Université Pierre et Marie Curie – Paris VI

GranuLab : un système d'aide à la découverte scientifique pour la physique des milieux granulaires

THÈSE

présentée et soutenue publiquement le 24 Janvier 2002

pour obtenir le titre de

Docteur de l'Université Pierre et Marie Curie - Paris VI

(spécialité Informatique, Télécommunications et Electronique)

par

Laurent Breton

Composition du jury

Rapporteurs :	Antoine Cornuéjols Jean-Pierre Müller
Examinateurs :	Eric Clément Jean-Gabriel Ganascia Mourad Oussalah
Invités :	Alexis Drogoul Jérôme Euzenat
Directeur :	Jean-Daniel Zucker

Laboratoire d'Informatique de Paris VI

Remerciements

Je tiens à exprimer ma profonde gratitude à Jean-Daniel Zucker, mon directeur de thèse, pour ces trois années de collaboration active. Sa confiance et son optimisme m'ont souvent été d'un grand recours pour mener à bien les objectifs que nous nous étions fixés.

Je n'oublierais pas les longues discussions que nous avons eues avec Eric Clément sur la physique des milieux granulaires, souvent autour d'une pizza. Je lui suis gré de la patience et de l'énergie dont il a su faire preuve au quotidien lors de mon séjour de plusieurs mois au sein de son équipe.

Cette thèse n'aurait pas vu le jour sans la collaboration entre Jean-Daniel Zucker, Eric Clément et leurs équipes respectives. Je tiens à les remercier pour m'avoir permis de traiter un sujet aussi passionnant et de contribuer ainsi à la recherche dans le domaine de la physique des milieux granulaires.

Je remercie également Philippe Claudin, membre de la "Jeune Equipe Dynamique" d'Eric Clément, sans qui je n'aurais pas pu réaliser toutes les courbes et analyses dont j'ai eu besoin au cours de ma thèse.

Je tiens également à remercier Antoine Cornuéjols et Jean-Pierre Müller d'avoir accepté avec enthousiasme d'être les rapporteurs de cette thèse. Je remercie Mourad Chabane Oussalah de s'être porté volontaire pour présider le jury de ma soutenance, Jérôme Euzenat, Alexis Drogoul et Jean-Gabriel Ganascia pour avoir été les examinateurs de cette thèse.

Je remercie Jean-Gabriel Ganascia de m'avoir accueilli au sein de son équipe ACASA ainsi que les membres de cette équipe avec qui j'ai pu discuter de nos problèmes respectifs. Je tiens particulièrement à remercier Bruno Robisson pour son aide à modéliser les équations du modèle physique ainsi que Mélanie Courtine et Valérie Renault pour leur soutien moral et gastronomique. Je tiens à remercier Valérie pour sa collaboration à l'utilisation de XML alors que cette technologie, très récente, nécessitait une conception nouvelle de représentation et de traitement des données. Je remercie également Mélanie pour ses commentaires constructifs et rigoureux au cours de ses nombreuses relectures de ce manuscrit.

Je tiens à féliciter Anneli Lénica et Cécile Le Pape qui ont implémenté le noyau de GranuLab dans le cadre d'un stage de Maîtrise.

Je remercie également Anne Liret et Philippe Codognet pour leurs conseils dans le domaine de la résolution de problèmes par contraintes.

Je remercie Estelle pour ses nombreux coups de fil et Sébastien pour ses judicieux conseils dans les domaines de la finance et de la bière mexicaine.

Je remercie tous ces grains de lumière grâce à qui mes nuits ne furent pas tout à fait noires mais parsemées de lueurs éclairant mon chemin d'espoir.

Enfin, je tiens à remercier ma famille qui m'a supporté ces trois années de thèse. Je dédie cette thèse à Noémie, notre premier enfant, qui a vu le jour au cours de ma thèse et m'a permis de prendre quotidiennement du recul par rapport à ma thèse.

A Noémie, Murielle et au futur bébé.

Table des matières

Chapitre 2
Problèmes de physique granulaire et motivations pour la simulation numérique

Chapitre 3
Modélisation du problème physique et algorithmes classiques

Chapitre 4
GranuSolve : Une approche Multi-Agent pour la «résolution de tas de sable»

Chapitre 5
Vers une modélisation du processus de découverte en physique granulaire

Chapitre 7
Expérimentations avec GranuLab : étude détaillée de la «fonction de Green»

Conclusion et perspectives

Bibliographie

Annexes

Table des figures

Liste des tableaux

Liste des algorithmes

Introduction

Domaine de cette thèse

En intelligence artificielle, les chercheurs tentent de créer des systèmes informatiques capables de simuler l'esprit humain pour des activités intellectuelles non triviales. L'*aide à la découverte scientifique* est un domaine de l'intelligence artificielle qui s'inscrit naturellement dans ce projet. Le but des systèmes d'aide à la découverte est de simuler l'activité de recherche d'un expert d'un domaine particulier. Pour cette tâche, nous pouvons distinguer au moins deux types d'activités intellectuelles. La première, souvent rationnelle, est liée au domaine scientifique étudié. La seconde, de plus haut niveau, est liée au processus général de la découverte.

Des systèmes tentant de reproduire ce processus ont permis de redécouvrir des lois physiques connues telles que la loi de Boyle des gaz parfaits ou la troisième loi de Kepler des périodes orbitales des planètes [Langley *et al.* 1987].

Les premiers systèmes qui ont vu le jour avec l'avènement de la découverte scientifique comme un domaine à part entière de l'intelligence artificielle permettaient de démontrer de nouveaux théorèmes à partir d'une théorie donnée en entrée du système. Ces systèmes fonctionnaient principalement en boucle fermée jusqu'à saturation des règles ou théorèmes que le système pouvait inférer [Lenat 1977a].

Par la suite, un grand nombre de systèmes basés sur des données d'expérimentations réelles ont vu le jour. Le but de ces systèmes est généralement d'inférer des lois symboliques à partir de données issues d'expérimentations réelles. Ainsi, de nombreux systèmes se sont attachés à redécouvrir des lois dans les domaines de la physique et de la chimie [Langley 1998, Valdés-Pérez 1995d].

Plus récemment, des systèmes de fouille de données de troisième génération permettent de modéliser et de représenter visuellement le processus de fouille de données mené par l'expert [Potts 1997, Khabaza et Shearer 1995]. Ces systèmes ne proposent pas de découvrir directement des règles ou des théorèmes à partir des données mais fournissent à l'expert les outils nécessaires à ses recherches. Ces systèmes introduisent un haut niveau d'interaction avec l'expert. Ainsi, ces systèmes ont une approche du type *collaboration homme-machine*, initiée notemment par Raùl Valdés-Pérez [Valdés-Pérez 1999]. Avec cette approche, la machine conserve son rôle de calculateur et l'expert son rôle de chercheur en guidant lui-même la recherhe à l'aide de ses propres heuristiques. Les données en entrée de ces systèmes sont généralement issues de bases de données de tailles importantes.

Aujourd'hui, il serait intéressant de pouvoir proposer des sytèmes d'aide à la découverte scientifique basés sur des données virtuelles issues de simulations numériques et qui s'inspireraient des systèmes décrits ci-dessus en proposant notamment un haut niveau d'interaction avec l'expert. Un tel système devrait pouvoir modéliser et représenter de manière explicite le processus de recherche d'un expert du domaine.

Problématiques et enjeux

La physique des milieux granulaires est un domaine de la physique mécanique en plein essor depuis quelques années. Après plusieurs siècles d'études de la mécanique des solides, la physique s'est intéressée à la mécanique des fluides, puis à l'extension de la mécanique newtonienne avec la mécanique quantique. Aujourd'hui, un nouveau défi est lancé à la communauté de chercheurs en physique. Il s'agit de l'étude de systèmes composés d'un grand nombre de particules élémentaires (grains corps célestes, bulles, etc.), un nouveau domaine appellé la *physique des systèmes complexes*. L'existence de nombreuses interactions entre ces particules élémentaires aux comportements individuels, pourtant bien connus, rend difficile la compréhension et la modélisation de ces systèmes physiques dans leur ensemble.

Il y a quelques années, Pierre-Gilles de Gennes a lancé un appel à la communauté des chercheurs en physique mécanique pour aborder le problème de la physique des milieux granulaires. De Gennes décrit le domaine des milieux granulaires comme étant un exemple simple de systèmes physiques complexes [de Gennes 1995, de Gennes 1999]. Effectivement, les empilements de grains posent de nouveaux types de problèmes aux physiciens. Car, contrairement à l'étude d'un solide, les *granulaires* se comportent de façon apparemment imprévisible.

Le but de cette thèse est donc de proposer un *système d'aide à la découverte scientifique appliqué à la physique des milieux granulaires*. L'intérêt d'appliquer un tel système au domaine de la physique granulaire repose sur la nouveauté de ce domaine. En effet, un grand nombre de lois de comportements macroscopiques des empilements de grains restent certainement à découvrir. Ainsi, un tel système pourrait aider à découvrir des lois qui soient tout à la fois *nouvelles*, *intéressantes*, *plausibles* et *intélligibles* [Valdés-Pérez 1999].

Cependant, les chercheurs de ce domaine rencontrent des difficultés d'ordres expérimentales et numériques pour mener à bien leurs recherches. Une difficulté tient à la réalisation d'expérimentations réelles au moins partiellement reproductibles. Des mesures macroscopiques de contraintes effectuées sur des empilements réels ont montré que les fluctuations des forces de contact sont de l'ordre de la valeur moyenne [Liu *et al.* 1995, Mueth *et al.* 1998]. De plus, des problèmes liés à la taille finie des capteurs, au calibrage des grains, etc. , rendent les mesures difficiles. D'une expérimentation à l'autre, les résultats observés peuvent être très différents. Ainsi, les chercheurs en physique granulaire ne cherchent pas à modéliser un tas mais plutôt à comprendre le *comportement moyen* d'un empilement à l'équilibre statique. Ces difficultés pour la réalisation d'expérimentations réelles les ont conduit naturellement à envisager la simulation numérique d'empilements de grains à l'équilibre statique pour remédier à ces difficultés.

Mais, le nombre d'interactions entre particules combiné au degré de liberté de chacunes d'entre elles rendent difficiles la simulation de tels empilements. Actuellement, les algorithmes de simulation classiquement employés par la communauté de chercheurs de ce domaine ne permettent pas de résoudre des tas de plus de deux cents grains en largeur. De plus, les performances en temps de calcul de ces algorithmes interdisent une véritable étude statistique des phénomènes macroscopiques observés.

La réalisation d'un environnement d'aide à la découverte appliqué à la physique des milieux granulaires requiert, dans un premier temps, de concevoir un solveur capable de résoudre de grands tas en des temps raisonnables. Cette première étape s'avère nécessaire si nous voulons que notre système puisse effectivement aider à la découverte de comportements macroscopiques fiables.

Approches proposées

La première étape de notre travail consiste à concevoir un algorithme de simulation d'empilements de grains rapide et permettant d'aborder la résolution de grands tas (plusieurs milliers de grains). Pour cela, nous proposons à travers GranuSolve une approche multi-agent pour résoudre des tas de sable à l'équilibre statique. Nous considérons un grain comme un agent essentiellement réactif dont le but est de trouver son équilibre en fonction de ses voisins. Un empilement est donc à l'équilibre statique lorsque tous les *agents-grains* qui le composent sont eux-mêmes à l'équilibre.

Cet algorithme repose sur deux méthodes de conception de systèmes multi-agents. La première est la méthode d'Eco-Résolution que nous utilisons pour résoudre un tas composé d'agents-grains [Ferber 1990, Drogoul *et al.* 1991, Drogoul et Dubreuil 1991]. La seconde est la méthodologie de conception de systèmes multi-agents Cassiopée qui nous permet de modéliser un agent-grain en couches et rôles successifs [Collinot et Drogoul 1996].

Les différentes versions de notre solveur sont données pour illustrer et justifier l'ajout de connaissances du domaine. Ces connaissances sont représentées sous la forme d'heuristiques pour l'Eco-Résolution du tas et de comportements de résolution pour les agents-grains. La validation, les performances et les limitations du système sont discutées pour chacune des versions implémentées.

Les performances de notre solveur nous ont permis, dans un deuxième temps, de construire GranuLab : un environnement d'aide à la découverte appliqué à la physique des milieux granulaires. Cet environnement est construit sur une arhcitecture componentielle traitant des flux de données au format XML. Dans cet environnement, notre solveur GranuSolve ne représente qu'un seul composant parmi un ensemble d'opérateurs de transformations, de calculs statistiques et d'affichage de courbes et de graphiques.

GranuLab permet de décrire, de mémoriser et de visualiser une série d'expérimentations menées en physique des milieux granulaires, à partir de données issues d'un solveur de tas. Les traitements appliqués aux données issues du solveur sont décrites à l'aide d'un nombre restreint de composants élémentaires réalisant chacun une opération. Les données en sortie des composants sont alors connectées aux entrées d'autres composants pour réaliser au final une chaîne de traitements qui peut être complexe. GranuLab permet ainsi d'exécuter une expérimentation virtuelle en gérant les flux de données entre composant ainsi que la réalisation des traitements eux-mêmes.

Enfin, nous proposons une expérimentation virtuelle réalisée avec GranuLab : l'étude de la fonction de réponse du système à une surcharge localisée. Cette étude illustre l'intérêt de notre approche et des choix technologiques effectués.

Plan

Dans le chapitre 1, nous donnons une classification originale des systèmes d'aide à la découverte scientifique appliqués à la physique. L'aide à la découverte scientifique est d'abord présentée comme un champ de l'Intelligence Artificielle. Les notions d'heuristiques et de données en entrée d'un tel système y sont décrites. Puis, nous présentons ce que nous pensons être quatre grandes familles de systèmes d'aide à la découverte scientifique sous l'angle original du niveau d'interaction explicite avec l'expert. Les deux premières familles regroupent les systèmes autonomes basés sur une théorie et ceux basés sur des résultats d'expérimentations réelles. Ces systèmes n'offrent que peu d'interactions avec l'utilisateur. Les systèmes interactifs tels que ceux réalisant de la fouille de données sont ensuite présentés. Enfin, les systèmes interactifs basés sur des données d'expérimentations virtuelles sont décrits. Notre système GranuLab s'inscrit dans cette dernière famille de systèmes d'aide à la découverte scientifique.

Le domaine d'application de GranuLab est la physique des milieux granulaires. Nous présentons, dans le chapitre 2, les problématiques liées à ce domaine. A partir de problèmes concrets que posent quotidiennement et dans divers domaines les matériaux granulaires, nous présentons une des problématiques de recherche actuelles de la communauté des chercheurs des milieux granulaires.

Dans le chapitre 3, nous présentons les principaux algorithmes de simulation numérique classiquement employés en physique des milieux granulaire pour simuler un empilement à l'équilibre statique. Les domaines d'inspiration de chacun de ces algorithmes ainsi que leurs performances y sont présentés.

Le chapitre 4 détaille GranuSolve, notre algorithme de résolution de tas de sable à l'équilibre statique. Cet algorithme repose sur une représentation distribuée, à l'aide d'agents-grains, de la résolution d'un tas. Dans ce chapitre, nous considérons un grain comme un agent essentiellement réactif dont le but est de trouver son équilibre en fonction de ses voisins. Notre algorithme repose sur deux méthodes de conception et de résolution de systèmes multi-agents. La première est la méthodologie de conception de système multi-agent Cassiopée qui nous a aidé à modéliser le comportement d'un agent-grain [Collinot et Drogoul 1996]. La seconde est la méthode d'Eco-Résolution que nous avons mis en œuvre dans le cas de la résolution d'un empilement constitué d'agents-grains [Ferber 1990, Drogoul *et al.* 1991, Drogoul et Dubreuil 1991]. Les différentes versions de notre solveur sont alors données pour illustrer et justifier l'ajout de connaissances du domaine. Ces connaissances sont représentées sous la forme d'heuristiques pour l'Eco-Résolution du tas et de comportements de résolution pour les agents-grains. La validation, les performances et les limitations du système sont discutées pour chacune des versions implémentées.

Dans le chapitre 5, nous définissons les fonctionnalités que nous attendons d'un environnement d'aide à la découverte appliqué à la physique des milieux granulaires. La construction d'un tel environnement pose deux types de problèmes. D'une part, des problèmes d'ordres méthodologiques, pour la représentation du processus de découverte, sont détaillés sous la forme d'un cahier des charges fonctionnel. D'autre part, nous traitons des problèmes technologiques pour la réalisation concrète de cet environnement.

La réalisation concrète du système GranuLab est détaillée dans le chapitre 6. L'architecture de notre système est basée sur l'utilisation de composants traitant des données au format XML. Le choix des technologies utilisées sont justifiés. Nous présentons le modèle de composant que nous avons conçu pour représenter les opérateurs de transformation que nous avons défini dans le chapitre 5. Le modèle d'un schéma d'expérimentations, permettant de créer et d'exécuter une expérimentation virtuelle est également détaillé. La généricité de l'approche est enfin discutée.

Le chapitre 7 présente une étude détaillée d'une expérimentation virtuelle avec GranuLab. Cette expérimentation tente d'apporter des éléments de réponse concernant la question ouverte du comportement macroscopique d'un empilement à l'équilibre statique.

Les conditions limites et les opérateurs de transformation utilisés sont détaillés. Les résultats de notre expérimentation sont ensuite comparés à ceux d'une théorie mathématique portant sur le caractère diffusif et propagatif du milieu granulaire.

Enfin, dans la conclusion, nous présentons les contributions, les limitations et les perspectives de recherche qui découlent de notre travail de thèse.

Guide de lecture

Il est préférable de lire cette thèse dans son intégralité et en respectant l'ordre des chapitres. Cependant, plusieurs domaines distincts sont abordés au cours de ce manuscrit. Ainsi, le lecteur intéressé par les aspects Intelligence Artificielle pourra trouver, dans le chapitre 1, une présentation originale des systèmes d'aide à la découverte scientifique appliqués à un domaine de la physique. Il pourra ensuite parcourir le chapitre 5 pour avoir une définition fonctionnelle de notre système. Le chapitre 6 présente une implémentation d'un tel environnement et le chapitre 7 illustre son utilisation à travers une expérimentation dans le domaine de la physique des milieux granulaires..

Le lecteur essentiellement intéressé par les aspects multi-agents pourra se limiter au chapitre 4 qui détaille notre algorithme de résolution de tas de sable à l'équilibre statique.

Enfin, le lecteur que les aspects physiques des milieux granulaires intéresse, pourra parcourir le chapitre 2 qui donne une vision synthétique des problèmes concrets que posent les matériaux granulaires. Le chapitre 3 lui donnera ensuite un aperçu des algorithmes classiques de simulation numérique employés par les physiciens de ce domaine. Enfin, le chapitre 7 détaille une expérimentation complète menée à l'aide de notre système.

Chapitre 1

L'aide à la découverte scientifique appliquée à la physique

Sommaire

1.1 L'aide à la découverte scientifique : un champ de l'Intelligence Artificielle

La *découverte scientifique*, comme champ d'investigation en Intelligence Artificielle (IA), est souvent décrite comme étant une tâche d'apprentissage non supervisé [Langley et Żytkow 1989]. Langley et Żytkow distinguent trois aspects différents de la découverte scientifique. Le premier concerne l'organisation d'objets en catégories et taxonomies, un champ de l'apprentissage appelé *regroupement conceptuel* [Michalski et Stepp 1983, Bournaud *et al.* 2000]. Un second aspect consiste à induire des régularités à partir de données décrivant le phénomène étudié. Enfin, un troisième aspect est celui de systèmes qui tentent de formuler des théories expliquant les données, par exemple en utilisant le principe de l'analogie [Falkenhainer 1987, Cornuéjols 1996a, Cornuéjols 1996b, Craig et Nersessian 2001]. Dans cette thèse, nous nous intéressons plus particulièrement au deuxième aspect.

Deux grands types de systèmes de découverte scientifique sont habituellement distingués en Intelligence Artificielle : ceux qui sont *guidés par les données* («data-driven») et ceux qui sont *guidés par la théorie* ou par un problème («theory-driven»). Des systèmes de découverte scientifique couvrant ces deux types d'approches ont vu le jour dès la naissance de l'Intelligence Artificielle comme un domaine scientifique à part entière.

Au milieu des années 60, Edward A. Feigenbaum, en collaboration avec le généticien et Prix Nobel Joshua Lederberg, travaille sur un système permettant d'identifier la structure géométrique d'une molécule : DENDRAL [Feigenbaum *et al.* 1971, Lindsay *et al.* 1993]. Le but de ce système était d'effectuer automatiquement une des tâches d'un chimiste, à savoir la découverte de la géométrie d'une molécule à partir des éléments de base qui la constituent et de son analyse au spectromètre de masse. Le processus se décompose en trois étapes :

1. *Le plan* : L'expert donne la description des fragments de molécules qui doivent se trouver dans la molécule finale et ceux qui ne doivent pas y appartenir.
2. *La génération* : Le système génère toutes les combinaisons possibles de molécules à partir des fragments de base en respectant la contrainte des fragments interdits.
3. *L'analyse* : Cette dernière étape permet de simuler l'analyse au spectromètre de masse d'une molécule candidate et de la comparer avec l'analyse effectuée sur la molécule de base. Le système peut ainsi éliminer un grand nombre de solutions inadéquates.

Le système DENDRAL fut le premier à intégrer explicitement des heuristiques d'un domaine d'expertise particulier pour tenter de contraindre la recherche dans un espace de possibilités et permettre ainsi une réduction notable de cet espace. Ces heuristiques sont données par les experts en chimie moléculaire et c'est, grâce à ces connaissances, que le système réalise un gain, d'un facteur deux cents environ, du nombre de possibilités pour la géométrie de la molécule.

A partir de ces expérimentations, Feigenbaum a souligné l'importance des connaissances *a priori* dans un système d'aide à la découverte («Knowledge Principle»). A l'époque, les chercheurs pensaient que plus un système possédait de connaissances du domaine d'expertise, meilleurs seraient ses résultats [Lenat *et al.* 1983, Lenat et Guha 1990]. Aujourd'hui, nous savons que la représentation des connaissances [Euzenat 1999, Euzenat 1993] joue un rôle aussi important et que la quantité de connaissances ne garantit pas la découverte de connaissances nouvelles et intéressantes. En effet, lorsque trop de connaissances sont données à un système d'aide à la découverte, il devient alors difficile de cerner celles qui pourraient lui être utiles à un instant donné.

Ses travaux furent à la base des systèmes experts développés dans les années 1970 (MYCIN [Shortliffe 1976, Buchanan et Shortliffe 1984b]). En effet, l'idée de Shortliffe et Buchanan [Buchanan et Shortliffe 1984a] fut d'adjoindre aux hypothèses générées, des cœfficients de probabilités mesurant le degré de confiance dans les hypothèses inférées par le système. Ainsi, dans le cas du diagnostic médical où de multiples règles peuvent s'appliquer sur un même cas clinique, plusieurs diagnostics sont souvent possibles. Ces diagnostics sont issus de l'application itérée des règles d'inférence et de déduction à partir des symptômes et des connaissances médicales concernant la pathologie observée.

Néanmoins, l'acquisition de ces connaissances du domaine requiert un travail d'introspection de la part des experts qui est long et contraignant, surtout lorsque des connaissances précises leurs sont demandées.

A la même époque, D. Lenat [Lenat 1977a, Lenat 1984a] écrit un programme en LISP qui crée automatiquement des concepts mathématiques à partir d'axiomes tirés de la théorie des ensembles : le système AM («Automated Mathematician»). Lenat a introduit des connaissances issues du domaine des mathématiques dans son système sous forme d'axiomes et d'un nombre impressionnant d'heuristiques (plus de deux cents). Ce système était novateur dans le sens où il pouvait fonctionner tout seul, sans aucune intervention avec un utilisateur et sans posséder de données en entrée. Il conduisait néanmoins à la génération de nouveaux concepts en mathématiques. Le système AM a ainsi redécouvert plus de trois cents concepts mathématiques dont la conjecture de Goldbach : «Tout nombre entier supérieur à 3 est la somme de deux nombres premiers».

Par la suite, les difficultés du système à trouver de nouveaux concepts ont montré que les heuristiques seules ne suffisaient pas à découvrir des connaissances vraiment nouvelles dans un système fonctionnant en boucle fermée. Lenat en a conclu que les connaissances apprises se trouvaient à la frontière de ce qui était déjà connu (les connaissances données sous forme d'axiomes au système) et étaient d'une nature très proche des connaissances de base [Lenat et Feigenbaum 1991]. Il était donc nécessaire d'avoir une palette de connaissances la plus large possible afin de pouvoir apprendre des concepts réellement nouveaux [Lenat *et al.* 1983, Lenat et Guha 1990]. Lenat a ensuite souligné l'importance du formalisme de représentation des connaissances dans ses systèmes pour expliquer les réussites de AM et d'EURISKO [Lenat et Brown 1984].

Ces deux exemples de systèmes d'aide à la découverte scientifique dans les domaines de la chimie et des mathématiques ont montré l'importance cruciale des connaissances du domaine d'expertise pour la qualité des hypothèses générées par le système. Ainsi, il ressort que les connaissances d'un expert peuvent apparaître, dans un système d'IA, notamment sous la forme de :

- **données en entrée du système :** ce sont une représentation possible des connaissances modélisant le problème à étudier. Ces connaissances sont souvent difficiles à représenter car il faut trouver un juste équilibre entre précision et concision de la description ;
- **heuristiques :** il s'agit des contraintes inhérentes à la théorie étudiée. Ces connaissances sont les plus difficiles à obtenir des experts, car s'il leur est facile de décrire ce sur quoi ils travaillent, il leur est beaucoup moins aisé d'expliciter comment ils travaillent, d'extérioriser les outils et les méthodes qu'ils emploient tous les jours presque «machinalement».

Ainsi, si la qualité des connaissances apprises dépend des connaissances introduites dans un système d'aide à la découverte, leur acquisition est une tâche difficile [Langley et Michalski 1986]. Plutôt que de représenter toutes les connaissances d'un domaine particulier dans un système d'aide à la découverte scientifique, nous avons préféré adopter une approche différente : la *collaboration homme-machine*, initiée notamment par R. E. Valdés-Pérez [Valdés-Pérez 1999, Valdés-Pérez 1995d]. Considérant les difficultés à acquérir les connaissances des experts pour les mettre dans un système d'IA, nous avons préféré modéliser le processus de recherche lui-même et tenter d'automatiser le plus grand nombre d'étapes possibles de ce processus de découverte.

Notre but est d'arriver à accélérer le processus de découverte tout en laissant à l'expert son rôle de chercheur dans le sens où il va lui-même diriger l'exploration de l'espace de recherche en utilisant ses propres heuristiques. Quant à la machine, elle conserve son rôle de calculateur en automatisant les étapes intermédiaires.

A présent, nous allons nous pencher sur l'étude des systèmes d'aide à la découverte scientifique lorsqu'ils sont appliqués à l'un des nombreux domaines de la physique (particules sub-atomiques, mécanique classique, électronique de puissance, etc.). Nous allons tenter de montrer l'étendue du spectre des systèmes de découverte scientifique en IA à travers le prisme du *niveau d'interaction explicite* proposé par ces différents systèmes.

1.2 L'aide à la découverte scientifique en physique

Depuis les débuts de l'IA, la plupart des systèmes d'aide à la découverte sont dédiés à un domaine de la physique ou de la chimie (cf. Fig. 1.1).

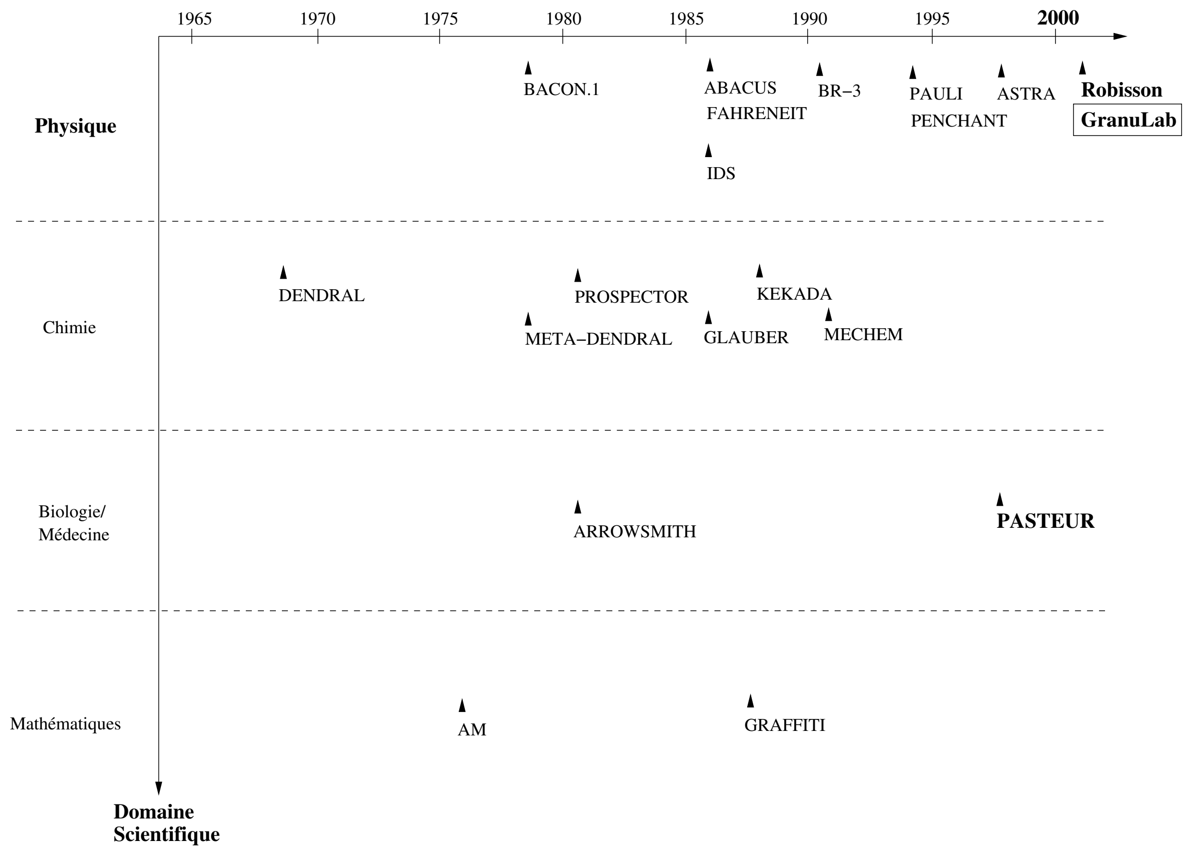


FIG. 1.1 – Historique des systèmes d'aide à la découverte scientifique par domaines d'application

Nous abordons les systèmes de découverte ou d'aide à la découverte scientifique sous l'angle original du niveau d'interaction explicite existant avec l'utilisateur. En effet, l'approche traditionnellement employée «guidée par la théorie» ou «guidée par les données» ne nous permettrait pas de situer clairement notre travail. Nous avons une première phase qui nous permet de générer des données à partir d'une théorie et une deuxième phase où ces données vont servir à inférer de nouvelles connaissances.

Afin de situer notre problématique dans un cadre plus général, nous présentons les systèmes autonomes basés sur une théorie et ceux basés sur des résultats d'expérimentations réelles. Ces systèmes n'offrent que peu d'interactions avec l'utilisateur. Les systèmes interactifs tels que ceux réalisant de la fouille de données sont ensuite présentés. Enfin, les systèmes interactifs basés sur des données d'expérimentations virtuelles sont décrits. Notre système GranuLab s'inscrit dans cette dernière famille de systèmes d'aide à la découverte scientifique.

1.2.1 Les systèmes autonomes basés sur une théorie

Dans les années 1960, les débuts de l'IA faisaient naître de grands espoirs dans la communauté scientifique. Certains pensaient que les ordinateurs pourraient très certainement faire seuls de grandes découvertes scientifiques et ainsi accélérer le développement de la recherche.

Le système Am [Lenat 1977a, Lenat 1984a], mentionné précédemment, a été conçu avec l'idée qu'un programme peut faire des découvertes dans le domaine des mathématiques de façon complètement autonome.

A partir d'une *théorie* initiale contenant les axiomes de la théorie des ensembles, le système essaie de déduire de nouveaux théorèmes à l'aide d'heuristiques de déduction et d'inférence mathématique.

D'un point de vue des expérimentations, il est évident que l'utilisateur-concepteur du système a eu de nombreuses interactions avec son système avant d'obtenir des résultats probants. Il est très improbable qu'il ait pu donner à son système les quelques centaines d'heuristiques et d'axiomes dès la première utilisation. Les interactions ont dû être nombreuses, même si les résultats qui ont été publiés n'en font pas clairement état.

Sur le schéma (cf. Fig. 1.2), le *moteur* est principalement constitué de règles d'inférences. Il propose des lois qui seront ensuite *validées* à l'aide d'un ensemble de règles de déduction mathématique. L'*interprétation*, par l'expert du domaine, des règles découvertes peut alors l'amener à corriger certaines erreurs dans les modules d'inférence et de déduction, voire d'ajouter des règles qui avaient été oubliées. L'interaction peut être aussi beaucoup plus riche qu'une simple mise au point de programme. En effet, lorsque l'expert omet volontairement certains axiomes, il peut alors analyser la théorie que le système propose dans le cadre d'hypothèses réduites.

Les découvertes faites par ce de système furent nombreuses [Lenat 1977a, Lenat 1982]

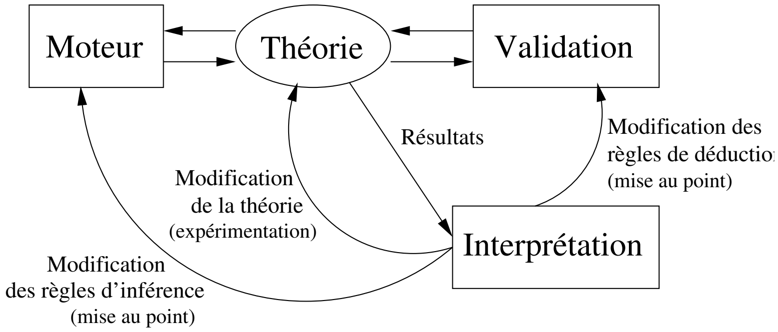


FIG. 1.2 – Architecture des systèmes d'aide à la découverte scientifique autonomes basés sur une théorie

mais la plupart d'entre elles étaient déjà connues et donc sans intérêt pour les mathématiciens sinon celui de montrer qu'une machine pouvait réaliser des démonstrations. Parmi les nouvelles connaissances découvertes par le système, la grande majorité étaient inintéressantes car elles étaient difficiles à appréhender (propriétés «exotiques» ou trop complexes).

AM illustre la difficulté de réaliser automatiquement la découverte de connaissances qui soient tout à la fois *nouvelles*, *intéressantes*, *plausibles* et *intelligibles* [Valdés-Pérez 1999]. Margaret Boden [Boden 1990] distingue une connaissance découverte qui est nouvelle pour la personne (le système) et parle de p-créativité (p pour «personal») d'une découverte qui n'avait jamais été faite auparavant et parle alors de h-créativité (h pour «historic»).

Ainsi, il est très difficile de découvrir des connaissances h-créatives à partir d'une théorie abstraite comme les mathématiques. La découverte de connaissances moins universelles que des théorèmes peut être alors envisagée, en particulier en se basant sur des données issues d'expérimentations réelles.

1.2.2 Les systèmes autonomes basés sur des résultats d'expérimentations réelles

Le plus représentatif de ce type de système est sans conteste le système BACON [Langley 1978] et ses différentes versions (BACON.1 : [Langley 1978] et BACON.4 : [Bradshaw *et al.* 1980]). Ils ont permis de redécouvrir des lois dans les domaines de la physique [Langley 1981] et de la chimie [Langley *et al.* 1983].

BACON prend en entrée une liste de descripteurs et tente d'expliquer les descripteurs «dépendants» à l'aide de ceux qui sont notés «indépendants». Par exemple, un ensemble de mesures est donné au système. Chacune comporte trois descripteurs notés indépendants (T : pour la température du gaz, P : la pression et N : la quantité de gaz) et une seule variable notée dépendante V : le volume du gaz (cf. Fig. 1.3).

P	V	N	T
12.12	20.523	52.241	21.7
15.13	13.578	45.678	20.5
16.81	8.592	33.245	19.8
19.23	35.110	131.499	23.4
9.96	7.995	16.496	22.0
13.65	16.349	50.102	20.3
16.78	18.262	75.085	18.6
⋮	⋮	⋮	⋮

FIG. 1.3 – Les données en entrée du système BACON

BACON donne alors comme résultat trois types de règles (cf. Tab. 1.1). Le premier type de règles ne comporte que des lois numériques simples (valeurs de constantes et relations linéaires), le second des lois symboliques entre paires de variables et enfin un type contenant les propriétés intrinsèques des objets étudiés (relations liant les descripteurs numériques d'un objet à ses descripteurs symboliques).

Lois numériques simples	Lois symboliques simples	Propriétés intrinsèques
X = 8.32	X = Y/T	masse(A) = 1.43
U = 1.57 V	Y = PV	température(A) = 18.9

TAB. 1.1 – Les trois types de règles fournies en sortie par le système BACON

D'un point de vue général, ce type de système prend en entrée des *données* qui ont été *saisies le plus souvent par un expert*. Dans un deuxième temps, le moteur d'inférence tente de détecter des régularités dans ces données. La validation de la théorie issue de cette deuxième phase peut amener le système à «demander» plus de données afin de valider la théorie sur un spectre plus large ou plus précis. Enfin, la validation par l'expert de la théorie peut l'amener à modifier les données (l'ordre par exemple) et à guider ainsi le moteur d'inférence par des heuristiques implicites. Ces heuristiques ne sont pas spécifiées dans ce type de système alors qu'elles sont souvent employées par l'utilisateur/expert. En effet, les résultats obtenus sont souvent le fruit d'expérimentations de plusieurs heuristiques portant sur les données (organisation, filtrage, etc.) menées par l'utilisateur.

Le système BACON a permis de poser les bases d'un moteur d'inférence pour la découverte de lois symboliques à partir de données. Il repose sur trois principes :

1. *La création de descripteurs d'un niveau supérieur.* La création d'un descripteur à partir de deux descripteurs existants permet au système de ne manipuler que des

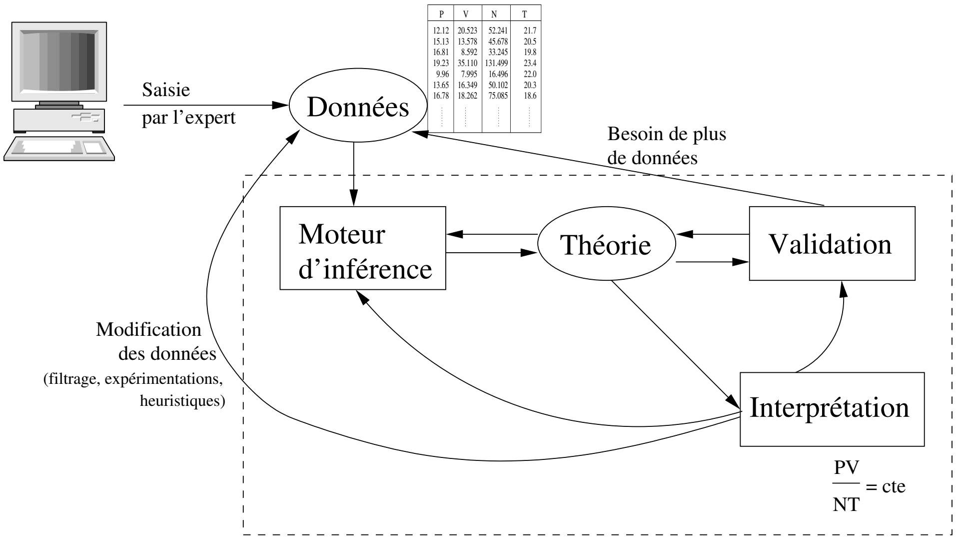


FIG. 1.4 – Architecture des systèmes d'aide à la découverte scientifique autonomes basés sur des résultats d'expérimentations réelles
Cette architecture contient celle des systèmes d'aide à la découverte basés sur une théorie (en pointillé sur le schéma) vue à la section précédente.

relations simples du type $X = AB$. Ces relations ne font intervenir qu'au plus trois descripteurs (matrice triangulaire supérieure), ce qui permet de simplifier grandement la tâche de découverte d'une loi entre descripteurs. Ainsi, la loi des gaz parfaits $P \times V = cte \times N \times T$ peut s'écrire uniquement à l'aide des formules suivantes :

$$\begin{aligned} X &= 8.32 \\ X &= Y/T \\ Y &= Z/N \\ Z &= PV \end{aligned}$$

2. *La recherche heuristique.* La recherche heuristique permet de ne pas considérer inutilement toutes les combinaisons d'opérations deux à deux ou trois à trois de descripteurs. Une heuristique souvent utilisée est la *mesure des corrélations* existantes entre descripteurs. Deux descripteurs A et B peu ou pas corrélés ont très peu de chance de former directement une relation du type $AB = cte$. Par contre, deux descripteurs corrélés positivement vont amener le système à ne considérer que le quotient et la différence de ces descripteurs, i.e. la création et l'étude des nouveaux descripteurs $X_1 = A/B$ et $X_2 = A - B$.

3. *La récursion.* C'est le mécanisme essentiel de ce genre de méthode. La récursion permet d'introduire dans le système les descripteurs qui ont été créés à l'étape 1. Les formules découvertes faisant intervenir les descripteurs créés par le système sont donc de plus en plus complexes. Le système peut ainsi aborder aisément le problème de la découverte de lois symboliques complexes.

Grâce à ces trois principes simples, BACON a permis de redécouvrir la loi de Kepler sur les périodes orbitales des planètes [Langley 1981]. Cependant, les lois que peut découvrir BACON se limitent à des égalités entre variables symboliques. Même si ces formules peuvent être complexes, elles ne peuvent pas rendre compte d'inégalités ou de transitions de phases en physique, par exemple. De même, BACON prend en considération toutes les variables et leurs valeurs d'un seul bloc, il lui est donc impossible de trouver des lois qui ne seraient vérifiées que pour un sous-groupe de valeurs expérimentales, ni de donner une description du domaine sur lequel une loi est vérifiée [Langley et Nordhausen 1986].

Par la suite, les systèmes de ce type ont évolué pour prendre en compte des connaissances de plus en plus structurées et ainsi autoriser la découverte de connaissances d'un ordre supérieur à ce que pouvaient obtenir les premiers systèmes. Cependant, il est important de noter que les différents systèmes créés par la suite se sont appuyés sur l'expérience acquise des systèmes antérieurs et ont tenté d'apporter une réponse aux carences et limitations de ces systèmes.

C'est ainsi que le système FAHRENHEIT [Koehn *et al.* 1986, Żytkow 1987] donne, pour chaque loi découverte, la description et les bornes du domaine dans lequel la loi est valide. En fait, FAHRENHEIT exprime les bornes *min* et *max* de ce domaine sous forme de lois et ceci en employant la technique de l'ajout de nouveaux termes (cf. étape 1) : D_{min} et D_{max}. Ensuite, le système cherche à «expliquer» ces nouveaux descripteurs en appliquant l'algorithme de recherche de lois (cf. étape 2). Le système applique alors récursivement cette méthode (cf. étape 3) pour trouver des bornes pour les termes qui entreraient dans la description des lois des bornes D_{min} et D_{max}.

Les bornes des lois trouvées pouvant être elles-mêmes des fonctions complexes, la limitation de la validité des lois découvertes à des hyper-rectangles n'existe plus. Des espaces beaucoup plus complexes peuvent être envisagés. Cependant, ce système reste confiné à la découverte de lois concernant des descripteurs numériques. Et même, si l'espace des lois s'est considérablement enrichi, il ne permet toujours pas de représenter des transitions de phases en physique mais uniquement des transitions binaires.

Comme pour BACON, ce système tient compte de l'ordre dans lequel les descripteurs ont été donnés en entrée. Pour tenter de s'affranchir de ce biais, le système propose une permutation cyclique des descripteurs, l'ensemble des arrangements possibles étant beaucoup trop coûteux à prendre en considération. FAHRENHEIT apporte de plus quelques améliorations au niveau des heuristiques pour la découverte de lois, comme la détection de variables non corrélées, qui permettent de réduire sensiblement l'espace de recherche de lois.

D'autres systèmes, tels que IDS [Langley et Nordhausen 1986] ou GLAUBER [Falkenhainer et Michalski 1986], tentent de répondre à la nécessité de prendre en considération des comportements qualitatifs qui représentent des caractéristiques importantes pour des domaines tels que la physique. Par exemple, IDS requiert une structuration des données en entrée du système sous forme d'objets dont la valeur des attributs évolue au cours du temps. Le système est ainsi capable d'induire des schémas qualitatifs à partir de données temporelles d'expérimentations. Ensuite, un algorithme similaire à ceux de BACON ou de FAHRENHEIT permet d'induire des lois pour chaque phase qualitative distincte.

1.2.3 Les systèmes qui proposent la réalisation d'expérimentations réelles à l'expert

Le système KEKADA [Kulkarni et Simon 1988, Kulkarni et Simon 1990] propose à un expérimentateur humain de réaliser des expérimentations réelles en chimie pour vérifier certaines hypothèses. Le système choisit, parmi les expériences possibles, celle qui sera la moins coûteuse à réaliser en termes de coût financier des produits impliqués mais aussi en termes de faisabilité (conditions particulières pour mener l'expérience à bien, temps de réalisation, ...) pour vérifier une hypothèse.

Les données en entrée du système sont de deux natures :

1. *Les connaissances du domaine* telles que le prix et les conditions d'utilisation des produits chimiques, etc. ;
2. Les résultats des mesures d'*expérimentations réelles* réalisées par l'expérimentateur humain.

Aux vues des résultats des expérimentations, il va émettre des hypothèses et proposer de nouvelles expérimentations pour les vérifier. Ainsi, ce système fonctionne de façon quasi-autonome avec une seule phase d'interaction avec le monde réel.

Cette étude systématique autorise, *a priori*, une exploration non subjective et automatique de l'espace de recherche de lois. En pratique, il peut être très long et coûteux de

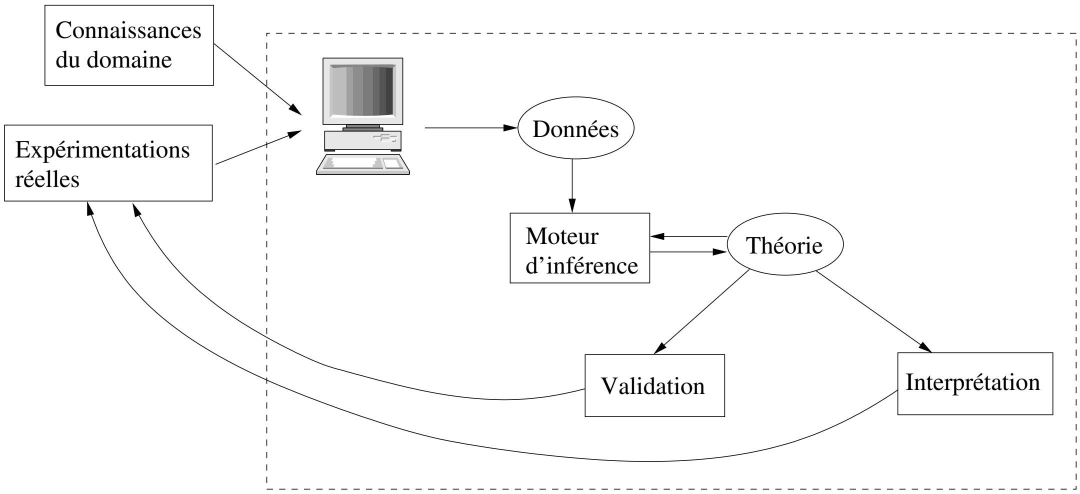


FIG. 1.5 – Architecture des systèmes d'aide à la découverte scientifique autonomes en interaction avec un expérimentateur humain

Cette architecture contient celle des systèmes autonomes basés sur des résultats d'expérimentations réelles (en pointillé sur la figure) présentés dans la section précédente.

laisser le système explorer des parties inintéressantes de l'espace parce qu'il va redécouvrir des lois ou réactions déjà connues.

Il manque donc à KEKADA la possibilité pour l'expert d'interagir avec le système.

1.2.4 Les systèmes interactifs basés sur des résultats d'expérimentations réelles

Ces systèmes ont vu le jour avec l'avènement des systèmes d'information à grande échelle, dans les années 1980. La quantité d'information à gérer étant de plus en plus importante, un procédé pour tenter d'extraire de façon semi-automatique des régularités dans les bases de données a été proposé. Il n'était plus possible de traiter à la main ces gigantesques volumes de données. Des outils de *fouille de données* («data-mining») ont donc vu le jour en fournissant aux utilisateurs/experts des outils élémentaires pour manipuler, filtrer, trier, etc. ces données afin d'en extraire des connaissances nouvelles.

Les données proviennent, dans la majorité des cas, de bases de données alimentées plus ou moins directement par des humains. La saisie d'une commande, par exemple, pourra être effectuée par un employé ou même directement par le client qui commande en ligne. Même si, de plus en plus d'enregistrements sont mémorisés automatiquement, c'est un humain qui reste le plus souvent la cause de ces données.

Dans un tel système, les interactions avec l'expert sont nombreuses et l'une des caractéristiques importantes est la possibilité de créer ses propres expérimentations. En effet, il est impossible d'explorer de façon systématique l'espace de recherche de régularités dans de telles bases. Les concepteurs de ces systèmes fournissent donc à l'expert des outils pour qu'il puisse effectuer ses propres recherches. C'est en cela que ce sont de véritables outils d'*aide à la découverte*. (cf. Fig. 1.6).

Clémentine [Khabaza et Shearer 1995] est un système de fouille de données de troisième génération. Il permet de représenter et d'opérationnaliser le processus d'aide à la découverte. A partir d'outils d'analyse, d'opérateurs de transformation et de visualisation, l'utilisateur peut créer ses propres expérimentations virtuelles et y inclure explicitement ses propres heuristiques de recherche.

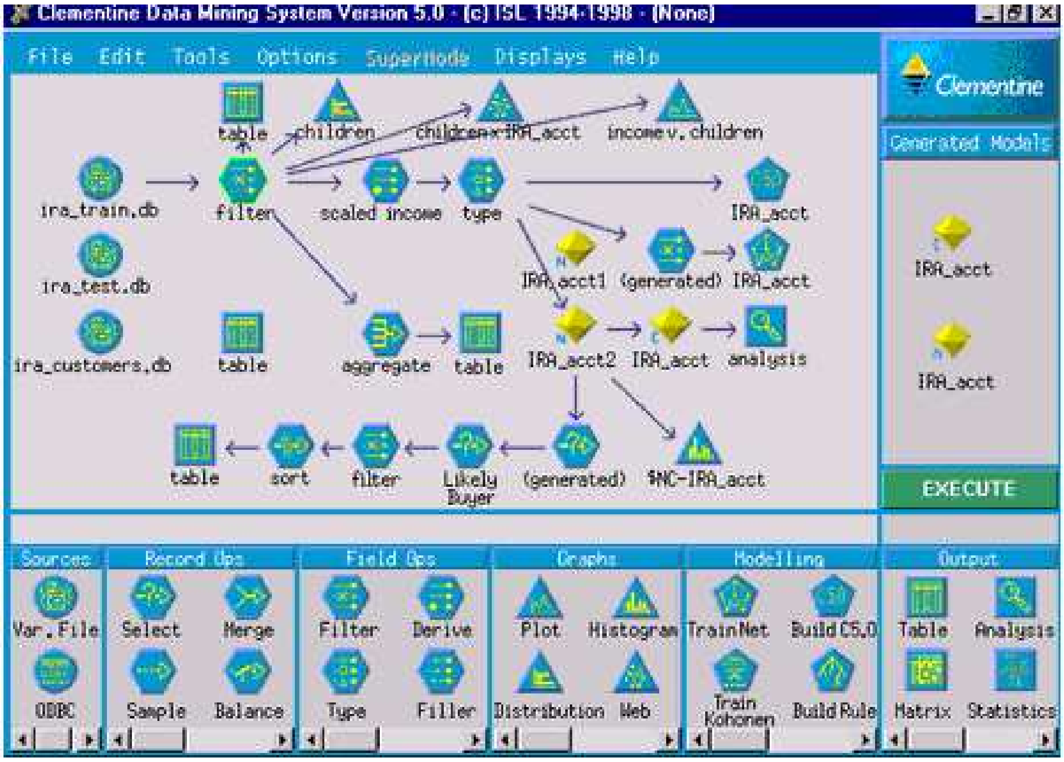


Fig. 1.6 – Clémentine : un système de fouille de données de troisième génération
Le processus d'aide à la découverte est représenté dans la fenêtre principale. En bas, les outils et opérateurs disponibles et, à droite, le contrôle de l'exécution du processus.

Clémentine n'est pas le seul système de ce type (K-wiz, Enterprise Miner [Potts 1997], etc.) et il a été dégagé un schéma générique de la fouille de données (cf. Fig. 1.7). Ce schéma se décompose en plusieurs phases :

1. *L'acquisition des données.* Les données sont souvent issues de bases de données complexes à partir desquelles un traitement est effectué. Il faut donc que le système puisse interroger directement une base de données pour en extraire les données utiles

et changer le format de représentation de ces données pour qu'il corresponde à celui de l'application.

2. *La sélection.* La phase de sélection permet de sélectionner les données à importer dans le système à partir des bases de données extérieures.

3. *Les pré-traitements sur les attributs.* La reformulation et le changement de type sont les deux principales opérations de cette phase du processus de découverte. Ces opérations ne concernent que les attributs des tables de la base de données. Elles sont souvent nécessaires pour pouvoir appliquer, par la suite, les opérateurs fournis dans l'environnement de fouille de données.

4. *Les pré-traitements sur les données.* Cette phase permet principalement un filtrage des données pour ne considérer que celles qui seront utiles pour le traitement que l'expert souhaite effectuer.

5. *L'analyse.* Le traitement, à proprement parler, regroupe un ensemble d'outils d'analyse, de classification et d'apprentissage. C'est le cas des réseaux de neurones qui apprennent une fonction, des classifieurs qui permettent de trouver des regroupements pertinents, ou plus simplement des fonctions statistiques (fit, lissage, etc.).

6. *L'interprétation.* L'interprétation est une phase essentielle qui demande toute l'attention de l'expert. Les systèmes de fouille de données proposent en général de visualiser les résultats sous forme de courbes, nuages de points, i.e. tous les types de visualisation susceptibles d'aider l'expert à comprendre les résultats.

A l'issue d'un traitement, l'expert peut, au regard des résultats, modifier les paramètres du pré-traitement ou de l'analyse. C'est grâce à ces multiples *interactions* que l'expert peut avancer dans ses recherches. A noter qu'il est tout à fait possible d'imaginer un outil d'analyse du type BACON qui permette d'inférer des lois symboliques à partir des données.

Ce type de système d'aide à la découverte est déjà très complet. Il est néanmoins possible d'adjoindre une étape supplémentaire dans le processus de découverte en amont des données. Cette étape consiste simplement en la génération des données à traiter. Ces données sont alors issues de simulations numériques basées sur un modèle ou une théorie du domaine d'expertise.

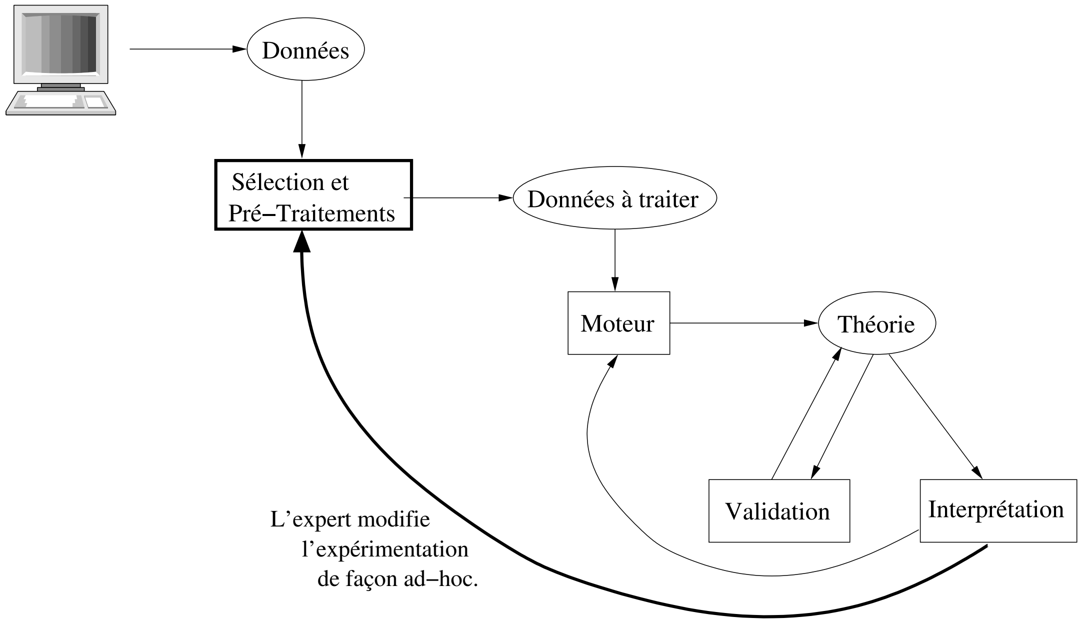


FIG. 1.7 – Architecture des systèmes interactifs de fouille de données basés sur des résultats d'expérimentations réelles
Cette architecture contient celle des systèmes autonomes d'aide à la découverte en interaction avec un expérimentateur humain (en police normale sur le schéma) présentés à la section précédente.

1.2.5 Les systèmes interactifs basés sur des résultats d'expérimentations virtuelles

L'avènement des systèmes de fouille de données de troisième génération, comme nous les avons décrits précédemment, facilite aujourd'hui l'ajout d'une étape supplémentaire dans le processus d'aide à la découverte scientifique. En effet, l'application de ces environnements d'aide à la découverte dans le cas de données issues d'expérimentations virtuelles et non plus réelles est un pas qu'il est facile de franchir.

Dans ce cadre, les données à analyser ne préexistent pas mais sont issues d'*expérimentations virtuelles* menées par l'expert ou la machine. Ces expérimentations se basent sur un modèle ou une *théorie* issue du domaine d'expertise (cf. Fig. 1.8).

Il est important de noter que le cadre théorique servant à générer les données n'est pas nécessairement identique à celui de la théorie découverte par le système. Par exemple, dans le cas de l'étude des systèmes complexes en physique, la théorie initiale, permettant de générer les données, décrit des phénomènes au niveau microscopique. La théorie «résultante» tente alors d'expliquer les phénomènes qu'il est possible de rencontrer au niveau macroscopique à partir des solutions issues de la première théorie. Il s'agit alors de modéliser et de caractériser le processus de transition du niveau micro au niveau macro.

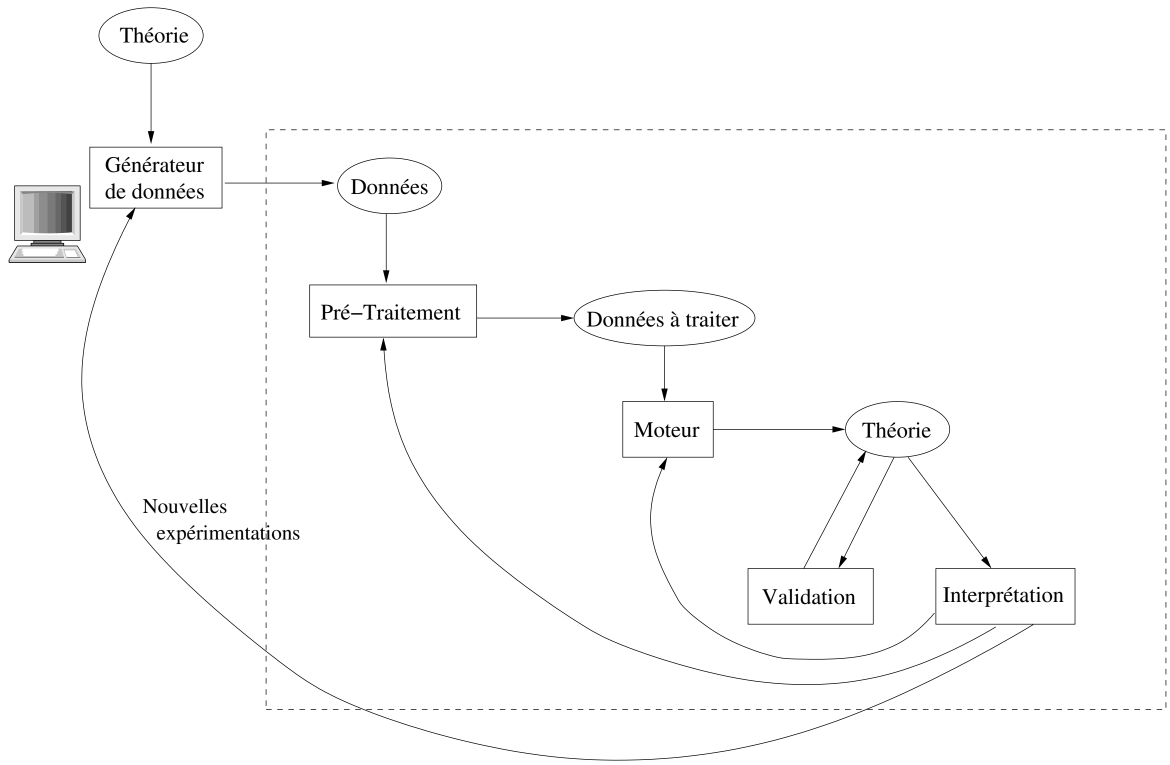


FIG. 1.8 – Architecture des systèmes d'aide à la découverte scientifique basés sur des résultats d'expérimentations virtuelles
Cette architecture contient celle des systèmes interactifs basés sur des données de résultats d'expérimentations réelles (en pointillé sur le schéma) vue à la section précédente.

C'est dans ce cadre que se situe le système GRANULAB : un système d'aide à la découverte scientifique appliqué à la physique des milieux granulaires. Nous avons essayé d'inférer, par régression, des lois de comportements macroscopiques fiables de tas de sable à partir de simulations numériques. Notre algorithme de simulation numérique GRANU-SOLVE se base sur un modèle caractérisant l'équilibre statique d'un grain en fonction de leurs rugosités (cf. Chap. 4).

1.3 Conclusion

Dans ce premier chapitre, nous avons d'abord introduit la notion d'aide à la découverte scientifique comme un champ de l'Intelligence Artificielle. Des systèmes d'aide à la découverte scientifique ont été présentés suivant l'axe original des interactions possibles avec l'expert. Cet axe nous a permis de décrire à la fois les différents types de systèmes de façon incrémentale mais aussi d'inclure les environnements de fouille de données comme

Famille de systèmes d'aide à la découverte scientifique	**Systèmes**	**Rôle de la machine**	**Type de données en entrée**
Autonomes basés sur une théorie	AM [Lenat 1977a]	Moteur d'inférence et de déduction (démonstration de théorèmes)	Axiomes de la théorie étudiée
Autonomes basés sur des résultats d'expérimentations réelles	BACON [Langley 1978], ABACUS [Falkenhainer et Michalski 1986], IDS [Langley et Nordhausen 1986], GRAFFITI [Fajtlowicz 1988], ...	Découverte de lois symboliques	Numériques et/ou qualitatives
Autonomes en interaction avec un expérimentateur humain	KEKADA [Kulkarni et Simon 1988]	Optimiser les expérimentations à réaliser pour vérifier les hypothèses inférées	Résultats d'expérimentations réelles
Interactifs basés sur des résultats d'expérimentations réelles	CLÉMENTINE [Khabaza et Shearer 1995], ENTERPRISE MINER [Potts 1997], K-WIZ	Modéliser le processus de découverte, rôle calculatoire	Bases de données
Interactifs basés sur des résultats d'expérimentations virtuelles	Robisson [Robisson 2001], **GRANULAB** [Breton *et al.* 2000]	Générer les données, modéliser et mémoriser le processus de découverte	Modèle ou théorie du domaine

TAB. 1.2 – Classification des systèmes d'aide à la découverte selon le niveau d'interaction avec l'utilisateur/expert

des systèmes d'aide à la découverte (cf. Tab. 1.2).

Notre démarche incrémentale nous a permis d'aboutir aux systèmes d'aide à la découverte à partir de données virtuelles qui caractérisent GRANULAB. Ce dernier, étant appliqué à la physique des milieux granulaires, nous détaillons, dans le chapitre suivant, la problématique liée à ce domaine.

Chapitre 2

Problèmes de physique granulaire et motivations pour la simulation numérique

Sommaire

2.1 Introduction

Notre travail est soutenu par un projet «Jeune Equipe» du CNRS. Notre équipe est constituée pour moitié de physiciens du Laboratoire des Milieux Désordonnés et Hétérogènes (LMDH) de l'université de Paris 6. La première étape à la conception d'un environnement d'aide à la découverte scientifique appliqué aux milieux granulaires a été de comprendre et d'intégrer la problématique des physiciens de notre équipe. Pour cela, j'ai passé plusieurs mois au sein de leur laboratoire afin de comprendre les problèmes liés à la réalisation d'expérimentations réelles et virtuelles d'empilement de grains. Mais tout d'abord, pourquoi les milieux granulaires sont-ils autant «en vogue» dans la communauté des chercheurs en physique mécanique ?

Après plusieurs siècles d'études de la mécanique des solides, la physique s'est intéressée à la mécanique des fluides, puis à l'extension de la mécanique newtonienne avec la mécanique quantique.

Aujourd'hui, un nouveau défi est lancé à la communauté de chercheurs en physique. Il s'agit de l'étude de systèmes composés d'un grand nombre de particules élémentaires (grains, corps célestes, bulles, etc.). Les lois de comportement physique sont connues pour chacune de ces particules prises indépendamment les unes des autres. Ce nouveau domaine s'appelle la physique des systèmes complexes.

Les lois régissant l'équilibre et la dynamique d'un grain solide sont connues depuis Newton. L'étude du comportement d'objets célestes relève du domaine de l'astrophysique et la modélisation d'une bulle de la mécanique des fluides. Mais l'existence de nombreuses interactions entre ces particules élémentaires rend difficile la compréhension et la modélisation de ces systèmes physiques dans leur ensemble. En effet, si les lois physiques qui régissent le comportement de chaque particule sont connues, le comportement d'un système mettant en jeu un grand nombre de ces particules élémentaires n'est pas prédictible. C'est le cas du problème des N corps où, pour $N > 2$, la trajectoire des corps en interaction ne peut plus être prédite par le calcul des équations.

Il y a quelques années, Pierre-Gilles de Gennes a lancé un appel à la communauté des chercheurs en physique mécanique pour aborder le problème de la physique des milieux granulaires. De Gennes décrit le champ des milieux granulaires comme étant un exemple simple de systèmes physiques complexes [de Gennes 1995, de Gennes 1999]. Effectivement, les empilements de grains posent de nouveaux types de problèmes aux physiciens. Car, contrairement à l'étude d'un solide, les *granulaires* se comportent de façon apparemment imprévisible. Prenons quelques exemples concrets que posent quotidiennement les milieux granulaires, notamment dans les processus industriels.

2.2 Des exemples de problèmes concrets liés aux milieux granulaires

2.2.1 L'exemple des silos

Les silos sont de plus en plus présents dans l'industrie (pharmaceutique, chimique, alimentaire, etc.) pour stocker les matières premières sous la forme de granules ou de poudre. En effet, un nombre croissant de matériaux qui n'ont pas naturellement la forme de grains sont d'abord transformés en granulaires avant d'être employés dans les processus industriels. C'est le cas notamment en chimie où des produits qui entrent dans la composition de réactions chimiques, lorsqu'ils sont sous la forme de granules, ont la propriété de se dissoudre et de réagir beaucoup plus rapidement que lorsqu'ils sont sous la forme d'un bloc solide. Ces grains, plus ou moins calibrés, peuvent ainsi être directement traités dans la chaîne de transformation de ces matières premières en produits finis. Mais le stockage industriel de tels matériaux pose des problèmes et peut même entraîner des catastrophes.

L'écoulement d'un silo est parfois stoppé par la formation d'une voûte de grains (cf. Fig. 2.1). Cette voûte, très solide, supporte alors une bonne partie de la charge du silo et elle est difficile à briser. A l'heure actuelle, les ingénieurs n'ont souvent pas trouvé d'autre moyen que de placer des petits marteaux autour de la paroi. Les marteaux entrent en action et lorsqu'ils atteignent la résonance, la voûte se brise et l'écoulement du silo reprend son cours normal.

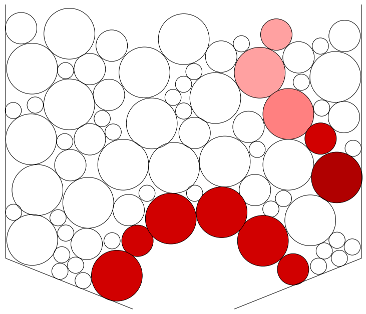


FIG. 2.1 – Voûte dans un silo
La voûte (grains en noir sur la figure) empêche le granulaire de s'écouler librement. Une autre chaîne de forces appuie sur la paroi du silo qui doit pouvoir résister à cette charge.

Sur les silos à faible valeur ajoutée (sable, graviers, etc.), les ouvriers utilisent le plus souvent une masse pour frapper la paroi du silo. Sous ces coups répétés, la paroi se déforme et la probabilité de la formation de voûtes dans le silo est ainsi démultipliée. Récemment, les chercheurs ont trouvé le moyen de briser les voûtes sans déformer le silo en employant de nombreux petits marteaux répartis autour du silo. Ces marteaux sont commandés électroniquement. Ils font vibrer la paroi du silo et donc le granulaire, jusqu'à atteindre la résonance et briser la voûte. Seulement, ce type d'installation est coûteux et n'est employé que pour les grains à forte valeur ajoutée ou dont la manipulation peut s'avérer dangereuse.

De plus, lorsque la voûte est enfin brisée, le granulaire se trouve mélangé à une grande quantité d'air. Il se comporte alors comme un liquide et s'écoule brutalement vers le réceptacle. Le granulaire peut ainsi déborder et causer des pertes voire des dégâts matériels ou mort d'hommes.

La formation de telles voûtes pose également la question de la charge maximale supportée par la paroi au niveau de la voûte. Les ingénieurs qui conçoivent de tels silos se basent sur des théories effectives issues de la mécanique des milieux continus et sur leur savoir expérimental pour évaluer la résistance nécessaire des matériaux à utiliser. Ils intègrent donc souvent des marges de sécurité importantes dans leurs calculs de résistance du silo aux sollicitations mécaniques.

Les industriels souhaiteraient pouvoir acquérir des silos conçus spécialement pour un type de granulaire particulier, avec la garantie d'un écoulement fluide et d'une résistance suffisante sous conditions normales d'utilisation.

Les silos à céréales sont parmi les plus problématiques en termes de sécurité. En effet, certains des grains qui appartiennent à des chaînes de forces vont être broyés (cf. section 2.2.2) si la charge qui transite via cette chaîne est trop importante. Le grain se réduit en une fine poudre. La farine commence alors à fermenter et génère du gaz inflammable. Ajoutons à cela des charançons qui vont eux-aussi grignoter les grains et donc générer de la poussière et du gaz par fermentation. Enfin, plaçons le tout en plein soleil par une chaude journée du mois d'août. Nous obtenons alors un mélange de gaz et de poussières comprimé qui peut exploser spontanément, sans l'aide une étincelle.

La compréhension de ces phénomènes d'écroulement et la capacité à prédire les catastrophes qui en sont parfois les conséquences est d'une importance vitale.

2.2.2 L'exemple du broyage

Un autre problème industriel que posent les granulaires est celui du broyage. Le broyage consiste à transformer un matériau granulaire brut (minerai, gravier, etc.) en grains d'une taille fixée pour pouvoir ensuite être traités automatiquement.

Un procédé très employé, dans le cas du broyage industriel, consiste à introduire une certaine quantité de grains non calibrés dans un tambour à l'intérieur duquel est placé un

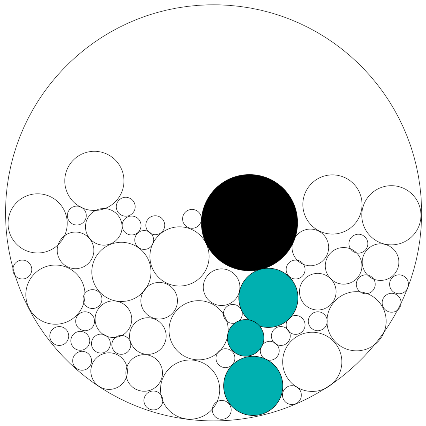


FIG. 2.2 – Exemple d'une ligne de forces dans un broyeur pour granulaires
La charge du boulet, en noir, est supportée par quelques grains seulement. Cette charge étant supérieure à ce que les grains peuvent supporter, un des grains de la chaîne va se briser.

lourd boulet métallique (représenté par le grain noir sur la Figure 2.2). Il suffit ensuite de faire tourner le tambour pour que le poids du boulet, en tombant, écrase le granulaire. Le boulet appuie alors sur les grains de la surface et il se crée un réseau de *chaînes de forces* dans le granulaire, jusqu'à la base du tambour (représentées en grisé sur la figure 2.2). La charge étant plus importante que ce que les grains peuvent supporter, c'est le grain le plus faible de ces chaînes qui va être broyé le premier. En faisant tourner ainsi le système, il est possible d'obtenir une proportion majoritaire de grains de la taille voulue.

Le broyage pose des problèmes essentiels de rendement et de coût énergétique. En effet, la préparation et le calibrage du système pour obtenir une quantité optimale de grains de la taille désirée ne sont pas prédictibles. Les questions qui se posent sont : combien de temps faire tourner le tambour ? quelle charge placer à l'intérieur ? et d'une façon générale, quelles sont les conditions limites qui permettent d'atteindre le rendement optimal [Tsoungui 1998] ? En effet, à chaque fois qu'un grain est broyé, il va donner naissance à des grains plus petits, même s'il s'agissait d'un grain de la taille recherchée. Ainsi, si le système tourne trop longtemps, une grande quantité de poudre risque d'être obtenue au final.

Ce problème de rendement est d'une importance cruciale pour les industriels sachant que l'énergie dépensée pour l'ensemble des opérations de broyage en France est supérieure à celle du secteur économique du transport [Guyon et Troadec 1994, Duran 1997]. Il est alors clair qu'un gain, même minime, du rendement de ces opérations de broyage aurait des retombées économiques importantes.

2.2.3 L'exemple des plaques tectoniques

Un modèle très simple de la géologie terrestre considère qu'il y a du granulaire à l'équilibre statique entre deux plaques tectoniques [Mora et Place 1993, Carlson *et al.* 1994].

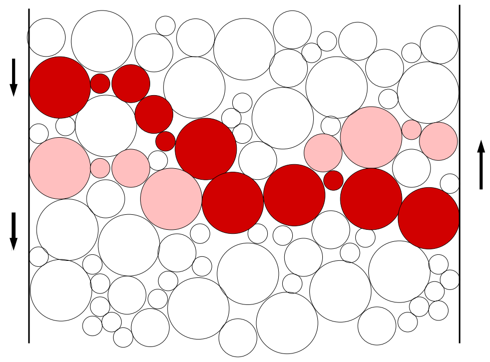


FIG. 2.3 – Exemple de lignes de forces dans un granulaire à l'équilibre entre deux plaques tectoniques
Les deux plaques cisaillent le granulaire qui arrive à résister au déplacement en garantissant un état provisoire d'équilibre. Les lignes de forces qui apparaissent dans le granulaire (en noir et grisé sur la figure) garantissent cet équilibre.

Les plaques cisaillent le milieu granulaire et tant qu'il reste à l'équilibre, il ne se passe rien de visible. Par contre, à l'intérieur du granulaire, des chemins de forces reliant les deux plaques se forment (cf. Fig 2.3). Ces chaînes de forces supportent alors des charges colossales. La valeur d'une chaîne étant toujours la valeur de son maillon le plus faible, ce sera donc le grain le plus faible d'une chaîne qui cédera le premier. La faiblesse d'un grain peut aussi bien se traduire par la fragilité de sa structure que par la précarité de son équilibre. Dans le premier cas, il sera broyé (cf. exemple précédent) et dans le second, il sera contraint à se déplacer sous la pression de son voisinage. Lors de tels incidents, si le granulaire réussit à retrouver un état d'équilibre, une simple secousse sismique sera perçue. Par contre, dans le cas où le granulaire ne peut plus garantir l'équilibre statique, les deux plaques vont brusquement glisser l'une sur l'autre (ou l'une contre l'autre) et engendrer alors un véritable séisme.

Cet exemple nous montre que la compréhension du comportement d'un milieu granulaire à l'équilibre statique est une étape nécessaire pour la conception d'un modèle prédictif pour les séismes.

2.3 Un problème central : comprendre la structure du réseau de forces dans un tas

Les problèmes décrits ci-dessus sont de nature très différentes mais ils ont un point en commun : *le réseau de forces dans un granulaire à l'équilibre statique.* Ainsi, les chercheurs en physique des milieux granulaires tentent actuellement de comprendre la structure et la dynamique du réseau de forces qui sous-tend l'équilibre d'un milieu granulaire. C'est le problème central qui permettra ensuite de développer des applications pour nombre de problèmes concrets.

Visualisation de réseaux de forces

Les premières expérimentations [Dantu 1957], menées sur des empilements de cylindres de plexiglas, ont permis de mettre visuellement en évidence la présence d'un réseau de forces dans un tas à l'équilibre statique.

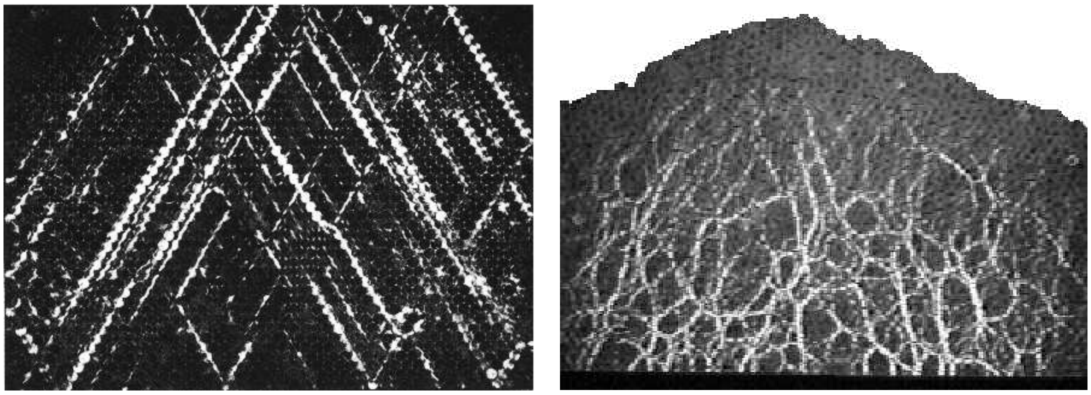


FIG. 2.4 – Visualisation du réseau de lignes de forces dans des tas réels
a) un tas confiné et fortement comprimé b) un tas ne supportant que son propre poids

Ces expérimentations se basent sur le principe de la photo-élasticité [Travers *et al.* 1986, Travers *et al.* 1987, De Josselin de Jong et Verruijt 1969]. L'empilement de cylindres en plexiglas est éclairé entre lumières polarisantes[1] et sans aucune contrainte, il apparaît comme sombre. Les premières expérimentations ont été menées avec des grains de type sphères dures et régulières. Ces expériences nécessitaient l'application d'une force très importante sur l'empilement (par confinement, par exemple) pour pouvoir observer des lignes de forces dans le milieu granulaire. En effet, il faut que la déformation du cylindre soit suffisamment importante pour qu'un grain puisse faire tourner

[1] il s'agit de polariseurs circulaires inverses

le plan de polarisation de la lumière et ainsi apparaître localement plus lumineux (cf. Fig. 2.4.a). D'une façon générale, si la déformation du cylindre n'est pas du même ordre de grandeur que la longueur d'onde de la lumière émise, rien n'apparaît.

Ce n'est que très récemment que l'équipe du Professeur Robert P. Behringer de l'Université de DUKE a visualisé [Howell *et al.* 1997], en utilisant des matériaux extrêmement sensibles et des propriétés de biréfringence de la lumière, les chemins de forces dans un tas qui ne supporte *que son propre poids* (cf. Fig. 2.4.b). Dans ce cas, les grains utilisés étant très malléables, il n'y a pas besoin d'appuyer fortement sur le tas pour déformer suffisamment les grains et voir apparaître les lignes de forces.

2.4 Les difficultés techniques pour réaliser des expérimentations réelles

D'une manière générale, les mesures macroscopiques de contraintes effectuées sur des empilements réels demandent des conditions expérimentales draconiennes pour obtenir des valeurs reproductibles [Kolb *et al.* 1999, Vanel et Clément 1999]. Des fluctuations importantes sont souvent observées à cause de la taille finie des capteurs de contraintes [Howell *et al.* 1997, Kolb *et al.* 1999, Vanel et Clément 1999]. De plus, des mesures locales sous un tas ont montré que les fluctuations des forces de contact sont de l'ordre de la moyenne [Liu *et al.* 1995, Mueth *et al.* 1998]. Il semble que ces chaînes de forces soient la principale cause de ces fluctuations et qu'elles soient aussi responsables de la propagation anormale du son observée dans les milieux granulaires [Liu et Nagel 1992]. La première difficulté surgit lorsqu'il s'agit simplement de mesurer, à l'aide de capteurs, la charge à la base d'un empilement.

2.4.1 Le problème des capteurs

En physique des milieux granulaires, il est difficile d'obtenir des expérimentations réelles reproductibles car, en plus du désordre mécanique dû à la présence de lignes de forces, la mesure est souvent «destructive».

En effet, des chercheurs ont mesuré la taille du diamètre de la surface de contact entre deux grains, elle est de l'ordre de 10 à 20 μm. Ainsi, l'utilisation d'un capteur de force qui se déformerait de plus de 10 μm sous la pression du tas entraînerait localement un affaissement du grain et la possible perte de contact avec les grains du dessus. Cet affaissement va surtout entraîner localement un réarrangement de la charge sur les grains voisins. Loïc Vanel [Vanel 1999] préconise donc l'utilisation de capteurs quasi-infiniment raides (par exemple, de type capacitif) afin de ne pas biaiser les mesures par une déformation locale des contraintes. Une fois ce problème de capteurs résolu, d'autres problèmes surviennent, comme la maîtrise des conditions expérimentales (conditions limites).

2.4.2 Les conditions limites

Des expérimentations qui ont été menées, en continu sur plusieurs jours, ont montré que la pression en bas d'un silo au repos variait avec la température de la pièce. Ainsi, lorsque la température augmente (de 2 à 3 degrés Celcius pendant la journée), les grains vont se dilater très légèrement et la pression en bas du silo va diminuer du fait de la formation de voûtes dans le tas. A l'inverse, lorsque la température diminue (pendant la nuit), les grains se rétractent et les contraintes vont alors se relaxer, les voûtes se briser et la pression en bas du silo va augmenter.

De même, le simple fait de marcher dans le couloir qui jouxte la salle d'expérimentations peut fausser les mesures à cause des vibrations. Il faut alors pouvoir isoler le montage expérimental de telles vibrations, voire déménager entièrement le montage expérimental dans un endroit stable où il y a peu de passage.

Enfin, si l'on souhaite réaliser un empilement parfaitement régulier de type «boulets de canon», il faut alors prendre des grains fabriqués dans un matériau très dur et qui soient calibrés au micron près de façon à être à peu près sûr de la régularité du réseau des contacts effectifs dans le tas.

Ces exemples soulignent l'importance de la maîtrise des conditions limites d'une expérimentation comme étant une condition nécessaire pour l'obtention de résultats reproductibles. Les chercheurs en physique des milieux granulaires sont ainsi confrontés quotidiennement à des dizaines de petits problèmes techniques avant de pouvoir mener une expérimentation jusqu'à son terme. Pierre-Gilles de Gennes [Duran 1997] parle de la physique des milieux granulaires comme étant non pas une physique coûteuse mais soignée.

La maîtrise des conditions limites d'une expérimentation étant une condition nécessaire à l'obtention de résultats reproductibles, les chercheurs en physique des milieux granulaires étudient des empilements les plus simples possibles mais qui puissent néanmoins rendre compte de phénomènes physiques macroscopiques intéressants et reproductibles.

2.5 Des empilements simples

La simplicité d'un empilement se définit plutôt comme l'absence de désordre des grains, mais d'autres paramètres entrent aussi en considération.

La dimension de l'espace est le premier élément caractéristique d'une expérimentation. Les difficultés techniques pour la réalisation d'un empilement en 3D sont bien plus importantes que pour un empilement en 2D, surtout si les chercheurs souhaitent maîtriser parfaitement la géométrie du réseau des contacts entre les grains.

Le désordre géométrique va entraîner un désordre du réseau des contacts entre les grains. Ce désordre est mesuré à l'aide de descripteurs comme la *densité* de l'empilement mais aussi en regardant la répartition de la distribution des contacts sur l'ensemble du tas.

Le désordre mécanique est induit par la présence de phénomènes tels que le frottement entre grains ou grain-paroi, la cohésion entre grains ou encore l'élasticité des contacts. Ainsi, à partir d'un empilement géométrique parfaitement régulier, il est possible de jouer sur la rugosité des grains pour créer un certain désordre mécanique.

La méthode d'empilement est un autre paramètre qui permet de créer simultanément un désordre géométrique et mécanique. En effet, la structure géométrique d'un empilement est très différente selon qu'il a été créé à partir d'une pluie uniforme de grains ou à partir d'une source ponctuelle. La dynamique d'empilement du tas va alors se traduire par la mémorisation des contraintes et de leurs directions successives pour conduire, au final, à des configurations géométriques et mécaniques différentes.

Les conditions limites du système permettent aussi de jouer sur le désordre de l'empilement. Par exemple, si un appui local est exercé en haut du tas, cette surcharge va alors diriger les contraintes dans le sens de l'appui. De même, le cisaillement d'un tas confiné va orienter les contraintes dans le sens du déplacement latéral.

Ainsi, les chercheurs en physique des milieux granulaires ont un très large éventail de possibilités pour créer du désordre dans un empilement. Mais, d'une manière générale, comme la caractérisation de l'influence de chacun de ces paramètres n'est pas clairement établie, les chercheurs étudient les conséquences mécaniques pour un seul de ces paramètres à la fois.

2.6 Le besoin de simulations numériques

Les mesures réalisées dans le cadre des expérimentations réelles sont assez peu informatives sur la structure interne du réseau de forces. Elles ne permettent pas d'appréhender facilement la structure et le désordre des réseaux de forces dans un empilement.

La simulation numérique est donc un outil essentiel [Radjai 1995] pour la réalisation des modèles conçus par les physiciens. Ces simulations permettent souvent de faire des hypothèses et les expérimentations réelles jouent alors un rôle clé dans la validation de ces modèles. L'intérêt de cette interaction entre expérimentations réelles et virtuelles est double.

Dans un premier temps, les expérimentations réelles permettent de faire des hypothèses sur la présence de phénomènes macroscopiques. La vérification de ces hypothèses est cependant souvent irréalisable car elle demanderait un trop grand nombre d'expérimentations réelles, difficiles à réaliser, pour couvrir l'espace des solutions. La simulation permet alors de valider ou d'infirmer certaines hypothèses et même de cerner les conditions limites pour lesquelles une hypothèse est vérifiée. C'est un point important, car ensuite, les chercheurs vont pouvoir conduire à nouveau des expérimentations réelles en ayant des informations précises sur les conditions limites et les mesures à réaliser.

Dans un deuxième temps, les expérimentations réelles sont indispensables pour la validation des modèles théoriques. Si une hypothèse qui est valide d'un point de vue numérique n'est pas vérifiée en pratique, les chercheurs vont alors chercher à comprendre la cause de cette apparente incohérence, peut-être modifier le modèle théorique et recommencer les étapes de simulation/validation.

2.7 Conclusion

Dans ce chapitre, nous avons présenté la physique des milieux granulaires comme un exemple simple de systèmes physiques complexes. A partir d'exemples concrets de problèmes liés aux milieux granulaires, la problématique de l'étude du réseau de forces dans un tas à l'équilibre statique s'est révélée être la clé de voûte de nombreuses applications industrielles et humaines.

Ensuite, nous avons exposé les difficultés techniques rencontrées par les chercheurs en physique des milieux granulaires pour la réalisation d'expérimentations réelles. Ces difficultés poussent les physiciens de ce domaine à considérer des empilements les plus simples possibles afin de pouvoir obtenir des phénomènes reproductibles. A l'aide de ces expérimentations, les chercheurs émettent des hypothèses, qu'en pratique, seules des simulations numériques peuvent venir étayer.

Dans le chapitre suivant, nous abordons la modélisation d'un empilement compact de sphères dures et régulières. Des algorithmes classiques de simulation de l'équilibre statique pour ce type d'empilement y sont décrits.

Chapitre 3

Modélisation du problème physique et algorithmes classiques

Sommaire

3.1 Introduction

Actuellement, la découverte de modèles physiques permettant d'expliquer les phénomènes décrits au chapitre précédent intéresse nombre de chercheurs. En effet, les conséquences d'une telle découverte et de la compréhension de lois dans ce domaine auraient des répercussions multiples.

D'une part, cela permettrait de faire progresser la recherche en physique des systèmes complexes. Des algorithmes de simulation et bien d'autres techniques pourraient être recyclées dans d'autres domaines de la physique tels que la physique particulaire, l'astrophysique, etc. Ainsi, plutôt que de commencer ex-nihilo la recherche dans un nouveau domaine, l'analogie avec les milieux granulaires pourrait être utilisée pour fournir une base appréciable de protocoles expérimentaux. Des lois de comportements macroscopiques vérifiées pour les granulaires pourraient ainsi être le point de départ d'une recherche dans un domaine de la physique des systèmes complexes.

D'autre part, d'un point de vue économique, les retombées d'une telle découverte seraient très importantes car elle permettrait notamment de réduire le coût énergétique de transformation des granulaires mais aussi de sécuriser les installations industrielles où sont stockés de tels matériaux.

Cette quête se heurte cependant à des difficultés d'ordres physique et algorithmique comme nous allons le voir dans ce chapitre. La section suivante présente d'abord les difficultés théoriques liées à la modélisation physique des interactions entre les particules. Dans cette section, nous présentons plus en détail un modèle théorique proposé par Eric Clément et Christophe Eloy pour caractériser l'équilibre d'un grain dans un tas à l'équilibre statique ainsi que leur algorithme de simulation. Ensuite, nous présentons les algorithmes classiquement employés par la communauté «matériaux granulaires» et leurs limitations.

3.2 Les difficultés du choix d'un modèle théorique

Afin de simuler des empilements de grains à l'équilibre statique, les chercheurs en physique ont besoin d'un modèle caractérisant les interactions entre les particules. Pour définir un tel modèle, il leur est nécessaire de se poser les questions suivantes :

Quoi mesurer et à quelle échelle se positionner ? La première étape est de savoir ce qu'il est souhaitable de mesurer et d'observer ainsi que le niveau auquel il faut se placer pour effectuer ces mesures. En effet, si par la suite le désordre mécanique dans un empilement doit être mesuré globalement, l'utilisation d'un modèle «jouet» purement théorique comme celui de Claudin [Claudin 1999] est souvent suffisant.

Dans ce modèle, les lignes de forces sont représentées par des rayons lumineux que des défauts dans le milieu (représentant localement un désordre mécanique) vont diviser et dévier. Il peut être également souhaitable d'obtenir une solution mathématique exacte des équations d'équilibre statique pour chaque grain, etc. D'autre part, le niveau auquel les mesures sont effectuées est appelé le niveau macroscopique. Ce niveau définit une borne supérieure pour la modélisation des interactions locales qui sera forcément d'un niveau d'ordre inférieur.

A quelle échelle modéliser les interactions locales ? Vient ensuite un problème important d'échelle. A quel niveau de détails faut-il descendre pour rendre compte des phénomènes d'interactions entre grains ? La question est de savoir s'il faut se placer à l'échelle microscopique du contact inter-grain ou bien à l'échelle du grain et considérer les contacts comme étant ponctuels ou même à l'échelle d'une ligne de forces comme dans le cas du modèle de Claudin. Le premier réflexe est de considérer l'échelle la plus petite et d'essayer de résoudre toutes les équations de la physique qui entrent en jeu. Seulement, les difficultés numériques pour simuler des processus multi-échelles rendent les chercheurs pragmatiques. Ils se placent donc à un niveau qui n'est pas trop exagérément petit pour ce qu'ils veulent mesurer au niveau macroscopique.

Quelle est la nature des interactions locales ? Enfin, la dernière difficulté consiste à choisir une des théories de l'interaction solide sur solide. Certains modélisent le contact inter-grain sous sa forme élastique quand d'autres introduisent des forces électrostatiques, de la cohésion ou du frottement (au sens de Coulomb [de Coulomb 1776] ou de Signorini [Signorini 1933]), etc. Ensuite l'équilibre d'un grain peut être aussi interprété de diverses façons. Certains considèrent simplement les équations d'équilibre statique, d'autres ajoutent des contraintes sur le couple de rotation du grain ou encore imposent que les forces normales soient dirigées vers le grain, etc.

Les multiples façons de modéliser les contacts inter-grains, l'équilibre d'un grain et celui d'un tas peuvent conduire à appréhender des milliers de modèles possibles. Devant cette combinatoire de modèles, les chercheurs ne se concentrent que sur les modèles cohérents et dont les solutions peuvent être calculées par simulation numérique.

Une première heuristique pour limiter encore la complexité de l'espace des modèles est de considérer un type de modèle «jouet», dont la complexité est réduite au maximum, afin de vérifier qu'il donne des résultats macroscopiques intéressants. Ensuite, en complexifiant ce premier modèle, il est possible de s'interroger sur la «robustesse» des lois découvertes auparavant, en vérifiant qu'elles sont toujours présentes. Ce processus cognitif est assez proche du cycle incrémental en spirale de conception de systèmes en Génie Logiciel.

De même, les conditions limites peuvent avoir une grande influence sur les comportements des empilements. Les chercheurs considèrent donc, en première approche, des conditions limites simplifiées, voire non réalistes d'un point de vue physique. Toutes ces simplifications permettent de limiter les effets dûs aux conditions limites et donc de dégager des hypothèses fiables pour un comportement observé.

Pour passer du niveau micro au niveau macro, i.e. du modèle du grain à des éléments de réponses concernant des lois macroscopiques dans un tas, la simulation numérique est un partenaire essentiel pour ces chercheurs [Radjai 1995]. Ils en ont besoin pour valider les hypothèses et les lois qu'ils pensent avoir découvertes lors d'expérimentations réelles ou au cours de simulations basées sur des modèles plus simples.

3.3 Les algorithmes classiques de simulation numérique

Depuis quelques années déjà, les chercheurs en physique des milieux granulaires se sont intéressés à la simulation des tas de sable et des algorithmes généraux ont émergé de ces diverses expériences. Nous allons présenter les principaux algorithmes existants et leurs conditions d'application aux milieux granulaires à l'équilibre statique uniquement. Ces algorithmes sont le plus souvent issus d'analogies avec un autre domaine de la physique pour lequel des algorithmes originaux avaient été développés auparavant. C'est le cas de la *dynamique moléculaire* empruntée au domaine des gaz et de la *méthode de calculs par éléments finis* issue d'études sur les matériaux continus. Les deux autres algorithmes présentés sont dérivés des équations des équilibres des particules dans le système. Le tableau 3.1 résume les conditions limites et les domaines d'inspirations de ces algorithmes.

3.3.1 Automates Cellulaires (CA)

Les automates cellulaires [Codd 1968] offrent des possibilités intéressantes quant à la simulation d'empilements à l'équilibre statique. Chacun des grains peut être représenté par un automate dont les entrées sont les grains voisins du grain courant. Les états de l'automate peuvent être identifiés aux états d'un grain (à l'équilibre, contact glissant, en rotation, etc.). Les opérateurs de passage d'un état à un autre peuvent être également définis.

Cette voie n'a pas été explorée par les chercheurs en physique des milieux granulaires car cette approche leur pose un problème physique conceptuel. Cet obstacle réside dans le

Type d'algorithme	**Domaine d'inspiration**	**Modélisation microscopique du grain**	**Domaine d'application et problématique**	**Taille des tas résolus**	**Temps de calcul approximatif**
AUTOMATES CELLULAIRES (CA) [Mora et Place 1993]	Théorie des jeux	Une cellule spatiale avec des conditions limites propres	Géologie (flux magmatiques)	200×200	1/2 jour
MÉTHODE PAR ELÉMENTS FINIS (FEM) [Tsoungui 1998]	Mécanique des matériaux continus	Chaque cellule possède des propriétés mécaniques propres	Compression et broyage des milieux granulaires	40×30	1/2 jour
DYNAMIQUE DES CONTACTS (CD) [Moreau et Jean 1996]	Mécanique des solides	Elasticité des contacts + équations de la dynamique du grain	Tas à l'équilibre statique ou en mouvement (cisaillement, compression, réponse de Green, etc.)	40×30	1/2 jour
DYNAMIQUE MOLÉCULAIRE (MD) [Radjai 1995]	Théorie des gaz	Agitation d'un grain, collisions avec pertes d'énergie	Milieux granulaires (équilibre quasi-statique)	100×100	1/2 jour (ne converge plus à grande précision)
MC-GRANU [Eloy et Clément 1997]	Algorithme de type Monte-Carlo	Equations de l'équilibre du grain + frottement de Coulomb aux contacts	Empilement statique de type «boulets de canon»	max. 50 grains en largeur, hauteur non limitée	25×25, $\mu = 0.5$ en 4 min 40 secs
GRANUSOLVE [Breton *et al.* 2000]	Systèmes multi-agents, Eco-résolution	idem MC-Granu	idem MC-Granu	100 millions de grains	300×300, $\mu = 0.5$ en 35 secs

TAB. 3.1 – Les différents algorithmes de simulation numérique de tas de sable

caractère «artificiel» du passage d'un état physique à un autre. Pour un physicien, il est difficilement concevable d'avoir des états intermédiaires qui vont à l'encontre des lois de la physique, même si ce n'est que de façon temporaire. Par contre, les automates cellulaires ont été utilisés dans le cadre de la simulation de flux de matières, les cellules caractérisant alors une portion de l'espace [Mora et Place 1993, Carlson *et al.* 1994, Bubak 1995].

Le seul défaut de cette méthode vient de l'aspect centralisé de la résolution globale du système et du caractère homogène des automates. La résolution centralisée et déterministe ne permet pas l'ajout de connaissances du domaine pour la résolution du problème. De plus, il est difficile d'ajouter des entités hétérogènes dans un tel système (parois, plancher, etc.).

3.3.2 MÉTHODE PAR ELÉMENTS FINIS (FEM)

La méthode de calcul par éléments finis (FEM) a été conçue, à l'origine, pour simuler le comportement de matériaux continus soumis à de fortes sollicitations. Le principe de cette méthode consiste à décomposer géométriquement la structure à étudier en des éléments plus petits de dimensions finies. Ces éléments sont alors inter-connectés par des nœuds représentant leurs frontières respectives. Les déplacements à l'intérieur de chaque élément sont caractérisés de façon simple à l'aide des nœuds et des lois de résistance du matériau dont l'élément est constitué.

Cette méthode permet ainsi de simuler de façon très précise le comportement d'une structure géométrique soumise à de fortes sollicitations mécaniques comme, par exemple, la carrosserie d'une voiture. Les succès industriels de cette méthode sont incontestables (résistance de matériaux hétérogènes) mais son application à un milieu discret est difficile.

Dans le cas des milieux granulaires, l'application de cette méthode requiert de diviser l'empilement en autant d'éléments que de grains. Les nœuds représentent alors les contacts inter-grains. La difficulté est qu'il faut définir pour chaque grain ses conditions limites et ses propriétés mécaniques. Cette opération est coûteuse en temps de calcul lorsqu'il y a plusieurs centaines de grains dans le système. Ensuite, pour chaque élément, la propagation des contraintes et le calcul de la déformation mécanique engendrée limitent la taille des tas considérés à quelques centaines de grains tout au plus [Tsoungui 1998].

Cette méthode pourrait s'appliquer à des problèmes de fortes contraintes sur des empilements, comme dans le cas du broyage ou de la compression de grains mous. Mais pour une étude statistique de larges empilements, cette méthode se révèle impraticable.

3.3.3 Dynamique des Contacts (CD)

La méthode de simulation de dynamique des contacts (CD) a été introduite par Jean-Jacques Moreau [Moreau et Jean 1996, Radjai 1998]. Cette méthode est directement issue des équations de la dynamique du grain et de ses contacts. Ces équations caractérisent les différents états des contacts et du grain au cours du temps. Cette méthode s'applique donc tout particulièrement au cas d'empilements avec une certaine dynamique des grains, comme le cisaillement [Radjai 1995, Radjai *et al.* 1997, Radjai 1997, Radjai *et al.* 1998].

A l'aide de ces équations, l'état de chacun des contacts du grain est choisi aléatoirement. Un contact peut-être à l'équilibre statique ou bien glissant. Les liens glissants sont considérés inactifs et seuls les liens non glissants sont pris en compte dans les équations de la dynamique du grain.

Dans le cas de l'étude d'empilements à l'équilibre statique, la méthode de la dynamique des contacts autorise un certain degré de liberté au niveau de chaque grain et de ses contacts. Si les forces qui sont appliquées sur le grain ne permettent pas d'atteindre l'équilibre statique, l'état des contacts (actifs/inactifs) peut être modifié pour arriver à une solution.

Les résultats obtenus sont assez encourageants et la taille des empilements résolus est de l'ordre de quelques milliers de grains (typiquement un tas 200x200 prend quelques heures de calculs) [Radjai 1998].

Le problème de la méthode de la dynamique des contacts est qu'elle se prête mal à l'étude statistique de larges empilements à l'équilibre statique et non en mouvement. Les limitations en taille et en temps de calcul de cette méthode rendent difficile une étude statistique d'empilements à l'équilibre statique. En effet, les fluctuations des paramètres observés étant de l'ordre de la moyenne, plusieurs milliers de réalisations sont nécessaires pour pouvoir dégager des comportements macroscopiques fiables.

3.3.4 Dynamique Moléculaire (MD)

La méthode de simulation dite dynamique moléculaire (MD) est empruntée au domaine des gaz [Rapaport 1995]. Un empilement de grains est considéré, dans ce cadre, comme étant constitué de particules élémentaires ayant chacune une énergie propre. Des collisions, avec dissipation d'énergie, se produisent entre les particules. Les physiciens tentent alors de «refroidir» le système en faisant baisser l'énergie totale jusqu'à ce que l'agitation soit suffisamment faible pour considérer le tas comme étant globalement à l'équilibre statique [Brendel et Dippel 1998].

Malheureusement, la convergence n'est plus assurée lorsque la précision souhaitée pour les équations de l'équilibre statique est trop importante, i.e. pour une agitation inférieure à un ϵ donné. En effet, pour de faibles énergies de mouvement, les chocs ne sont presque plus dissipatifs et les grains restent dans un état faiblement agité mais non statique. Le problème est que si cet état est considéré comme étant un état d'équilibre quasi-statique, cela revient à approximer la loi de frottement de Coulomb des contacts inter-grains [de Coulomb 1776]. Cette approximation peut alors entraîner l'obtention de solutions localement non physiques mais surtout elle implique de considérer des contacts déformables et donc d'introduire implicitement une loi d'élasticité dans le modèle [Radjai 1995].

Cette méthode donne des résultats très analogues en qualité et en temps de calculs à ceux de la Dynamique des Contacts, malgré des domaines d'inspiration très différents.

3.4 Le modèle de Clément et d'Eloy

Dans cette section, nous allons nous pencher sur l'étude d'un modèle simple de l'équilibre statique d'un empilement proposé par Eric Clément et Christophe Eloy [Eloy et Clément 1997] du Laboratoire des Milieux Désordonnés et Hétérogènes (Paris

6). Ce modèle caractérise l'équilibre d'un grain sous la contrainte de frottement à ses contacts au sens de Coulomb.

Nous détaillons les équations et les conditions limites de ce modèle car notre algorithme GranuSolve (cf. Chap. 4) se fonde sur ces équations pour caractériser l'équilibre d'un grain.

3.4.1 La description des conditions limites

Clément et Eloy proposent de s'intéresser à l'étude d'empilements bidimensionnels en boulets de canon de sphères dures et régulières (cf. Fig. 3.1). Le modèle qu'ils proposent quoique très académique représente une étape nécessaire selon eux à la réalisation de simulations mais aussi à l'élagage de l'espace des hypothèses sur les lois observées.

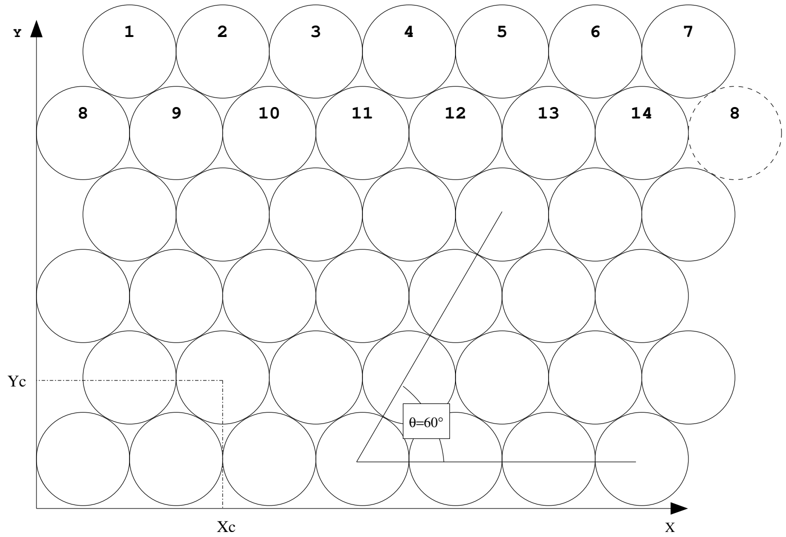


Fig. 3.1 – Empilement compact triangulaire

Afin de s'affranchir des effets dûs aux conditions aux limites, ils supposent que le tas est cylindrique et qu'il ne possède pas de plancher. Ainsi, le grain numéro 7 aura pour voisins les grains numéros 1, 6, 8 et 14 (cf. Fig. 3.1). Matériellement, il faut s'imaginer un mur de grains (2D), sans paroi donc cylindrique, i.e. un mur qui se referme sur lui-même.

L'empilement est de type «boulets de canon», i.e. tous les grains sont parfaitement réguliers avec un diamètre unitaire identique. Cet empilement est le plus compact et le plus stable qui puisse exister en dimension deux et trois. De plus, les contacts latéraux (il n'y a pas de contact effectif entre les grains 4 et 5) ont été volontairement négligés pour simplifier encore le modèle. Enfin, l'ensemble du système est soumis à un champ gravitique g paramétrable.

De plus, pour que le tas soit à l'équilibre statique, chaque grain doit être à l'équilibre au sens des forces qui s'appliquent sur lui. Ce sont simplement les équations de l'équilibre d'un solide de Newton.

3.4.2 Le frottement μ : un paramètre de désordre mécanique

Le frottement est le seul paramètre de désordre dans ce modèle. En effet, le désordre géométrique est inexistant du fait de la régularité de l'empilement et donc du réseau de contacts. Le désordre mécanique est représenté par la présence de frottement qui traduit, d'une certaine manière, la rugosité des grains.

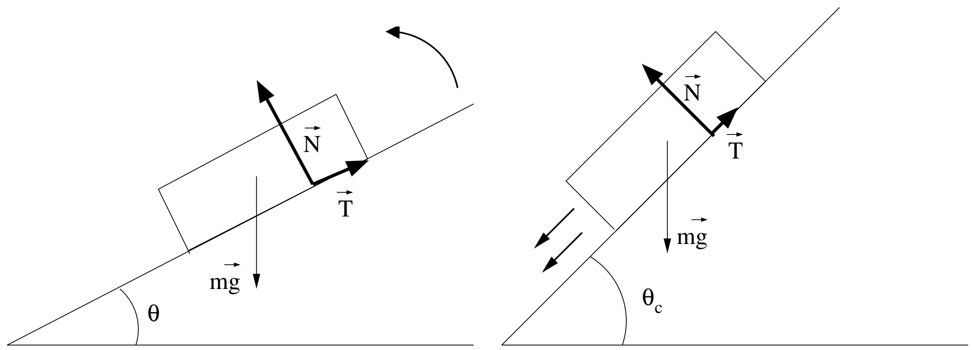


FIG. 3.2 – Le frottement solide sur solide
Tant que l'angle du plan incliné est inférieur à θ_c, la brique reste à l'équilibre. Au-delà, la brique décroche.

Prenons l'exemple d'une brique posée sur un plan incliné (cf. Fig. 3.2). En partant d'une configuration initiale où le plan est à l'horizontale, montons progressivement le plan incliné jusqu'à ce que la brique décroche et se mette en mouvement. Notons θ_c l'angle critique à partir duquel la brique décroche. Nous expérimentons ainsi la loi découverte par Coulomb [de Coulomb 1776] qui postule que cet angle est constant quel que soit la surface de contact (à poids constant). Ainsi, à chaque paire de matériaux correspond un cœfficient de frottement unique, noté μ, qui n'est autre que la tangente de l'angle critique : $\mu = \tan(\theta_c)$.

D'un point de vue mécanique, cela signifie que pour être à l'équilibre, la force qu'exerce la brique sur le plan incliné doit être comprise entre deux valeurs qui définissent le *cône de Coulomb*. Mais il est important de noter que cette force peut s'orienter dans n'importe quelle direction à l'intérieur de ce cône. La présence de frottement dans notre modèle, au sens de Coulomb, entraîne donc une indétermination des forces de contact. Il en va de même pour l'équilibre statique entre deux grains, il faut que la force de contact soit comprise dans son cône de frottement de Coulomb. Ainsi, plus le cœfficient de frottement est faible (grains peu rugueux) et moins il y aura de valeurs possibles pour la valeur des forces de contact inter-grains. Inversement, plus la valeur de frottement est grande, plus l'espace des possibles sera grand et ce en chacun des points de contact inter-grains.

3.4.3 Les équations du modèle

Les équations de l'équilibre d'un grain soumis à des forces de contact et à la gravité sont données par les lois de Newton (cf. Fig. 3.3). Une étude détaillée du modèle est donnée en annexe A.2.

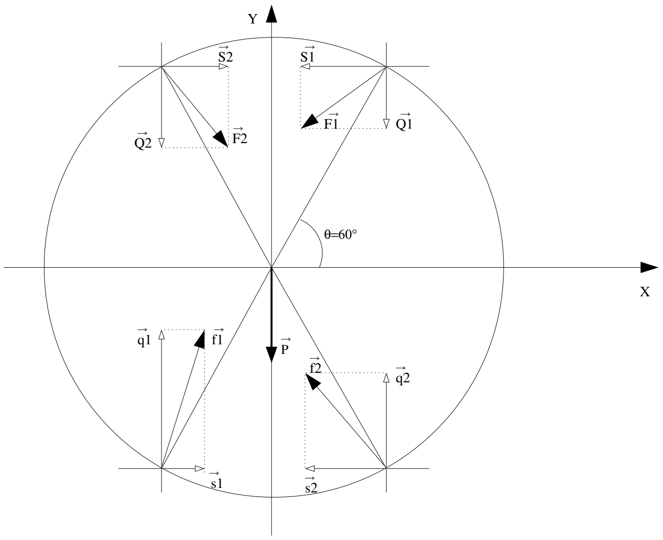


FIG. 3.3 – Forces de contact s'appliquant sur un grain
Pour que le grain soit à l'équilibre statique, il faut que la somme vectorielle des forces de contacts qui lui sont appliquées soit nulle.

Tous les grains doivent satisfaire le même système de trois équations suivant :

$$\begin{cases} S_1 + S_2 + s_1 + s_2 & = & 0 \\ Q_1 + Q_2 + q_1 + q_2 + mg & = & 0 \\ Q_1 - Q_2 - q_1 + q_2 + (-S_1 - S_2 + s_1 + s_2)\tan(\theta) & = & 0 \end{cases} \quad (3.1)$$

Les deux premières équations traduisent l'équilibre pour les projections horizontales et verticales, la troisième l'équilibre pour le couple (produit tensoriel nul). Sur la figure 3.3, nous avons représenté un exemple de jeux de forces qui s'appliquent sur un grain pour un angle de contact $\theta = 60°$. Les forces qui s'appliquent sur la partie supérieure du grain sont notées en majuscule (S_1, Q_1, S_2 et Q_2) et celles qui s'appliquent sur la partie inférieure en minuscule (s_1, q_1, s_2 et q_2). Les Q (resp. S) représentent la projection verticale (resp. horizontale) de la force de contact pour un angle de contact $\theta = 60°$.

De plus, deux autres contraintes sont ajoutées en chacun des contacts. D'une part, les forces normales doivent être en compression, i.e. les forces ne doivent pas «remonter» (elles doivent s'appliquer sur le grain et non l'inverse). Cette première contrainte se traduit par une inégalité pour chacune des forces de contact. Par exemple, pour le premier contact en haut à droite du grain :

$$Q_1 \tan(\theta) + S_1 \quad \leq \quad 0$$

D'autre part, une deuxième contrainte est ajoutée au niveau des forces de contact afin qu'elles soient contenues dans leur cône de frottement de Coulomb. Ce cône définit la limite statique du contact solide sur solide. Cette contrainte se traduit par une double inégalité pour chacune des forces de contact. Par exemple, cette seconde contrainte se traduit de la façon suivante, en notant μ le cœfficient de frottement du matériau :

$$\begin{aligned} Q_1(\mu \tan(\theta) - 1) + S_1(\tan(\theta) + \mu) &< 0 \\ Q_1(\mu \tan(\theta) + 1) + S_1(-\tan(\theta) + \mu) &< 0 \end{aligned}$$

Chaque grain va donc posséder un jeu de forces qui satisfait simultanément les conditions d'équilibre statique et les contraintes d'orientation et de frottement à ses quatre points de contact. A noter que la condition de Coulomb, valable en chaque contact, est une double inégalité, un empilement admet ainsi un grand nombre (en fait, une infinité) de solutions qui satisfont toutes les conditions de l'équilibre statique.

3.4.4 Le degré de liberté p d'une solution induit par le frottement

D'un point de vue algorithmique, maintenant, la présence de frottement se traduit par un intervalle de possibilités pour chaque force de contact. Ainsi, pour les équations d'équilibre du grain (cf. équations 3.1) et lorsque les forces supérieures sont connues (en majuscules), le système de trois équations à quatre inconnues (q_1, s_1, q_2 et s_2) à résoudre est le suivant :

$$\left\{ \begin{array}{lcl} s_1 + s_2 & = & -S_1 - S_2 \\ q_1 + q_2 & = & -Q_1 - Q_2 - mg \\ q_1 - q_2 - \tan(\theta)(s_1 + s_2) & = & Q_1 - Q_2 - \tan(\theta)(S_1 + S_2) \end{array} \right. \tag{3.2}$$

Nous sommes donc face à une indétermination qui nous oblige à faire un choix pour une des deux forces de contact. Plus précisemment, le système d'équations (3.2) montre que les valeurs de q_1 et de q_2 sont parfaitement déterminées par les valeurs des forces des contacts supérieurs. Seules les valeurs des composantes horizontales (s_1 et s_2) sont indéterminées. Posons $p = s_1 - s_2$, le système devient alors :

$$\begin{cases} q_1 &= -Q_2 - \frac{mg}{2} - \tan(\theta)(S_1 + S_2) \\ q_2 &= -Q_1 - \frac{mg}{2} + \tan(\theta)(S_1 + S_2) \\ s_1 &= -\frac{1}{2}(S_1 + S_2 - p) \\ s_2 &= -\frac{1}{2}(S_1 + S_2 + p) \end{cases} \quad (3.3)$$

Le paramètre p sera à choisir dans un intervalle défini par les contraintes d'orientation et de frottement des forces de contact. Mais, il se peut très bien que les forces que nous considérons comme connues (ici, les forces des contacts supérieurs) conduisent à une impossibilité qui se traduit alors par un intervalle vide pour le choix de la valeur de p. Cette situation est illustrée par le schéma 3.4.

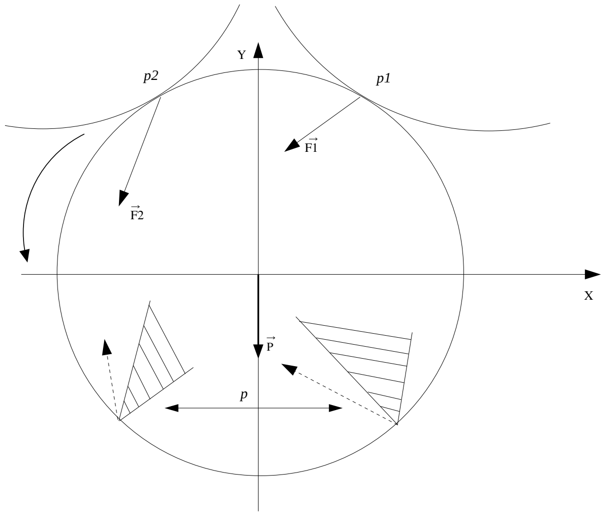


FIG. 3.4 – Un cas où les forces des contacts supérieurs rendent impossible l'équilibre du grain
Intuitivement, cela revient à trop soulever le plan incliné sur lequel est posé la brique (forces supérieures). La valeur de frottement (forces inférieures) ne suffit pas pour garantir des forces de contacts statiques.

Sur ce schéma, les forces qui s'appliquent sur les contacts supérieurs du grain vont l'obliger à tourner et il ne peut alors pas, compte tenu des contraintes liées au frottement, équilibrer ses forces. Il faudrait que les forces des contacts inférieurs soient en-dehors de leurs cônes de frottement de Coulomb. Ainsi, le choix aléatoire des valeurs p_1 et p_2 pour les deux grains de la couche supérieure peut être une source d'échec pour le grain courant (intervalle de possibilités vide pour p).

Ainsi, même si l'espace des solutions est *a priori* infini, l'espace des non-solutions est lui aussi infini. Le problème revient alors à trouver les solutions dans cet espace de recherche infini.

3.4.5 Les problématiques posées par ce modèle

Ce modèle, malgré son apparente simplicité, permet déjà de poser de nombreuses questions sur le comportement mécanique d'un empilement à l'équilibre statique. La géométrie de l'empilement étant fixe et parfaitement régulière, le frottement des grains est le seul paramètre de désordre modulable pour ce modèle.

Il est donc possible d'analyser comment évolue le désordre mécanique du réseau de forces avec le frottement. Ce paramètre est-il vraiment lié au désordre mécanique et dans quelles mesures ?

Un autre paramètre à étudier est la taille de l'empilement. En effet, le tas étant cylindrique, quel sera l'effet de ces conditions limites particulières avec une très haute colonne de grains ?

Le dernier paramètre du modèle est la gravité que subit l'empilement. Il est possible d'envisager un tas sans gravité, i.e. un tas posé sur une table à l'horizontale. La gravité n'entre alors pas en jeu dans le calcul de la charge des grains. Cet empilement peut être considéré sous confinement et il est alors possible d'étudier la réponse du système à un appui localisé. Cette expérimentation est connue sous le nom de la fonction de Green (cf. Chap. 7 pour une étude détaillée de cette expérimentation) et permet de caractériser la diffusion et la propagation du milieu. L'intérêt de se placer dans des conditions sans gravité est que le poids des grains n'entre pas en concurrence avec la surcharge due à l'appui sur le tas. La propagation du signal et sa diffusion peuvent ainsi être mesurés sur de longues distances dans le milieu.

3.5 L'algorithme MC-Granu

Clément et Eloy [Eloy et Clément 1997] ont proposé un algorithme de simulation numérique qui utilise une méthode stochastique de résolution des équations exactes de l'équilibre statique d'un tas. Elle se rapproche des méthodes dites de Monte-Carlo qui font intervenir un tirage aléatoire de variables. Nous nommerons cet algorithme MC-Granu dans la suite de notre exposé. La résolution d'un tas s'effectue de haut en bas, par échantillonnage aléatoire des couches. Nous allons d'abord montrer comment l'équilibre des grains est atteint et ensuite expliciter l'algorithme de façon globale (cf. Algorithme 3.1).

Algorithme 3.1 MC-Granu

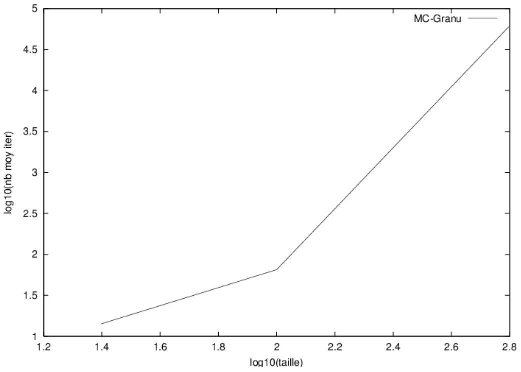
nbEchecs $\leftarrow 0$
Tant Que la dernière couche n'a pas été atteinte **Faire**
Echantillonnage : Créer un échantillon de la couche courante, de taille N, en choisissant aléatoirement le paramètre p pour tous les grains de chaque couche.
Regard en avant : Eliminer de l'échantillon les couches qui génèrent des échecs à la couche suivante.
Si l'échantillon de couches est vide **Alors**
nbEchecs $\leftarrow$ nbEchecs $+1$
Effectuer un retour arrière de $2^{\mathrm{nbEchecs}-1}$ couches
Sinon
Choisir aléatoirement une couche dans l'échantillon et propager les valeurs de cette couche à la couche suivante.
Fin Si
Si la couche courante n'a pas encore été atteinte **Alors**
nbEchecs $\leftarrow 0$
Fin Si
Fin Tant Que

Résolution de l'équilibre d'un grain. En résolvant le tas de haut en bas, nous considérons que les valeurs de forces supérieures du grain (cf. Fig. 3.3 : Q1, S1, Q2 et S2) sont fixées. Il nous faut alors résoudre un système de trois équations à quatre inconnues (q1, s1, q2 et s2). MC-Granu est un algorithme stochastique : chaque grain fait un tirage aléatoire de son paramètre p, i.e. il choisit la façon de répartir la charge qu'il supporte aux grains de la couche inférieure. Mais, il arrive souvent que l'équilibre d'un grain ne puisse être atteint aux vues des forces des contacts supérieurs qui lui sont imposées, il n'y a alors pas de solution possible, l'intervalle des valeurs possibles pour p est donc vide.

Résolution du tas. MC-Granu génère donc, à partir de la couche courante, un échantillon de taille N de cette même couche où le paramètre p de chacun des grains est choisi aléatoirement. Ensuite, MC-Granu procède à un *regard en avant* qui lui permet d'éliminer de l'échantillon les couches qui vont créer des échecs pour au moins un des grains de la couche suivante. Malheureusement, il arrive souvent qu'il n'y ait plus de couches ne générant pas d'échec à la couche suivante parmi celles échantillonnées. Dans ce cas, l'algorithme MC-Granu procède à un retour-arrière proportionnel au nombre d'échecs rencontrés ($2^{\mathrm{nbEchecs}-1}$). Ainsi, le tas peut être reconstruit entièrement des centaines de fois, avant d'avoir la «chance» de générer un tas avec peu d'échecs.

En pratique, cet algorithme est d'une complexité exponentielle avec la largeur du tas (cf. Tab. 3.2). En effet, plus le tas est large, plus la probabilité est grande que deux grains voisins, dont les paramètres p sont choisis aléatoirement, génèrent une impossibilité pour leur fils commun. Ce phénomène est encore décuplé pour les hautes valeurs de frottement car les grains ont alors plus de chance de choisir des forces excentrées par rapport à la direction normale du contact. Et dans ce cas, la couche toute entière est refusée et supprimée de l'échantillon des couches candidates.

Taille	Temps (secs)	Nb d'iter	Nb moy d'iter
25 (5^2)	0,07	355	14,21
100 (10^2)	0,30	6515	65,13
625 (25^2)	277,14	37,4 10^6	59 776



TAB. 3.2 – Performances de l'algorithme MC-GRANU pour un cœfficient de frottement $\mu = 0.5$

Rachid Bouamrane [Bouamrane 2001] a proposé une variante de cet algorithme en considérant qu'il était inutile de constituer un échantillon de taille N de la même couche pour ensuite choisir aléatoirement une couche parmi celles qui ne génèrent pas d'erreurs à la couche suivante. Il propose donc de générer et de tester aussitôt les couches jusqu'à trouver une couche qui réussisse le test. Le gain, en termes de performances est conséquent, car s'il est difficile de trouver une couche qui ne conduise pas à un échec à la couche suivante, il est N fois plus long d'en trouver N. Cependant, pour les même raisons que MC-GRANU, la complexité, en pratique, de cet algorithme croît de manière exponentielle avec la valeur de frottement.

3.6 Conclusion

Dans un premier temps, nous avons montré les difficultés théoriques que rencontrent les chercheurs en physique dans le choix d'un modèle pour représenter l'équilibre local d'un grain.

Nous avons ensuite décrit brièvement les principaux algorithmes utilisés en physique des milieux granulaires pour simuler des empilements de grains. Ces algorithmes se montrent souvent inefficaces pour l'étude statistique d'empilements à l'équilibre statique et donc du réseau de forces sous-jacent. En effet, les calculs sont souvent trop coûteux pour aborder une étude statistique du modèle et dans certains cas, la méthode elle-même est inadaptée à l'étude d'empilements à l'équilibre statique.

Des physiciens chercheurs du Laboratoire des Milieux Désordonnés et Hétérogènes ont donc proposé un modèle simple de l'équilibre statique d'un empilement de sphères dures et régulières, dont les solutions peuvent être calculées par simulation numérique. L'algorithme qu'ils ont conçu pour ce modèle est malheureusement d'une complexité exponentielle, en pratique, avec la largeur du tas.

Dans le chapitre suivant, nous décrivons GranuSolve : un algorithme original basé sur une approche multi-agent pour la résolution de tas de sable à l'équilibre statique. Cet algorithme se fonde sur le modèle de Clément et d'Eloy pour l'équilibre des grains et du tas dans son ensemble.

Chapitre 4

GranuSolve : Une approche Multi-Agent pour la «résolution de tas de sable»

Sommaire

4.1 Introduction

De nombreuses applications ont été mises en œuvre dans le domaine de la robotique collective en utilisant des systèmes multi-agents (SMA) [Drogoul et Zucker 1998]. Cependant, les systèmes multi-agents sont aussi étudiés dans le cadre de simulations de systèmes complexes [Servat *et al.* 1998b, Breton *et al.* 2000, Rouchier *et al.* 2000]. Nous proposons une approche multi-agent pour résoudre des tas de sable à l'équilibre statique.

Dans ce chapitre, nous considérons un grain comme un agent essentiellement réactif dont le but est de trouver son équilibre en fonction de ses voisins. Un empilement est donc à l'équilibre statique lorsque tous les agents-grains qui le composent sont eux-mêmes à l'équilibre.

Nous présentons la méthode d'ECO-RÉSOLUTION d'un tas que nous utilisons [Ferber 1990, Drogoul *et al.* 1991, Drogoul et Dubreuil 1991]. Nous détaillons ensuite, à l'aide de la méthodologie de conception de systèmes multi-agents CASSIOPÉE, le modèle d'un agent-grain [Collinot et Drogoul 1996].

Les différentes versions de notre solveur sont alors données pour illustrer et justifier l'ajout de connaissances du domaine. Ces connaissances sont représentées sous la forme d'heuristiques pour l'ECO-RÉSOLUTION du tas et de comportements de résolution pour les agents-grains. La validation, les performances et les limitations du système sont discutées pour chacune des versions implémentées.

Mais, tout d'abord, nous exposons les motivations qui nous ont conduit à utiliser les SMA pour la résolution de tas de sable à l'équilibre statique, selon le modèle de Clément et d'Eloy vu au chapitre 3.

4.2 Une modélisation distribuée

L'utilisation de systèmes multi-agents pour la simulation de processus physiques est récente. Dans une étude théorique sur l'apport des SMA pour la physique, [Treuil *et al.* 1997] introduisent la notion de «simulons» pour décrire des agents simulant des processus physiques. Les simulons ne sont pas les processus physiques eux-mêmes, mais ils correspondent au raisonnement mené par des physiciens pour réaliser ces processus. Ils ont dégagé trois caractéristiques essentielles d'un agent-simulon. Cet agent doit être capable d'effectuer des calculs locaux afin de distribuer la résolution globale du système entre les agents. Il doit aussi être autonome et pouvoir communiquer avec son environnement.

Cette modélisation a été adoptée par David Servat [Servat *et al.* 1998a]

[Servat *et al.* 1998b] pour l'étude du ruissellement de *boules d'eau* par simulation à l'aide d'un système multi-agent. Il modélise chaque boule d'eau par un agent autonome dans un environnement topographique donné. Chaque agent boule d'eau calcule sa trajectoire et les calculs sont ainsi répartis entre les agents et non centralisés dans un immense système d'équations à résoudre. De plus, leurs capacités à communiquer avec leur voisinage leur permet d'initier la création de groupes d'agents. Lorsque des interactions stables et durables s'installent entre des agents individuels, ils vont alors former un nouveau groupe d'agents. Par exemple, des boules d'eau qui empruntent des trajectoires similaires se regroupent en une ravine. La formation de tels groupes n'est pas ordonnée de façon centralisée mais émerge des interactions locales entre les agents boules d'eau.

Pour le problème qui nous intéresse, considérer un grain comme un agent permet de décentraliser la résolution du tas localement à chaque grain. En effet, l'espace de recherche des solutions pour un tas est beaucoup trop grand pour être appréhendé de façon globale, alors que, localement, les possibilités qui s'offrent à chaque grain sont données par ses équations. En outre, nos agents-grains sont suffisamment autonomes pour pouvoir introduire de la dynamique dans le système (avalanches, écoulement de silos, etc.) ou d'autres modifications du modèle. Enfin, leurs capacités à communiquer nous ont permis de définir, en collaboration avec les physiciens de l'équipe, des comportements simples de résolution.

Nous allons maintenant montrer comment résoudre un tas de sable à l'équilibre statique. Notre algorithme GRANUSOLVE se base sur deux méthodes de conception de systèmes multi-agents.

La première est la méthode d'ECO-RÉSOLUTION développée par Alexis Drogoul et Jacques Ferber [Drogoul *et al.* 1991, Drogoul et Dubreuil 1991, Drogoul 1993]. Cette méthode est générique car elle permet de dégager les heuristiques issues du domaine pour la résolution distribuée du problème. Cette méthode de résolution est présentée dans la section suivante.

La seconde est la méthodologie de conception de systèmes multi-agents CASSIOPÉE [Collinot et Drogoul 1996] à l'aide de laquelle nous avons implémenté nos agents-grains.

Enfin, les diverses versions de GRANUSOLVE sont présentées afin d'illustrer les choix successifs des heuristiques de résolution (cf. Tab. 4.1) et les évolutions du modèle de l'agent-grain. La validation et les performances de notre algorithme sont discutées pour chacune de versions implémentées.

4.3 Application de l'ECO-RÉSOLUTION aux tas de sable

La méthode d'ECO-RÉSOLUTION d'Alexis Drogoul et de Jacques Ferber est une méthode générique de résolution de problèmes distribués [Drogoul 1993]. La généricité de la méthode résulte de la distinction qui est faite entre les heuristiques concernant le domaine et la méthode de résolution distribuée. La résolution d'un empilement d'agents-grains se fait donc très naturellement (cf. Algorithme 4.1).

ALGORITHME **4.1** ECO-RÉSOLUTION d'un empilement

```
ListeAgentsActifs ← {grains de l'empilement}
Tant Que ListeAgentsActifs ≠ ∅ Faire
  grain = ChoixHeuristiqueAgentPrioritaire(ListeAgentsActifs)
  ListeAgentsActifs ← ListeAgentsActifs - {grain}
  grain.Resoudre()
Fin Tant Que
```

A l'initialisation de la résolution, tous les agents-grains sont non-résolus et demandent «à avoir la main» pour se résoudre. Ils sont marqués comme étant actifs. Notre algorithme d'ECO-RÉSOLUTION itère donc le choix et la résolution d'un grain jusqu'à ce que tous les grains soient résolus, i.e. qu'il ne reste plus de grains actifs. La fonction *ChoixHeuristiqueAgentPrioritaire()* est la seule qui dépende de notre domaine d'application : les milieux granulaires. Nous avons discuté avec les physiciens de l'équipe des heuristiques possibles dans le but d'obtenir un algorithme d'ECO-RÉSOLUTION qui ait du sens pour eux, à l'inverse d'un algorithme de type «boîte noire». L'interface de visualisation de la résolution d'un tas devait permettre à un expert de comprendre la résolution des grains et du tas dans son ensemble. Pour notre problème, nous avons donc proposé plusieurs heuristiques pour le choix du grain prioritaire (cf. Tab. 4.1).

Toutes ces heuristiques sont parvenues à résoudre des empilements de grains à l'équilibre statique. Mais, les heuristiques qui résolvent le tas, couche par couche, de haut en bas se sont révélées beaucoup plus efficaces que les autres. Par exemple, l'heuristique ALÉATOIRE_TAS est d'une complexité, en pratique, quadratique avec le nombre de grains de l'empilement. Celles qui résolvent le tas couche par couche ont un comportement linéaire avec le nombre de grains, pour une valeur de frottement fixée.

Il semble donc que la gravité soit une direction de résolution privilégiée pour le tas. Cela peut s'expliquer du fait que l'information des conditions limites supérieures doit être transmise aux grains des couches inférieures avant de résoudre plus avant le tas.

Nous venons d'expliciter la résolution du tas à l'aide de la méthode d'ECO-RÉSOLUTION. La section suivante décrit le modèle d'un agent-grain et explique comment la résolution

Heuristique	**Description**
Pression_Max	L'heuristique Pression_Max choisit prioritairement le grain qui subit le plus de pression de ses voisins. Ainsi, nous espérons nous rapprocher d'un état-solution en diminuant la distance qui sépare chaque grain d'un état résolu.
Pression_Min	Il s'agit de l'heuristique inverse de Pression_Max. Aussi surprenant que cela puisse paraître, cette heuristique fonctionne aussi bien car nous choisissons en priorité les grains qui devraient se résoudre facilement.
Aléatoire_Tas	Ici, il s'agit de prendre un grain aléatoirement dans le tas, parmi ceux qui ne sont pas résolus. Cette heuristique est assez mauvaise car un grain est résolu alors que ses voisins ne le sont pas encore. Lorsque ses voisins se résolvent à leur tour, ils vont forcément demander au premier de se résoudre à nouveau, car leurs valeurs de forces de contact auront changé.
Aléatoire_ Couche	Cette heuristique permet de résoudre le tas de haut en bas, couche par couche. Ensuite, sur la couche courante, un grain est choisi aléatoirement. La résolution se fait donc «en pluie».
Lecture_ Gauche	Le choix du grain prioritaire est le même que le sens de lecture à l'occidentale : de haut en bas et de gauche à droite.
Lecture_Droite	Toujours de haut en bas, mais de droite à gauche.
Direction_ Aléat_Couche	La direction de lecture sur une couche est fixée aléatoirement.
Direction_ Aléat_Fixée	La direction de parcours d'une couche est choisie une fois pour toute à l'initialisation de la résolution. Toutes les couches se résolvent donc en parcourant les grains dans la même direction.
Serpentin_ Gauche	Ici, la résolution se fait de haut en bas mais successivement de gauche à droite, puis de droite à gauche. Le sens du parcours des couches s'inverse à chaque couche.
Serpentin_ Droit	Idem Serpentin_Gauche à l'envers.
Démarrage_ Et_Dir_Aléat	Le premier grain choisi sur la couche et la direction de parcours sont aléatoires pour chacune des couches.
Bords_Externes	La résolution se fait toujours de haut en bas mais, sur la couche courante, les grains qui se trouvent aux bords d'un groupe de grains non résolus sont prioritaires. Cette heuristique tente ainsi de résorber des groupes de grains non résolus.

Tab. 4.1 – Heuristiques pour l'algorithme d'Eco-Résolution

locale des conflits s'effectue entre grains.

4.4 Application de la méthodologie de conception CASSIOPÉE aux agents-grains

Du fait de la nature «inerte» d'un grain réel, la modélisation du grain par un agent essentiellement réactif est naturelle. Nous nous sommes appuyés sur la méthodologie de conception de SMA CASSIOPÉE [Collinot et Drogoul 1996, Drogoul et Zucker 1998] pour concevoir l'agent-grain.

Cette méthodologie permet de concevoir les agents d'un système multi-agent pour que globalement, le SMA calcule la fonction f recherchée. Le principe qui guide cette méthodologie est une conception progressive des agents, par couches de plus en plus complexes. Cette façon de procéder est appelée «bottom-up», par opposition au mode de programmation classique «top-down» où le problème à résoudre est décomposé en sous-problèmes plus simples jusqu'à un niveau de détails suffisamment fin pour pouvoir l'implémenter directement.

Ainsi, cette méthodologie s'applique parfaitement à notre problème car si un modèle de l'équilibre du grain est connu depuis Newton, le modèle de l'empilement, lui, est inconnu et c'est justement ce que nous cherchons à découvrir. L'idée est donc de résoudre le problème de l'équilibre statique d'un empilement localement à partir des contraintes entre les grains.

L'autre avantage de la méthodologie CASSIOPÉE est qu'un système peut être conçu selon le principe de *parcimonie* [Drogoul 2000]. Ce principe nous permet d'arrêter de complexifier le modèle d'un agent lorsque le système multi-agent calcule effectivement la fonction globale f recherchée. Ainsi, le modèle de l'agent obtenu est le modèle minimal qui permet de calculer la fonction globale f qui permet de résoudre le problème, parmi l'ensemble des modèles de cette classe.

La méthodologie CASSIOPÉE est basée sur la définition incrémentale d'un agent en rôles et niveaux successifs que nous allons maintenant détailler. La première entité à définir est l'environnement dans lequel sont placés nos agents-grains.

4.4.1 L'environnement

L'environnement, qui a généralement un rôle important dans les systèmes multi-agents, est ici réduit à son expression la plus simple, à savoir la définition d'un cadre rectangulaire dans lequel l'empilement est placé. Le champ gravitique, auquel cet empilement est soumis à l'intérieur de ce cadre, est également défini.

4.4.2 Le rôle individuel de l'agent-grain

Un grain, même seul, est déjà une entité physique à part entière. Il possède des caractéristiques physiques comme sa masse, son rayon et sa position dans l'environnement qui sont modélisées à ce niveau de description (cf. Fig. 4.1).

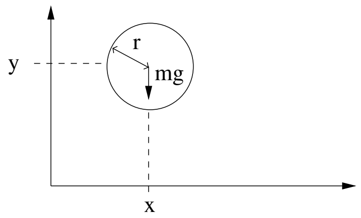


Fig. 4.1 – Modèle d'un agent-grain individuel
Un agent-grain placé seul dans son environnement. Il ne possède pas de voisin à ce niveau de conception.

Notre travail portant sur l'étude d'empilements à l'équilibre statique, un grain n'aura donc aucun but particulier. Puisqu'il connaît par avance sa position, un grain n'a pas à «tomber» dans l'environnement jusqu'à trouver une position d'équilibre.

4.4.3 Le niveau topologique de l'agent-grain

Au niveau topologique, les contraintes physiques liées à la géométrie de l'empilement sont insérées dans le modèle. Un agent-grain topologique peut donc être considéré comme étant un agent-grain individuel «augmenté» de la connaissance de son voisinage (cf. Fig. 4.2).

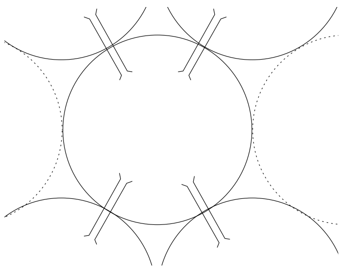


Fig. 4.2 – Modèle d'un agent-grain topologique
L'agent-grain connaît maintenant ses voisins et peut communiquer avec eux.

De plus, nous ajoutons des comportements simples de résolution et de vérification des équations de l'équilibre statique ainsi que des contraintes de frottement et d'orientation en chaque point de contact. Grâce aux équations du modèle, connaissant les valeurs des forces des contacts supérieurs, nous pouvons en déduire immédiatement l'intervalle des valeurs possibles pour les forces inférieures (qui peut être, dans certains cas, vide).

A ce niveau, les primitives de communication entre grains, i.e. l'envoi et la réception de messages, sont également définies.

4.4.4 Le rôle social de l'agent-grain

Afin d'alléger les commentaires sur notre algorithme, nous parlerons de «parents» pour désigner les deux grains supérieurs, de «père» pour un des grains supérieur, «fils» pour un grain inférieur et «frère» pour un grain latéral, voisins du grain courant.

Le «rôle social» d'un agent-grain est de trouver son équilibre vis-à-vis de ses voisins. Le grain reçoit, par message, les valeurs de forces que les grains des contacts supérieurs ont calculé pour leur propre équilibre. Le grain vérifie alors que les contraintes qui lui sont appliquées sont compatibles avec son propre équilibre (comportement du niveau topologique). Si tel n'est pas le cas, le grain va choisir aléatoirement un de ses parents et lui demander de changer sa valeur de force. Et ainsi de suite jusqu'à obtenir de ses parents un jeu de forces tel que le grain puisse se résoudre à son tour. Il va alors choisir aléatoirement la valeur de son paramètre p et distribuer la charge qu'il reçoit des grains supérieurs aux grains inférieurs (cf. Fig. 4.3).

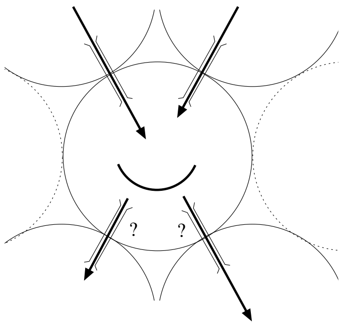


FIG. 4.3 – Modèle d'un agent-grain social
L'agent-grain implémente des comportements de résolution «sociaux» de son équilibre. Il reçoit les forces de ses parents et transmet les forces correspondant à son équilibre statique à ses fils.

Les différentes configurations possibles pour la résolution sociale d'un agent-grain sont représentées sur les schémas suivants.

Dans le premier cas d'une résolution sans échec (cf. Fig. 4.4), le grain reçoit, des grains supérieurs, un jeu de forces compatible avec son propre équilibre statique. Il a alors le choix, représenté par les cônes de frottement de Coulomb, de répartir la charge qu'il reçoit aux grains inférieurs. Il choisit donc, aléatoirement, une valeur pour son paramètre p et communique ces informations à ses voisins inférieurs.

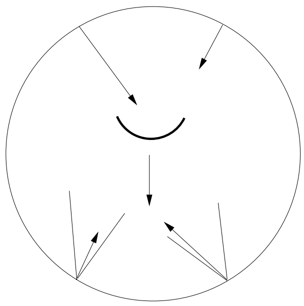


Fig. 4.4 – Le cas simple d'une résolution sans échec
Les forces des contacts supérieurs autorisent l'équilibre de l'agent-grain. Le degré de liberté pour les forces des contacts inférieurs est illustré par des cônes dont l'écartement dépend de la valeur de frottement (loi de Coulomb).

Dans le deuxième cas (cf. Fig. 4.5), le grain ne peut pas garantir l'équilibre statique aux vues des forces qui lui sont appliquées. Il va donc choisir aléatoirement un de ses parents et lui demander de «se pousser» en modifiant sa force pour ce contact. A cet instant, le grain se désactive et attend que son parent lui ait envoyé une force modifiée pour tenter de se résoudre à nouveau.

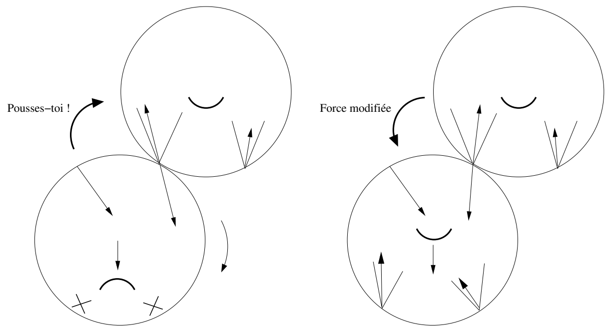


Fig. 4.5 – Le cas où le grain ne peut pas atteindre directement l'équilibre

Enfin, le dernier cas (cf. Fig. 4.6) est celui où l'un des fils du grain courant, incapable d'atteindre son équilibre, lui demande de changer sa force pour leur contact commun. Le grain courant pouvant se résoudre, il lui suffit de refaire un tirage aléatoire des forces inférieures et de communiquer son nouveau jeu de forces à ses fils. Des problèmes de convergence apparaissent alors (cf Fig 4.6).

Le fils droit étant déjà résolu, supposons que le fils gauche du grain courant lui ait demandé de changer sa force de contact (cf. Fig. 4.6, cas 1). En changeant son jeu de forces, le grain va modifier les forces de contact pour ses deux fils simultanément (cf. Fig.

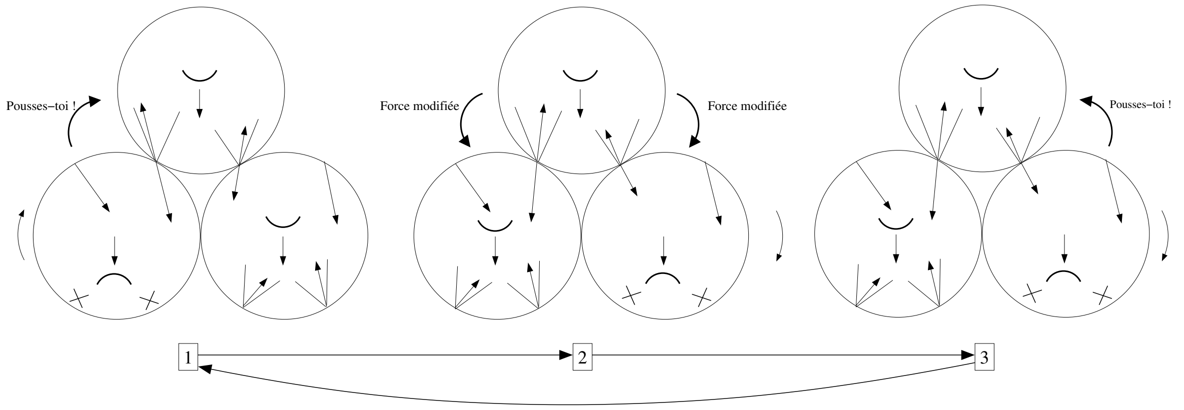


FIG. 4.6 – Le problème de l'inter-blocage

4.6, cas 2). Il arrive alors que le fils gauche soit satisfait de cette nouvelle force, mais qu'à l'inverse, le fils droit soit à son tour mis en échec. Le fils droit va alors peut-être pousser son père gauche et les grains se retrouvent alors dans le premier cas de figure (cf. Fig. 4.6, du cas 3 au cas 1). Il arrive parfois que le système boucle ainsi indéfiniment, les deux frères se renvoyant mutuellement la faute.

Afin d'éviter cette situation d'inter-blocage, nous avons donné aux grains la possibilité d'«arbitrer» ce genre de conflit. Nous maintenons, pour chaque grain, un compteur du nombre de fois consécutives où un autre agent-grain lui demande de changer ses forces des contacts inférieurs. A partir d'un certain seuil appelé *nb_poussées_max* (typiquement de l'ordre de 5), le grain décide de pousser à son tour ses propres parents pour tenter de débloquer la situation. Le compteur est alors remis à zéro. Le nombre maximum de poussées consécutives que peut supporter un grain devient alors un paramètre de notre algorithme.

4.4.5 Les groupes et le rôle collectif

Pour cette première version, nous n'avons pas eu besoin de modéliser des groupes car avec seulement les trois couches logicielles de notre modèle d'agent-grain décrites précédemment, l'algorithme arrive à résoudre les tas. D'après le principe de parcimonie, nous arrêtons donc de complexifier le modèle de nos agents puisque la fonction globale est atteinte par le système.

4.4.6 Le schéma des classes de GRANUSOLVE

La figure 4.7 montre le schéma des classes implémentées dans GRANUSOLVE. La classe d'un agent-grain est décomposée en plusieurs couches successives selon la méthodologie

Cassiopée.

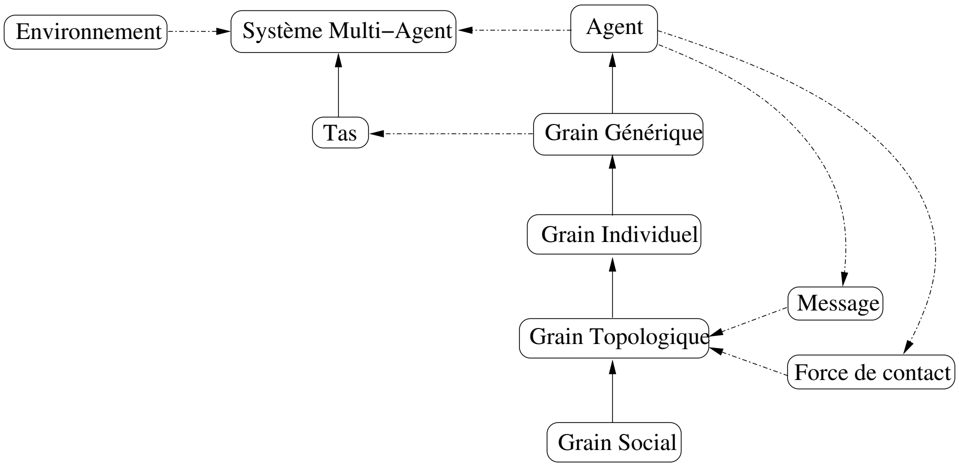


Fig. 4.7 – Schéma des classes de GranuSolve

4.5 La validation des solutions de l'algorithme

Afin de valider notre algorithme GranuSolve, nous avons définit une «batterie» de tests pour accepter ou non les solutions générées par notre algorithme.

1. *Equilibre d'un grain.* Pour chaque grain du tas, nous vérifions que la somme des forces qui s'appliquent sur ce grain est bien nulle, soit $\sum Q_i + mg = 0$ pour les composantes verticales des forces de contact et $\sum S_i = 0$ pour les composantes horizontales.
2. *Contraintes liées à la loi de frottement de Coulomb.* Nous vérifions également pour chacun des grains que les contraintes de frottement de Coulomb sont respectées en chaque point de contact, i.e. que $\forall_i, \frac{|N_i|}{|T_i|} < \mu$.
3. *Forces en compression.* Les forces normales de contact doivent être dirigées vers l'intérieur du grain et donc en chaque point de contact, nous avons : $\forall_i, N_i <= 0$.
4. *Loi de conservation de la charge verticale.* Le tas étant cylindrique, la charge verticale de la couche $ch\_v(N)$ est entièrement transmise à la couche $N+1$. Nous calculons donc la charge verticale de toutes les couches de grains pour laquelle nous devons avoir la relation suivante : $ch\_v(N+1) = ch\_v(N) + largeur \times mg$.
5. *Loi de conservation de la charge horizontale.* Toujours à cause de l'absence de parois, la charge horizontale d'une couche doit être nulle, sinon elle se déplacerait horizontalement, nous avons donc : $ch\_h(N) = 0$.

6. *Test qualitatif de la fonction de Green.* Ce dernier test est le plus précis d'entre tous. Le test de la fonction de Green considère la réponse d'un tas sur lequel un appui est exercé localement sur un grain de la première couche (cf. Chap. 7 pour une étude détaillée de cette expérimentation). En moyennant des milliers de réalisations pour ces conditions limites, l'œil de l'expert peut déceler des différences notables dans la répartition globale de cette surcharge. Ainsi, si un algorithme possède un biais implicite, cela peut se traduire par une dissymétrie de la réponse du système à cette surcharge. Par exemple, il peut être observé, en moyenne, une plus forte charge dans la moitié droite de l'empilement ou encore une dissymétrie dans la façon dont cette surcharge se propage et se diffuse dans le tas.

4.6 Heuristiques d'ECO-RÉSOLUTION d'un tas

Dans cette section, nous présentons plusieurs heuristiques pour l'algorithme d'ECO-RÉSOLUTION du tas.

4.6.1 LECTURE-GAUCHE

Résolution globale du tas

La résolution du tas se fait de façon distribuée, en «donnant la main» à chaque grain pour qu'il se résolve lui-même. Nous disposons donc, au niveau du tas, d'un ordonnanceur qui nous permet d'ordonner la résolution des agents. C'est à ce niveau que l'heuristique d'ECO-RÉSOLUTION est implémentée en donnant la priorité à certains agents.

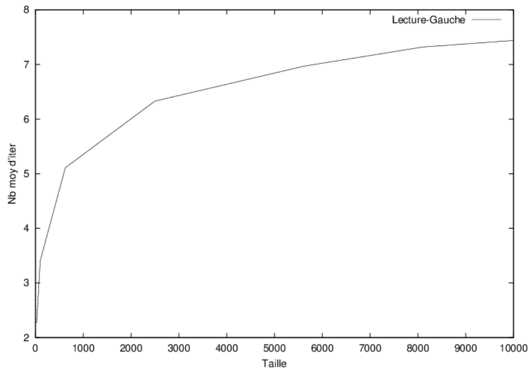
Nous avons essayé plusieurs heuristiques parmi celles que nous avons citées auparavant. Nous avons alors rapidement constaté que celles qui résolvaient le tas de haut en bas étaient les plus rapides. En effet, il ne sert à rien de résoudre un grain au milieu du tas tant que l'ordre de grandeur des forces qui vont lui être appliquées par ses voisins n'est pas connu. Dans ce cas, nous sommes obligés de résoudre chaque grain un grand nombre de fois car les grains voisins vont, à leur tour, choisir un jeu de forces et le transmettre au grain qui va devoir se résoudre à nouveau et ainsi de suite. La complexité, observée en pratique à frottement constant, de l'heuristique ALÉATOIRE_TAS est quadratique avec le nombre de grains de l'empilement.

Il semble donc que les contraintes doivent être propagées dans le sens de la gravité, de haut en bas. L'heuristique LECTURE_GAUCHE qui résout le tas de haut en bas et de gauche à droite (sens de lecture à l'occidentale) est, quant à elle, linéaire en pratique avec le nombre de grains. C'est donc cette heuristique que nous avons retenu pour cette première version de l'algorithme.

Performances

L'algorithme d'ECO-RÉSOLUTION LECTURE-GAUCHE est, en pratique, linéaire avec le nombre de grains mais à frottement constant. Ces performances sont déjà de plusieurs ordres de grandeurs meilleurs que l'algorithme MC-GRANU qui avait été proposé (cf Chap. 3.5).

Taille	Temps moy (secs)	Nb moy d'iter
25 (5^2)	0,01	2,27
100 (10^2)	0,10	3,41
625 (25^2)	0,91	5,11
2 500 (50^2)	5,03	6,33
5 625 (75^2)	11,46	6,97
8 100 (90^2)	17,61	7,32
10 000 (100^2)	22,29	7,44



TAB. 4.2 – Performances de l'algorithme LECTURE-GAUCHE
Tests réalisés pour une valeur de frottement $\mu = 0.5$ sur un Mac G4 400Mhz, sous LinuxPPC

Limitations

L'algorithme LECTURE-GAUCHE, comme MC-Granu et sa variante, reste exponentiel avec la valeur du frottement. La valeur de frottement maximale que nous pouvons atteindre pour des tas moyens (100x100) est de 0.8, ce qui correspond à un angle de décrochage de la brique sur le plan incliné de 38,66° (cf. chapitre précédent, Fig. 3.2).

Validation

Notre algorithme LECTURE-GAUCHE a passé avec succès tous les tests définis en 4.5 sauf le test qualitatif de la fonction de Green. Notre heuristique a révélé un biais implicite car elle favorisait la propagation de la charge vers la droite beaucoup plus souvent que vers la gauche.

Pour expliquer ce phénomène, il semble que l'hypothèse suivante soit la plus probable. La résolution se faisant de gauche à droite, nous favorisons systématiquement les grains non résolus les plus à gauche de la couche.

Nous pouvons décomposer une séquence type de résolution où le grain A (cf. Fig. 4.8) est soumis à une surcharge par rapport aux grains avoisinants.

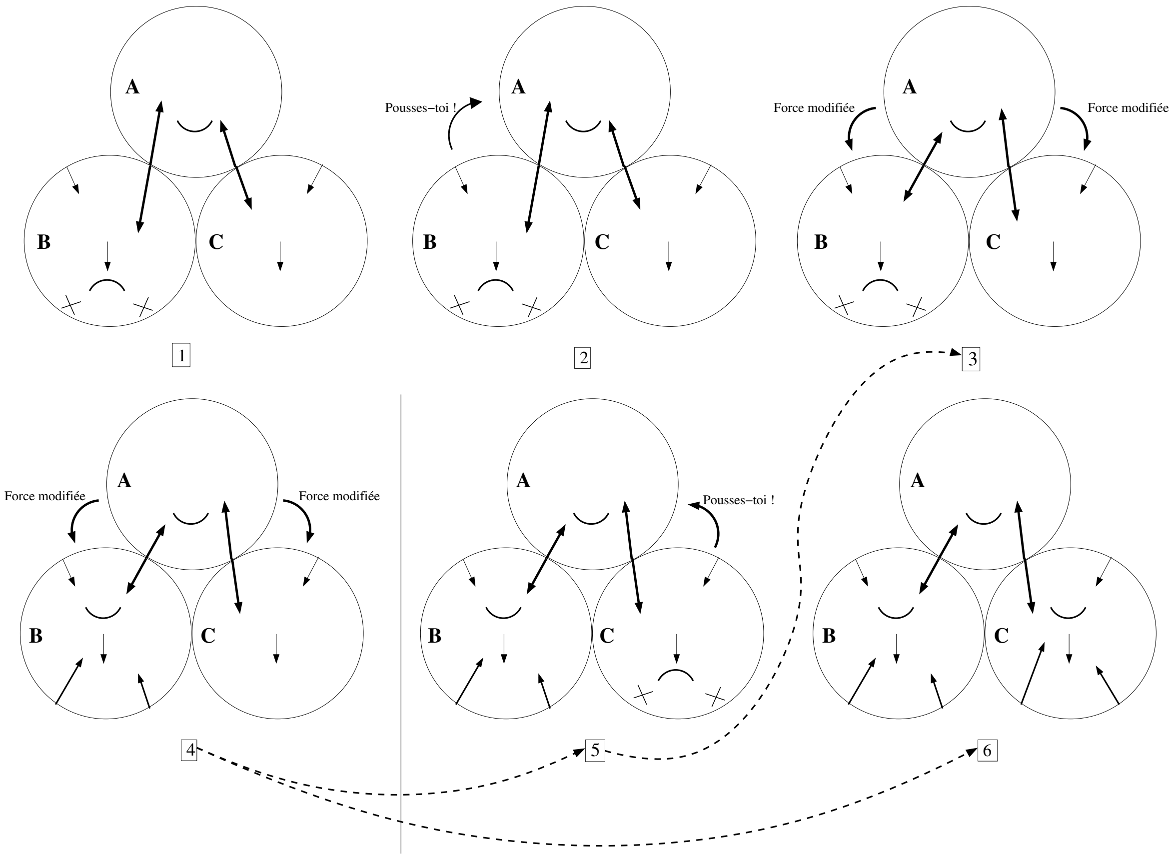


FIG. 4.8 – Un explication possible du biais de l'heuristique LECTURE-GAUCHE
Le grain B rejetant systématiquement la faute sur son frère C, le grain C a une grande probabilité de supporter une plus forte charge que B.

1. *B reçoit une trop forte charge de A.* Le grain A est résolu mais il envoie une forte charge au grain B qui ne peut se résoudre. Les grains de la couche supérieure étant tous résolus, c'est le grain B qui a la main.
2. *B pousse A.* Le grain B demande à son père droit (le grain A) de se pousser car il ne peut pas contrebalancer une telle surcharge.
3. *A modifie son jeu de forces.* A reçoit la demande de B et modifie donc aléatoirement la façon de répartir sa charge entre ses deux fils B et C. Ses deux fils étant modifiés, c'est le fils gauche qui aura le premier la main ensuite.
4. *B reçoit une nouvelle force de A.* Le grain B reçoit donc une nouvelle valeur de force de A. Si cette valeur ne lui permet toujours pas de trouver son équilibre, il va redemander à son père (A) de changer de force et donc recommencer les étapes

2 et 3 jusqu'à ce que la force de A lui permette de trouver son équilibre. A ce moment-là, il est très probable que la force de A soit beaucoup moins importante qu'auparavant puisqu'il n'arrivait pas à supporter une trop forte charge. Le grain B choisit aléatoirement un jeu de forces et le communique à ses fils.

5. *C a la main.* Le grain C a maintenant la main et peut se résoudre. Mais, comme son frère (B) a pu demander à leur père commun de changer et donc de diminuer sa force, le grain C se retrouve donc avec une charge plus importante que le grain B. S'il n'arrive pas à se résoudre, il va demander à son père (A) de changer son jeu de forces et il se retrouve alors à l'étape 3. Le processus va ainsi se répéter jusqu'à ce que le grain B soit résolu et qu'ensuite le grain C soit résolu.
6. *Le grain C est résolu.* Le grain C est résolu mais tout au long de ce processus, le grain B a toujours rejeté la faute sur son frère (C) et par conséquent, le grain C a une plus grande probabilité de supporter une plus forte charge que B. Bien sûr, tout ceci n'est valable que dans le cas où le grain A supporte lui-même une surcharge qui empêche l'un de ses fils de trouver facilement une solution, ce qui est le cas pour le test de la fonction de Green.

Cet exemple illustre bien la difficulté de la programmation distribuée par agents. En effet, un changement même minime dans le comportement d'un agent peut engendrer un comportement collectif très différent, voire la non convergence du système.

4.6.2 ALÉATOIRE-COUCHE

Pour remédier à ce biais, nous avons simplement changé d'heuristique pour l'ECO-RÉSOLUTION du tas, sans modifier le comportement des agents-grains. Nous avons alors adopté l'heuristique ALÉATOIRE-COUCHE qui considère les grains non résolus d'une même couche avec une priorité uniforme. Nous avons ainsi une résolution «en pluie» des grains de la couche courante.

Performances et limitations

Cette nouvelle heuristique a passé avec succès tous les tests, y compris celui de la fonction de Green. Les performances et limitations de cette heuristique sont les mêmes que précédemment, i.e. l'algorithme est linéaire avec le nombre de grains à frottement constant, mais exponentiel avec la valeur du frottement. Nous avons même une limite pour la valeur du frottement à 0.8, au-delà de laquelle l'algorithme ne converge plus.

4.6.3 COMMUNIQUE-INTERVALLES

En analysant les interactions entre les grains, les processus de résolution et de gestion de conflit, il nous est apparu que la communication entre les grains pouvait être enrichie. En effet, lorsqu'un grain demande à son père de se pousser, le père va alors modifier aveuglement son jeu de forces sans savoir si cette nouvelle force sera ou non acceptée par son fils. En plus d'être incertaine quant aux résultats, cette modification va entraîner la résolution du deuxième fils qui, à son tour, peut provoquer la résolution de son autre père, etc.

L'idée est donc qu'un père puisse communiquer à son fils, non plus uniquement la force de contact, mais aussi l'intervalle de ses possibilités pour cette force de contact. Ainsi, au lieu de demander à son père de changer aléatoirement sa force de contact, un grain pourra d'abord regarder s'il existe une force qui soit compatible avec son équilibre et demander ensuite à son père de changer sa force pour celle qu'il vient de calculer.

Performances et limitations

Le gain en nombre moyen de résolutions par grain est important. Le tableau (cf. Tab. 4.3) et la figure (cf. Fig. 4.9) illustrent les performances de cette nouvelle version de notre algorithme de résolution d'un agent-grain associée à l'heuristique d'ECO-RÉSOLUTION COMMUNIQUE-INTERVALLES.

Taille	Temps	NbIter/grain
10x10	0,014	1,20
25x25	0,120	1,36
50x50	0,571	1,44
75x75	1,426	1,49
100x100	2,702	1,51
150x150	6,727	1,54
200x200	12,881	1,56
250x250	21,555	1,57
300x300	33,136	1,58
350x350	47,513	1,59

Frott.	Temps	NbIter/grain
0,1	0,619	1,02
0,3	1,681	1,21
0,5	2,702	1,51
0,7	4,191	1,97
0,9	6,035	2,54
1,1	11,233	3,34
1,3	15,756	4,71
1,4	19,932	5,96
1,5	37,936	8,04
1,6	48,971	13,83

TAB. 4.3 – Performances de l'algorithme COMMUNIQUE-INTERVALLES
Tests réalisés pour a) une valeur de frottement $\mu = 0.5$ et b) une taille de tas fixe (100x100)

Ce changement minime du modèle nous permet, à présent, de résoudre des tas pour des valeurs de frottement jusqu'à 1.5 environ, ce qui représente un angle de décrochage de la brique sur le plan incliné de 56,3° (cf. chapitre précédent, Fig. 3.2). Cette valeur de rugosité est importante et notre algorithme permet de couvrir en grande partie l'espace

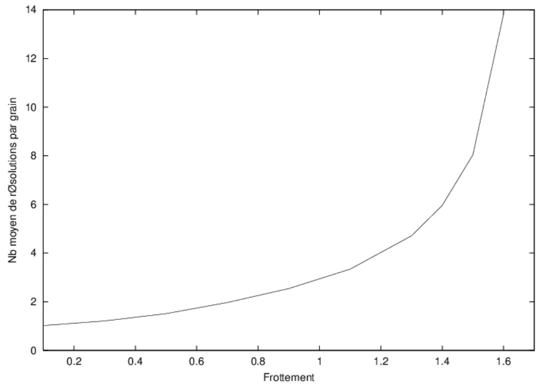


Fig. 4.9 – Courbe de performance de l'algorithme Communique-Intervalles
Tests réalisés pour une taille de tas fixe (100x100), la valeur de frottement variant.

des tas réalisables à partir du modèle. En effet, notre modèle n'a d'intérêt que jusqu'à une certaine valeur de frottement (1,7 soit un angle de 59,5°) car au-delà, le modèle ne donne plus de bornes supérieures pour les intervalles des forces de contact. En théorie, cela signifie que nous pouvons appliquer des forces infiniment grandes et qu'il sera toujours possible de les équilibrer, ce qui n'a pas beaucoup de sens dans le monde réel.

4.6.4 Bords-Externes

Toujours en visualisant la résolution des tas, nous nous sommes aperçus d'un phénomène d'étalement en largeur de la résolution. En effet, nous avons limité les remontées en hauteur avec le paramètre *nb_poussées_max* de chaque grain mais, pour les fortes valeurs de frottement, la probabilité d'échec des grains est forte et la résolution s'étale alors en largeur.

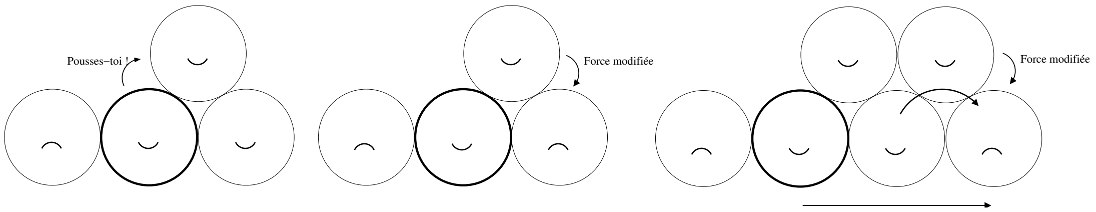


Fig. 4.10 – Etalement en largeur de la résolution avec l'heuristique Aléatoire-Couche

Par exemple, prenons la situation (cf. Fig. 4.10) où le grain courant et son frère gauche sont en échec et le frère droit résolu. Lorsqu'un grain est en échec, il choisit aléatoirement un de ses parents et lui demande de changer sa valeur de force pour leur contact en commun. Si le grain courant choisit de pousser son parent droit cela peut entraîner un échec pour son frère droit qui était résolu et qui va à son tour pousser un de parents et peut-être générer d'autres échecs. Il aurait mieux fallu pousser son parent gauche (si cela était possible) afin de ne pas potentiellement diminuer le nombre de grains résolus dans le système. C'est le principe même des heuristiques d'essayer d'atteindre, si possible de façon strictement monotone, un état qui se rapproche de l'état final recherché.

Nous avons donc proposé l'heuristique d'ECO-RÉSOLUTION BORDS-EXTERNES qui donne la priorité aux grains qui appartiennent à la couche non résolue la plus haute dans le tas et parmi ceux-ci aux grains qui se trouvent aux bords de groupes de grains non résolus. Nous donnons alors la main à de tels grains avec un paramètre supplémentaire qui est la direction privilégiée du parent à pousser si jamais il est en échec.

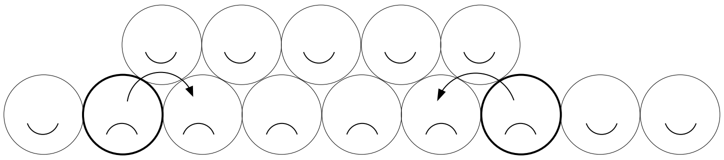


FIG. 4.11 – Le principe de l'heuristique BORDS-EXTERNES

Cette heuristique permet de remédier au problème d'étalement de la résolution car elle permet de résorber des groupes de grains non résolus. Ainsi, même si la résolution d'un grain est difficile, tant que l'algorithme n'a pas résolu le grain qui se trouve au bord d'un groupe de grains non résolus, il ne résout pas inutilement d'autres grains voisins (cf. Fig. 4.11).

Validation

Cette nouvelle version de notre solveur a passé avec succès tous les tests de validité que nous avons définis. De plus, cette heuristique ne semble pas introduire de biais dans la résolution car le test de la fonction de Green est probant.

Performances

Les performances sont encore accrues avec cette nouvelle heuristique (cf. Tab. 4.4, Fig. 4.12).

Limitations

Les limitations sont maintenant celles du modèle. En effet, nous arrivons à résoudre des tas pour des valeurs de frottement jusqu'à 1.7 environ (la limite de notre modèle étant fixée à 1.732) en des temps raisonnables.

Taille	Temps	Iter/grain
100 (10^2)	0,010	1,15
625 (25^2)	0,087	1,25
2 500 (50^2)	0,482	1,34
5 625 (75^2)	1,249	1,39
10 000 (100^2)	2,441	1,42
22 500 (150^2)	6,467	1,48
40 000 (200^2)	12,736	1,51
62 500 (250^2)	21,629	1,54
90 000 (300^2)	33,392	1,56
122 500 (350^2)	48,511	1,58

Frott.	Temps	Iter/grain
0,1	0,593	1,02
0,3	1,446	1,17
0,5	2,441	1,42
0,7	4,466	1,89
0,9	7,422	2,49
1,1	14,935	3,19
1,3	21,632	4,22
1,4	25,824	4,97
1,5	44,583	6,36
1,6	63,717	9,06
1,7	185,427	24,23

TAB. 4.4 – Performances de l'algorithme BORDS-EXTERNES
Tests réalisés pour a) une valeur de frottement $\mu = 0.5$ et b) une taille de tas fixe (100x100)

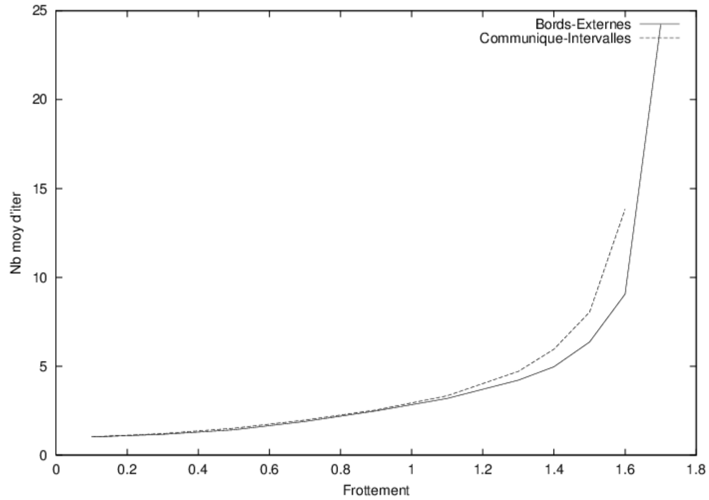


FIG. 4.12 – Courbe de performance de l'algorithme BORDS-EXTERNES
Tests réalisés pour une taille de tas fixe (100x100), la valeur de frottement variant.

4.7 Des agents et des agents-groupes

Dans cette section, nous allons nous intéresser à la conception de groupes dans notre modèle d'agent-grain. Il s'agit, en fait de l'extension de notre modèle initial d'un agent-grain aux groupes comme définis dans la méthodologie CASSIOPÉE.

Tout d'abord, les niveaux individuel, topologique et social du modèle de l'agent-grain restent inchangés. Seule la définition d'un agent change, i.e. un agent peut être de type simple ou de type groupe.

4.7.1 Définition des agents-groupes

Un groupe est aussi un agent, nous parlerons donc d'agent-groupe. Cette définition récursive permet d'avoir des agents de granularité différentes dans le système.

Nous avons défini deux types de groupes :

L'agent-couche est un agent-groupe contenant les grains d'une même couche horizontale. Une couche est résolue lorsque tous les grains qui la composent sont eux-mêmes résolus. Le comportement de résolution d'une couche est alors de donner la priorité aux grains qui se trouvent aux bords d'un groupe de grains non résolus. Lorsqu'un grain pousse un de ses parents, il arrête la résolution pour son groupe parent (i.e. la couche), considérant que les grains de la couche supérieure sont prioritaires.

L'agent-tas est un agent-groupe contenant les couches du tas. Le tas est résolu lorsque toutes les couches qui le composent sont résolues. Ainsi, le comportement de résolution d'un tas est de donner une priorité aux couches afin qu'elles se résolvent. L'heuristique utilisée est celle de la couche la plus haute.

En décomposant ainsi le problème de la résolution d'un tas composé de grains en la résolution d'un tas composé de couches, elles-mêmes composées de grains, nous définissons plus clairement les heuristiques de résolution pour chacun des niveaux.

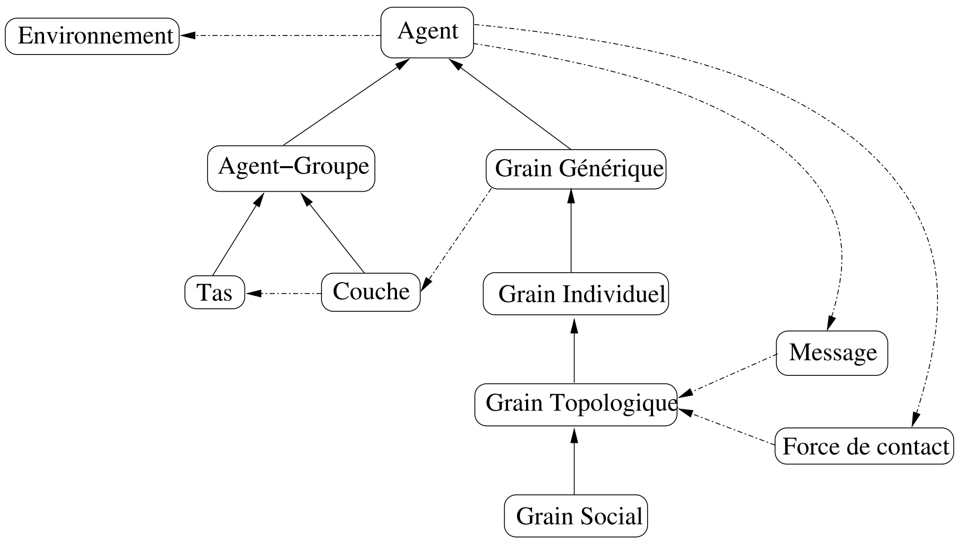


FIG. 4.13 – Schéma des classes des agents simples et des agents-groupes

Cette approche permet d'unifier le concept d'agent employé. Nous n'avons plus l'entité abstraite «tas» du premier modèle qui comprenait un ordonnanceur complexe du fait de la mise en œuvre de l'heuristique BORDS-EXTERNES. En effet, la méthode *play()* d'un agent permet à la fois de résoudre un grain mais aussi de résoudre un agent-groupe. Ainsi, cette méthode est aussi celle qui résout une couche de grains et un empilement de couches. Pour les agents-groupes, la méthode *play()* utilise une méthode *heuristique()* pour le choix du prochain agent prioritaire à résoudre. Il suffit ensuite de redéfinir uniquement la méthode *heuristique()* pour chacun des groupes «tas» et «couche». La factorisation du code des agents est ainsi beaucoup plus importante et permet de développer et de maintenir plus facilement le système.

Chaque niveau prend en charge sa résolution et met en œuvre l'heuristique appropriée. C'est une façon de distribuer encore un peu plus la résolution en supprimant le contrôle global des agents.

4.7.2 Le rôle collectif

Le rôle collectif d'un agent est assez simple. Lorsqu'un agent devient non résolu, il en informe son groupe parent et rend la main. De la même façon, lorsqu'un agent est activé par un autre (réception d'un message, par exemple), il active également son groupe parent afin que celui-ci puisse résoudre les agents qui le composent.

Performances et imitations

Cette version remaniée de l'algorithme précédent BORDS-EXTERNES a passé avec succès toutes les étapes du processus de validation. Elle n'a pas modifié les performances car nous avons simplement modifié la conception du modèle d'agent en introduisant des groupes dans notre simulation. L'heuristique BORDS-EXTERNES est toujours employée mais nous l'avons décomposée en deux heuristiques : COUCHE-HAUT-BAS pour l'agent-groupe *tas* et BORDS-EXTERNES pour l'agent-groupe *couche*.

4.7.3 FENÊTRE-GLISSANTE

L'introduction de groupes dans notre modèle, en plus de le simplifier, nous a permis d'implémenter l'algorithme FENÊTRE-GLISSANTE pour la résolution du tas. Nous avons constaté empiriquement lors des précédentes expérimentations que la résolution ne «remontait» jamais plus de quelques couches (5 maximum) dans le tas. Nous avons donc implémenté la résolution du tas par groupes de couches successives, permettant ainsi de résoudre des tas dont la taille dépasse largement celle de la mémoire vive de l'ordinateur. Des tas d'une taille encore jamais atteinte (plusieurs millions de grains) ont ainsi pu être résolus grâce à l'ajout d'un nouveau groupe d'agents : la «fenêtre de couches». Le tas résout alors des parties entières du tas représentées par des groupes de couches qui sont allouées et sauvegardées sur disque au fur et à mesure de la résolution.

Validation et performances

Cette version de l'algorithme ne modifie en rien les heuristiques précédemment employées et a passé avec succès tous les tests de validation.

Les performances, en termes de temps de calcul, sont les mêmes qu'auparavant. Par contre, la taille des tas que nous pouvons résoudre avec cet algorithme est colossale.

4.8 Validation qualitative

La première validation que nous nous étions imposée a permis de valider «mathématiquement» et assez finement les solutions générées par nos algorithmes.

Les lignes de forces

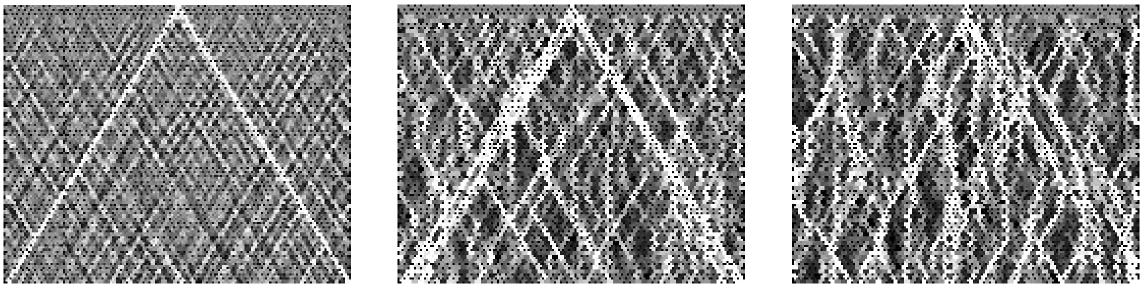


FIG. 4.14 – Visualisation des lignes de forces dans les tas simulés avec une valeur de frottement a) 0.1 b) 0.5 et c) 1.7

Après s'être assurés que les solutions étaient mathématiquement correctes, nous nous sommes intéressés à savoir si les experts retrouvaient visuellement des structures connues comme les réseaux de lignes de forces. Effectivement, des lignes de forces apparaissent dans le tas et le désordre mécanique augmente avec le frottement (cf. Fig 4.14). De plus, à partir d'une certaine valeur de frottement (0.5 typiquement), apparaissent des structures mécaniques intéressantes : les *tortillons*. Ce sont des lignes de forces qui collectent la charge verticalement. Ces structures, du point de vue de la propagation des contraintes mécaniques, vont jouer un rôle important en augmentant le caractère diffusif du milieu à forte valeur de frottement.

De plus, un autre élément important pour l'acceptation de notre algorithme par les experts, est lié au fait qu'ils comprennent ce que font les grains. A l'inverse d'un algorithme du type «boîte noire», nous avons pu facilement discuter de l'algorithme avec les experts et des heuristiques à mettre en œuvre pour résoudre les tas. La méthodologie de conception multi-agent CASSIOPÉE nous a permis de traduire des comportements physiques réels en comportements d'agents virtuels et de décomposer la conception de nos agents-grains en rôles et niveaux successifs plus simples à concevoir.

4.9 Conclusion

En conclusion, nous parlerons d'abord des performances de l'algorithme FenêtreGlissante qui permet de résoudre de très grands tas en des temps raisonnables (1 heure pour 1 million de grains à valeur de frottement $\mu = 1,4$).

Les performances

La taille des tas que nous pouvons résoudre est de quelques ordres de grandeurs supérieure à celle que peuvent résoudre les algorithmes classiques employés jusqu'à présent par les physiciens. De plus, les équations sont toutes résolues avec une précision de l'ordre de 10^{-6}, ce qui donne une confiance supplémentaire dans les solutions générées. Le tableau (4.5) donne une idée des temps de réalisation pour différentes combinaisons de tailles de tas et de valeurs de frottement.

Frottement Taille	0,1	0,3	0,5	0,7	0,9	1,1	1,3	1,5	1,7
2 500 (50^2)	0,20	0,23	0,30	0,40	0,52	0,75	1,76	5,08	28,84
5 625 (75^2)	0,48	0,55	0,72	0,99	1,31	1,95	4,42	13,57	68,83
10 000 (100^2)	0,90	1,05	1,38	1,95	2,57	3,86	8,81	26,41	125,1
40 000 (200^2)	4,32	5,17	7,03	10,32	13,77	21,08	45,62	137,1	581,4
90 000 (300^2)	11,51	14,03	19,32	28,68	38,63	59,53	121,5	366,8	1850
250 000 (500^2)	42,56	58,27	76,20	111,2	145,2	224,1	439,9	1230	5982
490 000 (700^2)	100,8	130,5	193,7	275,9	367,3	563,2	1048	2857	12684
1 E^6 (1000^2)	250,8	308,1	487,7	684,6	887,1	1402	2536	6964	31541

Tab. 4.5 – Tableau des temps de réalisation (secs) estimatifs pour différentes tailles de tas et valeurs de frottement
Moyennes réalisées pour 25 réalisations sur un G4-400Mhz sous LinuxPPC (env. 47 millions de grains ont été résolus pour réaliser ce tableau).

Grâce à ces performances en temps et en taille de tas, nous avons pu mener une étude statistique du modèle afin de dégager des comportements macroscopiques fiables des tas de sable (cf. Chap. 7). Ainsi, GranuSolve est le premier composant de notre environnement d'aide à la découverte scientifique appliqué aux milieux granulaires : GranuLab. Il était, en effet, nécessaire d'implémenter un solveur qui puisse résoudre des tas de grandes tailles en des temps relativement courts de façon à pouvoir générer des milliers de tas et obtenir une statistique fiable.

L'intelligence artificielle distribuée

D'un point de vue de l'Intelligence Artificielle Distribuée, GranuSolve représente une application de l'algorithme d'Eco-Résolution supportant un très grand nombre d'agents. L'idée de séparer la résolution des agents de la résolution du système multi-agent dans son ensemble (ou de groupes d'agents) nous a permis, en relation avec les physiciens de l'équipe, de mettre des connaissances du domaine dans notre système. De plus, l'interface de visualisation que nous avons implémentée (cf. Fig. 4.15) nous a beaucoup aidé à comprendre la résolution collective du système et à imaginer de nouvelles heuristiques pour l'Eco-Résolution.

La nature des solutions

Nous nous sommes aussi intéressés à la nature des solutions générées par notre solveur, à savoir s'il générait des familles «types» de solutions. Alexandre Pauchet, que j'ai eu le plaisir d'encadrer au cours d'un stage de DEA d'Intelligence Artificielle, Reconnaissance des Formes et Applications, a travaillé sur l'analyse des solutions de GranuSolve par classification automatique [Pauchet 2001]. L'idée de ce travail était d'essayer de caractériser l'espace des solutions atteintes par notre solveur, en vue d'une comparaison avec d'autres algorithmes.

Le nombre de descripteurs contenus dans un tas est beaucoup trop important pour pouvoir être analysés de façon automatique. Il était donc nécessaire de recourir à une description abstraite des tas, de manière à pouvoir dégager des zones homogènes significatives à l'aide d'un algorithme de classification efficace. La construction de cette abstraction se fait de manière automatique à partir des données brutes. Nous avons interrogé les physiciens de l'équipe pour déterminer les descripteurs qui leurs semblaient intéressants et caractérisant au mieux une solution donnée.

La classification des solutions de GranuSolve à partir de ces descripteurs de plus haut niveau a permis de dégager quelques propriétés des tas. Par exemple, il est possible de prédire la valeur de frottement μ pour une taille de tas fixée et inversement de prédire la taille d'un tas connaissant la valeur de frottement μ. Il est aussi ressorti que toutes les solutions pour des tas de taille 100x100 à valeur de frottement $\mu = 0,5$ sont homogènes et se distinguent nettement des solutions dont la valeur de frottement est distincte. Enfin, la valeur de frottement μ est le descripteur qui discrimine le mieux la structure mécanique

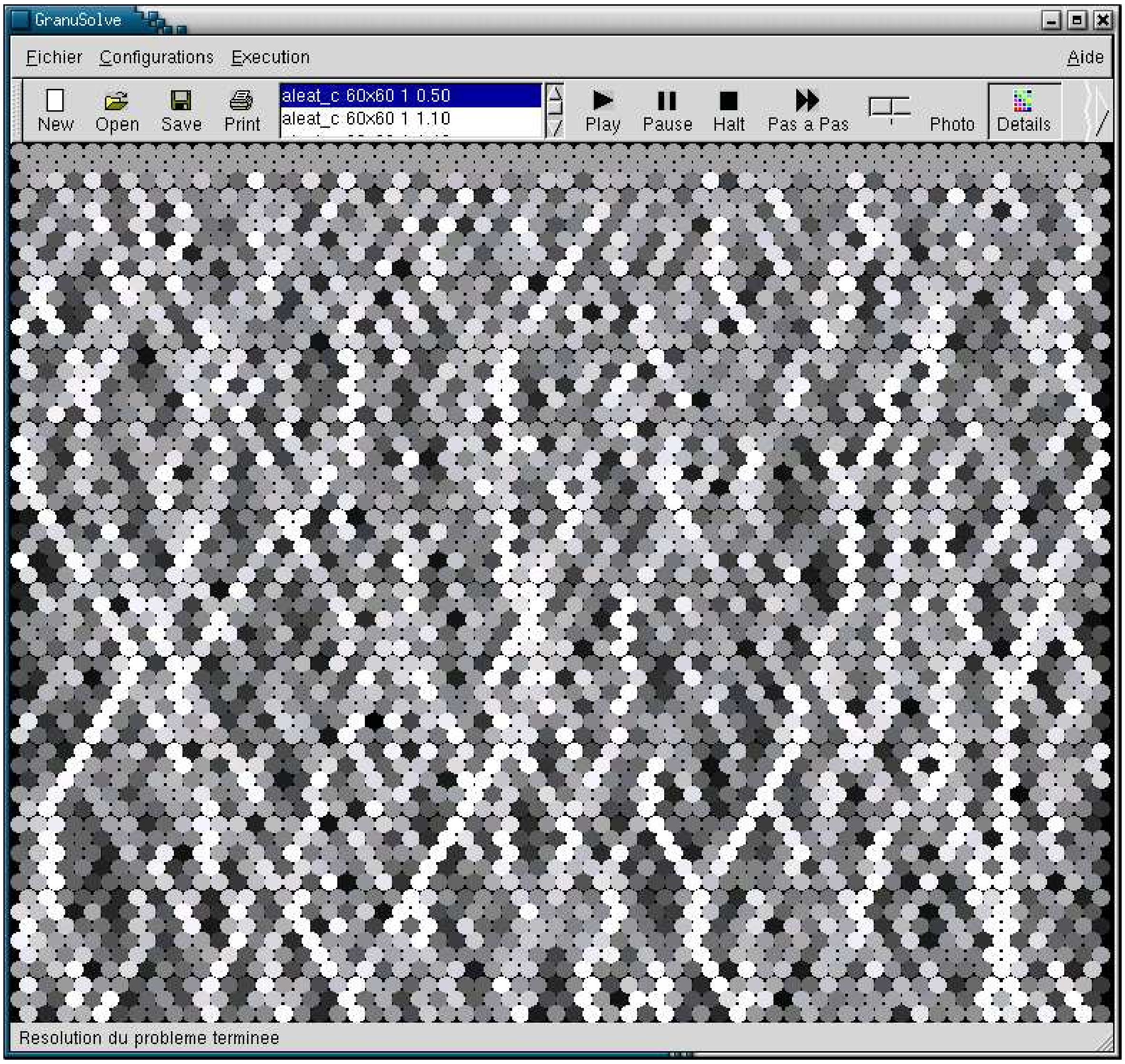


Fig. 4.15 – L'interface graphique de GranuSolve
Ecrite en C++ et avec l'aide des librairies graphiques Qt, l'interface permet de visualiser la résolution du tas. Des boutons de contrôle permettent d'arrêter la résolution ou de passer en mode «pas-à-pas».

des solutions et en particulier pour le cas des faibles valeurs de désordre.

Cette étude nous a permis de valider certaines hypothèses comme le fait que la valeur de frottement était considérée, par les physiciens de l'équipe, comme celle qui caractérise le désordre mécanique dans l'empilement. Les conclusions de ce travail nous ont convaincu de la nécessité de pouvoir manipuler des entités intermédiaires dans un système multi-agents de manière à avoir plusieurs niveaux de description d'un même système.

4.9.1 Perspectives

Plusieurs axes de recherche restent encore à explorer. Nous présentons ici les principales voies que nous pensons être intéressantes pour poursuivre nos recherches.

Les CSP

Nous avons considéré une façon simple de résoudre les équations de l'équilibre d'un grain car le problème était simple et toujours identique (4 contacts fixes). Dans le cas de tas plus réalistes (grains de tailles différentes), il faudrait pouvoir résoudre de façon plus générale les équations d'équilibre dans le cas de N contacts variables. En effet, dans un cas où le nombre de contacts et les angles associés varient, les équations du modèle ne sont plus adaptées. Il faudra alors envisager une résolution des équations par des méthodes de résolution de contraintes (CSP). L'expérience acquise d'un point de vue multi-agent peut être réutilisée pour la propagation des contraintes entre les grains et n'introduire des méthodes de CSP que dans la résolution locale d'un grain.

Les groupes d'agents

Nous avons défini deux types de groupes pour la résolution des tas de sable : les couches (groupes de grains) et le tas (groupe de couches). Il serait vraiment intéressant d'avoir une conception dynamique des groupes afin de capturer les phénomènes émergents. Ainsi, des entités intermédiaires telles que les lignes de forces (groupe de grains contigus ayant tous une charge supérieure à la moyenne) et les voûtes (groupe de deux lignes de forces ayant la même origine), etc., pourraient être manipulées et mesurées directement par le système.

La programmation des agents par composants

La définition des groupes pour notre système multi-agent a grandement simplifié l'algorithme en le décomposant en plusieurs niveaux récursifs. Aujourd'hui, il apparaît évident que les groupes apportent clarté et simplicité du code des agents.

A l'instar des groupes pour les agents, il faudrait introduire un niveau d'abstraction intermédiaire entre la programmation par agents et la programmation objet. Nous avons pu constater, au cours de l'implémentation de nos algorithmes, que le code d'un agent était peu robuste. En effet, le changement d'une ligne de code dans le comportement d'un agent peut avoir des conséquences catastrophiques au niveau de la résolution collective du système multi-agent. Aujourd'hui, il nous semble clair qu'il est nécessaire d'avoir un niveau d'abstraction intermédiaire entre le code objet et la conception d'agents. Ce niveau pourrait être comblé par la technologie des composants logiciels. Il serait très intéressant de pouvoir composer le comportement d'un agent (ou de façon récursive, les comportements eux-mêmes) à l'aide de briques élémentaires totalement indépendantes [Guillemet *et al.* 1999]. Nous pourrions ainsi tester, voire optimiser par évolution, différentes configurations possibles de comportements aussi bien au niveau de l'ordre d'exécution que des méthodes employées. Il est aussi possible d'imaginer de permuter facilement des éléments, par exemple une brique logicielle RésolutionDesEquations contre une autre RésolutionParCSP.

Les conditions limites

Enfin, une perspective importante serait d'utiliser un modèle plus réaliste des empilements de grains. Par exemple, il est envisagé d'ajouter des conditions limites (parois, plancher, etc.) pour simuler des systèmes de type silo ou broyeur.

L'autre possibilité pour complexifier le modèle est la prise en compte de la dynamique des grains. Avec cette dynamique, il est alors possible d'envisager un grand nombre d'applications (avalanches, rôle de l'histoire de l'empilement, phénomènes de fractures, etc.). Seulement, il ne faut pas perdre de vue que, pour l'instant, très peu de paramètres sont parfaitement maîtrisés dans les milieux granulaires.

Le chapitre suivant présente un modèle des expérimentations menées par les physiciens à partir de ces données. Nous décrivons ensuite le système GranuLab : un laboratoire virtuel d'expérimentations appliqué à la physique des milieux granulaires utilisant le solveur GranuSolve.

Chapitre 5

Vers une modélisation du processus de découverte en physique granulaire

Sommaire

5.1 Introduction

Dans le chapitre précédent, nous avons décrit notre algorithme de résolution de tas de sable à l'équilibre statique : GranuSolve. Cet algorithme est la condition nécessaire à la réalisation d'un environnement d'aide à la découverte. En effet, un tel environnement requiert naturellement des données pour fonctionner. Mais surtout, compte tenu des variations observées des descripteurs macroscopiques à l'échelle du tas, il est nécessaire de disposer d'un solveur capable de résoudre des tas de grandes tailles en des temps raisonnables afin d'obtenir une statistique fiable. Ces deux conditions étant réunies, nous pouvons maintenant envisager la construction de GranuLab, un environnement d'aide à la découverte scientifique appliqué à la physique des milieux granulaires.

Mais, la construction d'un tel environnement pose deux types de problèmes. D'une part, des problèmes d'ordres méthodologiques pour la représentation du processus de découverte. D'autre part, des problèmes technologiques pour la réalisation concrète de cet environnement. Dans la section 5.2 suivante, nous décrivons les problèmes méthodologiques que posent un tel environnement sous la forme d'un cahier des charges fonctionnel du système à implémenter. Nous traiterons ensuite, dans la section 5.3, des problèmes technologiques pour la mise en œuvre du système.

5.2 Un problème méthodologique : la modélisation du processus de découverte

Dans un système d'aide à la découverte, la modélisation du processus de découverte est la clé de voûte du système. La modélisation de ce processus repose en grande partie sur quatre aspects «fonctionnels» du système.

Le premier aspect tient à la définition correcte et exhaustive des objets d'études que l'expert souhaite manipuler dans le système. Ensuite, un deuxième aspect concerne les opérateurs élémentaires qu'il est nécessaire de fournir pour que l'expert puisse effectivement manipuler et opérer des traitements sur les objets d'études. Le troisième aspect relève de la modélisation d'une expérimentation virtuelle, i.e. de l'enchaînement correct des opérateurs et des objets étudiés. Enfin, au niveau de la conception globale du système, le dernier aspect traite de la modélisation du processus de recherche de l'expert.

Ces quatre aspects sont détaillés dans les sections suivantes.

5.2.1 Les objets d'études

La première étape consiste à définir les objets sur lesquels les physiciens travaillent. Ce sont les objets d'études. En physique des milieux granulaires, les chercheurs manipulent principalement des tas dont ils cherchent à extraire un comportement à l'aide de quelques descripteurs caractéristiques du milieu.

Mais avant de définir un tas, nous avons besoin de caractériser la façon dont ce tas est engendré en amont par le système : ce sont les conditions limites d'une expérimentation virtuelle.

Les conditions limites de génération d'un tas

Cet objet d'études regroupe les notions de préparation d'une expérimentation et de définition des *conditions limites* d'une manière générale. Ces conditions limites peuvent se décomposer en plusieurs objets de plus petites tailles :

- l'*environnement* permet essentiellement de donner un cadre physique à l'expérimentation (valeur de la gravité, taille maximale du système).
- la *géométrie du système* définit de façon plus précise la taille et la position des objets qui vont contenir le tas (parois, plancher, piston, etc.).
- le *grain* précise le nombre, la masse et la taille des grains utilisés pour construire l'empilement. Le système doit offrir la possibilité de choisir une taille de grain selon une loi de distribution gaussienne, par exemple. Des grains de tailles différentes[2] permettent d'ajouter un certain désordre géométrique dans l'empilement.
- la *méthode d'empilement* décrit la façon dont l'empilement doit être réalisé. De même que pour la taille des grains, des méthodes d'empilement «en pluie» ou à partir d'une source ponctuelle doivent être proposés à l'expert.

Toutes ces caractéristiques définissent la manière de préparer un empilement virtuel, à l'instar des expérimentations réelles.

Les tas

Il existe deux types de tas selon qu'ils soient résolus ou non. Cette distinction est importante car la résolution d'un tas déjà résolu n'a pas de sens et inversement le calcul d'un descripteur global sur un tas non résolu est voué à l'échec.

Un *tas non résolu* est un tas qui a été généré à partir d'une description de ses conditions limites par un *générateur de tas* (cf. opérateurs dans la section suivante). Ses caractéris-

[2]Les physiciens disent «polydisperses»

tiques sont essentiellement géométriques (position des grains, etc.) car elles sont issues de conditions limites particulières.

A l'inverse, un *tas résolu* est un tas initialement non résolu dont un solveur, par exemple GranuSolve, a donné une solution. En plus de ses caractéristiques géométriques, l'énumération complète des valeurs de forces en chacun des contacts est donnée. Un tas résolu possède ainsi la liste exhaustive des descripteurs de forces mécaniques exercées en chaque contact. Cette description est déjà beaucoup plus riche que la mesure donnée par un seul capteur sous le tas, dans le cas réel.

Les moyennes de tas

Cet objet est essentiel dans notre application, car il s'agit du principal objet d'études. En effet, les physiciens vont devoir moyenner un grand nombre de réalisations (tas résolus) afin de réduire le bruit et de dégager du signal. Cet objet *moyenne de tas* ne correspond pourtant pas à un tas résolu. Il s'agit en fait d'une abstraction d'un tas «moyen» et c'est sur cet objet que portent principalement les mesures.

Les objets de visualisation

Pour pouvoir interpréter les résultats donnés par le système, il est essentiel d'offrir des moyens de représenter l'information de façon concise. C'est le rôle des *courbes* et des *images*. Les *courbes* représentent simplement une série de couples de données [x, y] avec, éventuellement, des informations textuelles supplémentaires (légendes, etc.). Les *images* sont un moyen de visualiser une courbe ou la répartition de la charge (ou tout autre descripteur) dans un tas résolu.

Après avoir défini les principaux objets d'études de notre application, nous allons maintenant décrire les opérateurs permettant de traiter et de transformer ces objets.

5.2.2 Les opérateurs élémentaires

Dans cette section, nous décrivons les opérateurs que les physiciens utilisent quotidiennement dans leurs recherches. Il est important de bien définir ces opérateurs car ils sont le «pain quotidien» du chercheur en physique des milieux granulaires. L'oubli d'un opérateur peut être très gênant pour l'expert et rendre le système inutilisable.

Ces opérateurs peuvent être issus de l'IA (réseaux de neurones, classifications, etc.). Ils doivent constituer la boîte à outils de base de l'utilisateur. Il est donc important de mettre en œuvre les connaissances du domaine pour la conception de ces opérateurs. Cette tâche est souvent difficile car il faut demander aux experts un travail d'introspection pour décrire précisément les opérateurs qu'ils utilisent tous les jours. En effet, ces connaissances sont souvent «compilées», utilisées de façon implicites par les experts mais elles sont indispensables à ce type d'environnement. Un environnement d'aide à la découverte qui n'aurait pas la même approche que l'expert serait difficile à appréhender pour ce dernier.

Lecture d'un objet

La *lecture d'un objet* est un opérateur qui permet de charger un objet qui avait été enregistré auparavant pour le rendre accessible pour l'expérimentation courante (notion de persistance des objets d'études). Il est possible, à partir de la fonction de cet opérateur, d'imaginer une base de données d'objets prédéfinis parmi lesquels l'expert pourra sélectionner l'objet d'études qu'il désire (tas résolu intéressant pour l'expert, courbe, etc.).

Générateur de tas

Le *générateur de tas* crée simplement, à partir d'une description des *conditions limites*, un *tas non résolu*, i.e. le squelette géométrique d'un tas qui ne contient pas la totalité de l'information mécanique (forces de contact). La présence de quelques forces de contact est indispensable pour représenter des expérimentations du type fonction de Green (cf. Chap. 7) où un appui local est exercé sur la première couche de l'empilement. De plus, cette représentation autorise des tas partiellement résolus à être résolus de nouveau (notion de réentrance). C'est le cas lors d'une résolution inachevée d'un tas ou bien de la modification des conditions limites d'un tas résolu pour étudier les conséquences de ces modifications sur la structure de l'empilement existante.

Solveur de tas

Le *solveur de tas* génère une solution possible à partir d'un tas non résolu. Il est important de noter que GranuLab n'a pas pour vocation de reposer sur un unique solveur. D'autres solveurs que GranuSolve pourraient trouver leur place dans cet environnement et permettre ainsi la comparaison qualitative des solutions générées par des algorithmes différents. Actuellement, aucune représentation standard d'un empilement n'a été proposée parmi la communauté des physiciens qui travaillent sur les milieux granulaires. Les chercheurs de ce domaine qui réalisent des simulations numériques ne peuvent pas comparer directement leurs solutions. En effet, chacun ayant son format de représentation propre et ses opérateurs de transformation, il est alors difficile de comparer des résultats qualitatifs issus de multiples transformations à partir de tas résolus (moyennes, estimations des paramètres d'une fonction de distribution, etc.). GranuLab permettra d'avoir une représentation uniforme des objets d'études, de partager les opérateurs de transformation couramment utilisés et donc de pouvoir comparer directement deux méthodes distinctes. Ainsi, les biais implicites des algorithmes de simulation numérique pourront être mis en évidence. La connaissance des propriétés des algorithmes de résolution de tas en serait accrue.

Sommation de tas

La *sommation de tas* est une opération qui permet de calculer la moyenne des descripteurs de tas résolus. Le calcul «en place» de la moyenne de tas au fur et à mesure que les tas sont résolus par le solveur est une fonction qu'il est important d'intégrer. En effet, le calcul «en place» de la moyenne des descripteurs de tas résolus permet de ne pas bloquer la chaîne de traitement en aval qui est en attente du résultat de la moyenne des tas. Ainsi, il devra être possible de visualiser une courbe résultant de multiples transformations se dessiner et s'affiner au cours du temps, avec le nombre de tas résolus. Cette fonctionnalité devra permettre d'arrêter la résolution des tas lorsqu'une précision donnée a été atteinte sur la courbe-résultat finale.

Transformation d'un tas moyenné

Cet opérateur regroupe les *transformations* mathématiques sur les descripteurs ainsi que les changements de représentation. Les transformations mathématiques peuvent correspondre à une remise à l'échelle des données, au calcul d'une grandeur statistique, etc. Le changement de représentation consistera le plus souvent en une réduction du nombre de descripteurs (projections, courbes, etc.).

Transformation d'une courbe en une image

La transformation d'une *courbe* en une *image* peut être considérée comme une transformation avec changement de représentation.

Visualisation d'une image

Le système doit proposer un moyen de visualiser les images générées par l'utilisateur (visualisation de courbes, images de tas, etc.). Ces images se trouvent généralement à la fin d'une expérimentation virtuelle pour aider l'expert à interpréter les résultats. D'autre part, des fonctionnalités avancées en matière d'agrandissement et de recadrage de l'image devront être proposées car l'expert a souvent besoin de regarder plus précisemment une partie de courbe (asymptote, convergence, etc.).

Enregistrement d'un objet

Il est important pour l'expert d'avoir la possibilité d'enregistrer un objet particulier. Il permettra à l'utilisateur de ne pas avoir à ré-écrire à chaque utilisation ses configurations d'objets ou d'opérateurs. Pour qu'il puisse les réutiliser à volonté, le système devra donc mettre en œuvre les notions de persistance et de réutilisabilité des objets d'études.

Dans cette section, nous avons défini les principaux opérateurs qu'il est nécessaire d'implémenter dans notre système. Ces opérateurs sont définis pour effectuer des traitements et des transformations unitaires sur les objets d'études. Maintenant, nous allons nous intéresser à la spécification d'une expérimentation virtuelle ou comment enchaîner les opérateurs successifs qui permettent, à partir de la génération d'un tas, d'arriver à une courbe synthétisant un descripteur de l'empilement.

5.2.3 La modélisation d'une expérimentation virtuelle

La réalisation d'une *expérimentation virtuelle* nécessite de pouvoir enchaîner correctement les opérateurs avec les objets étudiés décrits précédemment.

En physique, la validation d'une expérimentation nécessite qu'elle soit au moins reproductible. Pour un système artificiel, il est donc important de pouvoir reproduire complètement une expérimentation donnée. Pour cela, il faut ajouter à chacun des opérateurs définis ci-dessus les paramètres nécessaires à la reproduction à l'identique de leurs comportements. Par exemple, un *générateur de tas* pourra prendre en paramètre la valeur de la «graine» du générateur de nombres aléatoires qui a été utilisée pour générer les tailles des grains selon une distribution gaussienne. Ainsi, la mémorisation de cette valeur nous garantit la totale reproduction du tas généré à partir de ses *conditions limites*.

5.2.4 La modélisation du processus de découverte

Du point de vue de l'aide à la découverte, la mémorisation d'une expérimentation complète est la condition nécessaire à l'utilisation de GranuLab comme d'un «carnet d'expérimentations virtuelles» du chercheur en physique. La mémorisation de chacune des expérimentations virtuelles menées par l'expert représente la trace visible du cheminement parcouru par le chercheur pour arriver au résultat final. De plus, il devient alors possible de comparer les voies empruntées par deux chercheurs distincts sur un même problème et de se communiquer ainsi plus facilement leurs raisonnements respectifs.

Du point de vue des systèmes d'informations, cette mémorisation va faciliter le partage des expérimentations et leur publication directe en ligne. Il est tout à fait envisageable de créer un Intranet pour une équipe de chercheurs et d'y déposer les expérimentations et résultats de chacun. Ainsi, tous les membres de l'équipe pourront bénéficier à la fois des résultats et des matériaux qui ont servi à obtenir ces résultats.

5.3 Un problème technologique : l'opérationnalisation du processus de découverte

Nous avons définis, dans la section précédente, les besoins et les fonctionnalités nécessaires à implémenter dans notre système. Nous proposons, dans cette section, des modèles pour chacune des entités et des fonctionnalités requises. Nous commençons par les entités atomiques de notre système : les objets d'études.

5.3.1 Le choix d'un langage uniforme de représentation des objets d'études

Des objets d'études très hétérogènes peuvent être appliqués à certains opérateurs comme l'opérateur de transformation (tas, courbes, etc.). Nous avons donc besoin d'un langage de représentation homogène des objets d'études pour ne pas avoir à multiplier le nombre d'opérateurs par le nombre de représentations. De plus, du point de vue de l'extensibilité du système à d'autres objets d'études, il est préférable d'adopter un langage de représentation uniforme pour tous les objets dès la conception du système.

Nous avons choisi, pour notre système GranuLab, de décrire tous les objets d'études à l'aide du langage XML ([XML 1998, Eckstein et Casabianca 2000]). Ce langage est assez récent et il a hérité de l'expérience acquise par les concepteurs de son ancêtre : SGML. XML est un langage de balises comme l'est HTML. Mais, contrairement au langage HTML, l'ensemble des balises utilisables avec XML n'est pas prédéfini et figé. C'est à l'utilisateur de créer son propre ensemble de balises, selon ses besoins. Ainsi, XML peut être considéré comme un méta-langage. Par conséquent, les langages dérivés de XML doivent tous respecter un certain nombre de règles syntaxiques propres à XML. Un document XML qui respecte ces règles est dit «bien formé».

L'ensemble des balises définies pour un nouveau type de document XML est appelé *Document Type Definition* (DTD). Il s'agit d'une définition du nom des balises, de leur contenu (attributs et types) mais aussi de la grammaire qu'un document XML doit respecter pour être «valide» par rapport à sa DTD. Aujourd'hui, il existe des éditeurs XML qui permettent de valider le document par rapport à la DTD à laquelle il a été déclaré conforme ([ChannelPoint 2001]).

Plusieurs communautés de scientifiques ont déjà proposé, de façon collective, des DTD pour leurs domaines respectifs. Par exemple, en mathématiques, il existe une DTD appelée MathML qui permet de décrire des objets mathématiques. L'intérêt de spécifier ainsi la description d'objets d'études est double. D'une part, l'échange et la communication entre chercheurs éloignés est facilitée : une application qui lit les documents XML respectant une certaine DTD pourra lire le document XML produit par un autre utilisateur. D'autre part, le format d'un fichier de données n'est plus propriétaire mais explicité dans la DTD. Cette «ouverture» permet de libérer les applications de la gestion des différentes versions du format de représentation. En effet, les concepteurs de logiciels rencontrent toujours des problèmes lorsqu'ils changent de format de représentation des données d'une application. Ils doivent normalement assurer une «compatibilité ascendante» et implémenter dans l'application des filtres pour pouvoir lire les documents décrits dans l'ancien format. Avec le langage XML, les applications n'ont plus à gérer les formats des documents. En effet, comme les DTD de l'ancien et du nouveau format sont connues, il suffit de fournir un compilateur pour transformer les documents de l'ancien format vers le nouveau. L'application, quant à elle ne lira que les documents au nouveau format. Il est même possible de fournir un compilateur pour assurer une «compatibilité descendante» vers l'ancien format.

Les conditions limites de génération d'un tas

Nous avons écrit une DTD pour représenter les *conditions limites* de génération d'un tas. La version complète de la DTD est détaillée dans l'annexe B.1. Cette DTD permet de décrire la préparation d'un empilement confiné entre parois ainsi que la méthode de construction : «en pluie» ou à partir d'une source ponctuelle.

Les tas

Nous avons également proposé une DTD pour décrire un empilement (détaillé dans l'annexe B.2). Cette grammaire permet de décrire un tas en deux parties. La première concerne les aspects géométriques (position, taille des grains, parois, etc.). La seconde permet de décrire les contraintes mécaniques (les forces de contacts inter-grains) qui s'appliquent sur les entités géométriques (grain, parois). Nous n'avons pas fait de distinction entre les *tas résolus* et les *tas non résolus* pour des raisons de simplicité. C'est le solveur qui, en chargeant un tas, vérifie s'il est résolu ou non.

Les moyennes de tas

L'objet *moyenne de tas* est représenté avec la même DTD qu'un tas. Il faut simplement préciser le nombre de tas représentés. Un tas unique aura donc une valeur unitaire pour l'attribut *nb_tas_représentés*.

Les courbes

En ce qui concerne les courbes, nous avons écrit une grammaire simple décrivant une liste de couples [x, y] (cf. annexe **??**).

Les images

Pour les images, nous avons utilisé une norme existante : SVG (Scalable Vector Graphics) [SVG 2000]. Cette grammaire permet de décrire une scène graphique à l'aide d'objets géométriques simples ou d'images. Les effets appliqués sur ces objets (filtres graphiques standards) et des animations peuvent aussi être décrits dans ce format. Des logiciels de rendu graphique [Adobe 2001, BATIK 2001] de ces fichiers permettent d'effectuer des manipulations graphiques à la souris tels que les agrandissements, les rotations mais aussi la sauvegarde sous un format de fichier de type image (jpeg ou png).

5.3.2 Le choix d'un modèle générique d'opérateurs

Les opérateurs sont de plusieurs natures mais nous avons dégagé un modèle générique d'opérateurs sous la forme d'un composant logiciel possédant des canaux d'entrées et de sorties. Ce composant générique encapsule une méthode de traitement des données qu'il reçoit sur ses canaux d'entrée. Ce modèle de composant permet de représenter tous les autres composants dont nous avons besoin pour notre système et ainsi de réduire le coût du développement. Notre but était de minimiser le nombre de composants à implémenter mais également de réduire le coût de leurs implémentations. L'utilisation du langage XML nous a ainsi permis d'utiliser les librairies existantes pour le traitement des données au format XML. Nous présentons maintenant les cinq types de composants que nous avons implémentés et qui suffisent pour représenter tous les opérateurs dont nous avions besoin.

Le composant ChargerDisque

Ce composant est très simple. Il ne prend aucune entrée et envoie, sur son canal de sortie, les noms des fichiers sélectionnés par l'utilisateur.

Le composant SauverDisque

Ce composant est le symétrique du précédent. Il ne génère pas de sortie réutilisable par le système, mais sauvegarde sur le disque les fichiers qui lui sont donnés en entrée.

Le composant LigneDeCommande

Ce composant permet d'exécuter une commande système extérieure à l'application. Par exemple, un solveur écrit dans un langage différent de celui du système pourra être appelé à l'aide de ce composant.

Le composant XSLT

Ce composant permet d'appliquer des règles de transformation XSLT à un fichier XML en entrée. La norme XSLT [XSLT 1999] permet de décrire, au format XML, des règles de réécriture d'un document XML vers un document texte (essentiellement XML). La DTD du document source étant connue, il est alors possible d'écrire des règles de réécriture basées sur les noms des balises du document fourni en entrée. Ces règles sont du type : «Si la balise [nom] est rencontrée, Alors exécuter actions d'écriture». Il est donc possible d'écrire simplement un ensemble de règles pour renommer certaines balises, calculer la somme d'un attribut pour un ensemble d'éléments, etc.

Ainsi, tous les opérateurs de notre système peuvent être décrits à l'aide de cet unique composant, sauf le *solveur* et le *générateur de tas* qui nécessitent des structures issues des langages de programmation.

Le composant VisualiseurSVG

Ce dernier composant apporte une solution au problème de la visualisation des *images*. La norme SVG [SVG 2000] permet de décrire des scènes graphiques. Ce composant utilise une librairie de rendu graphique de documents au format SVG. Nous n'avons donc pas eu à implémenter directement de fonctions graphiques complexes pour notre système. Cette norme répond parfaitement à nos attentes, puisqu'il s'agit de dessin vectoriel, les opérations d'agrandissement et de redimensionnement de l'image se font sans aucune perte de qualité.

5.3.3 Le choix d'un modèle de représentation et d'exécution d'une expérimentation virtuelle

Les cinq composants de base de notre système étant définis, nous avons proposé un modèle de représentation d'une expérimentation virtuelle à l'aide d'un graphe orienté. Ce graphe décrit les composants utilisés (les nœuds) et les liens entre composants (les arcs). Des contraintes peuvent alors être ajoutées au graphe (acyclique, etc.) afin de limiter les graphes qu'il est possible d'engendrer par l'utilisateur.

Nous avons également écrit une DTD pour représenter une *expérimentation virtuelle* et pouvoir l'enregistrer dans ce format. La DTD est détaillée en annexe B.4.

5.3.4 Le choix de la représentation et de la mémorisation du processus de découverte

Enfin, le processus de découverte est aussi décrit dans un format XML. La DTD d'une «étude» (présentée en annexe B.5) permet de décrire une problématique et les sous-problématiques qui en découlent. Il est aussi possible de consigner clairement l'énoncé d'une problématique, les conditions limites pour lesquelles elle s'applique (domaine d'investigation), les expérimentations virtuelles menées pour tenter d'y répondre et enfin de résumer les résultats obtenus. Le chercheur a ainsi une vision globale des problèmes, des résultats obtenus et d'une façon générale de l'avancement de ses travaux et des différentes voies explorées.

5.4 Conclusion

Dans ce chapitre, nous avons donné un cahier des charges fonctionnel pour notre application GranuLab. Ce cahier des charges nous a permi de dégager les objets d'études et les opérateurs dont les physiciens ont besoin.

Ensuite, nous avons proposé un format uniforme de représentation des entités de notre système à l'aide du langage XML. Nous avons donc décrit les DTD associées à la représentation de chacune de ces entités. De plus, l'utilisation du langage XML nous a permis d'exploiter des standards existants comme XSLT pour transformer des documents XML et SVG pour décrire des scènes graphiques vectorielles. L'utilisation de librairies associées à ces standards nous a enfin permi l'économie de conception d'outils de traitement des documents pour ces formats.

Dans le chapitre suivant, nous présentons l'implémentation de notre système.

Chapitre 6

Mise en œuvre dans GranuLab

Sommaire

6.1 Introduction

Le principal objectif du système GranuLab est de concevoir et de développer un laboratoire virtuel d'expérimentations pour l'aide à la découverte scientifique en physique des milieux granulaires.

Des environnements de fouille de données de troisième génération tels que Clémentine ou Enterprise Miner [Khabaza et Shearer 1995, Potts 1997] permettent de décrire le processus de la fouille. Ainsi les étapes fastidieuses de filtrage de données, de choix d'algorithmes et de visualisation des résultats sont tous représentés. Si leur approche, d'un point de vue ingénierie, est un succès incontestable, ces systèmes recourent souvent à des formats de données propriétaires. Dans le cadre du développement de GranuLab et pour une souplesse de réutilisation nous avons choisi d'utiliser XML [XML 1998] comme standard de représentation des connaissances.

Les principaux avantages de cette solution sont d'une part de permettre une plus grande modularité afin que des algorithmes déjà existants puissent être facilement ajoutés à l'environnement sous forme de nouveaux composants et d'autre part d'initier la création d'une représentation standardisée (DTD) des données physiques relatives aux expérimentations. Cela facilite ainsi l'échange de données entre physiciens et permet la reproductibilité des expérimentations virtuelles, ce qui est un des points cruciaux pour leur acceptation de GranuLab comme méthode d'investigation scientifique.

Dans ce chapitre, nous détaillons l'implémentation de GranuLab. Je tiens à remercier Anneli Lénica et Cécile Le Pape qui ont implémenté l'architecture componentielle de GranuLab dans le cadre d'un stage de Maîtrise d'Informatique.

6.2 Choix technologiques

6.2.1 Le langage XML pour représenter les connaissances

Le langage XML, grâce à la présence d'une grammaire explicite et séparée du document, permet une évolutivité de la structure même des documents générés.

D'un point de vue un peu plus pratique, le langage XML autorise l'édition d'un document contrainte par sa grammaire (DTD). Ainsi, à l'aide d'un simple éditeur XML [ChannelPoint 2001], un physicien se voit proposer des choix contextuels de balises à insérer afin de garantir la validité du document au cours de son édition.

L'un des objectifs de GranuLab est de représenter puis d'exécuter les différentes étapes d'une expérimentation virtuelle menée par un chercheur en physique granulaire.

Par la suite, ces étapes sont exécutées automatiquement par le système.

6.2.2 Le langage Java pour l'implémentation

Le langage Java, comme support pour l'implémentation de notre architecture s'est naturellement imposé. En effet, de nombreuses librairies et logiciels exploitant le format XML sont écrits en Java. De plus, la dynamique de ce langage et de la communauté de ses utilisateurs nous ont séduit. Mais surtout, la portabilité d'une application écrite en Java était un atout indispensable au développement de notre système car les membres de notre équipe travaillons sur des systèmes d'exploitation différents (LinuxPPC, Linux«Intel» et Windows). Enfin, les aspects réflexifs de ce langage nous ont permis de rendre notre système un peu plus extensible par l'instanciation dynamique d'un type de composant à l'exécution à partir de son nom.

6.3 Les opérateurs élémentaires

Les opérateurs élémentaires ont été définis dans le chapitre précédent. La modélisation de ces opérateurs à l'aide de composants traitant des données XML est décrite dans les sections suivantes.

6.3.1 Le ComposantGénérique

Afin de pouvoir relier les composants entre eux, nous avions besoin d'un modèle objet de composant homogène. Nous avons choisi de considérer un modèle de composant ne traitant que des données XML en entrée. Les données de sortie pouvant bien sûr être dans un format différent (HTML, image, etc.).

Un composant générique est implémenté par une classe abstraite qui possède un état actif, un nombre de canaux de communication, des liens déjà connectés à ses canaux et certaines propriétés (cf. Fig. 6.1). Les propriétés d'un composant définies dans un format XML servent à paramétrer le comportement du composant. Par exemple, un fichier de règles XSLT est le paramètre d'un composant du type Xslt et les filtres de noms de fichiers sont ceux d'un composant ChargerDisque.

Le principal contrôle effectué par le ComposantGénérique concerne son état d'activation. Il est activable si tous ses canaux déclarés sont effectivement liés à d'autres composants. L'activation d'un composant entraîne son exécution asynchrone, à savoir attendre l'arrivée de données sur ses canaux d'entrée, traiter ces données et les transmettre au composant suivant via ses canaux de sortie. Nous avons choisi de ne faire transiter sur

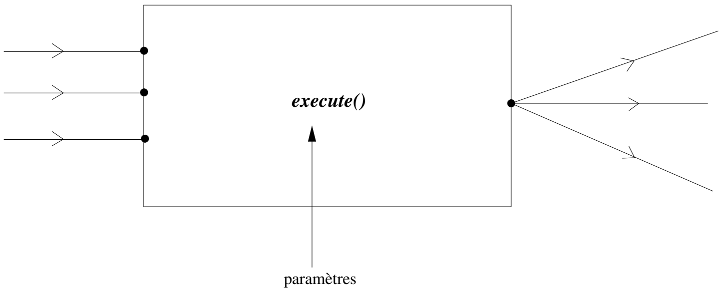


FIG. 6.1 – Représentation schématique de notre modèle de composant
Un composant possède N canaux d'entrée de données au format XML. Les données en sortie sont au format texte. Les paramètres du composant permettent de configurer le traitement effectué.

les canaux de communication entre composants que les noms des fichiers XML à traiter. En effet, pour notre application à la physique des milieux granulaires, nous avons des volumes importants de données à traiter, ce qui rend impossible le chargement complet de l'arbre XML en mémoire (modèle DOM).

Afin d'abstraire encore le modèle de composant générique, nous avons choisi d'implémenter deux contrôleurs pour gérer les flux de données en entrée et en sortie. Deux contrôleurs INPUTUNIT et OUTPUTUNIT sont ainsi ajoutés au modèle de composant générique. Le module INPUTUNIT se charge de contrôler les entrées sur les canaux d'entrée du composant (cf. Fig. 6.2). Lorsque toutes les entrées ont été connectées et ont reçu un message, le module renvoie la main au composant générique qui peut alors traiter les données. Inversement, le module OUTPUTUNIT se charge de transmettre les données générées par le composant aux composants suivants.

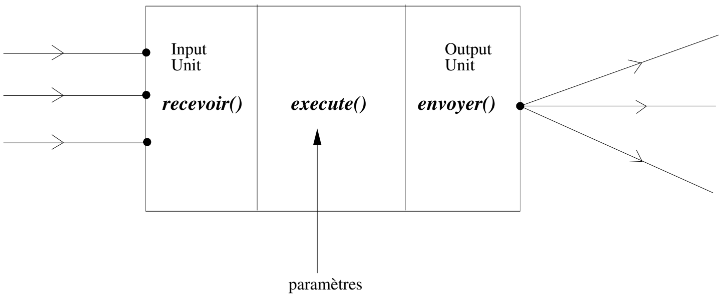


FIG. 6.2 – Représentation schématique de notre modèle de composant générique
La gestion des entrées et des sorties est encapsulée dans les modules INPUTUNIT et OUTPUTUNIT.

La méthode d'exécution d'un COMPOSANTGÉNÉRIQUE *execute()* est alors la suivante :

Ces deux modules de gestion des entrées/sorties permettent d'abstraire un peu plus

Algorithme 6.1 ComposantGénérique.*execute*()

données ← InputUnit.*recevoir*()
Tant Que il y a des données à traiter en entrée **Faire**
  résultats = *traitement*(données)
  OutputUnit.*envoyer*(résultats)
  données = InputUnit.*recevoir*()
Fin Tant Que

le modèle d'un composant générique et de factoriser le code. Ainsi, pour un composant ChargerDisque, la méthode InputUnit.*recevoir*() renverra directement la main au composant, sans attendre aucune donnée en entrée. Inversement, pour un composant SauverDisque, la méthode OutputUnit.*envoyer*() ne fera rien.

Les composants actuellement disponibles dans GranuLab sont décrits dans les sections suivantes. Ils sont tous dérivés de la classe ComposantGénérique.

6.3.2 Le composant ChargerDisque

Ce composant charge une liste de noms de fichiers à partir du disque et communique cette liste au composant suivant. Ce composant permet d'automatiser, par exemple, la résolution de plusieurs problèmes physiques stockés dans un même dossier.

6.3.3 Le composant SauverDisque

Il s'agit du composant inverse du précédent. Ce composant enregistre sur le disque les fichiers dont les noms lui sont donnés en entrée. Un paramètre *compteur* permet d'enregistrer une série de fichiers sous des noms distincts (*nom+numéro*).

6.3.4 Le composant CommandeSystème

Ce composant sert à exécuter un programme externe à l'application Ce programme doit prendre en entrée des données XML. Nous utilisons ce composant pour le solveur de tas qui est appelé avec le nom du fichier décrivant le tas à résoudre. En effet, le solveur est écrit en C++ et il est plus simple, en Java, d'exécuter une commande externe que d'appeler directement une fonction C++. Ce composant est aussi utile pour le cas du générateur de tas. Pour l'instant, le générateur de tas n'a pas été implémenté car, jusqu'à présent, nous générions directement les tas à résoudre.

6.3.5 Le composant Xslt

Ce composant prend un fichier de règles XSLT en paramètre. Il permet d'appliquer les règles aux fichiers d'entrée et renvoie les fichiers résultats [XSLT 1999]. Les extensions du langage XSLT permettent de traduire des opérations numériques simples (calcul de maxima, lissage gaussien, ...) sans avoir à écrire de longs programmes. L'extension la plus courante est la prise en charge du langage JavaScript, mais ce n'est pas le seul langage (Perl, etc.). Les extensions sont possibles dans la norme XSLT mais leur prise en charge (interprétation) dépend du processeur qui effectue la transformation. Par exemple, Xalan gère les extensions en JavaScript [Xalan 2001]. Toutes les opérations de traitements des données peuvent ainsi être traduites sous la forme d'une séquence de traitements élémentaires utilisant ce composant.

Nous utilisons le processeur Xalan écrit en Java pour appliquer les transformations requises sur les documents XML. Il est à noter que, grâce au composant CommandeSystème, il est possible d'envisager l'utilisation de processeurs XSLT qui ne soient pas basés sur le langage Java mais sur d'autres tels que Perl, Python, etc. Une telle permutation des composants peut être envisagée dans le cas de traitements de documents XML très volumineux (plus de 2 Mo).

6.3.6 Le composant VisualisationSVG

Ce composant utilise les librairies Batik [BATIK 2001] pour effectuer un rendu graphique de dessins vectoriels (cf. Fig. 6.3). Cette librairie, écrite en Java, permet en outre de réaliser des agrandissements et des rotations ainsi que de sauvegarder la vue sous un format d'image standard (jpeg ou png).

6.3.7 Le composant Groupe

La notion de groupe apporte une définition récursive des composants, un groupe de composants étant lui-même un composant. Ce composant autorise la création de chaînes typiques de traitement que l'utilisateur peut alors sauvegarder comme étant un seul méta-composant. Le temps qui nous était imparti pour cette thèse ne nous a pas permis d'implémenter ce composant qui est cependant indispensable à la création de schémas d'expérimentations complexes.

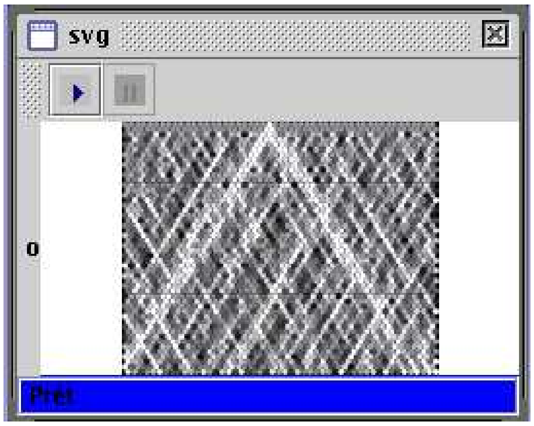


FIG. 6.3 – Visualisation au format SVG des charges des grains dans un tas
Le tas est soumis à une surcharge sur un grain de la première couche. Les lignes de forces apparaissent en blanc sur le dessin.

6.3.8 La représentation graphique d'un composant

Du point de vue graphique un composant est représenté par une fenêtre (cf. Fig. 6.4).

FIG. 6.4 – Représentation graphique d'un composant d'un schéma d'expérimentations
Les boutons permettent de contrôler le composant indépendamment des autres (*play*, *stop*). Une barre d'état informe de l'état du composant.

La fenêtre d'un composant se décompose en quatre parties :

La barre de boutons. Les boutons *play* et *stop* permettent de contrôler l'activation du composant indépendamment des autres.

Les barres d'état. La barre de progression donne une estimation du temps restant pour finir de réaliser la tâche en cours. La barre d'état textuelle, en bas de la fenêtre donne, quant à elle, l'état d'activation du composant. Une diode, de couleur verte pour un état activé et rouge sinon, est en cours d'étude.

Les points de connexion. Les points de connexion, de chaque côté de la fenêtre, permettent de connecter le composant avec un autre composant.

Le milieu de la fenêtre. Au milieu de la fenêtre, une zone de dessin est prévue notamment, pour le composant VISUALISATIONSVG. La visualisation d'un dessin vectoriel se fait directement dans cette zone.

Nous venons de passer en revue les six types de composants qui sont implémentés dans notre système. A l'aide de ces composants, il est possible d'effectuer un très grand nombre d'opérations, notamment grâce au composant XSLT. Il est aussi important de noter que la charge de programmation a été réduite au maximum par la réutilisation de librairies Java existantes pour le traitement de documents XML. Ainsi, notre application profite des développements et mises à jour régulières de ces librairies.

Enfin, pour faciliter l'extensibilité de notre système, nous avons utilisé les capacités de réflexivité du langage Java pour créer dynamiquement les composants suivant leurs noms issus du fichier XML. Ainsi, l'ajout d'un nouveau type de composant nécessitera seulement de rendre accessible, pour l'application, le fichier compilé de ce nouveau composant et la modification de la DTD d'une expérimentation virtuelle. Cela signifie qu'il n'y aura pas une seule ligne de code à modifier dans l'application, ni à recompiler tout le système, ce qui est très appréciable.

6.4 Les expérimentations virtuelles

6.4.1 La représentation sous forme d'un graphe

Un schéma d'expérimentation est représenté, en mémoire, par un graphe orienté de composants. Ce formalisme permet de traduire facilement des contraintes de structure du schéma en termes de propriétés du graphe. Ainsi, les conditions à l'exécution d'un schéma sont que le graphe soit sans cycle et que ses sommets source et puits uniques. Nous avons utilisé la librairie Tetrad développée à l'université de Carnegie Mellon qui autorise l'ajout de contraintes sur un graphe [Tetrad 2001]. Par exemple, un graphe orienté acyclique pourra refuser l'ajout d'un arc qui créerait un cycle dans le graphe et violerait ainsi une des contraintes. Ainsi, la modélisation fine d'un schéma d'expérimentations en termes de graphe nous libère de la gestion de l'édition du schéma. De plus, cette librairie, propose des représentations graphiques pour les liens entre les nœuds du graphe.

6.4.2 L'exécution d'un schéma d'expérimentation

L'exécution d'un schéma, une fois sa validité vérifiée (contraintes sur le graphe), s'opère en activant le premier composant source du graphe. Le premier composant va alors activer les composants suivants en même temps qu'il leur envoie les résultats de son traitement et ainsi de suite jusqu'au dernier. Un composant qui a terminé son exécution envoie un message de fin aux suivants. Les composants qui reçoivent un message de fin sur l'ensemble de leurs entrées s'arrêtent également et envoient, à leur tour, un message de fin d'exécution.

L'exécution du traitement de chaque composant est pris en charge par un processus léger (thread), ce qui autorise un certain parallélisme dans l'exécution du schéma.

6.4.3 La représentation graphique d'un schéma

La représentation graphique d'un schéma est basée sur la représentation des composants et des liens entre ces composants (cf. Fig. 6.5).

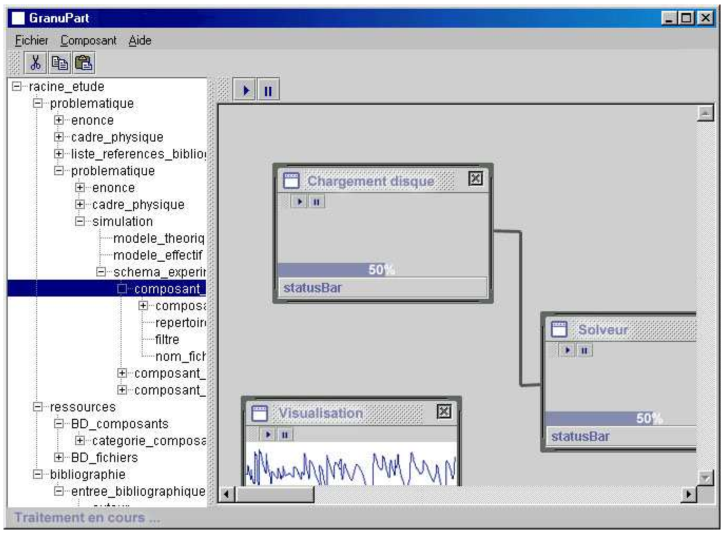


FIG. 6.5 – Représentation graphique d'un schéma d'expérimentations

6.5 L'opérationnalisation du processus de recherche en physique

6.5.1 La représentation d'une *étude*

La notion d'étude a fait l'objet d'une modélisation sous forme de grammaire XML (cf. Annexe B.5). Ce format autorise la représentation de connaissances hétérogènes dans un même document et sous une forme arborescente. Nous avons ainsi pu représenter la notion de problématique posée par le chercheur et l'associer directement aux expérimentations virtuelles réalisées et aux résultats obtenus.

D'un point de vue de la représentation en mémoire d'une *étude*, nous avons utilisé le modèle DOM («Document Object Model»). Ce modèle de représentation d'un document XML est normalisé par le consortium W3C («World Wide Web Consortium»). Les librairies qui implémentent ce modèle sont donc tenues de fournir un certain nombre de primitives pour manipuler l'arborescence d'un document XML en mémoire. Nous avons, là encore, utilisé une librairie Java existante : Xerces [Xerces 2001] qui implémente la norme DOM.

6.5.2 La représentation graphique de l'étude

La représentation graphique de l'étude a été implémentée à l'aide d'un composant JTree qui permet de manipuler une arborescence (cf. Fig. 6.6).

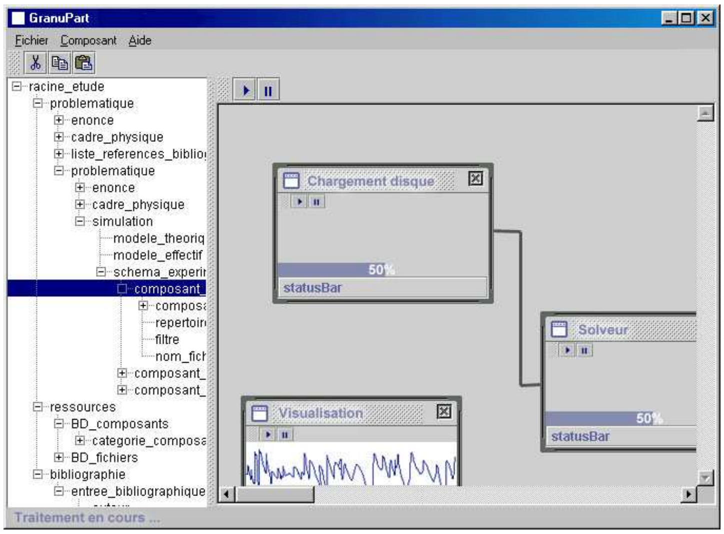


Fig. 6.6 – GranuLab : vue de l'étude XML

6.6 Conclusion

L'application GranuLab présentée est basée sur une architecture componentielle de traitement de documents XML. L'association d'une technologie de composants et du langage XML nous a permis d'implémenter simplement notre application, en utilisant au maximum les librairies existantes.

Nous avons proposé plusieurs standards XML sous forme de grammaires (DTD) pour décrire notamment des tas de sable à l'équilibre statique et des protocoles expérimentaux. Ces propositions sont actuellement en discussion avec les physiciens de l'équipe, spécialistes des milieux granulaires, en vue d'une procédure de pré-normalisation.

Cette architecture componentielle associée à XML est aujourd'hui vue par les physiciens comme une occasion unique de regrouper les différents algorithmes de résolutions qui, pour la plupart, ne sont utilisables que par leurs concepteurs. La définition de standards de représentations est aussi un premier pas vers la reproductibilité des expérimentations faites en simulation, garante de la démarche scientifique des physiciens.

A noter que l'application prend en charge l'internationalisation (y compris les raccourcis clavier) et que la plupart des propriétés graphiques sont configurables grâce à un fichier externe regroupant les propriétés de l'application.

Perspectives

Certains aspects de notre système demandent à être améliorés ou étendus. C'est le cas du composant Groupe qui permettrait d'abstraire de longues chaînes de traitement sous la forme d'un unique composant.

La représentation et la mémorisation du processus de recherche de l'expert au format XML permet de générer simplement des pages HTML décrivant les expérimentations et les résultats obtenus par chacun des membres de l'équipe. Ces pages peuvent être alors mises à disposition sur l'Intranet d'une équipe de chercheurs afin qu'ils puissent communiquer plus facilement la matière de leurs travaux. Ce moyen de communication à l'intérieur d'un groupe de chercheurs autorise la comparaison des méthodes et des outils employés par plusieurs chercheurs différents sur un même problème. Ainsi, chacun peut comparer ses études avec celles réalisées par les autres membres de son équipe.

Une des perspectives de recherche serait de pouvoir mettre en évidence les différences entre deux arbres XML afin de mettre en évidence les parties de l'arbre d'une étude qui diffèrent d'un chercheur à l'autre.

Chapitre 7

Expérimentations avec GranuLab : étude détaillée de la «fonction de Green»

Sommaire

7.1 Introduction

Un des problèmes majeurs rencontrés par les chercheurs en physique des milieux granulaires est la réalisation d'expérimentations réelles avec des tas de sable et l'analyse de leurs réactions lorsqu'un des grains est soumis à une pression. Cette expérimentation porte le nom de *fonction réponse* ou *fonction de Green*. Ces expérimentations étant coûteuses en temps et en équipements, nous illustrons l'utilisation de GRANULAB pour déterminer la réponse du système en fonction de la valeur de frottement et de la force d'appui initiale. La taille des tas de sable que peut résoudre notre algorithme GRANUSOLVE (millions de grains) est d'un ordre de magnitude de quelques degrés supérieur à ceux que les algorithmes classiques de simulation peuvent résoudre. Ces performances ont ouvert la voie à la modélisation du processus d'aide à la découverte pour ce problème précis.

Nous avons élaboré un modèle complexe pour explorer l'espace de ces lois. Pour chaque valeur de frottement, des milliers de tas sont résolus et leurs réponses analysées, renormalisées (pour obtenir une courbe universelle) afin d'apprendre la forme de la réponse par régression. Après plusieurs mois de calculs, le système a permis la validation d'une théorie mathématique caractérisant le milieu.

Dans la section 7.2 suivante, nous replaçons cette problématique dans son contexte de recherche scientifique en physique des milieux granulaires. Nous présentons en 7.3 la modélisation de cette expérimentation par l'enchaînement d'opérateurs de GRANULAB. Le détail des expérimentations menées pour l'étude la fonction de Green fait l'objet de la partie 7.4. Enfin, les résultats obtenus sont illustrés dans la section 7.5.

7.2 Le contexte physique de la fonction de Green

Les chercheurs en physique des milieux granulaires cherchent à *caractériser macroscopiquement le milieu granulaire* en tant que système physique complexe.

Pour des milieux denses et homogènes, il existe souvent des méthodes simples et efficaces pour mesurer et pour caractériser les propriétés du milieu étudié. C'est le cas dans le domaine de l'optique où une méthode pour caractériser le comportement macroscopique d'un solide translucide consiste à envoyer un rayon lumineux à travers le milieu et à mesurer ensuite la réponse du système à ce signal. Pour des verres très opaques, le système se caractérise par une importante *diffusion* de la lumière, alors que pour des verres parfaitement transparents, le milieu se caractérise par une plus importante *propagation* du signal.

Pour mesurer le caractère propagatif et/ou diffusif d'un milieu granulaire, les chercheurs ont eu l'idée de procéder de façon analogue à la méthode employée en optique. L'expérimentation consiste à mesurer la réponse du système «milieu granulaire» sur lequel une surcharge est appliquée localement sur un grain (cf. Fig. 7.1).

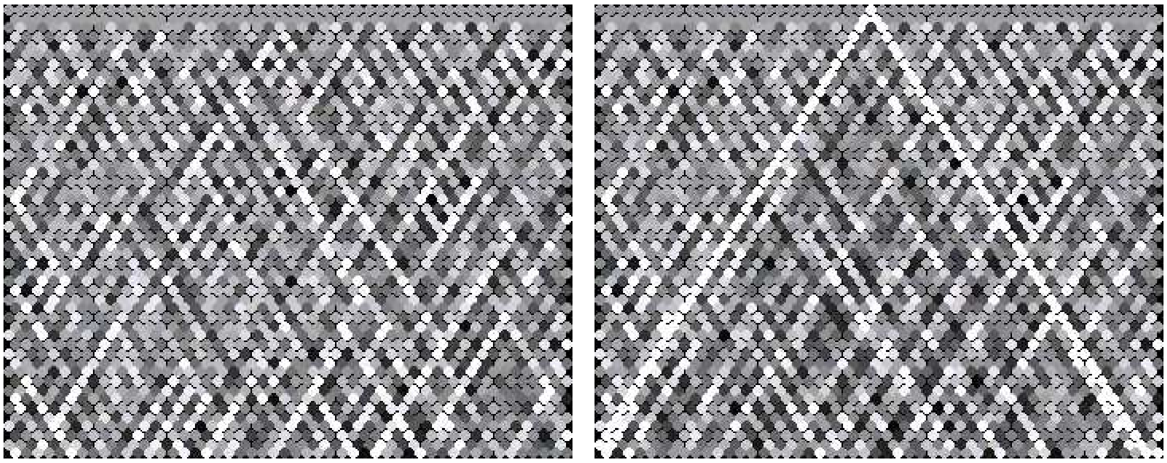


FIG. 7.1 – Exemple de visualisation de la réponse de la fonction de Green
Pour une faible valeur de frottement (μ =0,20), deux cas sont illustrés a) Charge uniforme sur la première couche b) Surcharge sur un grain de la première couche

Seulement, la mesure de la réponse du système à ce signal est difficile à réaliser dans le cas réel (cf. Chap. 3). Dans le meilleur des cas, les chercheurs arrivent à obtenir la réponse pour un seul grain à la base du tas. Et, pour obtenir la réponse de tous les grains à la base du tas, il leur faut reconstruire autant de fois le tas qu'il y a de grains à la base en déplaçant à chaque fois la position du capteur d'une distance équivalente à la taille d'un grain. Ces expérimentations étant très coûteuses en temps, en équipements et en moyens humains, les chercheurs trouvent dans la simulation numérique un partenaire indispensable pour la réalisation de ces expérimentations dans un environnement virtuel.

Cependant, des chercheurs ont aussi proposé un modèle théorique pour caractériser la réponse du milieu à une telle surcharge.

7.2.1 Un modèle théorique

Philippe Claudin propose un modèle analytique reliant les propriétés propagatives et diffusives d'un empilement sous condition de faible désordre : le modèle BCC (Bouchaud, Claudin et Clément) [Claudin 1999].

Philippe Claudin a trouvé cette relation après de longs calculs de développements limités jusqu'au second ordre. Cette relation n'est valable que pour un faible désordre car au-delà, les termes issus du développement limité pour des ordres supérieurs à deux ne sont plus négligeables.

Pour simplifier cette relation, nous pouvons écrire que le cœfficient de propagation au carré varie linéairement avec le cœfficient de diffusion, ce qui s'écrit :

$$C^2 = -kD, \text{avec } k > 0 \tag{7.1}$$

Ainsi, lorsque la diffusion est nulle, le cœfficient de propagation est maximal et a pour valeur la tangente de l'angle de contact entre les grains par rapport à la verticale. Dans notre cas, la géométrie de l'empilement étant canonique, cette valeur est l'angle caractéristique de notre réseau régulier, i.e. : ${C_0}^2 = \tan(\frac{\pi}{6})^2 = \frac{1}{3}$.

Le modèle sur lequel s'appuie cette théorie est très simple assez générique. Claudin représente les lignes de forces par des rayons lumineux que des défauts dans le milieu vont séparer et dévier de leurs trajectoires.

La géométrie de notre empilement étant parfaitement régulière, le seul paramètre de désordre dans le modèle de Clément et d'Eloy [Eloy et Clément 1997] est la valeur de frottement μ. Une grande valeur de frottement autorisera une redirection plus importante de la force subie par un grain. Ainsi, chaque grain est un défaut potentiel comme décrit dans le modèle de Claudin.

7.2.2 Un problème ouvert

Plusieurs questions restent ouvertes concernant l'adéquation de ce modèle mathématique avec la réalité physique. La première question qui se pose est d'abord de savoir si le comportement théorique est effectivement observé dans les expérimentations réelles.

Une deuxième question concerne l'extrapolation de la théorie pour des désordres plus importants, car au-delà d'une valeur de frottement de 0,5 environ, la théorie ne s'applique plus.

Enfin, la troisième question, sur laquelle se concentrent aujourd'hui un grand nombre de chercheurs de la communauté «milieux granulaires secs» est de caractériser la transition qui s'opère lorsque l'on passe d'un milieu sans aucun désordre (frottement nul) à un milieu à faible désordre. Ils cherchent donc à caractériser le comportement du système face à un désordre «élémentaire» (le plus petit désordre qu'il soit possible de générer).

7.3 La modélisation de la fonction de Green dans GranuLab

Pour des raisons de calendrier, toutes les expérimentations n'ont pas pu être réalisées avec l'interface de GranuLab qui n'était pas disponible à l'époque où nous avons commencé ces expérimentations. Cependant, nous avons réalisé toutes les expérimentations avec les composants de GranuLab en mimant, à l'aide de scripts, le flux de données qui est aujourd'hui piloté par le système.

Nous décrivons les conditions limites de la réalisation de nos expérimentations virtuelles et le modèle de régression des données que nous avons mis en œuvre dans GranuLab.

7.3.1 La description des conditions limites

Les conditions limites de nos expérimentations concernent des empilements bidimensionnels du type «boulets de canon» de grains durs et réguliers. Il n'y a pas de parois, i.e. l'empilement est cylindrique. Une surcharge est appliquée sur un des grains de la première couche. Afin que ce signal ne se diffuse trop rapidement dans le milieu, nous considérons une valeur de gravité nulle. Il faut donc se représenter un «mur de grains» posé à l'horizontale sur une table. Ainsi, la gravité n'entre pas en concurrence avec le signal et ce dernier se propage plus loin dans le milieu. Mais, à gravité nulle et sans confinement, aucune charge ne traverse le milieu. Pour notre étude, tous les grains de la première couche subissent donc un appui unitaire, sauf le grain du milieu qui subit une charge plus importante.

De plus, le tas étant cylindrique, pour ne pas obtenir des effets de recouvrement du signal, nous considérons des tas beaucoup plus grands en largeur qu'en hauteur. La taille des tas étudiés est de 501 grains de large par 200 grains de haut.

Reste à donner une valeur de l'appui initial sur le grain qui se trouve au milieu de la première couche. Nous avions d'abord essayé une valeur d'appui de 10 unités. Malheureusement, après plusieurs semaines de travail, nous nous sommes aperçus que cette valeur d'appui était trop importante. En effet, l'ordre de grandeur des valeurs des forces de contact était beaucoup plus grand que celui des forces environnantes, pour les premières couches. Il fallait donc attendre que le signal s'infléchisse suffisamment pour qu'il soit du même ordre de grandeur que celui des grains avoisinants. A ce moment-là, une deuxième phase de diffusion s'opérait, puis peut-être un début de troisième. Le problème réside dans la profondeur des tas à laquelle il fallait arriver pour commencer à voir des choses intéressantes du point de vue des physiciens de l'équipe. Pour mémoire, plus le tas est profond et plus l'espace des solutions est important et donc plus il faut générer de réalisations pour espérer obtenir une statistique fiable. Nous avons donc décidé, pour simplifier les calculs mais surtout l'analyse, de fixer la valeur de l'appui à 2.

Dans la section suivante, nous décrivons le modèle de régression des données issues de notre solveur GRANUSOLVE et correspondant aux conditions limites que nous venons d'exposer.

7.3.2 Le modèle de régression pour l'étude de la fonction de Green

Dans cette section, nous explicitons le processus de régression des données issues du solveur tel qu'il a été défini par les physiciens de l'équipe. Ce processus, illustré par la figure 7.2, consiste à inférer, par régression sur les données brutes, des propriétés macroscopiques du milieu.

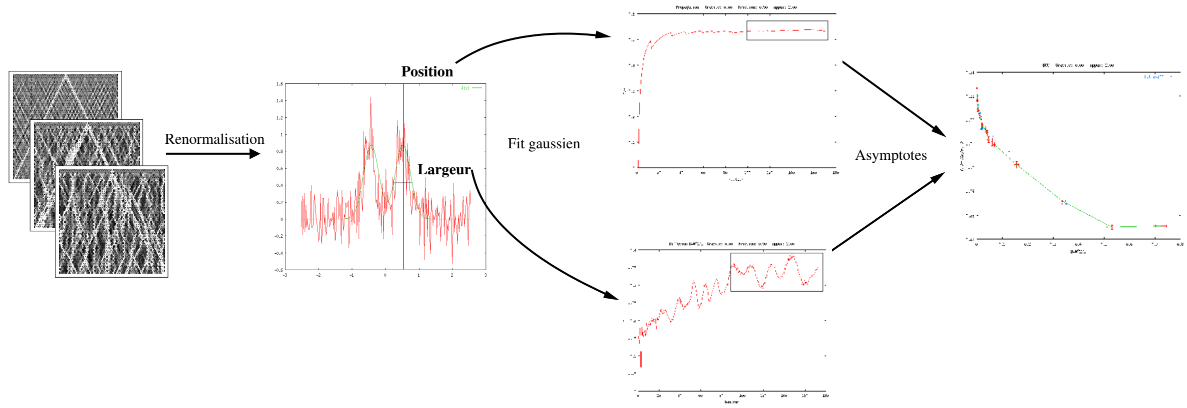


FIG. 7.2 – Processus de régression ou comment passer des données brutes à une courbe caractérisant le milieu ?

La première étape permet, à partir d'un tas solution issu de notre solveur, de renormaliser la réponse du système pour chacune des couches de l'empilement. De plus, cette renormalisation autorise la superposition de courbes de réponse de la fonction de Green quelles que soient les conditions limites initiales de génération de l'empilement. Une série d'opérateurs, décrits dans la section 7.4.1, permettent d'effectuer cette première tâche. Ensuite, afin de réduire le bruit et de dégager le signal, ces courbes renormalisées sont moyennées pour plusieurs milliers de solutions. Un opérateur *moyenne de tas* a été implémenté dans ce but.

La deuxième phase consiste ensuite à extraire les positions et les largeurs des pics de la fonction de réponse pour chaque couche. Cet opérateur a été implémenté sous la forme d'un programme C++ optimisant l'adéquation entre les données et une double gaussienne (cf. section 7.4.2).

Enfin, la dernière phase du processus de régression consiste à calculer les valeurs asymptotiques des positions et des largeurs des pics (cf. 7.4.4). Cette fois, un opérateur basé sur une feuille de styles XSLT a été proposé. Cet opérateur permet d'extraire des descripteurs du tas par régression sur les descripteurs des couches.

Ces trois étapes sont détaillées dans la suite de ce chapitre.

7.4 Les expérimentations menées autour de la fonction de Green

Dans cette section, nous détaillons le schéma d'expérimentations que nous avons construit pour l'étude la fonction de green. Chaque opérateur de transformation est présenté, décrit et souvent illustré d'un exemple. Pour des raisons de lisibilité, nous ne donnons que des vues partielles des composants de transformation implémentés. Plusieurs exemples de traitement des données XML par feuilles de style XSLT sont détaillés dans l'annexe B.7.

Une expérimentation se déroule en trois phases. Dans la section 7.4.1 suivante, nous décrivons le processus de renormalisation de la charge verticale des couches dans un tas. L'extraction de descripteurs au niveau de chaque couche est détaillée dans la section 7.4.2. Ensuite, dans la section 7.4.3, nous traçons le graphe de ces descripteurs pour toutes les couches de l'empilement Enfin, dans la dernière section 7.4.4, des descripteurs du tas, dans sa globalité, sont extraits à partir des valeurs asymptotiques des descripteurs des couches.

7.4.1 Le processus de renormalisation

Ce processus est important car il permet de réaliser la première partie de la transition entre les données brutes fournies par le composant GranuSolve et une courbe qui caractérise la diffusion et la propagation dans le milieu (cf Fig. 7.2).

L'extraction de l'information : la *charge verticale renormalisée du grain*

Le première opération à appliquer sur le tas est d'extraire l'information de la charge verticale de chaque grain à partir de ses forces de contact. Cette valeur de charge verticale du grain est renormalisée par la moyenne des charges verticales des grains de la couche.

$$charge\_verticale\_normalisee = \frac{charge\_verticale}{< charge\_verticale\_couche >} \tag{7.2}$$

Ainsi, un grain de charge verticale renormalisée à 0,86 signifie que ce grain possède une charge verticale un peu inférieure à la moyenne des grains de la couche (cf. Fig. 7.3[3]). A l'inverse, un grain avec une valeur de charge renormalisée de 2,78 possède une charge verticale 2,78 fois supérieure à la moyenne des grains de la couche.

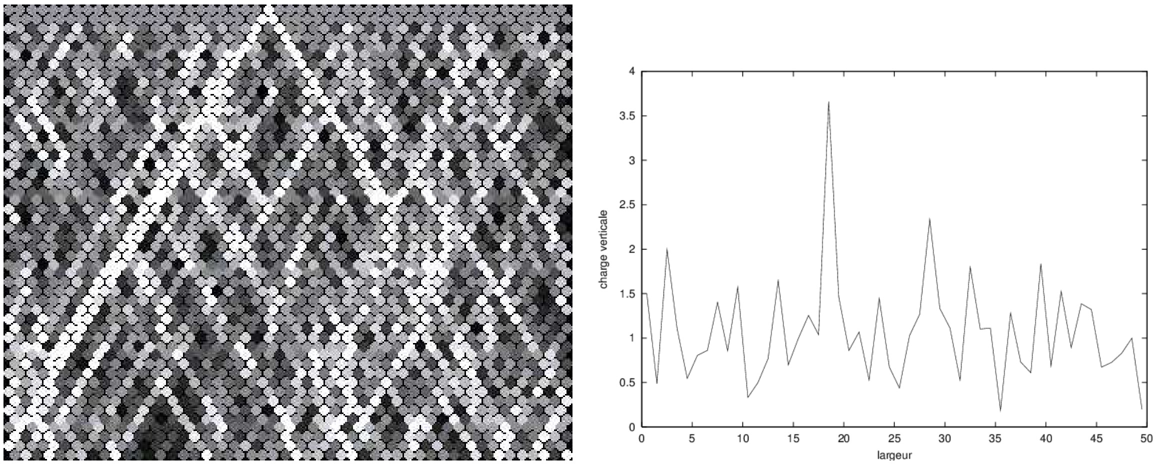


FIG. 7.3 – Visualisation des charges verticales renormalisées et de la réponse de Green a) Visualisation des charges verticales de chacun des grains (en blanc, les grains qui supportent une charge supérieure à la moyenne) b) Visualisation de la répartition de la charge pour la couche numéro 10.

La suppression du bruit de fond

Dans cette deuxième opération, l'idée est de soustraire au signal la valeur moyenne des charges verticales des grains de la couche que l'on aurait eu sans appui (le bruit de fond). Nous obtenons ainsi un bruit résiduel pour les grains en-dehors de la région du signal qui doit être centré en zéro sur l'axe des ordonnées (cf. Fig. 7.4).

La moyenne des réalisations

Nous moyennons ensuite les réalisations de plusieurs tas afin de réduire le bruit et espérer ainsi dégager plus facilement le signal du bruit. Il est important de remarquer que plus la valeur de frottement est grande et plus l'espace des possibilités est important

[3]pour des raisons de taille et de lisibilité de l'image, nous avons préféré illustrer notre propos avec une image et une courbe issues d'un tas 50x50 plutôt que de 501x200

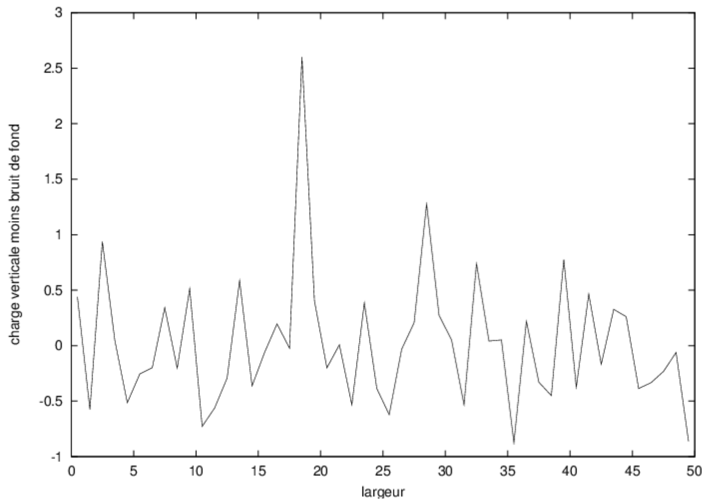


FIG. 7.4 – Visualisation de la réponse de Green sans le bruit de fond
Le bruit de fond a été soustrait au signal afin de centrer la courbe en zéro sur l'axe des abscisses.

et donc plus il faut générer de tas pour obtenir un bruit suffisamment petit pour être considéré comme négligeable dans la suite du processus (cf. Fig. 7.5). A cette étape, le bruit (en-dehors de la région où un signal se dégage) doit être centré en zéro et relativement faible devant le signal pour pouvoir être négligé dans la suite du processus.

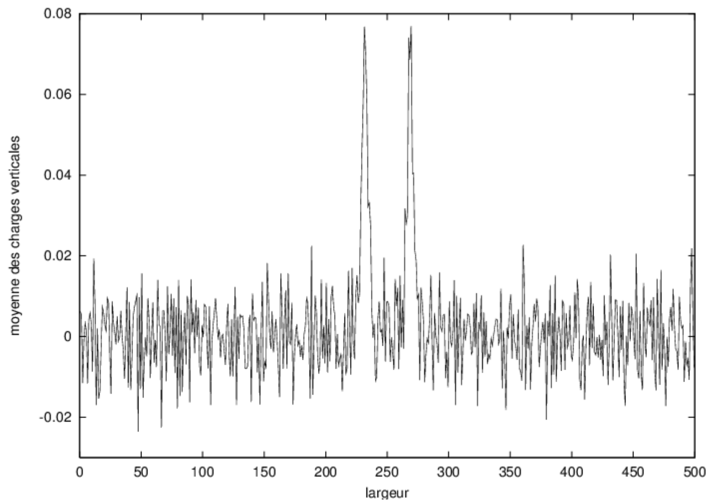


FIG. 7.5 – Visualisation de la réponse de Green pour 3461 réalisations moyennées
Cette courbe représente la moyenne de 3461 courbes comme la précédente. Il s'agit d'un tas 501x200 avec une valeur de frottement $\mu = 0,50$ et pour la couche numéro 40.

La renormalisation en x et en y

La courbe précédente est ensuite renormalisée de façon à obtenir une «courbe universelle». Ainsi, il sera possible de superposer les courbes issues de tas aux conditions limites ou de couches différentes pour pouvoir les comparer. Les quatre opérateurs suivants sont appliqués successivement sur les données moyennées.

1. $x_1 = x - largeur/2$ - La courbe moyennée est renormalisée en x une première fois afin de la recentrer en zéro sur l'axe des abscisses. Ainsi, quelle que soit la largeur des tas considérés, le point d'appui sera toujours centré en zéro (cf. Fig. 7.6).

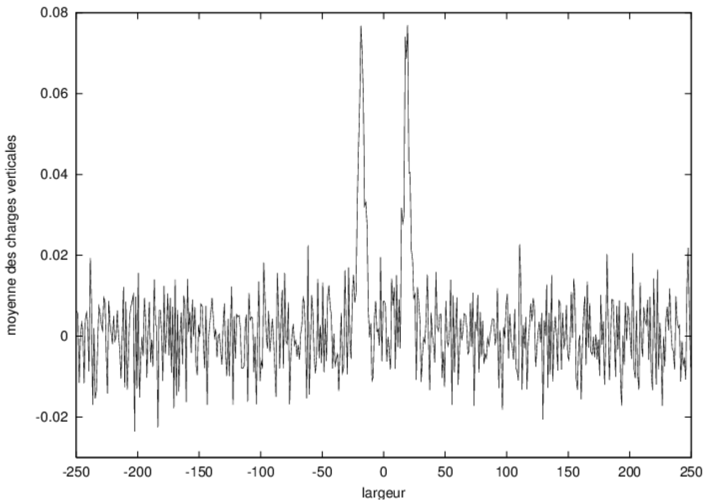


Fig. 7.6 – Visualisation de la réponse de Green, moyennée et recentrée en $y = 0$
Cette courbe est simplement issue de la translation horizontale de la courbe précédente.

2. $y_1 = y/A$ - Le signal est renormalisé, en y, par la valeur d'appui, notée A. Ainsi, quelle que soit la valeur d'appui sur les tas, les courbes auront toutes la même surface renormalisée à un. La hauteur des pics ne dépend plus de cette valeur d'appui (cf. Fig. 7.7).

3. $y_2 = y_1 * z$ - Le signal de chaque couche est multiplié par sa profondeur z. De cette façon, nous nous affranchissons de la profondeur à laquelle la réponse de Green est observée. Il sera donc possible de superposer et comparer successivement les courbes de réponse pour toutes les couches de l'empilement (cf. Fig. 7.8).

4. $x_2 = x_1/z$ - Pour conserver une surface unitaire sous la courbe renormalisée par la profondeur z de la couche, les valeurs en abscisse sont divisées par la profondeur z. De plus, la position du pic doit alors rester constante avec la profondeur dans le cas d'un comportement entièrement propagatif (cas d'une valeur de frottement nulle).

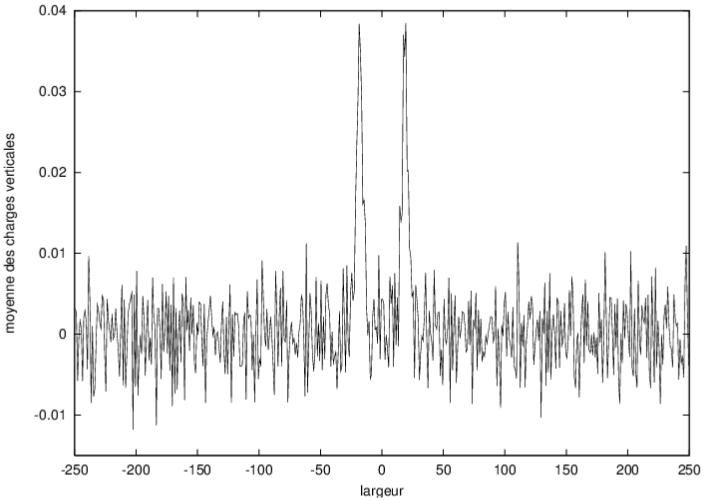


FIG. 7.7 – Visualisation de la réponse de Green, renormalisée par la valeur d'appui
La surface de la courbe est donc unitaire, quelle que soit la valeur de l'appui initial.

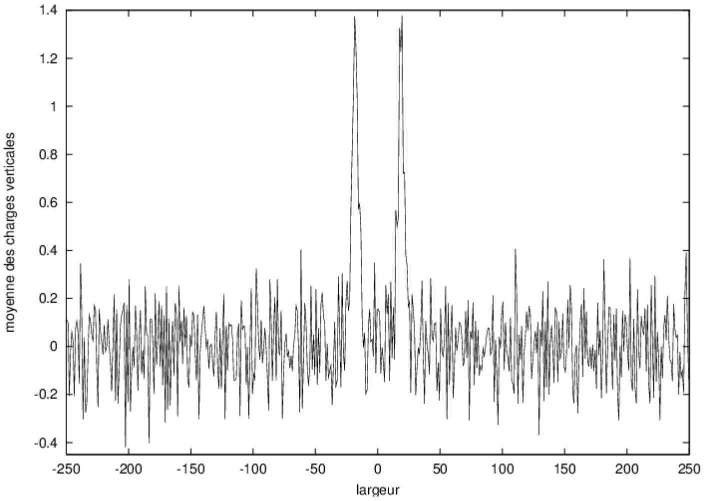


FIG. 7.8 – Visualisation de la réponse de Green, renormalisée par la valeur d'appui et par la profondeur de la couche
Le signal a été multiplié par la profondeur z de l'empilement. Les courbes de réponse de couches distinctes peuvent être comparées.

Des animations vidéo[4] ont été réalisées afin de visualiser l'évolution de la hauteur et de la position des pics en fonction de la profondeur dans le tas (à raison d'une image par couche) (cf. Fig. 7.9).

A la fin de cette phase de renormalisation, les courbes de réponse de Green peuvent être superposées et comparées, quelles que soient les conditions limites appliquées sur

[4]cf. le site oueb de GRANULAB : http ://terk.lip6.fr/~breton/granulab pour des animations de ces courbes

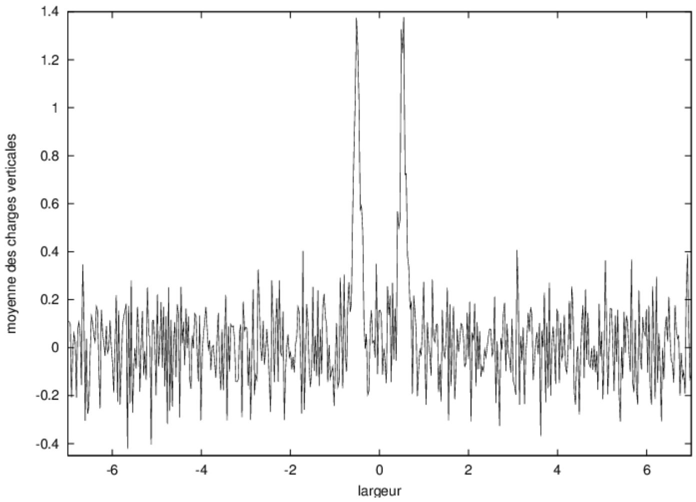


FIG. 7.9 – Visualisation de la réponse de Green après le processus de renormalisation En réalisant une animation à raison d'une image par couche, il est possible de visualiser l'évolution de l'écartement relatif des deux pics et de leurs tailles respectives.

l'empilement et la profondeur des couches étudiées.

7.4.2 L'adéquation des données avec une double gaussienne symétrique à deux degrés de liberté

L'étape suivante consiste à estimer la position et la largeur des pics en fonction de la profondeur. Pour la position du pic, calculer la position du maxima de la courbe en présence de bruit donne de très mauvais résultats. De même, le calcul de la largeur à mi-hauteur du pic centré autour de la valeur calculée comme décrit ci-dessus donne des résultats encore plus approximatifs.

Aux vues de la forme générale des doubles pics, nous avons décidé d'estimer les paramètres par un fit de la courbe avec une double gaussienne symétrique à deux degrés de liberté : g_μ pour la position et g_σ pour la largeur du pic. Nous comparons donc l'adéquation entre la distribution des données et la distribution théorique suivante :

$$G(g_\mu, g_\sigma) = \frac{1}{2\sqrt{2\pi {g_\sigma}^2}}(\exp\frac{(x+g_\mu)^2}{2{g_\sigma}^2} + \exp\frac{(x-g_\mu)^2}{2{g_\sigma}^2}) \tag{7.3}$$

Le problème du pic unique pour les premières couches est résolu par le fait que la double gaussienne peut être considérée comme la somme de deux gaussiennes centrées au même point.

Un premier algorithme d'estimation des paramètres g_μ et g_σ a été écrit et validé visuellement par les physiciens de l'équipe (cf. Fig. 7.10)

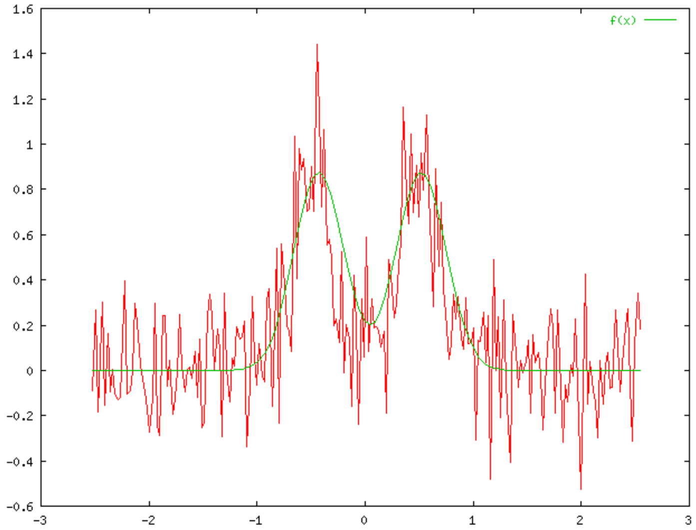


FIG. 7.10 – Validation visuelle du fit de la réponse de Green avec une double gaussienne

Seulement, le fait d'exécuter la procédure de fit pour toutes les couches du tas et pour toutes les positions et largeurs possibles des pics sur chaque couche demandait presque une journée de calcul pour avoir une estimation de ces deux paramètres pour un tas complet (cf Algorithme 7.1).

ALGORITHME **7.1** Estimation des paramètres g_μ et g_σ

Pour toutes les couches i du tas **Faire**
  Pour toutes les positions possibles g_μ du pic de la partie droite de la courbe **Faire**
    erreur $\leftarrow +\infty$
    Pour toutes les largeurs g_σ possibles des pics **Faire**
      erreur $\leftarrow$ calculer_fit_double_gaussienne(g_μ,g_σ)
    Fin Pour
    g_σ_*opt* $\leftarrow$ la meilleure largeur g_σ pour la position g_μ courante
  Fin Pour
  g_μ_*opt* $\leftarrow$ la meilleure position g_μ associée à la meilleure largeur g_σ_*opt*
Fin Pour

Cette première version calcule de façon correcte la position et la largeur des pics. Cet opérateur a d'ailleurs été validé par les physiciens. Le seul problème de cet algorithme réside dans sa complexité quadratique avec la largeur du tas. En effet, pour de grands tas, l'algorithme essaie de trouver la position optimale du pic parmi toutes les positions possibles pour la partie droite de la couche. Ensuite, pour chacune de ces positions, l'algorithme teste toutes les largeurs possibles du pic. Notre algorithme couvre entièrement l'espace de recherche et nous savons que certaines parties de cet espace ne contiennent pas la solution optimale. Nous avons donc proposé une heuristique afin de restreindre l'espace de recherche et donc la complexité de notre algorithme. Le gain en complexité à ce niveau est important car c'est le maillon le plus coûteux de notre chaîne de traitement (hormis le solveur). De plus, une complexité quadratique nous interdit l'estimation des paramètres g_μ et g_σ pour toutes les couches d'un tas mais limite cette recherche à quatre ou cinq couches choisies par l'expert. Cette limitation est très gênante pour pouvoir caractériser, par la suite, l'évolution de la largeur et de la position des pics en fonction de la profondeur de l'empilement.

Nous avons donc proposé une deuxième version de notre algorithme d'estimation basée sur l'hypothèse suivante :

HYPOTHÈSE 7.1 **Hypothèse de continuité**
La position (resp. largeur) du pic à une profondeur z donnée ne doit pas s'être écartée de plus de n grains autour de la position (resp. largeur) du pic à la couche précédente.

En prenant une valeur de l'ordre de cinq grains autour de la position du pic de la couche précédente, l'algorithme ne parcourt plus que dix possibilités pour la position du pic de la partie droite de la courbe au lieu de deux cent cinquante pour un tas de cinq cents grains de largeur. Et, de même pour la largeur du pic, l'algorithme couvre cinquante fois moins de valeurs que précédemment. Avec ce gain, d'un facteur 1250 pour cet exemple, nous avons obtenu les mêmes résultats en des temps tout à fait raisonnables (de l'ordre de quatre minutes). Ainsi, nous avons pu, par la suite, tracer la courbe de l'évolution de la largeur et de la position des pics en fonction de la profondeur pour toutes les couches de l'empilement.

7.4.3 Le graphe des positions et des largeurs des pics en fonction de la profondeur

Une fois les positions et les largeurs des pics estimées par le fit précédent, nous traçons l'évolution de ces paramètres en fonction de la profondeur des couches. A condition d'effectuer des mesures sur des tas suffisamment profonds, nous devrions obtenir un régime quasi-stationnaire pour les couches les plus profondes et en déduire ainsi les cœfficients de propagation et de diffusion du milieu (cf. Fig. 7.11).

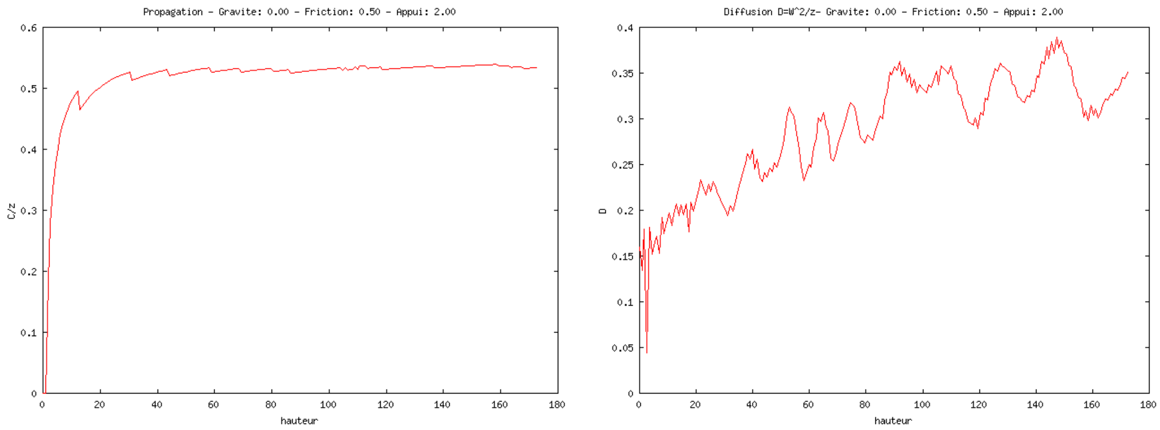


FIG. 7.11 – Visualisation de la position du pic et de sa largeur en fonction de la profondeur Chaque point représente l'estimation d'un paramètre (g_μ et g_σ) pour une couche du tas.

7.4.4 Le calcul des valeurs asymptotiques de la position et de la largeur des pics

L'estimation des valeurs asymptotiques de ces courbes est réalisée par le calcul de la moyenne des valeurs à partir d'une certaine profondeur dans le tas, donnée par l'utilisateur. L'écart-type des valeurs par rapport à la moyenne calculée nous donne, de plus, des barres d'erreur pour cette valeur asymptotique. Depuis le début de notre exposé concernant le processus de renormalisation et d'extraction de grandeurs macroscopiques, nous avons toujours considéré une valeur de frottement $\mu = 0,50$. Nous obtenons donc, à cette étape, un unique couple de valeurs pour les cœfficients de propagation et de diffusion, notés respectivement C et D. Il n'est pas possible d'effectuer de régression sur les données plus avant et une itération de ce même processus est nécessaire pour obtenir d'autres points et tracer une nouvelle courbe. Ces points sont indispensables si nous souhaitons vérifier que les données issues de nos simulations corroborent la théorie BCC. Pour obtenir ces points, il faut réitérer toutes les étapes que nous venons de décrire, depuis la section 7.4.1, pour des valeurs de frottement différentes.

Les courbes suivantes ont été obtenues par l'application du processus de régression sur les données pour différentes valeurs de frottement (cf. Fig. 7.12).

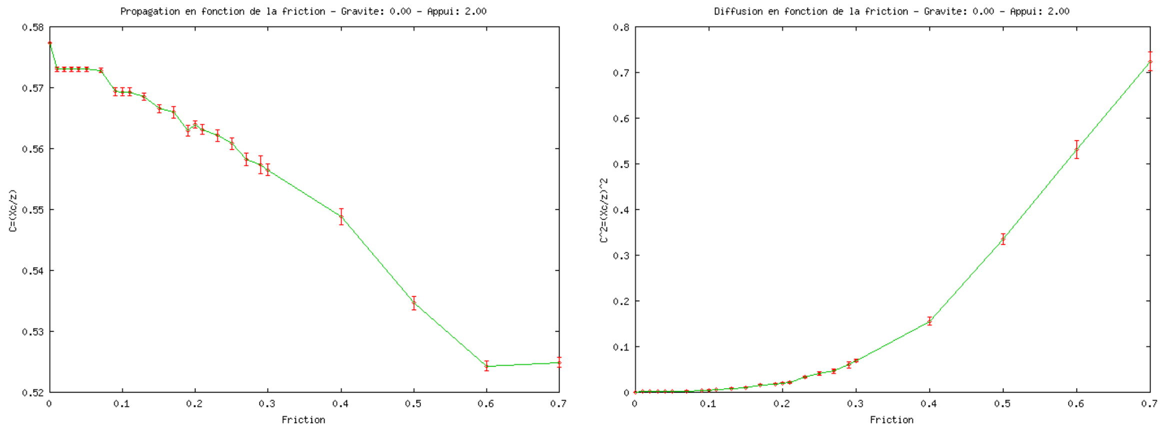


Fig. 7.12 – Visualisation des valeurs asymptotiques de la position du pic et de sa largeur en fonction de la valeur de frottement

Chaque point représente l'estimation de la valeur asymptotique de la position (resp. largeur) du pic tracée sur la courbe précédente. Il y a donc autant de points que de valeurs de frottement.

7.5 Les résultats obtenus pour l'étude de la fonction de Green

A l'issue du processus de régression décrit dans la section précédente, nous avons extrait suffisamment de valeurs des descripteurs macroscopiques d'un tas pour tracer la courbe de la théorie BCC. Nous comparons ensuite nos résultats avec ceux de la théorie.

7.5.1 La comparaison des simulations avec la théorie BCC

La théorie de Bouchaud et Claudin prédit que le carré du cœfficient de propagation doit varier de façon strictement linéaire avec le cœfficient de diffusion. Seulement, cette théorie est issue de calculs complexes menés à la main par Philippe Claudin qui a donc effectué des développements limités jusqu'à l'ordre 2. Les imprécisions issues des ordres supérieurs à 2 sont tout à fait négligeables pour les faibles désordres (le frottement pour notre problème), mais deviennent beaucoup plus importantes quand le désordre devient plus grand (le cas des fortes valeurs de frottement). Nous avons donc tracé la courbe $C^2 = f(D)$ qui, d'après cette théorie doit être une droite de pente strictement négative, au moins pour les faibles valeurs de frottement (cf. Fig. 7.13).

La courbe que nous avons tracé possède effectivement un régime linéaire avec une pente approximative de -0,1. Le dernier point en bas à droite n'est pas significatif car nous manquons de réalisations pour extraire des valeurs fiables des cœfficients de propagation et de diffusion. Ce point correspond à une valeur de frottement $\mu = 0,70$ et manque de

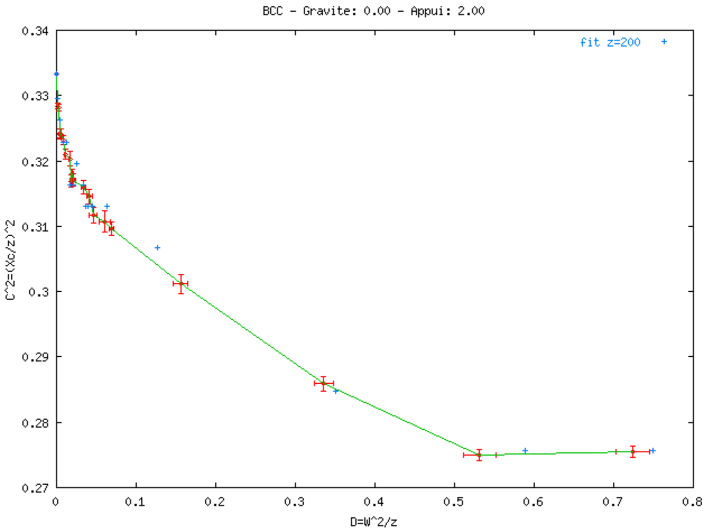


FIG. 7.13 – Visualisation de la courbe correspondant à la théorie BCC
Cette courbe représente les couples (C^2, D) pour des valeurs de frottement de 0,1 à 0,7.

précision malgré 26.000 réalisations moyennées.

Mais, les points proches du zéro, en abscisse, ne forment pas vraiment une droite. Nous avons donc multiplié le nombre de points dans cette zone afin d'obtenir la forme générale de la courbe pour un faible désordre mécanique (réponse du système à un désordre «élémentaire»).

La deuxième phase, pour des valeurs de frottement comprises entre 0,3 et 0,6, corrobore la théorie BCC de Claudin. La première phase, pour de très faibles désordres, semble énoncer le fait que, pour des tas simulés basés sur le modèle de Clément et d'Eloy, le cœfficient de propagation décroît beaucoup plus vite que le cœfficient de diffusion. Cela signifie que la position des deux pics se recentre assez rapidement, dès qu'il y a un peu de désordre dans le milieu. Mais, dans le même temps, les pics ne s'élargissent que très peu.

7.5.2 Quelques chiffres

Ces expérimentations ont débuté il y a environ huit mois. Durant cette période, nous avons eu beaucoup d'interactions avec les physiciens de l'équipe pour acquérir les connaissances et les opérateurs dont nous avions besoin pour mener à bien ces expérimentations. Chacun des opérateurs décrits dans ce chapitre a fait l'objet d'une validation systématique par les physiciens de l'équipe avant d'aller plus avant dans la mise en œuvre du processus.

Le graphique (cf. Fig. 7.13) montre la courbe de Claudin achevée. Pour réaliser cette courbe, il nous aura fallu résoudre pas moins de 50.000 tas, l'équivalent de 2 mois de calculs en continu pour un seul ordinateur de type G4 400Mhz. Chaque tas représentant environ 42 Mo sur disque, la totalité des tas résolus représente quelques 2 To[5] de données qui ont été manipulées pour n'obtenir, au final, que quelques points sur cette unique courbe.

7.6 Conclusion

Nous avons décrit, dans ce chapitre, le processus de régression mené sur les données issues d'un solveur de tas. Nous avons illustré ce processus par l'extraction de descripteurs macroscopiques d'un tas : les cœfficients de propagation et de frottement. Ces descripteurs nous ont permis de tracer la fonction issue de la théorie BCC de Claudin [Claudin 1999] et de comparer les résultats de nos simulations basées sur le modèle de Clément et d'Eloy [Eloy et Clément 1997] avec cette théorie mathématique.

La validation de la théorie BCC, pour au moins un intervalle de valeurs de frottement, à l'aide d'expérimentations virtuelles est essentielle car la réalisation de ces expérimentations en laboratoire s'avère impossible. En effet, le nombre de tas et de mesures, la quantité d'informations nécessaire, les erreurs et imperfections du montage rendent impraticable la réalisation complète du processus que nous avons décrit. De plus, l'observation d'une phase non-linéaire pour les faibles valeurs de frottement est un point qui suscite beaucoup d'intérêt dans la communauté des chercheurs en milieux granulaires. Ces deux résultats font l'objet d'un article en cours de préparation pour la revue *Euro. Physics Letters* [Breton *et al.* 2001].

L'utilisation d'un système tel que GranuLab permettant à l'expert de décrire successivement les étapes du processus de recherche s'est révélée très fructueuse. L'extraction des connaissances du domaine au travers des opérateurs de transformation est une condition nécessaire à l'optimisation de ces opérateurs. Mais ces opérateurs ont été difficiles a obtenir des physiciens de l'équipe qui les utilisent quotidiennement sans plus chercher à les optimiser ni même à les configurer pour leurs problèmes. Actuellement, ils utilisent des logiciels de traitement statistique du type tableur (Origin). Ce genre de logiciel propriétaire interdit l'ajout d'heuristiques ou l'optimisation des fonctions statistiques intégrées. En utilisant les outils standards du commerce, ils ont pris l'habitude de transformer systématiquement leurs données pour pouvoir appliquer une fonction ou un opérateur. Parfois, la transformation des données vers le format propriétaire de l'opérateur entraîne une perte ou une dégradation des données. Mais le plus souvent, c'est l'opérateur lui-même qui n'est pas adapté à leur problème. De plus, ces systèmes ne proposent pas de modélisation du processus de traitement des données dans son ensemble. L'intérêt d'un système ouvert

[5]Tera-octets. 1 To = 1024 Go

réside dans la possibilité d'offrir aux experts un ensemble d'opérateurs qui correspondent à leurs besoins réels et à leur approche du problème.

La représentation, dans son ensemble, du processus de régression sur les données issues du solveur nous a permis de valider, auprès des physiciens, chacune des étapes du processus. Cette validation est la condition pour l'utilisation de GranuLab comme d'un environnement d'aide à la découverte appliqué au domaine de la physique des milieux granulaires. De plus, la mémorisation et la vue globale des expérimentations est essentielle pour la reproductibilité des expérimentations mais aussi pour communiquer et expliquer ses travaux à d'autres chercheurs. En effet, une courbe ou un résultat expérimental n'a de sens que dans le cadre détaillé des conditions limites, des opérateurs de transformations appliqués, etc., qui ont été utilisés pour arriver à ce résultat.

GranuLab, tel que nous l'avons implémenté, est très facilement extensible à d'autres domaines que la physique des milieux granulaires. En effet, le système repose sur une grammaire XML de l'étude et une autre pour les expérimentations. Ces deux grammaires sont très facilement extensibles à la prise en charge d'autres objets d'études et de composants.

Enfin, maintenant que nous disposons d'un ensemble relativement complet d'opérateurs pour le traitement des données granulaires, les physiciens de l'équipe peuvent commencer à l'utiliser pleinement et quotidiennement pour leurs recherches. Nous espérons que cet environnement leur sera un allié quotidien et indispensable pour leurs travaux.

Conclusion et perspectives

Au cours de cette thèse, nous avons présenté GranuLab : un système d'aide à la découverte scientifique appliqué à la physique des milieux granulaires. Cet environnement permet à un expert du domaine de réaliser une série d'expérimentations virtuelles basées sur les solutions du solveur GranuSolve. Ce projet a reçu le soutien du CNRS sous la forme d'une «Jeune Equipe» interdisciplinaire regroupant les physiciens du Laboratoire des Milieux Désordonnés et Hétérogènes de Paris 6, Jean-Daniel Zucker, mon directeur de thèse et moi-même.

Dans les sections suivantes, les résultats, les limitations et les perspectives pour les trois domaines abordés que sont l'intelligence artificielle, les systèmes multi-agents et la physique des milieux granulaires sont résumés.

Contributions

Intelligence Artificielle

Le système GranuLab permet de représenter la partie visible du processus de recherche d'un expert en physique des milieux granulaires. Ce système s'inscrit dans une nouvelle génération de systèmes d'aide à la découverte scientifique qui autorisent un haut niveau d'interaction entre l'expert et la machine. GranuLab s'inspire de l'architecture des systèmes de fouille de données de troisième génération [Khabaza et Shearer 1995, Potts 1997] avec l'ajout d'une étape supplémentaire : la génération de données virtuelles à partir d'une théorie du domaine.

Les choix technologiques effectués au cours de cette thèse se sont révélés judicieux pour la représentation et l'opérationnalisation du processus de recherche d'un physicien. Le langage XML et les technologies associées nous ont permis de représenter à l'aide d'un format «ouvert», de façon simple et uniforme, les connaissances du domaine, i.e. les objets d'études dans GranuLab.

Les composants logiciels ont permis une modélisation simple du processus de régression effectué sur les données issues d'un solveur de tas. Nous avons ainsi représenté les opérateurs de transformation des données par un ensemble restreint de composants logiciels élémentaires. Le modèle de composant traitant des flux de documents XML associé aux technologies de ce langage ont réduit le nombre de composants à implémenter. Ainsi, les six types de composants implémentés couvrent l'ensemble des opérateurs dont nous avions besoin dans notre environnement d'aide à la découverte. De plus, ces composants utilisent le plus souvent les librairies existantes pour traiter des documents XML. Notre système bénéficie donc des fonctionnalités nouvelles et des progrès réalisés par les concepteurs de ces librairies. L'évolution du système est facilité par la simple mise à jour des librairies utilisées.

Cette architecture componentielle associée à XML est aujourd'hui vue par les physiciens comme une occasion unique de regrouper les différents algorithmes de résolution qui, pour la plupart, ne sont utilisables que par leurs concepteurs. La définition de standards de représentation est aussi un premier pas vers la reproductibilité, pour les autres chercheurs de la communauté, des expérimentations faites en simulation, garante de la démarche scientifique des physiciens.

Systèmes multi-agents

Du point de vue de l'Intelligence Artificielle Distribuée, GranuSolve représente une application de l'algorithme d'Eco-Résolution supportant un très grand nombre d'agents. L'idée de séparer la résolution des agents de la résolution du système multi-agent dans son ensemble (ou de groupes d'agents) nous a permis, en relation avec les physiciens de l'équipe, de mettre des connaissances du domaine dans notre système. L'étude systématique des heuristiques montre l'importance du choix de la stratégie pour l'Eco-Résolution. Ainsi, pour des conditions limites différentes de celles de notre modèle, par exemple avec des parois et sans gravité, une autre heuristique pourrait se révéler meilleure que celle que nous avons choisie.

De plus, l'interface de visualisation que nous avons implémentée nous a beaucoup aidé à comprendre la résolution collective du système et à imaginer de nouvelles heuristiques pour l'Eco-Résolution. Ce fut aussi un outil pédagogique efficace vis-à-vis des physiciens de l'équipe à qui nous avons pu expliquer les concepts des systèmes multi-agents très simplement en visualisant la résolution d'empilements.

Les performances de GranuSolve en temps et tailles de tas résolus (1 heure pour 1 million de grains) sont de quelques ordres de grandeurs supérieures à celles des algorithmes classiques employés jusqu'à présent par les physiciens. Nous avons également mis en place une batterie de tests numériques pour valider les solutions de GranuSolve afin que les physiciens de l'équipe aient une entière confiance dans notre solveur. De plus, les équations sont toutes résolues avec une précision de l'ordre de 10^{-6}, ce qui donne une confiance supplémentaire dans les solutions générées.

Grâce à ces performances en temps et en taille de tas, nous avons pu mener une véritable étude statistique de la fonction réponse d'un empilement («fonction de Green»).

Physique des milieux granulaires

La première contribution importante, pour la communauté de chercheurs en physique des milieux granulaires, est la réalisation d'un solveur de tas de sable à l'équilibre statique qui permette de résoudre des tas de grandes tailles en des temps raisonnables. Ces performances nous ont permis de mener une étude statistique de la fonction réponse (fonction de Green) d'un milieu granulaire.

Notre deuxième contribution concerne les résultats des expérimentations menées à l'aide de notre solveur dans l'environnement GranuLab. En effet, après un long processus de régression sur les données et de renormalisation, nous avons tracé la courbe caractérisant les caractères propagatif et diffusif du milieu (cf. Fig. 1). La théorie mathématique de Philippe Claudin [Claudin 1999] prédit une décroissance linéaire du cœfficient de propagation au carré en fonction du cœfficient de diffusion du milieu. Nos expérimentations ont permis de valider cette théorie pour une intervalle de valeurs de frottements (typiquement des valeurs de frottement entre 0,2 et 0,6). La validation de la théorie BCC, pour au moins un intervalle de valeurs de frottement, à l'aide d'expérimentations virtuelles est essentielle car la réalisation de ces expérimentations en laboratoire s'avère impossible. En effet, le nombre de tas et de mesures, la quantité d'informations nécessaire, les erreurs et imperfections du montage rendent impraticable la réalisation complète du processus que nous avons décrit.

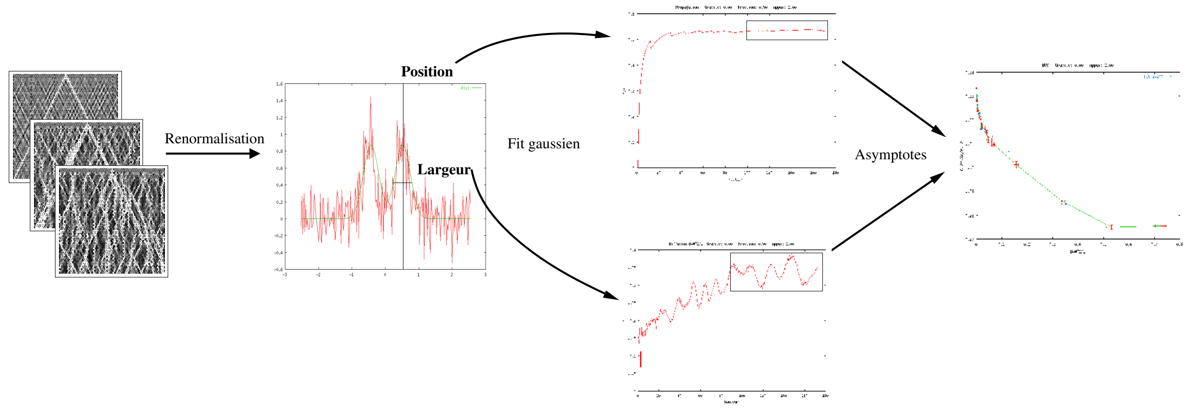


FIG. 1 – Processus de régression ou comment passer des données brutes à une courbe caractérisant le milieu ?

A partir de cette courbe, nous avons tiré un autre résultat intéressant. L'observation d'une phase non-linéaire pour les faibles valeurs de frottement est un point qui suscite beaucoup d'intérêt dans la communauté des chercheurs des milieux granulaires. Ils cherchent actuellement à caractériser le comportement du système face à un désordre «élémentaire» (le plus petit désordre qu'il soit possible de générer). Un article pour la revue

Euro. Physics Letters, illustrant ces résultats, est en cours de préparation [Breton *et al.* 2001].

Perspectives

Intelligence Artificielle

Le système GranuLab, tel que nous l'avons implémenté, est très facilement extensible à d'autres domaines que la physique des milieux granulaires. En effet, le système repose sur une grammaire XML de l'étude et une autre pour les expérimentations. Ces deux grammaires sont très facilement extensibles à la prise en charge d'autres objets d'études et de composants. La seule limitation de l'application de l'architecture de GranuLab à un autre domaine est l'utilisation du langage XML. Il est prévu, à très court terme, d'étendre l'utilisation de GranuLab dans le cadre du projet ObeLinks [Bournaud *et al.* 2000]. Le but de ce projet est d'extraire des régularités à partir de bases de données concernant le génome humain et dans le cas d'une pathologie particulière qu'est l'obésité.

D'autre part, maintenant que nous disposons d'un ensemble relativement complet d'opérateurs pour le traitement des données granulaires, les physiciens de l'équipe peuvent commencer à utiliser GranuLab pleinement et quotidiennement pour leurs recherches. Nous espérons que cet environnement leur sera un allié quotidien et indispensable pour leurs travaux.

Systèmes multi-agents

Nous avons considéré une façon simple de résoudre les équations de l'équilibre d'un grain car notre solveur était basé sur un modèle à contacts fixes. Dans le cas de tas plus réalistes (grains de tailles différentes), il faudra pouvoir résoudre de façon plus générale les équations d'équilibre. Une voie possible est alors de résoudre les équations par des méthodes de résolution de contraintes à domaines réels (CSP). Cet outil de résolution peut n'être utilisé que pour la résolution locale des grains et conserver la méthode d'Eco-Résolution pour la résolution globale du tas.

Une autre perspective concerne les groupes d'agents. Pour notre solveur, nous avions défini deux groupes d'agents : les couches et le tas. Il serait vraiment intéressant d'avoir une conception plus dynamique des groupes afin de pouvoir capturer les phénomènes émergents. Ainsi, des entités intermédiaires telles que les lignes de forces et les voûtes pourraient être manipulées et faire directement l'objet de mesures dans le système.

La troisième voie de recherche qui nous apparaît comme intéressante concerne la programmation d'agents par composants et non plus directement par objets. Nous avons pu constater, au cours de l'implémentation de nos algorithmes, que le comportement collectif de résolution du système multi-agent était très sensible à la moindre modification du code d'un seul type d'agent. Aujourd'hui, il nous semble clair qu'il est nécessaire d'introduire un niveau d'abstraction intermédiaire entre le code objet et la programmation par agents. Ce niveau pourrait être comblé par la technologie des composants logiciels. Le système multi-agent serait plus robuste si l'on pouvait composer le comportement d'un agent à l'aide de briques élémentaires de plus haut niveau d'abstraction que du simple code objet. La lisibilité et la maintenance d'un tel système en serait améliorée.

Enfin, une dernière perspective importante sera d'utiliser un modèle plus réaliste des empilements de grains. Par exemple, il est envisagé d'ajouter des conditions limites particulières (parois, plancher, etc.) pour simuler des systèmes de type silo ou broyeur.

Physique des milieux granulaires

Dans le domaine de la physique des milieux granulaires, GranuLab devra être utilisé plus largement par les physiciens de l'équipe et à terme dépasser les frontières de notre équipe. Dans un premier temps, nous nous limiterons aux chercheurs de notre équipe afin d'obtenir très rapidement des retours sur son utilisation au quotidien avant de proposer une version stable qui sera diffusée plus largement à la communauté des chercheurs dans ce domaine. Il sera alors intéressant d'observer le comportement de ces chercheurs face à un tel système. En effet, peu d'entre eux conçoivent des algorithmes de simulation numérique pour les milieux granulaires. La plupart des chercheurs sont avant tout des expérimentateurs mais aussi des théoriciens. Nous espérons ainsi pouvoir contribuer à une meilleure communication entre les membres de cette communauté de chercheurs.

Une autre perspective sera d'implémenter ou d'encapsuler d'autres algorithmes de résolution dans le système. Ainsi, les comportements macroscopiques des solutions issues d'algorithmes différents pourront être comparées et les biais implicites éventuels détectés.

Bilan

Nous avons essayé de montrer, à travers nos travaux, que l'utilisation d'outils de l'Intelligence Artificielle en interaction avec un chercheur de la physique des milieux granulaires pouvait aider à la découverte scientifique dans ce domaine. Nous avons également cherché à comprendre, à extraire et à représenter les connaissances du domaine aussi bien pour la résolution d'empilements de grains que pour la représentation du processus de recherche de l'expert.

Bibliographie

1 Découverte scientifique

[AAAI 1995] *Working notes of the AAAI Spring Symposium on Systematic Methods of Scientific Discovery*, Stanford, CA, 27-29 March 1995. AAAI Press.

[ABACUS 1986] Le système ABACUS. cf. [Falkenhainer et Michalski 1986], 1986.

[AM 1977] Le système AM. Création de concepts et découverte de lois etconjectures en mathématiques, cf. [Lenat 1977a, Lenat 1982, Lenat 1984a, Lenat et Brown 1984], 1977.

[ASTRA 1998] Le système ASTRA. Découverte de chaînes de réactions en astrophysique nucléaire, cf. [Kocabas et Langley 1998], 1998.

[Bacon 1960] Francis Bacon. *The new organon and related writings.* Liberal Arts Press, New York, 1960.

[BACON 1978] Le système BACON. Découverte automatique de lois symboliques à partir de l'analyse des corrélations entre les données, cf. [Langley 1978, Bradshaw *et al.* 1980, Langley 1981, Langley *et al.* 1983], 1978.

[Boden 1990] Margaret A. Boden. *The creative mind : Myths and mechanisms.* London : Weidenfeld and Nicolson, 1990.

[Bournaud *et al.* 2000] Isabelle Bournaud, Mélanie Courtine et Jean-Daniel Zucker. Abstractions for knowledge organization of relational descriptions. *Lecture Notes in Artificial Intelligence, Symposium of Abstraction, Reformulation and Approximation (SARA'2000)*, 1864 : pp87–106, 2000.

[Box et Hunter 1965] George E. P. Box et William G. Hunter. The experimental study of physical mechanisms. *Technometrics*, 7(1) : 23–42, 1965.

[BR-3 1991] Le système BR-3. Découverte dans le domaine de la physique des particules, cf. [Kocabas 1991], 1991.

[Bradley et Stolle 1996] Elizabeth Bradley et Reinhard Stolle. Automatic construction of accurate models ofphysical systems. *Annals of Mathematics and Artificial Intelligence*, 17(1-2) : 1–28, 1996.

[Bradshaw *et al.* 1980] Gary L. Bradshaw, Pat Langley et Herbert A. Simon. BACON.4 : The discovery of intrinsic properties. In *Proceedings of the Third Biennal Conference of theCanadian Society for Computational Studies ofIntelligence*, pages 19–25, Victoria, BC. 1980.

[Breton *et al.* 2000] Laurent Breton, Jean-Daniel Zucker et Eric Clément. GranuLab : un laboratoire virtuel d'expérimentations pour la découverte scientifique en physique granulaire. In *RFIA'2000*, Paris, France. 2000.

[Breton *et al.* 2001] Laurent Breton, Jean-Daniel Zucker et Eric Clément. Modeling scientific research in granular physics. In Magnani et al. [2001].

[Breton 2000] Laurent Breton. *GranuLab : Un système d'aide à la découverte scientifique appliqué à la physique des milieux granulaires.* Mémoire de pré-soutenance de thèse, Université Paris 6, LIP6, 2000.

[Bruno 2001] Sandra Bruno. Model-based reasonning, analogical reasonning and conceptualization in physics : A case study. In Magnani et al. [2001].

[Buchanan *et al.* 1969] Bruce G. Buchanan, G. L. Sutherland et Edward A. Feigenbaum. Heuristic DENDRAL : a program for generating explanatory processes in organic chemistry. *Machine Intelligence 4*, 1969.

[Buchanan et Shortliffe 1984a] Bruce G. Buchanan et Edward H. Shortliffe. *A model of inexact reasoning in medicine*, chapter 11, pages 233–262. In Buchanan [1984b], 1984.

[Buchanan et Shortliffe 1984b] Bruce G. Buchanan et Edward H. Shortliffe. *Rule-Based Expert Systems : The MYCIN Experiments of th Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1984.

[Campbell *et al.* 1982] A. N. Campbell, V. F. Hollister, R. O. Duda et P. E. Hart. Recognition of a hidden mineral deposit by anartificial intelligence program. *Science*, 217 : 927–929, September 1982.

[Charniak et McDermott 1985] Eugene Charniak et Drew McDermott. *Introduction to Artificial Intelligence.* Addison Wesley, 1985.

[Chung 1988] F. R. K. Chung. The average distance and the independence number. *Journal of Graph Theory*, 12(2) : 229–235, 1988.

[Cornuéjols *et al.* 1999a] A. Cornuéjols, A. Tiberghien et Collet G. L'effet tunnel en cognition : un mécanisme de transfert entre domaines conceptuels. In Sebag M., editor, *cap99*, juin 1999.

[Cornuéjols *et al.* 1999b] A. Cornuéjols, A. Tiberghien et Collet G. Tunnel effects in cognition : A transfer mechanism from known conceptual domains to new ones. In *aisb99*, pages 9–18, april 1999.

[Cornuéjols 1993] A. Cornuéjols. Training Issues in Incremental Learning. In *AAAI-93 Spring Symposium.* AAAI Press, 23-25 mars 1993.

[Cornuéjols 1996a] A. Cornuéjols. Analogie principe d'économie et complexité algorithmique. In *11èmes Journées Françaises d'Apprentissage (JFA'96), Sète, France*, pages 233–247, May 1996.

[Cornuéjols 1996b] A. Cornuéjols. *Machine Learning and Statistics : The interface*, chapter Analogy as Minimization of Description Length. Wiley, 1996.

[Corruble et Ganascia 1997] Vincent Corruble et Jean-Gabriel Ganascia. Induction and the discovery of the causes of scurvy :a computational reconstruction. *Artificial Intelligence*, 91(2) : 205–223, April 1997.

[Corruble 1996] Vincent Corruble. *Une approche inductive de la découverte en médecine : les cas du scorbut et de la lèpre.* Thèse de Doctorat, Université Paris 6, LAFORIA TH96/18, Juin 1996.

[Craig et Nersessian 2001] D. L. Craig et N. J. Nersessian. Model-base reasoning in architectural design : therole of perceptual simulations in analogy. In Magnani et al. [2001].

[CYC 1983] Le système CYC. cf. [Lenat *et al.* 1983, Lenat et Guha 1990], 1983.

[Davis *et al.* 1993] Randall Davis, Howard Shrobe et Peter Szolovits. What is a knowledge representation ? *AI Magazine*, 14(1) : 17–33, Spring 1993.

[DENDRAL 1971] Le système DENDRAL. Découverte de la structure géométrique de molécules en chimie organique, cf. [Feigenbaum *et al.* 1971, Lindsay *et al.* 1980, Feigenbaum et Buchanan 1993, Lindsay *et al.* 1993], 1971.

[Euzenat 1993] Jérôme Euzenat. Représentation granulaire du temps. *Revue d'intelligence artificielle*, 7(3) : 329–361, 1993.

[Euzenat 1999] Jérôme Euzenat. Représentations de connaissance : de l'approximation à la confrontation, 20 janvier 1999.

[FAHRENHEIT 1986] Le système FAHRENHEIT. cf. [Koehn *et al.* 1986, Żytkow 1987], 1986.

[Fajtlowicz 1988] Siemion Fajtlowicz. On conjectures of GRAFFITI. *Discrete Mathematics*, 72 : 113–118, 1988.

[Fajtlowicz 1995] Siemion Fajtlowicz. On conjectures on GRAFFITI V. In *Proceedings of the 7th International Quadrienna Conference on Graph Theory, Combinatorics andApplications*, volume 1, pages 367–376, 1995.

[Falkenhainer et Michalski 1986] Brian C. Falkenhainer et Ryszards S. Michalski. Integrating quantitative and qualitative discovery :the ABACUS system. *Machine Learning*, 1 : 367–401, 1986.

[Falkenhainer 1987] Brian C. Falkenhainer. Scientific theory formation through analogicalinference. In Langley [1987], pages 218–229.

[Feigenbaum *et al.* 1971] Edward A. Feigenbaum, Bruce G. Buchanan et Joshua Lederberg. On generality and problem solving : A case study using the DENDRAL program. *Machine Intelligence*, 6 : 165–190, 1971.

[Feigenbaum et Buchanan 1993] Edward A. Feigenbaum et Bruce G. Buchanan. DENDRAL and Meta-Dendral : Roots of knowledge systems and expert system applications. *Artificial Intelligence*, 59(1-2) : 233–240, February 1993.

[Feigenbaum et Feldman 1963] Edward A. Feigenbaum et Julian Feldman, editors. *Computers and Thought*. New York : McGraw-Hill, 1963.

[Feigenbaum 1977] Edward A. Feigenbaum. The art of Artificial Intelligence : Themes and case studies in knowledge ingeneering. In *International Joint Conference on Artificial Intelligence*, pages 1014–1029. Cambridge, MA, August 1977.

[Forbus 1984] Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24(1-3) : 85–168, December 1984.

[Ganascia 1990] Jean-Gabriel Ganascia. *L'Ame-Machine, les enjeux de l'Intelligence Artificielle*. Seuil, Science ouverte, 1990.

[Ganascia 1993] Jean-Gabriel Ganascia. *L'Intelligence Artificielle*. 1993.

[Gennari *et al.* 1989] John H. Gennari, Pat Langley et Doug Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1-3) : 11–61, 1989.

[Gerwin 1974] D. G. Gerwin. Information processing, data inferences and scientific generalization. *Behav. Sci.*, 19 : 314–325, 1974.

[GLAUBER 1986] Le système GLAUBER. cf. [Jones 1986], 1986.

[Good 1994] P. Good. *Permutation Tests*. Springer Verlag, New York, 1994.

[GRAFFITI 1988] Le système GRAFFITI. Enonce des conjectures dans le domaine de la théorie des graphes et application à la chimie moléculaire, cf. [Chung 1988, Fajtlowicz 1988, Fajtlowicz 1995], 1988.

[Griffith 1997] T. W. Griffith. *A Computational Theory of Dynamical Modeling in Scientific Reasoning*. PhD thesis, Georgia Institute of Technology, Cognitive Science Program, College of Computing, 1997.

[Hendrickson 1995] James B. Hendrickson. Systematic synthesis design : The SynGen program. In AAAI [1995], pages 13–17.

[IDS 1986] Le système IDS. cf. [Langley et Nordhausen 1986], 1986.

[Jones 1986] R. Jones. Generating predictions to aid the scientific discovery process. In *Proceedings of the Fifth National Conference on Artificial Intelligence - (AAAI-86)*, pages 513–517, Philadelphia, PA. Morgan Kaufmann, 1986.

[Kaplan et Simon 1990] C. A. Kaplan et Herbert A. Simon. In search of insight. *Cognitive psychology*, 22 : 374–419, 1990.

[KEKADA 1988] Le système KEKADA. Générateur automatique d'expérimentations réelles,application à la chimie, cf. [Kulkarni et Simon 1988, Kulkarni et Simon 1990], 1988.

[Khabaza et Shearer 1995] T. Khabaza et C. Shearer. Data mining with Clementine. In *IEEE Colloquium on Knowledge Discovery in Databases*, volume Digest 1995/021(B), London. 1995.

[Kocabas et Langley 1995] Sakir Kocabas et Pat Langley. Integration of research tasks for modeling discoveries in particle physics. In AAAI [1995], pages 87–92.

[Kocabas et Langley 1998] Sakir Kocabas et Pat Langley. Generating process explanations in nuclear astrophysics. In *Proceedings of the ECAI-98 Workshop on Machine Discovery*, Brighton, England. 1998.

[Kocabas 1991] Sakir Kocabas. Conflict resolution as discovery in particle physics. *Machine Learning*, 6(3) : 277–309, 1991.

[Koehn *et al.* 1986] B. W. Koehn, Pat Langley et Jan M. Żytkow. Experimenting and theorizing in theory formation. In *Proceedings ACM SIGART International Symposium on Methodologies for Intelligent Systems*, pages 296–307, Knoxville, TN. October 22-24 1986.

[Kokar 1986] Mieczyslaw M. Kokar. Determining arguments of invariant functional descriptions. *Machine Learning*, 1 : 403–422, 1986.

[Kuhn 1970] Thomas S. Kuhn. *The Structure of Scientific Revolutions*. The University of Chicago Press, Chicago, 1970.

[Kuipers 1986] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3) : 289–338, September 1986.

[Kulkarni et Simon 1988] Deepak Kulkarni et Herbert A. Simon. The processes of scientific discovery : The strategy of experimentation. *Cognitive Science*, 12 : 139–175, 1988.

[Kulkarni et Simon 1990] Deepak Kulkarni et Herbert A. Simon. Experimentation in machine discovery. In Shrager et Langley [1990], pages 255–273.

[Langley *et al.* 1983] Pat Langley, Gary L. Bradshaw et Herbert A. Simon. Rediscovering chemistry with the BACON system. In Michalski et al. [1983], pages 307–330.

[Langley *et al.* 1987] Pat Langley, Herbert A. Simon, Gary L. Bradshaw et Jan M. Żytkow. *Scientific Discovery : Computational Explorations of the Creative Processes.* Cambridge, Mass. : MIT Press, 1987.

[Langley et Michalski 1986] Pat Langley et Ryszards S. Michalski. Machine learning and discovery. *Machine Learning*, 1 : 363–366, 1986.

[Langley et Nordhausen 1986] Pat Langley et Bernd Nordhausen. A framework for empirical discovery. In *Proceedings of the Internatinal Meeting on Advances in Learning*, Les Arcs, France. 1986.

[Langley et Simon 1995] Pat Langley et Herbert A. Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38(11) : 54–64, 1995.

[Langley et Żytkow 1989] Pat Langley et Jan M. Żytkow. Data-driven approaches to empirical discovery. *Artificial Intelligence*, 40(1-3) : 283–312, 1989.

[Langley 1978] Pat Langley. BACON.1 : a general discovery system. In *Proceedings of the Second Biennal Conference of the Canadian Society for Computational Studies of Intelligence*, pages 173–180, Toronto, Ont. 1978.

[Langley 1981] Pat Langley. Data-driven discovery of physical laws. *Cognitive Science*, 5 : 31–54, 1981.

[Langley 1987] Pat Langley, editor. *Proceedings of the Fourth International Workshop on Machine Learning*, Irvine, CA, 1987.

[Langley 1998] Pat Langley. The computer-aided discovery of scientific knowledge. In *Discovery Science*, pages 25–39, 1998.

[Lenat *et al.* 1983] Douglas B. Lenat, D. Borning, D. McDonald, S. Taylor et S. Weyer. Knowsphere : Design of an expert system with an encyclopedic knowledge base. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany. August 1983.

[Lenat et Brown 1984] Douglas B. Lenat et John Seely Brown. The nature of heuristics IV : Why AM and EURISKO appear to work. *Artificial Intelligence*, pages 269–294, July 1984.

[Lenat et Feigenbaum 1991] Douglas B. Lenat et Edward A. Feigenbaum. On the thresholds of knowledge. *Artificial Intelligence*, 47 : 185–250, 1991.

[Lenat et Guha 1990] Douglas B. Lenat et R. V. Guha. *Building Large Knowledge-Based Systems : Representation and Inference in the CYC Project.* MA : Addison-Wesley, 1990.

[Lenat et Guha 1994] Douglas B. Lenat et R. V. Guha. Enabling agents to work together. *Communication of ACM*, pages 126–142, July 1994.

[Lenat 1975] Douglas B. Lenat. BEINGS : Knowledge as interacting experts. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, Tbilisi, U.S.S.R. September 1975.

[Lenat 1977a] Douglas B. Lenat. Automated theory formation in mathematics. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 833–842. Cambridge, MA, Morgan Kaufmann, 1977.

[Lenat 1977b] Douglas B. Lenat. The ubiquity of discovery. *The Journal of Artificial Intelligence*, pages 257–285, December 1977.

[Lenat 1982] D. B. Lenat. AM : discovery in mathematics as heuristic search. In R. Davis et D. B. Lenat, editors, *Knowledge-based systems in artificial intelligence*. McGraw Hill, New York, N. Y., 1982.

[Lenat 1983] Douglas B. Lenat. Three case studies in learning. In Michalski et al. [1983].

[Lenat 1984a] Douglas B. Lenat. Automated theory formation in mathematics. In W. Bledsoe et D. Loveland, editors, *Theorem Proving : After 25 Years*, pages 287–314. American Mathematical Society, Providence, 1984.

[Lenat 1984b] Douglas B. Lenat. Computer software for intelligent systems : An underview of AI. *Scientific American*, pages 204–213, September 1984.

[Lindsay *et al.* 1980] R. K. Lindsay, Bruce G. Buchanan, Edward A. Feigenbaum et Joshua Lederberg. *Applications of Artificial Intelligence for Organic Chemistry : The DENDRAL Project*. McGraw-Hill, New York, 1980.

[Lindsay *et al.* 1993] R. K. Lindsay, Bruce G. Buchanan, Edward A. Feigenbaum et Joshua Lederberg. DENDRAL : A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence*, 61(2) : 209–261, June 1993.

[Magnani *et al.* 2001] Lorenzo Magnani, Nancy J. Nersessian et Kenneth J. Knoespel, editors. *Model-Based Reasonning : Scientific Discovery,Technological Innovation, Values (MBR'01)*, Pavia, Italy, May 17-19 2001. à paraître.

[Margerie 1992] Moulet Margerie. ARC.2 : Linear regression in ABACUS. In *Proceedings of the International Conference on Machine Learning, Workshop on Machine Discovery*, 1992.

[MECHEM 1992] Le système MECHEM. Découverte de chaînes de réactions chimiques, cf. [Valdés-Pérez 1992, Valdés-Pérez 1994b, Valdés-Pérez 1995b], 1992.

[Michalski *et al.* 1983] Ryszard S. Michalski, Jaime G. Carbonell et Tom M. Mitchell, editors. *Machine Learning : An Artifical Intelligence Approach*, Tioga, Palo Alto, CA, 1983.

[Michalski et Stepp 1983] Ryszard S. Michalski et Robert Stepp. Learning from observation : Conceptual clustering. In Michalski et al. [1983].

[Moore 1995] A. W. Moore. A brief history of infinity. *Scientific American*, 272(4) : 112–116, 1995.

[Moulet 1993] Marjorie Moulet. *ARC : Découverte Empirique de Lois Numériques ou ABACUS Revu et Corrigé*. Thèse de Doctorat, Orsay, Université de Paris Sud, 22 janvier 1993.

[Moulet 1994] Marjorie Moulet. Iterative model construction with regression. In *European Conference on Artificial Intelligence*, pages 448–452, 1994.

[MYCIN 1974] Le système MYCIN. Système expert d'aide au diagnostic médical, cf. [Shortliffe 1974, Shortliffe 1976, Buchanan et Shortliffe 1984b], 1974.

[Newell et Simon 1963] Allen Newell et Herbert A. Simon. *GPS : A program that simulates human thought*, pages 279–293. In Feigenbaum et Feldman [1963], 1963.

[Newell et Simon 1972] Allen Newell et Herbert A. Simon. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

[Newell et Simon 1976] Allen Newell et Herbert A. Simon. Computer science as empirical inquiry : Symbols and search. *Communication of the ACM*, 19 : 111–126, 1976.

[PAULI 1994] Le système PAULI. Découverte dans le domaine de la physique des particules, cf. [Valdés-Pérez 1994a], 1994.

[PENCHANT 1994] Le système PENCHANT. Recherche heuristique de patrons de comportements complexes, application à la biologie cellulaire, cf. [Valdés-Pérez et Pérez 1994], 1994.

[Potts 1997] W. Potts. Introduction to predictive data mining using enterprise miner software. Technical report, SAS Institute, Cary, 1997.

[PROSPECTOR 1982] Le système PROSPECTOR. Découverte dans le domaine de la chimie minérale,cf. [Campbell *et al.* 1982], 1982.

[Quinlan 1993] J. Ross Quinlan. C4.5 : Programs for machine learning, 1993.

[Robisson 2001] Bruno Robisson. *Découverte automatique de circuits de l'électronique de puissance*. Thèse de Doctorat, Université Paris 6, 2001.

[Shasha et Lazere 1998] Dennis Shasha et Cathy Lazere. *Out of their minds. The lives and discoveries of 15 great computer scientists*. Springer-Verlag, 1998.

[Shortliffe 1974] Edward H. Shortliffe. *MYCIN : A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection.* PhD thesis, Stanford, CA, 1974.

[Shortliffe 1976] Edward H. Shortliffe. *Computer-based medical consultations : MYCIN.* New-York : American Elsevier, 1976.

[Shrager et Langley 1990] Jeff Shrager et Pat Langley, editors. *Computational models of scientific discovery and theory formation.* San Mateo, Calif., Morgan Kaufmann Publishers, 1990.

[Simon *et al.* 1997] Herbert A. Simon, Raúl E. Valdés-Pérez et D. H. Sleeman. Editorial : Scientific discovery and simplicity of method. *Artificial Intelligence*, 91(2) : 177–181, 1997.

[Simon 1979] Herbert A. Simon. Artificial intelligence research strategies in light of AI models of scientific discovery. In *Proceedings of the Sixth Internatinal Joint Conference on Artificial Intelligence*, pages 1086–1093, Menlo Park, Calif. 1979.

[Simon 1992] Herbert A. Simon. Scientific discovery as problem solving : reply to critics. *International Studies in the Philosophy of Science*, 6(1) : 69–88, 1992.

[Simon 1995] Herbert A. Simon. Artificial Intelligence : an empirical science. *Artificial Intelligence*, 1995.

[Spirtes *et al.* 1993] P. Spirtes, C. Glymour et R. Scheines. *Causation, Prediction and Search.* Springer Verlag, New York, 1993.

[Valdés-Pérez *et al.* 1992] Raúl E. Valdés-Pérez, Herbert A. Simon et R. F. Murphy. Discovery of pathways in science. In *Proceedings of the International Conference on Machine Learning, Workshop on Machine Discovery*, pages 51–57, 1992.

[Valdés-Pérez *et al.* 1993] Raúl E. Valdés-Pérez, Jan M. Żytkow et Herbert A. Simon. Scientific model-building as search in matrix spaces. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 472–478, Menlo park, Calif. American Association for Artificial Intelligence, 1993.

[Valdés-Pérez et Pérez 1994] Raúl E. Valdés-Pérez et Aurora Pérez. A powerful heuristic for the discovery of complex patterned behavior. *Proceedings of the Eleventh International Conference on Machine Learning*, pages 236–334, 1994.

[Valdés-Pérez 1992] Raúl E. Valdés-Pérez. Theory-driven discovery of reaction pathways in the MECHEM system. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 63–69, 1992.

[Valdés-Pérez 1993] Raúl E. Valdés-Pérez. A scientific basis for computational science. Technical Report CS-93-162, 1993.

[Valdés-Pérez 1994a] Raúl E. Valdés-Pérez. Algebraic reasonning about reactions : discovery of conserved properties in particle physics. *Machine Learning*, 17(1) : 47–68, 1994.

[Valdés-Pérez 1994b] Raúl E. Valdés-Pérez. Conjecturing hidden entities via simplicity and conservation laws : machine discovery in chemistry. *Artificial intelligence*, 65(2) : 247–280, 1994.

[Valdés-Pérez 1994c] Raúl E. Valdés-Pérez. Human-computer interactive elucidation of reaction mechanisms : application to catalysed hydrogenolysis of ethane. *Catalysis Letters*, 28(1) : 79–87, September 1994.

[Valdés-Pérez 1994d] Raúl E. Valdés-Pérez. Systematic induction and parsimony of phenomenological conservation laws. *Computer Physics Cmmunication*, 83(2-3) : 171–180, 1994.

[Valdés-Pérez 1995a] Raúl E. Valdés-Pérez. Generic tasks of scientific discovery. In AAAI [1995].

[Valdés-Pérez 1995b] Raúl E. Valdés-Pérez. Machine discovery in chemistry : New results. *Artificial Intelligence*, 74(1) : 191–201, March 1995.

[Valdés-Pérez 1995c] Raúl E. Valdés-Pérez. Praxis. *Artificial Intelligence*, 74(1) : 191–201, March 1995.

[Valdés-Pérez 1995d] Raúl E. Valdés-Pérez. Some recent human-computer discoveries in science and what accounts for them. *AI Magazine*, 16(3) : 37–44, Fall 1995.

[Valdés-Pérez 1996] Raúl E. Valdés-Pérez. A new theorem in particle physics enabled by machine discovery. *Artificial Intelligence*, 82(1-2) : 331–339, April 1996.

[Valdés-Pérez 1998] Raúl E. Valdés-Pérez. Why some programs do knowledge discovery well : Experiences from computational scientific discovery. Unpublished Manuscript, 1998.

[Valdés-Pérez 1999] Raúl E. Valdés-Pérez. Principles of human-computer collaboration for knowledge discovery in science. *Artificial Intelligence Journal*, 107(2) : 335–346, 1999.

[Weld et de Kleer 1990] Daniel S. Weld et Johan de Kleer. *Readings in Qualitative Reasoning About Physical ystems*. Morgan Kaufmann, San Mateo, CA, 1990.

[Winston 1989] P. H. Winston. *Artificial Intelligence*. Addison-Wesley, Reading, MA, USA, 1989.

[Wipke *et al.* 1978] W. T. Wipke, G. I. Ouchi et S. Krishnan. Simulation and evaluation of chemical synthesis - secs : an application of artificial intelligence techniques. *Artificial intelligence*, 11 : 173–193, 1978.

[Żytkow 1987] Jan M. Żytkow. Combining many searches in the FAHRENHEIT discoverysystem. In Langley [1987], pages 281–287.

2 Physique des milieux granulaires

[Automates Cellulaires (CA) 1993] Simulation par Automates Cellulaires (CA). cf. [Mora et Place 1993, Carlson *et al.* 1994, Bubak 1995], 1993.

[Bagster et Kirk 1985] D. F. Bagster et R. Kirk. Computer generation of a model to simulate granular material behavior. *Journal of Powder and Bulk solids Technology*, 9 : 19–24, 1985.

[Bastolla *et al.* 1997] U. Bastolla, Farhang Radjai et D. E. Wolf. Influence of friction on the force network in granular packings. In Wolf et Grassberger [1997], pages 193–199.

[Baxter 1997] G. W. Baxter. Stress distributions in a two dimensional granular material. In Behringer et Jenkins [1997], pages 345–348.

[Behringer et Jenkins 1997] Robert P. Behringer et J. T. Jenkins, editors. *Powders and Grains 97*, A. A. Balkema, Rotterdam, 1997.

[Bouamrane 2001] Rachid Bouamrane. Private communications, 2001.

[Brendel et Dippel 1998] L. Brendel et S. Dippel. Lasting contacts in molecular dynamics simulations. In Herrmann et al. [1998], page 313.

[Breton *et al.* 2001] Laurent Breton, Philippe Claudin, Eric Clément et Jean-Daniel Zucker. Stress response function of a two-dimensional ordered packing of frictional beads. *Europhysics Letters*, en préparation, 2001.

[Bubak 1995] M. Bubak. 2-D cellular automata and short range molecular dynamics programs for simulations on networked workstations and parallel computers. *Lecture Notes in Computer Science*, 1041 : 61–68, 1995.

[Carlson *et al.* 1994] J. M. Carlson, J. S. Langer et Shaw B. E. Dynamics of earthquake faults. *Rev. Mod. Phys.*, 66 : 657, 1994.

[Claudin 1999] Philippe Claudin. *La physique des tas de sable : Description phénoménologique de la propagation des contraintes dans les matériaux granulaires.* Thèse de Doctorat, Université Paris XI, Service de Physique de l'Etat Condensé du CEA Saclay, 1999.

[Codd 1968] E. F. Codd. *Cellular Automata.* Academic Press, New York, 1968.

[Coppersmith *et al.* 1996] S. N. Coppersmith, C.-H. Liu, S. Majumbar, O. Narayan et Thomas A. Witten. Model for force fluctuations in bead packs. *Phys. Rev. E*, 53(5) : 4673–4685, 1996.

[Cundall et Strack 1979] P. A. Cundall et O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1) : 47–65, 1979.

[Dantu 1957] P. Dantu. Contribution à l'étude mécanique et géométrique des milieux pulvérulents. In *Proc. of the 4th International Conference on Soil Mechanics and Foundation Engineering*, volume 1, pages 144–148. Butterworths Scientific Publications, London, 1957.

[de Coulomb 1776] Charles Augustin de Coulomb. Sur une application des règles de maximis et minimis à quelques problèmes de statique relatifs à l'architecture. *Mémoires de Mathématiques de l'Académie Royale des Sciences*, page 343, 1776.

[de Gennes 1995] Pierre-Gilles de Gennes. Dynamique superficielle d'un matériau granulaire. *C. R. Acad. Sciences*, 10(321 [IIb]) : 501–506, 1995.

[de Gennes 1999] Pierre-Gilles de Gennes. *Rev. Mod. Phys.*, 71 : S374, 1999.

[De Josselin de Jong et Verruijt 1969] G. De Josselin de Jong et A. Verruijt. Etude photoélastique d'un empilement de disques. *Cah. Gr. Franc. Rhéol.*, 2 : 73, 1969.

[Duran 1997] Jacques Duran. *Sables, poudres et grains.* Eyrolles Sciences, 1997.

[Dynamique des Contacts (CD) 1995] Simulation par Dynamique des Contacts (CD). cf. [Moreau et Jean 1996, Radjai 1995, Radjai *et al.* 1997, Radjai 1998], 1995.

[Dynamique Moléculaire (MD) 1995] Simulation par Dynamique Moléculaire (MD). cf. [Rapaport 1995, Brendel et Dippel 1998], 1995.

[Eloy et Clément 1997] Christophe Eloy et Eric Clément. Stochastic aspects of the force network in a regular granular piling. *J. Phys. I*, 7 : 1541–1558, december 1997.

[GranuSolve 1999] Simulation par Système Multi-Agent GranuSolve. cf. [Breton *et al.* 2001] et [Breton *et al.* 1999, Breton 1999, Breton *et al.* 2000] dans la partie systèmes multi-agents, 1999.

[Guyon et Troadec 1994] Etienne Guyon et Jean-Paul Troadec. *Du sac de billes au tas de sable.* Editions Odile Jacob, Sciences, 1994.

[Hadeler et Kuttler 1999] K. Hadeler et C. Kuttler. Dynamical models for granular matter. *Granular Matter*, 2 : 9–18, 1999.

[Herrmann *et al.* 1998] H. J. Herrmann, Stephan Luding et J. P. Hovi, editors. *Physics of Dry Granular Media*, NATO ASI, Amsterdam, 1998. Kluwer.

[Hong 1993] D. C. Hong. Stress distribution of a hexagonally packed granular pile. *Phys. Rev. E*, 47(1) : 760–762, 1993.

[Howell *et al.* 1997] Daniel Howell, B. Miller, C. O'Hern et Robert P. Behringer. Stress fluctuations for granular materials. In Wolf et Grassberger [1997], pages 133–147.

[Jullien et Meakin 1987] Rémi Jullien et Paul Meakin. Simple three-dimensional models for ballistic deposition with restructuring. *Eur. Phys. Lett.*, 4(12) : 1385–1390, December 1987.

[Jullien et Meakin 1988] Rémi Jullien et Paul Meakin. Ballistic deposit and segregation of polydisperse spheres. *Eur. Phys. Lett.*, 6(7) : 629–634, August 1988.

[Jullien et Meakin 1990] Rémi Jullien et Paul Meakin. A mechanism for particle size segregation in three dimensions. *Nature*, 344 : 425–427, March 1990.

[Kolb *et al.* 1999] Evelyne Kolb, T. Mazozi, Eric Clément et Jacques Duran. Force fluctuations in a vertically pushed granular column. *Eur. Phys. J. B*, 8 : 483–491, 1999.

[Liffman *et al.* 1994] K. Liffman, D. Y. C. Chan et B. D. Hughes. On the stress depression under a sandpile. *Powder Tech.*, 78 : 263, 1994.

[Liu *et al.* 1995] C.H. Liu, S.R. Nagel, D.A. Schecter, S.N. Coppersmith, S. Majumdar, O. Narayan et Thomas A. Witten. Force fluctuations in bead packs. *Science*, 269 : 513–515, 1995.

[Liu et Nagel 1992] C.-H. Liu et S. R. Nagel. Sound in sand. *Phys. Rev. Lett.*, 68(15) : 2301–2304, 1992.

[Luding 1997] Stephan Luding. Stress distribution in static two dimensional granular model media in the abscence of friction. *Phys. Rev. E*, 55(4) : 4720–4729, 1997.

[Matuttis *et al.* 1999] H. G. Matuttis, Stephan Luding et H. J. Herrmann. Discrete element simulations of dense packings and heaps made of spherical and non-spherical particles. *Powder Technology*, 1999.

[Matuttis 1998] H. G. Matuttis. Simulations of the pressure distribution under a two dimensional sand-pile of polygonal particles. *Granular Matter*, 1(2) : 83–91, 1998.

[Mitchell 1996] Tom M. Mitchell. Computation in cellular automata : a selected review. Technical report, Santa Fe Institute Working Paper, 9609 -074, 1996.

[Monte-Carlo (MC-Granu) 1997] Simulation par méthode du type Monte-Carlo (MC-Granu). cf. [Eloy et Clément 1997], 1997.

[Mora et Place 1993] P. Mora et D. Place. A lattice solid model for the nonlinear dynamics of earthquakes. *Int. J. Mod. Phys.*, pages 1059–1074, 1993.

[Moreau et Jean 1996] Jean-Jacques Moreau et M. Jean. Numerical treatment of contact and friction : the contact dynamics method. In *Proc. Engineering systems design and analysis conference*, volume 4, page 201. ASME, New York, 1996.

[Méthode des Eléments Finis (FEM) 1998] Simulation par la Méthode des Eléments Finis (FEM). cf. [Matuttis 1998, Matuttis 1998, Tsoungui 1998], 1998.

[Mueth *et al.* 1998] D. M. Mueth, H. M. Jaeger et S. R. Nagel. Force distribution in a granular medium. *Phys. Rev. E*, 57(3) : 3164–3169, 1998.

[Oron et Herrmann 1998] G. Oron et H. J. Herrmann. Exact calculation of force networks in granular piles. *Phys. Rev. E*, 58 : 2079–2089, 1998.

[Ouaguenouni et Roux 1997] S. Ouaguenouni et Jean-Noël Roux. Arching without friction : a simple model. In Wolf et Grassberger [1997].

[Radjai *et al.* 1996] Farhang Radjai, M. Jean, Jean-Jacques Moreau et S. Roux. Force distribution in dense two-dimensional granular systems. *Phys. Rev. E*, 77(2) : 274, 1996.

[Radjai *et al.* 1997] Farhang Radjai, D. E. Wolf, S. Roux, M. Jean et Jean-Jacques Moreau. Force networks in granular packings. In Wolf et Grassberger [1997], pages 169–179.

[Radjai *et al.* 1998] Farhang Radjai, D. E. Wolf, M. Jean et Jean-Jacques Moreau. Bimodal character of stress transmission in granular packings. *Phys. Rev. Lett.*, 80(1) : 61–64, 1998.

[Radjai 1995] Farhang Radjai. *Dynamique des rotations et frottement collectif dans les systèmes granulaires*. Thèse de Doctorat, Université de Paris-Sud U.F.R. Scientifique d'Orsay, décembre 1995.

[Radjai 1997] Farhang Radjai. Force networks in dense granular media. In Behringer et Jenkins [1997], page 211.

[Radjai 1998] Farhang Radjai. Multi-contact dynamics. In Herrmann et al. [1998], page 305.

[Rapaport 1995] D. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, Cambridge, 1995.

[Rosato 1987] A. Rosato. Why the brazil nuts are on top : Size segregation of particulate matter by shaking. *Phys. Rev. Lett.*, 58 : 1038, 1987.

[Signorini 1933] A. Signorini. Sopra alcune questioni di elastostatica. *Atti Soc. Ital. Progr. Sci.*, 1933.

[Socolar 1998] Joshua E. S. Socolar. Average stresses and force fluctuations in noncohesive granular materials. *Phys. Rev. E*, 57(3) : 3204–3215, 1998.

[Tkachenko et Witten 1999] Alexei V. Tkachenko et Thomas A. Witten. Stress propagation through frictionless granular material. *Phys. Rev. E*, 60(1) : 687–696, July 1999.

[Travers *et al.* 1986] T. Travers, M. Ammi, D. Bideau et A. Gervois. Uniaxial compression effects on 2d mixtures of 'hard' and 'soft' cylinders. *J. Phys. A*, 19 : L1033–L1038, 1986.

[Travers *et al.* 1987] T. Travers, M. Ammi, D. Bideau, A. Gervois, J. Messager et Jean-Paul Troadec. Uniaxial compression of 2d packings of cylinders. effects of weak disorder. *Eur. Phys. Lett.*, 4(3) : 329–332, 1987.

[Tsoungui *et al.* 1998] Olivier Laurent Tsoungui, D. Vallet et J.-C. Charmet. Use of contact area trace to study the force distribution inside 2d granular systems. *Granular Matter*, 1 : 65–69, 1998.

[Tsoungui 1998] Olivier Laurent Tsoungui. *Modélisation numérique et expérimentale du broyage des grains par compression.* Thèse de Doctorat, Octobre 1998.

[Vanel *et al.* 1999] Loïc Vanel, Daniel Howell, Robert Behringer et Clément Eric. Effect of construction technique on stress profiles for granular heaps. *Phys. rev. E*, 1999.

[Vanel et Clément 1999] Loïc Vanel et Eric Clément. Pressure screening and fluctuations at the bottom of a granular column. *Eur. Phys. J. B*, 8 : 483, 1999.

[Vanel 1999] Loïc Vanel. *Etude expérimentale de l'équilibre mécanique d'un milieu granulaire : exemples du silo et du tas de sable.* Thèse de Doctorat, Université de Paris 6, Juin 1999.

[Visscher et Bolsterli 1972] William M. Visscher et M. Bolsterli. Random packing of equal and unequal spheres in two and three dimensions. *Nature*, 239 : 504–507, October 1972.

[Wittmer *et al.* 1996] Joachim P. Wittmer, Philippe Claudin, Michael E. Cates et Jean-Philippe Bouchaud. An explanation for the central stress minimum in sandpiles. *Nature*, 382 : 336, 1996.

[Wittmer *et al.* 1997a] Joachim P. Wittmer, Michael E. Cates et Philippe Claudin. Stress propagation and arching in static sandpiles. *J. Phys. I*, 7 : 39–80, 1997.

[Wittmer *et al.* 1997b] Joachim P. Wittmer, Philippe Claudin, Michael E. Cates et Jean-Philippe Bouchaud. A new approach to stress propagation in sandpiles and silos. In Wolf et Grassberger [1997], page 153.

[Wolf et Grassberger 1997] D. E. Wolf et P. Grassberger, editors. *Friction, Arching, Contact Dynamics*, World Scienctific, Singapore, 1997.

3 Laboratoires Virtuels

[Augustin *et al.* 1967] Donald C. Augustin, Mark S. Fineberg, Bruce C. Johnson, Robert N. Linebarger, F. John Sansom et Jon C. Strauss. The CSI continuous system simulation language (CSSL). *Simulation*, 9 : 281–303, 1967.

[Broenink 1990] Jan F. Broenink. *Computer-aided Physical-Systems Modeling and Simulations : A Bong-Graph Approach.* PhD thesis, University of Twente, Enschede, The Netherlands, 1990.

[Cellier 1991] François E. Cellier. *Continuous system modeling.* Springer-Verlag, New York, 1991.

[Cremer et Stewart 1989] James F. Cremer et A. James Stewart. The architecture of *newton*, a general-purpose dynamics simulator. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 1806–1811, May 1989.

[Hoffmann 1989] Christoph M. Hoffmann. *Geometric and solid modeling.* Morgan Kaufmann, San Mateo, California, 1989.

[Mitchell et Assoc. 1986a] Mitchell et Gauthier Assoc. *ACSL : Advanced Continuous Simulation Language - User Guide and Reference Manual.* Concord, Mass., 1986.

[Mitchell et Assoc. 1986b] Mitchell et Gauthier Assoc. *AUTOSIM.* Concord, Mass., 1986.

[Nicholson 1997] S. Nicholson. The need for component methodologies in global applications. In *SGML/XML '97 Conference*, pages 241–247, Washington, USA. Graphic Communications Association, 1997.

[Pritsker 1986] A. B. Pritsker. *Introduction to simulation and SLAM II.* System Publishing Corporation, 1986.

[Rana *et al.* 1999] O. Rana, M. Li, D. Walker et M. Shields. An XML based component model for generating scientific applications and performing large scale simulations in a meta-computing environment. In *Generative Component Based Software Engineering*, Erfert, Germany. 1999.

[Xu 2000] L. Xu. Using JINI and XML to build a component based distributed system. Technical report, University of Maryland Baltimore County, 2000.

[Yarimagan 1999] Y. Yarimagan. *A Component Based Workflow System for Enacting Process Defined in XML.* Ms thesis, Middle East Technical University, 1999.

4 Langages

[Adobe 2001] Adobe. SVGViewer : logiciel de rendu grahique pour les navigateurs internet explorer et netscape. http ://www.adobe.com/svg/, août 2001.

[BATIK 2001] BATIK. BATIK : rendu de scènes graphiques vectorielles et librairies java. http ://xml.apache.org/batik, août 2001.

[Catania *et al.* 2000] Barbara Catania, Elena Ferrari, Alon Y. Levy et Alberto O. Mendelzon. XML and object technology. In *ECOOP Workshops*, pages 191–202, 2000.

[ChannelPoint 2001] ChannelPoint. Merlot : éditeur XML. http ://www.merlotxml.org, août 2001.

[Eckstein et Casabianca 2000] Robert Eckstein et Michel Casabianca. *XML précis et concis*. O'Reilly, 2000.

[Fontaine 1997] Alain Bernard Fontaine. *La bibliothèque standard STL du C++*. InterEditions, 1997.

[Géron et Tawbi 1999] Aurélien Géron et Fatmé Tawbi. *Pour mieux développer avec C++*. Dunod, Informatiques, 1999.

[Kernighan et Ritchie 1994] B.W. Kernighan et D.M. Ritchie. *Le Langage C, 2ème édition*. Masson, Paris, 1994.

[Lamport 1985] Leslie Lamport. *LaTeX - A Document Preparation System*. Addison-Wesley, 1985.

[Michard 1999] Alain Michard. *XML, langage et applications*. Eyrolles, 1999.

[Press *et al.* 1999] William H. Press, Saul A. Teukolsky, William T. Vetterling et Brian P. Flannery. *Numerical recipes in C, The art of scientific programming, Second edition*. Cambridge University Press, 1999.

[SVG 2000] SVG. Scalable vector graphics (SVG) 1.0 specification. Technical report, http ://www.w3.org/TR/SVG, W3C, 2000.

[Tetrad 2001] Tetrad. Tetrad : librairies java de gestion des contraintes sur des graphes. http ://www.phil.cmu.edu/tetrad/, août 2001.

[Xalan 2001] Xalan. Xalan : application de règles de transformation à un document XML. http ://xml.apache.org/xalan-j, août 2001.

[Xerces 2001] Xerces. Xerces : lecture et chargement en mémoire de documents XML. http ://xml.apache.org/xerces-j, août 2001.

[XML 1998] XML. Extensible markup language (XML) 1.0. Technical report, http ://www.w3.org/TR/REC-xml, W3C, Février 1998.

[XSLT 1999] XSLT. XSL transformations (XSLT) version 1.0. Technical report, http ://www.w3.org/TR/xslt, W3C, 1999.

5 Systèmes Multi-Agents

[Breton *et al.* 1999] Laurent Breton, Jean-Daniel Zucker et Eric Clément. Une approche multi-agents pour la résolution d'équations en physique des milieux granulaires. In *Ingénierie des systèmes multi-agents - JFIADSMA'99*, pages 281–293, La Réunion. Editions Hermès, Paris, 1999.

[Breton *et al.* 2000] Laurent Breton, Jean-Daniel Zucker et Eric Clément. A multi-agent based simulation of sand piles in a static equilibrium. *Multi-Agent Based Simulation Workshop in Lecture Notes in Artificial Intelligence*, 1979 : 108–118, 2000.

[Breton 1999] Laurent Breton. *Approche Multi-Agents pour l'étude du réseau de forces dans un tas de sable.* Mémoire de stage de D.E.A. IARFA, Université Paris 6, Septembre 1999.

[Collinot et Drogoul 1996] Anne Collinot et Alexis Drogoul. La méthode de conception multi-agents CASSIOPÉE : application à la robotique collective. Technical report, LAFORIA, LIP6, 1996.

[Demazeau et Müller 1990] Yves Demazeau et Jean-Pierre Müller, editors. *Decentralized A.I. 2*, Saint-Quentin en Yvelines, France, August 13-16 1990. Proceedings of the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW), Elsevier Science Publishers B.V.

[Drogoul *et al.* 1991] Alexis Drogoul, Jacques Ferber et Eric Jacopin. Pengi : Applying eco-problem solving for behavior modeling in an abstract eco-system. In E. Mosekilde, editor, *European Simulation Multiconference - ESM'91*, 1991.

[Drogoul et Dubreuil 1991] Alexis Drogoul et C. Dubreuil. Eco-problem solving : Results of the n-puzzle. In Werner et Demazeau [1991].

[Drogoul et Zucker 1998] Alexis Drogoul et Jean-Daniel Zucker. Methodological issues for designing multi-agent systems with machine learning techniques : Capitalizing experiences from the robocup challenge. Technical report, Université Paris 6, LIP6, 1998.

[Drogoul 1993] Alexis Drogoul. *De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents.* Thèse de Doctorat, Université Paris 6, 1993.

[Drogoul 2000] Alexis Drogoul. Systèmes multi-agents situés, 17 mars 2000.

[Ferber 1990] Jacques Ferber. Eco-problem-solving : How to solve problems by interactions. Technical Report 05/90, Laboratoire Formes et Intelligence Artificielle, Paris 6, janvier 1990.

[Guillemet *et al.* 1999] Alexandre Guillemet, Grégory Haïk, Thomas Meurisse, Briot Jean-Pierre et Marc Lhuillier. Mise en oeuvre d'une approche componentielle pour la conception d'agents. In *Ingénierie des systèmes multi-agents - JFIADSMA'99*, La Réunion. Editions Hermès, Paris, 1999.

[Jacopin 1993] Eric Jacopin. *Algorithmique de l'interaction : le cas de la planification.* PhD thesis, Université Paris 6, 1993.

[Pauchet 2001] Alexandre Pauchet. *Analyse des solutions de* GranuSolve *par classification automatique.* Mémoire de stage de D.E.A. IARFA, Université Paris 6, Septembre 2001.

[Renault 2001] Valérie Renault. *Organisation de sociétés d'agents pour la visualisation d'informations dynamiques.* Thèse de Doctorat, Université Paris 6, Décembre 2001.

[Rouchier *et al.* 2000] Juliette Rouchier, François Bousquet, Olivier Barreteau, Christophe Le Page et Jean-Luc Bonnefoy. Multi-agent modelling and renewable ressources issues : The relevance of shared representations for interacting agents. In *Proc. of the Multi-Agent Based Simulation Workshop*, volume 1979, pages 181–197. Springer, 2000.

[Servat *et al.* 1998a] David Servat, Edith Perrier, J.-P. Treuil et Alexis Drogoul. Towards virtual experiments laboratories : How multi-agent simulations can cope with multiple scales of analysis and viewpoints. In *Virtual Worlds'98*, 1998.

[Servat *et al.* 1998b] David Servat, Edith Perrier, J.-P. Treuil et Alexis Drogoul. When agents emerge from agents : Introducing multi-scale viewpoints in multi-agent simulations. In *Proc. of the Multi-Agent Based Simulation Workshop*, Paris, France. Springer-Verlag, 1998.

[Stone et Veloso 2000] Peter Stone et Manuela M. Veloso. Multiagent systems : A survey from a machine learning perspective. *Autonomous Robots*, 8(3) : 345–383, 2000.

[Treuil *et al.* 1997] Jean-Pierre Treuil, Edith Perrier et Christophe Cambier. Directions pour une approche multi-agent de la simulation de processus physiques spatialisés. pages 211–228. Hermes, 1997.

[Weiss 1994] G. Weiss. Some studies in distributed machine learning and organizational design. Technical report, Institut für Informatik, TU München, 1994.

[Werner et Demazeau 1991] Eric Werner et Yves Demazeau, editors. *Decentralized A.I. 3*, Kaiserslauter, Germany, August 5-7 1991. Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW), Elsevier Science Publishers B.V.

6 Problèmes de Satisfaction de Contraintes

[Hirayama et Yokoo 1997] K. Hirayama et M. Yokoo. Distributed partial constraint satisfaction problem. In *Proc. of the Third Internatinal Conference on Principles and Practice of Constraint Programming (CP-97)*, pages 222–236, 1997.

[Lemaitre et Verfaillie 1997] M. Lemaitre et G. Verfaillie. An incompltete method for solving distributed valued constraint satisfaction problems. In *Proc. of AAAI-97 Workshop on Constraints and Agents*, 1997.

[Tsang 1993] E. Tsang. *Foundations of Constraint Satisfaction.* Academic Press, 1993.

[Yokoo *et al.* 1998] M. Yokoo, E.H. Durfee, T. Ishida et K. Kuwabara. The distributed constraint satisfaction problem : formalization and algorithms. In *IEEE Trans. on Knowledge and Data Engineering*, volume 10, pages 673–685, 1998.

Annexe A

Les équations détaillées du modèle de Clément et d'Eloy

Sommaire

A.1 Le modèle à N forces de contact

A.1.1 L'empilement

La structure de l'empilement est celle d'un empilement compact triangulaire du type “boulets de canon” (cf. Fig. A.1). De plus, afin de s'affranchir des conditions aux limites, nous considèrons un tas cylindrique. Par exemple, le grain numéro 7, sur la figure suivante, est en contact avec les grains numéros 8 et 14.

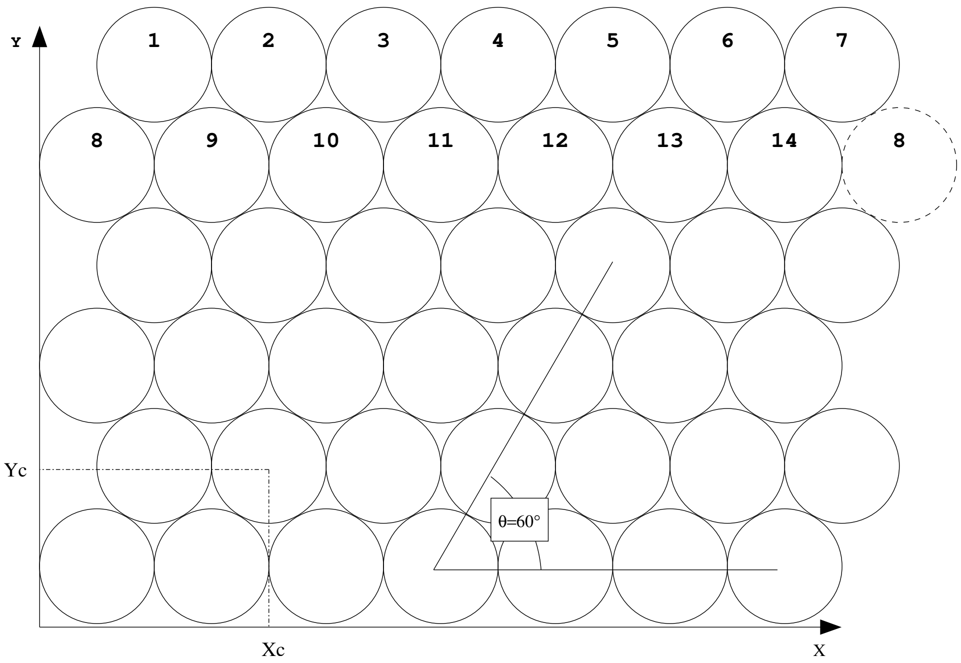


FIG. A.1 – Empilement compact triangulaire

Soient H et L, respectivement le nombre de couches et le nombre de colonnes de grains du tas. Soient i et j les indices de position d'un grain dans un tableau de grains ($i = [0 \ldots H-1]$ et $j = [0 \ldots L-1]$) La position du centre d'un grain dans le repère $(O, \vec{x}, \vec{y})$ est donnée en fonction de la ligne i et de la colonne j du grain par la relation :

$$\left\{ \begin{array}{lcl} X_c & = & rayon + (j \times 2 \times rayon) + (rayon \times (i\%2)) \\ Y_c & = & rayon + (hauteur - 1 - i) \times (\sqrt{3} \times rayon) \end{array} \right. \tag{A.1}$$

A.1.2 Le système *grain*

Considérons le système physique *grain* possédant une masse m dans un champ gravitique de valeur g. L'équilibre d'un grain, sur lequel N forces de contact sont appliquées,

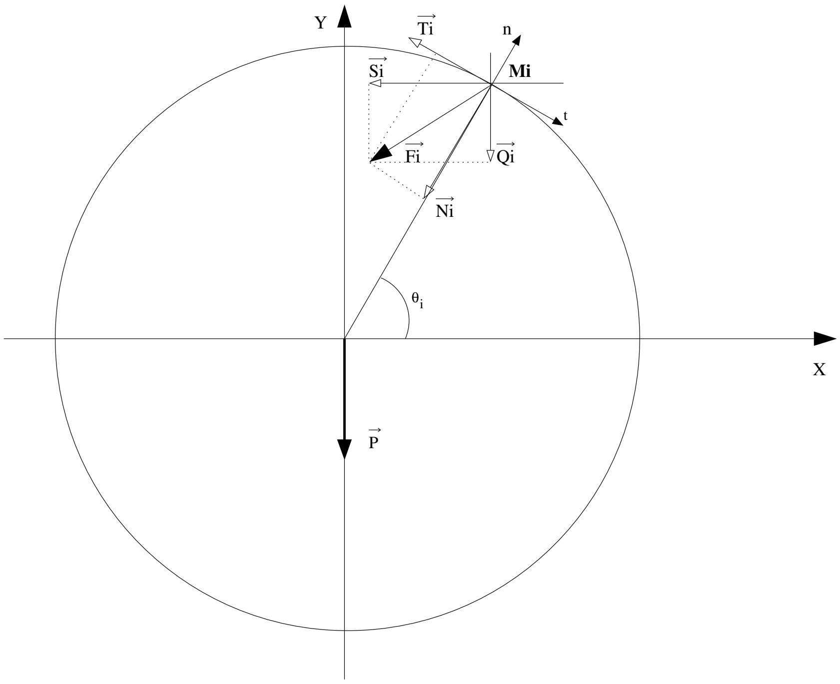


FIG. A.2 – N forces s'appliquant sur un grain

est donné par la loi de Newton (cf. Fig. A.2) :

$$\sum_{i=0}^{n} \overrightarrow{F_i} + \overrightarrow{P} = \overrightarrow{0} \tag{A.2}$$

De plus, nous imposons qu'il n'y ait pas de rotation sur le grain. Le produit tensoriel des contraintes doit donc être nul :

$$\sum_{i=0}^{n} \overrightarrow{OM_i} \wedge \overrightarrow{F_i} = 0 \tag{A.3}$$

La loi de Coulomb en chacun des contacts donne :

$$\forall_i \quad |T_i| < \mu |N_i| \tag{A.4}$$

Enfin, la condition d'unilatéralité des forces de contact :

$$\forall_i \quad N_i \leq 0 \tag{A.5}$$

Les forces $\vec{F_i}$ peuvent être décomposées en composantes normales et tangentielles ou en composantes horizontales et verticales. Les relations de conversion entre les deux repères

$(O, \vec{x}, \vec{y})$ et $(M_i, \vec{t}, \vec{n})$ sont les suivantes :

$$\left\{\begin{array}{lcl} q_i & = & N_i \sin(\theta_i) - T_i \cos(\theta_i) \\ s_i & = & N_i \cos(\theta_i) + T_i \sin(\theta_i) \\ T_i & = & -q_i \cos(\theta_i) + s_i \sin(\theta_i) \\ N_i & = & q_i \sin(\theta_i) + s_i \cos(\theta_i) \end{array}\right. \tag{A.6}$$

Dans les repères $(O, \vec{x}, \vec{y})$ et $(M_i, \vec{t}, \vec{n})$, l'équation (A.2) devient respectivement :

$$\sum_{i=0}^{n} s_i = 0 \qquad \sum_{i=0}^{n} (N_i \cos(\theta_i) + T_i \sin(\theta_i)) = 0 \tag{A.7}$$

$$\sum_{i=0}^{n} q_i + m.g = 0 \qquad \sum_{i=0}^{n} (N_i \sin(\theta_i) - T_i \cos(\theta_i)) + mg = 0 \tag{A.8}$$

De même, dans les repères $(O, \vec{x}, \vec{y})$ et $(M_i, \vec{t}, \vec{n})$, l'équation (A.3) s'écrit :

$$\sum_{i=0}^{n} (q_i \cos(\theta_i) - s_i \sin(\theta_i)) = 0 \qquad \sum_{i=0}^{n} T_i = 0 \tag{A.9}$$

Dans les repères $(O, \vec{x}, \vec{y})$ et $(M_i, \vec{t}, \vec{n})$, la contrainte de Coulomb sur les contacts (A.4) s'exprime :

$$\forall_i \quad |-q_i \cos(\theta_i) + s_i \sin(\theta_i)| < \mu |q_i \sin(\theta_i) + s_i \cos(\theta_i)| \qquad \forall_i,\ |T_i| < \mu |N_i| \tag{A.10}$$

Enfin, dans les repères $(O, \vec{x}, \vec{y})$ et $(M_i, \vec{t}, \vec{n})$, la contrainte d'orientation de la force normale (A.5) devient :

$$\forall_i \quad q_i \sin(\theta_i) + s_i \cos(\theta_i) \leq 0 \qquad \forall_i\ N_i \leq 0 \tag{A.11}$$

Des équations (A.10) et (A.11), les relations suivantes sont déduites :

$$\forall_i \quad q_i(\mu \sin(\theta_i) - \cos(\theta_i)) + s_i(\sin(\theta_i) + \mu \cos(\theta_i)) < 0 \qquad \forall_i\ T_i < -\mu N_i \tag{A.12}$$

$$\forall_i \quad q_i(\mu \sin(\theta_i) + \cos(\theta_i)) + s_i(-\sin(\theta_i) + \mu \cos(\theta_i)) < 0 \qquad \forall_i\ T_i > \mu N_i \tag{A.13}$$

Ainsi, un grain de masse m soumis à un champ gravitique g sur lequel est appliqué N forces de contact est à l'équilibre statique si et seulement si les relations : (A.8), (A.8), (A.9), (A.11), (A.13) et (A.13) sont vérifiées.

Les équations (A.8) et (A.8) permettent de déduire les relations suivantes :

$$\begin{aligned} \forall_j\, q_j &= -\sum_{i\neq j} q_i - mg & \text{(A.14)} \\ \forall_j\, s_j &= -\sum_{i\neq j} s_i & \text{(A.15)} \end{aligned}$$

En remplaçant dans (A.9), N relations qui lui sont équivalentes sont obtnues :

$$\forall_j \sum_{i\neq j}(q_i(\cos(\theta_i) - \cos(\theta_j)) + s_i(-\sin(\theta_i) + \sin(\theta_j)) - mg\cos(\theta_j) = 0 \quad \text{(A.16)}$$

A.2 Le modèle à 4 contacts fixes

A.2.1 L'empilement

Dans le cas du modèle de Clément et d'Eloy [Eloy et Clément 1997], les contacts latéraux inter-grains sont volontairement négligés pour des raisons de simplification du modèle (cf. Fig. A.1, le grain numéro 7 possède des contacts actifs uniquement avec les grains 8 et 14, mais pas avec les grains numéros 1 et 6).

A.2.2 Le système *grain*

Dans ce cas, nous pouvons donc simplifier le modèle décrit ci-dessus (A.1) en le ramenant à 4 forces de contacts (cf. Fig. A.3).

Les angles des contacts sont fixés et ont pour valeur, dans le sens trigonométrique à partir du contact en haut à droite : $\frac{\pi}{3}$, $\frac{2\pi}{3}$, $\frac{4\pi}{3}$ et $\frac{5\pi}{3}$. Nous avons les relations trigonométrique suivantes, en notant $\theta = \frac{\pi}{3}$:

$$\left\{ \begin{array}{lllllll} \cos(\frac{\pi}{3}) &=& \cos(\theta), & \sin(\frac{\pi}{3}) &=& sin(\theta) \\ \cos(\frac{2\pi}{3}) &=& -\cos(\theta), & \sin(\frac{2\pi}{3}) &=& sin(\theta) \\ \cos(\frac{4\pi}{3}) &=& -\cos(\theta), & \sin(\frac{4\pi}{3}) &=& -sin(\theta) \\ \cos(\frac{5\pi}{3}) &=& cos(\theta), & \sin(\frac{5\pi}{3}) &=& -sin(\theta) \end{array} \right. \quad \text{(A.17)}$$

Les forces qui s'appliquent sur la partie supérieure du grain sont notées en majuscule (S_1, Q_1, S_2 et Q_2) et celles qui s'appliquent sur la partie inférieure en minuscules (s_1, q_1, s_2 et q_2).

Nous pouvons ré-écrire les équations de la section précédente avec quatre contacts fixes.

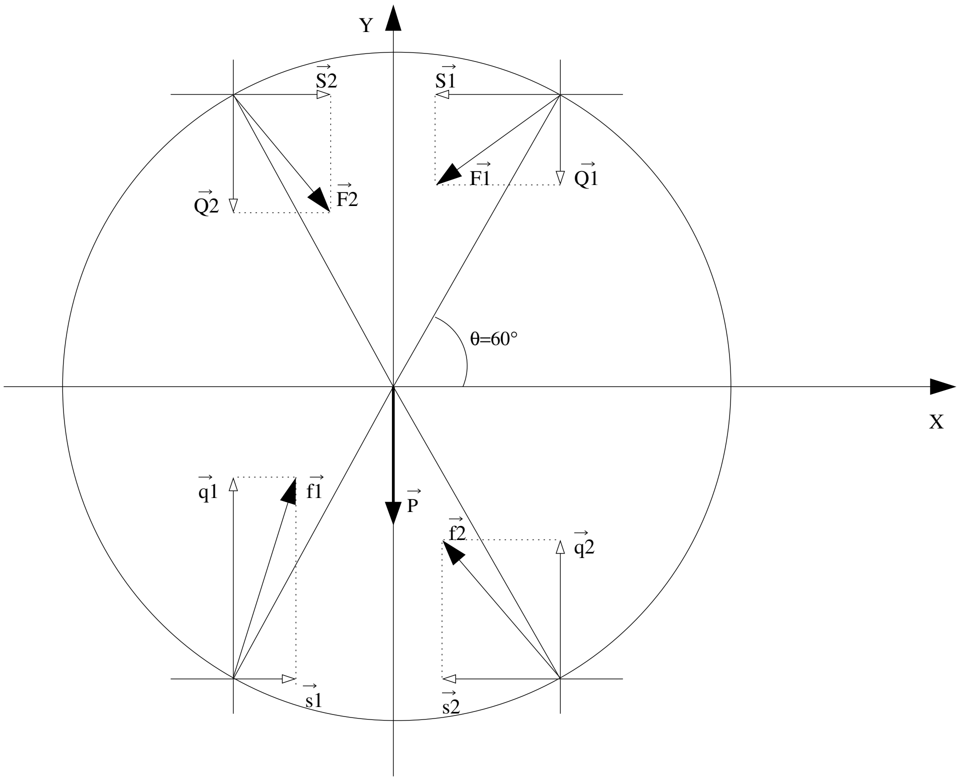


FIG. A.3 – Modèle à 4 forces de contact

L'équation concernant l'équilibre par translation horizontale (A.8) s'écrit maintenant :

$$S_1 + S_2 + s_1 + s_2 = 0 \tag{A.18}$$

De même, l'équation de l'équilibre par translation verticale (A.8) devient :

$$Q_1 + Q_2 + q_1 + q_2 + mg = 0 \tag{A.19}$$

L'équation garantissant l'équilibre par rotation (A.9) :

$$Q_1 - Q_2 - q_1 + q_2 + (-S_1 - S_2 + s_1 + s_2)\tan(\theta) = 0 \tag{A.20}$$

Et les conditions en chacun des contacts, d'une part l'orientation des forces normales (A.11) :

$$\left\{\begin{array}{lcl} Q_1 \tan(\theta) + S_1 & \leq & 0 \\ Q_2 \tan(\theta) - S_2 & \leq & 0 \\ -q_1 \tan(\theta) - s_1 & \leq & 0 \\ -q_2 \tan(\theta) + s_2 & \leq & 0 \end{array}\right. \tag{A.21}$$

et, enfin, la contrainte de frottement de Coulomb (A.13) et (A.13) :

$$\left\{\begin{array}{lll} Q_1(\mu\tan(\theta)-1)+S_1(\tan(\theta)+\mu) & < & 0 \\ Q_2(\mu\tan(\theta)+1)+S_2(\tan(\theta)-\mu) & < & 0 \\ q_1(-\mu\tan(\theta)+1)+s_1(-\tan(\theta)-\mu) & < & 0 \\ q_2(-\mu\tan(\theta)-1)+s_2(-\tan(\theta)+\mu) & < & 0 \end{array}\right. \tag{A.22}$$

et

$$\left\{\begin{array}{lll} Q_1(\mu\tan(\theta)+1)+S_1(-\tan(\theta)+\mu) & < & 0 \\ Q_2(\mu\tan(\theta)-1)+S_2(-\tan(\theta)-\mu) & < & 0 \\ q_1(-\mu\tan(\theta)-1)+s_1(\tan(\theta)-\mu) & < & 0 \\ q_2(-\mu\tan(\theta)+1)+s_2(\tan(\theta)+\mu) & < & 0 \end{array}\right. \tag{A.23}$$

Ainsi, un grain à 4 contacts fixés, de masse m et soumis à un champ gravitique de valeur g est à l'équilibre statique si et seulement si les relations : (A.18), (A.19), (A.20), (A.21), (A.22) et (A.23) sont vérifiées.

Si nous posons que les forces supérieures sont fixées, nous avons alors un système de 3 équations à 4 inconnues à résoudre :

$$\left\{\begin{array}{lll} s_1+s_2 & = & -S_1-S_2 \\ q_1+q_2 & = & -Q_1-Q_2-mg \\ q_1-q_2-\tan(\theta)(s_1+s_2) & = & Q_1-Q_2-\tan(\theta)(S_1+S_2) \end{array}\right. \tag{A.24}$$

sous les contraintes suivantes :

$$\left\{\begin{array}{lll} -q_1\tan(\theta)-s_1 & \leq & 0 \\ -q_2\tan(\theta)+s_2 & \leq & 0 \\ q_1(-\mu\tan(\theta)+1)+s_1(-\tan(\theta)-\mu) & < & 0 \\ q_2(-\mu\tan(\theta)-1)+s_2(-\tan(\theta)+\mu) & < & 0 \\ q_1(-\mu\tan(\theta)-1)+s_1(\tan(\theta)-\mu) & < & 0 \\ q_2(-\mu\tan(\theta)+1)+s_2(\tan(\theta)+\mu) & < & 0 \end{array}\right. \tag{A.25}$$

De (A.24), nous pouvons en déduire :

$$\left\{\begin{array}{lll} s_1+s_2 & = & -S_1-S_2 \\ q_2 & = & -Q_1-\frac{mg}{2}+\tan(\theta)(S_1+S_2) \\ q_1 & = & -Q_2-\frac{mg}{2}-\tan(\theta)(S_1+S_2) \end{array}\right. \tag{A.26}$$

Ainsi, dans ce modèle, les valeurs de q_1 et de q_2 sont complètement déterminées par les valeurs des forces supérieures. Il ne reste que les valeurs s_1 et s_2 à choisir dans les intervalles :

$$\left\{\begin{array}{lll} s_1 & \geq & -q_1\tan(\theta) \\ s_1 & > & \frac{q_1(-\mu\tan(\theta)+1)}{\tan(\theta)+\mu} \\ s_1 & < & \frac{q_1(\mu\tan(\theta)+1)}{\tan(\theta)-\mu} \quad \text{si } \mu < \tan(\theta) \\ s_2 & \leq & q_2\tan(\theta) \\ s_2 & > & \frac{q_2(-\mu\tan(\theta)-1)}{\tan(\theta)-\mu} \quad \text{si } \mu < \tan(\theta) \\ s_2 & < & \frac{q_2(\mu\tan(\theta)-1)}{\tan(\theta)+\mu} \end{array}\right. \tag{A.27}$$

C'est donc à partir des relations (A.26) et (A.27) que nous avons élaboré notre algorithme de résolution d'un grain. Il est important de noter que dans le cas où $\mu > \tan(\theta)$ ($\simeq 1.732$), le modèle ne donne pas de borne supérieure (resp. inférieure) pour le choix de s_1 (resp. de s_2).

Annexe B

La représentation des données au format XML

Sommaire

B.1 Les *conditions limites* de génération d'un tas

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Auteur : Laurent Breton
Date creation : 25/09/2001
Date maj : 25/09/2001
-->

<!-- Les conditions limites a‘ la re’alisation physique de l’empilement -->
<!ELEMENT conditions_limites (environement, paroi+, distribution_taille_grain,
                              methode_construction)>

<!-- Le nombre de grains de l’empilement -->
<!ATTLIST conditions_limites nb_grains CDATA #REQUIRED> <!ELEMENT environement EMPTY>
<!ATTLIST environement taille_x CDATA #REQUIRED>
<!ATTLIST environement taille_y CDATA #REQUIRED>
<!ATTLIST environement gravite_x CDATA #REQUIRED>
<!ATTLIST environement gravite_y CDATA #REQUIRED>

<!-- Une paroi est de’finie par une ligne et sa valeur de friction -->
<!ELEMENT paroi #EMPTY>
<!ATTLIST paroi x1 CDATA #REQUIRED>
<!ATTLIST paroi y1 CDATA #REQUIRED>
<!ATTLIST paroi x2 CDATA #REQUIRED>
<!ATTLIST paroi y2 CDATA #REQUIRED>
<!ATTLIST paroi friction CDATA "0.0">

<!-- La distribution des tailles des grains de l’empilement -->
<!ELEMENT distribution_taille_grain (distribution_constante|
                                     distribution_bi_modale|
                                     ditribution_aleatoire)>

<!-- Pour une distribution constante, la taille des grains est fixe -->
<!ELEMENT distribution_constante #EMPTY>

<!ATTLIST distribution_constante parametre CDATA "1.0">

<!-- Pour une distribution bi-modale, il y a 2 tailles de grains -->
<!ELEMENT distribution_bi_modale #EMPTY>

<!ATTLIST distribution_bi_modale valeur1 CDATA "0.5">
<!ATTLIST distribution_bi_modale valeur2 CDATA "1.0">

<!-- Pour une distribution aleatoire, les valeurs min et max de l’intervalle
     des valeurs a generer sont donnees -->
<!ELEMENT distribution_aleatoire (distribution_uniforme|distribution_gaussienne)>

<!ATTLIST distribution_aleatoire valeur_min CDATA "0.5">
<!ATTLIST distribution_aleatoire valeur_max CDATA "1.0">

<!-- La distribution uniforme est deja completement specifiee par les valeurs min et max de l’intervalle
     des valeurs a generer -->
<!ELEMENT distribution_uniforme #EMPTY>

<!-- La distribution gaussienne requiert ses deux parametres mu et sigma -->
<!ELEMENT distribution_gaussienne #EMPTY>

<!ATTLIST distribution_gaussienne mu CDATA "0.0">
<!ATTLIST distribution_gaussienne sigma CDATA "1.0">

<!-- La methode de construction est soit en pluie soit a‘ partir d’une position fixe a preciser -->
<!ELEMENT methode_construction (methode_point_fixe|methode_pluie)>


<!-- La methode du point fixe demande juste de preciser la position du point -->
<!ELEMENT methode_point_fixe #EMPTY>
```

```
<!ATTLIST methode_point_fixe position_x CDATA #REQUIRED>


<!-- La methode de construction en pluie requiert une distribution
     des valeurs des positions des grains qui vont tomber dans l'environement -->
<!ELEMENT methode_pluie (distribution_bi_modale|distribution_aleatoire)>
```

B.2 Le *tas*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
   Auteur : Laurent Breton
   Date creation :   23/03/2001
   Date maj :  19/06/2001
-->

<!ELEMENT tas (environement, couche*)>
<!ATTLIST tas largeur CDATA #REQUIRED>
<!ATTLIST tas hauteur CDATA #REQUIRED>
<!ATTLIST tas frottement CDATA #REQUIRED>

<!ELEMENT environement EMPTY>
<!ATTLIST environement taille_x CDATA #REQUIRED>
<!ATTLIST environement taille_y CDATA #REQUIRED>
<!ATTLIST environement gravite_x CDATA #REQUIRED>
<!ATTLIST environement gravite_y CDATA #REQUIRED>

<!ELEMENT couche (grain*)>
<!ATTLIST couche id CDATA #REQUIRED>
<!ATTLIST couche y CDATA #REQUIRED>

<!ELEMENT grain (fdc*)>
<!ATTLIST grain id CDATA #REQUIRED>
<!ATTLIST grain masse CDATA "1.0">
<!ATTLIST grain rayon CDATA "0.5">
<!ATTLIST grain x CDATA #REQUIRED>
<!ATTLIST grain y CDATA #REQUIRED>

<!ELEMENT fdc EMPTY>
<!ATTLIST fdc angle CDATA #REQUIRED>
<!ATTLIST fdc ref_grain CDATA #REQUIRED>
<!ATTLIST fdc x CDATA "0.0">
<!ATTLIST fdc y CDATA "0.0">
```

B.3 Un exemple d'un *tas*

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE tas SYSTEM "tas.dtd">
<tas largeur="10" hauteur="10" frottement="0.5">
  <environement taille_x="10.5" taille_y="8.79422863406"
  gravite_x="0.0" gravite_y="-1.0"/>
  <couche id="1" y="8.29422863406">
    <grain id="1" masse="1" rayon="0.5" x="0.5" y="8.29422863406">
      <fdc angle="4.18879020479" ref_grain="20"  x="0.4676889231" y="0.5"/>
      <fdc angle="5.23598775598" ref_grain="11"  x="-0.4676889231" y="0.5"/>
    </grain>
    <grain id="2" masse="1" rayon="0.5" x="1.5" y="8.29422863406">
      <fdc angle="4.18879020479" ref_grain="11"  x="0.1728344051" y="0.5"/>
      <fdc angle="5.23598775598" ref_grain="12"  x="-0.1728344051" y="0.5"/>
    </grain>
    <grain id="3" masse="1" rayon="0.5" x="2.5" y="8.29422863406">
      <fdc angle="4.18879020479" ref_grain="12"  x="0.5323267016" y="0.5"/>
```

```
      <fdc angle="5.23598775598" ref_grain="13"  x="-0.5323267016" y="0.5"/>
    </grain>
    <grain id="4" masse="1" rayon="0.5" x="3.5" y="8.29422863406">
      <fdc angle="4.18879020479" ref_grain="13"  x="0.4735412215" y="0.5"/>
      <fdc angle="5.23598775598" ref_grain="14"  x="-0.4735412215" y="0.5"/>
    </grain>
    <grain id="5" masse="1" rayon="0.5" x="4.5" y="8.29422863406">
      <fdc angle="4.18879020479" ref_grain="14"  x="0.650521808" y="0.5"/>
      <fdc angle="5.23598775598" ref_grain="15"  x="-0.650521808" y="0.5"/>
    </grain>
    ...
  </couche>
  <couche id="2" y="7.42820323028">
    <grain id="11" masse="1" rayon="0.5" x="1" y="7.42820323028">
      ...
    </grain>
  </couche>
</tas>
```


B.4 L'*expérimentation virtuelle*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
   Auteur : Laurent Breton
   Date creation :   15/03/2001
   Date maj :  04/04/2001
-->

<!-- Definition d'un schema d'experimentation  = un graphe muni de :
     - noeuds = composants
     - arcs = liens entre composants -->
<!ELEMENT schema_experimentation (liste_composants, liste_liens?,
               liste_resultats?)>
<!ATTLIST schema_experimentation date_creation CDATA #IMPLIED>
<!ATTLIST schema_experimentation date_maj CDATA #IMPLIED>
<!ATTLIST schema_experimentation auteur CDATA #IMPLIED>
<!ATTLIST schema_experimentation version CDATA #IMPLIED>

<!-- Une liste de composants = liste de noeuds du graphe -->
<!ELEMENT liste_composants ((composant_charger_disque|
              composant_sauver_disque|
              composant_commande_systeme|
              composant_xslt|
              composant_visualiseur|
              composant_groupe)+)>

<!-- Une liste de liens inter-composants = liste d'arcs du graphe -->
<!ELEMENT liste_liens (lien+)>

<!-- Un lien a besoin de 3 informations :
     - la ref. sur le composant d'entree et son numero de canal
     - idem pour le composant de sortie -->
<!ELEMENT lien EMPTY>
<!ATTLIST lien id_composant_entree CDATA #REQUIRED> <!--nodeA-->
<!ATTLIST lien id_composant_sortie CDATA #REQUIRED> <!--nodeB-->
<!ATTLIST lien id_canal_sortie CDATA #REQUIRED> <!--canal de nodeB-->
<!--ATTLIST lien ref_canal_entree CDATA #REQUIRED-->

<!-- Une liste de resultats = liste des fichiers sauvegardes au cours
     de l'execution du schema -->
<!ELEMENT liste_resultats (nom_fichier+)>

<!-- Definition du composant generique -->
<!ELEMENT composant_generique EMPTY >
<!ATTLIST composant_generique id CDATA #REQUIRED>
```

```
<!ATTLIST composant_generique nb_canaux_entree CDATA #REQUIRED>
<!ATTLIST composant_generique nb_canaux_sortie CDATA #REQUIRED>

<!-- Le composant charger_disque charge les fichiers du disque et les envoie
     un par un sur sa sortie. Si le parametre "noms_fichiers" reste a vide, on
     applique alors le filtre en chargeant tous les fichiers qui match le
     filtre. Sinon, "noms_fichiers" contient la liste des noms de fichiers
     a charger, separes par des blancs -->
<!ELEMENT composant_charger_disque (composant_generique)>
<!ATTLIST composant_charger_disque nom CDATA #IMPLIED>
<!ATTLIST composant_charger_disque classe_java CDATA #FIXED
         "LoadComponent">
<!ATTLIST composant_charger_disque repertoire CDATA ".">
<!--Anneli : plus simple lors de la construction du pathname-->
<!ATTLIST composant_charger_disque filtre CDATA ".xml">
<!ATTLIST composant_charger_disque noms_fichiers CDATA "">

<!-- Le composant sauver_disque prend le fichier qui arrive en entree et le
     sauve sur le disque. Il possede un increment qui lui permet de sauver
     plusieurs fichiers sous des noms differents = prefixe + numero + suffixe
     -->
<!ELEMENT composant_sauver_disque (composant_generique)>
<!ATTLIST composant_sauver_disque classe_java CDATA #FIXED "SaveComponent">
<!ATTLIST composant_sauver_disque nom CDATA #IMPLIED>
<!--Anneli : numero_courant disparait-->
<!--Anneli : incrementer disparait-->
<!ATTLIST composant_sauver_disque repertoire CDATA ".">
<!ATTLIST composant_sauver_disque prefixe_nom_fichier CDATA "resultat">
<!ATTLIST composant_sauver_disque suffixe_nom_fichier CDATA ".xml">

<!-- Le composant "commande systeme" prend en parametre le nom
     de la commande a executer. Le fichier d'entree est alors passe comme
     argument de cette commande. Il ne reste plus qu'a indiquer au composant
     ou aller chercher les fichiers resultats.-->
<!ELEMENT composant_commande_systeme (composant_generique) >
<!ATTLIST composant_commande_systeme classe_java CDATA #FIXED
"CommandLineComponent">
<!ATTLIST composant_commande_systeme commande CDATA #IMPLIED >
<!ATTLIST composant_commande_systeme nom CDATA #IMPLIED>
<!ATTLIST composant_commande_systeme repertoire_resultat CDATA ".">
<!ATTLIST composant_commande_systeme filtre_resultat CDATA ".xml">
<!--Anneli : a quoi sert fichier_resultat == ""? -->
<!--ATTLIST composant_commande_systeme fichier_resultat CDATA ""-->
<!ATTLIST composant_commande_systeme prefixe_fichier_resultat CDATA "">

<!-- Le composant xslt prend un nom de fichier en parametre qui contient les
     regles xslt a appliquer sur l'entree -->
<!ELEMENT composant_xslt (composant_generique) >
<!ATTLIST composant_xslt classe_java CDATA #FIXED "XSLTComponent">
<!ATTLIST composant_xslt nom CDATA #IMPLIED>
<!ATTLIST composant_xslt nom_fichier_regles_xslt CDATA "regles.xsl">
<!ATTLIST composant_xslt prefixe_fichier_resultat CDATA "sortieXSLT">

<!-- Un composant visualiseur ne comporte pas de parametre specifique (Batik).
     Il affiche le fichier svg qui est en entree -->
<!ELEMENT composant_visualiseur (composant_generique) >
<!ATTLIST composant_visualiseur classe_java CDATA #FIXED "VisualComponent">
<!ATTLIST composant_visualiseur nom CDATA #IMPLIED>

<!-- Un composant groupe comporte au moins 2 composants.
     Un composant groupe a au plus une entree et au plus une sortie
     (non implemente pour l'instant)  -->
<!ELEMENT composant_groupe (composant_generique, liste_composants,
                           liste_liens)>
<!ATTLIST composant_groupe classe_java CDATA #FIXED "GroupComponent">
<!ATTLIST composant_groupe nom CDATA #IMPLIED>
<!ATTLIST composant_groupe ref_composant_entree CDATA #IMPLIED>
<!ATTLIST composant_groupe ref_composant_sortie CDATA #IMPLIED>
```

B.5 L'*étude*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
   Auteur : Laurent Breton
   Date creation :   15/03/2001
   Date maj :  04/04/2001
-->

<!-- Racine de l'arbre, une etude comporte une arborescence
de problematiques auxquelles on essaie de repondre. -->
<!ELEMENT racine_etude (problematique*, ressources, bibliographie)>
<!ATTLIST racine_etude date_creation CDATA #IMPLIED>
<!ATTLIST racine_etude date_maj CDATA #IMPLIED>
<!ATTLIST racine_etude auteur CDATA #IMPLIED>
<!ATTLIST racine_etude version CDATA #IMPLIED>

<!-- Les ressources sont constituees de fichiers/composants
que l'on souhaite conserver afin d'etre reutilisees. -->
<!ELEMENT ressources (BD_composants, BD_fichiers)>

<!-- Une Base de Donnees de composants est constituee de
composants plus ou moins specifies, ranges par categorie -->
<!ELEMENT BD_composants (categorie_composant+)>

<!-- Une Categorie de composants regroupe tous les composants
d'une meme categorie -->
<!ELEMENT categorie_composant ((composant_charger_disque|
             composant_sauver_disque|
             composant_commande_systeme|
             composant_xslt|
             composant_visualiseur|
             composant_groupe|
             categorie_composant)*)>
<!ATTLIST categorie_composant nom CDATA #REQUIRED>



<!-- Une Base de Donnees de fichiers est constituee de fichiers,
ranges par categorie -->
<!ELEMENT BD_fichiers (categorie_fichier+)>



<!-- Une Categorie de fichiers regroupe tous les fichiers d'un meme type -->
<!ELEMENT categorie_fichier ((nom_fichier|categorie_fichier)*)>
<!ATTLIST categorie_fichier nom CDATA #REQUIRED>

<!-- Un nom de fichier est une balise vide avec juste un attribut nom -->
<!ELEMENT nom_fichier EMPTY>
<!ATTLIST nom_fichier nom CDATA #REQUIRED>


<!-- Une problematique =
     - une partie textuelle (enonces, etc)
     - une partie simulation (conditions limites, etc) . -->
<!ELEMENT problematique (enonce, cadre_physique, problematique*,
                        simulation*, resultat_simulation*, reponse?)>
<!ATTLIST problematique nom CDATA #REQUIRED>

<!ELEMENT cadre_physique (#PCDATA)>
<!ELEMENT enonce (#PCDATA)>
<!ELEMENT reponse (#PCDATA)>
<!ELEMENT resultat_simulation (#PCDATA)>
<!ATTLIST resultat_simulation idref CDATA #IMPLIED>
```

```
<!-- Definition d'une simulation -->
<!ELEMENT simulation (modele_theorique?, modele_effectif?,
               schema_experimentation*, resultat_experimentation*,
               notes*)>
<!ATTLIST simulation date_creation CDATA #IMPLIED>
<!ATTLIST simulation date_maj CDATA #IMPLIED>

<!ELEMENT modele_theorique EMPTY>
<!ATTLIST modele_theorique nom_fichier CDATA #REQUIRED >
<!ELEMENT modele_effectif EMPTY>
<!ATTLIST modele_effectif nom_fichier CDATA #REQUIRED >

<!ELEMENT resultat_experimentation (#PCDATA)>
<!ATTLIST resultat_experimentation idref CDATA #REQUIRED>
```

B.6 La charge verticale des grains d'un tas

Cette DTd est une version allégée de la DTD d'un tas. Les éléments «forces de contact» sont supprimés et remplacés par un seul attribut : la charge verticale du grain (ch_y).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
   Auteur : Laurent Breton
   Date creation :   23/03/2001
   Date maj :  19/06/2001
-->

<!ELEMENT tas_ch (environement, couche*)>
<!ATTLIST tas_ch largeur CDATA #REQUIRED>
<!ATTLIST tas_ch hauteur CDATA #REQUIRED>
<!ATTLIST tas_ch frottement CDATA #REQUIRED>

<!ELEMENT couche (grain*)>
<!ATTLIST couche y CDATA #REQUIRED>
<!ATTLIST couche ch_y CDATA #REQUIRED>

<!ELEMENT grain EMPTY>
<!ATTLIST grain rayon CDATA "0.5">
<!ATTLIST grain x CDATA #REQUIRED>
<!ATTLIST grain y CDATA #REQUIRED>
<!ATTLIST grain ch_y CDATA #REQUIRED>
```

B.7 Les règles de transformation d'un *tas* en un *tas_ ch*

```
<xsl:stylesheet version='1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform' >

<xsl:output method="xml" doctype-system="tas_ch.dtd" indent="yes"/>

<xsl:template match="tas">
   <xsl:element name = "tas_ch">
      <xsl:apply-templates select="@*"/>
      <xsl:apply-templates/>
   </xsl:element>

</xsl:template>

<xsl:template match="couche">
   <xsl:element name = "couche">
```

```
      <xsl:attribute name = "y">
         <xsl:value-of select="@y"/>
      </xsl:attribute>

      <xsl:attribute name = "ch_y">
         <xsl:value-of select="sum(./grain/fdc[number(@angle) &lt; 3]/@y) +
(sum(./grain/@masse) * number(/tas/environement/@gravite_y))"/>
      </xsl:attribute>

      <xsl:apply-templates/>

   </xsl:element>
</xsl:template>

<xsl:template match="grain">
<xsl:element name = "grain">

   <xsl:attribute name = "rayon">
      <xsl:value-of select="@rayon"/>
   </xsl:attribute>

   <xsl:attribute name = "x">
      <xsl:value-of select="@x"/>
   </xsl:attribute>

   <xsl:attribute name = "y">
      <xsl:value-of select="@y"/>
   </xsl:attribute>

   <xsl:attribute name = "ch_y">
      <xsl:value-of select="sum(./fdc[number(@angle) &lt; 3]/@y) +
(number(./@masse) * number(/tas/environement/@gravite_y))"/>
   </xsl:attribute>

</xsl:element>
</xsl:template>

<xsl:template match="@*|node()">
   <xsl:copy>
      <xsl:apply-templates select="@*"/>
      <xsl:apply-templates />
   </xsl:copy>
</xsl:template>

</xsl:stylesheet>
```

B.8 Les règles de transformation d'un *tas_ chn* au format SVG

```
<?xml version="1.0" ?>
<!DOCTYPE xsl:stylesheet>
<xsl:stylesheet
 version='1.0'
 xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>

<xsl:output method="xml" indent="yes"
doctype-public="-//W3C//DTD SVG 20000802//EN//"/>

<xsl:variable name="max_chn">
   <xsl:variable name="nodes" select="/tas_chn/couche/grain/@chn_y"/>
   <xsl:value-of select="number($nodes[not($nodes > .)])"/>
</xsl:variable>

<xsl:template match="tas_chn">
```

```
   <xsl:element name = "svg">
      <xsl:attribute name = "width">555px</xsl:attribute>
      <xsl:attribute name = "height">477px</xsl:attribute>
      <xsl:attribute name="preserveAspectRatio">xMidYMid meet</xsl:attribute>
      <xsl:attribute name = "viewBox">0, 0, <xsl:value-of select="./environement/@taille_x"/>
, <xsl:value-of select="./environement/@taille_y"/></xsl:attribute>

      <xsl:apply-templates/>
   </xsl:element>
</xsl:template>

<xsl:template match="grain">

<xsl:element name = "circle">

   <xsl:attribute name = "cx">
      <xsl:value-of select="@x"/>
   </xsl:attribute>

   <xsl:attribute name = "cy">
      <xsl:value-of select="/tas_chn/environement/@taille_y - @y"/>
   </xsl:attribute>

   <xsl:attribute name = "r">
      <xsl:value-of select="@rayon"/>
   </xsl:attribute>

   <xsl:variable name="couleur"><xsl:number value="floor((@chn_y div $max_chn) * 255)"/></xsl:variable>
   <xsl:attribute name = "style">fill:rgb(<xsl:value-of select="$couleur"/>,
                                         <xsl:value-of select="$couleur"/>,
                                         <xsl:value-of select="$couleur"/>)</xsl:attribute>

</xsl:element>
</xsl:template>
</xsl:stylesheet>
```

B.9 Un exemple d'un tas au format SVG

```
<?xml version="1.0" encoding="UTF-8"?>
<svg width="555px" height="477px" preserveAspectRatio="xMidYMid meet"
     viewBox="0, 0, 10.5, 8.79422863406">

   ...
    <circle cx="0.5" cy="5.696152" r="0.5" style="fill:rgb(61, 61, 61)"/>
    <circle cx="1.5" cy="5.696152" r="0.5" style="fill:rgb(157, 157, 157)"/>
    <circle cx="2.5" cy="5.696152" r="0.5" style="fill:rgb(173, 173, 173)"/>
    <circle cx="3.5" cy="5.696152" r="0.5" style="fill:rgb(114, 114, 114)"/>
    <circle cx="4.5" cy="5.696152" r="0.5" style="fill:rgb(48, 48, 48)"/>
    <circle cx="5.5" cy="5.696152" r="0.5" style="fill:rgb(171, 171, 171)"/>
    <circle cx="6.5" cy="5.696152" r="0.5" style="fill:rgb(110, 110, 110)"/>
    <circle cx="7.5" cy="5.696152" r="0.5" style="fill:rgb(185, 185, 185)"/>
    <circle cx="8.5" cy="5.696152" r="0.5" style="fill:rgb(57, 57, 57)"/>
    <circle cx="9.5" cy="5.696152" r="0.5" style="fill:rgb(161, 161, 161)"/>
   ...
</svg>
```

Résumé

En physique des milieux granulaires, des méthodes de simulations numériques sont utilisées notamment pour tenter de comprendre la connexion entre les mécanismes d'interaction des grains et l'obtention de lois macroscopiques fiables contrôlant leurs aspects mécaniques. Les méthodes généralement employées sont d'une complexité telle qu'une étude systématique de larges empilements et leur utilisation pour identifier des lois macroscopiques de comportement, restent un problème ouvert. Nous avons proposé une modélisation multi-agent originale, appelée GranuSolve, qui permet de résoudre des tas de sable bidimensionnels à l'équilibre statique. Cette méthode d'éco-résolution a, dans la pratique, une complexité linéaire avec le nombre de grains de l'empilement. Nous montrons également l'intérêt de la conception de groupes d'agents pour la résolution distribuée de ce type de problème. Cette efficacité a ouvert les portes au développement de GranuLab, un système d'aide à la découverte scientifique appliqué à la physique des milieux granulaires. Plus précisément, ce laboratoire virtuel permet des expérimentations basées sur les solutions générées par le composant GranuSolve. Cet environnement permet de représenter et de mémoriser la démarche scientifique d'un chercheur en physique granulaire. Ainsi, le système permet de poser des hypothèses, de réaliser des expérimentations pour vérifier celles-ci et enfin de proposer des lois. Nous avons choisi une approche componentielle associée au langage XML pour représenter et opérationnaliser ces expérimentations. Tout d'abord, cette approche a contribué à réaliser simplement un laboratoire virtuel à l'aide d'un nombre restreint de composants élémentaires. Ensuite, nous avons initié la création d'une représentation standardisée des données physiques relatives aux expérimentations. Enfin, après de nombreuses expérimentations, le système a permis de vérifier une théorie analytique du comportement de propagation et de diffusion d'un tas de sable.

Mots-clés: Aide à la découverte scientifique, Simulation multi-agent, Laboratoire virtuel, XML, Milieux granulaires

Abstract

In granular physics, numerical simulations are often used to better explain the link between the microscopic interaction mechanisms (e.g. contact forces between beads) and macroscopic laws (e.g. the maximum load at the bottom of a silo). Up to now, the complexity of the numerical simulations prevented physicists to study large piles and their mechanical properties. Thus, we have first proposed a multi-agent based simulation : GranuSolve. This eco-solving method can compute piles in a static equilibrium. The complexity of our algorithm is linear with respect to the number of grains. Then, we show how relevant it is to model groups of agents for the distributed resolution of our problem. Theses performances have paved the way for the development of GranuLab, a human-computer collaboration system for scientific discovery in the domain of granular physics. To be more precisely, this is a virtual laboratory for experiments based on the solutions given by the GranuSolve component. Within this environment, a granular physic researcher can represent and memorize the scientific processes he is walking through. So, the system allows the researcher to state hypotheses and realize experiments in order to establish these hypotheses as laws. We have chosen a componential approach based on the XML language to represent and operationalize theses experiments. At first, this approach has contributed to easily achieve a virtual laboratory with the help of a limited number of elementary components. Then, we have also initiated the creation of a standardized representation of physical data related to granular piles. Finally, the system has allowed the verification of an analytical theory related to the propagative and diffusive behavior of sand piles.

Keywords: Human-computer collaboration, Multi-agent based simulation, Virtual laboratory, XML, Granular physics