion (0,3 %) et une corrélation acceptable entre les résultats des essais de flexion et de traction (6,5 %). De plus, les géométries de fissuration observées sont caractéristiques des ruptures fragiles :

- la figure 3.27a montre une géométrie de fissure dans un plan perpendiculaire à la ligne moyenne de la poutre macroscopique. En outre, la fissure s'initie sur la fibre extérieure, en extension, de la poutre macroscopique ;
- la figure 3.27b montre une géométrie de fissure orientée à 45° par rapport à la ligne moyenne macroscopique.

Ces éléments permettent de valider le critère de rupture d'un point de vue quantitatif et qualitatif. Les valeurs de contrainte à rupture macroscopique pour les essais de traction, flexion et torsion sont dans une gamme de dispersion ac-

Essais	Contraintes à rupture macroscopique		
	Traction	Flexion	Torsion
1	53,6 MPa	55,8 Mpa	49,7 MPa
2	48,6 Mpa	55,6 Mpa	52,1 Mpa
3	53,2 Mpa	52,9 Mpa	53,2 Mpa
4	51,8 Mpa	56,4 Mpa	51,3 Mpa
Moyennes	51,8 Mpa	55,2 Mpa	51,6 Mpa
Différences	<i>référence</i>	6,5 %	0,3 %

TABLE 3.7 – Comparaison des résultats de contrainte macroscopique à rupture pour des essais de traction, flexion et torsion

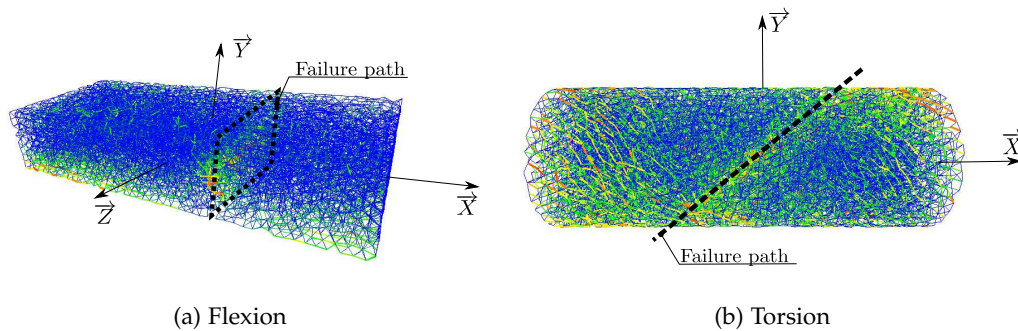


FIGURE 3.27 – Captures d'image de géométrie de fissuration

Le code couleur correspond à la contrainte principale maximale $\sigma_{\mu_{max}}$ des liaisons cohésives poutres

ceptable et les géométries de fissuration observées sont cohérentes par rapport à la théorie de la rupture fragile.

3.6.5 Étude de convergence

La figure 3.28 montre l'évolution de la contrainte à rupture macroscopique σ_{M_f} déduite d'essais de traction à rupture en fonction du nombre d'éléments discrets composant l'éprouvette (voir figures 3.19). Les résultats montrent que les valeurs de σ_{M_f} sont comprises autour d'une valeur moyenne à 10 % près.

Cette dispersion est supérieure à celle constatée pour les paramètres élastiques (voir section 3.4.5). En effet, contrairement aux phénomènes élastiques, la rupture est un phénomène local. Les fissures s'initient sur des défauts locaux dans une zone proche des contraintes maximales théoriques. La contrainte macroscopique à rupture résultante est alors soumise à la théorie du *maillon faible* où la valeur macroscopique de contrainte à rupture est conditionnée par la zone la plus fragile sollicitée.

La contrainte macroscopique à rupture est alors plus sensible aux défauts locaux et donne des valeurs plus dispersées que pour les caractéristiques élastiques macroscopiques. Toutefois, la valeur de 10 % sera considérée comme acceptable d'autant plus, que cette dernière valeur décroît lorsque le nombre d'éléments discrets augmente. Aussi, comme pour les propriétés élastiques, la valeur de la contrainte à rupture microscopique σ_{μ_f} déterminée à l'issue de la phase de calibration sera valable pour n'importe quel domaine discret, à condition que le

nombre d'éléments discrets soit suffisant et que le domaine soit compacté de façon à respecter les critères établis dans la section 3.3.

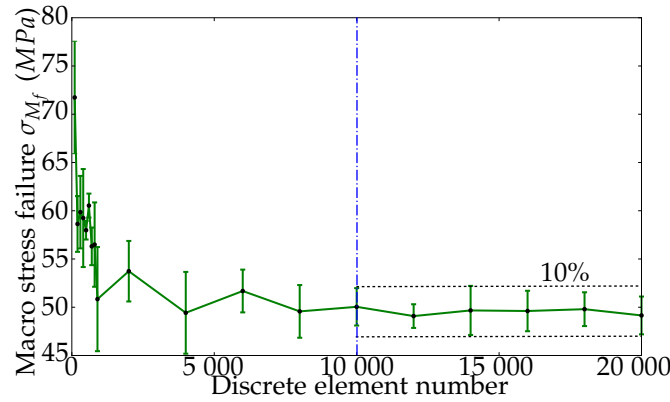


FIGURE 3.28 – Évolution de la contrainte à rupture macroscopique σ_{M_f} en fonction du nombre d'éléments discrets

3.6.6 Synthèse

Les sections 3.1 et 3.2 ont présenté le modèle hybride « lattice - particulaire » ainsi que le modèle de liaison cohésive poutre. Ce modèle permet grâce à l'introduction de rigidité de flexion au niveau microscopique d'obtenir un effet de striction au niveau macroscopique. C'est un avantage considérable de ce modèle.

La section 3.3 a permis de définir des critères de validation du domaine compact initial. Une méthode de mesure originale de l'anisotropie géométrique a été introduite. Puis un critère de finesse de discrétisation a été présenté afin d'assurer une stabilité des propriétés géométriques du domaine.

Les sections 3.4, 3.5 et 3.6 ont introduit des méthodes de calibration des paramètres microscopiques élastiques, massiques et de ruptures. Chacune de ces étapes a ensuite été validée par un jeu d'essais différents de ceux utilisés pour la calibration.

Des études de convergence ont également été menées. Elles ont montré que les relations liant les paramètres microscopiques aux comportements macroscopiques sont indépendantes de l'assemblage final à condition que le nombre d'éléments discrets soit suffisant et que le domaine soit compacté de façon à respecter les critères établis dans la section 3.3.

Des études préliminaires ont montré que cette indépendance n'est pas observée sur des domaines ne respectant pas les critères de compaction. Pour cela une gamme d'éprouvettes discrètes a été construite à l'aide d'un procédé de compaction partiel. Au cours de ce procédé, la compaction est arrêtée lorsque le nombre de coordination N_c du domaine attend la valeur de 5 au lieu de la valeur « standard » de 6,2. La figure 3.29 montre des études de convergence des paramètres élastiques et de rupture menées sur ces domaines discrets partiellement compactés. Il est alors observé que la précision des résultats est grandement affectée, et qu'il n'est plus possible de qualifier les paramètres macroscopiques de convergents. Aussi, l'indépendance entre paramètres microscopiques et comportements macroscopiques ne peut être imputée aux liaisons cohésives *poutres* seules, mais

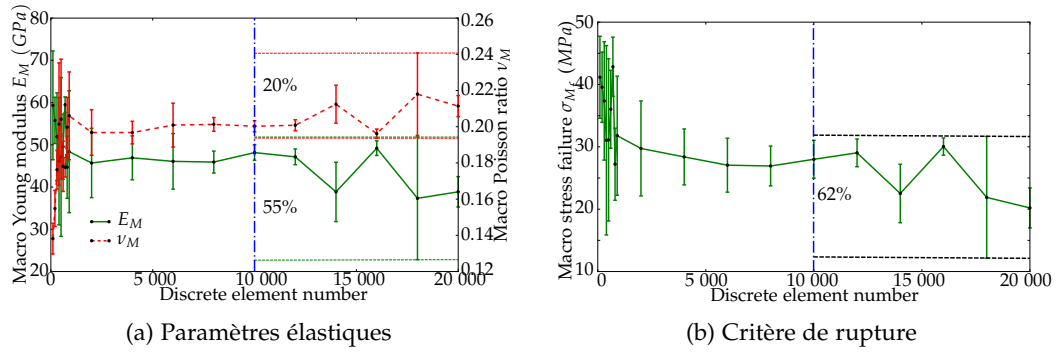


FIGURE 3.29 – Études de convergence pour des domaines discrets « mal compactés » caractérisés par un nombre de coordination de $N_c = 5$

résulte de l'ensemble du processus amont : création et validation du domaine compact initial.

Ce dernier point peut être considéré comme le résultat le plus important de ce chapitre. Les différentes méthodes évoquées dans la littérature ne permettent pas d'établir de façon aussi catégorique cette indépendance. Ces résultats permettent alors d'élargir le champ des perspectives d'utilisation de méthodes DEM quantitatives en permettant, par exemple, l'emploi de domaines discrets de forme complexe.

Enfin, ce chapitre a permis d'établir un modèle DEM de silice pouvant être supposé comme quantitativement *prédictif*. Cette hypothèse est confortée par les différents essais de validation détaillés au cours de ce chapitre. Au cours du chapitre suivant, ce modèle numérique de silice DEM sera exploité dans le cadre d'expériences numériques de surfacage.

CHAPITRE 4

Simulation du doucissage

*« La probabilité de réussir la mise
sur orbite d'une fusée est d'une
chance sur un million.
Dépêchons-nous de rater 999.999
lancements ! »*

Jacques Rouxel, *Les Shadoks*

Résumé

Il est proposé dans cette partie, d'exploiter les résultats obtenus au cours des précédents chapitres afin de simuler une des opérations de surfaçage de la silice : le doucissage. Le modèle développé est inspiré de modèles DEM dédiés à l'étude de phénomènes tribologiques. Ce travail a permis de jeter les bases d'un modèle DEM de doucissage ainsi que d'outils de post-traitement adaptés à cette gamme de problèmes. Il est ainsi possible de qualifier la rugosité de la surface érodée ainsi que de déterminer pour chaque expérience un profil de répartition des dommages sous-surfaciques dans le substrat de silice. Les premiers résultats prouvent la capacité de l'outil à fournir une analyse et une compréhension fine des effets du procédé sur la surface et la sous-surface obtenues. Toutefois, pour remplir pleinement l'objectif de simulations quantitatives du doucissage, des améliorations restent à apporter, notamment, dans la description de l'interface outil-matière.

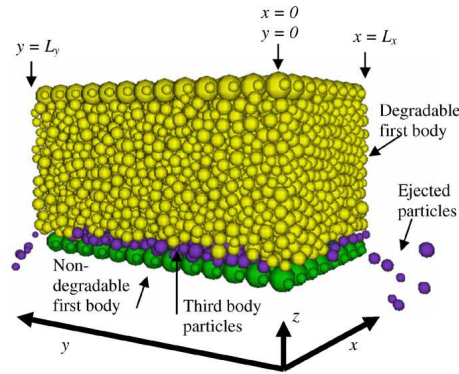


FIGURE 4.1 – Exemple de modèle DEM d'étude de problème tribologique
Image extraite [34]

Le chapitre 3 a permis d'établir et de valider un modèle DEM quantitatif de silice élastique fragile. Ce chapitre est maintenant dédié à l'exploitation de ce modèle au travers de simulations du surfacage. Il sera exclu de cette étude les phases d'ébauchage et de polissage pour ne s'intéresser qu'au doucissage (voir description des opérations de surfacage : chapitre 1, section 1.2). En effet, cette opération particulière est plus difficile à appréhender théoriquement en raison du caractère « libre » des abrasifs. Toutefois, elle constitue une étape clé et dimensionnante en regard des dommages lasers et de ses initiateurs : les SSD (voir chapitre 1, section 1.1.3). Le polissage, bien que mettant en œuvre également des abrasifs libres, ne sera pas étudié, car :

- la cartographie des SSD générée par polissage est, aujourd'hui, encore mal connue. La faible densité de micro-fissures générées par ce procédé induit des difficultés expérimentales de mesure de celles-ci ;
- les phénomènes physico-chimiques semblent être prépondérants lors de cette opération ([40, 21] et [63, §5.4.1]).

Au cours du chapitre 1, il a été montré que le doucissage peut être vu comme un problème tribologique. Aussi, le modèle numérique de doucissage développé au cours de ce chapitre sera inspiré de modèles DEM éprouvés, dédiés à l'étude de phénomènes tribologiques. Par exemple, Fillot, dans la référence [34], étudie qualitativement l'usure des corps frottants en considérant l'écoulement du troisième corps. La figure ?? illustre le modèle DEM utilisé pour cette étude où sont modélisés les deux premiers corps (*degradable first body* et *non-degradable first body*) ainsi que le troisième corps (*third body particles*).

L'une des différences principales de ce dernier modèle par rapport au doucissage est la présence du *slurry* à l'interface de contact. De plus, il est proposé, dans le cadre de cette étude d'étudier quantitativement et d'analyser la sous-surface obtenue dont, notamment, le profil de répartition des SSD provoquées durant le procédé.

En outre, il est nécessaire de procéder à la confrontation des résultats numériques et expérimentaux afin de vérifier la validité du modèle numérique. Les résultats de référence choisis sont ceux issus d'études paramétriques menées par l'équipe du Département des Lasers de Puissance (DLP)-CEA-CESTA [93]. Ces travaux expérimentaux ont pour objectif l'étude de l'influence des différents paramètres de doucissage sur la génération de SSD et la vitesse d'enlèvement de matière. Il est proposé de reproduire numériquement ces essais en respectant

quantitativement les plages de valeurs utilisées pour ces études expérimentales.

Outre les considérations expérimentales et théoriques décrites en introduction, le doucissage se révèle être un bon « candidat » pour une modélisation DEM. Les échelles mises en jeu (dimensions des abrasifs et profondeurs des SSD) permettent de développer un modèle local supposé représentatif du procédé, tout en respectant des temps de calcul raisonnables.

Au cours de ce chapitre, il sera décrit dans un premier temps, le dispositif expérimental de doucissage utilisé comme référence pour le modèle numérique. Dans un second temps, les différentes hypothèses d'études seront énumérées, puis il sera détaillé le modèle numérique de doucissage. Enfin, les résultats des simulations seront discutés en regard des données expérimentales issues du dispositif de référence.

4.1 Dispositif expérimental de référence

Le dispositif utilisé par le DLP du CEA-CESTA repose sur l'utilisation d'une polisseuse-doucisseuse *PM5* distribuée par la firme *Logitech Ltd*. La figure 4.2 décrit le principe global de la machine *PM5*. Le dispositif est composé :

- d'un plateau tournant** en fonte dont la vitesse de rotation (réglable) est imposée par un moteur électrique. La ductilité de la fonte permet aux particules abrasives de « s'enchâsser » dans l'outil. Ce phénomène est recherché au cours du doucissage [81, §2.3].

- d'un Jig** permettant de maintenir l'échantillon de silice par aspiration. Le *Jig* est maintenu en rotation par l'intermédiaire d'un bras de maintien et d'une paire de galets liés au châssis de la *PM5*.

- du slurry** composé de particules abrasives entraînées dans une solution aqueuse. Le *slurry* est délivré par un système de goutte-à-goutte au cours du doucissage.

Le *Jig*, et donc l'échantillon de silice, est entraîné en rotation sous l'action des efforts de frottement entre le plateau, les particules abrasives et l'échantillon de silice. L'échantillon de silice est préalablement poli, sa rugosité de surface est extrêmement faible (rugosité Rms inférieure à un nanomètre), d'un ordre bien inférieur aux dimensions des particules abrasives. Toutefois, au niveau microscopique, les conditions de contact entre particules abrasives et silice sont hétérogènes. Ces conditions sont variables suivant les géométries des abrasifs (voir chapitre 1, figure 1.9), la morphologie locale de la silice, la présence ou non de SSD proche du contact ou bien la vitesse et mouvement relatif entre l'abrasif et la silice. Cela induit une variabilité des efforts de contact *spatiale* (variant localement dans différentes zones de l'interface) et *temporelle* (variant au cours du procédé pour une zone donnée de l'interface).

Les actions de contact, et plus particulièrement leurs composantes tangentielles, sont l'élément moteur de la rotation du *Jig*. Aussi, le mouvement de rotation de celui-ci n'est pas parfaitement constant au cours du doucissage. Ce comportement est une propriété recherchée par les *polisseurs* en permettant de prévenir la formation de rayures et de garantir une usure uniforme de la surface de l'optique.

En outre, la machine *PM5* permet d'ajuster les paramètres du procédé : vitesse de rotation et pression. La vitesse de rotation est réglée *via* un potentiomètre

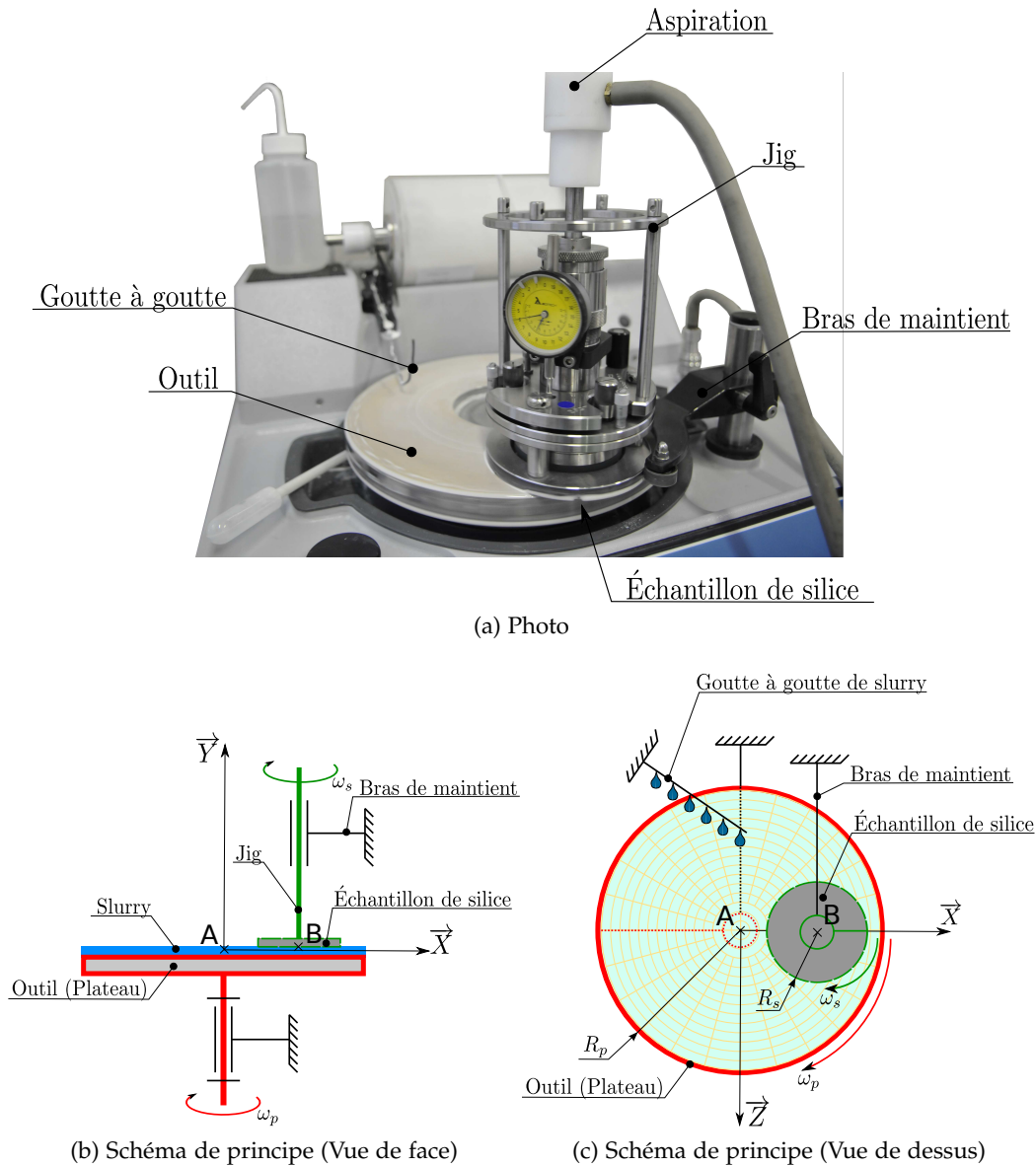


FIGURE 4.2 – Dispositif de polissage et de doucissage PM5

qui permet d'ajuster la vitesse de rotation du moteur électrique entraînant le plateau. La pression est ajustée grâce à un dispositif de précharge interne au *Jig* (voir figure 4.3) : au cours du fonctionnement, le *Jig* est en contact avec l'outil par l'intermédiaire de la couronne extérieure et de l'échantillon de silice. Le poids total du dispositif « *Jig* + silice » est équilibré par les efforts de réaction de ces deux contacts. Le réglage de la pression est effectué par l'intermédiaire du ressort de précharge qui permet d'ajuster la répartition des efforts de réaction entre l'échantillon de silice et la couronne extérieure. L'utilisateur peut régler la précontrainte subie par le ressort de précharge par l'intermédiaire d'une molette et d'un mécanisme vis-écrou. Ainsi, il est possible d'ajuster l'effort vertical exercé sur la silice dans une gamme de valeur allant de zéro au poids total du dispositif « *Jig* + silice ». Cet effort est calibré hors fonctionnement, de façon statique, à l'aide d'une balance.

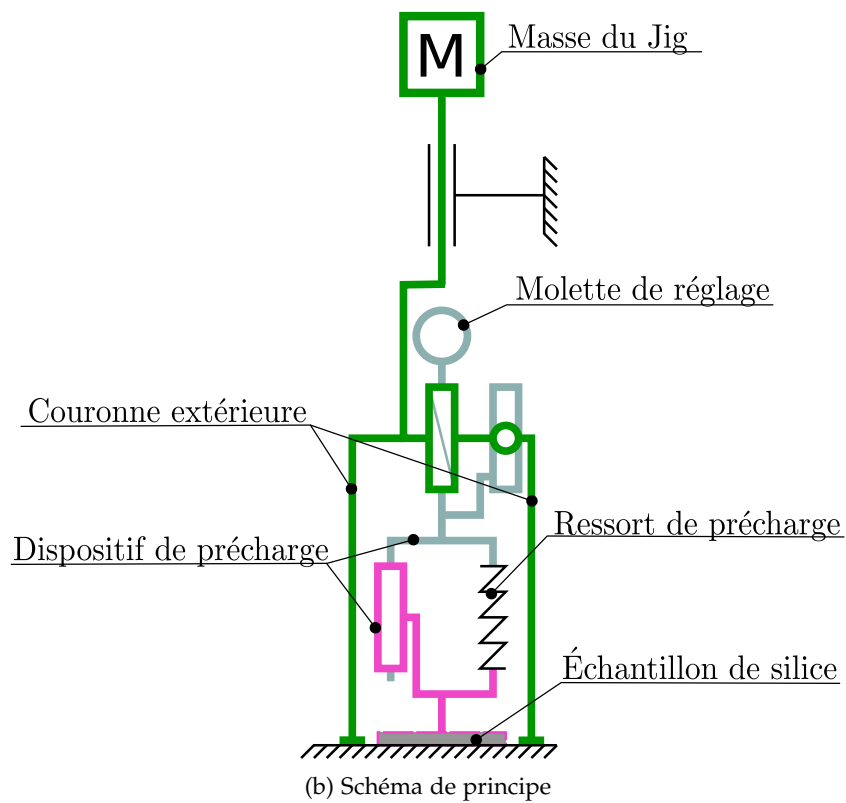
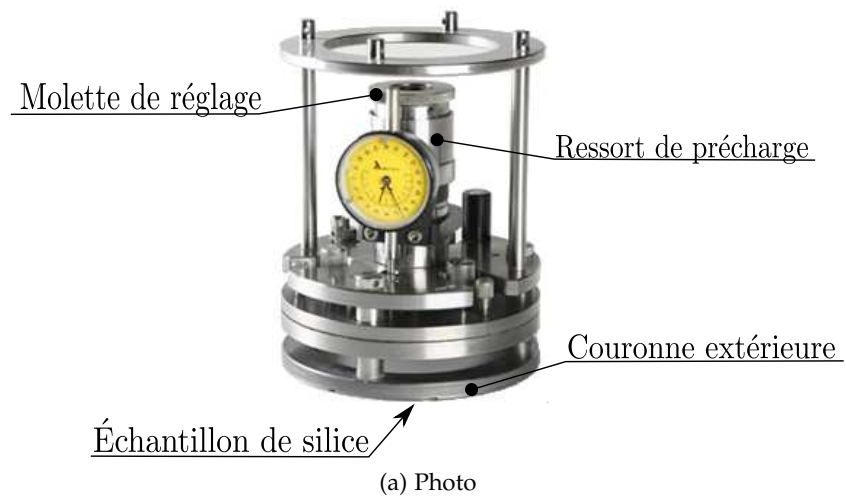


FIGURE 4.3 – Dispositif de précharge Jig

Types	Paramètres	C_{max}
Paramètres machine	Vitesse de rotation ω	↗
	Pression p	↘
Caractéristiques du slurry	Concentration $c\%$	↗
	Dimension moyenne $\tilde{d}$	↘

TABLE 4.1 – Synthèse des influences des paramètres du procédé sur la profondeur maximale de SSD

4.2 Résultats expérimentaux de référence

Les résultats de référence sont issus d'études paramétriques menées par l'équipe du DLP-CEA-CESTA sur la machine PM5 [93]. Ces travaux expérimentaux ont pour objectif l'étude de l'influence des différents paramètres de doucissage sur la génération de SSD et du taux d'enlèvement matière. Toutefois, les simulations actuelles ne permettent pas de rendre compte de façon réaliste du phénomène d'enlèvement matière. En effet, le taux d'enlèvement matière, de l'ordre d'une dizaine de micromètres par minute est, pour l'instant, incompatible avec les temps simulés de l'ordre de quelques millisecondes. Cette limitation ainsi que les solutions envisagées seront discutées en conclusion de ce chapitre.

Dans ces travaux, les SSD sont qualifiées par leur profondeur maximale, notée c_{max} et sont mesurées par *acidage* (voir section 1.4.1). Les paramètres de l'étude portent sur les paramètres du procédé ainsi que sur les caractéristiques du *slurry*, soit :

la pression statique exercée sur l'échantillon de silice. Ce paramètre est calibré hors fonctionnement et quantifié en kilogramme, relativement à la méthode de mesure *par pesée* ;

la vitesse de rotation de l'outil, définie en tour par minute ;

la concentration en abrasif du slurry. Ce paramètre est défini par la fraction volumique du volume occupé par les particules abrasives sur le volume total du *slurry* ;

la dimension moyenne des abrasifs mesurée en micromètre.

Les figures H.1, H.2, H.4, H.3 et H.5, disponibles en annexe, H montrent les évolutions des profondeurs maximales de SSD en fonction de ces quatre paramètres. Les tendances visibles sur ces figures sont synthétisées sur le tableau 4.1. Ces résultats, déjà discutés au cours de la section 1.5, montrent une contradiction avec le modèle d'indentation statique (nommé aussi « modèle classique », voir section 1.2). Un des enjeux des simulations est de reproduire ces tendances en apportant une description plus fine des phénomènes que le modèle « classique ».

4.3 Description du modèle de doucissage DEM

Afin de conserver des temps de calcul raisonnables, une petite fraction de la pièce de silice est simulée (voir figure 4.4). Le volume modélisé, contenu dans un cube de quelques centaines de micromètres, se situe dans la zone inférieure de la pièce de silice sollicitée par le processus d'abrasion. En plaçant le modèle d'étude dans cette zone, il est possible de simuler l'interface d'abrasion ainsi qu'un volume représentatif de silice. La hauteur du cube de silice est dimensionnée en

regard de la profondeur de SSD maximale mesurée expérimentalement. Pour les gammes de paramètres étudiées, les profondeurs de SSD ne dépassent pas la cinquantaine de micromètres. Aussi, il est choisi une hauteur de dimension trois fois supérieure à cette valeur, soit $150\ \mu\text{m}$.

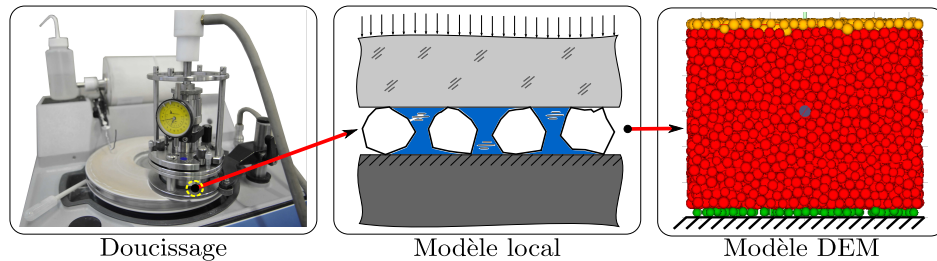


FIGURE 4.4 – Localisation du modèle DEM par rapport au procédé

4.3.1 Domaine d'étude et description générale

La figure 4.5 illustre le modèle DEM du doucissage. L'ensemble du domaine d'étude est contenu dans un parallélépipède. Des conditions périodiques permettent d'étendre virtuellement le domaine suivant les directions $\vec{X}$ et $\vec{Z}$ (voir figures 4.5) et de le considérer comme infini suivant ces directions. Les conditions périodiques sont traitées plus en détail en annexe (voir section J). Le domaine contient différents corps, dont :

- un volume représentatif de silice de forme parallélépipédique (éléments discrets rouges) ;
- les abrasifs, placés à l'interface du bloc de silice et de l'outil (éléments discrets verts) ;
- l'outil, modélisé par un élément discret unique de forme plane.

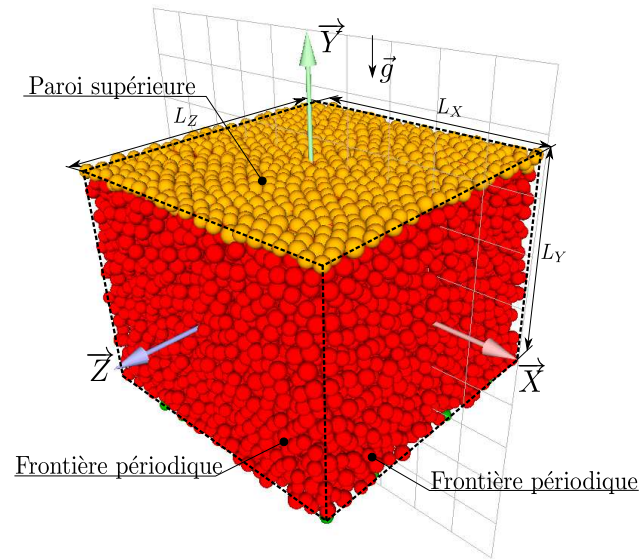
Des conditions limites permettent alors de simuler le procédé d'abrasion, dont :

- la pression, appliquée aux éléments discrets du bloc de silice formant la paroi supérieure (représentés en orange sur la figure 4.5) ;
- des déplacements imposés, dans le plan horizontal ($\vec{X}, \vec{Z}$), aux particules abrasives.

Il est proposé, dans les sections suivantes, de décrire plus en détail le modèle DEM. Une première partie sera consacrée à la modélisation des différents corps en présence. Une seconde partie sera dédiée à la description des modèles d'interaction. Une troisième partie détaillera les conditions limites et chargements extérieurs. Enfin, une dernière partie décrira la méthodologie de passage des conditions macroscopiques (doucissage expérimental) aux conditions microscopiques (modèle DEM).

4.3.2 Modélisation des différents corps

Le doucissage est un processus faisant interagir entre eux différents corps solides et liquides : l'outil, la solution aqueuse, les abrasifs, les particules d'usure et la silice (illustrés sur la figure 4.6).



(a) Vue de perspective

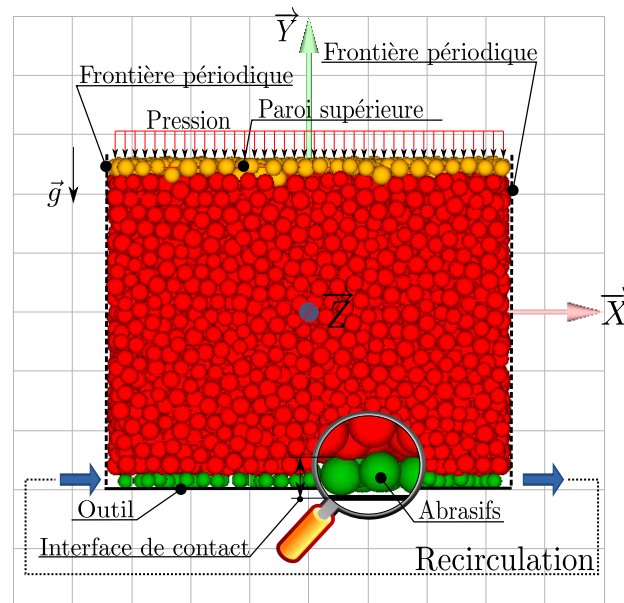
(b) Vue de face dans le plan $(\vec{X}, \vec{Y})$

FIGURE 4.5 – Illustration du modèle de doucissage DEM

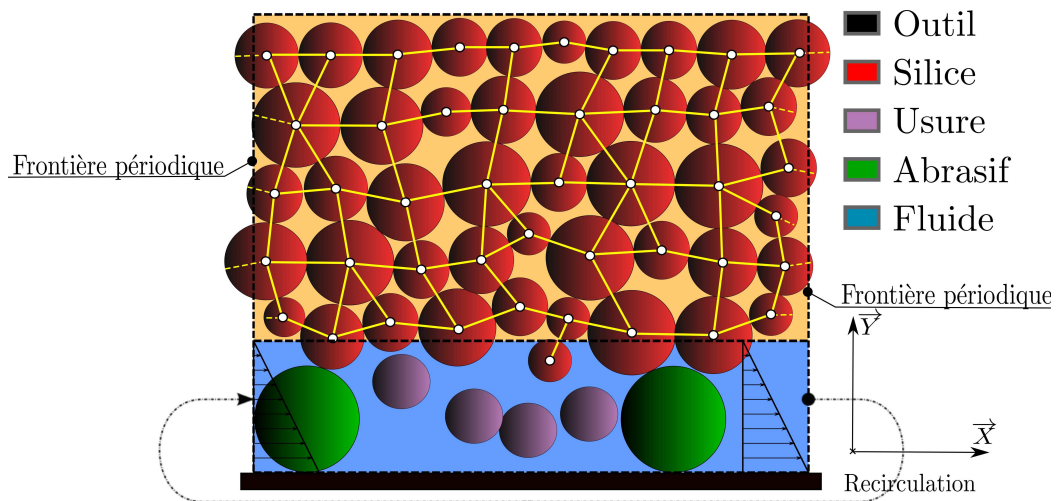


FIGURE 4.6 – Illustration 2D des différents corps du modèle DEM du doucissage

D'un point de vue informatique, la distinction entre les différents corps mis en jeu dans la simulation est réalisée grâce à l'utilisation des conteneurs de la *libDEM* : les *SetOf* (voir section 2.4.2). Ainsi, chacun des éléments discrets est affecté à un conteneur spécifique. Les relations entre conteneur et contenu étant bidirectionnelles, les objets *DiscreteElement* maintiennent à jour une liste de conteneurs dans lesquels ils sont référencés. Ce mécanisme permet de discriminer les éléments discrets et de leur affecter des comportements spécifiques.

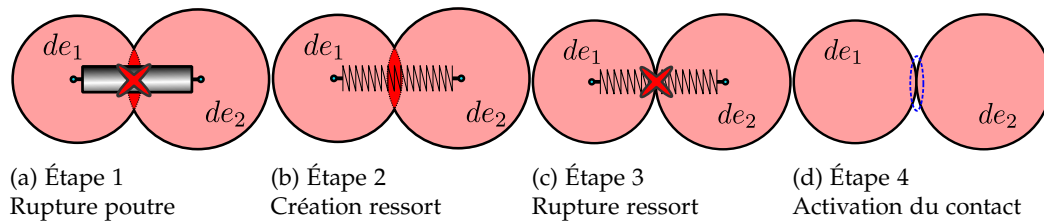
Un travail de « classification », dont le but est de référencer chacun des éléments discrets dans le conteneur adéquat, est réalisé durant une phase préliminaire au calcul DEM. Certains de ces conteneurs, comme le conteneur des particules d'usure, doivent être mis à jour au cours de la simulation.

La silice

La silice est composée d'un agrégat d'éléments discrets en interaction par des liaisons cohésives *poutres*. Le chapitre précédent a permis d'établir un modèle quantitatif de silice élastique fragile. Or, les essais de calibration et de validation en rupture, mis en œuvre au cours de ce chapitre (torsion, traction et flexion), sont de nature à solliciter exclusivement l'échantillon qu'en mode « d'ouverture de fissure ». Au cours du doucissage, il est fort probable que les fissures, créées dans la couche endommagée de la silice, soient sollicitées de façon cyclique en mode d'ouverture puis de fermeture de fissure. Aussi, il est nécessaire de mettre en application des mécanismes permettant de prendre en compte ce phénomène.

Pour ce faire, il est proposé d'activer la gestion des contacts uniquement lorsqu'une liaison cohésive, reliant deux éléments discrets *silice*, est rompue. Toutefois, il est possible que lors de la rupture d'une liaison cohésive *poutre*, les éléments discrets soient en interférence géométrique. Ce cas de figure pourrait alors engendrer des efforts de répulsion surévalués et rendre le système numériquement instable. Pour prévenir ce phénomène, la rupture est dite *progressive*. Elle est réalisée en quatre temps (voir figures 4.7) :

1. Rupture de la liaison cohésive *poutre* suivant les critères établis dans le chapitre précédent;

FIGURE 4.7 – Illustration du modèle de rupture *progressive*

2. Remplacement de la liaison cohésive *poutre* par une liaison cohésive de type *ressort de répulsion*. Ce type de liaison est unilatéral et peut être assimilé à un contact. La raideur est alors déterminée suivant la méthode présentée dans la section suivante (section 4.3.3). Il est imposé que la liaison possède une « longueur à vide » de façon à ce que les efforts de répulsion soient nuls lors de la création de la liaison. Cette liaison cohésive permet de ne pas générer de discontinuité des actions de répulsion si les deux éléments discrets sont initialement en interférence géométrique ;
3. Destruction de la liaison cohésive *ressort de répulsion* si les deux éléments discrets ne sont plus en interférence géométrique ;
4. Activation de la gestion classique des contacts entre les deux éléments discrets.

Une autre caractéristique du domaine discret *silice* est l'emploi de frontières périodiques suivant les directions $\vec{Z}$ et $\vec{X}$ (voir figure 4.5). Les frontières périodiques permettent d'étendre virtuellement le domaine d'étude à l'infini. Le domaine peut être considéré comme un parallélépipède d'épaisseur finie suivant l'axe $\vec{Y}$ et de longueur transversale, suivant les axes $\vec{Z}$ et $\vec{X}$, infinie. L'annexe J présente une description détaillée des traitements et algorithmes liés à l'utilisation des conditions périodiques.

Les particules d'usure

Au cours de la simulation, de l'endommagement apparaît dans le volume de silice. Certains éléments discrets peuvent alors se détacher de l'agrégat de silice. Ces éléments discrets, n'ayant plus de liaison les reliant à l'agrégat de silice, peuvent être qualifiés de « solitaires » et sont considérés comme des *particules d'usure*.

Dans le procédé réel, au cours du doucissage, ces particules sont évacuées hors de l'interface d'abrasion sous l'action du fluide par un réseau de rainures présent sur l'outil. Aussi, pour rendre compte de ce phénomène, les éléments discrets d'usure sont détruits lorsque ceux-ci franchissent les limites géométriques du domaine d'étude. Une limite de ce modèle est que les particules d'usure initialement proches des frontières du domaine sont évacuées plus vite que celles situées au centre.

La création des particules d'usure ne pouvant être prédite au cours de la simulation, il est alors nécessaire de mettre à jour régulièrement les conteneurs dédiés aux éléments discrets *silices* et aux éléments discrets *d'usure*. Pour ce faire, un algorithme de *reconnaissance d'agrégat* est utilisé. Un *agrégat* représente un ensemble d'éléments discrets reliés entre eux par des liaisons cohésives. Un *agrégat*

peut être composé de plusieurs ou d'un seul élément discret. L'algorithme de *reconnaissance d'agrégat* permet de construire « à la volée » la liste des agrégats présents dans la simulation. L'agrégat composé du nombre d'éléments discrets le plus élevé est considéré comme étant le bloc de silice. Les autres agrégats sont alors considérés comme des éléments discrets d'usure. L'algorithme est présenté en détail en annexe (section I).

Les abrasifs

La modélisation des particules abrasives est un point délicat. En effet, les images de microscopie (voir figure 1.9 du chapitre 1) montrent que chacune des particules abrasives possède une morphologie complexe et unique. Or les mesures expérimentales de taille des abrasifs, réalisées par granulométrie laser, donnent un diamètre correspondant à la diffusion de la lumière par une sphère. Afin d'être cohérent avec cette dernière hypothèse, il est supposé, dans une première approche, qu'une particule abrasive peut être assimilée à une sphère. Chaque abrasif sera représenté alors par un élément discret distinct de forme sphérique. Des approches plus fines consistent modéliser les abrasifs par des agglomérats d'éléments discrets dont les dimensions sont représentatives de l'angularité réelle des abrasifs ou par des géométries polyédriques. Toutefois, ces modèles ne seront pas développés dans le cadre de cette thèse en raison des coûts de calcul inhérents à ces méthodes.

Contrairement aux particules d'usure, les particules abrasives, lorsqu'elles franchissent les limites géométriques du domaine sont réintroduites « de l'autre côté » suivant les conditions de périodicité.

L'outil

L'outil est représenté par un unique élément discret de forme plane. Ce choix de modélisation permet de privilégier les performances de calcul pour une zone du modèle où une description fine n'est pas requise. L'outil permet de mettre en mouvement les particules abrasives dans le plan $(\vec{X}, \vec{Z})$ et de générer des efforts verticaux de répulsion sur les corps en interférence géométrique avec l'outil. En regard de la zone modélisée, le mouvement relatif de l'outil par rapport à la pièce de silice est approché par une translation suivant la direction $\vec{X}$.

La solution aqueuse

La solution aqueuse est située à l'interface d'abrasion. Il est supposé que le volume occupé par la solution aqueuse est contenu dans un parallélépipède contigu avec l'outil (voir figure 4.8). Les dimensions transversales sont celles du domaine. Par contre, l'épaisseur du film, notée e , est calculée à la volée durant la simulation.

Au cours de la simulation, un algorithme de reconnaissance des éléments discrets formant la limite inférieure du bloc de silice est utilisé. Cet algorithme (décrit en annexe, voir section K) permet de maintenir à jour un conteneur particulier, nommé *AbradedSurface* (voir figure 4.8), de façon à qualifier la couche abrasée de silice. L'épaisseur e du film de fluide est alors calculée en prenant la moyenne des positions projetées sur l'axe $\vec{Y}$ des éléments discrets de silice appartenant à la couche abrasée.

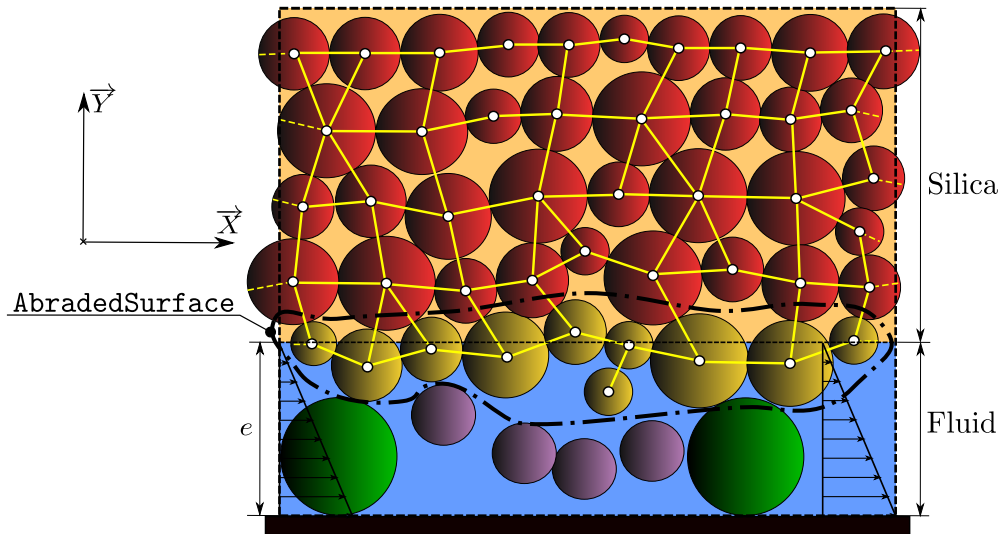


FIGURE 4.8 – Domaine de la solution aqueuse

4.3.3 Modélisation des interactions

Il ne sera pas fait mention, au cours de cette section, des interactions de type liaisons cohésives entre éléments discrets de *silice*, celles-ci faisant le sujet de descriptions détaillées au cours du chapitre précédent (chapitre 3).

Hormis le cas particulier des liaisons cohésives, les éléments discrets sont en interactions par contact. L'action de la solution aqueuse sur les éléments discrets en suspension sera également qualifiée de *contact*. Aussi, trois catégories de contact sont distinguées :

1. les contacts entre deux éléments discrets appartenant à un même matériau supposé comme initialement continu. Dans ce cas, le contact est dit *continu*. C'est le cas, par exemple, entre deux éléments discrets de *silice* non reliés par une liaison cohésive ;
2. les contacts entre deux éléments discrets représentant deux corps distincts. Dans ce cas, le modèle doit tenir compte de la géométrie sphérique des éléments discrets. Le modèle de contact choisi est alors celui de *Hertz* ;
3. les contacts entre le fluide et les éléments discrets en suspension dans ce milieu. Dans ce cas, le contact sera considéré comme *visqueux*.

Le tableau 4.2 synthétise les différents modèles d'interaction en fonction des corps en contacts. Les différents types d'éléments discrets sont décrits au cours de la section précédente (voir section 4.3.2).

Le contact continu

Un contact entre deux éléments discrets *silice* ne peut intervenir qu'après la rupture d'une liaison cohésive. La rupture d'une liaison cohésive est considérée alors comme une fissure au sein du matériau. Le but du contact *continu* est alors de prévenir l'interpénétration de deux éléments discrets *silice* en générant des efforts de répulsion lors d'une phase de refermeture de fissure.

Le travail présenté au chapitre 3, dont le modèle de liaison cohésive *poutre*, permet d'introduire d'un point de vue macroscopique, un comportement proche

	ED Silice	ED Usure	ED Abrasif	ED Outil	Fluide
ED Silice	<i>Continu</i>	-	-	-	-
ED Usure	<i>Hertz (S/S)</i>	<i>Hertz (S/S)</i>	-	-	-
ED Abrasif	<i>Hertz (S/S)</i>	<i>Hertz (S/S)</i>	<i>Hertz (S/S)</i>	-	-
ED Outil	<i>Hertz (P/S)</i>	<i>Hertz (P/S)</i>	<i>Hertz (P/S)</i>	<i>Imp</i>	-
Fluide	<i>Visqueux</i>	<i>Visqueux</i>	<i>Visqueux</i>	<i>S.O</i>	<i>S.O</i>

S : Sphère P : Plan Imp : Impossible S.O : Sans objet

TABLE 4.2 – Les différents modèles de contact suivant les différents types d'éléments discrets

d'un matériau continu. L'une des méthodes employées consiste à combler virtuellement les vides du domaine discret.

Dans le cas présent, une méthode similaire est employée. La raideur du contact *continu* est calculée de façon à compenser les vides générés par la forme sphérique des éléments discrets. Une forme virtuelle cubique est associée aux éléments discrets en contact. Cette méthode est également employée dans la référence [54]. La loi de répulsion est supposée parfaitement élastique. Les frottements sont négligés. L'effort de répulsion $\vec{f}$ est calculé suivant la loi :

$$\vec{f} = \delta K \vec{n} \quad \text{avec} \quad \delta = R_1 + R_2 - \left\| \overrightarrow{O_1 O_2} \right\| \quad \text{et} \quad \vec{n} = \frac{\overrightarrow{O_1 O_2}}{\left\| \overrightarrow{O_1 O_2} \right\|} \quad (4.1)$$

Où :

- $\vec{f}$ est l'effort de répulsion ;
- δ est l'interpénétration de contact ;
- K est la raideur de contact ;
- $\vec{n}$ est le vecteur unitaire suivant la direction normale au contact ;
- O_1 et O_2 sont les centres des éléments discrets en contact ;
- R_1 et R_2 sont les rayons des éléments discrets en contact.

Afin de déterminer la raideur K , il est associé au contact, un volume et une surface utile de forme parallélépipédique. Pour ce faire, des parallélépipèdes $\mathcal{C}_1$ et $\mathcal{C}_2$ (voir figure 4.9) sont associés aux éléments discrets de_1 et de_2 en contact. Les parallélépipèdes $\mathcal{C}_n$ ont pour longueur, le rayon R_n de l'élément discret n , et sont de section carrée, de coté $2R_n$.

Il est affecté, pour chacun des éléments discrets de la simulation, des propriétés matériaux spécifiques. Ces propriétés matériaux, dont le module de Young E_n , sont mises ensuite à profit pour déterminer la raideur équivalente K d'un contact entre deux éléments discrets. Ainsi, la raideur longitudinale K_n pour le parallélépipède $\mathcal{C}_n$ est déduite des lois de la résistance des matériaux par :

$$K_n = \frac{E_n S_n}{L_n} = \frac{E_n 4R_n^2}{R_n} = 4E_n R_n \quad (4.2)$$

Où :

- S_n et L_n sont respectivement la section et la longueur du parallélépipède $\mathcal{C}_n$;
- E_n est le module de Young **macroscopique** de l'élément discret n ;
- R_n est le rayon de l'élément discret n .

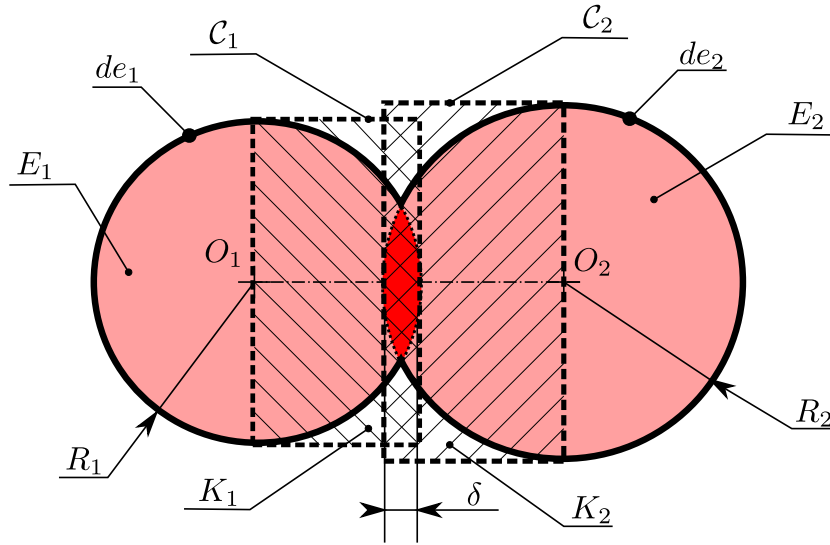


FIGURE 4.9 – Illustration de la méthode de calcul de la raideur équivalente de contact K

La raideur de contact K est alors donnée par la raideur des deux parallélépipèdes C_1 et C_2 pouvant être assimilés à deux ressorts en série. Ainsi :

$$K_n = \frac{K_1 K_2}{K_1 + K_2} \quad (4.3)$$

Or dans le cas d'un contact entre deux éléments discrets de silice les modules de Young E_1 et E_2 sont égaux entre eux. Donc :

$$K_n = 4E \frac{R_1 R_2}{R_1 + R_2} \text{ avec } E_1 = E_2 = E \quad (4.4)$$

Le contact de Hertz

Les particules d'usure, les abrasifs ainsi que l'outil sont modélisés par des éléments discrets dont les géométries, contrairement aux éléments discrets *silice*, sont supposées représentatives des morphologies réelles des corps en présence. Au cours de la simulation du doucissage, deux cas peuvent intervenir :

1. un contact entre deux sphères (voir figure 4.10a) ;
2. un contact entre un plan et une sphère. Ce deuxième cas traduit le contact possible entre un élément discret et l'outil de forme plane (voir figure 4.10b).

La relation donnant l'effort de répulsion d'un contact de Hertz est :

$$\vec{f} = \frac{4}{3} E^* \delta \sqrt{R^* \delta} \vec{n} \quad (4.5)$$

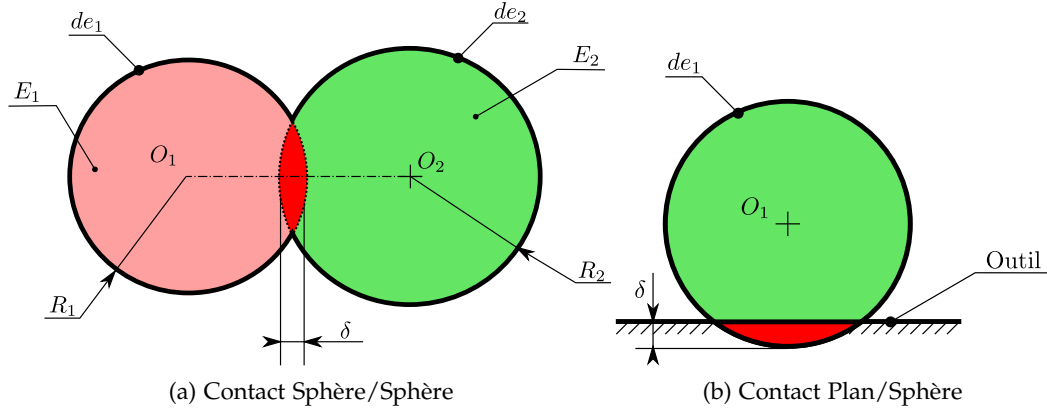


FIGURE 4.10 – Illustration des contacts de Hertz

Avec :

$$E^* = \frac{E_1(1 - \nu_2^2) + E_2(1 - \nu_1^2)}{E_1 E_2} \quad (4.6)$$

$$R^* = \begin{cases} \frac{R_1 R_2}{R_1 + R_2} & \text{Cas du contact Sphère/Sphère} \\ R_1 & \text{Cas du contact Plan/Sphère} \end{cases} \quad (4.7)$$

Où :

- E_1 et E_2 sont les modules de Young des matériaux en contact ;
- ν_1 et ν_2 sont les coefficients de Poisson des matériaux en contact ;
- R_1 et R_2 sont les rayons des géométries en contact ;
- δ est l'interpénétration de contact ;
- $\vec{n}$ est le vecteur unitaire normal au contact.

Amortissement

Dans la réalité, les matériaux possèdent des propriétés dissipatives lorsqu'ils sont sollicités mécaniquement. Aussi, de l'amortissement est introduit au niveau de chaque interaction selon :

$$\vec{f}_a = \dot{\delta} \alpha \, 2\sqrt{K^* M} \, \vec{n} \quad (4.8)$$

Où :

- α est le coefficient d'amortissement du contact. Le coefficient $2\sqrt{K^* M}$ représente l'amortissement critique de l'interaction. Le coefficient α permet d'exprimer l'amortissement sous la forme d'un ratio de cet amortissement critique. Il est également possible d'utiliser le coefficient de restitution e suivant la formulation donnée dans [56, §2.3] ;
- $\dot{\delta}$ est la vitesse de pénétration ;
- K^* est la raideur généralisée de l'interaction ;
- M est la masse équivalente de l'interaction ;
- $\vec{n}$ est la direction de l'interaction.

Cette formulation générale permet d'introduire de la dissipation d'énergie au niveau de toutes les interactions : contact de Hertz, contact continu et liaison cohésive *poutre*. Le tableau 4.3 résume l'intégration de l'amortissement ainsi que

les calculs des raideurs généralisées pour les différents modèles d'interaction. L'amortissement, dans le cas des liaisons cohésives *poutres*, n'est pas calculé pour toutes les directions de l'espace afin de réduire les temps de calcul. Seul, l'amortissement pour les modes de sollicitation prépondérants (traction et flexion) est pris en compte.

Ne possédant pas de données précises sur l'amortissement des matériaux mis en jeu, le coefficient α est fixé arbitrairement à une valeur de 10^{-2} pour les liaisons cohésives *poutres* et de 10^{-1} pour les autres interactions. Ces valeurs correspondent à des amortissements plutôt faibles, mais suffisants pour minimiser les conséquences de forts gradients de chargement induits par des ruptures ou des collisions, pouvant entraîner de fortes oscillations et même l'instabilité du système numérique.

	Contact Hertz		Contact continu	Liaison coh. poutre	
	Sphère/Sphère	Plan/Sphère		Traction	Flexion
K^*	$\frac{4}{3}E^*\sqrt{R^*\delta}$	$\frac{4}{3}E^*\sqrt{R^*\delta}$	$4E\frac{R_1R_2}{R_1+R_2}$	$\frac{ES}{l}$	$\frac{2EI_g}{l}$
M	$\frac{M_1M_2}{M_1+M_2}$	M	$\frac{M_1M_2}{M_1+M_2}$	$\frac{M_1M_2}{M_1+M_2}$	$\frac{I_1I_2}{I_1+I_2}$
α	10^{-1}	10^{-1}	10^{-1}	10^{-2}	10^{-2}

Où :

- E^* est le module de Young équivalent du contact de Hertz (voir égalité 4.6) ;
- R^* est le rayon équivalent du contact de Hertz (voir égalité 4.7) ;
- M_n est la masse de l'élément discret n ;
- M est la masse de l'élément discret sphérique dans le cas du contact Plan/-Sphère ;
- R_n est le rayon de l'élément discret n ;
- E est le module de Young de l'interaction (cas où $E_1 = E_2$) ;
- S est la section de la liaison cohésive *poutre* ;
- l est la longueur de la liaison cohésive *poutre* ;
- I_g est le moment quadratique de la liaison cohésive *poutre* ;
- I_n est l'inertie de l'élément discret n .

TABLE 4.3 – Intégration de l'amortissement dans les différents modèles d'interaction

Interaction avec le fluide, contact visqueux

La figure 4.11 illustre le modèle du fluide. Le fluide est supposé se mouvoir entre deux plans parallèles de distance e_f . Le premier plan $\mathcal{P}_t$ est représentatif de la position de l'outil et le plan $\mathcal{P}_s$ de la surface abrasée du bloc de silice (voir

section 4.3.2). Le mouvement du fluide est supposé laminaire. L'outil est animé d'un mouvement de translation de vitesse v_t suivant l'axe $\vec{X}$. Le bloc de silice est supposé immobile dans le plan $(\vec{X}, \vec{Z})$. En conséquence, le mouvement du fluide peut être approché par un *écoulement de couette*, caractérisé par :

- un gradient linéaire de vitesse $\vec{v}_f$ dans l'épaisseur du film de fluide ;
- des vitesses égales aux vitesses de déplacement des parois pour les zones du fluide en contact avec celles-ci, soit :
 - une vitesse égale à la vitesse de déplacement de l'outil v_t pour une altitude $Y = 0$;
 - une vitesse nulle pour une altitude $Y = e_f$.

L'action du fluide sur les éléments discrets immergés est alors déduite des forces d'entraînement visqueuses engendrées par le gradient de vitesse du fluide sur la surface de l'élément discret. L'élément discret, supposé sphérique, subira donc des efforts et moments visqueux, égaux à [47, §7.4] :

$$\vec{f} = 8\pi\eta R(\vec{v}_f - \vec{v}_d) \quad (4.9)$$

$$\vec{m} = 8\pi\eta R^3(\vec{\omega}_f - \vec{\omega}_d) \quad (4.10)$$

Où :

- η est la viscosité du fluide ;
- R est le rayon de l'élément discret ;
- $\vec{v}_f$ et $\vec{\omega}_f$ sont la vitesse linéaire et la vitesse de rotation du fluide calculées au centre de l'élément discret ;
- $\vec{v}_d$ et $\vec{\omega}_d$ est la vitesse linéaire et la vitesse de rotation de l'élément discret.

Les vitesses du fluide sont données en fonction de l'altitude y_d de l'élément discret :

$$\vec{v}_f = \|\vec{v}_t\| \left(1 - \frac{y_d}{e_f}\right) \vec{X} \quad (4.11)$$

$$\vec{\omega}_f = \frac{\|\vec{v}_t\|}{e_f} \vec{Z} \quad (4.12)$$

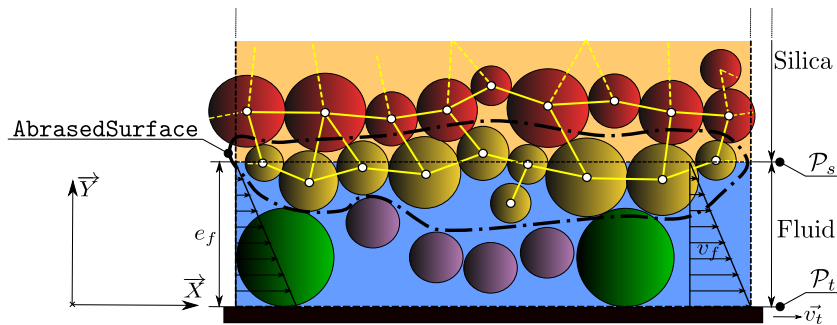


FIGURE 4.11 – Illustration 2D du modèle de fluide

4.3.4 Conditions limites et chargements

Durant le procédé réel, l'ensemble formé de l'échantillon de silice et du *slurry* est sollicité par des actions mécaniques provenant de la vitesse relative entre

l'échantillon de silice et le plateau ainsi qu'une partie du poids du *Jig* (précharge réglable).

Il est proposé de modéliser ces deux sollicitations externes par des conditions limites en déplacement appliquées à l'ensemble fluide et abrasif ainsi que des chargements verticaux appliqués aux éléments discrets formant la paroi supérieure du bloc de silice. Les sections suivantes détaillent la mise en œuvre de ces conditions limites et chargements dans le modèle DEM.

Fenêtre temporelle de simulation

Afin de conserver une stabilité du schéma numérique dynamique explicite, il est nécessaire d'introduire un pas de temps Δt inférieur au plus petit temps d'oscillation (t_{os}) du système, soit :

$$\Delta t = \frac{1}{k} \min(t_{os}) \quad \text{avec} \quad k \geq 2\pi \quad \text{et} \quad t_{os} = \sqrt{\frac{M_i}{K_i}} \quad (4.13)$$

Où M_i et K_i sont les masses et raideurs équivalentes d'une interaction élastique. La valeur de k déterminée par Cambou [17] est évaluée à 2π . Dans ces conditions, le pas de temps de simulation est de l'ordre $1,7 \cdot 10^{-10}$ s. Pour conserver des temps de calcul raisonnables, les calculs seront limités à environ dix millions d'itérations. La fenêtre temporelle de simulation est alors de $1,7$ ms. Ce temps est très court comparativement aux expériences réelles, pouvant durer quelques dizaines de minutes.

Aussi, la fenêtre temporelle de simulation ne permet pas d'étudier l'enlèvement matière, de l'ordre d'une dizaine de micromètres par minute en regard des dimensions moyennes des éléments discrets silices de l'ordre de la dizaine de micromètres. Il est proposé d'accélérer artificiellement ce phénomène en contraignant le déplacement de la paroi supérieure.

Une première solution consiste à considérer la paroi supérieure comme fixe. Cette condition entraîne alors des pics d'effort normal et tangentiel et permet de favoriser l'abrasion. Toutefois, cette méthode contraint le volume du domaine d'étude à être constant, ce qui est en désaccord avec l'hypothèse formulée à la section 4.3.2 où les particules d'usure sont évacuées. Afin de compenser la perte de volume induite par l'évacuation des particules d'usure, il est donné à la paroi supérieure la possibilité de ne se déplacer que « vers le bas » (axe $-\hat{Y}$). Cette condition de déplacement unidirectionnel permet un mouvement de corps rigide du bloc de silice lorsque le système atteint une phase de repos et de venir compenser les pertes de volume sous l'action d'efforts normaux de faibles intensités. Il est proposé que cette charge extérieure soit dimensionnée en fonction de la pression imposée du doucissage expérimental (précharge). Toutefois, dans ces conditions, le taux d'enlèvement matière simulé ne pourra pas être considéré comme représentatif du taux d'enlèvement matière expérimental.

Modélisation de la précharge

Le dispositif de précharge, décrit dans la section 4.1, permet d'ajuster l'effort normal. Il est considéré que l'effort de précharge, calibré hors fonctionnement, est constant spatialement et temporellement au cours du doucissage. Celui-ci est donc supposé intégralement transmis proportionnellement à la surface de silice

modélisée. Comme illustré sur la figure 4.12, l'effort global de précharge $\vec{F}_{pr}$ est appliqué aux éléments discrets appartenant à la paroi supérieure du bloc de silice DEM, tel que :

$$\vec{f}_{pr} = \frac{\vec{F}_{pr}}{N} \quad (4.14)$$

Où N est le nombre d'éléments discrets appartenant à la paroi supérieure.

En outre, dans le dispositif réel, le poids de l'échantillon de silice de l'ordre de 0,4 N. Cet effort est négligeable devant la précharge exercée, de l'ordre de la dizaine de Newton. Aussi, l'effet du poids sera négligé.

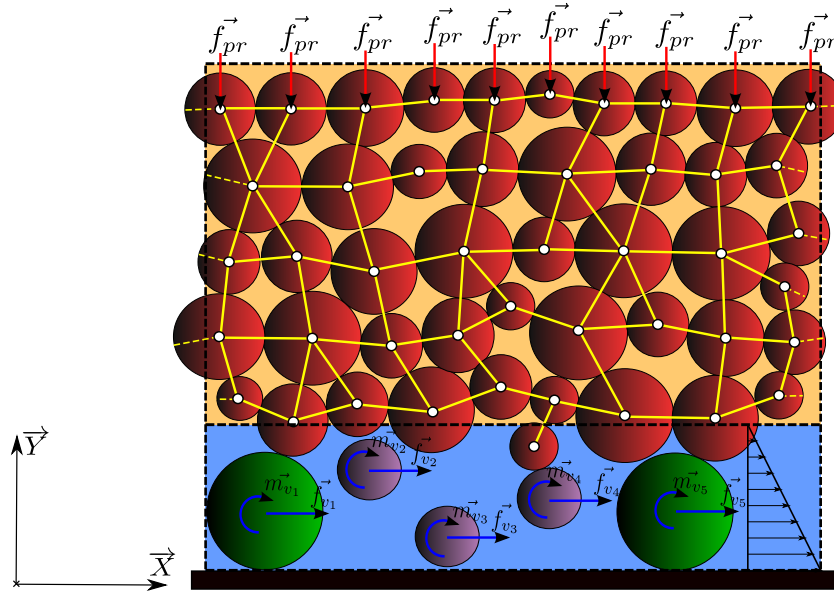


FIGURE 4.12 – Illustration 2D des chargements et conditions limites

Modélisation de l'avance de l'outil

À l'échelle dimensionnelle modélisée, il est supposé que l'outil se déplace en translation. Ce déplacement implique :

une mise en mouvement des particules abrasives. Il a été vu, au cours de la section 4.1, que certaines particules abrasives, au cours des opérations de doucissage sont « enchâssées » dans l'outil. Aussi, le taux de répartition entre le nombre d'abrasifs liés à l'outil et le nombre d'abrasifs libres au cours du procédé n'est pas connu. Entre ces deux possibilités, un modèle intermédiaire a été choisi. Il est imposé que les déplacements, suivant l'axe de translation de l'outil, des particules abrasives soient égaux au déplacement de l'outil. La composante verticale, selon $\vec{Y}$ est également imposée. Les abrasifs ne peuvent se déplacer verticalement afin d'éviter des phénomènes de rebond causés par les pics de charge (voir section 4.3.4). Par contre, les éléments discrets abrasifs sont libres de se déplacer suivant la composante horizontale $\vec{Z}$, perpendiculaire à la vitesse d'avance de l'outil. Ce modèle est qualifié d'abrasif « semi-libre ».

une mise en mouvement du fluide. La section 4.3.3 décrit le modèle d'interaction entre l'outil, le fluide et les éléments discrets immergés. Le déplace-

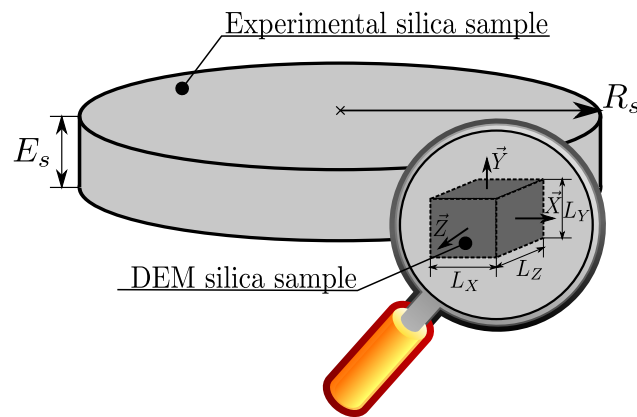
ment de l'outil entraîne une mise en mouvement du fluide suivant l'axe $\vec{X}$. Or, les vitesses des différents corps immergés et du fluide suivant les directions normales au déplacement de l'outil ($\vec{Y}$ et $\vec{Z}$) sont supposées négligeables par rapport aux vitesses mises en jeu suivant l'axe $\vec{X}$. Les actions du fluide par effet visqueux sur ces corps seront donc négligées suivant ces directions. Seules, les actions exercées suivant l'axe $\vec{X}$ seront prises en compte. Les éléments discrets immergés (*abrasifs* et *usures*) sont alors soumis à des forces et moments d'entraînement visqueux (notés $\vec{f}_{v_n}$ et $\vec{m}_{v_n}$ sur la figure 4.12). Toutefois, les éléments discrets *abrasifs* étant contraints à se déplacer à la même vitesse que l'outil, les actions du fluide sur ceux-ci ne sont pas calculées.

4.3.5 Méthodologies de quantification des paramètres du modèle

Un enjeu fort de ce travail de thèse est de proposer une modélisation *quantitative* du doucissage du verre. Aussi, il a été prêté une attention particulière à la quantification des différents chargements et conditions limites du modèle DEM.

L'étude peut être qualifiée de paramétrique, les conditions de doucissage (précharge, vitesse de l'outil, taille et concentration des abrasifs) définissent les paramètres d'entrées de l'étude. La démarche est alors simple. Il est retenu un jeu de paramètres définissant un cas de référence. Chacun des paramètres est ensuite étudié indépendamment en faisant varier la valeur de celui-ci à partir du cas de référence et de la plage des valeurs expérimentales.

Les sous sections suivantes décrivent les différentes méthodes permettant de traduire les conditions expérimentales du doucissage en regard de l'échelle dimensionnelle du modèle DEM (voir illustration des deux échelles sur la figure 4.13). Les paramètres de références sont distingués par un astérisque (*).



Symboles	Descriptions	Valeurs
E_s	Épaisseur de l'échantillon de silice	1 cm
R_s	Rayon de l'échantillon de silice	2,5 cm
L_X	Longueur du volume de silice DEM	150 μm
L_Y	Largeur du volume de silice DEM	150 μm
L_Z	Épaisseur du volume de silice DEM	150 μm

FIGURE 4.13 – Échelles de l'échantillon de silice réel et de l'échantillon numérique

Quantification de la précharge

Il est supposé que la précharge appliquée est constante spatialement et temporellement au cours de l'opération du doucissage. Aussi, il est possible de quantifier la précharge M_{pr} sous la forme d'une pression p appliquée sur la surface de l'échantillon réel de silice :

$$p = \frac{M_{pr}g}{\pi R_s^2} \quad (4.15)$$

Où g est la gravité. Il est également supposé que la contrainte due à la précharge est répartie uniformément dans le volume de silice. Aussi, la valeur de la pression p est identique à la pression précédemment calculée pour le modèle DEM. Il est possible de quantifier l'effort de précharge total $\vec{F}_{pr}$ (voir section 4.3.4 et figure 4.12) :

$$F_{pr} = \frac{M_{pr}gL_XL_Z}{\pi R_s^2} \quad (4.16)$$

En appliquant cette précédente égalité, il est possible d'établir une correspondance entre les valeurs de précharge expérimentales et celles du modèle DEM. Le tableau suivant (tableau 4.4) donne cette correspondance.

Désignation	Unité	Symbole	valeurs		
Précharge Exp.	Kg	M_{pr}	0,8	1,6 (*)	2,4
Précharge DEM	N	F_{pr}	$0,9 \cdot 10^{-4}$	$1,8 \cdot 10^{-4}$ (*)	$2,7 \cdot 10^{-4}$

(*) Valeurs de référence

TABLE 4.4 – Valeurs des paramètres de précharge

Quantification de la vitesse d'avance de l'outil

La gamme de vitesse de rotation de l'outil est comprise entre cinq et trente tours par minute. Aussi, pour une valeur maximale de rotation de l'outil $\omega_p = 30 \text{ tr.min}^{-1}$, et pour un temps total simulé $T = 1,7 \text{ ms}$, la rotation de l'outil n'est que de 0,3 degré. Compte tenu du temps simulé T et de la vitesse de rotation du plateau ω_p , l'angle de rotation du plateau est considéré comme négligeable.

Il sera admis que la zone simulée par le modèle DEM est située sur la circonférence de l'échantillon de silice. Cette zone est identifiée par le point « C » sur la figure 4.14. À cette échelle, le volume simulé du modèle DEM peut-être assimilé à un point. En conséquence, comme l'angle de rotation de l'outil est négligeable pour le modèle DEM, le mouvement de celui-ci sera supposé être une translation suivant la direction du vecteur vitesse $\vec{V}_{c,s/p}$.

Il sera également admis que pour le cas étudié la vitesse de rotation de l'échantillon de silice ω_s par rapport au bâti est nulle. Ainsi, la vitesse tangentielle de l'échantillon de silice par rapport à l'outil $\vec{V}_{c,s/p}$ est alors :

$$\|\vec{V}_{c,s/p}\| = R_{max}\omega_p \quad \text{avec} \quad R_{max} = 15 \text{ cm} \quad (4.17)$$

En appliquant cette précédente égalité, il est possible d'établir une correspondance entre les valeurs de vitesse de rotation expérimentales de l'outil et celles du modèle DEM. Le tableau suivant (tableau 4.5) donne cette correspondance.

Désignation	Unité	Symbole	valeurs		
Vitesse de rotation de l'outil Exp.	$tr.min^{-1}$	ω_p	5	15 (*)	25
Vitesse de translation de l'outil DEM	$m.s^{-1}$	v_p	0,08	0,24 (*)	0,40

(*) Valeurs de référence

TABLE 4.5 – Valeurs des vitesses d'avance de l'outil

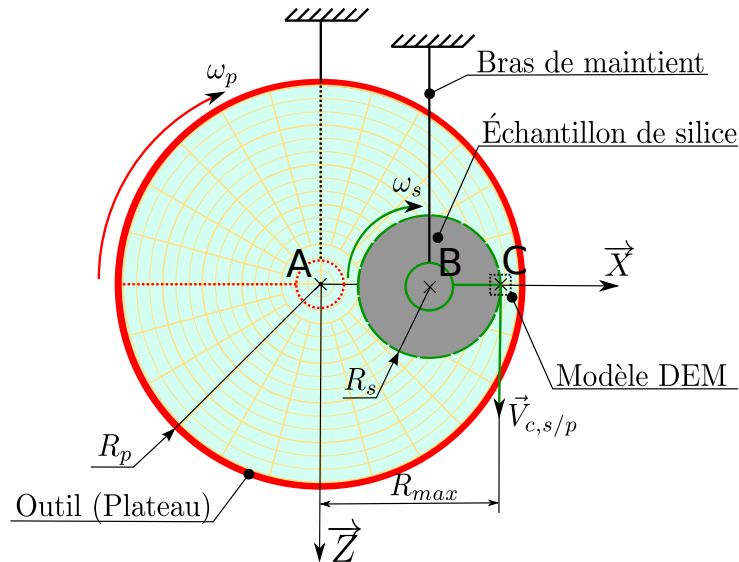


FIGURE 4.14 – Illustration du calcul vitesse de translation de l'outil

Quantification de la taille des particules abrasives

Dans ce cas particulier, il n'est pas nécessaire de recourir à une traduction des conditions expérimentales. La dimension moyenne expérimentale est directement injectée dans le modèle numérique. Par contre, la variabilité des dimensions réelles des abrasifs n'est pas prise en compte. Les abrasifs sont donc tous de dimension égale dans le modèle numérique. Le tableau suivant (tableau 4.6) donne le jeu des valeurs utilisées.

Désignation	Unité	Symbole	valeurs		
Dimension moyenne Exp.	μm	R_{ab}	5	10 (*)	15
Dimension DEM	μm	R_{ab}	5	10 (*)	15

(*) Valeurs de référence

TABLE 4.6 – Valeurs des dimensions des abrasifs

Quantification de la concentration en particules abrasives du slurry

Il a été choisi de modéliser les particules abrasives par des particules sphériques (voir section 4.3.2). Afin d'évaluer le nombre de particules abrasives du modèle DEM, il est nécessaire de connaître l'épaisseur de l'interface de doucissage. Une première hypothèse suppose la présence d'une seule couche d'abrasif

entre le substrat de silice et l'outil [124, §4.4]. Dans ce cas, l'épaisseur de l'interface est équivalente au diamètre moyen des abrasifs. Toutefois, cette dernière hypothèse est sujette à discussion [95, §4]. À ce jour, les études expérimentales ne permettent pas d'arbitrer cette question en raison des difficultés techniques à réaliser des mesures de l'interface de doucissage durant le procédé. Dans une première approche, il sera retenu l'hypothèse la plus simple où il est supposé une unique couche d'abrasif. Il est également supposé que la répartition spatiale des abrasifs est homogène au niveau de l'interface silice/outil.

Aussi, connaissant le rayon des particules abrasives R_{ab} , les dimensions du bloc de silice simulé L_X , L_Y et L_Z et la concentration $c\%$ en abrasif du *slurry*, il est possible d'en déduire le nombre d'abrasifs N_{ab} :

$$N_{ab} = c\% \frac{V_{sl}}{V_{ab}} \quad \text{avec} \quad V_{ab} = \frac{4}{3}\pi R_{ab}^3 \quad \text{et} \quad V_{sl} = 2L_X L_Z R_{ab} \quad (4.18)$$

Où :

- V_{ab} est le volume d'une particule abrasive ;
- V_{sl} est le volume du slurry à l'interface de contact.

Le nombre réel N_{ab} est ensuite approché par le nombre entier le plus proche. En appliquant cette précédente égalité, il est possible d'établir une correspondance entre les valeurs en concentration des particules abrasives expérimentales et le nombre de particules abrasives du modèle DEM. Le tableau suivant (tableau 4.7) donne cette correspondance pour une valeur $R_{ab} = 10\mu m$ (correspondant au cas référence).

Désignation	Unité	Symbole	valeurs		
Concentration Exp.	%	$c\%$	6% (*)	12%	18%
Nb abrasif DEM		N_{av}	7 (*)	13	19

(*) Valeurs de référence

TABLE 4.7 – Valeurs des paramètres de concentration

4.4 Résultats numériques et discussions

Cette section propose, dans un premier temps, une description des variables d'intérêt et des méthodes de post-traitement. Les principaux résultats seront ensuite décrits ainsi que les tendances pouvant être dégagées en regard des paramètres du procédé. En dernier lieu, il s'engagera une discussion des résultats obtenus par rapport aux données expérimentales ainsi qu'une critique du modèle et des différentes améliorations pouvant être apportées.

Le tableau 4.8 synthétise les différents jeux de paramètres utilisés pour chaque expérience de doucissage tels qu'ils sont définis dans la section précédente (section 4.3.5). Les valeurs des paramètres sont exprimées suivant les unités expérimentales (colonnes *exp*) et traduites dans des unités adaptées aux conditions numériques (colonnes *dem*).

Cette gamme d'essais numériques permet de tester indépendamment chacun des paramètres en prenant comme référence l'expérience n°1 intitulée « référé-

rence ». Dans ce tableau, les valeurs des paramètres marquées d'un « - » signifient qu'elles sont égales à celles de l'expérience de référence.

Ces expériences sont réalisées à l'aide d'un seul échantillon numérique de silice afin de vérifier que les résultats obtenus sont bien les conséquences des effets des paramètres du procédé plutôt que la signature de l'échantillon seul. En effet, les profils de rupture obtenus pourraient être la conséquence unique des fragilités intrinsèques de l'échantillon numérique utilisé.

Intitulé	Paramètres							
	exp M_{pr} Kg	dem F_{pr} mN	exp ω_p $tr.min^{-1}$	dem v_p $m.s^{-1}$	exp $\bar{R}_{ab}$ μm	dem R_{ab} μm	exp $c\%$ %	dem N_{av}
référence	1,6	0,18	15	0,24	10	10	6	7
pression -	0,8	0,09	-	-	-	-	-	-
pression +	2,4	0,27	-	-	-	-	-	-
vitesse -	-	-	5	0,08	-	-	-	-
vitesse +	-	-	25	0,40	-	-	-	-
rayon -	-	-	-	-	5	5	-	-
rayon +	-	-	-	-	15	15	-	-
concentration +	-	-	-	-	-	-	12	13
concentration ++	-	-	-	-	-	-	18	19

TABLE 4.8 – Vue synthétique des différentes conditions de doucissage

4.4.1 Variables d'intérêt et méthodologie de post-traitement

Variable d'intérêt

Les opérations de doucissage consistent, par passes successives, et à l'aide d'abrasifs de plus en plus fins, à atteindre de façon progressive une planéité et une rugosité de surface voulue (voir section 1.2). Ce processus permet également, au cours des passes successives, de diminuer progressivement la profondeur maximale de SSD dans l'épaisseur de la pièce de silice. Ce dernier processus peut être qualifié de lent puisque, pour chacune des passes, il s'agit d'éroder l'épaisseur de la pièce d'une dimension au moins égale à la profondeur de SSD générée par la précédente opération. De plus, malgré les précautions prises par les *polisseurs* certaines de ces SSD subsistent. Pour ces raisons, la profondeur maximale de SSD sera considérée comme la principale variable d'intérêt.

De plus, en raison de la condition de déplacement unidirectionnel qui joue le rôle d'accélérateur du processus d'abrasion (voir section 4.4.3), le taux d'enlèvement matière est largement surévalué par rapport aux observations expérimentales. Dans ces conditions, il n'est donc pas possible de réaliser une étude quantitative de cette variable. Aussi, dans une première approche, il est proposé de restreindre cette étude à l'exploitation des profondeurs de SSD.

Zones temporelles d'intérêt

La figure 4.15 montre l'évolution du nombre de liaisons cohésives *poutres* rompues au cours de la simulation. Cette évolution montre que la simulation est

ponctuée par de très brefs instants de « crise ». Ces crises sont révélées par des augmentations brutales du nombre de liaisons cohésives rompues, marquées sur la figure 4.15 par des traits en pointillés verts.

Le nombre d'itérations d'une simulation avoisine quelque dix millions. Dans ces conditions, il n'est pas possible d'effectuer des prises de données pour chaque itération en regard du volume de données et des temps de calcul nécessaires au post-traitement. Il n'est pas non plus indiqué de réaliser des prises de mesure à intervalles réguliers (toutes les n itérations), le risque étant de ne pas détecter une zone de crise. Pour pallier ces problèmes, les prises de mesure, au cours de la simulation, sont déclenchées de façon événementielle lorsqu'une liaison cohésive approche de sa contrainte à rupture.

Enfin, lors de la phase de post-traitement, seules les données issues des zones de crise sont traitées de façon à minimiser les temps de traitements.

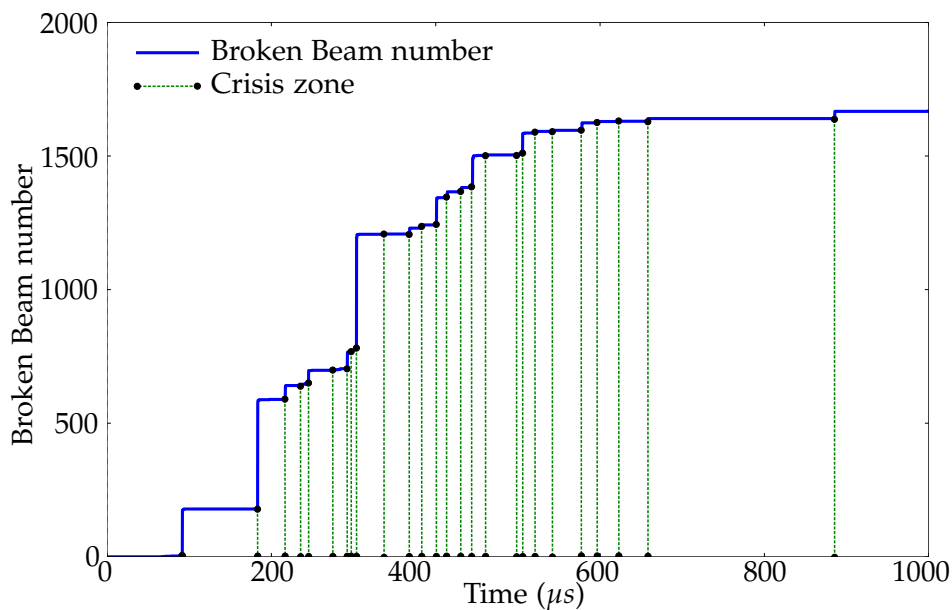


FIGURE 4.15 – Évolution du nombre de liaisons cohésives *poutres* rompues au cours du temps - *Conditions de référence* -

Méthodologie de mesure

Dans les parties qui vont suivre, il sera considéré qu'une liaison cohésive *poutre*, dont le critère de rupture a été atteint, représente une micro-fissure au sein du matériau. Il sera considéré alors que « l'altitude » d'une liaison cohésive rompue est représentative de la profondeur de la micro-fissure. Cette mesure est réalisée en prenant comme référence la surface abrasée du bloc de silice. Celle-ci est définie par les éléments discrets *silices* appartenant à la couche inférieure et potentiellement en interaction avec les particules abrasives. Il est ainsi possible d'établir une cartographie 3D de la surface abrasée selon la méthode suivante :

1. Au cours de la simulation, les positions des éléments discrets appartenant à la couche abrasée sont extraites ;
2. En retranchant, pour chaque position, le rayon de l'élément discret associé selon la composante $\vec{Y}$, il est déduit un nuage de points considéré comme

représentatif de la morphologie de la surface abrasée ;

3. Ce nuage de points est ensuite traité par interpolation linéaire pour construire une cartographie complète de cette surface.

La surface bleue (S) représentée sur la figure 4.16 montre le résultat obtenu pour un doucissage dans les conditions de référence et pour un temps simulé de 507 μs .

En outre, les positions des liaisons cohésives *poutres* rompues sont également sauvegardées au cours des simulations. Sur la figure 4.16, ces positions sont représentées par des points rouges : *Broken beams*. Un traitement numérique permet ensuite de déduire, pour chaque point P_n , représentant une liaison rompue, le point d'intersection I_n entre la surface interpolée et la droite $(P_n, \vec{Y})$. Les points I_n sont représentés sur cette figure par des carrés bleus : *Broken beams projected on (S)*. L'altitude de la liaison cohésive rompue i et donc la profondeur de SSD est définie par :

$$c_i = \left\| \overrightarrow{P_i I_i} \right\| \quad (4.19)$$

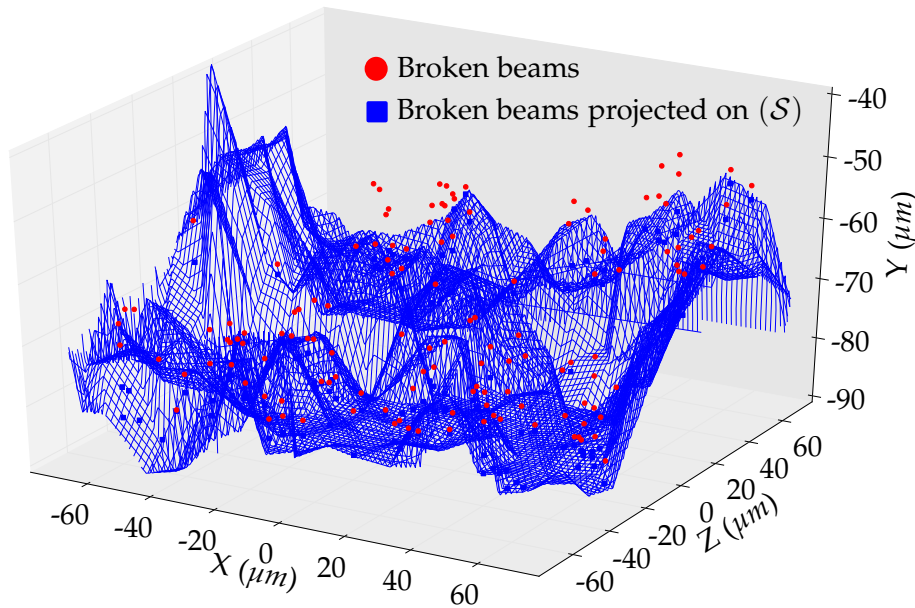


FIGURE 4.16 – Cartographie de la surface abrasée et des SSD - Conditions de référence, temps simulé de 507 μs -

À partir de ces données, il est possible d'établir un profil de répartition des SSD dans l'épaisseur du substrat de silice. Ces profils sont présentés sous forme d'histogrammes illustrés, par la figure 4.17, où apparaît le nombre de SSD en fonction de leurs profondeurs respectives. Les profondeurs de SSD sont comptabilisées par intervalle de 2,5 μm .

Des observations expérimentales ont montré que la loi de répartition peut être approchée par une fonction exponentielle décroissante de la forme [124] :

$$f(x) = ae^{-bx} \quad \text{avec } b \geq 0 \quad (4.20)$$

Afin de caractériser ce profil de répartition, cette dernière relation est utilisée. La méthode des moindres carrés est mise à profit pour calculer les coefficients a et b permettant de minimiser les écarts entre fonctions théoriques et nuages de points. Dans le cas du profil illustré sur la figure 4.17, ces valeurs sont :

$$a = 55 \quad \text{et} \quad b = 0,13 \quad (4.21)$$

Ces coefficients permettent de représenter la courbe de répartition théorique illustrée par la courbe rouge sur la figure 4.17. Les valeurs des coefficients a et b sont respectivement des indicateurs :

- du nombre de SSD proches de la surface abrasée (de profondeur nulle) ;
- du gradient de décroissance du nombre de SSD dans l'épaisseur.

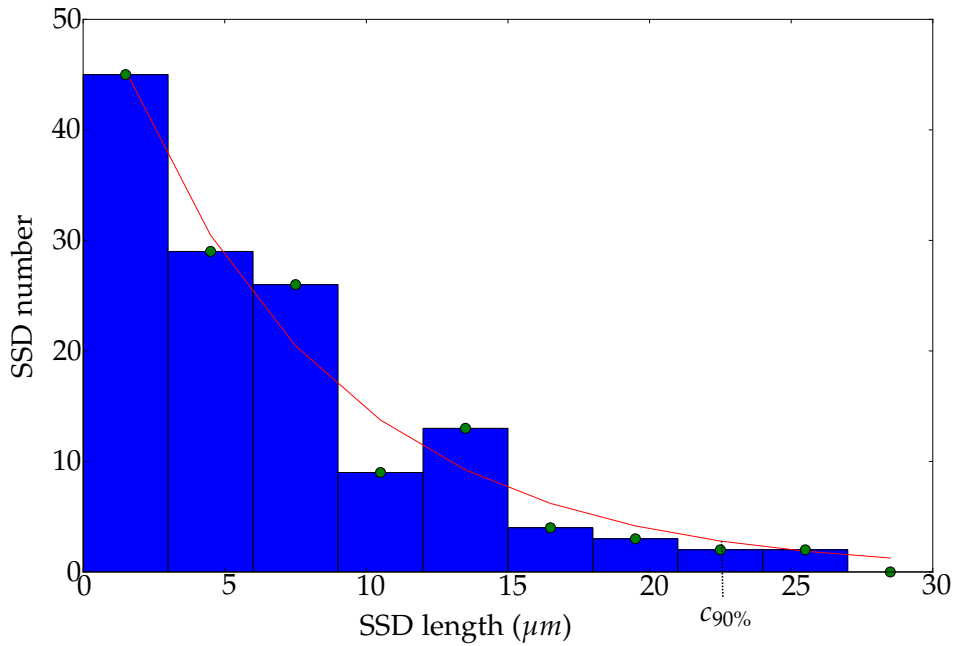


FIGURE 4.17 – Profil de répartition des SSD dans l'épaisseur - Conditions de référence, temps simulé de 507 μs -

Comme il a été vu en introduction de cette section, la principale variable d'intérêt dans ces calculs est la profondeur maximale de SSD. Une méthode de mesure directe serait de prendre la valeur maximale du nuage de points. Toutefois, cette dernière méthode induit le risque de baser des résultats sur des points statistiquement non significatifs. Il est plutôt choisi de mettre à profit la loi de répartition théorique précédemment définie et jugée plus fiable (car sa valeur est déduite de l'ensemble des points). En raison du faible nombre de l'échantillonnage (environ une centaine de SSD), il sera considéré que la profondeur maximale de SSD est définie par la valeur seuil de profondeur dont 90% des SSD sont inférieures. Cette valeur, notée $c_{90\%}$, est définie par :

$$c_{90\%} = \frac{\ln(0,1)}{-b} \quad (4.22)$$

Toutefois, cette valeur étant variable au cours de la simulation, il est également intéressant de tracer son évolution au cours du temps simulé (voir figure

4.18). Ces courbes peuvent permettre par la suite de comparer les évolutions pour différentes expériences en ayant une vue globale sur toute la durée de la simulation. La figure 4.18 montre que la profondeur maximale de SSD semble se stabiliser au cours de la simulation. Cette tendance semble se reproduire pour chacune des expériences numériques. Aussi, cette valeur stabilisée de $c_{90\%}$ sera prise comme référence pour qualifier une expérience numérique.

Toutefois, cette procédure de mesure, automatisée dans le cadre de cette étude, doit être accompagnée de vérification. Il s'agit de vérifier que le profil de SSD relevé correspond bien à une fonction exponentielle décroissante. Ces vérifications ont été réalisées visuellement pour chacune des expériences dans le domaine temporel de la zone stabilisée.

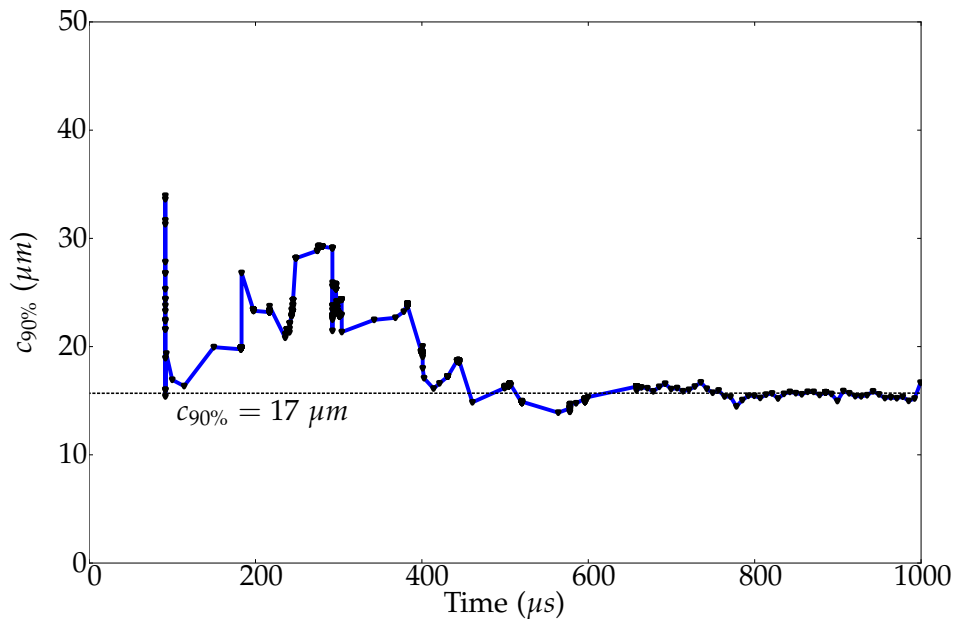


FIGURE 4.18 – Évolution de la profondeur maximale de SSD - Conditions de référence -

4.4.2 Étude de l'expérience de référence

Au cours de ce paragraphe, il est proposé d'étudier l'évolution de différents paramètres de l'expérience de référence au cours du temps.

Évolution de l'altitude de la paroi supérieure

La figure 4.19 illustre l'évolution de l'altitude de la paroi supérieure du bloc de silice¹.

Une première remarque concerne l'évolution décroissante de l'altitude. C'est ici une des conséquences directes des conditions imposées. En effet, la paroi supérieure est soumise à deux conditions (voir section 4.3.4) :

1. Un chargement vertical représentatif de la précharge imposée expérimentalement ;

1. Plus précisément, l'altitude est définie par la position moyenne des éléments discrets composant la paroi supérieure du bloc de silice projetée sur l'axe vertical $\vec{Y}$.

2. Une condition de déplacement unidirectionnel : la paroi supérieure de silice ne peut se déplacer que dans la direction descendante.

Cette tendance décroissante est donc la conséquence directe des conditions ci-dessus. Toutefois, l'altitude de la paroi supérieure traduit également un mouvement de corps rigide du bloc de silice. Celui-ci n'est rendu possible que par la perte de volume due à l'enlèvement de matière. En conséquence, l'évolution de l'altitude de la paroi supérieure peut être un indicateur du taux d'enlèvement de matière. En raison de la condition de déplacement unidirectionnel qui joue le rôle d'accélérateur du processus d'abrasion (voir section 4.4.3 et 4.4.1), le taux d'enlèvement matière ne sera pas exploité quantitativement dans le cadre de cette étude.

Il est intéressant également de tracer l'évolution de l'altitude (courbe bleue) en regard du nombre des liaisons cohésives rompues (courbe verte en pointillées). Ces évolutions montrent qu'il n'y a pas de correspondance temporelle précise des diminutions brusques d'altitude avec les augmentations brutales de liaisons rompues. En fait, une diminution d'altitude est permise lorsque les abrasifs sont positionnés dans une configuration où il subsiste « un jeu » entre la surface abrasée et les abrasifs. Par contre, une diminution de l'altitude du bloc de silice va entraîner une perte d'espace de mobilité des abrasifs. La probabilité est donc plus grande, pour les abrasifs, de se trouver dans une configuration où ceux-ci vont solliciter mécaniquement la silice et donc favoriser l'abrasion.

Une autre caractéristique de l'évolution de l'altitude est son séquençement en trois zones. La première zone, nommée *Stationnary zone*, montre une faible évolution de celle-ci. Il est observé ensuite une brusque diminution (*Decreasing zone*) puis une nouvelle zone stationnaire. Il sera vu par la suite que, pour chacune des expériences, ce profil sera retrouvé.

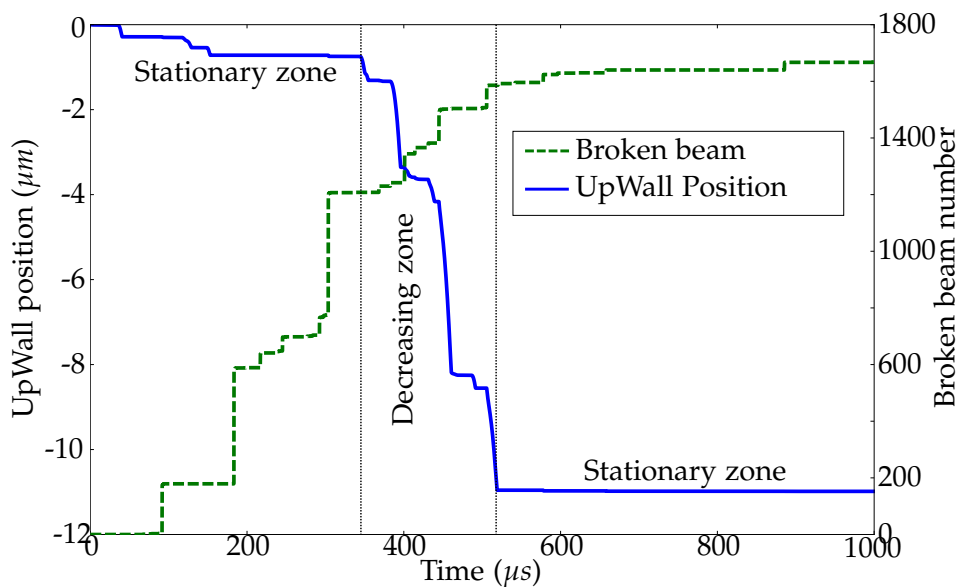


FIGURE 4.19 – Évolution de la position moyenne projetée sur $\vec{Y}$ de la paroi supérieure du bloc de silice - Conditions de référence -

Évolution de l'effort normal appliquée à la paroi supérieure

La condition de déplacement unidirectionnel de la paroi supérieure permet de favoriser l'abrasion. Mais, dans ces conditions quelles sont alors les véritables effets de la précharge ? Pour étudier cette question, il est proposé de tracer l'évolution des efforts normaux en fonction du temps (voir figure 4.20)².

Une première remarque porte sur la présence de surintensités de chargement par rapport à la précharge appliquée. La précharge étant de $0,18 \text{ mN}$ et les pics de charge de l'ordre de 180 mN , il existe un rapport de 10^3 entre ces deux valeurs. Dans ces conditions, il est difficile d'établir un réel effet de la précharge, celle-ci étant négligeable devant les pics de charge dus à la condition de déplacement.

Une seconde remarque porte sur les morphologies de ces pics de charge. Il en est observé de deux types :

1. les pics de charge n'entraînant pas de dommage au sein de la silice. C'est le cas par exemple des trois premières surintensités tracées sur la figure 4.20 ;
2. les pics de charge entraînant des dommages. Ils sont caractérisés par des discontinuités entraînant une « remise à zéro » brutale du chargement. Ces discontinuités correspondent à des relaxations de la silice induites par l'apparition de dommages. Cette correspondance temporelle est visible sur la figure 4.20 où les évolutions brutales de liaisons cohésives rompues sont accompagnées de discontinuités de chargement.

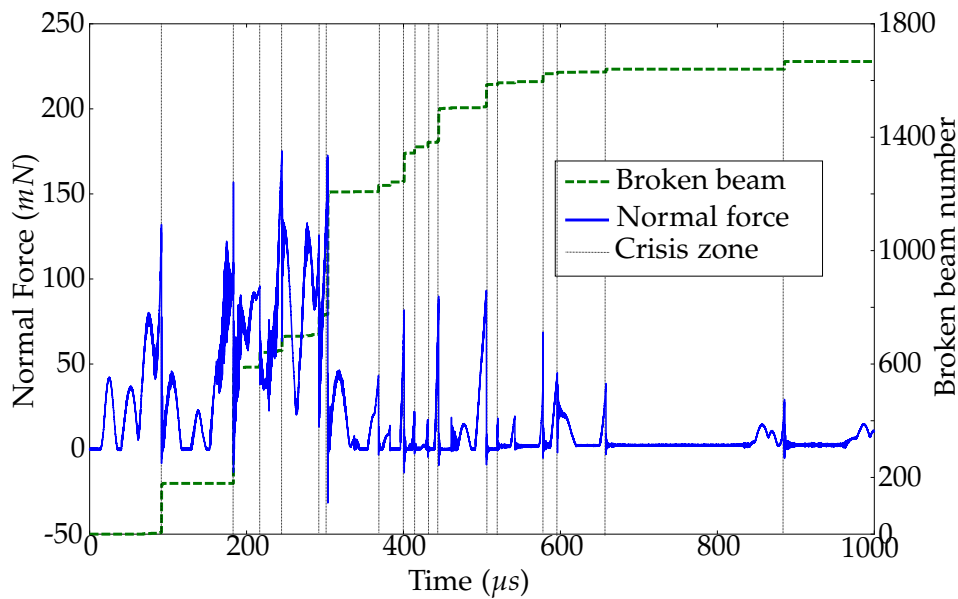


FIGURE 4.20 – Évolution de la force normale appliquée sur la paroi supérieure du bloc de silice - Conditions de référence -

Enfin, une troisième remarque porte sur les durées caractéristiques des pics de charge. Sur la figure 4.20, il semble que les premiers pics, n'entraînant pas de rupture, possèdent approximativement les mêmes durées. Cela est confirmé en établissant une mesure fine de ces durées. Il est obtenu une moyenne d'environ $21,5 \mu\text{s}$ pour un écart type d'environ $3,1 \mu\text{s}$.

2. Plus précisément, la somme des efforts appliqués sur les éléments discrets appartenant à la paroi supérieure du bloc de silice projetée sur l'axe vertical $\vec{Y}$.

Ces durées, compte tenu de la vitesse de déplacement suivant $\vec{X}$ des abrasifs de $0,24 \text{ m.s}^{-1}$, correspondent à un déplacement d'environ $5 \mu\text{m}$. Or, cette dernière valeur correspond au rayon moyen des éléments discrets *silice*. Pour expliquer ce phénomène, la figure 4.21 propose une illustration 2D simplifiée d'un cycle de chargement abrasif/*silice*. Cette illustration fait apparaître qu'un cycle de chargement complet correspond à une avance des abrasifs d'une valeur égale au diamètre des éléments discrets *silice*. Cette dernière conclusion est alors en contradiction avec les observations, puisqu'il est observé un déplacement équivalent à un rayon et non à un diamètre des éléments discrets *silice*.

En fait, le cas illustré sur la figure 4.21 correspond aux cas où les abrasifs sont « liés » et donc contraints à se déplacer suivant la direction d'avance de l'outil. Dans le cas étudié, les abrasifs sont libres, et peuvent donc « contourner l'obstacle » afin de minimiser les énergies en présence. Dans ce cas, la durée effective d'un cycle de chargement est deux fois plus courte. En effet, lorsque l'abrasif, en contact avec un élément discret *silice*, est contraint à se déplacer suivant $\vec{X}$, celui-ci effectue également un mouvement de contournement dans une direction perpendiculaire. Finalement, le contact est rompu lorsque le déplacement de l'abrasif sur $\vec{X}$ est équivalent au rayon de l'élément discret *silice* avec lequel il est en contact. La valeur des durées des pics de charge observées correspond bien à ce déplacement. Aussi, est-il légitime de penser que le modèle d'abrasif « semi-libre » (voir section 4.3.4), utilisé dans ces simulations, est plus proche des conditions réelles de doucissage (abrasif libre) que d'ébauchage (abrasif lié).

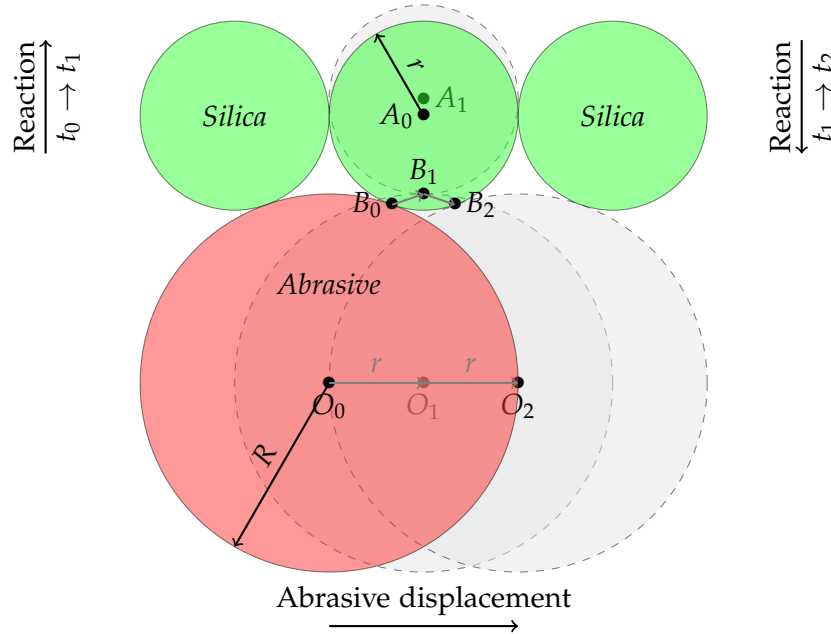


FIGURE 4.21 – Illustration 2D des mécanismes de chargement abrasif/*silice*

4.4.3 Influence de la pression (précharge)

Les courbes d'évolution des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour les différentes conditions de précharges sont visibles sur la figure 4.22.

Les courbes d'évolution des efforts normaux montrent que les pics de charge sont approximativement de mêmes intensités alors que les valeurs de précharge sont multipliées respectivement par 1,5 et par 3 entre l'expérience « pression - » et les expériences « référence » et « pression + ».

De plus, les valeurs stabilisées de profondeurs maximales de SSD, $c_{90\%}$ sont sensiblement similaires (bien que leurs évolutions ne le soient pas). Ces comparaisons montrent que l'effet de la précharge, associé à la condition de déplacement unidirectionnel, semble être minime sur les profils de SSD générés. Ces résultats sont en désaccord avec les observations expérimentales. Ces dernières montrent qu'une augmentation de la pression a pour effet de diminuer la profondeur maximale des SSD.

Par contre, les courbes d'évolution des altitudes des parois supérieures montrent un décalage temporel. En effet, les zones nommées *Decreasing zone* (voir section 4.4.2), débutent et finissent plus tôt lorsque la précharge augmente. Il semble que la précharge joue essentiellement un rôle d'accélérateur du phénomène d'abrasion et cela en accord avec la loi de Preston [107] décrite au cours du chapitre 1 (voir section 1.5).

Aussi, même si les valeurs des précharges semblent négligeables en comparaison des intensités de pic de charge et des déplacements induits, celles-ci semblent capables d'accélérer le processus d'abrasion. Par contre, il n'est observé aucun effet notable sur le profil de répartition des SSD.

Toutefois, les conclusions précédentes doivent être prises avec précaution. La présence de la condition de déplacement unidirectionnel entraînant des surintensités de charge ne permet pas d'établir une similitude entre l'étude paramétrique expérimentale portant sur la pression et la présente étude numérique. En effet, plusieurs questions restent ouvertes :

- Ces phénomènes de surintensités existent-ils dans la réalité ?
- Si ces phénomènes existent, sont-ils du même ordre de grandeur que ceux observés numériquement ?

Aussi, il est prévu, dans un futur proche, la mise en service d'un *Jig* équipé de différents capteurs dont les mesures devraient apporter un éclairage sur ces interrogations.

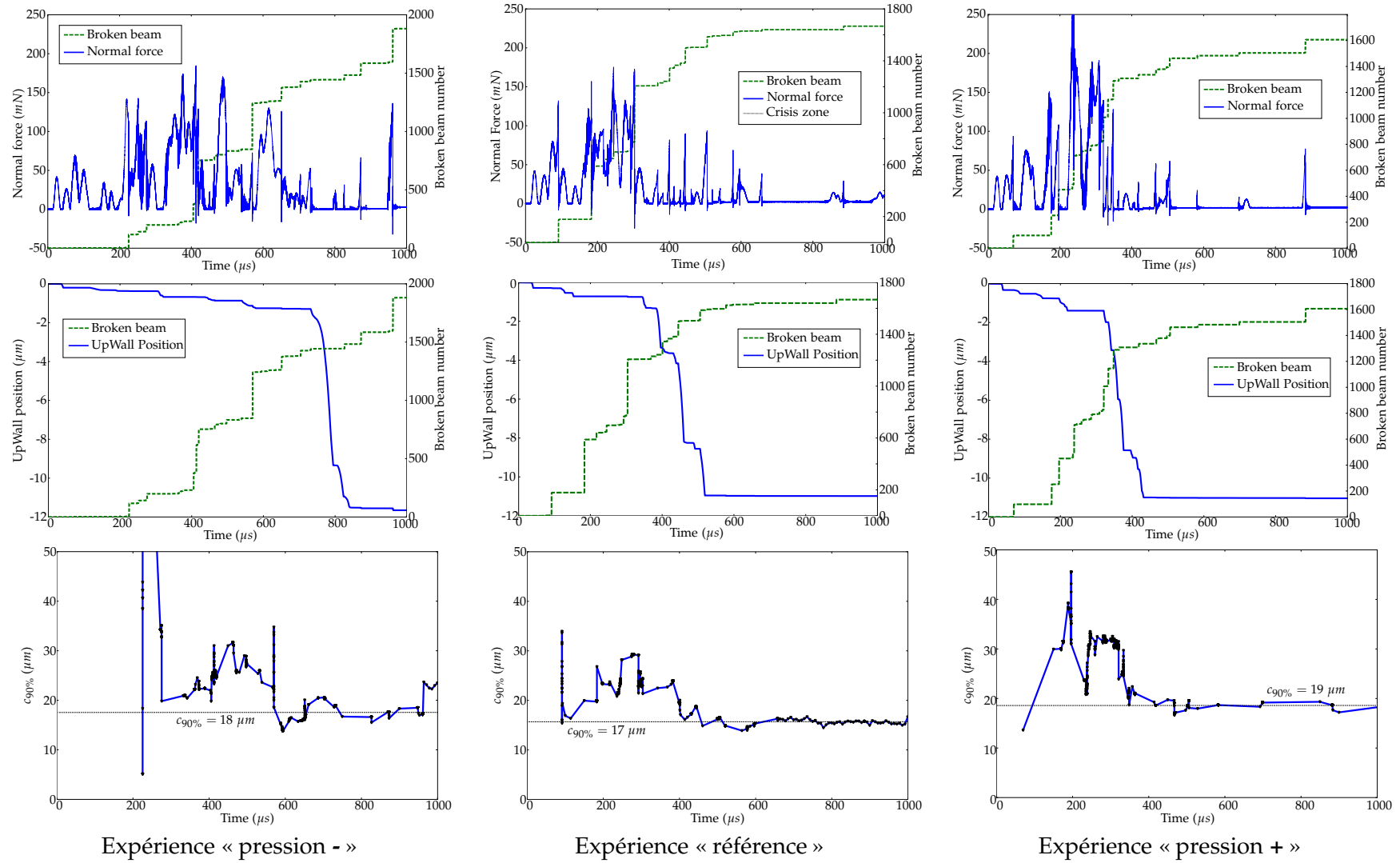


FIGURE 4.22 – Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes conditions de précharge

4.4.4 Influence de la vitesse de l'outil

Les courbes d'évolution des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour les différentes conditions d'avance de l'outil sont visibles sur la figure 4.23. Dans ce cas d'étude, l'expérience « vitesse - » a été poussée plus avant afin de compenser la faible vitesse d'avance des abrasifs. Le temps simulé pour cette expérience est environ de 2,5 *ms* alors que les autres expériences proposent une fenêtre temporelle de simulation de 1 *ms*.

En premier lieu, il apparaît que la vitesse d'abrasion semble fortement diminuer avec l'expérience « vitesse - ». Par contre, cette évolution est moins accentuée entre les expériences « référence » et « vitesse + ». Toutefois, cette dernière remarque est en mettre en perspective vis-à-vis des valeurs introduites. Les vitesses étant de 5, 10 et 15 tours par minute, il existe un rapport de 2 entre les expériences « vitesse - » et « référence » et seulement de 1,5 entre les expériences « référence » et « vitesse + ».

D'autre part, il est observé que les durées de pics de charge sont proportionnelles aux vitesses d'avance des abrasifs. Ce point permet de confirmer l'hypothèse établie dans la section précédente où il est supposé que cette durée est caractéristique du temps de parcours des abrasifs d'une distance équivalant au rayon moyen des éléments discrets *silice*.

Enfin, les évolutions des profondeurs maximales de SSD montrent une différence marquée entre l'expérience « vitesse - » et les autres expériences. En effet, la valeur de $c_{90\%}$ décroît régulièrement au cours de l'expérience pour atteindre une valeur d'environ 12 μm , alors que pour les expériences « référence » et « vitesse + », $c_{90\%}$ se stabilise autour d'une valeur de 18 μm .

De manière générale, de forts écarts de résultats sont constatés pour des vitesses d'avance faibles. Aussi de faibles vitesses d'avance de l'outil semblent diminuer la vitesse d'abrasion et ceci en accord avec la loi de Preston précédemment citée. Par contre, toujours pour de faibles vitesses, la profondeur maximale de SSD diminue. Cette dernière tendance est en désaccord avec les résultats expérimentaux synthétisés en annexe H sur la figure H.1.

Sur ce dernier point, une explication possible de cette différence entre résultats expérimentaux et numériques est sans doute une insuffisance du modèle numérique de l'interface d'abrasion. Dans le modèle numérique, les efforts visqueux exercés par le fluide sur la surface abrasée sont négligés. Or, en augmentant la vitesse de l'outil, ces efforts peuvent devenir non négligeables et influencer le procédé. Il est même possible, au-delà d'un certain ratio vitesse/précharge, que des phénomènes de portance apparaissent. Le fluide est alors capable de reprendre significativement une partie des efforts normaux exercés sur la pièce et peut donc lisser les efforts normaux.

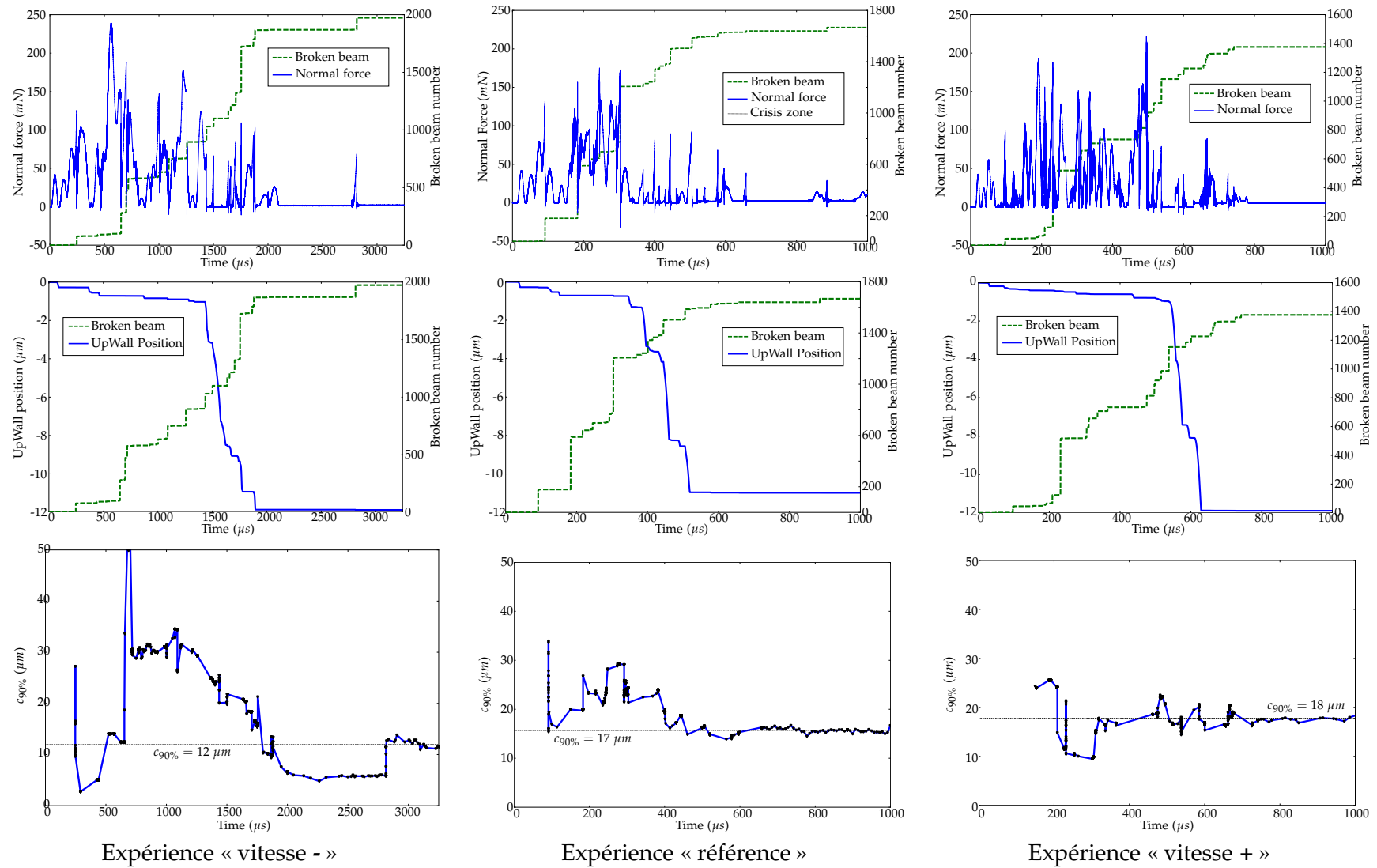


FIGURE 4.23 – Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes conditions de vitesse

4.4.5 Influence de la taille des abrasifs

Les courbes d'évolution des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes tailles des abrasifs sont visibles sur la figure 4.25.

Une première remarque porte sur la faible distance (altitude) parcourue par la paroi supérieure pour l'expérience « rayon - ». Dans ce cas de figure, la taille moyenne des éléments discrets *silice* est très légèrement inférieure (rayon moyen de $4,7 \mu m$) à la taille des abrasifs (rayon de $5 \mu m$). Au cours de la simulation, les particules d'usure ont alors des difficultés à être évacuées. Il subsiste alors un « lit » de particules d'usure. Ce lit de particules réduit les zones de vide entre la surface abrasée et l'outil³ et empêche significativement la descente du bloc de silice.

Une autre remarque porte sur les évolutions des profondeurs maximales de SSD. Il est observé que la dimension des particules abrasives influence les valeurs de $c_{90\%}$, soit :

- d'environ $11 \mu m$ pour l'expérience « rayon - » ;
- d'environ $17 \mu m$ pour l'expérience « référence » ;
- d'environ $22 \mu m$ pour l'expérience « rayon + ».

Aussi, cette tendance est en accord avec les observations expérimentales (voir figure H.3 en annexe H) où il est observé une croissance des profondeurs de SSD avec une augmentation de la taille moyenne des abrasifs. De plus, les valeurs des profondeurs de SSD obtenues numériquement semblent quantitativement en adéquation avec les observations expérimentales comme le montre la figure 4.24. Toutefois, ces résultats sont à prendre avec prudence dans la mesure où les influences des autres paramètres du procédé (précharge et vitesse) ont montré certaines divergences avec les observations expérimentales.

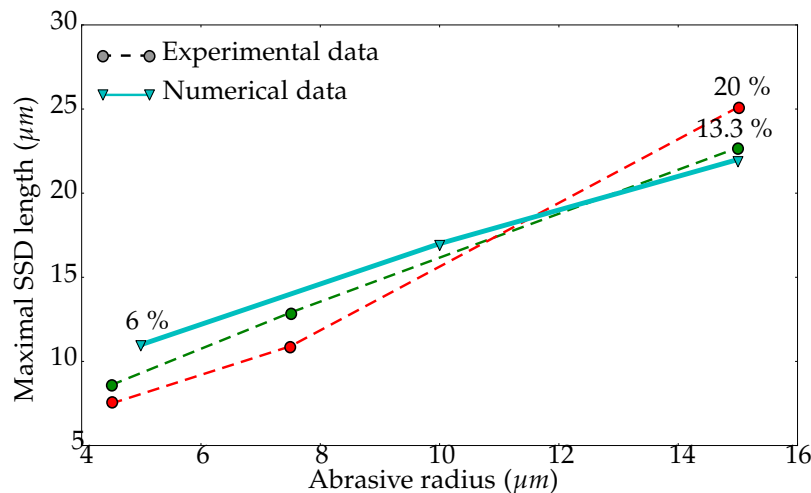


FIGURE 4.24 – Comparaisons des résultats expérimentaux et numériques de l'influence des dimensions des abrasifs pour différentes concentration

La tendance observée peut être la combinaison de deux phénomènes. Tout d'abord, pour obtenir des valeurs similaires de concentration en abrasif, une augmentation des dimensions des abrasifs s'accompagne « mécaniquement » d'une diminution du nombre de ceux-ci. Les nombres des abrasifs pour les expériences

3. Le terme de jeu peut être aussi employé, voir section 4.4.2

« rayon - », « référence » et « rayon + » sont respectivement de 28, 7 et 3. Aussi, il sera vu, au cours de la section suivante (section 4.4.6), qu'une diminution du nombre d'abrasifs entraîne une augmentation de la profondeur des SSD. En outre, cet effet peut également être expliqué par la « rugosité apparente » vue par les abrasifs. En effet, plus la taille des abrasifs augmente, plus la rugosité vue par les abrasifs de la surface abrasée diminue. Ainsi, il est plus aisé pour des abrasifs de petite taille de venir se loger dans une « faille » de rugosité. Ce phénomène est bien connu des utilisateurs de rugosimètre mécanique où il est nécessaire de pondérer la mesure effectuée par la dimension du palpeur utilisé. Aussi, en diminuant la taille des abrasifs, il est probable que les réactions tangentielles (suivant la direction d'avance des abrasifs) générées par les abrasifs soient favorisées. Les efforts tangentiels devenant plus grands, le phénomène d'abrasion est alors privilégié comparativement aux efforts normaux qui sont plus propices à la création de micro-fissures en sous surface par effet d'indentation.

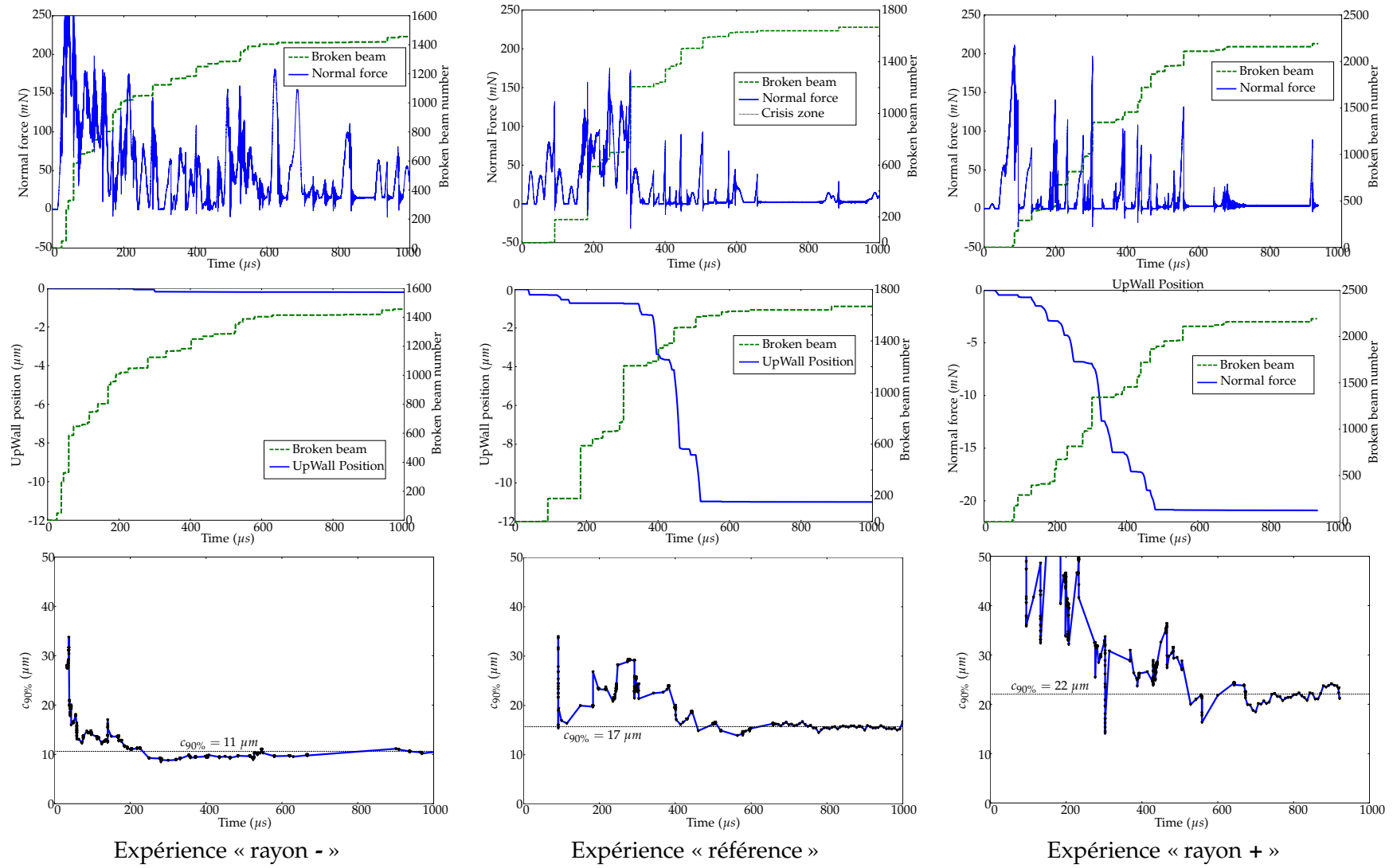


FIGURE 4.25 – Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes dimensions des abrasifs

4.4.6 Influence de la concentration en abrasif

Les courbes d'évolution des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes concentrations en abrasif sont visibles sur la figure 4.27.

En premier lieu, il apparaît que les vitesses d'abrasion diminuent fortement avec l'augmentation de la concentration en abrasif. En second lieu, la taille maximale de SSD semble légèrement diminuer avec l'augmentation du nombre d'abrasifs.

Ces deux tendances peuvent être expliquées par le fait que, si le nombre d'abrasifs augmente, le nombre de contacts entre la silice et les abrasifs augmente également. Cette multiplication des contacts entraîne une meilleure répartition des efforts de réaction sur la surface de la silice et diminue les concentrations de contrainte favorables à la création de SSD et à l'enlèvement de matière.

Cette tendance n'est que partiellement observée expérimentalement. En effet, l'évolution de la profondeur maximale des SSD en fonction de la concentration en abrasif est dépendante de la dimension moyenne des abrasifs. Pour des dimensions (en rayon) des abrasifs comprises, entre 5 et environ 12 μm , il est observé une décroissance puis une croissance pour des dimensions supérieures (voir figure H.5 en annexe). Or, dans les conditions simulées le rayon des abrasifs est de 10 μm . En regard de cette valeur, l'évolution décroissante semble en accord avec les observations expérimentales. Toutefois, en l'état, l'étude paramétrique numérique réalisée permet seulement une comparaison partielle. Il serait nécessaire d'étudier l'influence de la concentration en abrasif en faisant varier également les dimensions de ceux-ci.

En dernier lieu, il apparaît que les résultats numériques sont quantitativement en accord avec les observations expérimentales comme le montre la figure 4.26. Toutefois, ces résultats s'accompagnent de la même remarque que pour la section précédente. Ils sont à prendre avec prudence dans la mesure où les influences des autres paramètres du procédé ont montré des divergences avec les observations expérimentales.

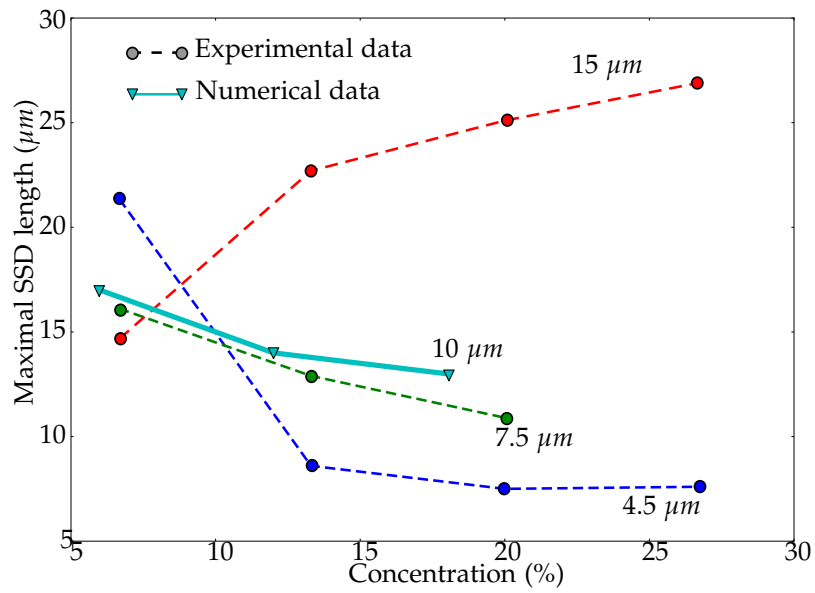


FIGURE 4.26 – Comparaisons des résultats expérimentaux et numériques de l'influence de la concentration en abrasif pour différents rayons d'abrasif

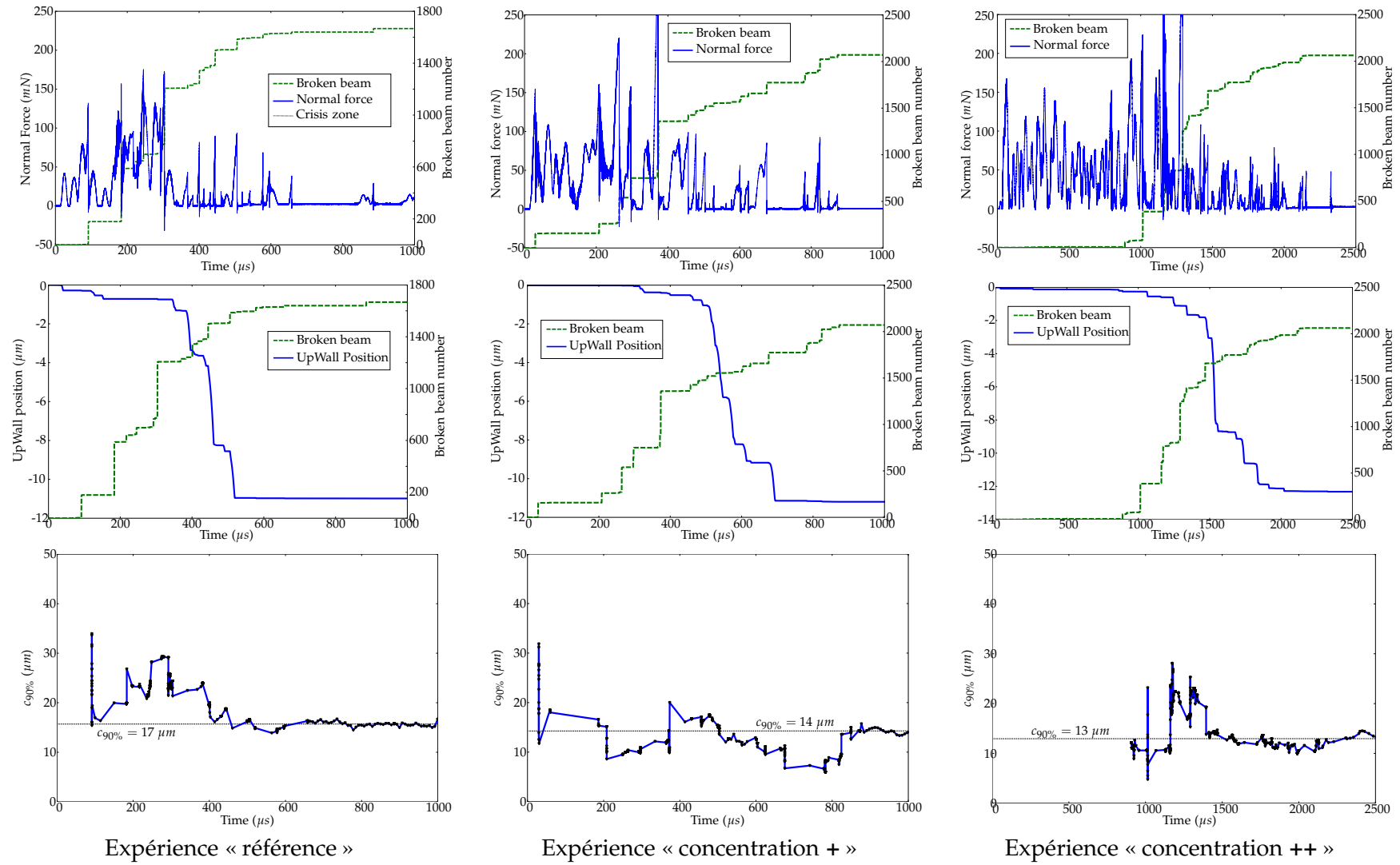


FIGURE 4.27 – Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes concentrations en abrasifs

4.4.7 Analyses des rugosités

Pour chacune des expériences, les évolutions des rugosités de la surface abrasée sont données sur les figures 4.28, 4.29, 4.30 et 4.31. Ces mesures sont obtenues à partir de la surface abrasée interpolée obtenue suivant la méthode décrite en section 4.4.1. Les graphiques montrent les évolutions des rugosités. La rugosité pv correspond à l'écart maximal obtenu, tandis que la rugosité rms correspond à l'écart type des altitudes mesurées. La rugosité rms sera considérée comme un indicateur plus fiable que la rugosité pv .

Il est observé, sur l'ensemble des relevés que, au cours du doucissage numérique, le procédé tend à augmenter la rugosité plutôt que de la diminuer, ce qui est en contradiction avec les observations expérimentales. Il est possible que les temps caractéristiques d'abrasions pour obtenir une surface doucie (de l'ordre de la minute) ne soient pas compatibles avec les temps simulés (de l'ordre de la milliseconde).

De plus, les valeurs de rugosité, de l'ordre de la dizaine de micromètres pour les rugosités rms et de l'ordre de la cinquantaine de micromètres pour les rugosités pv , sont bien supérieures à celles observées expérimentalement après doucissage (de l'ordre de la centaine de nanomètres). Les conditions d'abrasion du modèle numérique semblent donc beaucoup plus sévères que les conditions du doucissage expérimental. Cette différence peut être causée, par l'insuffisance du modèle de l'interface d'abrasion et, plus particulièrement, la modélisation du fluide (voir section 4.4.4). D'autre part, en raison des contraintes inhérentes aux performances de calcul, la discrétisation de la silice paraît insuffisante en regard des valeurs des rugosités expérimentales. Le diamètre moyen des éléments discrets silice étant de l'ordre de la dizaine de micromètres, il paraît peu probable d'obtenir des rugosités inférieures à cette valeur.

Toutefois, il est intéressant de noter une certaine correspondance entre les rugosités rms et les profondeurs de SSD relevées pour chacune des expériences. Le tableau 4.9 dresse un bilan de ces valeurs. Les ratios entre profondeur de SSD $c_{90\%}$ et rugosité rms varient autour d'une valeur moyenne de 1,6 pour un écart type de 0,2. Il paraît donc exister une relation de proportionnalité entre rugosité et profondeur maximale de SSD. Bien que fondée sur aucune proposition théorique, cette relation est également observée expérimentalement. Cette loi empirique, souvent exprimée en fonction de la rugosité pv , est intensément utilisée pour effectuer des mesures indirectes de profondeur de SSD (voir section 1.4.3). Ces derniers résultats doivent être toutefois pondérés au vu des écarts des ordres de grandeur des rugosités obtenus numériquement et celles observées expérimentalement.

Expérience	SSD_{max} $c_{90\%}$ μm	Rugosité rms μm	ratio $\frac{c_{90\%}}{rms}$
référence	17	10,8	1,6
pression -	18	10,2	1,8
pression +	19	10,5	1,8
vitesse -	12	9,0	1,3
vitesse +	18	9,5	1,9
rayon -	11	8,0	1,4
rayon +	22	13,0	1,7
concentration +	14	8,4	1,7
concentration ++	13	9,5	1,4
		moyenne	1,6
		écart type	0,2

TABLE 4.9 – Synthèse des résultats des profondeurs maximales de SSD, des rugosités et de ratios associés

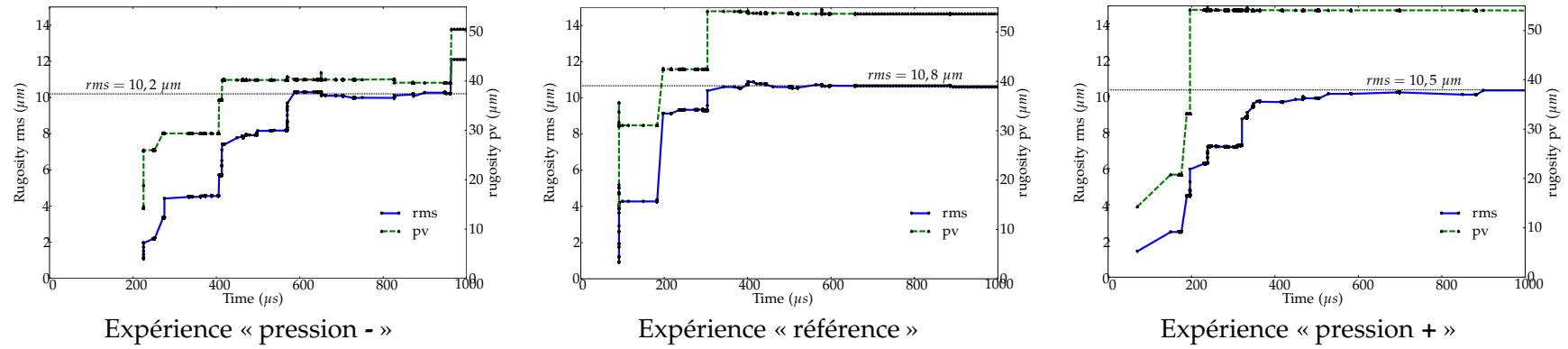


FIGURE 4.28 – Évolutions des rugosités rms pour différents conditions de précharges

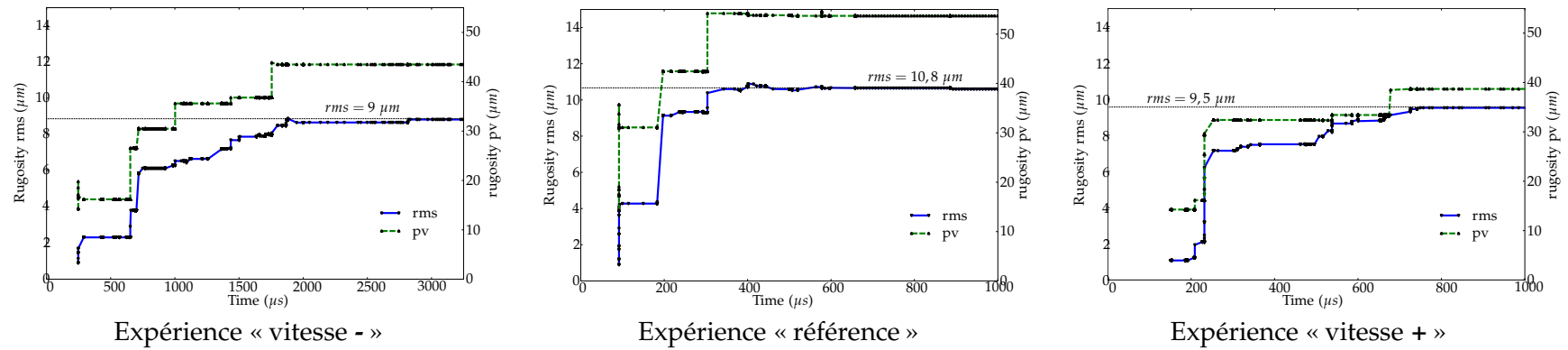


FIGURE 4.29 – Évolutions des rugosités rms pour différentes conditions de vitesses

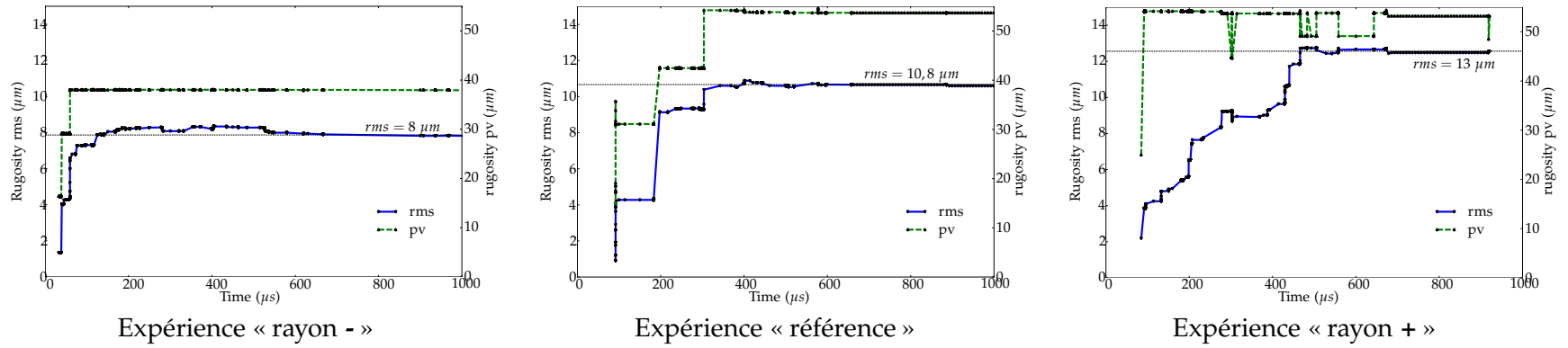


FIGURE 4.30 – Évolutions des rugosités rms pour différentes dimensions des abrasifs

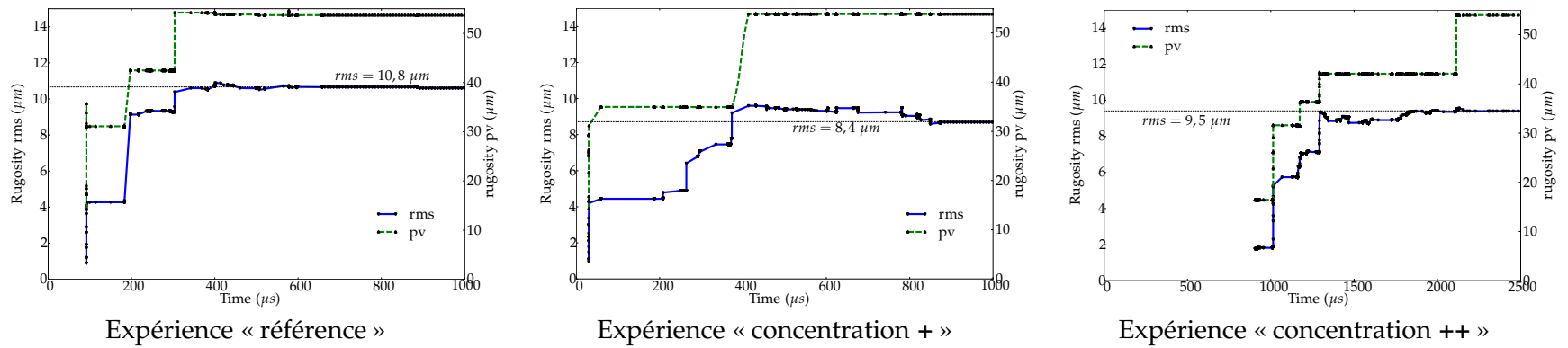


FIGURE 4.31 – Évolutions des rugosités rms pour différentes concentrations en abrasif

4.5 Synthèse et critiques des résultats

Le tableau 4.10 synthétise les tendances observées expérimentalement et numériquement sur la profondeur maximale de SSD en fonction des paramètres du procédé. En raison de la condition de déplacement unidirectionnel, il n'est pas possible d'étudier numériquement l'influence de la précharge. Ce tableau comparatif fait apparaître alors un principal point de divergence entre les résultats numériques et les observations expérimentales concernant l'influence de la vitesse. Cette divergence semble être due à des limitations du modèle numérique et notamment une insuffisance de la prise en compte du fluide à l'interface de doucissage (voir section 4.4.4).

D'autre part, concernant l'influence de la concentration en abrasif, il a été observé expérimentalement que celle-ci ne suit pas une évolution monotone en étant dépendante de la dimension des abrasifs. Il est observé expérimentalement une décroissance des profondeurs maximales de SSD pour des rayons inférieurs à $12\ \mu m$ puis une tendance croissante pour des dimensions supérieures. Dans les conditions simulées le rayon des abrasifs est de $10\ \mu m$. En regard de cette valeur, l'évolution décroissante semble en accord avec les observations expérimentales (voir section 4.4.6). Par contre, il paraît peu probable que le modèle numérique soit en mesure de reproduire la tendance croissante pour des rayons d'abrasifs supérieurs à $12\ \mu m$. En effet, cette évolution décroissante semble contredire le « bon sens physique » : si le nombre d'abrasifs augmente, cela devrait entraîner une multiplication des contacts, et donc, diminuer les pressions locales exercées. Aussi, l'hypothèse d'une unique couche d'abrasif dans l'interface d'abrasion semble être remise en question par ces observations puisqu'une variabilité du nombre de couche rendrait l'affirmation précédente caduque. Il est également possible de se questionner sur l'analyse des observations expérimentales.

À l'échelle du modèle numérique, les paramètres du procédé sont découplés, mais est-ce réellement le cas dans le contexte des études expérimentales ? Au cours de la section dédiée à la description du dispositif expérimental de référence, il a été vu que la vitesse de rotation relative entre l'outil et la pièce est erratique, celle-ci dépendant fortement d'un ensemble de conditions hétérogènes au niveau de l'interface d'abrasion. Or, dans ce cas, une augmentation de la pression ne peut-elle pas avoir d'influence sur la vitesse de rotation relative de l'outil par rapport à la pièce de silice ? Ne peut-il pas en être de même pour la concentration ?

À l'heure actuelle, ces questions sont sans réponse. Toutefois, dans un futur proche la mise en service d'un *Jig* équipé de capteurs devrait permettre de répondre à certaines de ces interrogations. De plus, ces données devraient permettre aussi un recalage plus fin des conditions limites et chargements du modèle numérique. Enfin, il serait possible de statuer sur le bien-fondé de certaines hypothèses : nombre de couches de particules abrasives à l'interface, présence ou non de pic de charge traduisant une condition de déplacement unidirectionnel, présence de phénomènes vibratoires, etc.

Enfin, en analysant les résultats obtenus et, plus particulièrement le tableau 4.9, il apparaît que le phénomène de premier ordre agissant sur la taille maximale des SSD est la dimension des abrasifs. Dans ce cas, l'étude numérique montre quantitativement de bonnes concordances avec les observations expérimentales. Ce phénomène est d'ailleurs mis à profit par les polisseurs en réalisant des opé-

Types	Paramètres	Profondeur de SSD	
		Exp.	Num.
Paramètres machine	Vitesse de rotation ω	$\searrow$	$\nearrow$
	Pression p	$\searrow$	?
Caractéristiques du slurry	Concentration $c\%$	$\searrow \nearrow$	$\searrow$
	Dimension moyenne $\bar{d}$	$\nearrow$	$\nearrow$

Exp. : Résultats expérimentaux

Num. : Résultats numériques

TABLE 4.10 – Synthèse des influences des paramètres du procédé sur la profondeur maximale de SSD suivant les observations expérimentales et les simulations numériques

rations de surfaçage à l'aide d'abrasifs de plus en plus fins. Aussi, même si les influences des autres paramètres n'ont pu être formellement retrouvées, la concordance des résultats pour le phénomène du premier ordre permet de conforter le modèle numérique de doucissage et la démarche réalisée.

4.6 Conclusion

En considérant le doucissage comme un phénomène tribologique, le choix du modèle numérique s'est porté sur la méthode des éléments discrets dont la mise en œuvre se révèle pertinente pour cette gamme de problèmes.

Une des limitations inhérentes à la méthode DEM est le fort coût de calcul de cette méthode ainsi que les contraintes de fenêtre temporelle induites par l'algorithme d'intégration explicite. Dans ces conditions, il n'est pas possible de simuler l'intégralité du dispositif expérimental (spatialement et temporellement).

Aussi, en regard du dispositif expérimental de référence (décrit au cours de la section 4.1), il a été nécessaire de définir les dimensions du domaine d'étude DEM mais aussi une fenêtre temporelle de simulation de manière à respecter des contraintes de temps de calcul tout en essayant de ne pas dégrader la physique du problème.

Le passage de l'expérience macroscopique au modèle DEM microscopique n'est pas sans difficulté. La description du modèle numérique réalisée au cours de la section 4.3 montre qu'en focalisant le problème sur l'interface de doucissage, les mesures expérimentales sont alors inexistantes. Il est nécessaire de procéder à une extrapolation, décrite au cours de la section 4.3.4, des données expérimentales macroscopiques au modèle numérique microscopique. Ce changement d'échelle conduit à élaborer des hypothèses dont la pertinence ne peut pas toujours être vérifiée.

Malgré ces difficultés, certaines tendances expérimentales ont pu être retrouvées (voir section 4.4) : décroissance exponentielle des dommages sous-surfaciques dans l'épaisseur de silice, influence de la dimension des abrasifs ainsi que la relation entre rugosité et profondeur de SSD. Ces résultats ont pu être obtenus grâce à une méthodologie de post-traitement originale et rigoureuse présentée au cours de la section 4.4.1. Par contre, les résultats numériques concernant les influences des vitesses de l'outil et des précharges appliquées n'ont pût être retrouvées. Plusieurs explications sont alors avancées, comme l'influence non négligeable du

fluide, et l'éventuel couplage des paramètres du procédé au niveau macroscopique (voir section 4.5).

Aussi, même si toutes les expériences numériques ne conduisent pas à retrouver les observations expérimentales, l'étendue des résultats obtenus montrent que ceux-ci sont bien induits par les conditions du procédé et ne sont pas la signature unique de l'échantillon numérique de silice.

En conclusion, même si l'outil numérique de simulation du doucissage développé dans le cadre de ce dernier chapitre ne remplit pas pleinement les objectifs du modèle « quantitatif », celui-ci permet de jeter les bases de modèles prenant plus finement en compte les différentes physiques mises en jeu. Des améliorations sont à envisager, dont :

- la modélisation du fluide ;
- le recalage des paramètres expérimentaux de doucissage et, plus particulièrement, l'hypothèse d'une unique couche d'abrasif ;
- la modélisation de la pression normale exercée sur le bloc de silice et la mise en cause de son déplacement unidirectionnel ;
- la mise en œuvre de domaines d'étude plus étendus ;
- la prise en compte de l'effet statistique, en réalisant plusieurs simulations pour un même jeu de paramètres.

Aussi, est-il nécessaire, pour pouvoir envisager les deux dernières propositions, d'augmenter significativement les performances du code de calcul. Des solutions informatiques comme la *parallélisation* du code de calcul *GranOO*, autorisant l'exploitation plusieurs processeurs simultanément, devraient permettre de réaliser ces objectifs.

En outre, les performances de calculs contraignent et limitent la fenêtre temporelle de simulation ainsi que le niveau de discrétisation du problème. Actuellement, ces contraintes empêchent l'étude du taux d'enlèvement matière et de la rugosité avec une précision acceptable. Un accroissement des performances de calcul permettrait donc d'accéder à l'étude de ces paramètres en réalisant des modélisations plus fines du doucissage.

Enfin, la mise en service prochaine d'un dispositif de doucissage, équipé de capteurs de mesure de l'épaisseur du fluide, de la vitesse de rotation du *jig* et de l'effort normal exercé, devrait permettre de se prononcer sur la pertinence de certaines hypothèses ainsi qu'un recalage plus fin des paramètres du modèle en regard de ces nouvelles données expérimentales.

Conclusion générale et perspective

Ce mémoire a permis de présenter les différents travaux portant sur la problématique de *la modélisation par éléments discrets des phases d'ébauchage et de doucissage du verre*.

La première partie de ce mémoire fut consacrée à la présentation de la problématique de tenue aux flux lasers des optiques en silice. Il est admis que la présence de micro-fractures en sous surface, induites par les procédés d'abrasion des composants optiques, jouent un rôle clé dans l'initiation des dommages lasers. Ces micro-fractures sont au cœur des préoccupations de ce mémoire. Une revue bibliographique des travaux expérimentaux, portant sur les relations entre opérations de surfaçage et la génération de micro-fissures, montre que ce phénomène est encore mal connu. Ce premier chapitre a permis également de définir un cadre théorique ainsi que des hypothèses de travail. Ce mémoire s'est intéressé plus particulièrement à l'opération de doucissage où il est considéré que les phénomènes prépondérants agissant sur la génération des micro-fissures sont d'origine mécanique. Puis, une revue des méthodes numériques pouvant être appliquées à cette gamme de problèmes a été entreprise. En raison des similitudes du doucissage avec les problèmes tribologiques, le choix de la méthode numérique s'est porté sur la méthode des éléments discrets *dynamiques explicites* qui s'avère être un outil pertinent pour l'étude de cette gamme de problèmes. Toutefois, la principale difficulté, inhérente à ces méthodes, réside dans l'obtention de résultats quantitatifs. Un effort particulier a été donc entrepris pour proposer un modèle quantitatif de doucissage du verre de silice.

Pour mener à bien ce projet, il a été décidé de refonder le code de calcul DEM *dynamique explicite Gran3D*, orienté vers la résolution de problèmes tribologiques, dans une démarche d'optimisation en termes d'évolutivité, de sécurité et de performance. Les différentes techniques de conception, de programmation, de déploiement et de gestion de projet ont été présentées au cours du deuxième chapitre. L'architecture de cette plate-forme de calcul est basée sur l'exploitation de deux notions fondamentales : les *greffons* et les *conteneurs intelligents*. Ces solutions permettent alors une grande souplesse d'utilisation, mais également d'assurer l'évolutivité et la pérennité du code de calcul.

Cette plateforme de calcul a été ensuite mise à profit pour proposer un modèle quantitatif de silice élastique fragile. Ce modèle, présenté au cours du cha-

pitre 3, repose sur l'utilisation de liaisons cohésives *poutres*. Une méthodologie de calibration des paramètres mécaniques microscopiques en regard des propriétés mécaniques de la silice est alors proposée. Une large gamme d'essais numériques a permis, par la suite, de valider l'approche retenue. Enfin, il a été montré que pour un jeu de paramètres microscopiques, les propriétés mécaniques macroscopiques sont indépendantes du maillage à condition que le domaine initial respecte un certain nombre de critères d'organisation. Ces critères, établis au cours du chapitre 3, ont permis de qualifier l'organisation d'un échantillon numérique selon sa compacité, son homogénéité et son isotropie géométrique.

Le modèle de silice élastique fragile a été ensuite introduit dans le modèle de doucissage présenté au chapitre 4. Afin de vérifier la pertinence des résultats numériques obtenus, il a été proposé de baser le modèle numérique sur un dispositif expérimental de référence, puis de comparer les résultats numériques obtenus avec les observations expérimentales issues de ce dispositif. Les données provenant des simulations ont été analysées suivant une méthodologie de post-traitement permettant de quantifier la rugosité de la surface érodée ainsi que de déterminer pour chaque expérience un profil de répartition des dommages sous-surfaciques dans le substrat de silice. Les résultats numériques ont montré alors une adéquation partielle avec les observations expérimentales. Toutefois, il apparaît que l'effet de la dimension des abrasifs, identifié expérimentalement comme prépondérant, est bien retrouvé numériquement. En outre, la mise en service, très prochainement, d'un dispositif de doucissage instrumenté permettra de connaître les évolutions dynamiques *in situ* de l'épaisseur de l'interface de doucissage, l'évolution de l'effort normal et des vitesses mises en jeux. Ces mesures permettront à la fois une connaissance plus fine du procédé, d'affiner le modèle de surfaçage local développé, et d'arbitrer la validité de certaines hypothèses d'étude, dont le nombre de couches d'abrasif dans l'interface de doucissage.

D'une manière générale l'objectif de modèles quantitatifs est partiellement rempli. S'agissant du modèle de silice élastique fragile, les résultats montrent une bonne adéquation de ceux-ci avec les modèles théoriques. Par contre, le modèle de doucissage semble souffrir de limitations. De façon à s'approcher des conditions réelles, il est envisagé de décrire de façon plus fine l'interface de doucissage. Il s'agit, notamment, d'améliorer la description géométrique des abrasifs en utilisant des formes polyédriques ou des agglomérats d'élément discret permettant une meilleure prise en compte de l'angularité des abrasifs. Un autre aspect, concerne l'amélioration du modèle du fluide abrasif. Pour parvenir à un modèle plus réaliste, il serait possible de coupler au modèle de doucissage, un modèle de résolution des problèmes de mécanique des fluides.

Aussi, la transition entre le modèle de silice élastique fragile et le modèle de doucissage est sans doute trop abrupte. La démarche de calibration des propriétés mécaniques décrites au cours du chapitre 3 donne des résultats satisfaisants concernant les propriétés macroscopiques de la silice : élasticité et rupture. Aussi, le modèle de doucissage développé exploite ces propriétés de façon locale. Notamment, les ruptures sont propagées sur quelques éléments discrets, contrairement aux essais de calibrations où la fissuration s'inscrit sur une section complète de l'éprouvette numérique. Pour s'assurer de la pertinence des propriétés locales à rupture, une étude plus fine peut alors être conduite. Il est envisagé d'introduire des étapes intermédiaires entre le modèle de doucissage et les essais macroscopiques à rupture. Ainsi, une première étape consisterait à développer une

expérience numérique d'indentation pour mieux qualifier la rupture locale de la silice. Des essais numériques de rayage pourraient également être conduits de façon à valider la rupture dans des conditions plus proches du doucissage. Pour mener à bien cette démarche, une collaboration avec des membres du laboratoire LARMAUR¹, dont les thématiques de recherche sont centrées sur l'étude et la qualification expérimentales des propriétés mécaniques du verre, est à l'étude.

Enfin, une limitation générale concerne les performances de calcul. Dans l'optique d'augmenter significativement ceux-ci, il est envisagé de mettre en œuvre des solutions informatiques comme *la parallélisation* du code de calcul de *GranOO* pour permettre d'exploiter pleinement des machines de calculs modernes utilisant les technologies multi-cœurs et multi-processeurs. Cet apport permettra de simuler des échantillons numériques comportant un nombre plus important d'éléments discrets. Les domaines d'études pourraient être de plus grandes dimensions ou bien d'un taux de raffinement plus élevé. Cela permettrait d'améliorer la représentativité des résultats compte tenue des aspects statistiques des phénomènes simulés. D'autre part, des performances de calcul accrues autoriseront des temps de simulation plus longs qui permettront, à terme, de modéliser l'enlèvement de matière.

1. <http://www.larmaur.univ-rennes1.fr>

Annexes

De l'usage des quaternions en mécanique du solide

A.1 Introduction

Les quaternions furent introduits par William Rowan Hamilton en 1843. Il sera décrit brièvement le raisonnement d'Hamilton l'amenant à créer cette entité mathématique.

Il est admis que la transformation d'un point M par un vecteur $\vec{V}$ donne un unique point N (où que deux points M et N définissent un unique vecteur $\vec{V}$).

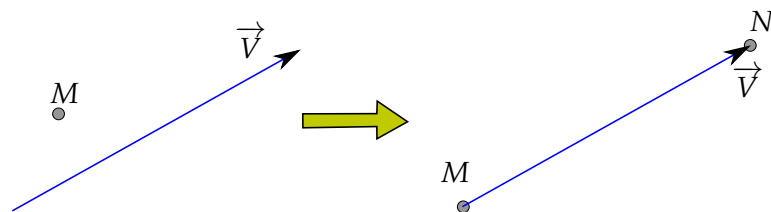


FIGURE A.1 – Vecteur liant deux points

De la même façon Hamilton veut trouver une entité permettant de transformer un vecteur en un autre vecteur unique. Cette entité sera nommée plus tard *quaternion* (cf figure A.2).

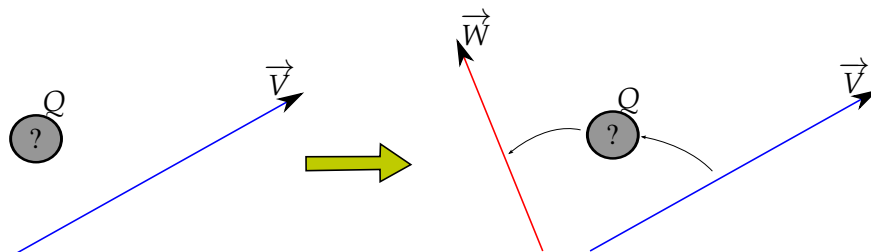


FIGURE A.2 – Quaternion liant deux vecteurs

Un vecteur étant défini par une norme et une direction, la transformé d'un vecteur par rapport à un autre vecteur, implique un changement de norme et un changement de direction. Le quaternion devra donc représenter cette double transformation.

A.2 Changement de norme : nombre scalaire

Tout scalaire peut représenter le ratio entre les normes de deux vecteurs parallèles. Ainsi, un scalaire est un opérateur qui modifie la norme d'un vecteur sans modifier la direction :

$$\vec{A} = s \bullet \vec{B} \text{ avec } \|\vec{A}\| = \|\vec{B}\|$$

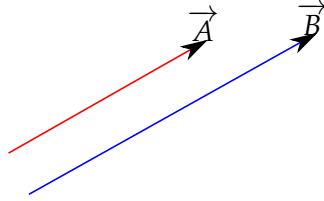


FIGURE A.3 – Définition d'un scalaire

A.3 Changement de direction : versor

Hamilton définit également le versor comme le quotient de deux vecteurs non parallèles mais de mêmes normes. Un versor est donc un opérateur qui modifie la direction d'un vecteur sans modifier la norme :

$$\vec{A} = v \bullet \vec{B}$$

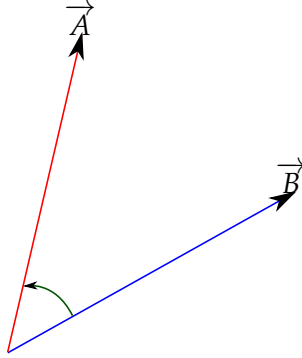


FIGURE A.4 – Définition d'un versor

A.4 Définition d'un quaternion

Hamilton définit donc le quaternion comme le quotient géométrique de deux vecteurs quelconques. Le quaternion est donc un opérateur qui modifie la direction et la norme d'un vecteur :

$$\vec{A} = Q \bullet \vec{B} \tag{A.1}$$

Le quaternion est la combinaison d'un scalaire et d'un versor. Cette combinaison requiert 4 coordonnées indépendantes : une pour le changement de norme (scalaire), et trois pour le changement d'orientation (versor).

A.5 Propriétés mathématiques

Hamilton décrivit un quaternion comme un quadruplet de nombres réels, le premier élément étant la partie scalaire réelle, et les trois autres formant un vecteur imaginaire pur. Tout quaternion Q peut être considéré comme une combinaison linéaire de quatres quaternions "unités" $1, i, j$, et k :

$$Q = q \cdot 1 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k \quad (\text{A.2})$$

Les quaternions peuvent également être écrits sous une forme faisant apparaître la partie vectorielle imaginaire :

$$Q = (q, \vec{q})$$

Dans la suite il sera préféré l'écriture des quaternions sous forme vectorielle.

Hamilton put ainsi définir une multiplication entre quaternions basée sur la multiplication entre les quaternions unités $1, i, j$, et k :

•	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

TABLE A.1 – Table de multiplication des quaternions unités

Cette table de multiplication appelle plusieurs remarques :

- La multiplication entre membres imaginaires est semblable au produit vectoriel ;
- La multiplication entre deux même membres imaginaires donne -1. Un membre imaginaire est comparable à un versor impliquant une rotation de 90° autour de son axe. En appliquant deux fois le même versor le vecteur résultant est l'opposé de celui avant transformation ;
- La matrice de multiplication n'est pas symétrique, la multiplication de deux quaternion n'est donc pas commutative.

A.5.1 Addition

L'addition est commutative et s'exprime par :

$$\begin{aligned} Q_1 &= (q_1, \vec{q}_1) \\ Q_2 &= (q_2, \vec{q}_2) \\ Q_1 + Q_2 &= (q_1 + q_2, \vec{q}_1 + \vec{q}_2) \end{aligned} \quad (\text{A.3})$$

A.5.2 Multiplication

La multiplication est non-commutative et s'exprime par :

$$Q_1 \bullet Q_2 = (q_1 q_2 - \vec{q}_1 \cdot \vec{q}_2, q_1 \vec{q}_2 + q_2 \vec{q}_1 + \vec{q}_1 \wedge \vec{q}_2) \quad (\text{A.4})$$

A.5.3 Forme conjuguée

Le conjugué est obtenu en conservant la partie scalaire et en prenant l'opposé de la partie vectorielle

$$\overline{Q} = (q, -\vec{q}) \quad (\text{A.5})$$

A.5.4 Norme

Le produit d'un quaternion Q par son conjugué $\overline{Q}$ donne un nombre réel positif :

$$Q \bullet \overline{Q} = q^2 + \|\vec{q}\|^2 \quad (\text{A.6})$$

La norme d'un quaternion est donnée par :

$$\|Q\| = \sqrt{Q \bullet \overline{Q}} = \sqrt{q^2 + \|\vec{q}\|^2} \quad (\text{A.7})$$

A.5.5 Inverse

Si un quaternion n'est pas nul, il possède un unique inverse :

$$Q^{-1} = \frac{\overline{Q}}{\|Q\|^2} \quad (\text{A.8})$$

A.5.6 Quelques opérations remarquables

$$\overline{(\overline{Q})} = Q \quad (\text{A.9})$$

$$\overline{(Q_1 + Q_2)} = \overline{Q_1} + \overline{Q_2} \quad (\text{A.10})$$

$$\overline{(Q_1 \bullet Q_2)} = \overline{Q_2} \bullet \overline{Q_1} \quad (\text{A.11})$$

$$(Q_1 \bullet Q_2)^{-1} = Q_2^{-1} \bullet Q_1^{-1} \quad (\text{A.12})$$

$$\frac{1}{2} (Q_2 \bullet Q_1 + Q_1 \bullet Q_2) = (q_2 q_1 - \vec{q}_2 \vec{q}_1, q_2 \vec{q}_1 + q_1 \vec{q}_2) \quad (\text{A.13})$$

$$\frac{1}{2} (Q_2 \bullet Q_1 - Q_1 \bullet Q_2) = (0, \vec{q}_2 \wedge \vec{q}_1) \quad (\text{A.14})$$

A.5.7 Quaternions unitaires et forme polaire

Les quaternions unitaires sont, par définition, les quaternions de norme un. Un quaternion est unitaire si, et seulement si, $Q^{-1} = \overline{Q}$. Tout quaternion peut s'écrire sous une forme polaire :

$$Q = (\rho \cos \theta, \rho \sin \theta \vec{u}) \quad (\text{A.15})$$

Où :

- ρ est un réel positif égal à la norme du quaternion : $\rho = \|Q\|$;
- $\vec{u}$ est un vecteur unitaire : $\|\vec{u}\| = 1$.

Si $\rho = 1$ le quaternion est unitaire et s'écrit :

$$Q = (\cos \theta, \sin \theta \vec{u}) \quad (\text{A.16})$$

Ces derniers permettent d'exprimer une rotation vectorielle.

A.6 Quaternion et rotation

A.6.1 Démonstration

Supposons l'opération suivante :

$$V' = N \bullet V \bullet N \quad (\text{A.17})$$

Où :

- V' est un quaternion quelconque : $V' = (v', \vec{v}')$;
- N est un quaternion pur : $N = (0, \vec{n})$;
- V est un quaternion pur : $V = (0, \vec{v})$.

En appliquant les règles de multiplication des quaternions, il est montré que :

$$v' = 0 \quad (\text{A.18})$$

$$\vec{v}' = (\vec{n} \wedge \vec{v}) \wedge \vec{n} - \vec{n}(\vec{n} \cdot \vec{v}) = \vec{v} - 2\vec{n}(\vec{n} \cdot \vec{v}) \quad (\text{A.19})$$

Les résultats ci-dessus montre que si $\vec{n}$ est un vecteur unité alors le vecteur $\vec{v}'$ est la symétrie du vecteur $\vec{v}$ dans le plan normal à $\vec{n}$. L'opération $V' = N \bullet V \bullet N$ exprime donc une symétrie. Or une rotation vectorielle peut être représentée comme le produit de deux symétries successives [22, §4].

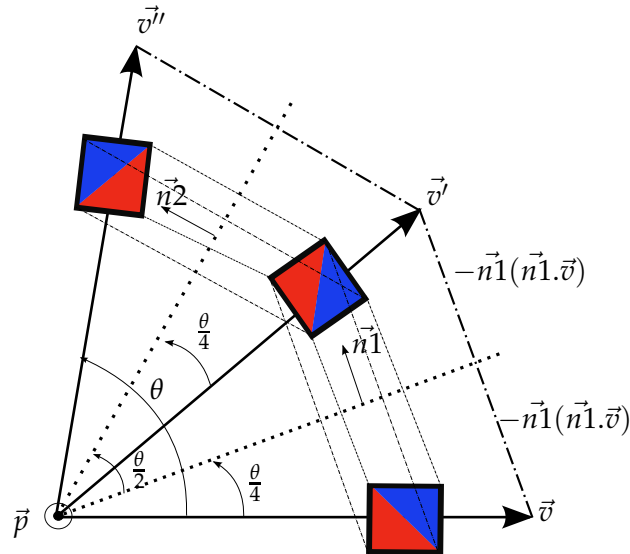


FIGURE A.5 – Rotation comme produit de deux symétries

La figure ci-dessus met en évidence ce théorème pour un espace à deux dimensions. Le raisonnement est également valable en trois dimensions. Une rotation de θ autour du vecteur $\vec{p}$ est équivalente à deux symétries successives dont les plans sont orientés suivant un angle de $\frac{\theta}{2}$. L'orientation de $\vec{p}$ est définie par l'intersection des plans de symétrie. Cette opération peut donc être écrite en utilisant les quaternions :

$$\left. \begin{aligned} V' &= N_1 \bullet V \bullet N_1 \\ V'' &= N_2 \bullet V' \bullet N_2 \end{aligned} \right\} V'' = N_2 \bullet N_1 V \bullet N_1 \bullet N_2$$

Or en développant cette dernière égalité, il est montré que pour $||\vec{p}'|| = 1$:

$$\left. \begin{aligned} N_2 \bullet N_1 &= (\cos \frac{\theta}{2}, \vec{p} \sin \frac{\theta}{2}) = Q \\ N_1 \bullet N_2 &= \overline{Q} \end{aligned} \right\} V'' = Q V \overline{Q}$$

Où V'' et V sont des quaternions purs.

Il a été démontré que la rotation vectorielle d'un vecteur quelconque $\vec{v}$ d'un angle θ autour d'un vecteur $\vec{p}$ unitaire peut être obtenue grâce au quaternion unitaire $Q(\cos \frac{\theta}{2}, \vec{p} \sin \frac{\theta}{2})$ selon :

$$(0, \vec{v}') = Q \bullet (0, \vec{v}) \bullet \overline{Q}$$

Par abus de notation, le quaternion pur $(0, \vec{v})$ sera confondu avec le vecteur $\vec{v}$. L'équation précédente devient :

$$\boxed{\vec{v}' = Q \bullet \vec{v} \bullet \overline{Q}} \quad (\text{A.20})$$

En conclusion, il est possible d'exprimer des changements de base à l'aide du quaternion unitaire polaire $Q(\cos \frac{\theta}{2}, -\vec{p} \sin \frac{\theta}{2})$ où :

- le vecteur $\vec{p}$ est unitaire et désigne l'axe de rotation ;
- θ est l'angle de rotation autour de $\vec{p}$.

A.6.2 Relation entre quaternions et matrice de passage

La relation entre quaternion et matrice de passage est donnée par [22, §6] :

$$\overline{A} = 2 \begin{pmatrix} \frac{q_0^2 + q_1^2 + q_2^2 + q_3^2}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & \frac{q_0^2 - q_1^2 + q_2^2 - q_3^2}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & \frac{q_0^2 - q_1^2 - q_2^2 + q_3^2}{2} \end{pmatrix} \quad (\text{A.21})$$

A.7 Quaternion et vitesse de rotation

Soit un solide S_1 repéré par le repère $R_1(0_1, \vec{x}_1, \vec{y}_1, \vec{z}_1)$ en mouvement par rapport à un repère $R_0(0_0, \vec{x}_0, \vec{y}_0, \vec{z}_0)$. La formule de la base mobile donne :

$$\frac{d\vec{x}_1}{dt} \Big|_{R_0} = \omega_{R_1/R_0} \wedge \vec{x}_1 \quad (\text{A.22})$$

Il est possible d'exprimer cette égalité à l'aide de quaternions purs :

$$\dot{\vec{x}}_1 = \vec{\omega} \wedge \vec{x}_1 \iff (0, \dot{\vec{x}}_1) = (0, \vec{\omega} \wedge \vec{x}_1)$$

Finalement le changement de base peut également être exprimé grâce aux

quaternions :

$$\begin{aligned}(0, \vec{x}_1) &= Q \bullet (0, \vec{x}_0) \bullet \bar{Q} \\ \frac{d}{dt}(0, \vec{x}_1) &= \frac{d}{dt} [Q \bullet (0, \vec{x}_0) \bullet \bar{Q}] \\ (0, \dot{\vec{x}}_1) &= \dot{Q} \bullet (0, \vec{x}_0) \bullet \bar{Q} + Q \bullet (0, \dot{\vec{x}}_0) \bullet \bar{Q} + Q \bullet (0, \vec{x}_0) \bullet \dot{\bar{Q}}\end{aligned}$$

Or $\dot{\vec{x}}_0 = \vec{0}$ car R_0 est fixe :

$$(0, \dot{\vec{x}}_1) = \dot{Q} \bullet (0, \vec{x}_0) \bullet \bar{Q} + Q \bullet (0, \vec{x}_0) \bullet \dot{\bar{Q}}$$

Or $(0, \vec{x}_0) = \bar{Q} \bullet (0, \vec{x}_1) \bullet Q$:

$$(0, \dot{\vec{x}}_1) = \dot{Q} \bullet \bar{Q} \bullet (0, \vec{x}_1) \bullet Q \bullet \bar{Q} + Q \bullet \bar{Q} \bullet (0, \vec{x}_1) \bullet Q \bullet \dot{\bar{Q}}$$

Or $Q \bullet \bar{Q} = 1$:

$$(0, \dot{\vec{x}}_1) = \dot{Q} \bullet \bar{Q} \bullet (0, \vec{x}_1) + (0, \vec{x}_1) \bullet Q \bullet \dot{\bar{Q}}$$

Et comme Q est unitaire :

$$Q \bullet \dot{\bar{Q}} = -\overline{\dot{Q} \bullet \bar{Q}} = -\dot{Q} \bullet \bar{Q}$$

D'où :

$$(0, \dot{\vec{x}}_1) = \dot{Q} \bullet \bar{Q} \bullet (0, \vec{x}_1) - (0, \vec{x}_1) \bullet \dot{Q} \bullet \bar{Q}$$

Or la forme $Q_1 \bullet Q_2 - Q_2 \bullet Q_1$ est dite antisymétrique et est égale à :

$$2(0, \vec{p} \wedge \vec{n})$$

En posant :

$$P = \dot{Q} \bullet \bar{Q} \text{ et } N = (0, \vec{x}_1)$$

Est obtenu :

$$(0, \dot{\vec{x}}_1) = 2(0, \vec{p} \wedge \vec{x}_1)$$

Or d'après la formule de la base mobile $(0, \dot{\vec{x}}_1) = (0, \vec{\omega} \wedge \vec{x}_1)$. Il est donc possible d'en déduire :

$$\vec{p} = \frac{1}{2}\vec{\omega} \Rightarrow P = \frac{1}{2}(0, \vec{\omega}) \Rightarrow \dot{Q} \bullet \bar{Q} = \frac{1}{2}(0, \vec{\omega}) \Rightarrow \dot{Q} = \frac{1}{2}(0, \vec{\omega}) \bullet Q$$

Ce qui donne finalement :

$$\boxed{\dot{Q} \bullet \bar{Q} = \frac{1}{2}(0, \vec{\omega})} \quad (\text{A.23})$$

$$\boxed{\dot{Q} = \frac{1}{2}(0, \vec{\omega}) \bullet Q} \quad (\text{A.24})$$

$$\boxed{\ddot{Q} = \frac{1}{2}(0, \dot{\vec{\omega}}) \bullet Q + \frac{1}{2}(0, \vec{\omega}) \bullet \dot{Q}} \quad (\text{A.25})$$

A.8 Application en dynamique du solide

Soit un solide S repéré par $R_1 (G, \vec{x}_1, \vec{y}_1, \vec{z}_1)$ en mouvement par rapport à un repère galiléen $R_0 (O, \vec{x}_0, \vec{y}_0, \vec{z}_0)$. G est le centre de masse de S . Les équations d'Euler-Newton donnent :

$$I\vec{\omega} + \vec{\omega} \wedge I\vec{\omega} = \vec{\tau} \quad (\text{A.26})$$

Où :

- I est la matrice d'inertie de S exprimé dans le repère $R_1 (G, \vec{x}_1, \vec{y}_1, \vec{z}_1)$;
- $\vec{\omega}$ est la vitesse de rotation du solide S par rapport à R_0 exprimé dans le repère $R_1 (G, \vec{x}_1, \vec{y}_1, \vec{z}_1)$;
- $\vec{\tau}$ est la somme des moments extérieurs appliqués sur le solide au point G , exprimé dans la base $R_1 (G, \vec{x}_1, \vec{y}_1, \vec{z}_1)$;

Cette dernière équation permet d'exprimer $\vec{\omega}$:

$$\vec{\omega} = I^{-1} (\vec{\tau} - \vec{\omega} \wedge I\vec{\omega})$$

En utilisant les égalités A.24 et A.25, il est possible d'écrire :

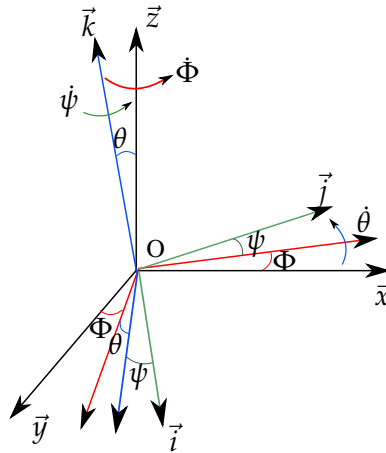
$$\ddot{q} = \dot{q} \bullet \bar{q} \bullet \dot{q} + \frac{1}{2} \dot{q} \bullet \left[I^{-1} (\vec{\tau} - 4 (\bar{q} \bullet \dot{q}) \wedge I. (\bar{q} \bullet \dot{q})) \right] \quad (\text{A.27})$$

Où :

- $q, \dot{q}, \ddot{q}$ représentent respectivement les orientations, vitesses, accélérations angulaires du solide ;
- le quaternion pur $\bar{q} \bullet \dot{q}$ est assimilé par abus de notation à sa partie vectorielle.

A.8.1 Quaternion et angles d'Euler

La figure ci-dessous représente un changement de base paramétré selon les 3 angles d'Euler.



Suivant cette configuration, il est alors possible de définir la matrice de passage [105] :

$$M = \begin{pmatrix} c\psi.c\theta.c\Phi - s\psi.s\Phi & c\psi.c\theta.s\Phi + s\psi.c\Phi & -c\psi.s\theta \\ -s\psi.c\theta.c\Phi - c\psi.s\Phi & -s\psi.c\theta.s\Phi - c\psi.c\Phi & s\psi.s\theta \\ s\theta - c\Phi & s\theta - s\Phi & c\theta \end{pmatrix}$$

Par commodité de lecture c représente le cosinus et s le sinus.

A.8.2 Singularité des angles d'Euler

Une singularité [105, 26] est observée lorsque $\theta = K.\pi$ où K est un nombre entier. Cette singularité s'exprime mathématiquement par :

$$M_s = \begin{pmatrix} c\psi.c\Phi - s\psi.s\Phi & c\psi.s\Phi + s\psi.c\Phi & 0 \\ -s\psi.c\Phi - c\psi.s\Phi & -s\psi.s\Phi + c\psi.c\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

La rotation induite par M_s est une rotation unique par rapport à l'axe $(O, \vec{z})$. Les angles ψ et Φ ont donc la même conséquence du point de vue cinématique. Cette singularité est aussi connue sous le nom de *Gimbal Lock*, la littérature explique ce phénomène par une perte d'un degré de liberté.

Cette singularité s'exprime encore plus fortement lorsque apparaissent les vitesses de rotation :

$$\begin{aligned} \dot{\Phi} &= -\omega_x \frac{\cos \Psi}{\sin \theta} + \omega_y \frac{\sin \Psi}{\sin \theta} \\ \dot{\theta} &= \omega_x \sin \Psi + \omega_y \cos \Psi \\ \dot{\Psi} &= \omega_x \frac{\cos \Psi \cdot \cos \theta}{\sin \theta} - \omega_y \frac{\sin \Psi \cdot \cos \theta}{\sin \theta} + \omega_z \end{aligned}$$

Le calcul des dérivés par rapport au temps des angles d'Euler devient problématique pour $\theta \rightarrow K.\pi$. En effet, le terme $\frac{1}{\sin \theta}$ diverge et les erreurs numériques deviennent non négligeables.

Un avantage reconnu des quaternions [26] est d'éviter cette singularité car la position angulaire d'un solide est paramétrée non pas par trois angles mais par un axe de rotation et un angle.

A.9 Synthèse

Au cours de cette section a été introduit l'outil mathématique *quaternion*. Son application en dynamique du solide a également été introduite. Les *quaternions* présentent de nombreux avantages pour des applications numériques. Ils permettent une grande rapidité de calcul en évitant l'utilisation des fonctions trigonométriques sinus et cosinus et ne souffrent pas de singularité pouvant amener à la divergence du schéma numérique. Le seul inconvénient est la renormalisation. Cette étape est nécessaire pour éliminer une possible dérive des résultats numériques lors de l'intégration [105, §2.5].

Description du fichier d'entrée : Tension.inp

L'organisation générale d'un fichier d'entrée GranOO permet de décrire suivant la syntaxe XML une expérience numérique DEM. Ainsi, un fichier d'entrée permet de définir :

1. les paramètres généraux de calcul : le nombre d'itération, pas d'intégration, etc. ;
2. le domaine discret à lire ;
3. la séquence de greffons à exécuter en phase de *preprocessing* ;
4. la séquence de greffons à exécuter en phase de *processing* ;
5. la séquence de greffons à exécuter en phase de *postprocessing* ;
6. la description de conditions limites et de chargements usuels.

Cette section propose une vue détaillée de ces étapes au travers d'un exemple de simulation DEM : un essai de traction où l'éprouvette est encastree à l'une de ses extrémités et chargée à l'autre extrémité.

B.1 Paramètres généraux de calcul

Les paramètres généraux sont décrits au travers de la balise `ComputeProblem` :

```
<ComputeProblem TotIteration="10000" OutDir="Results" TimeStep="1e-10"/>
```

- L'attribut `TotIteration` permet de spécifier le nombre total d'itérations de calcul ;
- L'attribut `OutDir` permet de spécifier le chemin relatif ou absolu du dossier ou seront stockés les différents résultats issus de la simulation ;
- L'attribut `TimeStep` permet de spécifier le pas d'intégration.

B.2 Lecture du domaine discret

La lecture d'un domaine discret est réalisée à l'aide de la balise `SampleFile` :

```
<SampleFile File="cylinder.gdd" />
```

L'attribut `File` permet de spécifier le chemin relatif ou absolu du fichier `gdd` à lire. Dans le cadre de l'essai de traction, le fichier `cylinder.gdd` correspond à une éprouvette cylindrique (voir figure 2.16).

B.3 Les greffons de *pre-processing*

La séquence de greffons à exécuter avant la boucle temporelle est spécifiée à l'aide de la balise `PreProcessing`.

```
<PreProcessing>
  <PlugIn Id="ConvertBondToBeam3D" Bond3DSet="Global"
    YoungModulus="10e9" RadiusRatio="0.6" PoissonRatio="0.3" FailureStress="100e9"/
  >
  <PlugIn Id="ComputeOptimalTimeStep3D" Ratio="0.14" />
</PreProcessing>
```

- Le greffon `ConvertBondToBeam3D` permet d'affecter des valeurs aux propriétés mécaniques des liaisons cohésives de type `Beam3D`;
- Le greffon `ComputeOptimalTimeStep3D` permet de calculer le pas d'intégration optimal en fonction des propriétés mécaniques et massiques des éléments discrets et des liaisons cohésives.

B.4 Les greffons de *processing*

La séquence de greffons à exécuter durant la boucle temporelle est spécifiée à l'aide de la balise `Processing`.

```
<Processing>
  <PlugIn Id="Check3D" />
  <PlugIn Id="ResetLoad3D" />
  <PlugIn Id="ApplyLoad3D" />
  <PlugIn Id="ApplyBondLoad3D" ComputeDeformationEnergy="yes"/>
  <PlugIn Id="IntegrateAccelerationLinear3D" />
  <PlugIn Id="IntegrateAccelerationAngular3D"/>
  <PlugIn Id="ApplyBoundaryCondition3D" />
  <PlugIn Id="SaveDomain3D" IterLoop="100"/>
</Processing>
```

- Le greffon `Check3D` balaye différents paramètres clés du calcul DEM et vérifie leurs cohérences. En cas d'erreur le greffon arrête le déroulement du calcul;
- Le greffon `ResetLoad3D` annule les efforts exercés sur les éléments discrets;
- Le greffon `ApplyLoad3D` permet de mettre à jour puis d'appliquer les chargements externes sur les éléments discrets;
- Le greffon `ApplyBondLoad3D` permet de mettre à jour puis d'appliquer les chargements exercés par les liaisons cohésives sur les éléments discrets;
- Le greffon `IntegrateAccelerationLinear3D` applique le schéma d'intégration de *verlet vitesse* sur les vitesses et positions linéaires;
- Le greffon `IntegrateAccelerationAngular3D` applique le schéma d'intégration de *verlet vitesse* sur les vitesses et positions angulaires;
- Le greffon `ApplyBoundaryCondition3D` met à jour les conditions limites;

- Le greffon SaveDomain3D permet de sauvegarder le domaine discret sous la forme d'un fichier gdd. L'attribut IterLoop permet d'exécuter ce greffon toutes les 100 itérations de calcul.

B.5 Les greffons de *post-processing*

La séquence de greffons à exécuter après la boucle temporelle est spécifiée à l'aide de la balise PostProcessing.

```
<PostProcessing>
  <PlugIn Id="CalculusInfo" />
</PostProcessing>
```

Le greffon CalculusInfo permet une vue synthétique des propriétés informatique du calcul : temps total de calcul, système d'exploitation, version du compilateur, etc.

B.6 Définition de chargements usuels

Les chargement ne sont appliqués que si le greffon ApplyLoad3D est exécuté et sont définis grâce à la balise Load. Ils sont appliqués sur un ensemble d'éléments discrets (SetOf). Il est possible de définir des cas de chargement variable à l'aide de la balise MathFunction.

```
<MathFunction Id="TensionRamp">
  <RampAndConstant Limit="1000." Constant="1.e9" VariableRef="Iteration"/>
</MathFunction>

<Load Id="TensionRampLoading" DiscreteElement3DSetId="Cylinder-xMax">
  <TotalForce>
    <Vector3D X="TensionRamp" Y="0." Z="0." />
  </TotalForce>
</Load>
```

- La balise RampAndConstant permet de construire une fonction mathématique composée d'une affine puis d'une constante. Cette fonction permet d'atteindre de façon progressive une valeur ;
- La balise TotalForce permet de déclarer un chargement s'appliquant sur l'ensemble d'éléments discrets Cylinder-xMax. La spécificité de se chargement est de répartir de façon équivalente sur chaque élément discret la force totale spécifiée ;
- La balise Vector3D permet de construire un vecteur force. Il est possible d'attribuer aux composantes X, Y et Z une valeur ou bien une fonction mathématique. Dans ce cas, le vecteur est suivant l'axe $\vec{X}$ et sa norme est définie par la fonction mathématique TensionRamp.

B.7 Définition de conditions limites usuelles

Les conditions limites ne sont appliquées que si le greffon ApplyBoundaryCondition3D est exécuté et sont définies grâce à la balise BoundaryCondition.

```
<BoundaryCondition Id="Fix" DiscreteElement3DSetId="Cylinder-xMin">
```

```

<Displacement>
  <Vector3D X="0." Y="0." Z="0." />
  <Quaternion3D Qx="0." Qy="0." Qz="0." Qreal="1.0" />
</Displacement>
</BoundaryCondition>

```

Les différentes conditions limites déclarées ci-dessus s'appliquent aux déplacements de l'ensemble d'éléments discrets Cylinder-xMin. Il s'agit de simuler, dans le cas de la traction, un encastrement de l'une des extrémités. Ainsi :

- La balise Vector3D permet d'affecter des vecteurs déplacements aux éléments discrets concernés. Dans le cas présent, celui-ci est nul ;
- La balise Quaternion3D permet d'affecter des rotations aux éléments discrets concernés. Dans ce cas, celles-ci sont fixées à leurs orientations initiales.

B.8 Vue de l'ensemble du fichier Tension.inp

```

<!DOCTYPE Gran00 SYSTEM "Gran00.dtd">
<Gran00 Version="1.0">

  <ComputeProblem TotIteration="10000" OutDir="Results"/>
  <SampleFile File="cylinder.gdd" />

  <PreProcessing>
    <PlugIn Id="ConvertBondToBeam3D" Bond3DSet="Global"
      YoungModulus="10e9" RadiusRatio="0.6" PoissonRatio="0.3" FailureStress="100e9"/>
    <PlugIn Id="ComputeOptimalTimeStep3D" Ratio="0.14" />
  </PreProcessing>

  <Processing>
    <PlugIn Id="Check3D" />
    <PlugIn Id="ResetLoad3D" />
    <PlugIn Id="ApplyLoad3D" />
    <PlugIn Id="ApplyBondLoad3D" ComputeDeformationEnergy="yes"/>
    <PlugIn Id="IntegrateAccelerationLinear3D" />
    <PlugIn Id="IntegrateAccelerationAngular3D"/>
    <PlugIn Id="ApplyBoundaryCondition3D" />
    <PlugIn Id="UpdateDiscreteShape3D" IterLoop="100"/>
    <PlugIn Id="SaveDomain3D" IterLoop="100"/>
  </Processing>

  <PostProcessing>
    <PlugIn Id="CalculusInfo" />
  </PostProcessing>

  <!-- Loading, normal force on X axis -->
  <MathFunction Id="TensionRamp">
    <RampAndConstant Limit="1000." Constant="1.e9" VariableRef="Iteration"/>
  </MathFunction>

  <Load Id="TensionRampLoading" DiscreteElement3DSetId="Cylinder-xMax">
    <TotalForce>
      <Vector3D X="TensionRamp" Y="0." Z="0." />
    </TotalForce>
  </Load>

```

```

<!--Boundary Conditions, the left face is fixed -->
<BoundaryCondition Id="Fix" DiscreteElement3DSetId="Cylinder-xMin">
  <Displacement>
    <Vector3D X="0." Y="0." Z="0." />
  </Displacement>
</BoundaryCondition>

<BoundaryCondition Id="FixRot" DiscreteElement3DSetId="Cylinder-xMin">
  <Displacement>
    <Quaternion3D Qx="0." Qy="0." Qz="0." Qreal="1.0" />
  </Displacement>
</BoundaryCondition>

</GranOO>

```

B.9 Synthèse

Ce format d'entrée permet à l'utilisateur d'organiser et de paramétrer son expérience numérique DEM. L'utilisation du format standard XML permet une écriture et une lecture intuitive proche des concepts de l'OO. Pour une description plus détaillée des fonctionnalités offertes par cette interface, le lecteur peut consulter la documentation officielle de GranOO dont :

- la description des greffons fournis par défaut¹ ;
- la description du fichier d'entrée et des balises XML supportées par le fichier d'entrée².

1. <http://plateformesn-m2p.ensam.eu/SphinxDoc/granoo/Docs/Plugins.pdf>

2. <http://plateformesn-m2p.ensam.eu/SphinxDoc/granoo/Docs/XmlTags.pdf>

Modélisation et implémentation du modèle de liaison cohésive *poutre*

C.1 Modélisation

Le modèle de liaison cohésive étudié est de type « poutre ». Les hypothèses générales de la résistance des matériaux seront donc retenues :

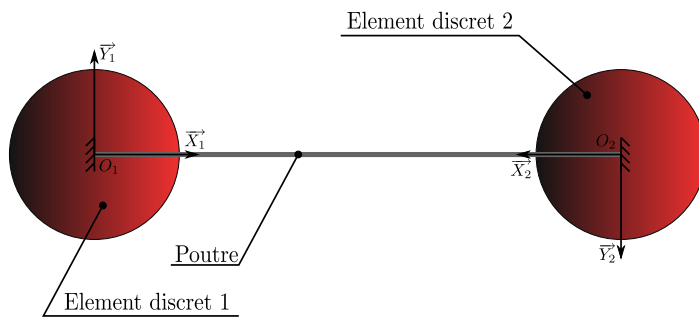
- Le matériau est supposé continu, homogène et isotrope ;
- Les dimensions longitudinales sont supposées grandes devant les dimensions transversales ;
- La géométrie est supposée cylindrique ;
- Les sections droites restent droites après déformation ;
- Les déformations sont supposées très faibles ;
- Les effets massiques des poutres sont négligés, le problème sera supposé statique.

Le principe de superposition sera appliqué afin de décomposer le problème en sollicitation simple de flexion, de traction-compression et de torsion.

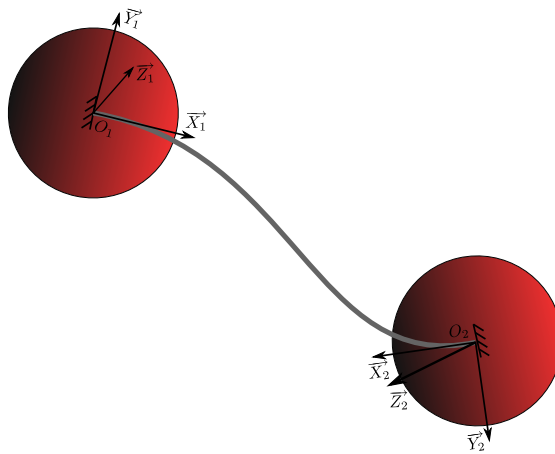
C.1.1 Paramétrage du problème

Soit une poutre cylindrique de rayon r , de longueur L , de section $s = \pi r^2$ encastrée à son extrémité à deux éléments discrets $de1$ et $de2$ (cf figure C.1a). Soit un point G d'abscisse x appartenant à la ligne moyenne de la poutre tel que $\overrightarrow{O_1G} = x \cdot \vec{X}$. Soit les moments quadratiques définis au point G selon les axes $\vec{x}$ et $\vec{y}$ $I_{Gz} = I_{Gy} = \frac{\pi d^4}{64}$. À l'instant initial, la poutre n'est pas sollicitée. Une déformation de celle-ci peut être observée au cours de la simulation comme l'illustre la figure C.1b.

Les repères $(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$ et $(O_2, \vec{X}_2, \vec{Y}_2, \vec{Z}_2)$ sont respectivement liés aux éléments discrets $de1$ et $de2$ (cf figure C.2). À chaque pas de temps un repère de travail $(O_1, \vec{X}, \vec{Y}, \vec{Z})$ est construit de façon que l'axe $(O_1, \vec{X})$ soit coïncidant à la droite (O_1, O_2) . Les actions mécaniques transitant par les liaisons encastres sont traduites en terme d'efforts résultants $\vec{R}_1, \vec{R}_2$, et de moments $\vec{M}_1, \vec{M}_2$ appliqués aux centres des éléments discrets O_1 et O_2 . Les vecteurs rotations $\vec{\theta}_1 = \theta_{1x} \vec{X} + \theta_{1y} \vec{Y} + \theta_{1z} \vec{Z}$ et $\vec{\theta}_2 = \theta_{2x} \vec{X} + \theta_{2y} \vec{Y} + \theta_{2z} \vec{Z}$ traduisent les rotations

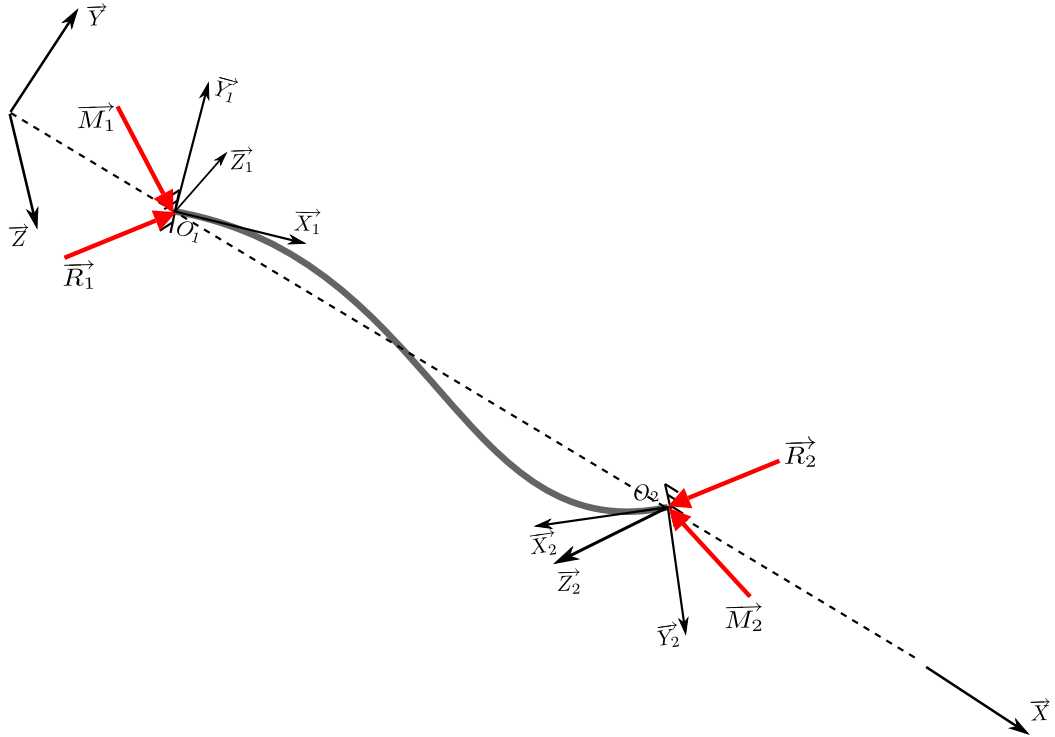


(a) État non sollicité



(b) État sollicité

FIGURE C.1 – Modèle de liaisons cohésive « poutre »

FIGURE C.2 – Modèle de liaison cohésive *poutre*, bilan des actions mécaniques

des sections droites de la poutre aux points O_1 et O_2 .

C.1.2 Résolution

L'écriture algébrique suivante sera adoptée :

$$\vec{R}_1 = R_{1x} \vec{X} + R_{1y} \vec{Y} + R_{1z} \vec{Z}$$

$$\vec{R}_2 = R_{2x} \vec{X} + R_{2y} \vec{Y} + R_{2z} \vec{Z}$$

$$\vec{M}_1 = M_{1x} \vec{X} + M_{1y} \vec{Y} + M_{1z} \vec{Z}$$

$$\vec{M}_2 = M_{2x} \vec{X} + M_{2y} \vec{Y} + M_{2z} \vec{Z}$$

L'écriture du principe fondamental de la statique au point O_2 , permet d'écrire :

$$R_{1x} = -R_{2x} \quad (C.1)$$

$$R_{1y} = -R_{2y} \quad (C.2)$$

$$R_{1z} = -R_{2z} \quad (C.3)$$

$$M_{1x} = -M_{2x} \quad (C.4)$$

$$M_{1y} = -M_{2y} + L.R_{2z} \quad (C.5)$$

$$M_{1z} = -M_{2z} - L.R_{2y} \quad (C.6)$$

En appliquant le principe fondamental de la statique au point G , il est possible d'écrire le torseur de cohésion :

$$[T_{coh}]_G = \begin{bmatrix} R_{2x} & M_{2x} \\ R_{2y} & M_{2y} - (L-x) R_{2z} \\ R_{2z} & M_{2z} + (L-x) R_{2y} \end{bmatrix}_{(G, \vec{X}, \vec{Y}, \vec{Z})} \quad (C.7)$$

C.2 Sollicitation de traction-compression

D'après les lois de la résistance des matériaux, contrainte normale σ , effort normal R_{2x} , et déformation ε sont reliés par :

$$\sigma = \frac{R_{2x}}{S} \quad (C.8)$$

$$\sigma = E\varepsilon \quad (C.9)$$

Avec :

- σ , contrainte normale ;
- R_{2x} , effort normal ;
- S , section de la poutre soit $S = \pi r^2$;
- E , module de Young ;
- ε , déformation soit pour une sollicitation de traction $\varepsilon = \frac{\Delta L}{L}$, avec L longueur de la poutre.

Les égalités C.8, C.9, C.7 et C.1 permettent d'écrire :

$$\boxed{R_{2x} = -R_{1x} = ES \frac{\Delta L}{L}} \quad (C.10)$$

C.3 Sollicitation de torsion

D'après les lois de la résistance des matériaux, le moment de torsion M_t , et l'angle unitaire de torsion α sont reliés par :

$$M_t = GI_o \alpha \quad (C.11)$$

Avec :

- M_t moment de torsion ;
- G module de cisaillement ;
- I_o moment quadratique polaire ;
- α angle unitaire de torsion, $\alpha = \frac{\theta_{2x} - \theta_{1x}}{L}$, avec L longueur de la poutre.

Les égalités C.11, C.7 et C.4 permettent d'écrire :

$$\boxed{M_{2x} = -M_{1x} = \frac{GI_o}{L} (\theta_{2x} - \theta_{1x})} \quad (C.12)$$

C.4 Sollicitation de flexion

D'après les lois de la résistance des matériaux, les rotations des sections droites $\theta_y(x)$, $\theta_z(x)$, les flèches $y(x)$, $z(x)$ et moments fléchissants M_{fz} , M_{fy} sont

reliés par :

$$y''(x) = \frac{M_{fz}}{EI_{Gz}} \quad (C.13)$$

$$\theta'_z(x) = \frac{M_{fz}}{EI_{Gz}} \quad (C.14)$$

$$z''(x) = \frac{M_{fy}}{EI_{Gy}} \quad (C.15)$$

$$\theta'_y(x) = \frac{M_{fy}}{EI_{Gy}} \quad (C.16)$$

Avec :

- M_{fz}, M_{fy} moments de flexion suivant les axes $\vec{Z}$ et $\vec{Y}$;
- E , module de Young du matériau ;
- I_{Gz}, I_{Gy} moments quadratiques en G de la poutre suivant les axes $\vec{Z}$ et $\vec{Y}$;
- $y(x), z(x)$ déformées suivant les axes $\vec{Y}$ et $\vec{Z}$;
- $\theta_y(x), \theta_z(x)$ rotations des sections droites suivant les axes $\vec{Y}$ et $\vec{Z}$.

L'écriture des conditions aux limites donne :

$$\theta_y(0) = \theta_{1y} \quad (C.17)$$

$$\theta_z(0) = \theta_{1z} \quad (C.18)$$

$$\theta_y(L) = \theta_{2y} \quad (C.19)$$

$$\theta_z(L) = \theta_{2z} \quad (C.20)$$

$$y(0) = 0 \quad (C.21)$$

$$z(0) = 0 \quad (C.22)$$

$$y(L) = 0 \quad (C.23)$$

$$z(L) = 0 \quad (C.24)$$

En intégrant l'équation C.14, il vient :

$$\theta_z(x) = \frac{1}{EI_{Gz}} \int M_{fz}.dx + A \quad (C.25)$$

Compte tenu de l'équation C.7 :

$$\begin{aligned} \theta_z(x) &= \frac{1}{EI_{Gz}} \int M_{2z} + (L - x) R_{2y}.dx + A \\ &= \frac{x}{EI_{Gz}} \left[M_{2z} + \left(L - \frac{x}{2} \right) R_{2y} \right] + A \end{aligned}$$

D'après la condition limite C.18 :

$$A = \theta_{1z} \quad (C.26)$$

D'après la condition limite C.19 :

$$\theta_{2z} - \theta_{1z} = \frac{L}{EI_{Gz}} \left(M_{2z} + \frac{LR_{2y}}{2} \right) \quad (C.27)$$

En intégrant maintenant l'équation C.15 :

$$\begin{aligned} y(x) &= \frac{1}{EI_{Gz}} \iint M_{fz}.dx + Ax + B \\ &= \frac{x^2}{2EI_{Gz}} \left[M_{2z} + \left(L - \frac{x}{3} \right) R_{2y} \right] + Ax + B \end{aligned}$$

Or d'après C.26 :

$$y(x) = \frac{x^2}{2EI_{Gz}} \left[M_{2z} + \left(L - \frac{x}{3} \right) R_{2y} \right] + \theta_{1z}x + B \quad (C.28)$$

D'après la condition limite C.22 :

$$B = 0 \quad (C.29)$$

Puis d'après la condition limite C.23 :

$$0 = \frac{l^2}{2EI_{Gz}} \left[M_{2z} + \frac{2L}{3} R_{2y} \right] + \theta_{1z}L \quad (C.30)$$

En résolvant le système d'équations formé par C.30 et C.27, est obtenu :

$$\begin{aligned} M_{2z} &= \frac{2EI}{L} (2\theta_{2z} + \theta_{1z}) \\ R_{2y} &= -\frac{6EI}{L^2} (\theta_{2z} + \theta_{1z}) \end{aligned}$$

Les équations C.7 permettent d'écrire :

$$\begin{aligned} M_{1z} &= \frac{2EI}{L} (\theta_{2z} + 2\theta_{1z}) \\ R_{1y} &= \frac{6EI}{L^2} (\theta_{2z} + \theta_{1z}) \end{aligned}$$

En appliquant la même démarche aux équations C.16 et C.15, et en prenant en compte les conditions aux limites C.17, C.19, C.22 et C.24, est obtenu :

$$\begin{aligned} M_{2y} &= \frac{2EI}{L} (2\theta_{2y} + \theta_{1y}) \\ R_{2z} &= \frac{6EI}{L^2} (\theta_{2y} + \theta_{1y}) \\ M_{1y} &= \frac{2EI}{L} (\theta_{2y} + 2\theta_{1y}) \\ R_{1z} &= -\frac{6EI}{L^2} (\theta_{2y} + \theta_{1y}) \end{aligned}$$

C.4.1 Bilan

Les résultats précédents permettent d'exprimer la réaction de la liaison cohésive *poutre* en fonction des positions et orientations des éléments discrets auxquels celle-ci est liée. Il est donc possible d'écrire le torseur des actions méca-

riques transmissibles :

$$[T_{b/de1}]_{O_1} = \begin{bmatrix} \overrightarrow{R_{p/de1}} \\ \overrightarrow{M_{O_1,p/de1}} \end{bmatrix} = \begin{bmatrix} ES \frac{\Delta L}{L} & \frac{G \cdot I_p}{L} (\theta_{2x} - \theta_{1x}) \\ -\frac{6EI}{L^2} (\theta_{2z} + \theta_{1z}) & -\frac{2EI}{L} (\theta_{2y} + 2\theta_{1y}) \\ \frac{6EI}{L^2} (\theta_{2y} + \theta_{1y}) & -\frac{2EI}{L} (\theta_{2z} + 2\theta_{1z}) \end{bmatrix}_{(\vec{X}, \vec{Y}, \vec{Z})} \quad (C.31)$$

$$[T_{b/de2}]_{O_2} = \begin{bmatrix} \overrightarrow{R_{p/de2}} \\ \overrightarrow{M_{O_2,p/de2}} \end{bmatrix} = \begin{bmatrix} -ES \frac{\Delta L}{L} & -\frac{G \cdot I_p}{L} (\theta_{2x} - \theta_{1x}) \\ \frac{6EI}{L^2} (\theta_{2z} + \theta_{1z}) & -\frac{2EI}{L} (2\theta_{2y} + \theta_{1y}) \\ -\frac{6EI}{L^2} (\theta_{2y} + \theta_{1y}) & -\frac{2EI}{L} (2\theta_{2z} + \theta_{1z}) \end{bmatrix}_{(\vec{X}, \vec{Y}, \vec{Z})} \quad (C.32)$$

C.5 Implémentation dans la plate-forme de simulation DEM « GranOO »

La liaison cohésive *poutre* est implémentée au travers de la classe Beam. Le tableau suivant synthétise les différents attributs nécessaires à la description d'une poutre de section circulaire.

Paramètres	Attributs de la classe Beam
Module de Young	double youngModulus_
Coefficient de Poisson	double poissonRatio_
Longueur initiale à l'état relaxé	double initialLength_
Rayon	double radius_

La classe Beam possède également des attributs permettant le repérage d'un objet dans l'espace.

Paramètres (cf figure C.1a)	Attributs de la classe Beam
Repère $F_1(O_1, \vec{X}_1, \vec{Y}_1, \vec{Z}_1)$	const Geom::Frame<N> F1_
Repère $F_2(O_2, \vec{X}_2, \vec{Y}_2, \vec{Z}_2)$	const Geom::Frame<N> F2_
Repère $F(O, \vec{X}, \vec{Y}, \vec{Z})$	const Geom::Frame<N> F_
Centre du repère F_1	const Geom::Point<N> O1_
Positionnement des repères	
Rotation du repère F_1	const Geom::Quaternion<N> Q1_
Centre du repère F_2	const Geom::Point<N> O2_
Rotation du repère F_2	const Geom::Quaternion<N> Q2_
Centre du repère F	const Geom::Point<N> O_
Rotation du repère F	Geom::Quaternion<N> Q_

C.5.1 Calcul du torseur des efforts transmis, méthode UpdateLoad()

L'extrait de code C++ qui suit permet d'aligner l'axe $\vec{X}$ du repère $F(O, \vec{X}, \vec{Y}, \vec{Z})$ avec le vecteur $\overrightarrow{O_1O_2}$.

```
//Update rotation of frame F
```

```
const Geom::Frame<N> &F0 = Geom::Frame<N>::Global;
const Geom::Vector<N> O1O2(de1_.GetGravityCenter(), de2_.GetGravityCenter());
assert(O1O2.IsNull()==false); //Check if de1 and de2 are not coincident
const Geom::Vector<N> X = O1O2.Unit();
Q_.SetVecFromTo(Geom::Vector<N>(1.,0.,0.,F1_),Geom::Vector<N>(X,F1_));
Q_.Normalize();
```

Code C++	Commentaires
const Geom::Frame<N> &F0 = Geom::Frame<N>::Global; const Geom::Vector<N> O1O2(de1_.GetGravityCenter(), de2_.GetGravityCenter()); assert(O1O2.IsNull()==false); const Geom::Vecteur<N> U = O1O2.Unit(); Q_.SetVecFromTo(Geom::Vector<N>(1.,0.,0.,F1_), Geom::Vector<N>(U,F1_)); Q_.Normalize();	Création d'un synonyme pour le repère global Construction du vecteur $\overrightarrow{O_1O_2}$ Pré-condition : $\ \overrightarrow{O_1O_2}\ \neq 0$ Construction du vecteur $\vec{U} = \overrightarrow{O_1O_2} / \ \overrightarrow{O_1O_2}\ $ Alignement de l'axe $\vec{X}$ du repère F avec le vecteur $\vec{U}$

C.5.2 Expression des vecteurs rotation $\vec{\theta}_1$ et $\vec{\theta}_2$

L'extrait de code C++ qui suit permet de calculer les vecteurs rotations $\vec{\theta}_1$ et $\vec{\theta}_2$ de chacune des deux particules par rapport au repère $F(O, \vec{X}, \vec{Y}, \vec{Z})$.

```
//Compute Angle1
Geom::Quaternion<N> Q1_F (Q1_, F_);
const Geom::Vector<N> rotAxis1(Q1_F[0]*2., Q1_F[1]*2., Q1_F[2]*2., F_);
const double theta1_x = rotAxis1*F_.GetXAxis();
const double theta1_y = rotAxis1*F_.GetYAxis();
const double theta1_z = rotAxis1*F_.GetZAxis();

//Idem for particule 2...
```

Code C++	Commentaires
Geom::Quaternion<N> Q1_F (Q1_, F_); const Geom::Vector<N> theta1(Q1_F[0]*2., Q1_F[1]*2., Q1_F[2]*2., F_); const double theta1_x = theta1[0]; const double theta1_y = theta1[1]; const double theta1_z = theta1[2];	Expression du Quaternion lié à la particule 1 dans le repère d'étude F Construction du vecteur rotation $\vec{\theta}_1$. La partie vectorielle du quaternion $Q1_F$ est extraite puis multipliée par 2. Cette opération est permise dans le cas où les déformations sont petites. Extraction des composantes du vecteur rotation $\vec{\theta}_1$ Même opération pour la particule 2

C.5.3 Calcul du moment

L'extrait de code C++ qui suit permet de calculer les moments de réaction de la poutre.

```
const double l = initialLength_;
//--flexion torque
const double Ig = M_PI*pow(radius_*2.,4)/64.;
const double factor_M = -2. * youngModulus_ * inertia / l;
const double Miy = factor_M * ( theta2_y + (2*theta1_y));
const double Miz = factor_M * ( theta2_z + (2*theta1_z));
```



```
const double M2y = factor_M * ((2*theta2_y) + theta1_y );
const double M2z = factor_M * ((2*theta2_z) + theta1_z );

//--Torsion torque
const double G = youngModulus_ / (2*(1+poissonRatio_));
const double Io = Ig*2.;
const double M1x = (theta2_x - theta1_x)*G*Io/l;
const double M2x = - M1x;
```

Code C++	Commentaires
<pre>const double Ig = M_PI*pow(radius_*2.,4)/64.; const double factor_M = -2. * youngModulus_ * Ig / l; const double M1y = factor_M * (theta2_y + (2*theta1_y)); const double M1z = factor_M * (theta2_z + (2*theta1_z)); const double M2y = factor_M * ((2*theta2_y) + theta1_y); const double M2z = factor_M * ((2*theta2_z) + theta1_z); const double G = youngModulus_ /(2*(1+poissonRatio_)); const double Io = Ig*2.; const double M1x = (theta2_x - theta1_x)*G*Io/l; const double M2x = - M1x;</pre>	Calcul du moment quadratique $I_G = I_{Gy} = I_{Gz} = \pi.d^4/64$ Calcul du facteur $\frac{-2EI}{l}$ Calcul des composantes sur $\vec{Y}$ et $\vec{Z}$ des moments de rappel $\vec{M}_{O_1,p/de1}$ et $\vec{M}_{O_2,p/de2}$ Calcul du module de cisaillement G Calcul du moment quadratique polaire $I_O = \pi.d^4/32$ Calcul de la composante sur $\vec{X}$ des moments de rappel $\vec{M}_{O_1,p/de1}$ et $\vec{M}_{O_2,p/de2}$

C.5.4 Calcul de la résultante

L'extrait de code C++ qui suit permet de calculer les résultantes de réaction de la poutre.

```
//--Flexion force
const double factor_R = 6. * youngModulus_ * inertia / pow(initialLength_,2);
const double R1y = -factor_R * (theta2_z + theta1_z);
const double R1z = factor_R * (theta2_y + theta1_y);
const double R2y = -R1y;
const double R2z = -R1z;
const Geom::Vector<N> T1(R1x, R1y, R1z, F_);
const Geom::Vector<N> T2(R2x, R2y, R2z, F_);
```

Code C++	Commentaires
<pre>const double factor_R = 6. * youngModulus_ * inertia / pow(l,2);</pre>	Calcul du facteur $\frac{6EI}{l^2}$
<pre>const double R1y = -factor_R * (theta2_z + theta1_z); const double R1z = factor_R * (theta2_y + theta1_y); const double R2y = -R1y; const double R2z = -R1z;</pre>	Calcul des composantes sur $\vec{Y}$ et $\vec{Z}$ des forces de rappel $\vec{R}_{p/de1}$ et $\vec{R}_{p/de2}$

C.5.5 Affectation des moments et des résultantes

L'extrait de code C++ qui suit permet de vérifier et d'appliquer les résultantes et moments de réaction de la poutre sur chacune des deux particules.

```
//Check results
```

```

assert(R1x==R1x);
assert(R1y==R1y);
assert(R1z==R1z);
assert(M1x==M1x);
assert(M1y==M1y);
assert(M1z==M1z);
assert(M2y==M2y);
assert(M2z==M2z);

//Build Torque
const Geom::Vector<N> M1(M1x, M1y, M1z, F_);
const Geom::Vector<N> M2(M2x, M2y, M2z, F_);

//Affect Torque
de1_.GetExternalTorque() += Geom::Vector<N>(M1, F0);
de2_.GetExternalTorque() += Geom::Vector<N>(M2, F0);

//Build Force
const Geom::Vector<N> R1(R1x, R1y, R1z, F_);
const Geom::Vector<N> R2(R2x, R2y, R2z, F_);

//Affect Force
de1_.GetExternalForce() += Geom::Vector<N>(R1, F0);
de2_.GetExternalForce() += Geom::Vector<N>(R2, F0);

```

Code C++	Commentaires
<pre> assert(R1x==R1x); assert(R1y==R1y); assert(R1z==R1z); assert(M1x==M1x); assert(M1y==M1y); assert(M1z==M1z); assert(M2y==M2y); assert(M2z==M2z); const Geom::Vector<N> M1(M1x, M1y, M1z, F_); const Geom::Vector<N> M2(M2x, M2y, M2z, F_); Interaction<N>::de1_.GetExternalTorque() += Geom::Vector<N>(M1, F0); Interaction<N>::de2_.GetExternalTorque() +=Geom::Vector<N>(M2, F0); const Geom::Vector<N> R1(R1x, R1y, R1z, F_); const Geom::Vector<N> R2(R2x, R2y, R2z, F_); de1_.GetExternalForce() += Geom::Vector<N>(R1, F0); de2_.GetExternalForce() += Geom::Vector<N>(R2, F0); </pre>	Post-condition :Vérification que les valeurs obtenues ne sont pas des nan ou des inf Construction des moments de rappel $\overrightarrow{M_{O_{1,p}/de1}}$ et $\overrightarrow{M_{O_{2,p}/de2}}$. Ces moments sont exprimés dans le repère de travail F. Affectation des moments. Ces moments doivent être exprimés dans le repère global F_0 Construction des résultantes de rappel $\overrightarrow{R_{p/de1}}$ et $\overrightarrow{R_{p/de2}}$. Ces résultantes sont exprimées dans le repère de travail F. Affectation des résultantes. Ces résultantes doivent être exprimées dans le repère global F_0

C.6 Validation de l'implémentation

Cette section propose de valider l'implémentation de la classe Beam au travers d'une simulation d'un système masse ressort (voir figure C.3). Cette étude permet de vérifier la cohérence des résultats dans un problème à trois dimensions. La figure C.3 montre une vue 3D du système obtenu. La position de l'élément discret *DE1* est fixe, et une impulsion est donnée à l'élément discret *DE2* afin d'initier le mouvement. La masse de l'élément *DE2* est très grande devant les autres éléments, permettant ainsi de négliger les effets inertiels du ressort lui-même.

Paramètres	Symbole	Valeur	Unité
Masse DE2	M_2	$4,18879 \times n \ n \in \mathbb{N}$	Kg
Masse du Ressort	M_R	0,0345575	Kg
Module de Young des poutres	E	10^9	Pa
Coefficient de Poisson des poutres	ν	0,3	-
Rayon des poutres	R_p	0,005	m
Rayon d'enroulement	R_e	0,04	m
Nombre de spire utile	n	6,5	-

TABLE C.1 – Paramètres de l'étude « essai de torsion 1D »

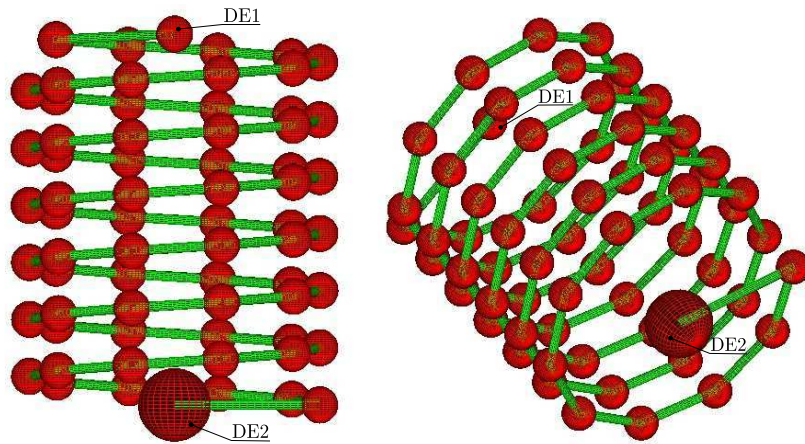


FIGURE C.3 – Modèle DEM du ressort

Le ressort est un composant technologique bien connu, sa raideur théorique est donnée par l'égalité :

$$k_{th} = \frac{G.d^4}{8.D^3.n} \quad (C.33)$$

Avec :

- k_{th} raideur théorique en Nm^{-1} ;
- D diamètre d'enroulement en m ;
- G module de cisaillement en Pa ;
- d diamètre de fil en m ;
- n nombre de spires utiles.

C.6.1 Pré-étude, Invariabilité de la raideur

Les paramètres numériques de l'étude sont indiqués dans le tableau C.1. La rigidité du ressort découle de l'assemblage des liaisons poutres entre les éléments discrets. Les paramètres mécaniques des poutres étant inchangés au cours de dix simulations, il s'agit de vérifier dans un premier temps l'invariabilité de la raideur du ressort, puis de comparer les résultats obtenus avec le résultat théorique.

Pour chaque simulation , la masse de l'élément discret DE2 est incrémentée d'une valeur constante.

Masse M_2 (Kg)	4,2	8,4	12,6	16,7	21,0	25,1	29,3	33,5	37,7	41,9
Masse du Ressort M_R (Kg)	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,04
Oscillation T (s)	1,0	1,4	1,8	2,0	2,3	2,5	2,7	2,9	3,0	3,2
Raideur K (N.m ⁻¹)	157,9	158,1	158,2	158,2	158,2	158,3	158,3	158,2	158,3	158,3

TABLE C.2 – Résultats des simulations « essai de torsion 1D »

Or, la période d'oscillation d'un système masse-ressort est donnée par

$$T = 2\pi\sqrt{\frac{m}{k}} \quad (\text{C.34})$$

Avec :

- T , période d'oscillation en s ;
- m , masse du système en kg ;
- k raideur du ressort Nm⁻¹.

Les effets d'inertie du ressort sont négligés devant celle de la masse en mouvement, ce qui correspond bien au cas mis en oeuvre au travers des simulations.

Cette dernière égalité permet également d'écrire :

$$k = \left(\frac{2\pi}{T}\right)^2 m \quad (\text{C.35})$$

Au cours des simulations, l'amplitude du déplacement de la masse est sauvegardée. À partir de ces résultats, les périodes d'oscillation peuvent être extraites. Les masses étant connues, il est possible de calculer la raideur du ressort grâce à l'égalité C.35. Le tableau C.2 récapitule les résultats obtenus. Le plus grand écart obtenu entre les raideurs sont seulement de 0,2%. Ce résultat permet donc de conclure à une bonne fiabilité du modèle mise en oeuvre.

C.6.2 Étude de convergence

Les résultats de l'étude de convergence sont présentés figure C.4. Les résultats des simulations montrent une convergence de la période d'oscillation vers la solution théorique lorsque le nombre d'éléments discrets augmente.

C.7 Conclusion

L'implémentation et l'intégration d'un modèle théorique dans un code de calcul sont des opérations délicates. Les sources d'erreurs sont multiples : fautes de frappe, écrasement de données non prévu, approximations non négligeables avec effet « boule de neige », erreurs causées par des bibliothèques externes, etc.

Aussi malgré l'attention portée lors de l'écriture du code et l'utilisation systématique de pré-conditions et de post-conditions[84] nul n'est à l'abri de calculs erronés. Les résultats des tests présentés ci-dessus ont été comparés de façon minutieuse avec des résultats théoriques. Dans les cas présentés, les solutions numériques et théoriques sont très proches. Il est donc légitime d'affirmer que l'implémentation du modèle de liaison poutre a été correctement réalisée. De plus, ces résultats permettent également de valider les méthodes numériques employées et l'architecture logicielle : schéma d'intégration explicite *Verlet-vitesse*, utilisation des quaternions pour représenter les rotations, etc.

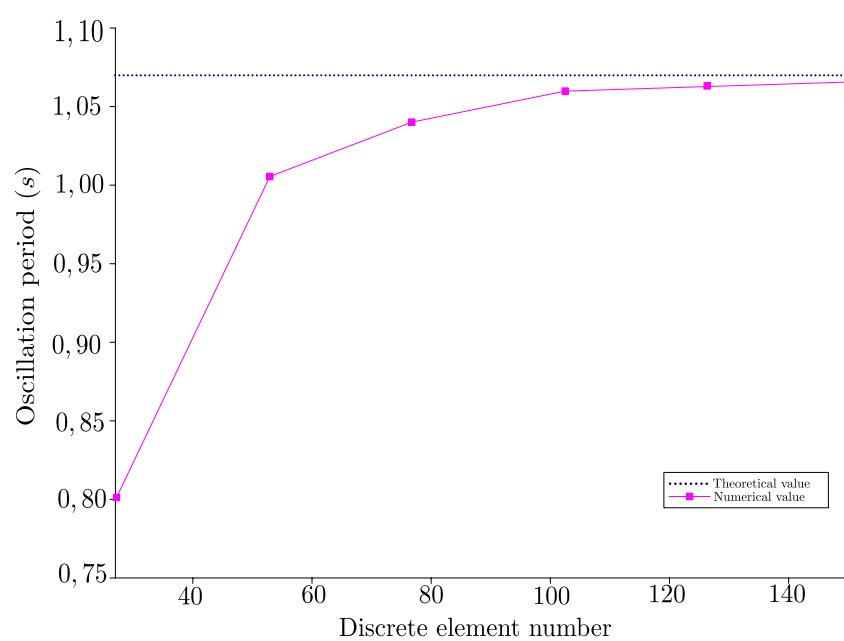


FIGURE C.4 – Étude de convergence des périodes d'oscillation « essai de torsion 1D »

Calcul des énergies : description des méthodes employées

D.1 Travail des efforts externes

Soit un échantillon numérique soumis à des actions mécaniques externes. Ces actions mécaniques se traduisent par des forces et des moments appliqués au centre de gravité des éléments discrets. Soit :

- $\vec{f}_i$ l'effort extérieur appliqué à l'élément discret i (en N) ;
- $\vec{m}_i$ le moment extérieur appliqué à l'élément discret i (en $N.m$) ;
- $d\vec{\theta}_i$ la rotation infinitésimale (en rad) de l'élément discret i ;
- $d\vec{\theta}_i$ le déplacement infinitésimal (en m) de l'élément discret i .

Dans la mesure où le pas de temps de calcul Δt est suffisamment faible, il est possible de considérer que les déplacements et rotations effectués entre deux pas de temps sont infinitésimaux. Ainsi, pour le pas de temps p correspondant au temps t :

$$d\vec{\theta}_i^t \approx \Delta\vec{\theta}_i^p = \vec{\theta}_i^p - \vec{\theta}_i^{p-1} \quad (D.1)$$

$$d\vec{u}_i^t \approx \Delta\vec{u}_i^p = \vec{u}_i^p - \vec{u}_i^{p-1} \quad (D.2)$$

Dans ce cas, le travail des efforts extérieurs peut être exprimé par :

$$E_T = \sum_{i=1}^n \sum_{p=1}^j \Delta\vec{\theta}_i^p \cdot \vec{m}_i^p + \Delta\vec{u}_i^p \cdot \vec{f}_i^p \quad (D.3)$$

Avec :

- n le nombre total des éléments discrets ;
- j le nombre total de pas de calcul ;
- E_T le travail des efforts extérieurs du domaine discret (en J).

D.2 Énergie interne de déformation

L'énergie interne de déformation est considérée comme étant le travail des efforts de réaction des liaisons cohésives. La même approximation de calcul que précédemment est appliquée (voir équations et). Il est possible de décomposer

l'énergie interne de déformation en énergie de :

- traction (travail de l'effort normal) ;
- flexion (travail des moments de flexions) ;
- torsion (travail du moment de torsion) ;
- cisaillement (travail des efforts tranchants).

Ainsi, en utilisant la notation utilisée dans la section 3.2 :

$$E_{trac} = \sum_{i=1}^n \sum_{p=1}^j \Delta u_{i_1X}^p \cdot F_{i_1X}^p + \Delta u_{i_2X}^p \cdot F_{i_2X}^p \quad (D.4)$$

$$E_{flex} = \sum_{i=1}^n \sum_{p=1}^j \Delta \theta_{i_1Y}^p \cdot T_{i_1Y}^p + \Delta \theta_{i_2Y}^p \cdot T_{i_2Y}^p + \Delta \theta_{i_1Z}^p \cdot T_{i_1Z}^p + \Delta \theta_{i_2Z}^p \cdot T_{i_2Z}^p \quad (D.5)$$

$$E_{tor} = \sum_{i=1}^n \sum_{p=1}^j \Delta \theta_{i_1X}^p \cdot T_{i_1X}^p + \Delta \theta_{i_2X}^p \cdot T_{i_2X}^p \quad (D.6)$$

$$E_{cis} = \sum_{i=1}^n \sum_{p=1}^j \Delta u_{i_1Y}^p \cdot F_{i_1Y}^p + \Delta u_{i_2Y}^p \cdot F_{i_2Y}^p + \Delta u_{i_1Z}^p \cdot F_{i_1Z}^p + \Delta u_{i_2Z}^p \cdot F_{i_2Z}^p \quad (D.7)$$

Avec :

- n le nombre total de joints cohésifs ;
- j le nombre total de pas de calcul ;
- E_{trac} , E_{flex} , E_{tor} et E_{cis} respectivement les énergies internes de déformation de traction, de flexion, de torsion et de cisaillement du domaine discret (en J) ;
- $\Delta \theta_{i_1X}^p$ rotation infinitésimale du joint cohésif d'indice i de la section droite relative à l'élément discret 1 projeté sur l'axe X au pas de calcul p (en rad) ;
- $\Delta u_{i_1X}^p$ déplacement infinitésimal du joint cohésif d'indice i de la section droite relative à l'élément discret 1 projeté sur l'axe X au pas de calcul p (en m).

D.3 Énergie cinétique

L'énergie cinétique d'un domaine discret est définie par la somme des énergies cinétiques des éléments discrets le composant. Les éléments discrets étant sphériques, les tenseurs d'inertie associés sont invariants quel que soit le repère d'expression. D'où :

$$E_{cin} = \sum_{i=1}^n \frac{1}{2} M_i v_i^2 + \frac{1}{2} I_i \omega_i^2 \quad (D.8)$$

Avec :

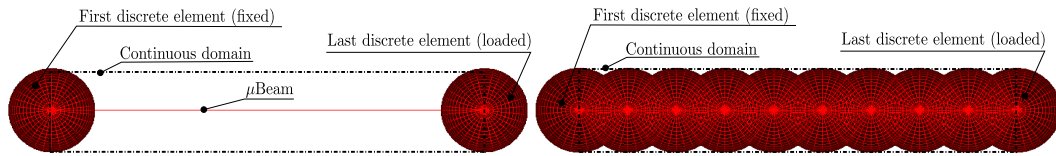
- n le nombre total des éléments discrets ;
- E_{cin} énergie cinétique du domaine discret (en J) ;
- M_i masse de l'élément discret i (en kg) ;
- v_i module de la vitesse de l'élément discret i par rapport au repère global (en $m.s^{-1}$) ;
- I_i inertie de l'élément discret i (en $kg.m^2$). $I_i = \frac{2}{5} M_i r_i^2$ où r_i est le rayon de l'élément discret i ;

- ω_i module de la vitesse de rotation de l'élément discret i par rapport au repère global (en $rad.s^{-1}$).

Essai de torsion 1D, démonstration de l'erreur de 10% de calibration de l'inertie

L'étude de convergence de domaine discret 1D (voir figure E.1) sous sollicitation de torsion dynamique montre une erreur de convergence de 10% (voir figure E.2). La solution théorique des périodes d'oscillation est donnée par l'équation suivante :

$$T = 4L\sqrt{\frac{\rho}{G}} \quad (\text{E.1})$$



(a) Domaine discrétisé en deux éléments discrets (b) Domaine discrétisé en vingt éléments discrets

FIGURE E.1 – Illustration des domaines d'étude de convergence dynamique

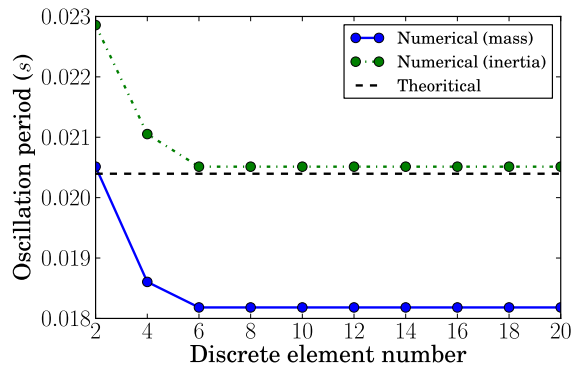


FIGURE E.2 – Période d'oscillation sous sollicitation de torsionlabel

Dans le cas des essais illustrés par la courbe « *Numerical (mass)* » sur la figure E.2, la masse volumique des éléments discrets est calculée de façon à assurer une égalité des masses entre domaine discret et domaine continu. La masse du domaine continu, de forme cylindrique, est donnée par :

$$M_c = \pi R^2 L \rho_c \quad (E.2)$$

Où :

- M_c est la masse du domaine continu ;
- R est le rayon du domaine continu de forme cylindrique ;
- L est la longueur du domaine continu de forme cylindrique ;
- ρ_c est la masse volumique du domaine continu.

La masse du domaine discret est donnée par :

$$M_D = N \frac{4}{3} \pi R^3 \rho_D \quad (E.3)$$

Où :

- M_D est la masse du domaine discret ;
- N est le nombre d'éléments discrets ;
- R est le rayon des éléments discrets, égal au rayon du domaine continu de forme cylindrique ;
- ρ_D est la masse volumique des éléments discrets.

Donc, en égalisant les équations E.2 et E.3, il est obtenu :

$$\rho_D^{mass} = \frac{3}{4} \frac{\rho_c L}{N} \quad (E.4)$$

L'autre courbe « *Numerical (inertia)* » sur la figure E.2, illustre les résultats de l'étude de convergence lorsque la masse volumique des éléments discrets est calculée de façon à assurer une égalité des inerties suivant l'axe du cylindre entre domaine discret et domaine continu. Dans le cas d'un cylindre d'axe $\vec{x}$, la composante diagonale du tenseur d'inertie selon son axe est donnée par :

$$I_c = \frac{1}{2} \pi R^4 L \rho_c \quad (E.5)$$

L'inertie totale du domaine discret suivant l'axe $\vec{x}$ est :

$$I_D = N \frac{8}{15} \pi R^5 \rho_c \quad (E.6)$$

Donc, en égalisant les équations E.5 et E.6, il est obtenu :

$$\rho_D^{inertia} = \frac{15}{16} \frac{\rho_c L}{N} \quad (E.7)$$

À partir de la solution théorique, il est possible de calculer l'écart sur les périodes d'oscillations entre les deux modèles de calibration de la masse volumique

du domaine discret ρ_D^{mass} et $\rho_D^{inertia}$. Ainsi :

$$\begin{aligned}
 \frac{T^{mass}}{T^{inertia}} &= \frac{4L\sqrt{\frac{\rho_D^{mass}}{G}}}{4L\sqrt{\frac{\rho_D^{inertia}}{G}}} \\
 &= \sqrt{\frac{\rho_D^{mass}}{\rho_D^{inertia}}} \\
 &= \sqrt{\frac{\frac{3}{4}\frac{\rho_c L}{N}}{\frac{15}{16}\frac{\rho_c L}{N}}} \\
 \frac{T^{mass}}{T^{inertia}} &= \sqrt{\frac{12}{15}} \approx 0,89
 \end{aligned} \tag{E.8}$$

En conséquence, l'écart de période d'oscillation entre le modèle de calibration de la masse volumique du domaine discret basée sur une équivalence de masse ρ_D^{mass} et sur une équivalence d'inertie $\rho_D^{inertia}$, est constant et égal à 10%.

Description de l'essai de flexion à rupture

La figure F.1 illustre l'essai de flexion à rupture. Une géométrie d'éprouvette parallélépipédique est préférée à la géométrie cylindrique. En effet, pour une éprouvette cylindrique sollicitée en flexion, la zone des contraintes maximales est localisée le long d'une ligne située en surface de l'éprouvette (voir figure F.2b). Dans le cas d'une éprouvette parallélépipédique, la zone des contraintes maximales est située sur l'une des faces des éprouvettes (voir figure F.2a).

Aussi, la zone des contraintes maximales étant très faible, la probabilité de solliciter mécaniquement un défaut (ou maillon faible) est donc beaucoup moins élevée. La géométrie parallélépipédique permet de ne pas surévaluer la contrainte à rupture en flexion.

L'essai choisi est de type *flexion pur*. Le moment de flexion est supposé constant le long de la poutre macroscopique. Des rotations $\phi(t).\vec{Z}$ et $-\phi(t).\vec{Z}$ sont appliqués progressivement aux extrémités de la poutre macroscopique jusqu'à l'apparition d'une fissure « franche ». Cette rotation se traduit par des déplacements et des rotations des éléments discrets appartenant aux faces $xMin$ et $xMax$:

$$\vec{d}_i^{xMax}.\vec{Y} = (\vec{OG}_i.\vec{Y}) \cos(\phi) \quad (F.1)$$

$$\vec{d}_i^{xMax}.\vec{X} = (\vec{OG}_i.\vec{Y}) \sin(\phi) \quad (F.2)$$

$$\vec{d}_i^{xMin}.\vec{X} = -(\vec{OG}_i.\vec{Y}) \sin(\phi) \quad (F.3)$$

$$\vec{d}_i^{xMin}.\vec{Y} = -(\vec{OG}_i.\vec{Y}) \cos(\phi) \quad (F.4)$$

$$\vec{\theta}_i^{xMax}.\vec{Z} = \phi \quad (F.5)$$

$$\vec{\theta}_i^{xMin}.\vec{Z} = -\phi \quad (F.6)$$

Où :

- $\vec{d}_i^{xMax}$ et $\vec{d}_i^{xMin}$ sont les vecteurs déplacements des éléments discrets i appartenant aux faces $xMin$ et $xMax$;
- $\vec{OG}_i$ est le vecteur position de l'élément discret i ;
- $\vec{\theta}_i^{xMax}$ et $\vec{\theta}_i^{xMin}$ sont les vecteurs rotations des éléments discrets i appartenant aux faces $xMin$ et $xMax$.

$xMin$

Pour déterminer la contrainte macroscopique à rupture, il est nécessaire de mesurer le moment fléchissant macroscopique m_{f_M} . Celui est supposé constant le long de la poutre macroscopique. En pratique, il sera considéré comme étant la moyenne des moments fléchissants, $m_{f_M}^{xMax}$ et $m_{f_M}^{xMin}$, mesurés aux extrémités de l'éprouvette. Ces derniers sont déduits des mesures des efforts $\vec{f}_i$ et moments $\vec{m}_i$ extérieurs appliqués aux éléments discrets appartenant aux faces $xMin$ et $xMax$:

$$m_{f_M}^{xMax} = \sum_{i=1}^{n_{xMax}} \left(\vec{m}_i + \overrightarrow{O_1 G_i} \wedge \vec{f}_i \right) \cdot \vec{Z} \quad (F.7)$$

$$m_{f_M}^{xMin} = \sum_{i=1}^{n_{xMin}} \left(\vec{m}_i + \overrightarrow{O_2 G_i} \wedge \vec{f}_i \right) \cdot \vec{Z} \quad (F.8)$$

$$m_{f_M} = \left(m_{f_M}^{xMax} + m_{f_M}^{xMin} \right) / 2 \quad (F.9)$$

Où les points O_2 et O_1 sont les centres géométriques des faces $xMin$ et $xMax$. La contrainte normale macroscopique maximale σ_M^{max} est alors déduite des lois de la résistance des matériaux par :

$$\sigma_M^{max} = \frac{m_{f_M} h_M}{I_{gz} 2} \quad (F.10)$$

Où I_{gz} est le moment quadratique de la poutre macroscopique par rapport à l'axe $\vec{Z}$, soit :

$$I_{gz} = \frac{b_M h_M^3}{12} \quad (F.11)$$

La contrainte à rupture macroscopique est déduite de la contrainte normale maximale selon le critère de *Rankine* (voir section 3.6.1) adapté aux matériaux fragiles :

$$\sigma_{M_f} = \sigma_{M_f}^{max} \quad (F.12)$$

Où $\sigma_{M_f}^{max}$ est la contrainte normale maximale mesurée lors de la rupture de l'échantillon numérique.

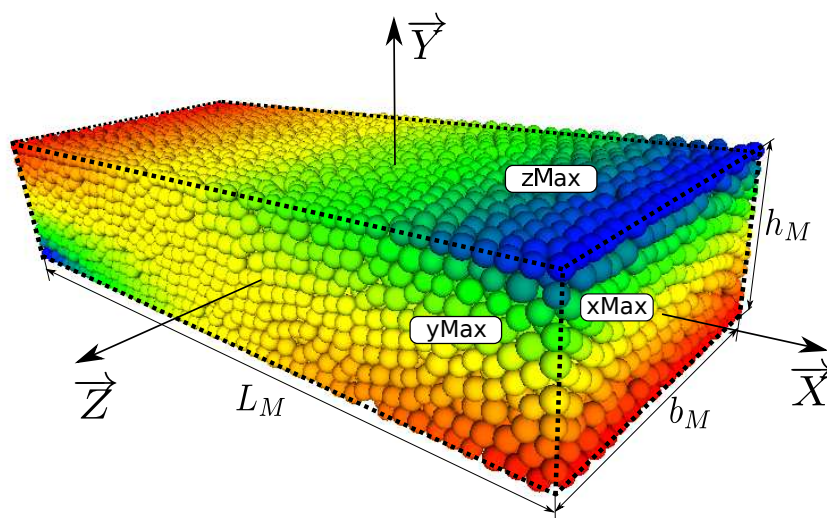


FIGURE F.1 – Illustration de l'essai de flexion à rupture
 Le code couleur correspond aux vecteurs déplacements des éléments discrets projetés sur l'axe $\vec{X}$

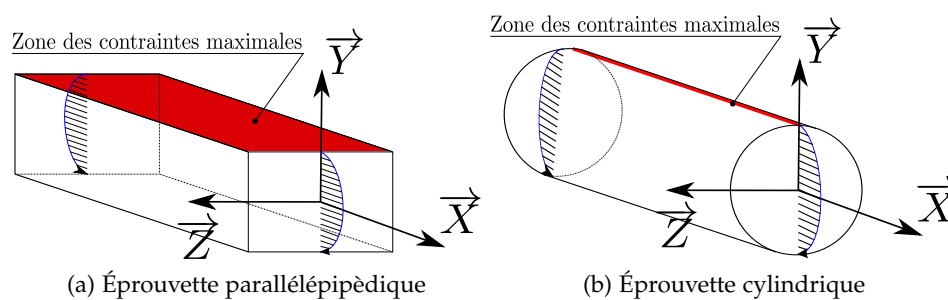


FIGURE F.2 – Zone des contraintes maximales pour des éprouvettes cylindriques et parallélépipédique en flexion pure

Présentation des essais de torsion à rupture

Le mode opératoire est similaire à l'essai de flexion à rupture. Des déplacements et rotations aux éléments discrets appartenant aux faces $xMin$ et $xMax$ (voir figure G.1) sont appliqués de façon à traduire une rotation de section droite des extrémités de la poutre macroscopique ϕ et $-\phi$ suivant l'axe $\vec{X}$.

$$\vec{d}_i^{xMax} \cdot \vec{Y} = \left\| \vec{OG}_i \cdot \vec{Y} \right\| \cos(\phi) \quad (G.1)$$

$$\vec{d}_i^{xMax} \cdot \vec{Z} = \left\| \vec{OG}_i \cdot \vec{Y} \right\| \sin(\phi) \quad (G.2)$$

$$\vec{d}_i^{xMin} \cdot \vec{Y} = - \left\| \vec{OG}_i \cdot \vec{Y} \right\| \cos(\phi) \quad (G.3)$$

$$\vec{d}_i^{xMin} \cdot \vec{Z} = - \left\| \vec{OG}_i \cdot \vec{Y} \right\| \sin(\phi) \quad (G.4)$$

$$\vec{\theta}_i^{xMax} \cdot \vec{X} = \phi \quad (G.5)$$

$$\vec{\theta}_i^{xMin} \cdot \vec{X} = -\phi \quad (G.6)$$

Où :

- $\vec{d}_i^{xMax}$ et $\vec{d}_i^{xMin}$ sont les vecteurs déplacements des éléments discrets i appartenant aux faces $xMin$ et $xMax$;
- $\vec{OG}_i$ est le vecteur position de l'élément discret i ;
- $\vec{\theta}_i^{xMax}$ et $\vec{\theta}_i^{xMin}$ sont les vecteurs rotations des éléments discrets i appartenant aux faces $xMin$ et $xMax$.

Le moment de torsion macroscopique m_{t_M} est ensuite déduit des efforts et moments extérieurs des éléments discrets appartenant aux faces $xMin$ et $xMax$.

$$m_{t_M}^{xMax} = \sum_{i=1}^{n_{xMax}} \left(\vec{m}_i + \vec{OG}_i \wedge \vec{f}_i \right) \cdot \vec{X} \quad (G.7)$$

$$m_{t_M}^{xMin} = \sum_{i=1}^{n_{xMin}} \left(\vec{m}_i + \vec{OG}_i \wedge \vec{f}_i \right) \cdot \vec{X} \quad (G.8)$$

$$m_{t_M} = \left(m_{t_M}^{xMax} + m_{t_M}^{xMin} \right) / 2 \quad (G.9)$$

Où les points O_2 et O_1 sont les centres géométriques des faces $xMin$ et $xMax$. La contrainte de cisaillement macroscopique maximale τ_M^{max} est alors déduite des

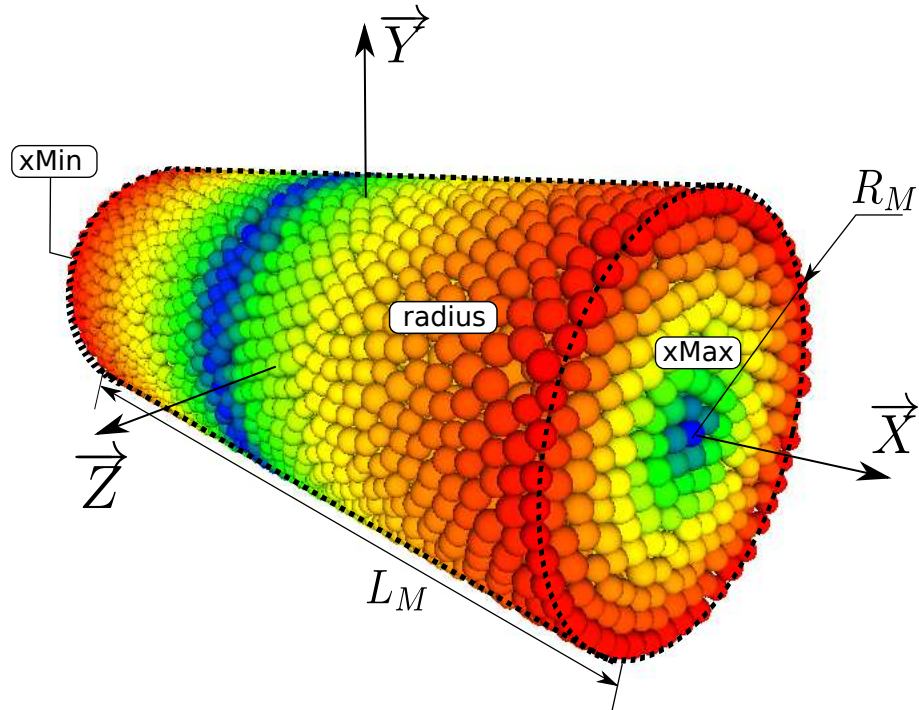


FIGURE G.1 – Illustration de l'essai de torsion à rupture
Le code couleur correspond à l'intensité des vecteurs déplacements des éléments discrets

lois de la résistance des matériaux par :

$$\tau_M^{max} = \frac{m_{t_M}}{I_o} R_M \quad (G.10)$$

Où I_o est le moment quadratique polaire de la poutre macroscopique, soit :

$$I_o = \frac{\pi R_M^4}{2} \quad (G.11)$$

La contrainte à rupture macroscopique est déduite de la contrainte de cisaillement maximale selon le critère de *Rankine* (voir section 3.6.1) adapté aux matériaux fragiles :

$$\sigma_{M_f} = \tau_{M_f}^{max} \quad (G.12)$$

Où $\tau_{M_f}^{max}$ est la contrainte de cisaillement maximale mesurée lors de la rupture de l'échantillon numérique.

ANNEXE H

Résultats expérimentaux de référence

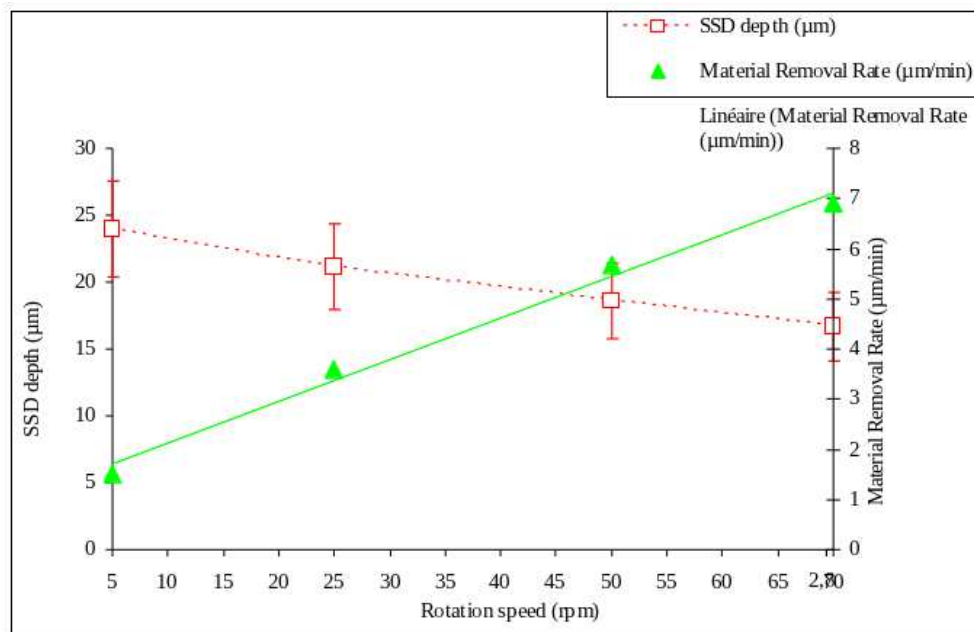


FIGURE H.1 – Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la vitesse de rotation de l'outil - image extraite de [93] -

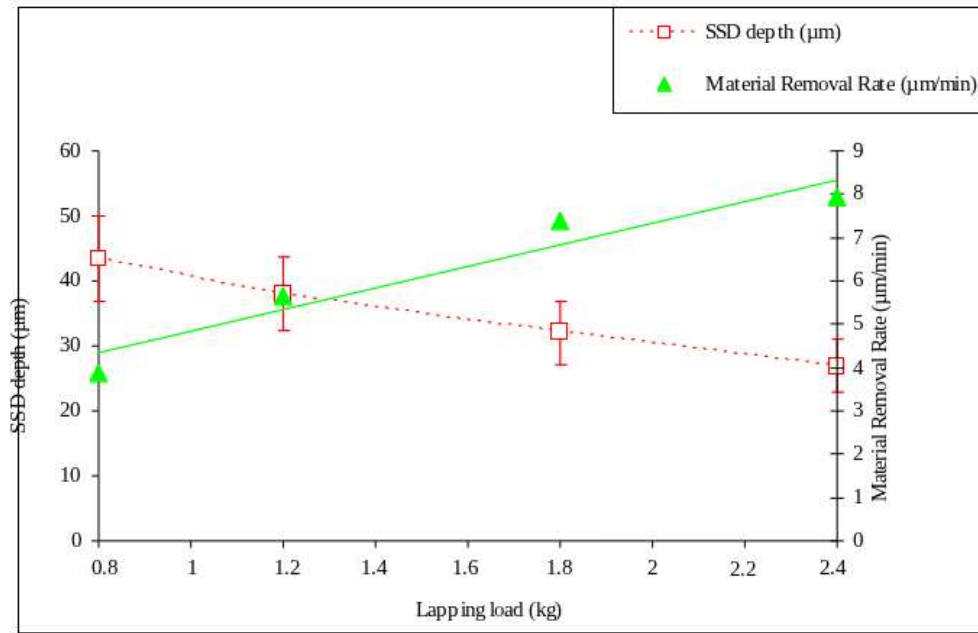


FIGURE H.2 – Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la vitesse de rotation de la pression exercée sur la silice - image extraite de [93] -

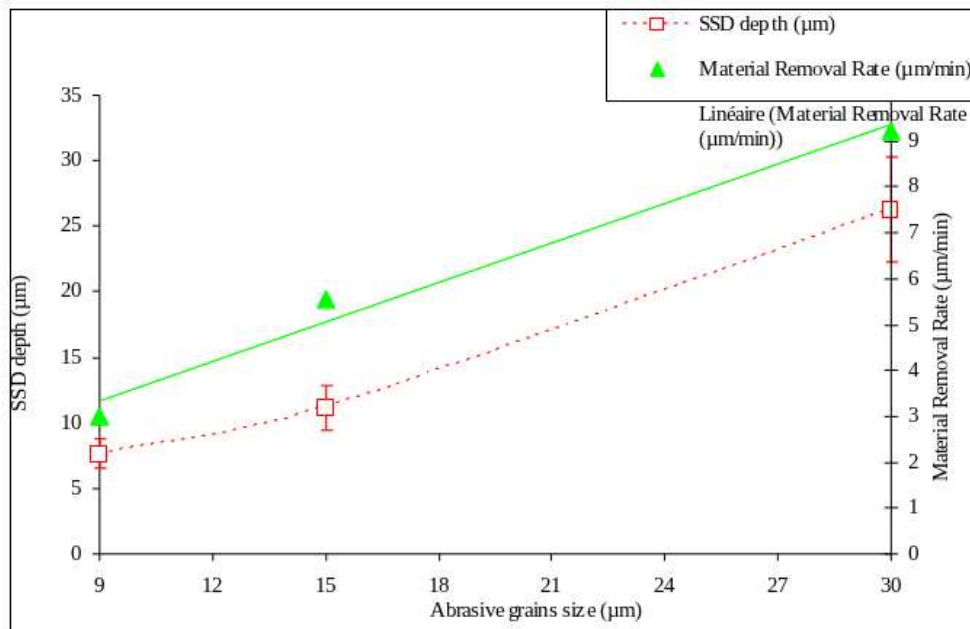


FIGURE H.3 – Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la taille moyenne des grains abrasifs - image extraite de [93] -

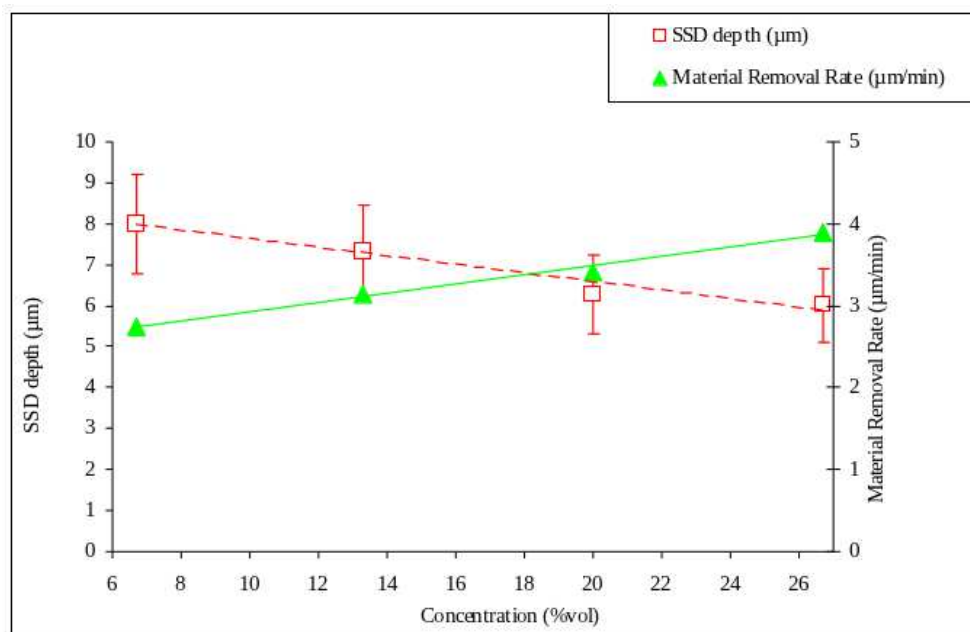


FIGURE H.4 – Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la concentration du *slurry* - image extraite de [93] -

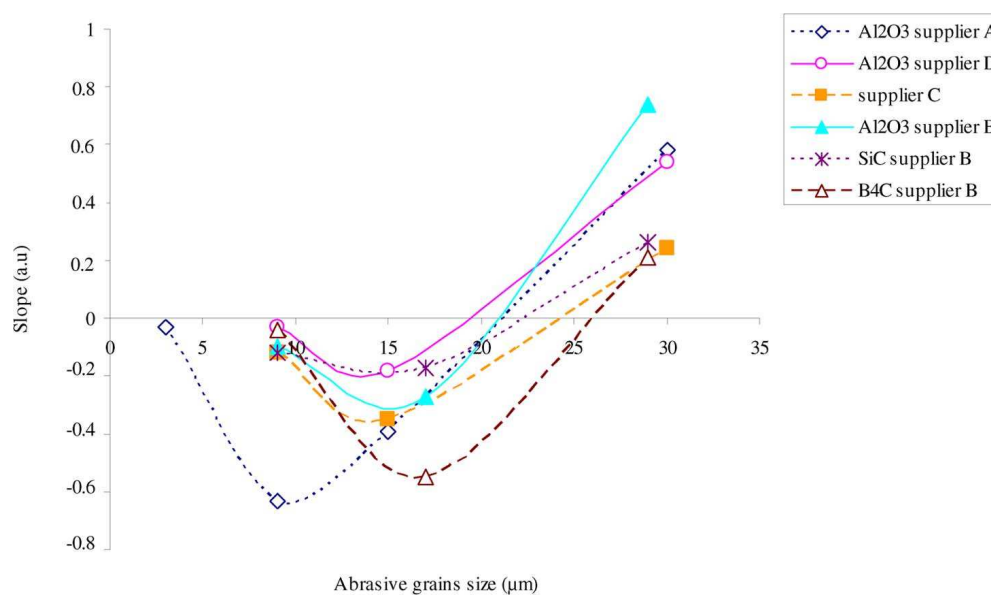


FIGURE H.5 – Évolution de la pente des évolutions des profondeurs max de SSD par rapport à la concentration du *slurry* en fonction du diamètres des grains abrasifs - image extraite de [93] -

Algorithme de reconnaissance d'agrégats

I.1 Fonction de détection d'un agrégat : DetectCluster()

L'algorithme de détection d'agrégat est basé sur la fonction récursive DetectCluster() (voir algorithme 2). Cette fonction permet (voir figures I.1) :

1. à partir d'un élément discret particulier, de balayer les liaisons auxquelles celui-ci est relié. Les éléments discrets maintiennent à jour une liste de liaisons, accessible *via* la méthode GetBondList() de la classe DiscreteElement ;
2. à partir de cette liste de liaison, de balayer les éléments discrets en interaction par ces liaisons. La méthode GetOtherDiscretElement() de la classe Bond, prenant en argument un élément discret, retourne l'autre élément discret lié par l'objet Bond ;
3. grâce à la récursivité de balayer toutes les liaisons possibles en prenant soin de ne pas balayer deux fois une même liaison. Cette vérification est faite grâce à la liste de liaison déjà balayée : ParsedBondList ;
4. d'enrichir la liste des éléments discrets composant l'agrégat (DEList) au fur et à mesure des appels de la fonction DetectCluster().

Algorithm 2 Fonction de détection d'agrégats - Fonction DetectCluster()

```

function DETECTCLUSTER(DE, DEList, ParsedBondList)
  if DEList.Contain(DE) is False then
    DEList.Add(DE)
  end if
  BondList ← de.GetBondList()
  for all Bond in BondList do
    if ParsedBondList.Contain(Bond) is False then
      ParsedBondList.Add(Bond)
      BondedDE ← Bond.GetOtherDiscretElement(DE)
      DETECTCLUSTER(BondedDE, DEList, ParsedBondList)
    end if
  end for
end function

```

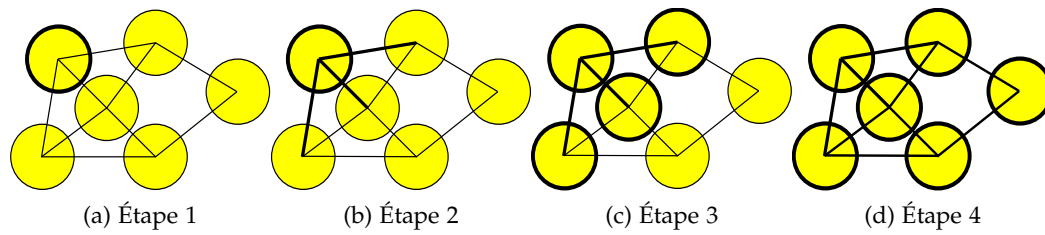


FIGURE I.1 – Illustration de la fonction de détection d'agrégats

I.2 Détection de tous les agrégats

La fonction précédente permet, à partir d'un élément discret de construire une liste définissant l'agrégat auquel cet élément appartient. L'algorithme 3 permet alors :

1. de créer une liste d'agrégat : *ClusterList* (liste de liste d'éléments discrets) ;
2. de construire un agrégat pour chaque élément discret « silice » ;
3. d'ajouter cet agrégat à la liste *ClusterList* ;
4. d'éviter les doublons d'agrégat pouvant être contenus dans *ClusterList* grâce à la fonction *MustBeParsed()* (voir algorithme 4). Cette fonction permet d'exclure la construction d'agrégat pour des éléments discrets déjà contenus dans un des agrégats existants.

Algorithm 3 Algorithme de détection d'agrégats - Algorithme principal

Require: *SilicaList*

ClusterList $\leftarrow$ *Empty*

for all *DE* in *SilicaList* **do**

Answer $\leftarrow$ *MUSTBEPARSED*(*DE*, *ClusterList*) (algo. 4)

if *Answer* is *True* **then**

Cluster $\leftarrow$ *Empty*

ParsedBondList $\leftarrow$ *Empty*

DETECTCLUSTER(*DE*, *Cluster*, *ParsedBondList*)

ClusterList.Add(*Cluster*)

end if

end for

Algorithm 4 Algorithme de détection d'agrégats - Fonction MustBeParsed()

```

function MUSTBEPARSED(DE, ClusterList)
  if ClusterList is Empty then
    return True
  else
    for all DEList in ClusterList do
      if DEList.Contain(DE) is True then return False
      end if
    end for
  end if
  return True
end function

```

Cet algorithme étant extrêmement coûteux en terme de temps de calcul, celui-ci est déclenché de façon sporadique. En effet, il est nécessaire de rafraîchir la liste des agrégats que, lorsqu'au cours d'une itération de calcul, des liaisons cohésives se sont rompues. Sur la totalité des itérations, les phases de ruptures ne représentent qu'un faible pourcentage de la totalité des itérations de calcul. Les pertes de performance, induit par cet algorithme, deviennent alors négligeables.

Condition de périodicité

Des conditions périodiques sont introduites dans le modèle DEM de surfacage afin d'étendre virtuellement les dimensions du domaine d'étude. Il s'agit, comme le montre la figure J.1, d'effectuer des copies du domaine discret puis de les déplacer suivant une *direction périodique* et d'une *distance périodique*.

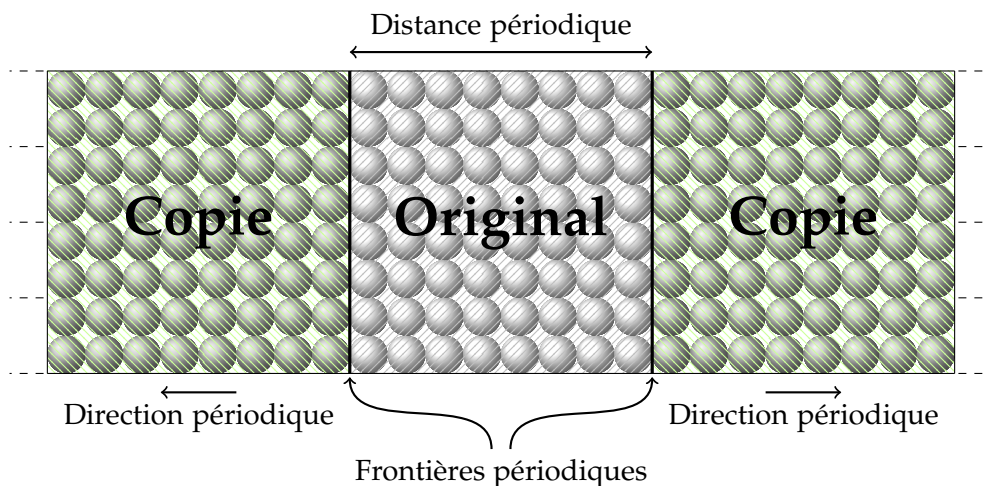


FIGURE J.1 – Copie puis translation d'un domaine discret suivant une condition périodique

Une condition périodique se traduit au niveau des éléments discrets par l'introduction de positions *virtuelles* en plus de leurs positions réelles. Il peut être considéré qu'un élément discret possède alors deux positions dans l'espace. La position virtuelle traduit la translation de l'élément discret suivant la direction de périodicité d'une valeur égale à la *distance périodique*.

Toutefois, l'implémentation des positions virtuelles n'est pas réalisée dans *GranOO*. Les conditions périodiques sont alors appliquées ponctuellement, lors de différents traitements nécessitant la prise en compte de ces conditions par simple translation de l'élément discret. Aussi, pour minimiser le nombre de calculs, les conditions périodiques ne sont appliquées que pour les éléments discrets proches des frontières périodiques, comme l'illustre la figure J.2.

Une application directe des conditions périodiques est la *recirculation*. La recirculation est appliquée si le centre de gravité d'un élément discret franchit une

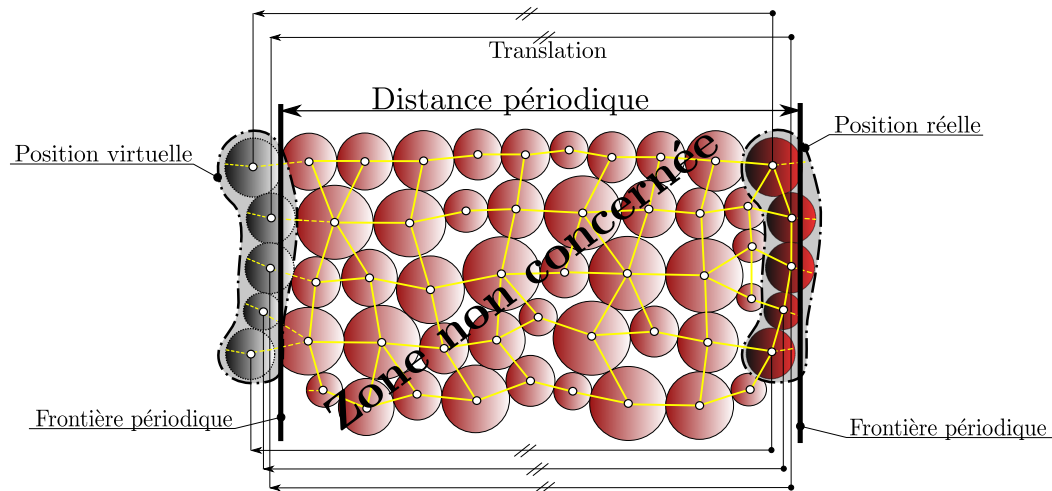


FIGURE J.2 – Mise en oeuvre d'une condition périodique dans un calcul DEM

paroi périodique. Celui-ci est alors réintroduit de « l'autre côté » du domaine suivant les conditions de périodicité.

D'autre part, certains traitements nécessitent la prise en compte de conditions périodiques. Deux traitements sont alors concernés : le calcul des réactions des liaisons cohésives et la détection des contacts.

J.1 Réaction des liaisons cohésives et conditions périodiques

Les conditions de périodicité permettent à deux éléments discrets, situés de part et d'autre du domaine, d'être en interaction. Lors du calcul des efforts et moments de réaction des liaisons cohésives, il est alors nécessaire de prendre en compte cette propriété.

D'un point de vue informatique, la classe *Beam* de la *libDEM* est dérivée en une classe *PeriodicBeam*. La méthode virtuelle *Update()*, de calcul des actions mécaniques de réaction, est surchargée pour prendre en compte les effets de la périodicité. L'algorithme 5 décrit la méthode *Update()* de la classe *PeriodicBeam*. Celle-ci permet :

1. de calculer la distance selon l'axe périodique (*PeriodicAxis*) séparant les deux éléments discrets en interaction. Le résultat est stocké dans la variable *Distance* ;
2. de tester si cette distance est supérieure à un certain ratio de la distance périodique (*PeriodicDistance*). Si le résultat du test est vrai, il est considéré que l'interaction est périodique et un des éléments discrets est translaté ;
3. d'appeler la méthode *Update()* de la classe de base *Beam* ;
4. de remettre l'élément discret déplacé à sa position d'origine, dans le cas où, l'interaction est périodique.

Algorithm 5 Prise en charge des conditions périodiques lors du calcul des réactions des liaisons cohésives - Méthode `Update()` de la classe `PeriodicBeam`

Require: *PeriodicDistance*

```

Method PERIODICBEAM::UPDATE()
    DiscreteElementMoved  $\leftarrow$  False
     $G_1 \leftarrow$  DiscreteElement1.GetGravityCenter()
     $G_2 \leftarrow$  DiscreteElement2.GetGravityCenter()
    Distance  $\leftarrow \overrightarrow{G_1 G_2} \cdot \vec{X}$ 
    if Distance is superior to PeriodicDistance/2 then
        TranslateAlongX(PeriodicDistance, PeriodicAxis)
        DiscreteElementMoved  $\leftarrow$  True
    end if
    Beam : Update()
    if DiscreteElementMoved is True then
        UnTranslateAlongX()
    end if

```

J.2 Détection des contacts et conditions périodiques

Afin de détecter les contacts entre deux éléments discrets situés de part et d'autre d'un domaine, le domaine est divisé en cinq zones suivant la figure J.3. Cette subdivision permet d'ignorer les éléments discrets de la zone *non concernée*. En plus des zones *xMax* et *xMin*, contiguës aux frontières périodiques, il est défini des « marges de sécurité » *xMax_m* et *xMin_m*. La fonction `DetectPeriodicContact()`, décrite par l'algorithme 6 permet :

1. de déplacer les éléments discrets (fonction `TranslateAlongX()`) appartenant à la zone *xMin*, de façon à la rendre contiguë à la zone *xMax* ;
2. de procéder à une détection des contacts « classique » entre les éléments discrets appartenant à :
 - la zone *xMax* et à la zone *xMin* ;
 - puis entre la zone *xMax_m* et la zone *xMin*.

La fonction `DetectAndProcessContactBetween()` est décrite au travers de l'algorithme 7. Afin d'optimiser les performances de calcul, la fonction `ProcessContact()`, calculant les efforts de réaction du contact, est appelée « à la volée », lorsqu'un contact est détecté entre deux éléments discrets.

3. de remettre les éléments discrets appartenant à la zone *xMin* à leurs positions d'origine (fonction `UnTranslateAllDiscreteElementIn()`) ;
4. de déplacer les éléments discrets appartenant à la zone *xMax* de façon à la rendre contiguë la zone *xMin* ;
5. de procéder à une détection des contacts « classiques » entre les éléments discrets appartenant à la zone *xMax* et à la zone *xMin_m* ;
6. de remettre les éléments discrets appartenant à la zone *xMax* à leurs positions d'origine.

Algorithm 6 Prise en charge des conditions périodiques - Fonction `DetectPeriodicContact()`

```

function DETECTPERIODICCONTACT
    TranslateAlongX( $xMin$ )
    DetectAndProcessContactBetween( $xMin$ ,  $xMax$ )
    DetectAndProcessContactBetween( $xMin$ ,  $xMax\_m$ )
    UnTranslateAlongX( $xMin$ )

    TranslateAlongX( $xMax$ )
    DetectAndProcessContactBetween( $xMax$ ,  $xMin\_m$ )
    UnTranslateAlongX( $xMin\_m$ )
end function

```

Algorithm 7 Fonction de détection des contacts entre deux conteneurs d'éléments discrets - Fonction `DetectAndProcessContactBetween()`

```

function DETECTANDPROCESSCONTACTBETWEEN( $DEList1$ ,  $DEList1$ )
    for all  $DiscreteElement1$  in  $DEList1$  do
        for all  $DiscreteElement2$  in  $DEList2$  do
             $G_1 \leftarrow DiscreteElement1.GetGravityCenter()$ 
             $G_2 \leftarrow DiscreteElement2.GetGravityCenter()$ 
             $R_1 \leftarrow DiscreteElement1.GetRadius()$ 
             $R_2 \leftarrow DiscreteElement2.GetRadius()$ 
             $distance \leftarrow \|\overrightarrow{G_1 G_2}\|$ 
            if  $distance$  is inferior to  $R_1 + R_2$  then
                ProcessContact( $DiscreteElement1$ ,  $DiscreteElement2$ )
            end if
        end for
    end for
end function

```

J.3 Prise en charge de la double périodicité

Afin de conserver une symétrie suivant les axes $\vec{X}$ et $\vec{Z}$, le domaine est doublement périodique, suivant ces deux directions.

La prise en charge de ces conditions, lors du calcul des réactions des liaisons cohésives, n'introduit pas de difficulté spécifique. La méthode `Update()` de la classe `DoublePeriodicBeam` permet de prendre en compte ces conditions. Cette méthode est décrite au travers de l'algorithme 8. Elle est très proche de la méthode `Update()` de la classe `PeriodicBeam` décrite précédemment au travers de l'algorithme 5.

L'algorithme de détection des contacts doit être adapté afin de prendre en compte la périodicité au niveau des arêtes du domaine. La fonction `DetectDoublePeriodicContact()`, décrite au travers de l'algorithme 9, permet :

1. de procéder à une détection des contacts pour chaque axe périodique selon l'algorithme 6 grâce aux fonctions `DetectPeriodicContactAlongX()` et `DetectPeriodicContactAlongZ()` ;
2. de déplacer les éléments discrets appartenant à la zone 1 de façon à la rendre contiguë la zone 13 ;

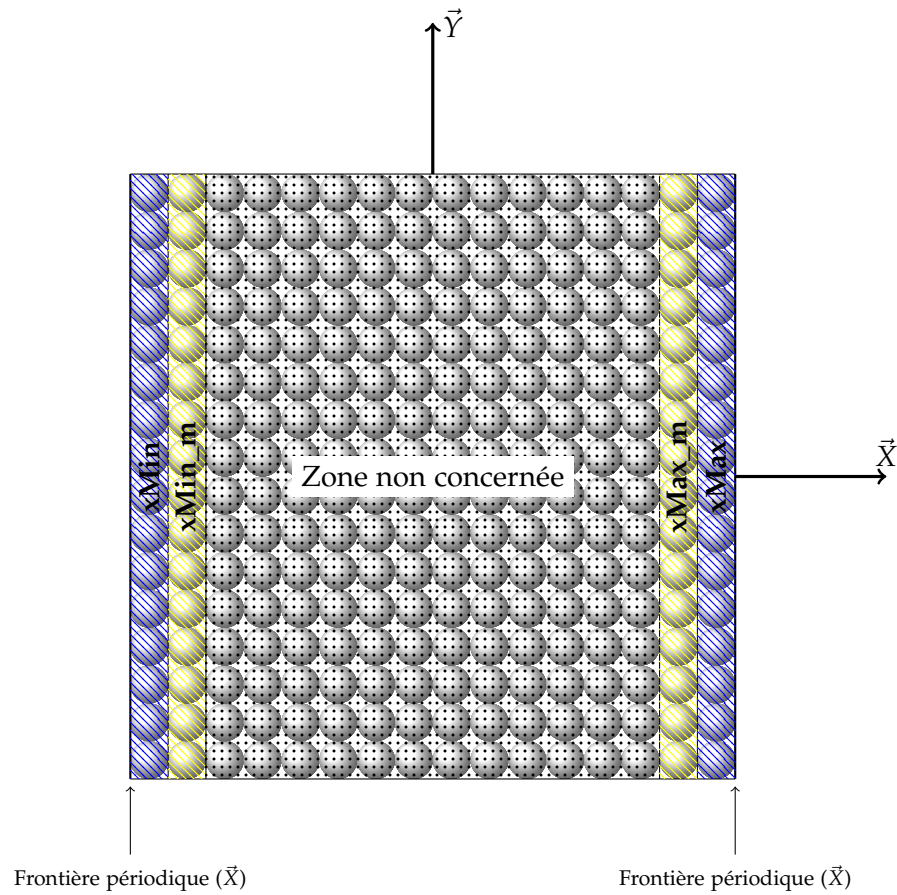


FIGURE J.3 – Subdivision du domaine

3. de rechercher les contacts entre la zone 1 et la zone $(13 \cup 14 \cup 15 \cup 16)$;
4. de remettre les éléments discrets appartenant à la zone 1 à leurs positions d'origine ;
5. de déplacer les éléments discrets appartenant à la zone 13 de façon à la rendre contiguë la zone 1 ;
6. de rechercher les contacts entre la zone 13 et la zone $(2 \cup 3 \cup 4)$;
7. de remettre les éléments discrets appartenant à la zone 13 à leurs positions d'origine ;
8. de prendre en compte les contacts entre zones $(5 \cup 6 \cup 7 \cup 8)$ et $(9 \cup 10 \cup 11 \cup 12)$ en procédant de la même façon.

Algorithm 8 Prise en charge des doubles conditions périodiques lors du calcul des réactions des liaisons cohésives - Méthode Update() de la classe DoublePeriodicBeam

Require: *PeriodicDistanceX*, *PeriodicDistanceZ*

Method DOUBLEPERIODICBEAM::UPDATE()

DiscreteElementMovedOnX $\leftarrow$ *False*

DiscreteElementMovedOnZ $\leftarrow$ *False*

$G_1 \leftarrow \text{DiscreteElement1.GetGravityCenter}()$

$G_2 \leftarrow \text{DiscreteElement2.GetGravityCenter}()$

$\text{DistanceX} \leftarrow \overrightarrow{G_1 G_2} \cdot \vec{X}$

$\text{DistanceZ} \leftarrow \overrightarrow{G_1 G_2} \cdot \vec{Z}$

if *DistanceX* is superior to *PeriodicDistanceX*/2 **then**

 TranslateAlongX()

DiscreteElementMovedOnX $\leftarrow$ *True*

end if

if *DistanceZ* is superior to *PeriodicDistanceZ*/2 **then**

 TranslateAlongZ()

DiscreteElementMovedOnZ $\leftarrow$ *True*

end if

Beam : :Update()

if *DiscreteElementMovedX* is *True* **then**

 UnTranslateAlongX()

end if

if *DiscreteElementMovedOnZ* is *True* **then**

 UnTranslateAlongZ()

end if

Algorithm 9 Prise en charge de la double périodicité lors de la phase de détection des contacts - Fonction DetectDoublePeriodicContact()

```

function DETECTDOUBLEPERIODICCONTACT
    DetectPeriodicContactAlongX()                ▷ similar to algorithme 6
    DetectPeriodicContactAlongZ()                ▷ similar to algorithme 6

    TranslateAlongXandZ(list 1)
    DetectAndProcessContactBetween(list 1, list (13 ∪ 14 ∪ 15 ∪ 16))
    UnTranslateAlongXandZ(list 1)

    TranslateAlongXandZ(list 13)
    DetectAndProcessContactBetween(list 1, list (2 ∪ 3 ∪ 4))
    UnTranslateAlongXandZ(list 13)

    TranslateAlongXandZ(list 5)
    DetectAndProcessContactBetween(list 5, list (9 ∪ 11 ∪ 12))
    UnTranslateAlongXandZ(list 5)

    TranslateAlongXandZ(list 9)
    DetectAndProcessContactBetween(list 9, list (6 ∪ 7 ∪ 8))
    UnTranslateAlongXandZ(list 9)
end function

```

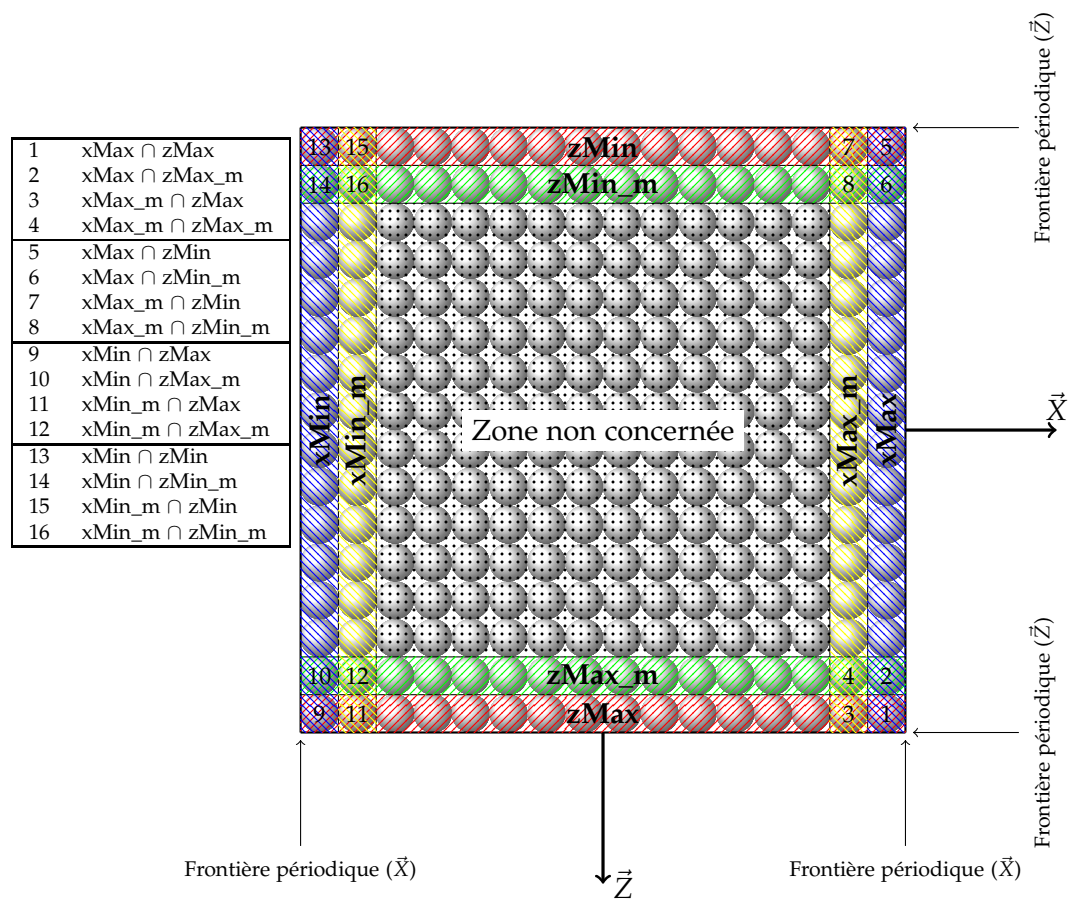


FIGURE J.4 – Subdivision du domaine pour une double périodicité

Algorithme de reconnaissance des surfaces externes

Pour différents besoins, il est nécessaire au cours de la simulation du doucissage de déterminer les limites d'un agrégat d'élément discret. Par exemple, il est souhaité pouvoir qualifier la couche de silice abrasée. Cette couche représente une surface externe de l'agrégat *silice*. Afin d'être en mesure de déterminer cette surface, la fonction `DetectExternalSurface()`, décrite au travers l'algorithme 10, est utilisé. Celle-ci prend en argument une liste d'éléments discrets (*DEList*) ainsi qu'un vecteur ($\vec{v}_0$) représentant la direction normale à la surface. La fonction renvoie ensuite une liste d'éléments discrets (*SurfaceDEList*) contenant les éléments de la surface. L'algorithme se décompose en plusieurs étapes (voir également figure K.1) :

1. création d'une liste vide d'élément discret (*SurfaceDEList*) qui sera enrichie au cours de l'algorithme ;
2. créations de vecteurs orthogonaux à $\vec{v}_0$: $\vec{v}_1$ et $\vec{v}_2$;
3. détection des limites du domaine suivant les directions orthogonales : d_1^{min} , d_1^{max} , d_2^{min} et d_2^{max} ;
4. création d'une grille rectangulaire dans le plan ($\vec{v}_1$, $\vec{v}_2$) dont les colonnes et les lignes sont espacées du rayon moyen ($\bar{r}$) des éléments discrets du domaine étudié. Les intersections sont stockées sous la forme de points (p_n). Des droites (Δ_n) passant par p_n et de direction $\vec{v}_0$ sont ensuite construites ;
5. détection de l'élément discret le plus éloigné par rapport à la direction défini par $\vec{v}_0$ en intersection avec Δ_n grâce à la méthode `GetMostDistantDE()` ;
6. ajout de l'élément discret, si celui-ci n'est pas déjà présent, à la liste *SurfaceDEList*.

Tout comme l'algorithme de détection d'agrégat (voir annexe, section I), la fonction de reconnaissance des surfaces externes est coûteuse en temps de calcul. Celle-ci est donc déclenchée de façon sporadique lorsqu'au cours d'une itération de calcul, des liaisons cohésives se sont rompues.

Algorithm 10 Fonction de reconnaissance des surfaces externes - Fonction DetectExternalSurface()

```

function DETECTEXTERNALSURFACE(DEList,  $\vec{v}_0$ )
    SurfaceDEList  $\leftarrow$  build an empty discrete element list

     $\bar{r} \leftarrow DEList.GetAverageRadius()$ 
     $\vec{v}_1 \leftarrow \text{ComputeOrthogonalVector}(\vec{v}_0)$ 
     $\vec{v}_2 \leftarrow \vec{v}_0 \wedge \vec{v}_1$ 

     $d_1^{min} \leftarrow DEList.GetMinDistanceValueAlong(\vec{v}_1)$ 
     $d_1^{max} \leftarrow DEList.GetMaxDistanceValueAlong(\vec{v}_1)$ 
     $d_2^{min} \leftarrow DEList.GetMinDistanceAlong(\vec{v}_2)$ 
     $d_2^{max} \leftarrow DEList.GetMaxDistanceAlong(\vec{v}_2)$ 

     $d_1 \leftarrow d_1^{min}$ 
    while  $d_1 \leq d_1^{max}$  do
         $d_2 \leftarrow d_2^{min}$ 
        while  $d_2 \leq d_2^{max}$  do
             $p_n \leftarrow d_1 \cdot \vec{v}_1 + d_2 \cdot \vec{v}_2$ 
             $\Delta_n \leftarrow (p_n, \vec{v}_0)$ 
             $DE \leftarrow DEList.GetMostDistantDE(\Delta_n)$ 
            if SurfaceDEList.Contain( $DE$ ) is False then
                SurfaceDEList.Add( $DE$ )
            end if
             $d_2 \leftarrow d_2 + \bar{r}$ 
        end while
         $d_1 \leftarrow d_1 + \bar{r}$ 
    end while

    return SurfaceDEList

end function

```

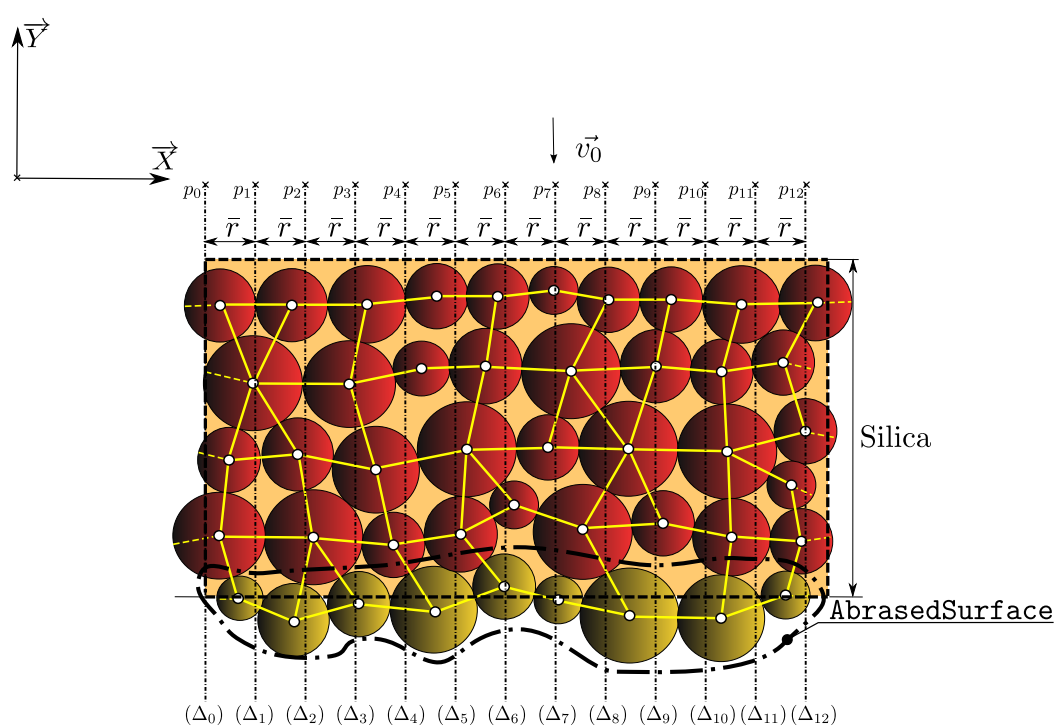


FIGURE K.1 – Illustration 2D de la fonction de reconnaissance des surfaces externes

Table des figures

1.1	Vue aérienne du LMJ, Août 2009 (http://www-lmj.cea.fr/)	2
1.2	Illustration de la cible (http://www-lmj.cea.fr/)	3
1.3	Photo de la chambre d'expérience (http://www-lmj.cea.fr/)	4
1.4	Schémas d'une chaîne laser	5
1.5	Domage laser - <i>image extraite de [135]</i> -	6
1.6	Relation entre puissance des installations lasers et travaux portant sur les endommagements lasers	6
1.7	Schéma de principe du doucissage	8
1.8	Illustration de différentes géométries de fissuration créées par indentation [124]	9
1.9	Image issue de microscopie électronique à balayage de poudres abrasives - <i>images extraites de [93]</i> -	9
1.10	Structure de l'interface de silice après polissage [120]	10
1.11	Procédé de quantification des SSD par polissage local (<i>Taper polishing</i>)	11
1.12	Le surfacage vue comme un problème tribologique	14
1.13	Structure de la silice SiO ₂	15
1.14	Zone des contraintes maximales sous sollicitations de traction et de flexion	16
1.15	Illustration d'un problème de compaction	18
1.16	Modélisation DEM d'un problème de flexion de poutre cylindrique	19
1.17	Modélisation DEM d'un problème de rupture de poutre cylindrique sous sollicitation de torsion	19
1.18	Graphique montrant l'évolution de la densité de SSD dans l'épaisseur de silice issue de simulation DEM. Résultats extraits de [54].	22
2.1	Exemple de représentation d'une classe dans le formalisme UML. Vue (simplifiée) de la classe <code>DiscreteElement</code>	31
2.2	Exemple de <i>diagramme d'objets</i> issu du formalisme UML. Vue de trois instances <code>de1</code> , <code>de2</code> et <code>de3</code> de la classe <code>DiscreteElement</code>	31
2.3	Classe portée par une relation. La classe <code>contact</code> est portée par l'association entre deux objets de la classe <code>DiscreteElement</code>	32
2.4	<i>Diagramme de classes</i> . La classe <code>Contact</code> est dérivée en classes <code>ElasticContact</code> , <code>ViscousContact</code> et <code>ThermalContact</code>	33
2.5	Illustration du principe d'encapsulation par l'allégorie de l'iceberg	34
2.6	Bibliothèques composant la plate-forme de calcul GranOO	36
2.7	Les entités géométriques au coeur de la <i>libGeometrical</i>	37
2.8	Les formes géométriques de la <i>libGeometrical</i>	38
2.9	Comparatif des performances de la <i>libGeometrical</i> et de <i>Blitz++</i>	40
2.10	Vue simplifiée de l'architecture générale de la <i>libDEM</i>	43

2.11	Vue détaillée de l'architecture générale des conteneurs de la <i>libDEM</i>	44
2.12	Architecture du système de greffon	46
2.13	Pages d'accueil de la plateforme M2P	49
2.14	Page d'accueil du site <i>granoo.org</i>	51
2.15	Vue synthétique de l'architecture d'exploitation	53
2.16	Visualisation du fichier <i>Cylinder.gdd</i> avec le programme <i>gddViewer</i>	55
3.1	Illustration de différents modèles de liaisons cohésives	60
3.2	Expérience de cisaillement à rupture utilisée par Schlangen et Garboczi. Image extraite de [115].	61
3.3	Illustration du modèle hybride « lattice - particulaire »	62
3.4	Illustration de la liaison cohésive poutre	63
3.5	Méthode de compaction	65
3.6	Solide platonique (géode) permettant de classifier et de comptabiliser les orientations des contacts	66
3.7	Arrangement géométrique de domaines compactés pour différentes valeurs de dispersion radiale.	67
3.8	Évolution de l'écart quadratique moyen e en fonction de la dispersion χ appliquée au rayon des éléments discrets	68
3.9	Évolution des paramètres géométriques en fonction du nombre d'éléments discrets	68
3.10	Illustration de l'essai numérique de traction	70
3.11	Évolution du chargement et répartition des énergies au cours d'un essai de traction quasi statique	71
3.12	Évolution des valeurs calculées de E_M et de ν_M en fonction du temps	72
3.13	Répartition des énergies de déformation pour une éprouvette cylindrique sous sollicitation quasi statique de traction, flexion et torsion	72
3.14	Évolution de E_M et de ν_M en fonction de ν_μ	73
3.15	Influence du module de Young microscopique E_μ sur les paramètres macroscopiques E_M et ν_M	75
3.16	Influence du rayon adimensionné microscopique $\tilde{r}_\mu$ sur les paramètres macroscopiques E_M et ν_M	75
3.17	Calibration des paramètres microscopiques $\tilde{r}_\mu$ et E_μ du matériau silice	75
3.18	Calcul du rayon du domaine continu équivalent	76
3.19	Différents échantillons numériques utilisés pour l'étude de convergence	77
3.20	Évolution des paramètres macroscopiques en fonction du nombre d'éléments discrets	77
3.21	Illustration des domaines d'étude de convergence dynamique	79
3.22	Période d'oscillation sous sollicitation dynamique	80
3.23	Illustration du déplacement du front d'onde mécanique <i>Le code couleur représente les vitesses des éléments discrets projetées sur l'axe $\vec{x}$</i>	81
3.24	Mesure de la vitesse moyenne des éléments discrets projetée sur l'axe $\vec{x}$ de la face $xMin$	82
3.25	Évolutions du nombre de liaisons rompues et des contraintes normales mesurées aux extrémités de la poutre	85
3.26	Calibration du paramètre microscopique de rupture σ_{μ_f} du matériau silice	86
3.27	Captures d'image de géométrie de fissuration <i>Le code couleur correspond à la contrainte principale maximale $\sigma_{\mu_{max}}$ des liaisons cohésives poutres</i>	87

3.28	Évolution de la contrainte à rupture macroscopique σ_{M_f} en fonction du nombre d'éléments discrets	88
3.29	Études de convergence pour des domaines discrets « mal compactés » caractérisés par un nombre de coordination de $N_c = 5$	89
4.1	Exemple de modèle DEM d'étude de problème tribologique <i>Image extraite [34]</i>	92
4.2	Dispositif de polissage et de doucissage <i>PM5</i>	94
4.3	Dispositif de précharge <i>Jig</i>	95
4.4	Localisation du modèle DEM par rapport au procédé	97
4.5	Illustration du modèle de doucissage DEM	98
4.6	Illustration 2D des différents corps du modèle DEM du doucissage	99
4.7	Illustration du modèle de rupture <i>progressive</i>	100
4.8	Domaine de la solution aqueuse	102
4.9	Illustration de la méthode de calcul de la raideur équivalente de contact <i>K</i>	104
4.10	Illustration des contacts de Hertz	105
4.11	Illustration 2D du modèle de fluide	107
4.12	Illustration 2D des chargements et conditions limites	109
4.13	Échelles de l'échantillon de silice réel et de l'échantillon numérique	110
4.14	Illustration du calcul vitesse de translation de l'outil	112
4.15	Évolution du nombre de liaisons cohésives <i>poutres</i> rompues au cours du temps - <i>Conditions de référence</i> -	115
4.16	Cartographie de la surface abrasée et des SSD - <i>Conditions de référence, temps simulé de 507 μs</i> -	116
4.17	Profil de répartition des SSD dans l'épaisseur - <i>Conditions de référence, temps simulé de 507 μs</i> -	117
4.18	Évolution de la profondeur maximale de SSD - <i>Conditions de référence</i> -	118
4.19	Évolution de la position moyenne projetée sur $\vec{Y}$ de la paroi supérieure du bloc de silice - <i>Conditions de référence</i> -	119
4.20	Évolution de la force normale appliquée sur la paroi supérieure du bloc de silice - <i>Conditions de référence</i> -	120
4.21	Illustration 2D des mécanismes de chargement abrasif/silice	121
4.22	Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes conditions de précharge	123
4.23	Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes conditions de vitesse	125
4.24	Comparaisons des résultats expérimentaux et numériques de l'influence des dimensions des abrasifs pour différentes concentration	126
4.25	Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes dimensions des abrasifs	128
4.26	Comparaisons des résultats expérimentaux et numériques de l'influence de la concentration en abrasif pour différents rayons d'abrasif	130
4.27	Évolutions des forces normales, de liaisons rompues, d'altitudes de la paroi supérieure et des profondeurs maximales de SSD pour différentes concentrations en abrasifs	131

4.28	Évolutions des rugosités rms pour différentes conditions de précharges	134
4.29	Évolutions des rugosités rms pour différentes conditions de vitesses	134
4.30	Évolutions des rugosités rms pour différentes dimensions des abrasifs	135
4.31	Évolutions des rugosités rms pour différentes concentrations en abrasif	135
A.1	Vecteur liant deux points	145
A.2	Quaternion liant deux vecteurs	145
A.3	Définition d'un scalaire	146
A.4	Définition d'un versor	146
A.5	Rotation comme produit de deux symétries	149
C.1	Modèle de liaisons cohésive « poutre »	162
C.2	Modèle de liaison cohésive <i>poutre</i> , bilan des actions mécaniques	163
C.3	Modèle DEM du ressort	171
C.4	Étude de convergence des périodes d'oscillation « essai de torsion 1D »	173
E.1	Illustration des domaines d'étude de convergence dynamique	179
E.2	Période d'oscillation sous sollicitation de torsionlabel	179
F.1	Illustration de l'essai de flexion à rupture <i>Le code couleur correspond aux vecteurs déplacements des éléments discrets projetés sur l'axe $\vec{X}$</i>	185
F.2	Zone des contraintes maximales pour des éprouvettes cylindriques et parallélépipédique en flexion pure	185
G.1	Illustration de l'essai de torsion à rupture <i>Le code couleur correspond à l'intensité des vecteurs déplacements des éléments discrets</i>	188
H.1	Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la vitesse de rotation de l'outil - <i>image extraite de [93]</i> -	189
H.2	Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la vitesse de rotation de la pression exercée sur la silice - <i>image extraite de [93]</i> -	190
H.3	Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la taille moyenne des grains abrasifs - <i>image extraite de [93]</i> -	190
H.4	Évolution de la profondeur max de SSD et du taux d'enlèvement de matière en fonction de la concentration du <i>slurry</i> - <i>image extraite de [93]</i> -	191
H.5	Évolution de la pente des évolutions des profondeurs max de SSD par rapport à la concentration du <i>slurry</i> en fonction du diamètres des grains abrasifs - <i>image extraite de [93]</i> -	191
I.1	Illustration de la fonction de détection d'agréats	194
J.1	Copie puis translation d'un domaine discret suivant une condition périodique	197
J.2	Mise en oeuvre d'une condition périodique dans un calcul DEM	198
J.3	Subdivision du domaine	201
J.4	Subdivision du domaine pour une double périodicité	204

K.1	Illustration 2D de la fonction de reconnaissance des surfaces externes	207
-----	--	-----

Liste des tableaux

1.1	Tableau synthétique des différentes relations $c_{max} = f(R_t)$	12
1.2	Synthèse de l'influence des paramètres du procédé sur la génération de SSD et du taux d'enlèvement de matière pour des opérations de surfacage par abrasifs libres. Résultats extraits de [93].	13
1.3	Caractéristiques mécaniques de la silice S312 à 20 °C	16
3.1	Influence de ν_μ sur les paramètres macroscopiques ν_M et E_M	73
3.2	Tableau synthétique des influences des paramètres microscopiques E_μ et $\tilde{r}_\mu$ sur les paramètres macroscopiques E_M et ν_M	73
3.3	Valeurs des paramètres élastiques macroscopiques de la silice	74
3.4	Valeurs, obtenues par calibration de la silice, des paramètres élastiques microscopiques des liaisons cohésives <i>poutres</i>	74
3.5	Comparaison de résultats numériques et théoriques pour des essais de traction, flexion et torsion	76
3.6	Comparaison des modèles numériques et théoriques pour des essais dynamiques de traction, flexion, torsion et choc	82
3.7	Comparaison des résultats de contrainte macroscopique à rupture pour des essais de traction, flexion et torsion	87
4.1	Synthèse des influences des paramètres du procédé sur la profondeur maximale de SSD	96
4.2	Les différents modèles de contact suivant les différents types d'éléments discrets	103
4.3	Intégration de l'amortissement dans les différents modèles d'interaction	106
4.4	Valeurs des paramètres de précharge	111
4.5	Valeurs des vitesses d'avance de l'outil	112
4.6	Valeurs des dimensions des abrasifs	112
4.7	Valeurs des paramètres de concentration	113
4.8	Vue synthétique des différentes conditions de doucissage	114
4.9	Synthèse des résultats des profondeurs maximales de SSD, des rugosités et de ratios associés	133
4.10	Synthèse des influences des paramètres du procédé sur la profondeur maximale de SSD suivant les observations expérimentales et les simulations numériques	137
A.1	Table de multiplication des quaternions unités	147
C.1	Paramètres de l'étude « essai de torsion 1D »	171
C.2	Résultats des simulations « essai de torsion 1D »	172

Bibliographie

- [1] David Abrahams and Aleksey Gurtovoy. *C++ Template Metaprogramming : Concepts, Tools, and Techniques from Boost and Beyond (C++ in Depth Series)*. Addison-Wesley Professional, 2004.
- [2] A. Alexandrescu. *Modern C++ design : generic programming and design patterns applied*. C++ in-depth series. Addison-Wesley, 2001.
- [3] O.L. Anderson. Effect of pressure on glass structure. *Journal of Applied Physics*, 27(8) :943–949, 1956. cited By (since 1996) 0.
- [4] A.S. Argon. Plastic deformation in metallic glasses. *Acta Metallurgica*, 27(1) :47–58, 1979. cited By (since 1996) 733.
- [5] B. R. B. R. Lawn. Fracture and deformation in brittle solids : A perspective on the issue of scale. *Journal of Material Research*, 19 :22–29, 2003.
- [6] G. T. Beilby. Surface flow in crystalline solids under mechanical disturbance. *Royal Society of London Proceedings Series I*, 72 :218–225, 1903.
- [7] H. Bercegol, A. Boscheron, J.-M. Di-Nicola, E. Journot, L. Lamaignère, J. Neauport, and G. Razě. Laser damage phenomena relevant to the design and operation of an icf laser driver. *Journal of Physics : Conference Series*, 112(3) :032013, 2008.
- [8] H. Bercegol, P. Grua, D. Hébert, and J.-P. Morreeuw. Progress in the understanding of fracture related laser damage of fused silica. In *Proceedings of SPIE*, volume 6720, pages 672003–1. SPIE, 2007.
- [9] H. Bersini. *L'orienté objet : cours et exercices en UML, Java, C# et C++*. Eyrolles, 2002.
- [10] K. L. Blaedel, J. S. Taylor, and C. J. Evans. *Machining of ceramics and composites*. Marcel Dekker, 1999.
- [11] N. Bloembergen. Role of cracks, pores, and absorbing inclusions on laser induced damage threshold at surfaces of transparent dielectrics. *Appl. Opt.*, 12(4) :661–664, Apr 1973.
- [12] A. Bobet, A. Fakhimi, S. Johnson, J. Morris, F. Tonon, and M. R. Yeung. Numerical models in discontinuous media : Review of advances for rock mechanics applications. *Journal of Geotechnical and Geoenvironmental Engineering*, 135(11) :1547–1561, 2009.
- [13] M. Buijs and K. Korpel-van Houten. A model for lapping of glass. *Journal of Materials Science*, 28(11) :3014–3020, 1993.

- [14] R. W. Cahn and E. J. Kramer. *Material science and technology, vol 9 : glasses and amorphous metals*. VCH, 1991.
- [15] F. Camborde, C. Mariotti, and F. V. Donzé. Numerical study of rock and concrete behaviour by discrete element modelling. *Computers and Geotechnics*, 27(4) :225–247, 2000.
- [16] B. Cambou. *Behaviour of Granular Materials*. Springer, illustrated edition edition, 1998.
- [17] B. Cambou, M. Jean, and F. Radjai. *Micromechanics of Granular Materials*. Wiley, 2009.
- [18] D. W. Camp, M. R. Kozlowski, L. M. Sheehan, M. A. Nichols, M. Dovik, R. G. Raether, and I. M. Thomas. Subsurface damage and polishing compound affect the 355-nm laser damage threshold of fused silica surfaces. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3244 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 356–364, April 1998.
- [19] H. A. Carmona, F. K. Wittel, F. Kun, and H. J. Herrmann. Fragmentation processes in impact of spheres. *Physical Review*, 77(5) :051302, 2008.
- [20] L. Chorel. *Développement d'un outil numérique et d'outils expérimentaux pour la conception de pyromécanismes utilisées en sécurité automobile*. PhD thesis, Université bordeaux 1, 210.
- [21] Lee M. Cook. Chemical processes in glass polishing. *Journal of Non-Crystalline Solids*, 120(1-3) :152–171, 1990.
- [22] A. Coutias and L. Romero. The quaternions with an application to rigid body dynamics. Technical report, University of New Mexico, Albuquerque, 1999.
- [23] P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29 :47–65, 1979.
- [24] G. A. D'Addetta, F. Kun, and E. Ramm. On the application of a discrete model to the fracture process of cohesive granular materials. *Granular Matter*, 4(2) :77–90, 2002.
- [25] G. A. D'Addetta and E. Ramm. A microstructure-based simulation environment on the basis of an interface enhanced particle model. *Granular Matter*, 8(3) :159–174, 2006.
- [26] J. Diebel. Representing attitude : Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University, California, 2006.
- [27] Karel Driesen and Urs Hölzle. The direct cost of virtual function calls in c++. *SIGPLAN Not.*, 31 :306–323, October 1996.
- [28] D. Eberly. *Game physics*. Elsevier, Morgan Kaufmann, 30 Corporate Drive, Suite 400, Burlington, MA 01803 USA, second edition edition, 2010.
- [29] A. Fakhimi and T. Villegas. Application of dimensional analysis in calibration of a discrete element model for rock deformation and fracture. *Rock Mechanics and Rock Engineering*, 40(2) :193–211, 2007.

- [30] M. Feit and A.M. Rubenchik. Initiation growth and mitigation of uv laser induced damage in fused silica. In *Laser-induced damage in optical materials : 2001 : 33rd Annual Boulder Damage Symposium, 1-2 October 2001, Boulder, Colorado : proceedings*, Proceedings of SPIE—the International Society for Optical Engineering, 2001.
- [31] M. D. Feit and A. M. Rubenchik. Influence of subsurface cracks on laser-induced surface damage. In G. J. Exarhos, A. H. Guenther, N. Kaiser, K. L. Lewis, M. J. Soileau, & C. J. Stolz, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5273 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 264–272, June 2004.
- [32] N. Fillot, I. Iordanoff, and Y. Berthier. A granular dynamic model for the degradation of material. *Journal of Tribology*, 126(3) :606–614, 2004.
- [33] N. Fillot, I. Iordanoff, and Y. Berthier. Simulation of wear through mass balance in a dry contact. *Journal of Tribology*, 127(1) :230–237, 2005.
- [34] N. Fillot, I. Iordanoff, and Y. Berthier. Modelling third body flows with a discrete element method—a tool for understanding wear with adhesive particles. *Tribology International*, 40(6) :973–981, 2007. Numerical Simulation Methods in Tribology : possibilities and limitations.
- [35] J. L. Finney. Random packings and the structure of simple liquids. i. the geometry of random close packing. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 319(1539) :pp. 479–493, 1970.
- [36] Florian Fleissner, Timo Gaugele, and Peter Eberhard. Applications of the discrete element method in mechanical engineering. *Multibody System Dynamics*, 18 :81–94, 2007. 10.1007/s11044-007-9066-2.
- [37] J. Fortin, M. Hjiiaj, and G. de Saxcé. An improved discrete element method based on a variational formulation of the frictional contact law. *Computers and Geotechnics*, 29(8) :609–640, 2002.
- [38] J. Fortin, O. Millet, and G. de Saxcé. Numerical simulation of granular materials by an improved discrete element method. *International Journal for Numerical Methods in Engineering*, 62(5) :639–663, 2005.
- [39] D. François. Essais de dureté. *Techniques de l'ingénieur*, (M4160), 2005.
- [40] A. Franks. Materials problems in the production of high quality optical surfaces. *Materials Science and Engineering*, 19(2) :169–183, 1975.
- [41] F. Y. Génin, A. Salleo, T. V. Pistor, and L. L. Chase. Role of light intensification by cracks in optical breakdown on surfaces. *J. Opt. Soc. Am. A*, 18(10) :2607–2616, Oct 2001.
- [42] A. J. Glass and A. H. Guenther. Laser induced damage of optical elements—a status report. *Appl. Opt.*, 12(4) :637–649, Apr 1973.
- [43] Maurice Godet. The third-body approach : A mechanical view of wear. *Wear*, 100(1-3) :437–452, 1984.
- [44] Keishi Gotoh and J. L. Finney. Statistical geometrical approach to random packing density of equal spheres. *Nature*, 252 :202–205, 1974.

- [45] A. A. Griffith. The phenomena of rupture and flow in solids. *Philosophical Transactions of the Royal Society of London*, 221 :163–198, 1920.
- [46] Gabe M. Guss, Isaac L. Bass, Richard P. Hackel, Christian Mailhiot, and Stavros G. Demos. In situ monitoring of surface postprocessing in large-aperture fused silica optics with optical coherence tomography. *Appl. Opt.*, 47(25) :4569–4573, Sep 2008.
- [47] J. Happel and H. Brenner. *Low Reynolds number hydrodynamics : with special applications to particulate media*. Mechanics of fluids and transport processes. M. Nijhoff, 1983.
- [48] P. Paul Hed and David F. Edwards. Optical glass fabrication technology. 2 : Relationship between surface roughness and subsurface damage. *Appl. Opt.*, 26(21) :4677–4680, Nov 1987.
- [49] S. Hentz, L. Daudeville, and F. V. Donzé. Identification and validation of a discrete element model for concret. *Journal of Engineering Mechanics*, 130(6) :709–719, 2004.
- [50] S. Hentz, F. V. Donzé, and L. Daudeville. Discrete element modelling of concrete submitted to dynamic loading at high strain rates. *Computers & Structures*, 82(29-30) :2509–2524, 2004.
- [51] H. J. Herrmann. Proceedings of the nato advanced study institute, cargèse, corsica, france, 18-31 july, 1988. In *Random Fluctuations and Pattern Growth : Experiments and Models Random Fluctuations and Pattern Growth : Experiments and Models*, volume 157 of *NATO Science Series E*, pages 149–160. Stanley, Harry Eugene ; Ostrowsky, N., 1988.
- [52] II R. A. House, J. R. Bettis, and A. H. Guenther. Subsurface structure and laser damage threshold. *IEEE Journal of Quantum Electronics*, 13 :363–+, May 1977.
- [53] A. Ibrahimbegovic and A. Delaplace. Microscale and mesoscale discrete models for dynamic fracture of structures built of brittle material. *Computers & Structures*, 81(21) :1255–1265, 2003.
- [54] I. Iordanoff, A. Battentier, J. Neauport, and J.L. Charles. A discrete element model to investigate sub-surface damage due to surface polishing. *Tribology International*, 41(11) :957–964, 2008.
- [55] I. Iordanoff, N. Fillot, and Y. Berthier. Numerical study of a thin layer of cohesive particles under plane shearing. *Powder Technology*, 159(1) :46–54, 2005.
- [56] I. Iordanoff and M. M. Khonsari. Granular lubrication : Toward an understanding of the transition between kinetic and quasi-fluid regime. *Journal of Tribology*, 126(1) :137–145, 2004.
- [57] I. Iordanoff, B. Seve, and Y. Berthier. Solid third body analysis using a discrete approach : Influence of adhesion and particle size on macroscopic properties. *Journal of Tribology*, 124(3) :530–538, 2002.
- [58] A. Jagota and S. Bennison. Spring-network and finite-element models for elasticity and fracture. In Kamal Bardhan, Bikas Chakrabarti, and Alex

- Hansen, editors, *Non-Linearity and Breakdown in Soft Condensed Matter*, volume 437 of *Lecture Notes in Physics*, pages 186–201. Springer Berlin / Heidelberg, 1994. 10.1007/3-540-58652-0_37.
- [59] M. Jebahi. *Couplages modèles discrets - modèles continus pour les simulations du comportement des matériaux sous sollicitation dynamique*. PhD thesis, Arts & Métiers ParisTech, I2M-DuMAS, UMR 5295 CNRS, F-33405, Talence, France, 2010.
- [60] J. F. Jerier, B. Harthong, D. Imbault, F.V. Donzé, and P. Dorémus. Numerical simulations of isostatic and die compaction of powder by the discrete element method. In M. Nakagawa & S. Luding, editor, *American Institute of Physics Conference Series*, volume 1145 of *American Institute of Physics Conference Series*, pages 457–460, 2009.
- [61] Jean-François Jerier, Didier Imbault, Frederic-Victor Donze, and Pierre Dorémus. A geometric algorithm based on tetrahedral meshes to generate a dense polydisperse sphere packing. *Granular Matter*, 11 :43–52, 2009. 10.1007/s10035-008-0116-0.
- [62] Jean-François Jerier, Vincent Richefeu, Didier Imbault, and Frédéric-Victor Donzé. Packing spherical discrete elements for large scale simulations. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28) :1668–1676, 2010.
- [63] Hank H. Karow. *Fabrication methods for precision optics*. Wiley-Interscience, 2004. ISBN 0-471-70379-6.
- [64] Alan C. Kay. The early history of smalltalk. *SIGPLAN Not.*, 28(3) :69–95, March 1993.
- [65] K. Ken-Ichi. Distribution of directional data and fabric tensors. *International Journal of Engineering Science*, 22(2) :149–164, 1984.
- [66] Sandeep Koranne. Standard libraries. In *Handbook of Open Source Tools*, pages 105–111. Springer US, 2011. 10.1007/978-1-4419-7719-9_4.
- [67] J. Kozicki and F.V. Donzé. A new open-source software developed for numerical simulations using discrete modeling methods. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50) :4429–4443, 2008.
- [68] Christian F. Kranenberg and Kenneth C. Jungling. Subsurface damage identification in optically transparent materials using a nondestructive method. *Appl. Opt.*, 33(19) :4248–4253, Jul 1994.
- [69] F. Kun and H. J. Herrmann. A study of fragmentation processes using a discrete element method. *Computer Methods in Applied Mechanics and Engineering*, 138(1-4) :3–18, 1996.
- [70] L. Lamaignère, R. Courchinoux, J.-C. Poncetta, and H. Bercegol. L’endommagement des optiques du lmj : problématique, mécanismes et métrologie. volume UVX 2010 - 10e Colloque sur les Sources Cohérentes et Incohérentes UV, VUV et X ; Applications et Développements Récents, pages 99–105, 2011.

- [71] Romain Laniel, Pierre Alart, and Stéphane Pagano. From discrete to continuous numerical identification of a geomaterial with an internal length. *Computer Methods in Applied Mechanics and Engineering*, 199(1-4) :113–122, 2009.
- [72] Brian Lawn. *Fracture of brittle solids - second edition*. Cambridge University Press, 1993.
- [73] Woo Hyong Lee and Morris Chang. A study of dynamic memory management in c++ programs. *Computer Languages, Systems & Structures*, 28(3) :237–272, 2002.
- [74] Ching-Lung Liao, Ta-Peng Chang, Dong-Hwa Young, and Ching S. Chang. Stress-strain relationship for granular materials based on the hypothesis of best fit. *International Journal of Solids and Structures*, 34(31-32) :4087–4100, 1997.
- [75] S.B. Lippman. *C++ gems*. SIGS reference library series. Cambridge University Press, 1998.
- [76] Kristin Lochmann, Luc Oger, and Dietrich Stoyan. Statistical analysis of random sphere packings with variable radius distribution. *Solid State Sciences*, 8(12) :1397–1413, 2006.
- [77] M.K. Loukides and A. Oram. *Programming with GNU software*. A Nutshell handbook. O'Reilly & Associates, 1997.
- [78] Boris D. Lubachevsky and Frank H. Stillinger. Geometric properties of random disk packings. *Journal of Statistical Physics*, 60 :561–583, 1990.
- [79] S. A. Magnier and F. V. Donzé. Numerical simulations of impacts using a discrete element method. *Mechanics of Cohesive-frictional Materials*, 3(3) :257–276, 1998.
- [80] Laurent Mahéo, Vincent Grolleau, and Gérard Rio. Damping efficiency of the tchamwa-wielgosz explicit dissipative scheme under instantaneous loading conditions. *Comptes Rendus Mécanique*, 337(11-12) :722–732, 2009.
- [81] J. P. Marioge. *Surfaces optiques, methode de fabrication et de controle, recherches*. P. Bozec, 2000.
- [82] C. L. Martin, D. Bouvard, and S. Shima. Study of particle rearrangement during powder compaction by the discrete element method. *Journal of the Mechanics and Physics of Solids*, 51(4) :667–693, 2003.
- [83] Joseph Menapace, Bernie Penetrante, Phil Miller, Tom Parham, Mike Nichols, John Peterson, and Don Golini. Combined advanced finishing and uv-laser conditioning for producing uv-damage-resistant fused silica optics. In *Optical Fabrication and Testing*, page OMB4. Optical Society of America, 2002.
- [84] Bertrand Meyer. Applying "design by contract". *IEEE*, 25, Issue 10 :40–51, 1992.
- [85] P. E. Miller, T. Suratwala, J. D. Bude, T. A. Laurence, N. Shen, W. A. Steele, M. Feit, J. Menapace, and L. Wong. Laser damage precursors in fused silica. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7504 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, October 2009.

- [86] Vahé Ter Minassian. Lmj, un méga outil de recherche. *Les défis du cea*, (151) :6–11, Juin 2010.
- [87] B. K. Mishra and Raj K. Rajamani. The discrete element method for the simulation of ball mills. *Applied Mathematical Modelling*, 16(11) :598–604, 1992.
- [88] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1) :131–150, 1999.
- [89] J.J. Moreau, P.D. Panagiotopoulos, and International Centre for Mechanical Sciences. *Nonsmooth mechanics and applications*. Courses and lectures - International Centre for Mechanical Sciences. Springer-Verlag, 1988.
- [90] A. Munjiza. *The combined Finite-Discrete Element Methode*. John Wiley & Sons, 2004.
- [91] J. Neauport, C. Ambard, P. Cormont, N. Darbois, J. Destribats, C. Luitot, and O. Rondeau. Subsurface damage measurement of ground fused silica parts by hf etching techniques. *Opt. Express*, 17(22) :20448–20456, Oct 2009.
- [92] J. Neauport, P. Cormont, P. Legros, C. Ambard, and J. Destribats. Imaging subsurface damage of grinded fused silica optics by confocal fluorescence microscopy. *Opt. Express*, 17(5) :3543–3554, Mar 2009.
- [93] J. Neauport, J. Destribats, C. Maunier, C. Ambard, P. Cormont, B. Pintault, and O. Rondeau. Loose abrasive slurries for optical glass lapping. *Appl. Opt.*, 49(30) :5736–5745, Oct 2010.
- [94] J. Neauport, L. Lamaignère, F. Pilon, and J.-C. Birolleau. Polishing-induced contamination of fused silica optics and laser induced damage density at 351 nm. *Opt. Express*, 13(25) :10163–10171, Dec 2005.
- [95] J. Neauport, D. Valla, J. Duchesne, P. Bouchut, L. Lamaignère, J. Bigarre, and N. Daurios. Building high-damage-threshold surfaces at 351 nm. In *Proceedings of SPIE*, volume 5252, pages 131–139. SPIE, 2004.
- [96] A. Needleman. A continuum model for void nucleation by inclusion debonding. *Journal of Applied Mechanics*, 54(3) :525–531, 1987.
- [97] M. Ortiz. Microcrack coalescence and macroscopic crack growth initiation in brittle solids. *International Journal of Solids and Structures*, 24(3) :231–250, 1988.
- [98] M. Ostoja-Starzewski. Lattice models in micromechanics. *Applied Mechanics Reviews*, 55 :35–60, 2002.
- [99] J. Pommier Y. Renard M. Salaun P. Laborde. Methodes xfem d’ordre supérieur en mecanique de la rupture. Technical report, Laboratoire MIP, ENSICA, 2002.
- [100] A. Pandolfi, P. Krysl, and M. Ortiz. Finite element simulation of ring expansion and fragmentation : The capturing of length and time scales through cohesive models of fracture. *International Journal of Fracture*, 95 :279–297, 1999. 10.1023/A :1018672922734.
- [101] T. Penders. *Introduction à UML*. OEM, 2002.

- [102] Frederic Lerouge Marc Rossetto Pierre Agati. *Resistance des materiaux*. Dunod, 2001.
- [103] Scott M. Pike, Bruce W. Weide, and Joseph E. Hollingsworth. Checkmate : cornering c++ dynamic memory errors with checked pointers. *SIGCSE Bull.*, 32 :352–356, March 2000.
- [104] T. Pöschel and S. Luding. *Granular Gases*, volume Vol. 564 of *Lecture Notes in Physics*. Pöschel, Thorsten ; Luding, Stefan, 2001.
- [105] T. Pöschel and T. Schwager. *Computational granular dynamics*. Springer, 2005.
- [106] D.O. Potyondy and P. A. Cundall. A bonded-particle model for rock. *International Journal of Rock Mechanics and Mining Sciences*, 41(8) :1329–1364, 2004. Rock Mechanics Results from the Underground Research Laboratory, Canada.
- [107] F W Preston. The theory and design of plate glass polishing machines. *Journal of the society of glass technology*, 11 :214–256, 1927.
- [108] Joseph A. Randi, John C. Lambropoulos, and Stephen D. Jacobs. Sub-surface damage in some single crystalline optical materials. *Appl. Opt.*, 44(12) :2241–2249, Apr 2005.
- [109] James R. Rice. Dislocation nucleation from a crack tip : An analysis based on the peierls concept. *Journal of the Mechanics and Physics of Solids*, 40(2) :239–271, 1992.
- [110] D. Richard, I. Iordanoff, Y. Berthier, M. Renouf, and N. Fillot. Friction coefficient as a macroscopic view of local dissipation. *Journal of Tribology*, 129(4) :829–835, 2007.
- [111] E. Rougier, A. Munjiza, and N. W. M. John. Numerical comparison of some explicit time integration schemes used in dem, fem/dem and molecular dynamics. *International Journal for Numerical Methods in Engineering*, 61(6) :856–879, 2004.
- [112] P. Rozycki, N. Moes, E. Bechet, and C. Dubois. X-fem explicit dynamics for constant strain elements to alleviate mesh constraints on internal or external boundaries. *Computer Methods in Applied Mechanics and Engineering*, 197(5) :349–363, 2008. Enriched Simulation Methods and Related Topics.
- [113] A. Salleo, F. Y. Genin, J. Yoshiyama, C. J. Stolz, and M. R. Kozlowski. Laser-induced damage of fused silica at 355 nm initiated at scratches. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3244 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 341–347, April 1998.
- [114] E. Schlangen and E. J. Garboczi. New method for simulating fracture using an elastically uniform random geometry lattice. *International Journal of Engineering Science*, 34(10) :1131–1144, 1996.
- [115] E. Schlangen and E.J. Garboczi. Fracture simulations of concrete using lattice models : Computational aspects. *Engineering Fracture Mechanics*, 57(2-3) :319–332, 1997.

- [116] E. Schlangen and J. G. M. van Mier. Experimental and numerical analysis of micromechanisms of fracture of cement-based composites. *Cement and Concrete Composites*, 14(2) :105–118, 1992. Special Issue on Micromechanics of Failure in Cementitious Composites.
- [117] E. Schlangen and J. G. M. van Mier. Simple lattice model for numerical simulation of fracture of concrete materials and structures. *Materials and Structures*, 25(9) :534–542, 1992.
- [118] L. Scholtès, B. Chareyre, F. Nicot, and F. Darve. Micromechanics of granular materials with capillary effects. *International Journal of Engineering Science*, 47(1) :64–75, 2009.
- [119] A. A. Shabana. *Vibration of discrete and continuous systems*. Mechanical engineering series. Springer, second edition, 1997.
- [120] J. Shen, S. Liu, K. Yi, H. He, J. Shao, and Z. Fan. Subsurface damage in optical substrates. *Optik - International Journal for Light and Electron Optics*, 116(6) :288–294, 2005.
- [121] Wenjie Shiu, Frederic Victor Donze, and Laurent Daudeville. Discrete element modelling of missile impacts on a reinforced concrete target. *International Journal of Computer Applications in Technology*, 34(1) :33–41, 2009.
- [122] Joerg Steinert, Stefan Glied, Andreas Wuttig, Angela Duparre, and Horst Truckenbrodt. Advanced methods for surface and subsurface defect characterization of optical components. 4099(1) :290–298, 2000.
- [123] B. Stroustrup. *The C++ programming language* :. Addison-Wesley, 2000.
- [124] T. Suratwala, P. Davis, L. Wong, P. Miller, M. Feit, J. Menapace, R. Steele, and D. Walmer. Sub-surface mechanical damage distributions during grinding of fused silica. *Journal of Non-Crystalline Solids*, 352 :5601–5617, 2006.
- [125] Y. Tan, D. Yang, and Y. Sheng. Discrete element method (dem) modeling of fracture and damage in the machining process of polycrystalline sic. *Journal of the European Ceramic Society*, 29(6) :1029–1037, 2009.
- [126] F. A. Tavarez and M. E. Plesha. Discrete element method for modelling solid and particulate materials. *International Journal for Numerical Methods in Engineering*, 70 :379–404, 2007.
- [127] E.W. Taylor. Plastic deformation of optical glass. *Nature*, 163(4139) :323, 1949. cited By (since 1996) 13.
- [128] B. Tchamwa. *Contribution à l'étude des méthodes d'intégration directe explicites en dynamique non linéaire des structures*. PhD thesis, Ecole Centrale de Nantes, 1997.
- [129] Aleta A. Tesar, Norman J. Brown, John R. Taylor, and Christopher J. Stolz. Subsurface polishing damage of fused silica : nature and effect on laser damage of coated surfaces. 1441(1) :154–172, 1991.
- [130] Todd Veldhuizen. *Expression templates*, pages 475–487. SIGS Publications, Inc., New York, NY, USA, 1996.
- [131] Todd L. Veldhuizen. Scientific computing : C++ versus fortran, focuses on optimizations for small linear algebra objects. *Dr. Dobb's Journal*, 1997.

- [132] Todd L. Veldhuizen. C++ templates as partial evaluation. Technical report, Indiana University Computer Science Department, 1999.
- [133] J. Wang, Y. Li, J. Han, Q. Xu, and Y. Guo. Evaluating subsurface damage in optical glasses. *ournal of the European Optical Society – Rapid Publications*, 6(11001), 2011.
- [134] D. Weatherley. *ESyS-Particle v2.0 user’s guide*, 2009.
- [135] J. Wong, J. L. Ferriera, E. F. Lindsey, D. L. Haupt, I. D. Hutcheon, and J. H. Kinney. Morphology and microstructure in fused silica induced by high fluence ultraviolet 3w (355 nm) laser pulses. *Journal of Non-Crystalline Solids*, 352(3) :255–272, 2006.
- [136] Zhang Xiu-li, Jiao Yu-yong, and Zhao Jian. Simulation of failure process of jointed rock. *Journal of Central South University of Technology*, 15 :888–894, 2008.
- [137] J. Zarzycki. Propriétés mécaniques des verres. *Revue de physique appliquée*, 12 :789–796, 1977.
- [138] Yiyang Zhou, Paul D. Funkenbusch, David J. Quesnel, Donald Golini, and Arne Lindquist. Effect of etching and imaging mode on the measurement of subsurface damage in microground optical glasses. *Journal of the American Ceramic Society*, 77(12) :3277–3280, 1994.
- [139] O. C. Zienkiewicz and R. L. Taylor. *The finite element method for solid and structural mechanics, 6th edition*. Elsevier, butterworth-heinemann edition, 2005.