HAL
open science

Simulation numérique de la dynamique des systèmes discrets par décomposition de domaine et application aux milieux granulaires

Damien Iceta

To cite this version:

Damien Iceta. Simulation numérique de la dynamique des systèmes discrets par décomposition de domaine et application aux milieux granulaires. Matériaux et structures en mécanique [physics.class-ph]. Université Montpellier II - Sciences et Techniques du Languedoc, 2010. Français. NNT : . tel-00800891

HAL Id: tel-00800891

https://theses.hal.science/tel-00800891

Submitted on 14 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE MONTPELLIER 2
— Sciences et Techniques du Languedoc —

THESE

pour obtenir le grade de
DOCTEUR DE L'UNIVERSITE MONTPELLIER 2

Spécialité :	Mécanique et Génie Civil.
Formation doctorale :	Mécanique des matériaux et des milieux complexes, des structures et des systèmes.
Ecole doctorale :	Informations, Structures, Systèmes.

Simulation numérique de la dynamique des systèmes discrets par décomposition de domaine et application aux milieux granulaires.

---

par

Damien ICETA

Soutenue publiquement le 16/07/2010 devant le jury composé de :

MM.	Laurent Champaney	Professeur - ENS Cachan	Rapporteur
	Gery De Saxcé	Professeur - Université Lille 1	Rapporteur
	Mikael Barboteu	Maître de conférences - Université de Perpignan	Examinateur
	Hachmi Ben Dhia	Professeur - Ecole centrale Paris	Examinateur
	Mathieu Renouf	Chargé de Recherche - CNRS	Examinateur
	Pierre Alart	Professeur - Université Montpellier II	Directeur
	David Dureisseix	Professeur - Université Montpellier II	Directeur
	Michel Jean	Directeur de recherche émérite	Invité

Table des matières

1 Décomposition de domaine en calcul des structures **9**
1.1 Introduction 10
1.2 Méthodes de décomposition de domaine 12
1.2.1 Décomposition de domaine et calcul multiprocesseur 12
1.2.2 Ajout macroscopique et extensibilité 17
1.3 Extension aux problèmes non linéaires 18
1.4 Découpage du domaine 21
1.5 Objectifs 24

2 Deux algorithmes dédiés à la dynamique granulaire **25**
2.1 Un modèle non régulier pour les milieux granulaires 26
2.1.1 La dynamique des grains 27
2.1.2 Passage de l'espace des grains à l'espace des contacts 30
2.1.3 Relations de comportement et interaction 33
2.1.4 Problème de référence 35
2.1.5 Problème sous structuré 35
2.2 Un algorithme LATIN multidomaine 38
2.2.1 Méthode LATIN monoéchelle 38
2.2.2 Recherche des paramètres optimaux 41
2.3 Un algorithme GSNL multidomaine 43
2.3.1 GSNL versus LATIN (monoéchelle et monodomaine) 43
2.3.2 Stratégies de type GSNL en décomposition de domaine 44
2.3.3 Analogies algorithmiques des deux méthodes 48
2.4 Comparaison de performances sur un incrément de temps 48
2.4.1 Résultats dans un cas linéaire abstrait 48
2.4.2 Résultats pour un milieu granulaire 53
2.5 Implantations dans LMGC90 55
2.5.1 Échanges avec MPI 55
2.5.2 Structure et mise en place 59

2.5.3 Temps de calcul et nombre d'itérations 60
2.5.4 Résultats avec frottement 61

3 Introduction d'une échelle macro 71
3.1 Définition des champs macro 72
3.1.1 Une stratégie multiéchelle 72
3.1.2 Représentation macro 73
3.2 Algorithme de type LATIN multiéchelle 78
3.2.1 Étape locale. 78
3.2.2 Étape linéaire. 79
3.3 Algorithme de type GSNL multiéchelle 82
3.4 Faible efficacité de l'échelle macroscopique 84

4 Comment enrichir un algorithme GSNL multidomaine ? 87
4.1 Analyse de convergence de GSNL multidomaine 89
4.1.1 Méthodologie 89
4.1.2 Analyse de convergence dans le cas linéaire 92
4.2 Une tentative d'enrichissement 93
4.2.1 Algorithme enrichi 93
4.2.2 Analyse de convergence dans le cas linéaire 96
4.3 Analyse asymptotique et non régulière 99
4.3.1 Cas limite rigide avec GSNL standard 100
4.3.2 Cas limite rigide avec GSNL enrichi 100
4.3.3 Cas non régulier avec GSNL standard 103
4.3.4 Cas non régulier avec GSNL enrichi 106
4.4 Premières validations numériques 111
4.4.1 Comparaison des versions générique et enrichie 111
4.4.2 Stratégies de résolutions incomplètes 113

Introduction générale

Le Laboratoire de Mécanique et Génie Civil de Montpellier développe une partie de son activité dans le domaine de la simulation numérique en mécanique, avec une compétence particulière dans la gestion des interactions non régulières. Il a l'ambition nouvelle de se doter d'un outil numérique capable de traiter de "grands systèmes granulaires". Le travail développé dans cette thèse fait suite aux recherches menées par le laboratoire sur la décomposition de domaine de milieux discrets non réguliers [50], en se focalisant ici sur les milieux granulaires.

Les applications concernées seront essentiellement des systèmes granulaires de grande dimension et stables, c'est-à-dire sans écoulements collectifs de grains. Les développements de cette thèse ont donné lieu à des échanges avec la SNCF, intéressée pour augmenter ses capacités de calcul pour des simulations de ce type. En effet, la société a pour objectif la simulation du ballast de Lignes Grandes Vitesses soumises à un trafic intense, sur une longueur de plusieurs traverses, soit un milieu granulaire 3D sous chargement dynamique. D'autres types de simulations peuvent être cités, comme le déclenchement d'avalanche neigeuse et les maçonneries soumises à des chargements sismiques ou cycliques thermiques. De telles simulations en dynamique granulaire impliquent des interactions non régulières entre grains répartis sur l'ensemble du système à traiter, et des connectivités qui varient — on suppose faiblement — au cours de la simulation. Même sans écoulement, les grains en contact potentiel ne sont pas les mêmes tout au long de la simulation. En effet, le "maillage" reliant les centres des grains qui peuvent entrer en contact est susceptible d'évoluer localement.

Afin de répondre au besoin de simulations granulaires de grande taille, l'idée dans cette thèse est de se placer au carrefour de deux champs d'intérêts : la mécanique non régulière "Non Smooth Contact Dynamics (NSCD)" et les méthodes de décomposition de domaine "Domain Decomposition Methods (DDM)", et de voir dans quelle mesure il est possible de les combiner. La première a fait ses preuves pour le calcul de milieux discrets denses avec des interactions de contact, tandis que la seconde a permis l'augmentation des capacités de résolution sur des milieux continus.

La mécanique non régulière (NSCD) a été développée par J.J. Moreau et M. Jean durant les deux dernières décennies [34, 49]. Elle est adaptée à de nombreuses applications mais tout particulièrement quand des collections de corps rigides ou déformables sont assemblés dans un échantillon dense, sujet à des chargements dynamiques. Aussi, beaucoup d'attention a été portée à la façon dont la méthode converge et à la manière d'évaluer la qualité d'une simulation numérique. C'est une difficulté importante quand on étudie un échantillon de grande taille pour lequel le nombre et la nature des solutions est loin d'être unique. En conséquence, la simulation numérique doit être réalisée à l'aide d'une résolution implicite des forces de contact. Le LMGC est doté de la plateforme de calcul LMGC90[1] particulièrement adaptée au calcul avec NSCD pour les milieux granulaires. À chaque pas du problème d'évolution, toutes les contraintes cinématiques sont prises en compte en même temps que les équations du mouvement afin de déterminer toutes les forces d'interaction. Ceci permet de traiter les transferts d'efforts non locaux qui peuvent être impliqués dans des collisions multiples, contrairement à la dynamique moléculaire classique dans laquelle on considère le système en évolution comme une succession de collisions binaires [13]. Même si le coût de calcul peut être assez élevé, le gain reste important. Par exemple dans les milieux granulaires, les répartitions classiques de contraintes que l'on peut trouver dans un milieu continu ne sont plus réalistes. Les chaînes d'effort agissent entre les grains de deux façons : le réseau fort des forces de contact est porteur des contraintes de cisaillement, tandis que le réseau faible supportera plutôt la pression interne [52].

Initialement prévues pour résoudre des problèmes sur des milieux continus discrétisés, les méthodes de décomposition de domaine (DDM) se sont beaucoup développées ces vingt dernières années, dans un contexte de calcul multiprocesseur. Pour éviter d'avoir à résoudre les équations d'équilibre sur l'ensemble de la structure, ces méthodes utilisent la décomposition en sous domaines pour n'avoir plus qu'un ensemble de résolutions de problèmes locaux sur chacun des sous domaines. On peut alors projeter ces résolutions dans une étape, généralement nommée condensation sur les interfaces, quand le problème est linéaire, qui correspond à la résolution du raccord entre sous domaines voisins. On pourra citer parmi les différents développements les approches primales, duales et mixtes [40, 26, 27, 9]. La recherche de méthodes à taux de convergence plus élevés a conduit au développement d'algorithmes mettant en jeu des problèmes grossiers ou macroscopiques. En effet, l'extensibilité numérique des algorithmes n'est garantie qu'au prix de l'ajout de problèmes, dits grossiers,

[1] http ://www.lmgc.univ-montp2.fr/~dubois/LMGC90/

éventuellement issus d'une description multi-échelle des champs inconnus d'interface [8]. Rappelons qu'un algorithme de décomposition de domaine est extensible si le nombre d'interactions ne s'accroît pas quand le nombre de sous domaines augmente en même temps que la taille du problème complet. Des techniques issues de méthodes multigrilles ou d'éléments finis hiérarchiques permettent de propager l'information rapidement à l'ensemble des sous domaines [44][29]. L'utilisation d'un problème global sur l'ensemble de la structure a été réalisée dans la méthode FETI, pour laquelle les mouvements de solide rigide des sous domaines doivent être obligatoirement introduits, ou dans la " balancing domain decomposition method", où l'on ajoute la prise en compte des mouvements de solide rigide dans la phase de pré-conditionnement du solveur de type gradient conjugué.

Pour combiner ces deux approches, nous partirons de la méthode NSCD et nous y adapterons les méthodes de Décomposition de Domaine ainsi que leurs développements multiéchelles. En lien avec la méthode NSCD, nous développerons une stratégie de Décomposition de Domaine (DDM) en utilisant uniquement des solveurs génériques, robustes et non réguliers. Le premier est dérivé de la méthode LATIN initiée par Ladevèze [38]. Le second, développé par Moreau dans le cadre de la NSCD [49], peut être interprété comme un algorithme de Gauss Seidel Non Linéaire (NLGS) [35], un des objectifs étant d'implanter et de tester ces solveurs dans la plate-forme de calcul LMGC90.

Les algorithmes de Gradient Conjugué Projeté traitant donc un problème d'optimisation sous contraintes unilatérales [15, 47, 55] n'ont pas été retenus. Car si ils sont aussi utilisés par les méthodes de décomposition de domaine pour des problèmes linéaires (BDD [44], FETI [27]), ces méthodes, encore efficaces pour le contact sans frottement [55], sont difficiles à étendre à des problèmes de contact avec frottement et pour des comportements non réguliers plus généraux tels que la capillarité, l'adhésion. Leurs performances ne sont alors pas très bonnes ; par exemple l'approche Gradient Conjugué montre des difficultés de convergence quand le coefficient de frottement augmente [55]. N'ont pas été non plus retenues les extensions dites "directes" des méthodes de type FETI pour les problèmes sans [21, 5, 4] ou avec frottement [54], de même que les approches de Dostál sans [16, 17], ou avec frottement [32, 18], qui concernent des problèmes d'assemblage, l'ensemble de ces méthodes limitant le contact aux interfaces des sous domaines, mais ne pouvant traiter des contacts diffus et nombreux sur l'ensemble d'une structure.

Ici nous élaborons un algorithme pour des problèmes de dynamique non régulière qui vise à être numériquement extensible, en vue d'une parallélisation de calculs de

taille importante.

Dans une première partie, quelques méthodes de décomposition de domaine seront présentées, de façon succincte en faisant le point sur leurs différentes utilisations des quantités d'interface. Les possibilités d'extension à des problèmes non linéaires par de telles méthodes seront alors étudiées. Après un choix de décomposition du milieu que nous présenterons, les différents objectifs de la thèse en seront déduits.

Pour répondre à ces objectifs, dans une deuxième partie, on introduira la dynamique granulaire nécessaire à la mise en place d'algorithmes dédiés. Deux algorithmes de décomposition de domaine adaptés à un système discret de type granulaire seront alors mis en place, testés et comparés. Enfin nous verrons, dans le contexte de l'utilisation d'un logiciel fonctionnant avec NSCD, l'implantation de ces algorithmes et quelles sont les difficultés numériques associées.

Pour augmenter les capacités de calcul du logiciel nous mettrons en place une échelle macroscopique pour laquelle les deux solveurs seront adaptés. Enfin dans une dernière partie nous verrons comment il est possible d'enrichir un des deux algorithmes, de manière à obtenir un passage des informations à grande longueur d'onde, et d'en déduire éventuellement un comportement homogénéisé tangent.

Dans tout le mémoire, les non linéarités seront toutes non régulières. Nous utiliserons cependant indifféremment les vocables non linéaires et non réguliers.

Chapitre 1

Décomposition de domaine en calcul des structures

Sommaire

1.1 Introduction 10
1.2 Méthodes de décomposition de domaine 12
1.3 Extension aux problèmes non linéaires 18
1.4 Découpage du domaine 21
1.5 Objectifs 24

1.1 Introduction

Dans de nombreuses simulations numériques (ballast, géomatériaux), on a affaire à des systèmes tridimensionnels de grande dimension, mettant en jeu des comportements non linéaires voire non réguliers. Par exemple, si l'on vise comme simulations de très grands systèmes granulaires comme dix mètres de ballast ferroviaire soumis à un chargement cyclique dynamique [58], ou le comportement des arènes de Nîmes, ou l'aqueduc d'Arles soumis à un chargement sismique [53], un challenge en mécanique numérique est d'arriver à réaliser les calculs en des temps raisonnables. Des études précédentes se sont focalisées sur l'aspect calcul parallèle [55, 56], en utilisant des architectures à mémoire partagée, de manière à éviter les difficultés de sous structuration d'un milieu, qui est une collection de particules en réarrangement permanent, allant dans certains cas jusqu'à un écoulement fluide.

Le LMGC est doté d'une plate-forme de calcul LMGC90 particulièrement adaptée au calcul avec NSCD pour les milieux granulaires. Ce logiciel peut fonctionner en utilisant une architecture parallèle avec mémoire distribuée, mais cela ne permet pas de réaliser des simulations à une échelle suffisamment grande, tant du point de vue de la taille mémoire que de celui de l'efficacité et du temps de calcul. En mécanique des milieux continus, la résolution des problèmes linéaires en utilisant des méthodes de décomposition de domaine est plus efficace lorsqu'elle est associée à une architecture à mémoire distribuée (la géométrie et les calculs sont séparés entre les différents processeurs, chacun avec sa propre mémoire). De plus, les machines multiprocesseurs connaissent actuellement un fort développement et équipent dorénavant des ordinateurs de bureau classiques et non plus seulement les gros calculateurs. Pour en tirer parti, des algorithmes adaptés doivent être développés.

Une première approche en décomposition de domaine utilisant le solveur LATIN a été réalisée sur un milieu discret modèle [50] : un système de tenségrité mettant en jeu des corps déformables avec des interactions non régulières. Le comportement de cette structure est ainsi par certains côtés comparable à celui d'un milieu granulaire sans frottement. Le solveur mis en place pour ce milieu dans [50] est basé sur le solveur LATIN initialement proposé par Ladevèze dans [37]. L'utilisation qui en est faite est en quelque sorte dégénérée vu que le coté évolution globale en temps n'est pas utilisé (il s'agit uniquement d'une version incrémentale). Toutefois les inconnues principales intervenant dans le problème sont à la fois les champs de déplacement et d'efforts aux interfaces, et en ce sens cette approche sera qualifiée de mixte.

En s'inspirant de ces travaux ainsi que de l'ensemble des méthodes de décomposition

de domaine, des algorithmes adaptés aux milieux granulaires sont mis en place. Le lien avec les systèmes de tenségrité sera fait, et on justifiera le choix de découpage réalisé sur le milieu granulaire et conservé sur l'ensemble du mémoire. Ce choix, ainsi que le descriptif des méthodes de décomposition de domaine nous permettra de mettre en place les objectifs auxquels devra répondre la suite du mémoire.

1.2 Méthodes de décomposition de domaine

Les méthodes de décomposition de domaine vont inspirer les développements de cette thèse et vont être parcourues ici de manière succincte. Le principe n'est pas ici de décrire complètement les algorithmes étudiés, que l'on peut trouver dans la littérature citée, mais bien de voir les avantages et points particuliers de chacune des méthodes et dans quelle mesure elles sont applicables à un milieu granulaire. De plus le milieu granulaire étant, en termes de comportement, fortement éloigné de celui de milieux continus il ne s'agit pas ici de comparer les méthodes sur un critère de vitesse de convergence mais bien de voir quels ont été les développements réalisés et comment on pourrait s'en inspirer.

1.2.1 Décomposition de domaine et calcul multiprocesseur

L'efficacité des méthodes de décomposition de domaine, associées à un contexte de calcul multiprocesseur, est bien établie, que ce soit d'un point de vue théorique ou pratique quand on traite des systèmes linéaires issus de la discrétisation de problèmes continus [40, 26].

Selon [41] on peut séparer les méthodes de décomposition de domaine en deux classes, les méthodes avec et sans recouvrement.

Les méthodes avec recouvrement ont l'avantage de ne pas introduire de nouvelles grandeurs mécaniques. En revanche les quantités d'interface sont calculées plusieurs fois et en fonction de la taille du problème, elles peuvent alourdir considérablement les calculs. Parmi les méthodes avec recouvrement on peut citer la méthode alternée de Schwarz introduite dans [59].

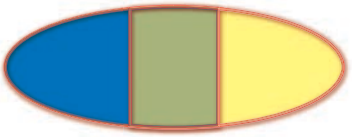


Fig. 1.1 – Deux sous domaines se recouvrant partiellement.

On a ici deux sous domaines se recouvrant partiellement. Dans des cas simples de recouvrement, l'algorithme de Schwarz présente une analogie avec la méthode de Gauss-Seidel par bloc et peut se ramener à une interprétation en tant que sous-partitionnement sans recouvrement avec un troisième sous domaine correspondant à la zone d'intersection [11].

Les résultats en terme de convergence peuvent être sensiblement différents de ceux des méthodes sans recouvrement, mais si l'on vise une application à un milieu granulaire où la non linéarité est diffuse dans toute la structure, celle-ci serait présente notamment dans la zone de recouvrement. On viendrait dupliquer des calculs non linéaires et risquer ainsi d'augmenter les temps de calculs.

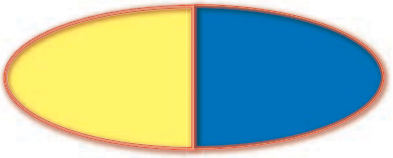


Fig. 1.2 – Deux sous domaines sans recouvrement.

Les méthodes sans recouvrement ont une interface de taille plus réduite et moins dépendante de la taille du problème traité, mais nécessite l'introduction de quantités d'interface. En effet le principe pour les méthodes sans recouvrement est de réduire le problème sur les inconnues d'interface. Ce problème peut alors être résolu soit par un solveur direct soit par une méthode itérative, par exemple une méthode du gradient conjugué pré-conditionné [40]. Il est alors primordial de trouver un pré-conditionneur adapté.

On peut à nouveau séparer ces méthodes en deux catégories : les méthodes primales et duales. Si on se restreint aux équations de la mécanique du solide. Soit on cherche à déterminer le champ de déplacement aux interfaces solution du problème soit le champ d'effort dual aux interfaces. La méthode primale permet de vérifier en permanence la continuité des déplacements aux interfaces et tendra à trouver une solution équilibrée en effort à convergence. La méthode duale quant à elle permet de vérifier en permanence l'équilibre en effort aux interfaces et va tendre vers une solution à déplacements continus à convergence. Enfin, les méthodes mixtes n'imposent ni la continuité des déplacements, ni l'équilibre des efforts au cours des itérations.

Méthodes primales.

Parmi les méthodes primales on peut citer la méthode de Schur primale initialement appelée méthode de sous-structuration. Cette méthode à la base était employée pour une analyse de structure où les non-linéarités sont confinées dans une zone a priori déterminée (un sous domaine) [36]. Cependant, de par le partitionnement du

domaine qu'elle entraîne, on a pu l'utiliser pour la création d'algorithmes adaptés aux calculateurs parallèles. En condensant les inconnues sur les interfaces, on pourra trouver une description détaillée dans [40]. On obtient alors un problème faisant intervenir la matrice de Schur et les seules inconnues d'interface.

On considère le problème de référence du type,

$$[K][q] = [f] \tag{1.1}$$

où K est la rigidité associé à un maillage élément fini, q le déplacement aux noeuds et f les efforts extérieurs correspondants. On découpe l'ensemble du domaine Ω en deux sous domaines Ω_E et Ω'_E. Le système peut être réécrit sous la forme,

$$\begin{bmatrix} K_{EE} & 0 & K_{E\Gamma} \\ 0 & K_{E'E} & K_{E'\Gamma} \\ K_{\Gamma E} & K_{\Gamma E'} & K_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} q_E \\ q_{E'} \\ q_\Gamma \end{bmatrix} = \begin{bmatrix} f_E \\ f_{E'} \\ f_\Gamma \end{bmatrix}. \tag{1.2}$$

Pour réaliser la condensation du problème on utilise une méthode de substitution pour les inconnues q_E et $q_{E'}$ (élimination de Gauss par bloc) de manière à réduire le problème sur les seules inconnues q_Γ,

$$[S][q_\Gamma] = [b] \tag{1.3}$$

avec

$$[S] = K_{\Gamma\Gamma} - \sum_{E=1}^{n_{sd}} K_{\Gamma E}(K_{EE})^{-1}K_{E\Gamma} \tag{1.4}$$

$$[b] = f_\Gamma - \sum_{E=1}^{n_{sd}} K_{\Gamma E}(K_{EE})^{-1}f_E \tag{1.5}$$

$[S]$ est appelée matrice du complément de Schur [7] et $[b]$ est le résultat de la condensation des efforts généralisés $[f]$ sur Γ et n_{sd} le nombre de sous domaines. On notera que pour revenir aux degrés de libertés internes aux sous domaines :

$$q_E = K_{EE}^{-1}(f_E - K_{E\Gamma}q_\Gamma) \tag{1.6}$$

Le problème peut alors être résolu par une méthode directe faisant intervenir l'assemblage complet de cette matrice de Schur. Celle-ci, bien qu'issue de la réduction sur les seules inconnues d'interface est pleine et peut devenir de taille imposante si le nombre de sous domaines est grand [19]. Il peut être envisagé des étapes supplémentaires dans la condensation pour arriver à des tailles de problèmes les plus réduites possibles [22].

L'autre façon de résoudre ce problème implique des méthodes itératives. On peut ainsi éviter d'expliciter la matrice de Schur grossissant avec le nombre de sous domaines. En revanche, le conditionnement de la matrice du problème réduit (matrice de Schur) est meilleur que celui de la matrice initiale du problème complet K. La différence entre ces deux conditionnements est montrée dans [40]. Cette différence a plaidé en faveur du développement d'algorithmes itératifs de résolution du problème condensé, de type gradient conjugué. Une étude plus détaillée de cette résolution pourra être trouvée dans [19]. Toutefois un phénomène important lors de cette résolution est à noter : il n'est pas nécessaire d'assembler $[S]$.

En effet, lors de la résolution, le produit $[S][q_\Gamma]$ est calculé [19] à partir de l'ajout de contributions locales $[S_E][q_{E\Gamma}]$; on n'a donc pas besoin d'assembler [S]. En revanche comme l'on veut s'assurer que le champ de déplacement est cinématiquement admissible (méthode primale) il faut prendre en compte la condition aux bords : $q_{E|\Gamma} = q_{E\Gamma}$. La résolution de ce problème de type Dirichlet fait intervenir K_{EE} qui devra être factorisée selon (1.6). On peut alors calculer les efforts de bords généralisés $S_E q_E = K_{\Gamma E} q_E + K^E_{\Gamma\Gamma} q^E_\Gamma$. On a assuré la continuité à l'interface et on fait tendre les efforts de bords généralisés vers l'équilibre entre sous domaines.

Le gradient conjugué sur le problème condensé peut s'interpréter comme un gradient conjugué pré-conditionné sur le problème de départ [19].

Méthodes duales.

La deuxième catégorie comporte les méthodes duales. Il s'agit ici de ne plus résoudre en déplacement mais en efforts sur les bords. Avec la méthode de Schur duale on arrive comme pour la résolution en déplacement à condenser le problème sur les seules inconnues d'interface qui sont cette fois-ci des efforts.

Si l'on considère à nouveau le problème de référence (1.1), mais qu'on ajoute cette fois un multiplicateur de Lagrange associé à la condition de continuité de déplacement [19], on peut alors exprimer les déplacements dans chaque sous domaine sous la forme (à partir du moment où K_E est inversible) :

$$q_E = (K_E)^{-1} \left(f_E - B_E^T F_\Gamma \right) \tag{1.7}$$

où B_E sont les matrices de restriction aux noeuds de bord à une convention de signe près et F_E le multiplicateur de Lagrange donnant les actions généralisées du sous domaine Ω_E sur $\Omega_{E'}$ ($F_E = -F_{E'} = F_\Gamma$).

Le problème après condensation s'écrit :

$$[\Delta][F_\Gamma] = [c] \tag{1.8}$$

avec $[\Delta] = \sum_{E=1}^{n_{sd}} B_E (K_E^{-1}) B_E^T = \sum_{E=1}^{n_{sd}} \Delta_E$

et $[c] = \sum_{E=1}^{n_{sd}} B_E (K_E)^{-1} f^E = \sum_{E=1}^{n_{sd}} c_E$
où $\sum$ représente l'assemblage des quantités locales aux sous domaines.

Si les matrices K_E sont toutes inversibles on peut alors utiliser l'exacte duale de la méthode primale. En effet les inconnues sont ici les efforts à l'interface et on peut toujours utiliser une méthode de gradient conjugué. Lors de la résolution on a cette fois besoin du produit $[\Delta][F_\Gamma]$ qui est calculé à partir de l'ajout des contributions locales : $[\Delta][F_\Gamma] = \sum_{E=1}^{n_{sd}} \Delta_E F_{E\Gamma}$ où $F_{E\Gamma} = B_E^T F_\Gamma$.

Le calcul se fera toujours en deux temps mais cette fois-ci la première étape, si l'on regarde l'équation (1.7), sera le calcul des déplacements correspondant au problème de Neuman avec efforts imposés $F_{E\Gamma}$. La deuxième étape n'est plus que la restriction des déplacements de sous domaines à l'interface $\Delta_E F_E = B_E q_E$ puis la somme de toutes ces contributions.

Δ_E permet alors à partir d'un champ d'effort de bord comme seul chargement, de trouver le champ de déplacement correspondant ; à un champ d'effort de bord, $[\Delta]$ fait donc correspondre le champ de saut de déplacement de bord. Il s'agit bien de la méthode duale de la précédente : au niveau des matrices locales, on a $\Delta_E = (S_E)^{-1}$

Dans le cas où les matrices K_E ne sont pas toutes inversibles (donc si il y a au moins un domaine flottant), on peut déterminer la solution du problème de Neumann précédent à un mouvement de solide rigide près [28]. (1.7) s'écrit alors :

$$q_E = K_E^+ \left(f_E - B_E^t F_\Gamma \right) + R_E \alpha_E \tag{1.9}$$

où K_E^+ est un inverse généralisé de la rigidité K_E, R_E l'ensemble des mouvements de solide rigide possibles du sous domaine Ω_E et α_E les coefficients d'une combinaison linéaire entre eux. Par conséquent on retrouve dans le problème condensé l'influence de ces mouvements de solide rigide [19]. Une condition de solvabilité est cependant nécessaire : les forces extérieures au sous domaine Ω_E doivent être globalement équilibrées.

Si l'on écrit l'équilibre sous forme discrétisée (il correspond à l'orthogonalité avec tous les champs de mouvement de solide rigide),

$$R_E^t \left(f_E - B_E^t F_\Gamma \right) = 0 \tag{1.10}$$

ce qui s'assemble en le système $R_B^T F_\Gamma = d$, où R_B est la matrice colonne des mouvements de solide rigide bord $B_E R_E$, et d la matrice colonne des déséquilibres $R_E^t f_E$.

On en déduit l'expression du nouveau système condensé, $\Delta F_\Gamma = c + R_B \alpha$ où α est le vecteur colonne des coefficients α_E , et c l'assemblage des charges condensées locales $B_E K_E^+ f_E$.

Dans le cas de sous domaines flottants, la méthode FETI a été développée de manière à traiter ces mouvements de solide rigide grâce à un projecteur, en gardant la logique d'utilisation de l'algorithme de gradient conjugué [29]. La projection se fera sur le sous espace des efforts de bord équilibrés par sous domaine. Un projecteur doit donc intervenir pour renvoyer un champ F_Γ dans $Ker(R_B^T)$. Par exemple [29], $p = I_d - R_B(R_B^T R_B)^{-1t} R_B$.

1.2.2 Ajout macroscopique et extensibilité

Dans cette thèse un critère important concernant les algorithmes étudiés sera l'extensibilité. Ce critère traduit la faculté d'un système à accroître sa capacité de calcul quand des ressources sont ajoutées. On cherche à adapter la puissance de la machine (par choix d'un certain nombre de processeurs) à la taille du problème. Travaillant à taille de problème local par processeur constant, un algorithme est dit numériquement extensible s'il n'y a pas (ou peu) de perte de convergence lorsque l'on augmente le nombre de processeurs, et donc la taille du problème à traiter.

L'extensibilité peut être prouvée de manière théorique quand un problème macroscopique est ajouté au pré-conditionneur de l'algorithme de gradient conjugué appliqué au problème d'interface [44].

L'avantage de la méthode FETI est que lors de la projection le système à résoudre est de taille réduite (nombre de degrés de liberté de chacun des sous domaines considérés comme un solide rigide) et que cette projection permet d'obtenir l'extensibilité du problème. On a introduit un problème macro qui correspond aux mouvements de solide rigide des sous domaines. Cette projection associée à un pré-conditionneur adapté (Dirichlet ou Lumped) permet de garantir un taux de convergence élevé et l'extensibilité des calculs [19].

De façon symétrique on peut obtenir l'extensibilité des méthodes primales en ajoutant un problème grossier et un pré-conditionneur adapté [44]. La phase supplémentaire consiste à corriger le déplacement obtenu par des mouvements de solide rigide sur chaque sous domaine : $q_E - R_E \alpha_E$ de façon, après prise de moyenne, à déterminer un nouveau résidu qui soit équilibré par sous domaine [19].

Dans le cas de problèmes de dynamique, on n'a plus de sous domaines flottants ; Schur dual peut s'appliquer directement mais n'est pas extensible, il est alors nécessaire d'ajouter un problème macro pour retrouver l'extensibilité [23]. Dans la suite, pour les milieux granulaires, la partie pré-conditionneur nous concernera moins que l'ajout d'un problème global permettant d'obtenir l'extensibilité en propageant l'information globalement.

L'enrichissement du problème grossier est apparemment nécessaire si l'on veut retrouver l'extensibilité sur des problèmes d'ordre 4 ; (FETI-DP [46, 24] ou les

méthodes duales BDDC [45]), les problèmes 3D exigent un enrichissement supplémentaire [25].

1.3 Extension aux problèmes non linéaires

L'extension des méthodes de décomposition de domaine à des problèmes non linéaires est généralement traitée comme la résolution en série de problèmes linéarisés. Chacun d'eux est résolu avec un solveur classique en décomposition de domaine fonctionnant comme une boîte noire [14, 6]. Peu d'études tentent de traiter la non linéarité et la décomposition de domaine dans un algorithme global ; nous avons pu trouver les références suivantes : [31, 9, 10, 21]. Une telle stratégie est difficile à étendre aux problèmes de type granulaire.

Dans cette thèse l'objectif est de pouvoir traiter des milieux granulaires ou des milieux discrets modèles comme la tenségrité dont la particularité est que leur non régularité est diffuse sur l'ensemble de la structure. De tels systèmes sont assez différent de structures discrétisées issues de milieu continus. Dans le milieu microfissuré considéré dans [50] le nombre de degrés de liberté liés à du contact est faible même si cet exemple implique un nombre important d'interfaces de contact.

Dans [51] une méthode de décomposition de domaine multiéchelle est utilisée à la fois pour simuler le comportement d'un milieu discret non régulier, et pour fournir un comportement numérique macroscopique de ce même système. De manière analogue au problème des "coins" ou "arêtes" des problèmes continus en 3D, des problèmes d'interfaces faibles apparaissent dans le problème macroscopique. Ils sont caractéristiques de la sous structuration d'un milieu discret. Cependant l'étude est limitée à la simulation quasi-statique d'une nappe de tenségrité de dimension importante. Le modèle dynamique utilisé pour simuler le comportement d'un milieu granulaire est assez différent comme on pourra le voir par la suite.

Un milieu modèle à non linéarité diffuse. Si un système discret est défini comme un ensemble de noeuds et de leurs interactions, alors la non linéarité ne peut apparaître que dans les relations de comportement des interactions. Un matériau granulaire avec du contact frottant entre ses grains est un exemple type de milieu non linéaire dans lequel chaque interaction a un comportement non linéaire : contact unilatéral avec frottement sec. Le problème est que pour commencer il n'est pas évident de concevoir les domaines issus de la décomposition et de mettre en place des tests de validation. En effet pour modéliser un milieu granulaire dans une bande large de comportements (de l'évolution quasi statique à l'écoulement dynamique) il est d'usage d'utiliser une formulation en vitesse-impulsion couplée à un schéma

d'intégration en temps de type "time stepping". Avec une telle formulation, la sous structuration du milieu granulaire doit être remise à jour régulièrement pour tenir compte des changements de connectivités.

Dans [50] il a donc été choisi d'étudier un problème statique mettant en jeu une structure discrète dite de tenségrité. Un tel système est plus proche des problèmes élastostatiques de milieux continus initialement traité par les méthodes de décomposition de domaine [40]. Les structures de tenségrité sont des assemblages de barres et de câbles utilisés en Génie Civil dont un exemple est donné Figure 1.3. Suivant une définition de Motro [50], ce sont des "systèmes autocontraints en état stable comportant un ensemble discontinu d'éléments en compression à l'intérieur d'un continuum d'éléments en tension". En conséquence le comportement unilatéral des câbles, chargeables uniquement en tension, dominant sur l'ensemble de la structure, confère aux grandes structures de tenségrité un caractère intéressant pour tester une approche non linéaire multiéchelle. Afin de les modéliser, il est effectivement nécessaire de prendre en compte la non régularité diffuse à travers toute la structure.

FIG. 1.3 – Needle Tower (Snelson 1968).

Le comportement des câbles, pouvant se tendre ou se détendre, est très proche de celui de grains rentrant en contact ou non. La principale simplification est que les connectivités restent les mêmes au cours de l'évolution dynamique. La loi de contact entre deux rigides fait intervenir l'impulsion de contact et les vitesses de ces deux corps et traduit la non inter-pénétrabilité des deux corps. La loi de comportement d'un câble relie tension et allongement (déplacement relatif) et exprime le fait qu'en

sollicitation de traction uniquement celui-ci a une certaine rigidité.

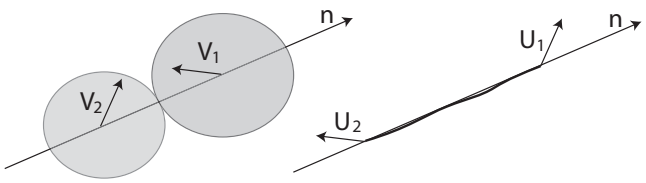


Fig. 1.4 – Conditions unilatérales : contact entre particules et inextensibilité d'un câble.

Les graphes correspondant aux comportements sont dessinés sur la Figure 1.5.

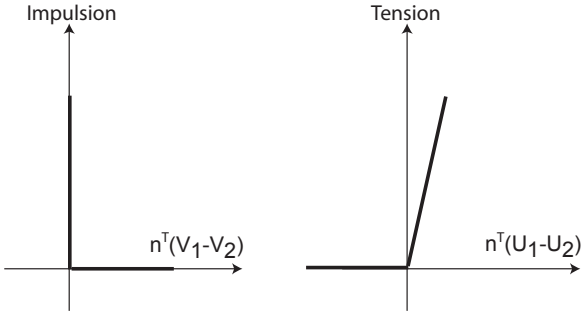


Fig. 1.5 – Loi de comportement de contact unilatéral entre particules, à gauche, et d'un câble extensible, à droite.

1.4 Découpage du domaine

La stratégie de calcul utilisée va reposer sur un partitionnement spatial du domaine en un ensemble de sous structures et interfaces, chacune de ces entités possédant ses propres variables et ses propres équations. Le découpage adapté d'une structure afin d'utiliser une approche en décomposition de domaine est un partitionnement géométrique du domaine occupé par cette structure. Dans le cas d'une structure discrète, il y a deux façons classiques de réaliser ce partitionnement : (i) distribuer les éléments, liens, ou interactions parmi les sous structures (Figure 1.7), ou (ii) distribuer les noeuds, ou grains parmi les sous structures (Figure 1.8). D'un point de vue topologique, elles correspondent au découpage de deux graphes différents : le graphe de connectivité ou son dual, celui des noeuds.

Dans notre cas, de la même façon que dans [50], puisque la non régularité peut apparaître dans les interactions entre grains, on choisit de distribuer ces interactions parmi les sous structures comme sur la Figure 1.7.

Cette option correspond à celle déjà utilisée par [6] pour les milieux cellulaires et [20] pour les systèmes de tenségrité. Elle se justifie dans tous ces cas, comme pour les milieux granulaires, par le fait que la non régularité (de contact) est diffuse à travers l'ensemble du domaine d'étude, et qu'elle sera donc forcément présente dans les sous domaines. Autant donc isoler la non régularité dans les sous domaines et chercher à conserver des interfaces parfaites facilitant par là-même la définition de quantités homogénéisées (donc régulières) à ces interfaces. Pour une approche "duale", avec non régularité aux interfaces et régularité dans les sous domaines, se référer à [21] et [39].

D'un point de vue pratique, l'interface est donc discrète et constituée des centres de masse de grains ayant au moins deux contacts avec des grains de sous domaines différents. Un algorithme de partitionnement doit éviter tous les cas pathologiques inhérents aux milieux granulaires. On définit au préalable un découpage du domaine total en sous domaines réguliers a priori parallélépipédiques, définissant des interfaces rectilignes. Les contacts sont ensuite répartis entre les sous domaines (on procède en distribuant le point milieu entre les centres de masse des grains en interaction, selon leur position). Ensuite on repère les grains qui ont des contacts avec deux sous domaines adjacents. On voit tout de suite surgir les difficultés liées à des grains appartenant à plus de deux sous domaines. Une réflexion sur ces interfaces dites faibles à déjà été entamée dans [3] dans le cadre des systèmes de tenségrité. Cependant la stratégie de décomposition en granulaire s'avère plus complexe que pour les systèmes de tenségrité, le plus souvent réguliers. Elle exige une grande rigueur dans la démarche et les tests réalisés dans ce mémoire n'ont certainement pas révélé toutes les situations pathologiques.

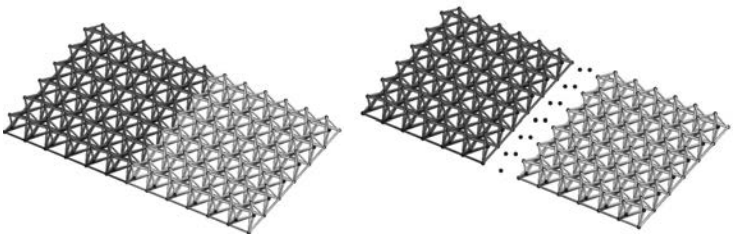


FIG. 1.6 – Partitionnement en sous domaines (à gauche) et interface discrète parfaite entre sous structures (à droite) pour une nappe de tenségrité.

Un changement important est qu'au lieu d'avoir des noeuds sans masse ni inertie aux interfaces, on a des grains dotés d'une masse. Les grains d'interface sont "coupés" en deux, chacune des deux parties appartenant à une sous structure différente. La masse et l'inertie de ces grains sont réparties lors de la construction des matrices de masse pour chacune des sous structures. Cet aspect diffère de la stratégie de décomposition de domaine pour les systèmes de tenségrité où la masse était prise en compte dans les liens eux-mêmes (barres essentiellement) donc au sein des sous structures et non aux interfaces.

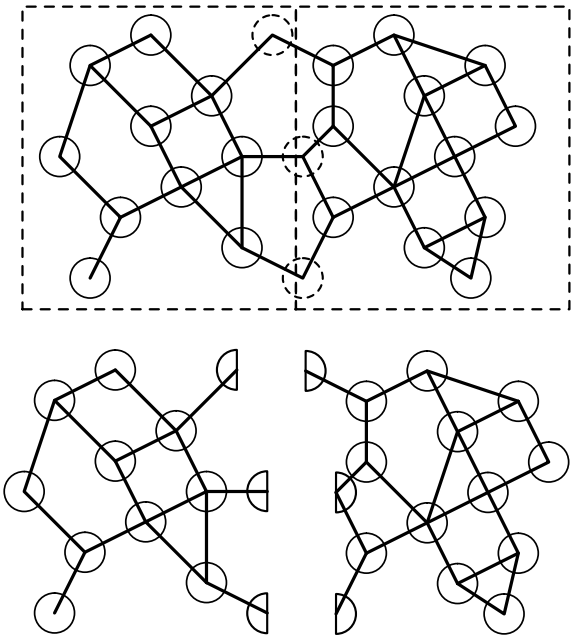


FIG. 1.7 – Partitionnement en sous domaines : distribution des interactions.

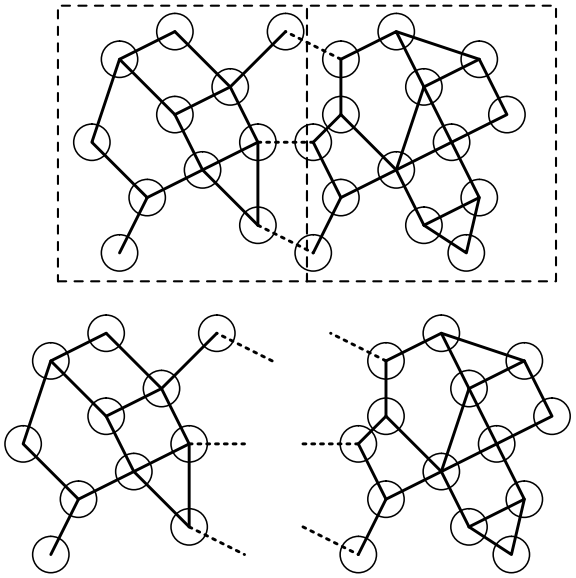


FIG. 1.8 – Partitionnement en sous domaines : distribution des grains.

L'interface entre sous domaines est définie comme l'ensemble de ces grains d'interface, joignant une sous structure à une autre. La non régularité est ainsi localisée à l'intérieur des sous structures uniquement. Ce choix de modélisation est identique à [6] et en quelque sorte le dual de celui proposé dans [39] où les non linéarités sont isolées aux interfaces. Une relation de "recollement" parfait des grains d'interface est donc à adjoindre au problème multidomaine.

1.5 Objectifs

Mise en place d'algorithmes en décomposition de domaine adaptés à un système discret de type granulaire. Les travaux précédents au sein du laboratoire ont permis de mettre en place un algorithme muldidomaine, multiéchelle adapté à un milieu discret modèle, un système de tenségrité. Une des premières questions, à laquelle on a commencé à répondre dans cette première partie, est : la méthode mise en place est-elle applicable au milieu granulaire et quels aménagements sont à prévoir d'une part pour faire une résolution multidomaine mais également pour avoir une double échelle (micro-macro) ?

Utilisation des algorithmes en décomposition de domaine pour paralléliser une plate forme de calcul existante. Une utilisation optimale de la démarche proposée sur les milieux granulaires demandera des améliorations algorithmiques, une adaptation à un environnement multiprocesseur, le recours à une stratégie de reconfiguration dynamique de la sous structuration. Les développements utiliseront la plate-forme logicielle LMGC90, dédiée à la simulation numérique des milieux discrets mais sans décomposition de domaine. Cette plate-forme fonctionnant déjà sous OpenMP pour faire les gros calculs en mémoire partagée, il est souhaitable pour diminuer la taille en mémoire, de combiner la décomposition de domaine avec une librairie d'échanges de messages, en l'occurence MPI, afin que chaque sous domaine soit traité par une instance de la plate-forme avec une taille mémoire réduite.

Optimisation des algorithmes. Une fois la décomposition de domaine fonctionnelle et permettant d'alléger les calculs, l'étape ultérieure consiste à améliorer la convergence des algorithmes en ciblant prioritairement les outils classiques de décomposition de domaine que sont, l'introduction d'une échelle macro, celle d'un problème grossier et son enrichissement. Les spécificités de notre problème nous amèneront à envisager d'autres stratégies.

Chapitre 2

Deux algorithmes dédiés à la dynamique granulaire

Sommaire

2.1 Un modèle non régulier pour les milieux granulaires . . 26
2.2 Un algorithme LATIN multidomaine 38
2.3 Un algorithme GSNL multidomaine 43
2.4 Comparaison de performances sur un incrément de temps 48
2.5 Implantations dans LMGC90 55

Dans cette partie on va s'attacher à décrire le milieu granulaire de manière à poser le problème à traiter dans la suite. Ce problème dit de référence va être utilisé pour comparer deux solveurs. Les deux solveurs sont basés sur la méthode LATIN pour l'un et GSNL pour le second. Ces algorithmes moins efficaces que par exemple, une méthode de gradient conjugué, quand ils sont appliqués à un milieu continu discrétisé ou à un problème de contact sans frottement, ont l'avantage d'être robustes et de fonctionner lorsqu'on introduit le frottement [55].

En effet le réel objectif est ici de traiter des problèmes granulaires avec frottement voire avec des interactions plus complexes (cohésion/décohésion, capillarité...). Dans un premier temps on pose le problème de référence. Seul le cas sans frottement sera traité dans un premier temps. Ce problème de référence plus simple à traiter numériquement sera étudié afin de comparer les solveurs sous MATLAB. Ensuite, dans la dernière partie de ce chapitre on intégrera à nos tests le cas avec frottement, en mettant en évidence l'implantation en parallèle dans une plate-forme de calcul dédiée au traitement du contact.

2.1 Un modèle non régulier pour les milieux granulaires

Dans le cadre d'une modélisation granulaire basée sur une collection de corps rigides, deux repères sont nécessaires, l'un lié au corps, dit global, pour écrire la dynamique, l'autre lié au contact, dit local, pour exprimer les interactions de contact. Pour cela des minuscules seront utilisées pour toutes les grandeurs exprimées dans une base locale de contact. Les majuscules seront quant à elles réservées aux grandeurs exprimées dans le repère des grains.

Les exposants ^{c} symbolisent le fait que la grandeur est issue d'un contact entre solide, on utilisera l'exposant ^{d} pour les autres. G est le centre de masse du corps.

Pour des raisons de simplicité on considérera des contacts unilatéraux sans frottement dans un premier temps.

Notation	Objet	Repère
F^c	Résultante de l'effort dû à un contact	grain
F^d	Résultante des actions extérieures	grain
M^c	Moment au centre de masse de l'effort dû à un contact	grain
M^d	Moment au centre de masse des actions extérieures	grain
P	Impulsion due à un contact	grain
K	Moments d'impulsion de contact au centre de masse	grain
r	Impulsion de contact	contact
k	Moment d'impulsion de contact au point de contact	contact
U	Vitesse du centre de masse pour un grain	grain
v	Vitesse relative entre deux grains	contact
Ω	Vitesse de rotation d'un grain dans la base principale d'inertie	grain
ω	Vitesse de rotation relative entre deux grains	contact
$\begin{pmatrix} v \\ \omega \end{pmatrix}$	Quantités cinématiques relatives entre deux grains	contact
$V = \begin{pmatrix} U \\ \Omega \end{pmatrix}$	Vitesse et vitesse de rotation	grain
$V^- = \begin{pmatrix} U^- \\ \Omega^- \end{pmatrix}$	Vitesse et vitesse de rotation juste avant l'instant du choc	grain
$R = \begin{pmatrix} P \\ K \end{pmatrix}$	Torseur des efforts de contact	grain
$R^d = \begin{pmatrix} P^d \\ K^d \end{pmatrix}$	Torseur des efforts donnés extérieurs	grain

TAB. 2.1 – Notations générales.

2.1.1 La dynamique des grains

Chaque grain (indicé par i), interagit avec ses voisins (indicés par $\bar{i}$) par un ensemble fini de contacts ponctuels. Sa masse est notée m_i, et son centre de masse G_i. Son opérateur d'inertie au point G_i est J_{iG}.

Au contact α, localisé au point M_α, les deux corps entrant en contact sont

nommés par convention candidat et antagoniste. Au point M_α le vecteur directeur normal n_α est dirigé vers le corps antagoniste.

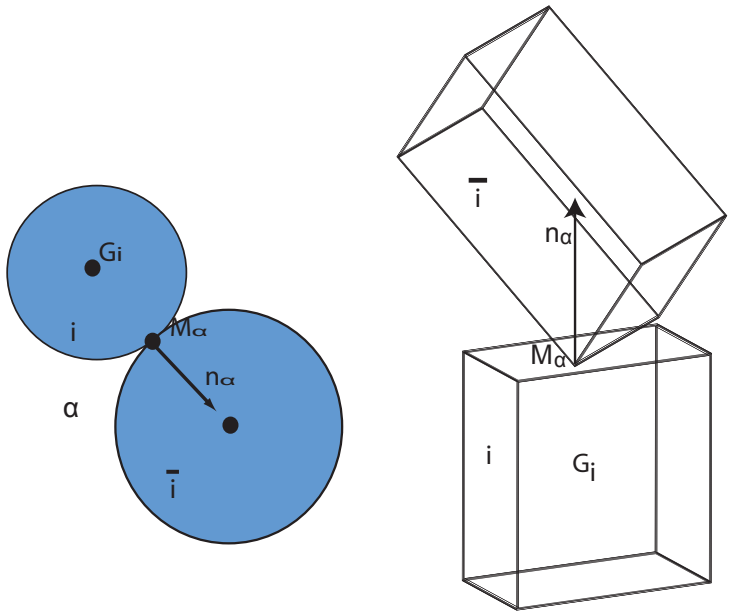


FIG. 2.1 – Grains et contacts.

La dynamique des corps est décrite par 6 paramètres en 3D : 3 pour les composantes du vecteur vitesse U_i du centre de masse et 3 pour les composantes du vecteur taux de rotation Ω_i.

Afin de calculer l'évolution du système dans le temps, un schéma d'intégration en temps est nécessaire et fournit une discrétisation des différentes équations différentielles liées à la dynamique. Dans le cas régulier, il est possible d'utiliser n'importe quel schéma d'intégration pour les équations différentielles, ces schémas ayant diverses propriétés en termes de précision, de conservation de l'énergie, de stabilité, de convergence [33]. Dans le cas non régulier [57], le problème est différent car les solutions perdent les propriétés nécessaires à la construction des schémas d'ordre élevé : dérivabilité, continuité.

À cause du très grand nombre de contacts mis en jeu dans un milieu granulaire dense, un schéma de type "time-stepping" (où la taille des pas de temps est fixée à l'avance) est choisi plutôt qu'un schéma "event-driven" (où chaque collision est isolée dans un unique pas de temps), qui conduirait à des pas de temps trop petits.

Plus spécifiquement, le schéma "event-driven" est incapable de prendre en compte des contacts simultanés [30, 42, 43].

Un schéma "time-stepping" donne une équation discrète de la dynamique mettant en jeu des impulsions dans son membre de droite, c'est à dire l'intégrale sur $]t^-, t^+[$ des quantités d'efforts et de moments. Les quantités cinématiques et sthéniques devront être déterminées à la fin du pas de temps t^+.

En intégrant entre 2 instants t^- et t^+, en supposant que les résultantes et moments des efforts extérieurs ont des limites finies non nulles lorsque la durée de l'intervalle de temps tend vers 0, on obtient l'expression de la discontinuité du vecteur quantité de mouvement et du moment cinétique,

$$\left\{ \begin{array}{c} m\left[\left[U_{(G_i,i/Rg)}\right]\right]_t \\ \left[\left[\delta_{(G_i,i/Rg)}\right]\right]_t \end{array} \right\}_{G_i} = \left\{ \begin{array}{c} \int_{t^-}^{t^+} F_i^c(\bar{i} \to i)dt + \int_{t^-}^{t^+} F_i^d(\bar{i} \to i)dt \\ \int_{t^-}^{t^+} M_{i_G i}^c(\bar{i} \to i)dt + \int_{t^-}^{t^+} M_{i_G i}^d(\bar{i} \to i)dt \end{array} \right\}_{G_i} \tag{2.1}$$

Ce qui s'écrit aussi,

$$\left\{ \begin{array}{c} m_i\left(U_i^+ - U_i^-\right) \\ J_{iG}\left(\Omega_i^+ - \Omega_i^-\right) \end{array} \right\}_{G_i} = \left\{ \begin{array}{c} P_i + P_i^d \\ K_i + K_i^d \end{array} \right\}_{G_i} \tag{2.2}$$

où les actions extérieures agissant sur le système sont : l'impulsion de contact résultante P_i, l'impulsion des actions autres que les contacts $P_i^d = \int_{t^-}^{t^+} F_i^d dt$ (où F_i^d est la résultante des actions autres que les contacts), le moment d'impulsion de contact en G_i, K_i, le moment d'impulsion des actions autres que les contacts exprimé en G_i, $K_i^d = \int_{t^-}^{t^+} M_i^d dt$ (où M_i^d est le moment des actions autres que les contacts en G_i).

En assemblant masse et inertie dans une matrice locale M_i, les variables cinématiques U_i et Ω_i dans un vecteur local V_i, les impulsions P_i et K_i dans un vecteur R_i, et finalement les quantités données P_i^d et K_i^d dans un vecteur R_i^d, l'admissibilité dynamique se synthétise en,

$$M_i V_i^+ = R_i + R_i^D + M_i V_i^-. \tag{2.3}$$

Pour des raisons de simplicité l'exposant + sera omis dans la suite. Ces vecteurs, associés à chaque grain, peuvent être assemblés en des vecteurs globaux de manière à réécrire la dynamique du milieu granulaire dans une équation où la matrice M est diagonale, si les coordonnées globales du vecteur taux de rotation sont exprimées dans la base d'inertie principale de chaque grain,

$$MV = R + R^D + MV^-. \tag{2.4}$$

Dans la suite on simplifiera à nouveau cette équation en considérant que la partie MV^- est incluse dans les impulsions données R^d. A chaque itéré on connaît :

$$R^d = R^D + MV^-, \tag{2.5}$$

et au final la dynamique est résumée dans l'expression,

$$MV = R + R^d. \tag{2.6}$$

2.1.2 Passage de l'espace des grains à l'espace des contacts

Pour les milieux granulaires on a régulièrement à faire le passage entre l'espace dit des grains (sur lequel on vient d'écrire la dynamique), et l'espace des contacts dans lequel on retrouve les lois d'interaction généralement écrites dans un repère local dépendant de chaque contact. Pour cela on met en place un opérateur noté H caractéristique du graphe des contacts entre les grains.

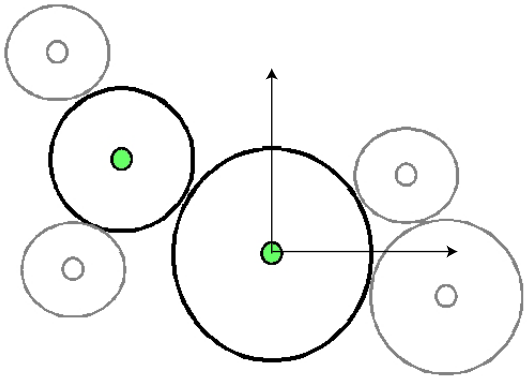


FIG. 2.2 – Repère lié aux grains.

Dans l'espace associé au grain i, les deux grandeurs à déterminer à $t = t^+$ sont les vitesses des grains V et les impulsions R tandis que dans l'espace associé au contact α, ce seront les vitesses relatives entre grains v et les impulsions de contact r. Ceci signifie que les repères de contact sont figés sur l'intervalle $]t^-, t^+]$.

Construction de la matrice de passage H.

Soit U la vitesse du centre de gravité, dans le repère global R_g, v la vitesse relative entre deux grains A et B en contact au point P, dans le repère local. La matrice H assure le passage du repère local au repère global [57].

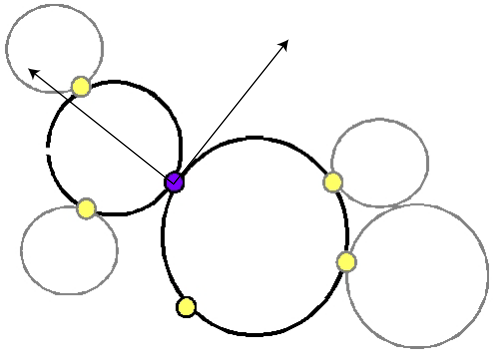


FIG. 2.3 – Repère lié aux contacts.

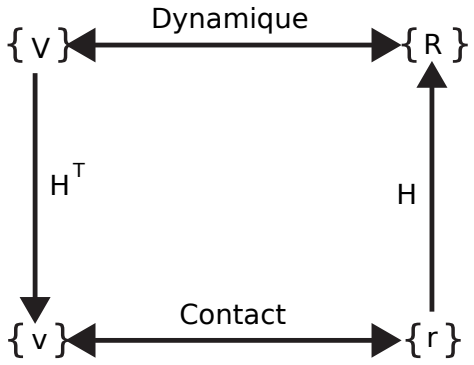


FIG. 2.4 – Diagramme commutatif.

$$\begin{aligned} v &= U_{(P\in A/Rg)} - U_{(P\in B/Rg)} \\ &= \left(U_{(G_A\in A/Rg)} - U_{(G_B\in B/Rg)}\right) + \left(\Omega_{(A/Rg)} \wedge G_AP - \Omega_{(B/Rg)} \wedge G_BP\right) \end{aligned} \tag{2.7}$$

On va chercher à exprimer v dans la base locale : Soit $\Omega_{(A/Rg)} = \Omega_\alpha\alpha_m + \Omega_\beta\beta_m + \Omega_\gamma\gamma_m$, où $(\alpha_m, \beta_m, \gamma_m)$ est la base principale d'inertie pour le grain A. On pose,

$$G_AP = G_AP_{(1)}e_1 + G_AP_{(2)}e_2 + G_AP_{(3)}e_3 \tag{2.8}$$

$$\begin{array}{l} \alpha_m = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 \\ \beta_m = \beta_1 e_1 + \beta_2 e_2 + \beta_3 e_3 \\ \gamma_m = \gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3 \end{array} . \tag{2.9}$$

Soit $P1$ la matrice (orthogonale) de passage de la base principale d'inertie au repère global,

$$P1 = \begin{pmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{pmatrix}, \tag{2.10}$$

et $P2$ la matrice (orthogonale) de passage de la base locale au contact (n, t, s) au repère global,

$$P2 = \begin{pmatrix} n_1 & t_1 & s_1 \\ n_2 & t_2 & s_2 \\ n_3 & t_3 & s_3 \end{pmatrix}. \tag{2.11}$$

En posant $M_A = P2^T R_A$ et $M_B = P2^T R_B$ où,

$$R_A = \begin{pmatrix} \alpha_2 G_A P_{(3)} - \alpha_3 G_A P_{(2)} & \beta_2 G_A P_{(3)} - \beta_3 G_A P_{(2)} & \gamma_2 G_A P_{(3)} - \gamma_3 G_A P_{(2)} \\ \alpha_3 G_A P_{(1)} - \alpha_1 G_A P_{(3)} & \beta_3 G_A P_{(1)} - \beta_1 G_A P_{(3)} & \gamma_3 G_A P_{(1)} - \gamma_1 G_A P_{(3)} \\ \alpha_1 G_A P_{(2)} - \alpha_2 G_A P_{(1)} & \beta_1 G_A P_{(2)} - \beta_2 G_A P_{(1)} & \gamma_1 G_A P_{(2)} - \gamma_2 G_A P_{(1)} \end{pmatrix}, \tag{2.12}$$

on obtient l'expression de la transposée de H^α pour un seul contact α dans le cas où on utiliserait toutes les inconnues cinématiques au contact,

$$v \;=\; \begin{pmatrix} P2^T & M_A & 0 & 0 \\ 0 & P2^T P1 & 0 & 0 \\ 0 & 0 & -P2^T & -M_B \\ 0 & 0 & 0 & -P2^T P1 \end{pmatrix} \begin{pmatrix} V_A \\ V_B \end{pmatrix} \tag{2.13}$$

où $V_A = \begin{pmatrix} U_G A(1) \\ U_G A(2) \\ U_G A(3) \\ \Omega_{A\alpha} \\ \Omega_{A\beta} \\ \Omega_{A\gamma} \end{pmatrix}$ et $V_B = \begin{pmatrix} U_G B(1) \\ U_G B(2) \\ U_G B(3) \\ \Omega_{B\alpha} \\ \Omega_{B\beta} \\ \Omega_{B\gamma} \end{pmatrix}$.

La matrice complète H sera l'assemblage des matrices de chaque contact.

Utilisation de la matrice H.

La vitesse relative entre le candidat et l'antagoniste pour le contact α est égale à la différence des vitesses des particules en contact, c'est à dire $v^\alpha = H^{\alpha T} V$ et de façon symétrique l'impulsion au contact r^α exercée par l'antagoniste sur le candidat contribue à la résultante R via $H^\alpha r^\alpha$. Les vitesses relatives aux contacts et les impulsions sont assemblées dans les équations de compatibilité,

$$\begin{cases} \text{en vitesse} : v = H^T V \\ \text{en efforts} : R = Hr \end{cases} . \tag{2.14}$$

2.1.3 Relations de comportement et interaction

Contact inélastique unilatéral sans frottement.

Un premier comportement est le contact unilatéral simple qui est naturellement exprimé comme une condition de complémentarité liant les efforts de contact à une distance inter grains notée ici u_n. En dynamique, Moreau a prouvé dans un lemme de viabilité [49] que l'on pouvait utiliser une loi de complémentarité en vitesse-impulsion. Dans le cas d'un contact inélastique unilatéral sans frottement, le couple (v_n,r_n) doit satisfaire la loi de comportement,

$$\begin{cases} \text{Si} \quad u_n > 0 \quad r_n = 0 \\ \text{Si} \quad u_n = 0 \quad 0 \leq \ v_n \perp r_n \ \geq 0 \end{cases} , \tag{2.15}$$

où $[0 \leq \ v_n \perp r_n \ \geq 0] \Leftrightarrow [v_{n\alpha} \geq 0 \quad r_{n\alpha} \geq 0 \quad v_{n\alpha} r_{n\alpha} = 0 \quad \forall \alpha]$.

En utilisant un schéma d'intégration en temps du type "time stepping" on obtient,

$$\begin{cases} \text{Si} \quad u_n^p > 0 \quad r_n = 0 \\ \text{Si} \quad u_n^p \leq 0 \quad 0 \leq \ v_n \perp r_n \ \geq 0 \end{cases} , \tag{2.16}$$

où u^p est la prédiction de l'espace inter grain (positive si aucun contact n'est détecté) ; on utilise ici le prédicteur 'leapfrog' classique [49] : $u^p = u^- + \frac{h}{2} v^-$, avec $h = t^+ - t^-$ comme pas de temps. Cette relation est résumée dans l'équation formelle suivante,

$$\mathcal{R}(v, r) = 0 \tag{2.17}$$

La relation de comportement dans le cas du contact sans frottement se réduit à une écriture sur la composante normale v_n de la vitesse relative v. Ce qui réduit H à la première ligne de la matrice décrite précédemment.

Contact inélastique unilatéral avec frottement de Coulomb.

Au lieu de se limiter ici aux composantes normales on a besoin des trois composantes tant en vitesse qu'en effort.

$$v = \begin{pmatrix} v_n \\ v_t \end{pmatrix}, r = \begin{pmatrix} r_n \\ r_t \end{pmatrix} \tag{2.18}$$

avec v_t et r_t les vitesses et impulsions tangentielles (vecteurs à deux composantes). On a toujours la relation précédente concernant les normales,

$$\begin{cases} \text{Si} \quad u_n^p > 0 \quad r_n = 0 \\ \text{Si} \quad u_n^p \leq 0 \quad 0 \leq \ v_n \perp r_n \ \geq 0 \end{cases} \tag{2.19}$$

où u_n^p est un prédicteur de la position cherchée. On peut alors la réécrire sous la forme algorithmique suivante,

$$\begin{cases} \text{Si} \quad u_n^p > 0 \quad r_n = 0 \\ \text{Si} \quad u_n^p \leq 0 \quad r_n - Proj_{R^+}\left(r_n - \rho_n v_n\right) = 0, \quad \rho_n > 0 \end{cases}. \tag{2.20}$$

A cela on ajoute la relation liant vitesse et impulsions tangentiel du au frottement, avec μ le coefficient de frottement,

$$\|r_t\| \leq \mu |r_n| \quad \text{et} \quad v_t \neq 0 \Rightarrow r_t = -\mu r_n \frac{v_t}{\|v_t\|}. \tag{2.21}$$

que l'on peut réécrire sous la forme d'une égalité,

$$r_t - Proj_{r_n D_\mu}\left(r_t - \rho_t v_t\right) = 0, \quad \rho_t > 0, \tag{2.22}$$

avec $D_\mu = \{s_t : ||s_t|| \leq \mu r_n\}$.

Le comportement de l'interaction de contact frottant, résumé dans la relation formelle $\mathcal{R}\left(v, r\right)$, est régi par le système d'équations et inéquations,

$$\mathcal{R}\left(v, r\right) = 0 \Leftrightarrow \begin{cases} \begin{cases} \text{Si} \quad u_n^p > 0 \quad r_n = 0 \\ \text{Si} \quad u_n^p \leq 0 \quad r_n - Proj_{R^+}\left(r_n - \rho_n v_n\right) = 0, \quad \rho_n > 0 \end{cases} \\ r_t - Proj_{r_n D_\mu}\left(r_t - \rho_t v_t\right) = 0, \quad \rho_t > 0 \end{cases}. \tag{2.23}$$

Il faut noter que puisque cette relation de comportement porte sur les composantes normales et tangentielles seules de v et r, on ne prend pas en compte les différences de rotation et les moments associés, ce qui pourrait être le cas avec un modèle de Coulomb généralisé dit aussi frottement de roulement. La matrice H sera réduite à ses trois premières lignes et les vecteurs v et r ne comprendront dans la suite que trois composantes v_n, v_t et r_n, r_t.

2.1.4 Problème de référence

En prenant en compte la dynamique des grains (2.6) et les équations de compatibilité (2.14), on peut obtenir une formulation en impulsion du problème de référence (que l'on appellera dynamique réduite),

$$v - Wr = v^d, \tag{2.24}$$

où W est l'opérateur de Delassus $W = H^T M^{-1} H$, et les quantités données sont $v^d = H^T M^{-1} R^d$.

On notera ici que cette équation est écrite classiquement en milieu granulaire avec l'introduction de ce que l'on appelle vitesse libre v^d, qui correspond à la vitesse de la particule si on enlève les contacts. Si l'on reprend l'expression de R^d de (2.5) on voit intervenir dans cette vitesse libre à la fois les efforts imposés mais aussi les vitesses initiales V^- qui traduisent la dynamique d'un grain sans contact.

Quand des vitesses sont imposées sur une partie Ω_1 des grains : $V|_{\Omega_1} = V_d$, les degrés de liberté des grains peuvent être séparés en deux ensembles : avec l'indice u les valeurs inconnues, et avec l'indice p les valeurs données. La dynamique réduite (2.24) s'écrit alors,

$$v - W_u r = v^d, \tag{2.25}$$

avec $H = \begin{pmatrix} H_u \\ H_p \end{pmatrix}$, $W_u = H_u^T M_u^{-1} H_u$, $v^d = H_u^T M_u^{-1} R_u^d + H_p^T V_p$.

Puisque ce problème a la même structure que (2.24), et pour simplifier les notations, (2.24) sera utilisé dans la suite.

Pour fermer le problème il reste à ajouter l'équation de comportement (2.17), ce qui donne le problème de référence suivant,

$$\begin{cases} Wr - v = -v^d \\ \mathcal{R}(v, r) = 0 \end{cases} . \tag{2.26}$$

2.1.5 Problème sous structuré

Introduire le découpage proposé au chapitre 1.4 induit la création de nouveaux champs d'inconnues :

- sur chaque interface entre un sous domaine Ω_E et un sous domaine $\Omega_{E'}$, une répartition discrète d'impulsions $F_{EE'}$ exprimant les actions du sous domaine $\Omega_{E'}$ sur le sous domaine Ω_E (impulsion de cohésion des grains coupés),
- un champ de vitesse discret $V_{EE'}$ qui est la réduction du champ de vitesse du sous domaine Ω_E sur son interface locale avec le sous domaine voisin $\Omega_{E'}$.

Une sous structure E, est soumise aux actions des interfaces voisines $\Gamma_{EE'}$ (aux impulsions $F_{EE'}$ et aux vitesses $V_{EE'}$). En considérant l'ensemble des interfaces locales de la sous structure E les assemblages des champs précédents sont notés $F_{E\Gamma}$ et $V_{E\Gamma}$. Une interface $\Gamma_{EE'}$ transfère les impulsions moyennes $F_{EE'}$ et les vitesses $V_{EE'}$ de part et d'autre de ses côtés.

Avec le choix précédent de découpage, le comportement de l'interface est parfait, c'est à dire qu'on assure la continuité des vitesses des centres de masses et l'équilibre des efforts entre demi-grains,

$$\begin{cases} F_{EE'} + F_{E'E} = 0 \\ V_{EE'} = V_{E'E} \end{cases} \tag{2.27}$$

La structure de données du problème de référence est donc complexifiée par l'ajout des quantités d'interface. Par exemple, pour un sous domaine E parmi n_{SD} sous domaines, l'indice $E\Gamma$ exprime l'ensemble de ses interfaces. On met en place pour la suite une matrice booléenne $C_{E\Gamma}$ qui fait le passage des degrés de liberté d'interface locale aux degrés de liberté du sous domaine. Elle est définie de manière à projeter l'ensemble des vitesses du sous domaine V_E sur $V_{E\Gamma}$ qui assemble les vitesses des interfaces du sous domaine,

$$V_{E\Gamma} = C_{E\Gamma} V_E. \tag{2.28}$$

L'équation duale, permettant de distribuer sur l'enssemble des grains du sous domaine les impultions d'interface, s'écrit à partir de cette même matrice $C_{E\Gamma}$,

$$F_E = C_{E\Gamma}^T F_{E\Gamma}. \tag{2.29}$$

Les quantités locales au sein des sous domaines sont indicées par E pour montrer qu'elles ne sont plus prises en compte sur l'ensemble du domaine mais bien sur sa limitation à E. Ainsi pour chaque sous domaine E on a les quantités locales (r_E, v_E).

La solution $s = \bigcup_E s_E$ avec $s_E = (v_E, r_E, V_E, R_E)$ du problème de référence doit vérifier sur tous les sous domaines Ω_E,

– La compatibilité des vitesses,

$$V_E|_{\Omega_1 \cap \Omega_E} = V_d|_{\Omega_1 \cap \Omega_E} \qquad v_E = H_E^T V_E \tag{2.30}$$

– L'admissibilité des efforts,

$$R_E = H_E r_E \tag{2.31}$$

– La dynamique non régulière, avec l'ajout des quantités d'interface s'écrit $M_E V_E = R_E - F_E + R_E^d$ et la réduction sur les inconnues de contact donne,

$$v_E - W_E r_E = v_E^d - H_E^T M_E^{-1} C_{E\Gamma}^T F_{E\Gamma} \tag{2.32}$$

où $W_E = H_E^T M_E^{-1} H_E$ et $v_E^d = H_E^T M_E^{-1} R_E^d$.

– La relation de comportement,

$$\mathcal{R}(v_E, r_E) = 0 \tag{2.33}$$

et sur toutes les interfaces $\Gamma_{EE'}$,
– Le comportement d'interface,

$$\left\{ \begin{array}{c} F_{EE'} + F_{E'E} = 0 \\ V_{EE'} = V_{E'E} \end{array} \right. . \tag{2.34}$$

2.2 Un algorithme LATIN multidomaine

Cette méthode de décomposition de domaine associée à la méthode LArge Time INcrement method (LATIN) créée au départ pour les milieux continus [38] fait intervenir une formulation mixte en déplacement et tension pour les systèmes de tenségrité en quasi-statique [50], ou vitesse et impulsion ici.

A cause de la non régularité dynamique des problèmes granulaires, on n'utilisera et on ne présentera ici qu'une version incrémentale de la méthode LATIN.

2.2.1 Méthode LATIN monoéchelle

Dans le cas présent d'une approche incrémentale, la méthode LATIN [38] est une méthode itérative qui à chaque pas nécessite deux étapes différentes, une étape locale, où les relations locales de comportement sont traitées, et une étape linéaire, où les conditions linéaires, d'admissibilité notamment, sont vérifiées. Ce schéma itératif conduit à des approximations successives de la solution, c'est à dire les vitesses relatives de contact et impulsions (v_E, r_E), $E = 1, ..., n_{SD}$, et les vitesses et impulsions sur chaque interface $(V_{EE'}, F_{EE'})$.

Étape locale.

Une fois (v_E, r_E) et $(V_{EE'}, F_{EE'})$ connus, l'étape locale consiste à trouver $(\widehat{v}_E, \widehat{r}_E)$ et $(\widehat{V}_{EE'}, \widehat{F}_{EE'})$ satisfaisant :

– pour chaque interface, le comportement d'interface parfaite :

$$\begin{cases} \widehat{F}_{EE'} + \widehat{F}_{E'E} = 0 \\ \widehat{V}_{EE'} = \widehat{V}_{E'E} \end{cases} \tag{2.35}$$

– et la direction de recherche sur l'interface :

$$(\widehat{F}_{EE'} - F_{EE'}) + d^+(\widehat{V}_{EE'} - V_{EE'}) = 0. \tag{2.36}$$

– pour chaque interaction de chaque sous domaine, la relation de comportement :

$$\mathcal{R}(\widehat{v}_E, \widehat{r}_E) = 0 \tag{2.37}$$

– et la direction de recherche pour les interactions :

$$(\widehat{r}_E - r_E) - l^+(\widehat{v}_E - v_E) = 0. \tag{2.38}$$

d^+ et l^+ sont deux paramètres positifs de la méthode. Pour des interfaces parfaites et par exemple pour des interactions de contact sans frottement, la solution de l'étape locale est explicite :

$$\left\{ \begin{array}{l} \widehat{F}_{E'E} = -\widehat{F}_{EE'} = \frac{1}{2}[(F_{E'E} - F_{EE'}) + d^+(V_{E'E} - V_{EE'})] \\ \widehat{V}_{EE'} = \widehat{V}_{E'E} = \frac{1}{2}[(V_{E'E} + V_{EE'}) + d^{+-1}(F_{E'E} + F_{EE'})] \end{array} \right. \tag{2.39}$$

$$\left\{ \begin{array}{ll} \text{si } u^p > 0, & \widehat{r}_E = 0 \quad \text{et} \quad \widehat{v}_E = v_E - (l^+)^{-1} r_E \\ \text{si } u^p \leq 0, & \widehat{r}_E =< r_E - l^+ v_E >_+ \quad \text{et} \\ & \widehat{v}_E = -(l^+)^{-1} < r_E - l^+ v_E >_- \end{array} \right. . \tag{2.40}$$

Étape linéaire.

Une fois $(\widehat{v}_E, \widehat{r}_E)$ et $(\widehat{V}_{EE'}, \widehat{F}_{EE'})$ connus, l'étape linéaire consiste à trouver (v_E, r_E) et $(V_{EE'}, F_{EE'})$ satisfaisant,

- pour chaque sous domaine, son évolution dynamique,

$$M_E V_E = R_E - F_E + R_E^d, \tag{2.41}$$

  avec les équations d'admissibilité : $v_E = H_E^T V_E$, $R_E = H_E r_E$, $F_E = C_{E\Gamma}^T F_{E\Gamma}$, $V_{E\Gamma} = C_{E\Gamma} V_E$. Par rapport à l'équation du cas non sous structuré (2.6) on ajoute ici F_E, les impulsions exercées par le sous domaine sur l'interface.
- pour chaque sous domaine et ses interfaces, les directions de recherche :

$$\left\{ \begin{array}{l} (r_E - \widehat{r}_E) + l^-(v_E - \widehat{v}_E) = 0 \\ (F_{E\Gamma} - \widehat{F}_{E\Gamma}) - d^-(V_{E\Gamma} - \widehat{V}_{E\Gamma}) = 0 \end{array} \right. . \tag{2.42}$$

d^- et l^- sont aussi deux paramètres positifs de la méthode. En utilisant les directions de recherche, le problème linéaire global sur chaque sous domaine est

$$(M_E + H_E l^- H_E^T + C_{E\Gamma}^T d^- C_{E\Gamma}) V_E = \widehat{R}_E + R_E^d \tag{2.43}$$

où $\widehat{R}_E = H_E(\widehat{r}_E + l^- \widehat{v}_E) - C_{E\Gamma}^T(\widehat{F}_{E\Gamma} - d^- \widehat{V}_{E\Gamma})$ est une quantité donnée à cette étape. Une fois V_E obtenu, les autres quantités sont obtenues par post-traitement,

$$\begin{array}{r} v_E = H_E^T V_E \\ r_E = \widehat{r}_E + l^- \widehat{v}_E - l^- H_E^T V_E \\ F_{E\Gamma} = \widehat{F}_{E\Gamma} - d^- \widehat{V}_{E\Gamma} + d^- V_{E\Gamma} \end{array} . \tag{2.44}$$

Une façon classique de représenter la méthode LATIN est donnée sur la Figure 2.5. A gauche se trouve la représentation pour une résolution de problèmes

non linéaires et à droite une représentation possible dans un cas de problèmes non réguliers. Ces non régularités sont représentées par la droite indicée NR mais ici la représentation est faite de manière simplifiée, avec une seule cassure, alors que le nombre de non régularités est beaucoup plus important et correspond au nombre de contacts. En notant s un élément solution d'un groupe d'équation, on a défini deux ensembles : Ad l'ensemble des éléments s_n vérifiant les équations linéaires et NL (respectivement NR) l'ensemble des éléments $\hat{s}_n$ vérifiant les équations locales non linéaires (respectivement non régulières). La solution du problème est alors l'intersection des ensembles A_d et NL (respectivement NR).

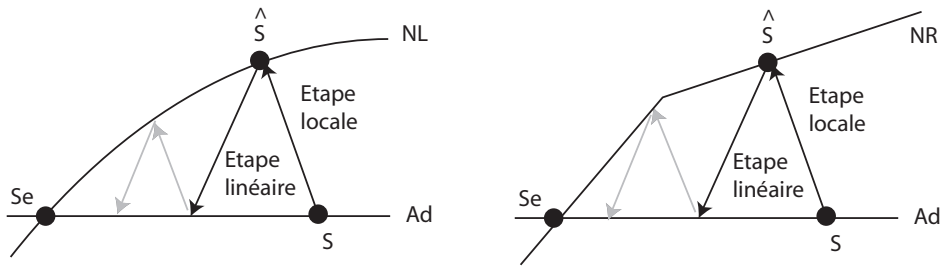


FIG. 2.5 – Principe de la méthode LATIN : illustration pour les cas régulier et non régulier.

Remarque concernant le contact frottant.

Dans le cas du contact frottant explicité à la section 2.1.3, à l'étape locale la direction de recherche s'écrit,

$$(\widehat{r} - r) - l^{+}(\widehat{v} - v) = 0. \tag{2.45}$$

Au final, on a le problème,

$$\begin{cases} (\widehat{r} - r) - l^{+}(\widehat{v} - v) = 0 \\ \mathcal{R}(\widehat{v}, \widehat{r}) = 0 \end{cases} \tag{2.46}$$

qui est semi-couplé entre les composantes normales et tangentielles. Dans le cas où la direction de recherche l^{+} est diagonale on a une expression explicite des diverses composantes.

$$\begin{array}{ll} \text{Si} \quad u_n^p > 0 & \left\{ \begin{array}{l} \widehat{r}_n = 0 \\ \widehat{v}_n = v_n - l_n^{-1} r_n \\ \widehat{v}_t = v_t - l_t^{-1} r_t \\ \widehat{r}_t = 0 \end{array} \right. \\ \text{Si} \quad u_n^p \leq 0 & \left\{ \begin{array}{l} \widehat{v}_n = < v_n - l_n^{-1} r_n >_+ \\ \widehat{r}_n = -l_n^{-1} < v_n - l_n^{-1} r_n >_- \\ \widehat{r}_t = Proj_{r_n D_\mu} (r_t - l_t v_t) \\ \widehat{v}_t = \widehat{r}_t - (r_t - l_t v_t) \end{array} \right. \end{array} \tag{2.47}$$

2.2.2 Recherche des paramètres optimaux

Dans le cas de la méthode LATIN classique, les directions de recherche, d^+ et l^+ et leurs homologues d^- et l^- sont généralement choisies identiques, $d^+ = d^-$ et $l^+ = l^-$, notamment pour justifier la convergence de la méthode. Les valeurs choisies pour ces directions de recherche doivent être déterminées. Dans [50] cette valeur pour la tenségrité en quasi-statique est liée à la matrice de rigidité qui joue le rôle de notre matrice de masse. Ici on teste la vitesse de convergence autour d'une valeur liée à la matrice de masse. Ainsi, les directions de recherche $l^+ = l^-$ sont prises égales à la moyenne de la matrice de masse et $d^+ = d^-$ sont prises égales à la moyenne des masses des seuls grains d'interface.

L'essai utilisé pour réaliser cette étude est un essai bidimensionnel de cisaillement avec un millier de grains, ces grains sont confinés dans une boite dont les bords latéraux sont penchés de manière à venir cisailler l'échantillon à l'intérieur, ce même essai est utilisé et décrit au chapitre 2.4.2. On lance alors une série de calculs non linéaires dans lesquels on ne fait varier que ces deux paramètres. Ceux-ci n'influent pas sur la solution à convergence mais sur le nombre d'itérations nécessaires pour arriver à un même niveau d'erreur par rapport à une référence (construite ici en laissant converger le calcul avec un très grand nombre d'itérations).

Sur la Figure 2.6 chaque point de la surface représente pour un même nombre d'itérations l'erreur par rapport à la solution de référence en fonction des valeurs de d et de l. Le point repéré de façon particulière correspond au choix précédent, avec $x = l$, $y = d$ et z l'erreur. Cette surface peut être décomposée en trois parties. Pour les décrire, on a une surface presque horizontale que l'on nommera "plancher" et deux surfaces verticales que l'on pourra nommer "murs". En fait le "plancher" n'est pas tout à fait plan, il est penché vers le coin où se trouve le point particularisé. Cette pente se présente suivant les deux paramètres l et d. La Figure 2.6 est un zoom sur des valeurs de paramètres proches du point particularisé, mais cette surface "plancher" conserve la même tendance pour des valeurs plus éloignées de l et d. Les

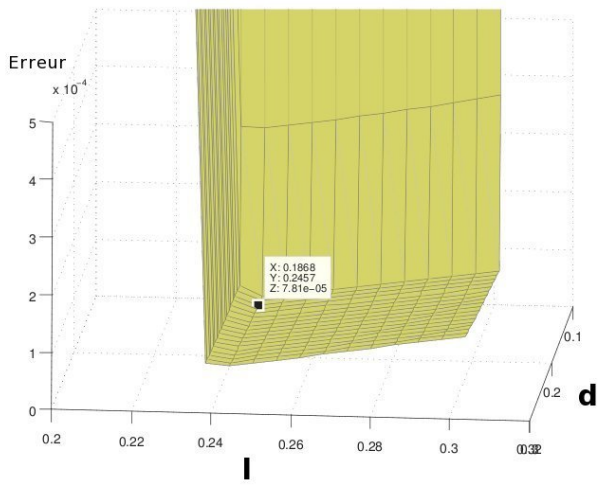


FIG. 2.6 – Erreur par rapport à la référence en fonction des valeurs choisies pour les paramètres de directions de recherche.

deux “murs” illustrent le fait que, proche des valeurs optimales par valeur inférieure, l’algorithme, appliqué à un milieu granulaire, diverge.

Le point particularisé semble bien être l’optimum, ce qui sera confirmé plus loin en comparant l’algorithme LATIN à une méthode de type Gauss Seidel non linéaire.

2.3 Un algorithme GSNL multidomaine

Avant de chercher à formuler un solveur de type Gauss-Seidel dans un contexte de décomposition de domaine, on s'attache dans une première partie à comparer la méthode LATIN à deux étapes avec l'algorithme de Gauss-Seidel ne comportant qu'une étape à chaque itération, cette étape consistant à résoudre successivement des problèmes non réguliers locaux (par contact), les contacts précédents ayant été mis à jour. Cette dernière actualisation est une caractéristique de Gauss-Seidel qui le distingue d'un algorithme de type Jacobi, où tous les contacts sont traités en parallèle. On trouvera dans [56] des expériences numériques pour une approche parallèle restreinte à une structure à mémoire partagée.

2.3.1 GSNL versus LATIN (monoéchelle et monodomaine)

Dans un premier temps on se place dans un cas monoéchelle, avec une seule structure, pour partir de l'écriture du solveur LATIN et montrer que par un choix de directions de recherche appropriées on retrouve une formulation de type GSNL.

À chaque itération on effectue deux étapes, une première étape dite linéaire qui va permettre de passer des variables $(\widehat{r}, \widehat{v})$ aux variables intermédiaires (r, v), puis une deuxième étape dite locale qui va permettre de passer de ces variables intermédiaires aux variables $(\widetilde{r}, \widetilde{v})$.

$$(\widehat{r}, \widehat{v}) \overset{\text{Étape 1}}{\longrightarrow} (r, v) \overset{\text{Étape 2}}{\longrightarrow} (\widetilde{r}, \widetilde{v}) \tag{2.48}$$

– Étape linéaire
  Après écriture de la dynamique réduite, cf. chapitre 2.1.4, les équations de cette étape se résument de la façon suivante,

$$\left\{ \begin{array}{rcll} v - WHr & = & v^d & \text{dynamique} \\ r + l^- v & = & \widehat{r} + l^- \widehat{v} & \text{direction de recherche} \end{array} \right. . \tag{2.49}$$

  En choisissant $l^- = 0$ on obtient,

$$\left\{ \begin{array}{c} v = v^d + W\widehat{r} \\ r = \widehat{r} \end{array} \right. \tag{2.50}$$

– Étape locale
  Les équations de cette étape se résument de la façon suivante,

$$\left\{ \begin{array}{c} \mathcal{R}\left(\widetilde{r}, \widetilde{v}\right) = 0 \quad \text{où } \mathcal{R} \text{ est la relation de comportement au contact} \\ \widetilde{r} - l^+ \widetilde{v} = r - l^+ v \quad \text{Direction de recherche} \end{array} \right. . \tag{2.51}$$

En notant $l^+ = W_D^{-1}$ on obtient,

$$\left\{ \begin{array}{c} W_D\widetilde{r} - \widetilde{v} = W_D r - v \\ \mathcal{R}\left(\widetilde{r}, \widetilde{v}\right) = 0 \end{array} \right. , \tag{2.52}$$

et avec les équations obtenues à l'étape linéaire,

$$\left\{ \begin{array}{c} W_D\widetilde{r} - \widetilde{v} = W_D\widehat{r} - \left(v^d + W\widehat{r}\right) \\ \mathcal{R}\left(\widetilde{r}, \widetilde{v}\right) = 0 \end{array} \right. . \tag{2.53}$$

On obtient alors une écriture GSNL,

$$\left\{ \begin{array}{c} W_D\widetilde{r} - \widetilde{v} = -v^d - \left(W - W_D\right)\widehat{r} \\ \mathcal{R}\left(\widetilde{r}, \widetilde{v}\right) = 0 \end{array} \right. . \tag{2.54}$$

Sous cette forme, le système non linéaire ci-dessus peut être résolu par un algorithme de Jacobi non linéaire (par bloc) si W_D est pris égal à la diagonale (par bloc de contact) de W, ou par un algorithme de Gauss-Seidel non linéaire (par bloc) si W_D est pris égal à la triangulaire inférieure (par bloc de contact), notée aussi W^L, de W. Dans la méthode LATIN il est cependant d'usage de choisir, afin de préserver la localité, une direction de recherche diagonale (par bloc éventuellement). Ceci rapproche donc pour les choix particuliers effectués ici la méthode LATIN incrémentale d'une méthode de Jacobi non linéaire par bloc.

2.3.2 Stratégies de type GSNL en décomposition de domaine

On souhaite remplacer le solveur de type LATIN tel que présenté dans les sections précédentes par un solveur de type GSNL dans un contexte de décomposition de domaine multiéchelle. La raison tient d'une part à la robustesse constatée de GSNL sur les problèmes granulaires, et d'autre part au constat effectué précédemment qu'une méthode de type LATIN incrémentale s'apparente plutôt à un algorithme de Jacobi a priori moins performant. Nous reprenons donc ici la formulation d'un problème non régulier à non régularité diffuse en utilisant des notations des méthodes de décomposition de domaine. Au delà de la dynamique granulaire (grains rigides) d'autres applications sont aussi visées : la dynamique de multicorps déformables, la dynamique de systèmes de tenségrité.

On adopte ici une formulation de type FETI, où un multiplicateur de Lagrange est introduit pour imposer le raccord cinématique aux interfaces. La condition de continuité du champ de vitesses sur l'interface Γ nécessite un opérateur booléen signé $A_{E\Gamma}$ qui extrait du champ de vitesses d'un sous domaine V_E sa restriction signée à l'interface. Cette condition de continuité s'écrit alors,

$$\sum_{E=1}^{n_{sd}} A_{E\Gamma} V_E = 0. \tag{2.55}$$

On définit F_Γ comme le multiplicateur de Lagrange associé à cette contrainte. (Par rapport à l'équation (2.32), F_Γ est l'assemblage par une matrice booléenne signée des quantités $F_{E\Gamma}$ de chaque sous domaine en un unique vecteur pour toutes les interfaces.) La dynamique réduite (2.24) sur chaque sous domaine s'écrit alors,

$$W_E r_E - v_E - H_E^T M_E^{-1} A_{E\Gamma}^T F_\Gamma = -v_E^d. \tag{2.56}$$

La combinaison de la dynamique (2.56), de la condition de continuité (2.55) et de la relation de comportement (2.17) donne la formulation,

$$\left\{ \begin{array}{c} \left.\begin{array}{c} W_E r_E - v_E - H_E^T M_E^{-1} A_{E\Gamma}^T F_\Gamma = -v_E^d \\ \mathcal{R}(v_E, r_E) = 0 \end{array}\right\} E = 1, ..., n_{SD} \\ \displaystyle\sum_{E=1}^{n_{sd}} A_{E\Gamma} V_E = 0 \end{array} \right. . \tag{2.57}$$

Les deux premières équations sont locales par sous domaine, la troisième est globale aux interfaces. Seule la seconde est non linéaire ; on peut donc aussi condenser la première équation (linéaire) sur les inconnues F_Γ et remplacer la troisième équation de raccord en privilégiant F_Γ comme inconnue principale de cette équation. On obtient alors,

$$\left\{ \begin{array}{c} \left.\begin{array}{c} W_E r_E - v_E - H_E^T M_E^{-1} A_{E\Gamma}^T F_\Gamma = -v_E^d \\ \mathcal{R}(v_E, r_E) = 0 \end{array}\right\} E = 1, ..., n_{SD} \\ \displaystyle X F_\Gamma - \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} R_E = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} R_E^d \end{array} \right. \tag{2.58}$$

où $X = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} A_{E\Gamma}^T$ et $R_E = H_E r_E$.

Le problème réduit sur (r_E,v_E,F_Γ), avec les notations $\hat{f} = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} R_E^d$ et $\hat{A}_{E\Gamma}^T = H_E^T M_E^{-1} A_{E\Gamma}^T$, s'écrit,

$$\left\{ \begin{array}{c} \left.\begin{array}{c} W_E r_E - v_E - \hat{A}_{E\Gamma}^T F_\Gamma = -v_E^d \\ \mathcal{R}(v_E, r_E) = 0 \end{array}\right\} E = 1, \ldots, n_{sd} \\ \displaystyle X F_\Gamma - \sum_{E=1}^{n_{sd}} \hat{A}_{E\Gamma} r_E = \hat{f} \end{array} \right. \tag{2.59}$$

Un algorithme de type Gauss Seidel appliqué à la résolution de ce système sera constitué de deux pas. Le premier contient la résolution d'une seule itération de Gauss-Seidel pour chaque sous domaine E,

$$\begin{cases} W_E^L r_E - v_E = \hat{A}_{E\Gamma}^T \tilde{F}_\Gamma - v_E^d - (W_E - W_E^L)\tilde{r}_E \\ \mathcal{R}(v_E, r_E) = 0 \end{cases} \tag{2.60}$$

où $\tilde{F}_\Gamma$ et $\tilde{r}_E$ sont fixés à cette étape.

Le découpage classique de Gauss-Seidel de W_E utilise la partie triangulaire inférieure W_E^L (incluant la partie diagonale W_E^D) : $W_E = W_E^L + (W_E - W_E^L)$. Ce premier pas est réalisé sous domaine par sous domaine et pourra être facilement parallélisé. Le problème (2.60) est résolu, une paire d'inconnues (v_E^α,r_E^α) après l'autre, pour lesquelles une technique d'intersection de graphes donne comme pour la méthode LATIN un résultat explicite (au moins dans un cas sans frottement).

Après mise à jour des quantités d'impulsion grain par grain R_E, le second pas consiste à résoudre la dernière équation du système (2.58),

$$XF_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1}(R_E + R_E^d). \tag{2.61}$$

Une particularité de l'opérateur X est qu'il est ici diagonal. En effet il est créé en sommant les projections des matrices de masse de chaque sous domaine sur l'interface globale. La matrice de masse, si elle est exprimée dans la base principale d'inertie, est diagonale et il en est donc de même pour son inverse. Elle ne permet pas de coupler les mouvements d'un grain à celui de ses voisins. En faisant la réduction sur l'interface, on ne fait que sélectionner les inverses des masses des grains d'interface qui ne sont donc pas reliés aux autres grains du sous domaine. La sommation ne fait alors que l'addition des inverses de la masse d'un grain d'un sous domaine et de celle de son homologue dans le sous domaine voisin. L'opérateur X est donc diagonal avec sur sa diagonale l'inverse de la masse des grains d'interface.

La formulation algorithmique associée est décrite dans l'algorithme 1, où N est le nombre de pas de temps et n_{GS} le nombre d'itérations de Gauss-Seidel réalisées afin de résoudre sur chaque sous domaine la dynamique non régulière.

Algorithm 1 (Version générique)

for $i = 1, \ldots, N$ **do** {Boucle sur les pas de temps}
  initialisation des inconnues (r_E, v_E, F_Γ) au temps t_i.
  while $\zeta < \zeta_{\max}$ (critère de convergence) **do**
    (r_E, v_E, F_Γ) sont connus, calcul de $(\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma)$:
    Étape 1 : Calcul de $\bar{r}_E, \bar{v}_E$ en parallèle pour $E = 1, \ldots, n_{sd}$
    avec n_{GS} itérations de Gauss-Seidel sur :

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E = -v_E^d + \hat{A}_{E\Gamma}^T F_\Gamma \\ \mathcal{R}(\bar{v}_E, \bar{r}_E) = 0 \end{cases}$$

    Étape 2 : Calcul de $\bar{F}_\Gamma$:

$$X \bar{F}_\Gamma = \hat{f} + \sum_{E=1}^{n_{sd}} \hat{A}_{E\Gamma} \bar{r}_E$$

    for $E = 1, \ldots, n_{sd}$ **do**
      $(r_E, v_E, F_\Gamma) \leftarrow (\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma)$
    end for
  end while
end for

2.3.3 Analogies algorithmiques des deux méthodes

Afin de comparer les deux méthodes multidomaines précédentes, il faut dégénérer chacune d'entre elles.

Premièrement, on considère une version Jacobi de l'approche Gauss-Seidel précédente, ce qui revient à prendre la partie diagonale W_E^D dans le découpage : $W_E = W_E^D + (W_E - W_E^D)$. Avec un tel choix, en passant de Gauss-Seidel à Jacobi, le taux de convergence de la méthode NLGS diminue.

Deuxièmement, un choix de direction de recherche dégénéré de la LATIN doit être fait de manière à identifier les deux algorithmes résultants. Pour identifier la méthode LATIN à l'algorithme de Jacobi précédent il faut choisir $d^+ = \frac{1}{2}X^{D^{-1}}$ et $l^+ = (W_E^D)^{-1}$ à l'étape locale et $d^- = l^- = 0$ (soit des directions de recherche "verticales" à l'étape linéaire. Avec ces choix, le taux de convergence de la méthode LATIN diminue aussi.

L'unification des deux algorithmes précédents est donc sujette à la dégénérescence de chacun d'eux.

2.4 Comparaison de performances sur un incrément de temps

2.4.1 Résultats dans un cas linéaire abstrait

Pour tester l'extensibilité des méthodes on traite ici un exemple de milieu modèle linéaire sans réalité physique, mais considéré comme un problème intermédiaire avant de s'attaquer au problème granulaire. Un échantillon granulaire est créé en utilisant la plate-forme LMGC90[1] dédiée aux problèmes multicontacts [34]. Les interactions de contact, écrites à l'origine avec une complémentarité vitesse-impulsion, sont remplacées par des liens entre les centres des grains. De cette manière, l'exemple peut être considéré comme un treillis composé de nœuds massifs et de liens linéaires. Le seul changement dans le problème de référence est la relation de comportement,

$$\mathcal{R}(v, r) = 0 \Leftrightarrow r = -kv \tag{2.62}$$

(k est le coefficient de linéarité homogène à une masse).

Dans un tel cas, le problème de référence peut être ramené en une seule équation,

$$(W + k^{-1})r = -v_d. \tag{2.63}$$

[1]http ://www.lmgc.univ-montp2.fr/~dubois/LMGC90/

Le membre gauche de l'équation est composée d'une partie diagonale régulière k^{-1} et de l'opérateur de Delassus W, qui est a priori singulier.

L'exemple testé est constitué de 991 grains, Figure 2.7, la vitesse des grains inférieurs est imposée nulle, tandis que les bords droit et gauche sont soumis à une rotation globale de manière à simuler un essai de cisaillement.

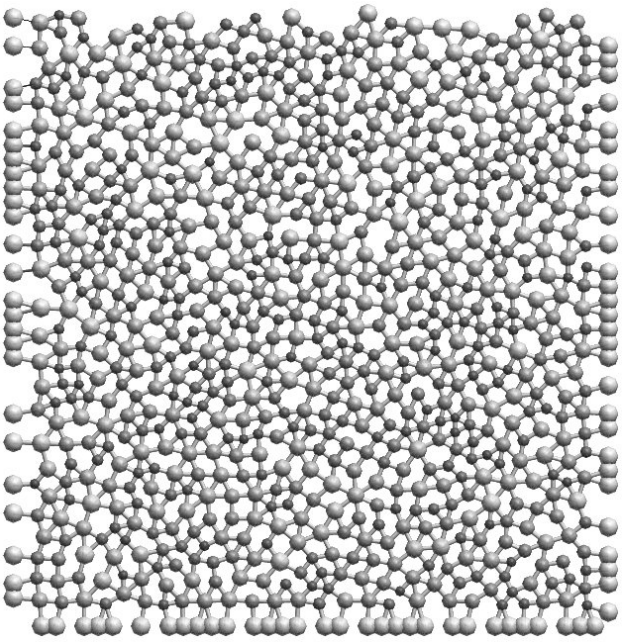


FIG. 2.7 – Réseau de 991 noeuds et 1700 interactions.

Deux valeurs de k sont testées :

- une valeur de référence k_0 choisie égale à la moyenne des termes (diagonaux) de $(W^D)^{-1}$
- une valeur k_1 réduisant la contribution régularisante : $k_1 = 50k_0$.

Plusieurs courbes de convergence sont tracées dans chacun des cas, Figures 2.8 et 2.9, pour chacune des méthodes, et des nombres différents de sous domaines, $n_{SD} = 1, 2, 4$ ou 8. Pour ce problème linéaire test, et contrairement au cas granulaire, la solution est unique en terme d'impulsions, r_{ref}. En conséquence, une erreur relative est utilisée pour tester la convergence,

$$\eta = \sqrt{\frac{(r - r_{\text{ref}})^T (r - r_{\text{ref}})}{(r + r_{\text{ref}})^T (r + r_{\text{ref}})}} \tag{2.64}$$

où r_{ref} est le vecteur contenant l'ensemble des impulsions des liens. On choisit de comparer la solution obtenue par rapport à la solution de référence en se basant sur les impulsions r_{ref} plutôt que sur les vitesses v_{ref}. Ce choix est en accord avec l'approche NSCD où les impulsions sont les variables privilégiées.

De plus lors des simulations on a remarqué une très grande sensibilité des données en impulsions au nombre de chiffres significatifs (proches de celui du zéro machine).

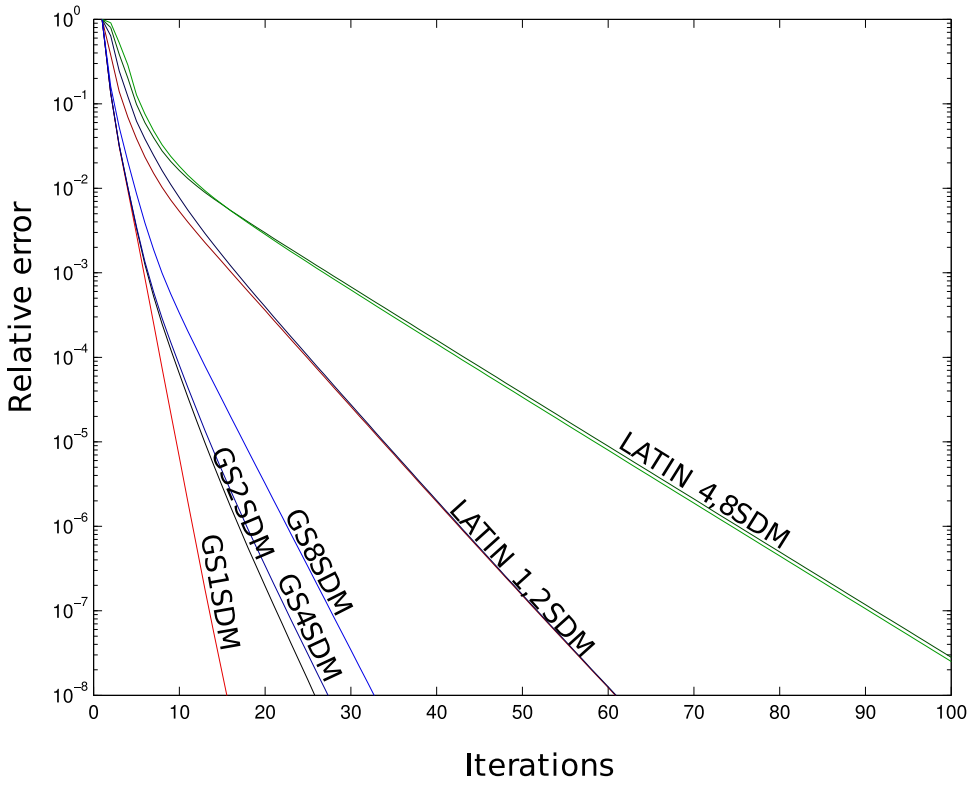


FIG. 2.8 – Évolution de l'erreur relative en fonction des itérations pour $k = k_0$.

Pour la méthode LATIN, les directions de recherche sont choisies telles que $l^+ = l^- = k$ et $d^+ = d^- = k$.

Une première remarque est que l'on vérifie bien que le taux de convergence de chacune des méthodes diminue quand l'influence de la partie régulière diminue. Sur ce test, NLGS montre un taux de convergence plus élevé que la LATIN, l'écart se réduisant pour le cas moins régulier ($k = k_1$, Figure 2.9).

Afin de pouvoir tirer des conclusions avec plus de sous domaines la taille de l'exemple a été augmentée de manière à ce que les sous domaines ne soient pas de

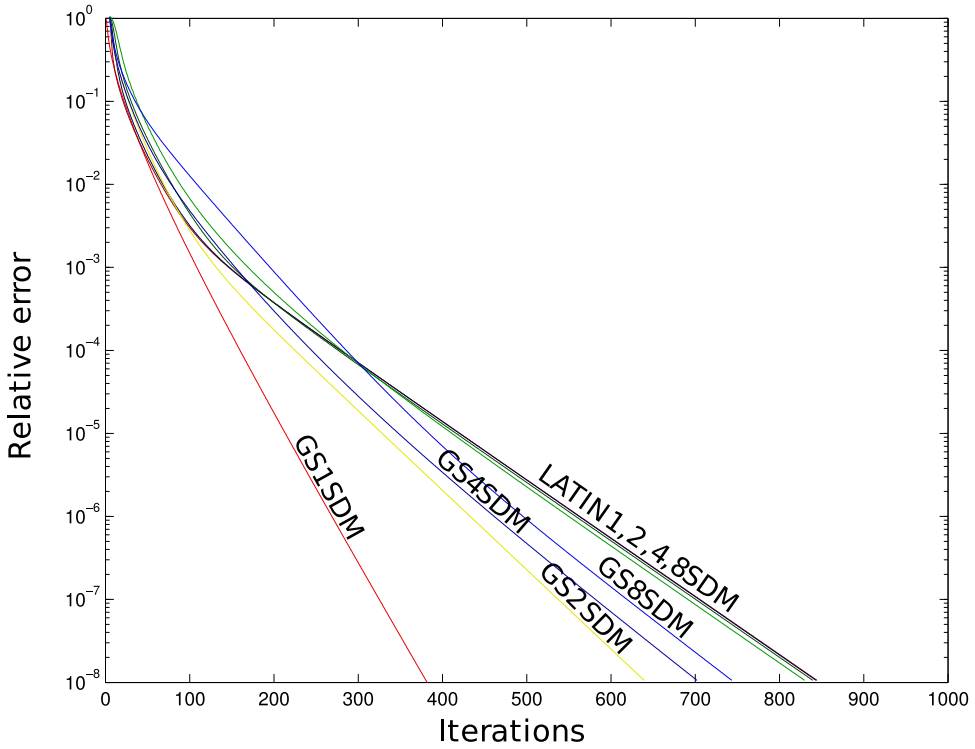


FIG. 2.9 – Évolution de l'erreur relative en fonction des itérations pour $k = k_1 = 50k_0$.

trop petite taille. On a déposé dans une boîte faisant la même hauteur mais 5 fois la largeur de la précédente 5 000 grains. Cet exemple est représenté Figure 2.10 avec par dessus en trait pointillé la grille de décomposition en 16 sous domaines. C'est cette grille qui va permettre de choisir à quels sous domaines appartiennent les contacts.

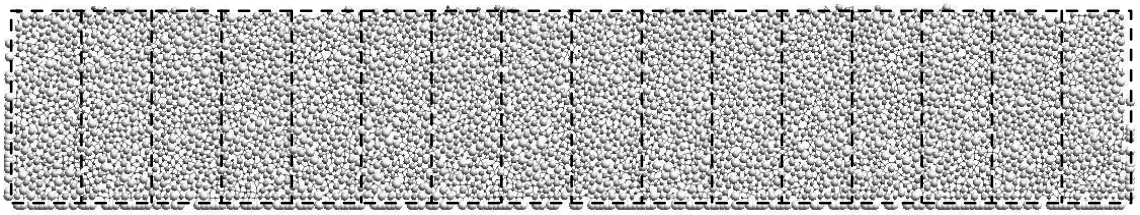


FIG. 2.10 – Réseau de 5000 noeuds et 10000 interactions avec grille de décomposition en 16 sous domaines.

Les résultats illustrés dans les Figures (2.11 et 2.12) sont alors très similaires aux premiers.

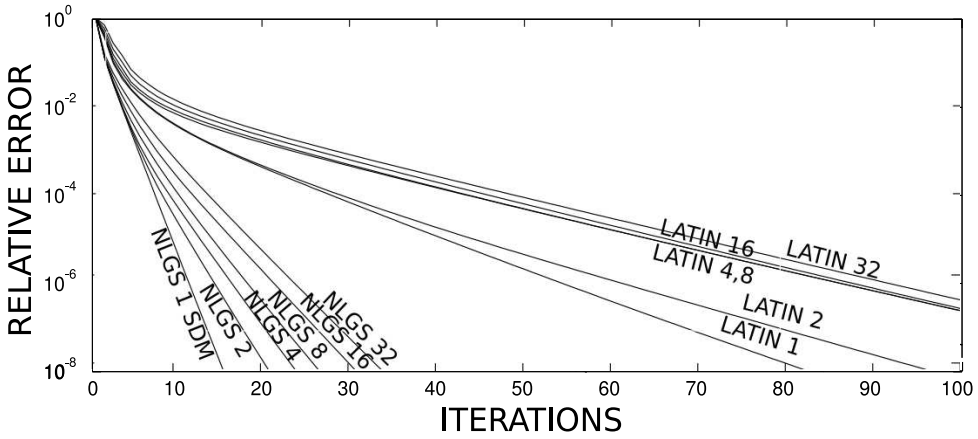


FIG. 2.11 – Évolution de l'erreur relative en fonction des itérations pour $k = k_0$.

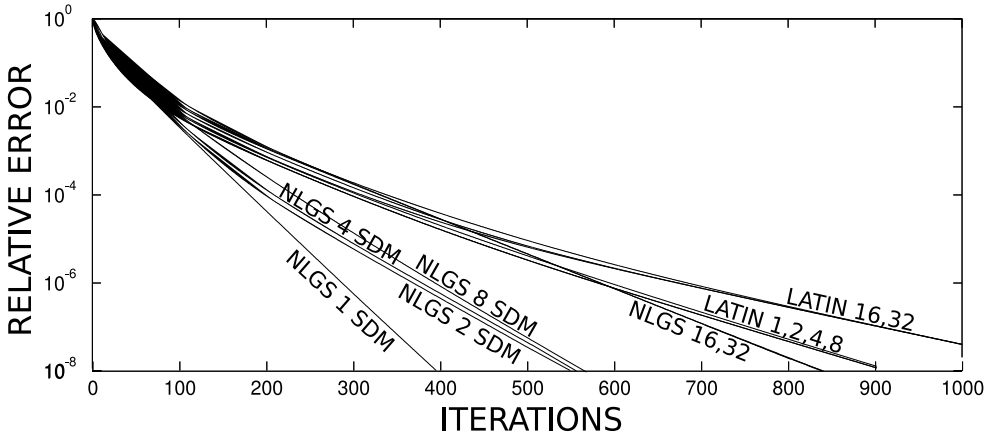


FIG. 2.12 – Évolution de l'erreur relative en fonction des itérations pour $k = k_1 = 50k_0$.

L'extensibilité numérique est perdue quand le nombre de sous domaines dans ce milieu discret augmente mais ce n'est pas systématique (contrairement aux résultats connus pour un milieu continu).

– Pour $k = k_0$ (Figure 2.11) l'extensibilité se dégrade très régulièrement pour

NLGS. Ceci est moins vrai pour LATIN avec un taux de convergence similaire pour $n_{sd} = 4, 8, 16, 32$.

– Pour $k = k_1$, correspondant à un treillis plus "rigide", la situation est moins claire, la dégradation de l'extensibilité s'effectuant "par saut". Ainsi pour NLGS on identifie aisément trois comportements : $n_{sd} = 1$, puis $n_{sd} = 2, 4, 8$ et enfin $n_{sd} = 16, 32$. Avec LATIN on ne retrouve que deux "paquets" : $n_{sd} = 1, 2, 4, 8$, puis $n_{sd} = 16, 32$.

2.4.2 Résultats pour un milieu granulaire

Le problème précédent est cette fois considéré avec des interactions de contact sans frottement (2.16). Dans ce cas on a pu constater lors de ces simulations une très forte sensibilité aux données initiales. Pour initialiser un pas de temps, utiliser des données stockées précédemment dans un fichier formaté avec 14 chiffres significatifs (notamment pour les impulsions de contact), ne s'avère pas suffisant pour avoir des comportements comparables des solveurs et de leurs solutions, il a fallu pour cela utiliser plus de 20 chiffres significatifs pour traiter les exemples qui possédaient une solution unique.

Cependant même sans frottement, l'unicité de la solution n'est pas garantie (en particulier pour les impulsions) dans le cas général, notamment avec un grand nombre de grains et des déplacements imposés en conditions aux limites. De ce point de vue, l'introduction du frottement amplifiera encore cet effet c'est pourquoi l'erreur relative est remplacée par un indicateur relatif de convergence, à l'itération i,

$$\zeta_i = \sqrt{\frac{(r_i - r_{i-1})^T (r_i - r_{i-1})}{r_i^T r_i}}. \tag{2.65}$$

La Figure 2.13 trace son évolution au cours des itérations d'un pas de temps, pour chacune des méthodes avec 1 à 8 sous domaines. Dans ce cas test, toutes les convergences sont semblables. On notera qu'avec un zoom très fin on retrouve l'ordre précédent des taux de convergence mais que les écarts sont négligeables.

Les deux méthodes différentes, avec leurs décompositions, semblent avoir le même comportement de convergence. Ces méthodes semblent être de plus en plus extensibles quand les comportements d'interaction sont de plus en plus non réguliers, alors qu'aucun enrichissement multiéchelle n'est pour l'instant introduit. Ceci peut être attribué d'une part à la rusticité des solveurs non linéaires (basés sur une méthode de point fixe) en comparaison à d'autres algorithmes plus sophistiqués de type gradient conjugué par exemple en décomposition de domaine [17, 21]. Il faut noter que cette dernière approche n'est pas adaptée au problème que l'on traite ici, où la non régularité est diffuse sur l'ensemble du domaine [51] et non restreinte aux interfaces.

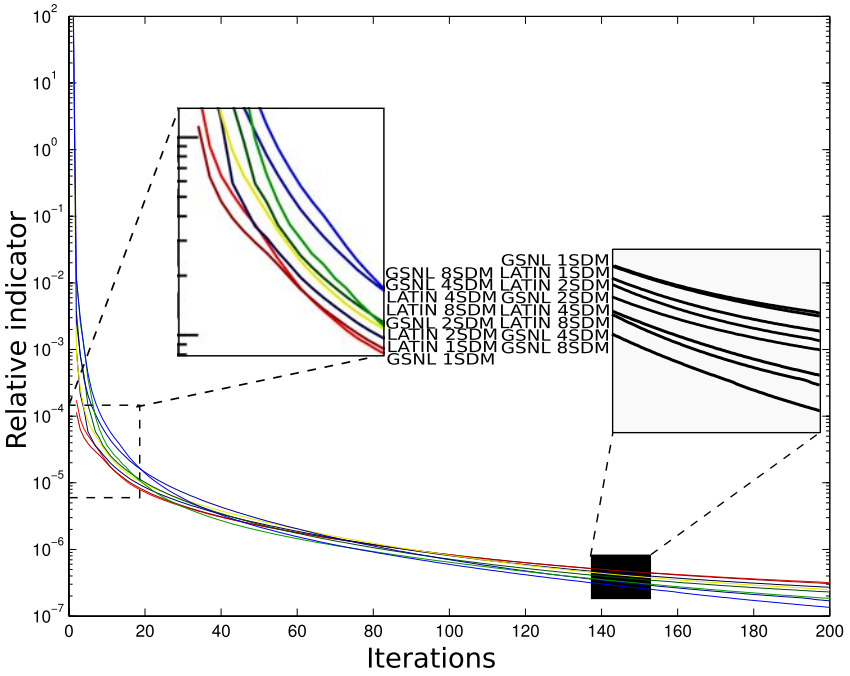


FIG. 2.13 – Évolution de l'indicateur de convergence en fonction des itérations pour le cas non linéaire.

Utiliser une version multiéchelle de la présente approche aurait tout de même certains avantages. Le premier est qu'elle permettrait l'accès à un comportement homogénéisé des sous domaines, un outil pratique pour permettre de coupler deux modèles à deux échelles différentes [48]. Le second est, au cours d'un problème d'évolution, de ne pas avoir à effectuer les calculs sur certains sous domaines pendant quelques pas. Cependant, avant de passer à cette échelle supplémentaire, on souhaite s'intéresser aux résultats obtenus dans le cas de contacts frottants qui sont véritablement l'objectif de cette thèse pour traiter des milieux granulaires réalistes. Jusqu'à présent les échantillons étaient générés par LMGC90 et après programmation d'un module permettant d'extraire les données de ce logiciel, les calculs étaient lancés sur MATLAB. La partie traitement du contact frottant étant plus lourde à mettre en place et existant déjà dans LMGC90, le traitement du contact frottant a été réalisé directement dans ce logiciel. Pour cela, il a fallu y incorporer différents éléments propres à la décomposition de domaine, ce que l'on va décrire dans le chapitre suivant.

2.5 Implantations dans LMGC90

L'objectif est maintenant de traiter des interactions plus complexes comme du frottement (définies à la section 2.1.3) en utilisant LMGC90, un code par élément discret, développé et maintenu par F. Dubois.

2.5.1 Échanges avec MPI

Dans les calculs effectués dans [50] la programmation était séquentielle. C'est à dire que le programme est traité par un et un seul processus (Figure 2.14). Toutes les variables et constantes du programme sont allouées dans la mémoire centrale affectée au processus, qui s'exécute sur un seul processeur physique de la machine. Des mécanismes internes au processeur peuvent alors permettre d'accélérer plus ou moins les calculs (vectorisation...).

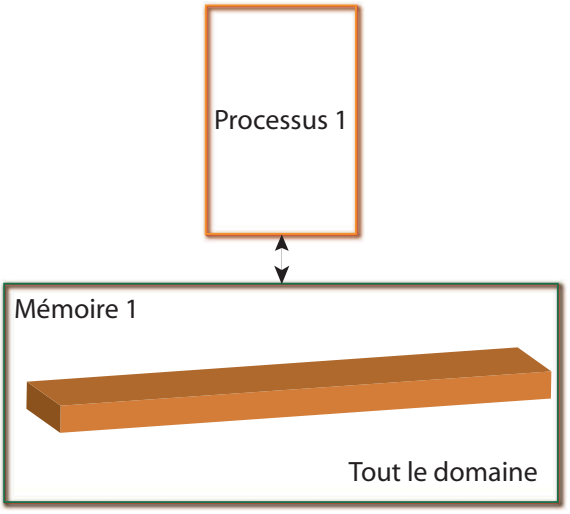


Fig. 2.14 – Architecture monoprocesseur.

Dans la plate-forme LMGC90, pour traiter des exemples de grande taille (de l'ordre de 50 000 grains en tridimensionnel ou de 500 000 en bidimensionnel) le code est développé de manière à pouvoir utiliser l'interface de programmation pour le calcul parallèle OpenMP sur des architectures à mémoire partagée. On utilise alors

un ensemble de processus qui ont tous accès à la même mémoire et on répartit les calculs à réaliser entre ces processus qui peuvent êtres répartis sur l'ensemble des processeurs physiques de la machine (Figure 2.15). On peut bénéficier ainsi des capacités de calculs des machines modernes multiprocesseurs jusqu'à un nombre modéré de processeurs efficacement utilisés en parallèle.

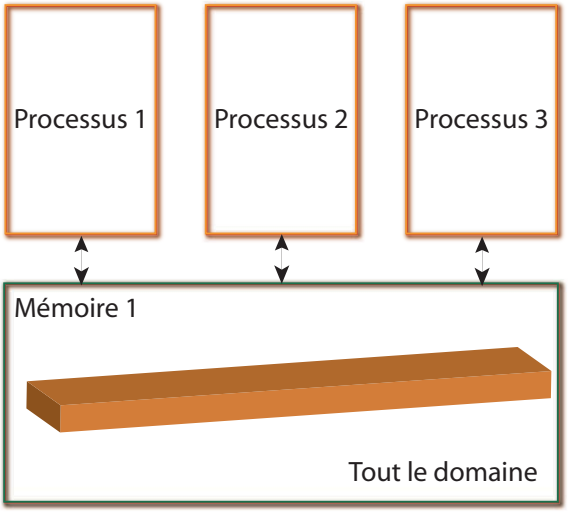


FIG. 2.15 – Architecture multiprocesseur à mémoire partagée.

Toutefois pour augmenter encore la taille des calculs accessibles, le passage à une architecture distribuée est nécessaire. Ce type d'architecture est, pour des milieux continus, plus efficace que la mémoire partagée, associée à de la décomposition de domaine. Elle permettra de ne stocker sur les différentes mémoires que l'information nécessaire à chaque processeur, les algorithmes locaux traitant des domaines plus petits (notamment s'il y a nécessité de réaliser des assemblages de matrices). Ces algorithmes peuvent être plus efficaces qu'un algorithme appliqué à tout le domaine, et on peut arriver à gérer les sous domaines où un critère déterminerait si on peut suspendre provisoirement le calcul non linéaire et ainsi alléger la charge. MPI[2] [60] "Message Passing Interface" est une norme définissant une bibliothèque de fonctions qui permet d'exploiter des ordinateurs distants ou multiprocesseurs

[2]http ://www.open-mpi.org

par passage de messages. Dans le modèle de programmation par échange de messages, chaque processus exécute un programme éventuellement différent, toutes les variables du programme sont privées et résident dans la mémoire locale allouée à chaque processus.

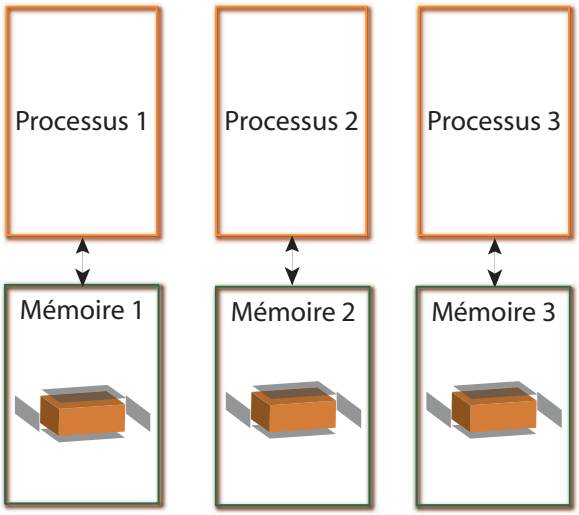


FIG. 2.16 – Architecture multiprocesseur à mémoire distribuée.

Un cas particulier qui nous intéresse ici est le modèle d'exécution SPMD "Single Program, Multiple Data" pour lequel le même programme est exécuté par tous les processus et qui a l'avantage d'être supporté par toutes les machines. En effet on dispose déjà du logiciel LMGC90, permettant de réaliser les calculs avec de nombreuses options de lois d'interactions, qui fonctionne bien pour effectuer les calculs désirés mais dont on veut augmenter les performances, pour permettre le calcul à plus grande échelle. LMGC90 est composé de plusieurs couches, une couche en langage Fortran à la base du logiciel, une couche en langage Python qui se veut plus éloignée de la programmation machine et une couche en langage C pour faire le lien entre les deux. L'idée est donc d'utiliser cette structure déjà fonctionnelle, d'en lancer plusieurs instances et d'arriver à les faire dialoguer entre elles. On voit dès lors que c'est la couche Python qui va permettre ce dialogue.

Il reste à savoir comment piloter ces instances à un niveau plus élevé, mettre en place le solveur multidomaine proposé, sans modifier le solveur du logiciel en

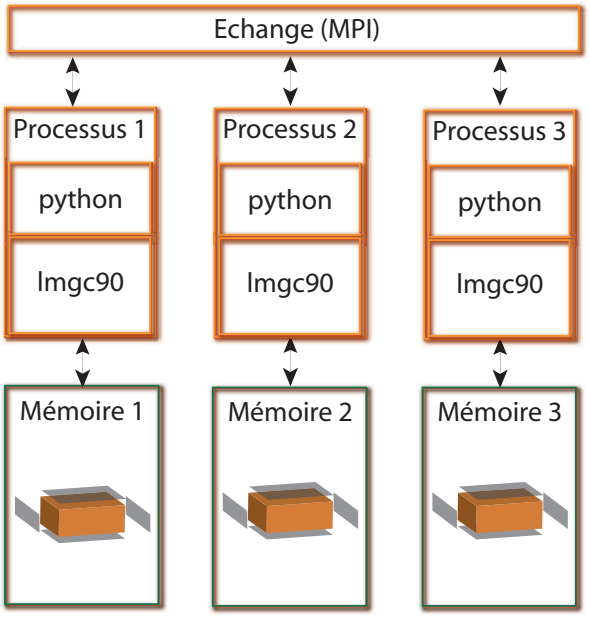


FIG. 2.17 – Architecture à mémoire distribuée : adaptation à la structure LMGC90.

lui-même. En fait, l'algorithme proposé est composé de deux étapes : la première est traitée par sous domaine où on exécute une résolution locale des interactions de contact. Cette opération est exactement la même que celle du solveur initial mono-domaine, opérant sur l'ensemble du domaine. Donc, en lançant plusieurs instances de LMGC90 on peut réaliser cette résolution directement par sous domaine. Cependant, il reste à ajouter le multiplicateur de Lagrange comme effort imposé pour faire le lien entre les sous domaines. Ainsi mises à part les difficultés inhérentes au développement numérique comme la récupération et le traitement des vitesses et impulsions imposées, la première étape sera réalisée par chaque instance avec son propre solveur (ici NLGS en l'occurrence) dans la couche Fortran.

La deuxième étape consiste à faire dialoguer l'ensemble des sous domaines et correspond à une synchronisation des différents processus. Chaque sous domaine, à la fin du calcul de l'étape précédente, réalise une projection de ses grandeurs nécessaires sur son interface. En établissant les différents liens nécessaires entre Fortran et Python on fait alors remonter des données par sous domaine à un niveau plus haut. Avec des procédures d'échange MPI on assure le dialogue entre ces interfaces et les calculs correspondant à la deuxième étape de l'algorithme sont exécutés, chacune ayant récupéré les données de l'interface globale.

2.5.2 Structure et mise en place

Sans entrer dans les détails du code, sa structure, basée sur l'algorithme 1 et permettant de répondre à la démarche proposée dans la section précédente 2.5.1, on s'attache dans cette section à mettre en évidence les flots d'échanges de données ainsi que le poids des différents calculs.

Un nom arbitraire a été fixé pour chacune des étapes du calcul de manière à pouvoir les identifier dans la suite. Sur la droite de la Figure 2.18 est représentée la taille mémoire de chacun des sous domaines. En effet, la procédure n'est ici pas optimale. De manière à pouvoir faire le découpage en sous domaines, à un moment donné chaque sous domaine charge l'ensemble des données du domaine complet en mémoire pour pouvoir calculer le graphe des contacts, y superposer sa grille et ensuite ne garder que ses propres contacts. On pourrait facilement optimiser ceci soit en utilisant un processus maître qui serait le seul à faire cette détermination, mais il devrait tout de même charger l'ensemble du domaine et réaliser des échanges lourds avec les autres processus une fois son calcul effectué. Une autre méthode serait d'avoir une grille grossière, et dans une case de cette grille grossière (au lieu de parcourir l'ensemble du domaine), chacun des sous domaines va chercher à savoir si les contacts lui appartiennent. Toutefois, afin de limiter la partie développement déjà importante, il a été choisi de laisser de côté les développements dits d'optimisation, n'empêchant pas le développement d'algorithmes performants et qui ne sont pas l'objet de ce manuscrit.

La notion de couche Fortran ou Python est importante car même en évitant de trop dupliquer l'information en utilisant des pointeurs, les temps d'échange entre couches ne sont pas forcément négligeables vis a vis des échanges MPI.

Les échanges MPI sont dans l'ensemble assez légers en termes de quantité de données, puisqu'il sont limités aux quantités d'interface. Ceux qui nous importent vis-à-vis du temps de calcul sont ceux réalisés en DD7 pour transmettre les efforts d'interface entre les sous domaines et ceux réalisés en DD6 pour récupérer les sauts de vitesses aux interfaces. Ces échanges sont réalisés à chaque itération et il a été nécessaire de les optimiser un minimum de manière à ne pas trop pénaliser les temps de calculs. En revanche, l'échange M1 réalisé en fin de pas de temps est assez important car il permet de transmettre l'ensemble des données à chacun des sous domaines. En effet, si l'on ne transfère pas toutes les données locales on risque, lorsque chacun des sous domaines calcule un nouveau découpage, que celui-ci ne soit pas identique pour tous les processus. Une possibilité pour les problèmes où les positions n'évoluent pas beaucoup est de ne pas refaire le découpage à chaque pas de temps mais de le faire évoluer dynamiquement. Par souci de simplicité ce n'est pas cette option qui a été choisie ici.

2.5.3 Temps de calcul et nombre d'itérations

On représente maintenant le temps pris par chacune de ces étapes à chaque moment ou elle est réalisée, sur un unique pas de temps. La première figure (Figure 2.19) illustre ce qui se passe pour une simulation avec mille grains (décrite ci-après) avec une décomposition en 4 sous domaines utilisant 4 processeurs. Les temps illustrés sont les temps CPU par processeur. Il apparaît clairement ici que les temps d'échange prennent le dessus devant les temps de calcul. Les étapes DD6 et DD7 sont prédominantes devant les temps de calcul (S2 et S3). De plus, le temps de transmission des informations en fin de pas de temps M1 semble lui aussi important devant les temps de calcul.

Toutefois, le nombre d'itérations effectuées par pas de temps est plus proche du millier que de la dizaine représentée ici afin de voir ces étapes au cours du calcul. Ainsi, il est difficile d'évaluer avec ces seules dix itérations l'importance de l'initialisation par rapport au calcul global. On représente sur la Figure 2.20 pour chaque étape les temps cumulés sur un pas de temps. De plus, on a choisi ici de décomposer l'étape d'échange DD6 en plusieurs sous étapes. (On pourrait faire de même pour DD7 et s'attendre à ce que les conclusions sur les temps d'échange pour transmettre les informations de l'interface jusque au sous domaines aient une répartition comparable à ceux de DD6, des sous domaines vers l'interface). L'intérêt est que l'on identifie les phases coûteuses : ici c'est la partie DD6a qui correspond à des échanges entre Python et Fortran. Ce surcoût est lié à la réalisation d'opérations de bas niveau dans un langage interprété comme Python. Le remède consisterait à coder ces opérations dans une routine Fortran, appelée une seule fois depuis l'interpréteur Python. La partie DD6b est bien plus légère (négligeable devant DD6a), elle correspond au calcul de la contribution du sous domaine au second membre $A_{E\Gamma}^T V_E$. Le temps d'échange et de sommation $\sum_{E=1}^{n_{sd}} A_{E\Gamma}^T V_E$ entre les sous domaines réalisé par MPI en DD6c lui n'est pas négligeable. Cependant ce temps d'échange est aussi un lieu de synchronisation entre les processus. Ainsi le déséquilibre de charge entre les sous domaines provoque un temps d'attente par processus cumulé sur l'ensemble des itérations et on peut imaginer qu'en diminuant les temps globaux (donc DD6a) les temps d'attente diminuent en conséquence. On comprend donc qu'ici l'optimisation ne pourra être effectuée qu'avec une étude lourde de la programmation. En revanche, la durée de mise en commun des données en fin de pas de temps (M1) est négligeable.

On voit très nettement que les temps de calculs S2 sont négligeables devant les temps d'échanges DD6 et DD7. Cependant, il convient de noter qu'on se place ici dans le cas le plus coûteux en échanges : une seule itération de Gauss Seidel (S2) entre chaque mise à jour de l'interface, ceci peut être modifié de manière à diminuer

le surcoût liés aux échanges ; cependant des optimisations de code restent à prévoir. Le but de ce manuscrit étant d'abord la mise en place d'algorithmes, il ne sera plus fait état dans la suite de ces temps CPU, nécessaires en revanche à la recherche d'optimisations qui devra être menée pour rendre le code efficace. On raisonnera alors sur le nombre d'itérations nécessaires à la résolution d'un pas de temps, en gardant en tête le coût des calculs en fonction de la taille du problème en mémoire (que ce soit un domaine complet ou un sous domaine plus ou moins grand).

On va tâcher d'étudier l'extensibilité des algorithmes, qui permet, traduisant la stabilité du nombre d'itérations à convergence vis à vis du nombre de sous domaine. Ainsi, avec un nombre identique d'itérations ainsi que des sous domaines plus petits donc moins lourds à calculer, on gagnera du temps de calcul.

2.5.4 Résultats avec frottement

Le premier exemple est un essai de grande taille à deux dimensions avec contact frottant. Le coefficient de frottement est $\mu = 0.3$, et 100 000 grains ont été déposés par gravité pour obtenir l'échantillon de la Figure 2.21. Les bords de droite et de gauche ont une rotation imposée. Le bord inférieur est bloqué. La boîte est découpée en deux sous domaines.

L'algorithme générique 1 est utilisé et deux indicateurs de convergence sont calculés au cours des itérations. Bien sûr, aucune erreur de référence n'est disponible du fait de la non unicité de la solution. L'indicateur relatif de convergence en volume est calculé à chaque itération, et sommé sur l'ensemble du domaine. Il est défini à l'itération i de la manière suivante,

$$\zeta_i = \sqrt{\frac{(r_i - r_{i-1})^T (r_i - r_{i-1})}{r_i^T r_i}}. \tag{2.66}$$

Le second indicateur est défini uniquement aux interfaces, et permet de mesurer la stationnarité des inter-impulsions,

$$Z_i = \sqrt{\frac{(F_{\Gamma,i} - F_{\Gamma,i-1})^T (F_{\Gamma,i} - F_{\Gamma,i-1})}{F_{\Gamma,i}^T F_{\Gamma,i}}}. \tag{2.67}$$

Leurs évolutions au cours des itérations sont décrites Figure 2.22. Ces évolutions sont assez similaires, et le deuxième indicateur sera utilisé dans la suite, étant moins coûteux à calculer.

On étudie maintenant la sensibilité au nombre de sous domaines, on compare un découpage en 2 et 4 sous domaines que l'on illustre Figure 2.23. En se basant uniquement sur l'indicateur de convergence, dont la courbe est tracée au cours des

itérations sur un même pas de temps pour les découpages en 2 et 4 sous domaines, Figure 2.24, les vitesses de convergence sont très similaires comme dans la partie sans frottement. Basé sur ce critère uniquement l'algorithme semble extensible.

Pour valider le calcul précédent dans un cas 3D on a réalisé l'implantation du code dans la version 3D de LMGC90. L'essai reste un essai de cisaillement, 1 000 sphères sont déposées dans une boite fermée (la paroi frontale étant transparente). On réalise comme précédemment un essai de cisaillement qui permet de passer de la Figure 2.25 à la Figure 2.26.

Sur la Figure 2.27 on trace, comme dans le cas 2D, les indicateurs de convergence en volume et à l'interface au cours des itérations sur un pas de temps. Si on se base sur l'indicateur en volume, le comportement de convergence est très similaire au cas 2D. Il n'est cependant pas possible de comparer les vitesses de convergence tant les problèmes sont différents, en effet on a d'un côté en 2 dimensions 100 000 disques et de l'autre en 3 dimensions 1 000 sphères. L'évolution des deux indicateurs n'est plus tout à fait identique. La convergence de l'indicateur d'interface est moins régulière. Ceci est peut-être dû à la taille réduite en terme de nombre de grains de l'interface.

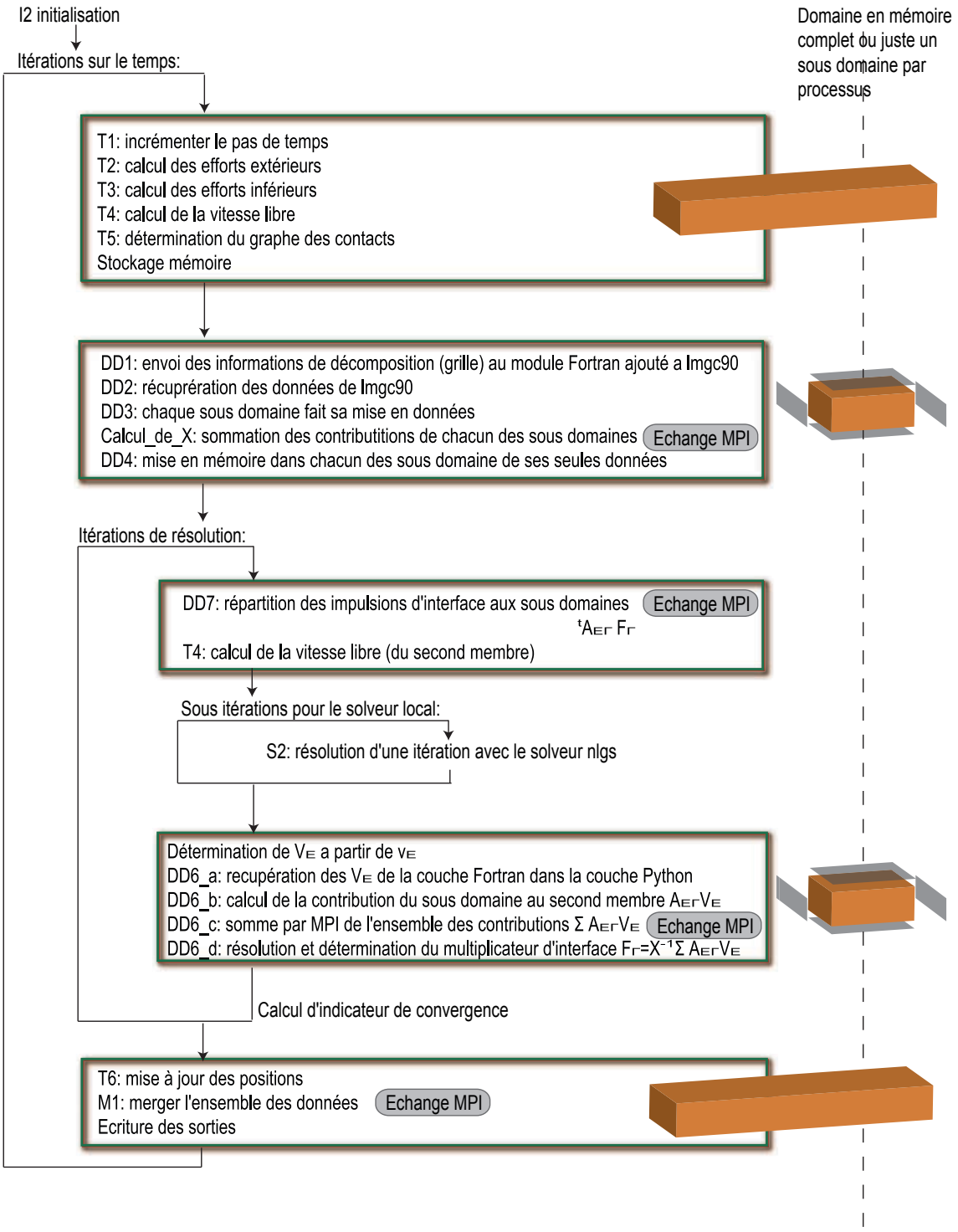


FIG. 2.18 – Structure du programme.

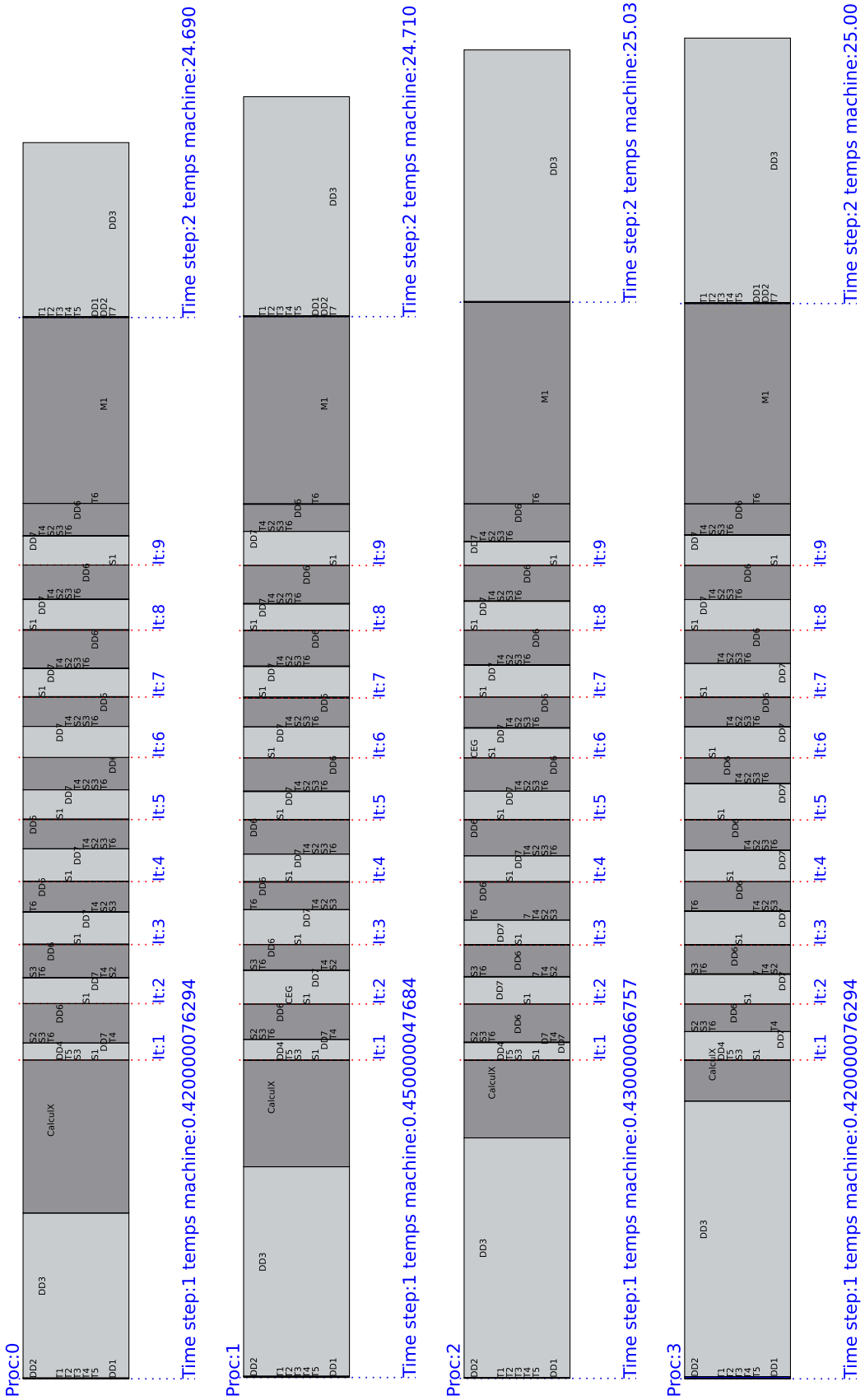


FIG. 2.19 – Répartition du temps cpu par étapes sur un pas de temps avec dix itérations.

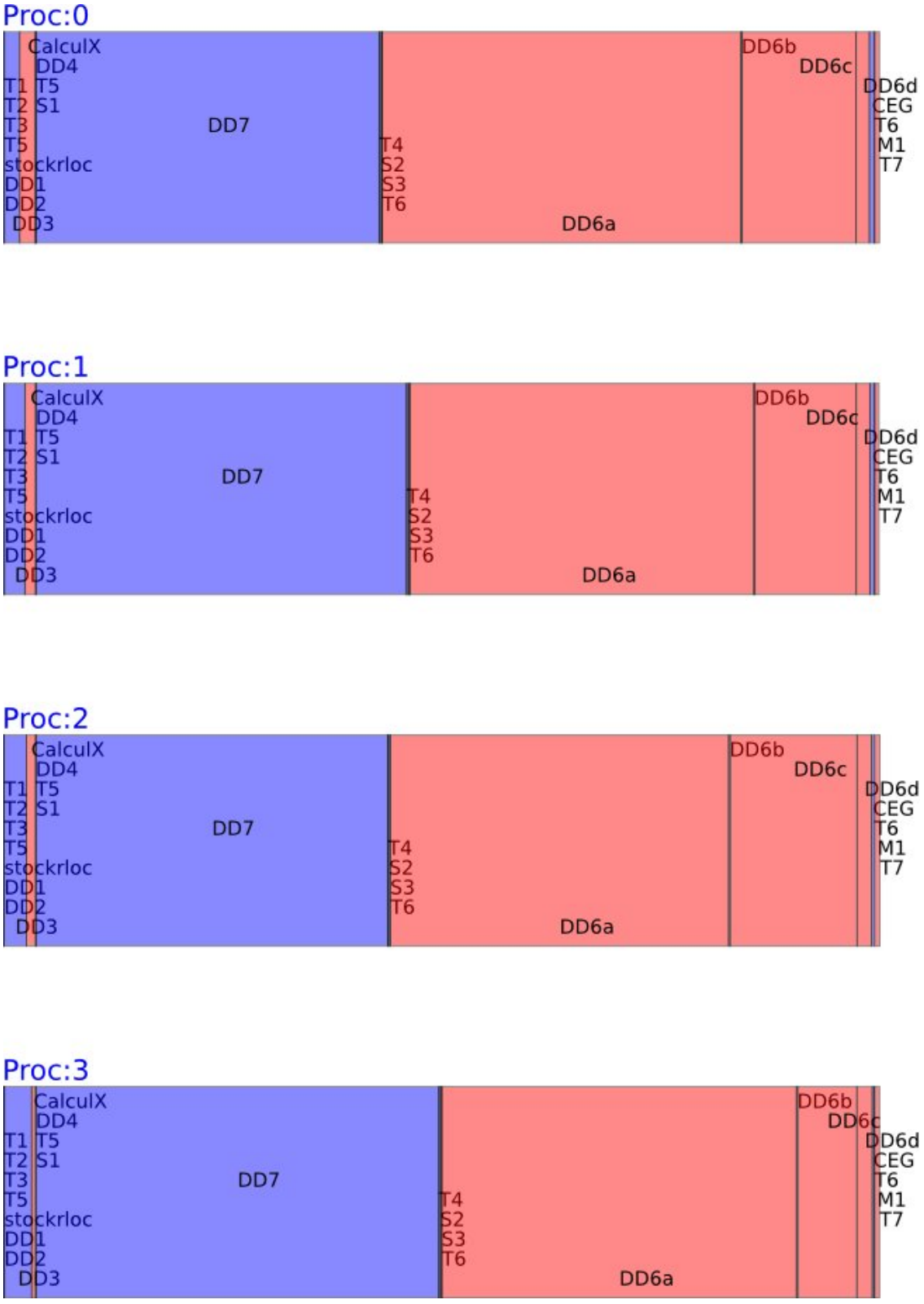


FIG. 2.20 – Répartition du temps CPU cumulé sur les itérations par étape sur un pas de temps avec 2 000 itérations.

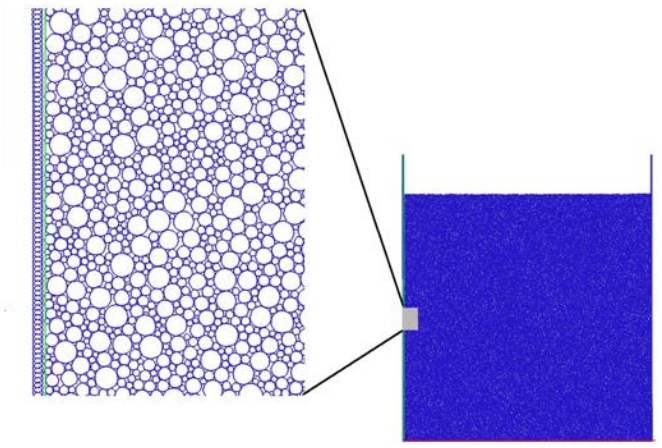


FIG. 2.21 – Echantillon bidimensionnel de 100 000 disques avec frottement pour essai de cisaillement.

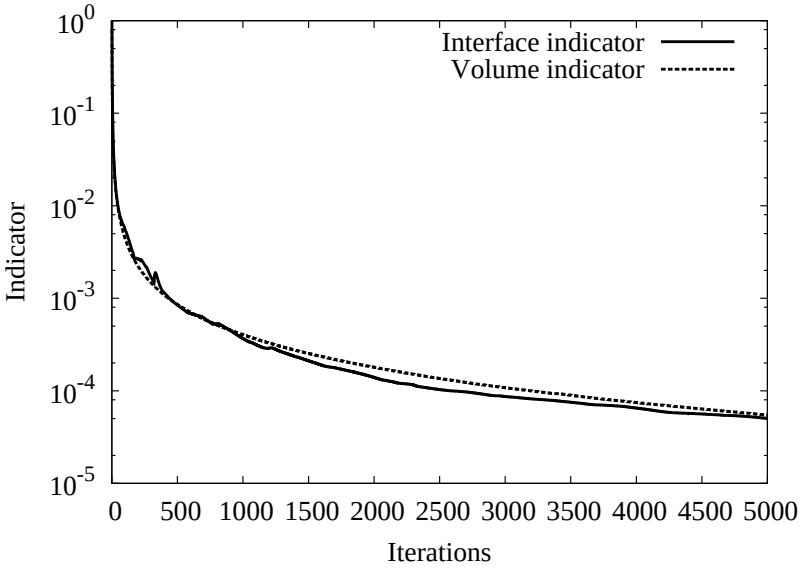


FIG. 2.22 – Comparaison des indicateurs de volume et d'interface sur l'essai de cisaillement bidimensionnel.

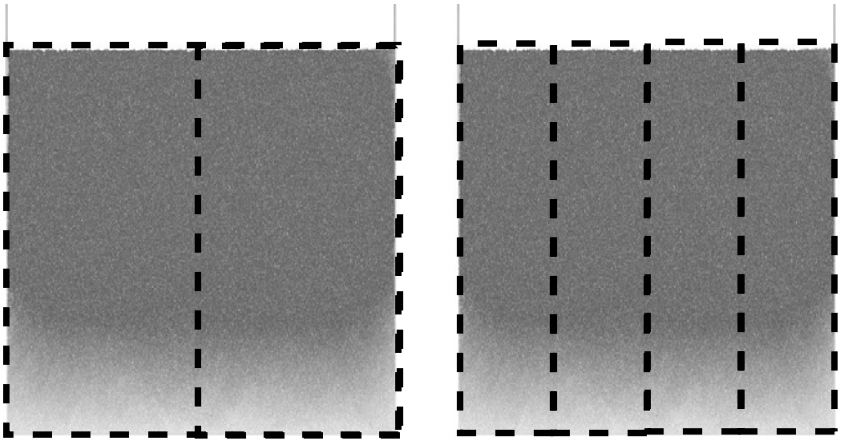


FIG. 2.23 – Découpage en 2 et 4 sous domaines sur l'essai de cisaillement bidimensionnel.

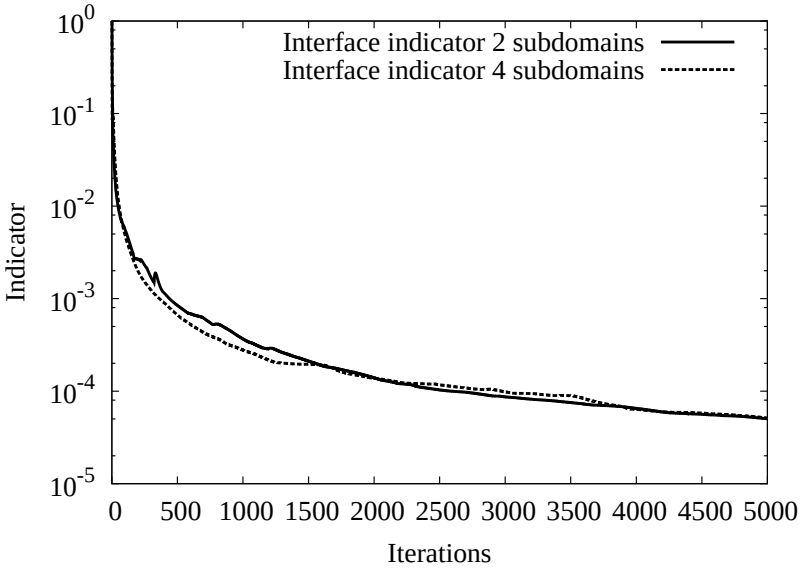


FIG. 2.24 – Comparaison des indicateurs d'interface avec 2 ou 4 sous domaines sur l'essai de cisaillement bidimensionnel

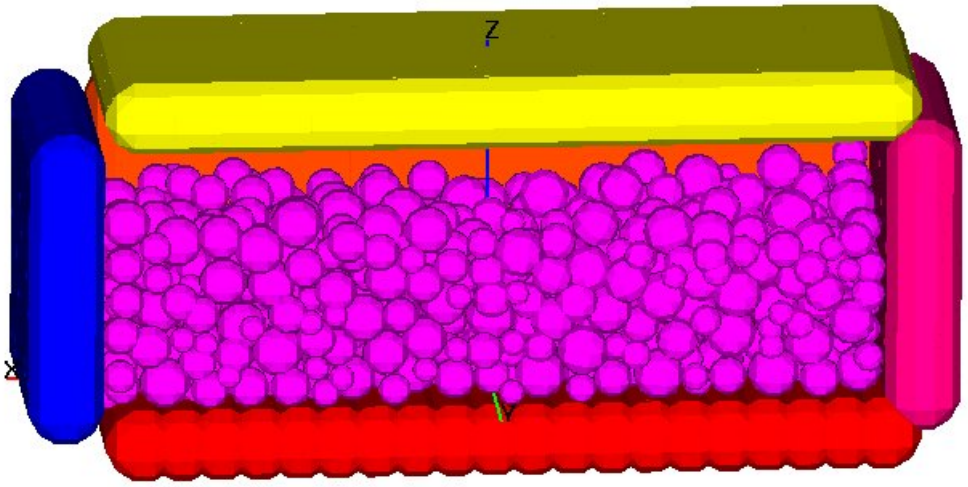


Fig. 2.25 – Début d'un essai de cisaillement tridimensionnel avec 1 000 sphères avec frottement.

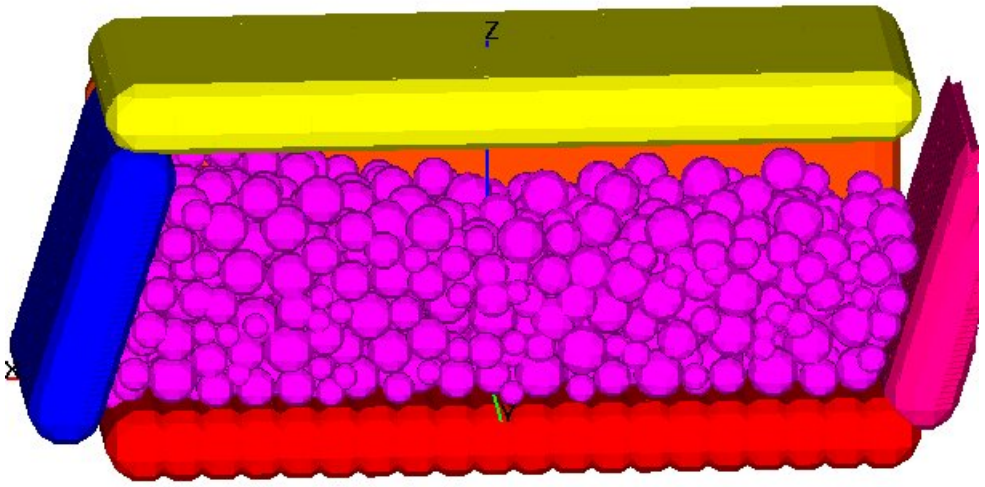


Fig. 2.26 – Fin d'un essai de cisaillement tridimensionnel avec 1 000 sphères avec frottement.

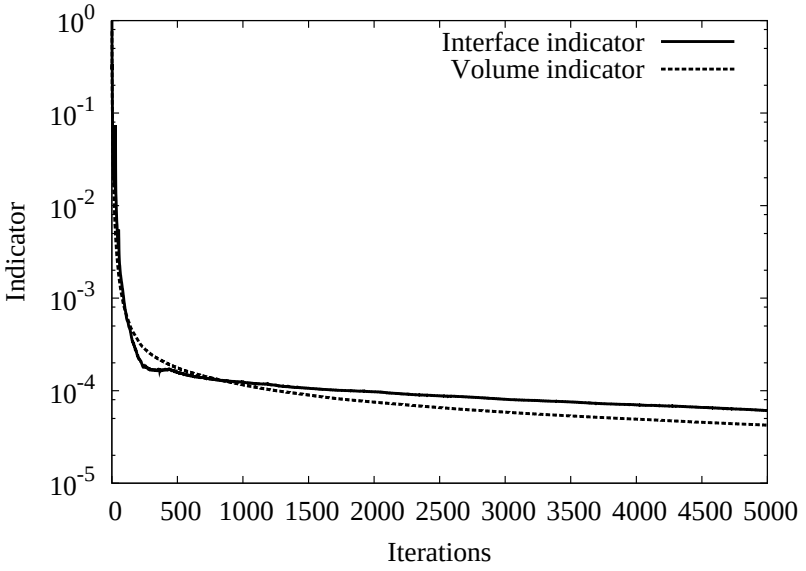


FIG. 2.27 – Comparaison des indicateurs de volume et d'interface sur l'essai de cisaillement tridimensionnel découpé en deux sous domaines.

Chapitre 3

Introduction d'une échelle macro

Sommaire

3.1 Définition des champs macro 72
3.2 Algorithme de type LATIN multiéchelle 78
3.3 Algorithme de type GSNL multiéchelle 82
3.4 Faible efficacité de l'échelle macroscopique 84

Sur la base des essais effectués dans le chapitre précédent, l'extensibilité numérique des algorithmes monoéchelles semble préservée sans recourir à un problème grossier ni à l'introduction d'une échelle macro. Cependant il peut être intéressant de disposer d'un comportement tangent homogénéisé à l'échelle d'un sous domaine. En effet si ce comportement s'avère linéaire et stable sur plusieurs pas de temps, il peut être conservé sans effectuer de résolution locale non linéaire sur un certain nombre de pas ultérieurs (la validation de cette stratégie est cependant hors du propos de ce mémoire).

Pour introduire cette échelle macro, on peut imaginer étendre les approches déjà utilisées en milieu continu, (comme pour la LATIN) au cas de la dynamique granulaire, voire même à un algorithme de type GSNL en s'appuyant sur les comparaisons algorithmiques déjà établies au chapitre 2.3.3.

La démarche employée est alors la suivante :

- on procède à un découpage du domaine discret en sous domaines et interfaces comme décrit au chapitre 1.4 ;
- pour les champs définis à l'interface, on utilise une décomposition additive en une partie macro, adaptée au cas des milieux granulaires, et une partie micro complémentaire ;
- en ayant choisi un partitionnement tel que le comportement d'interface est parfait (et linéaire), les parties macros vérifient aussi ce comportement ; on impose alors un raccord cinématique supplémentaire (macro) au cours des itérations.

  Ces différentes étapes vont être décrites dans la suite.

3.1 Définition des champs macro

3.1.1 Une stratégie multiéchelle

Une fois la sous-structuration effectuée, la première étape consiste toujours à définir le problème à partir des champs microscopiques et macroscopiques, champs portant sur les quantités d'interfaces du problème sous-structuré. Par interface locale $\Gamma_{EE'}$ entre deux sous structures E et E', étant donnés un champ discret d'impulsions moyennes $F_{EE'}$ aux noeuds de l'interface agissant sur E et un champ discret de vitesses $V_{EE'}$ aux noeuds de l'interface connectés à E, nous décomposons en deux partie ces champs : une partie Macro, marquée d'un exposant M, et une partie complémentaire micro, marquée d'un exposant m,

$$V_{EE'} = V^M_{EE'} + V^m_{EE'} \quad \text{et} \quad F_{EE'} = F^M_{EE'} + F^m_{EE'}. \tag{3.1}$$

Les espaces micro et macro doivent êtres définis de façon unique : ils doivent être orthogonaux, c'est pourquoi un projecteur orthogonal doit être utilisé dans chaque sous espace. En écrivant l'orthogonalité au sens de l'énergie, celle-ci s'écrit donc de la façon suivante,

$$\sum_{EE'} (F_{EE'})^T V_{EE'} = \sum_{EE'} (F^M_{EE'})^T V^M_{EE'} + (F^m_{EE'})^T V^m_{EE'}. \tag{3.2}$$

3.1.2 Représentation macro

La partie macro est représentée sur un espace de petite taille : les champs macros sont définis à l'aide de peu de paramètres ; une base des champs macros peut être choisie comme suit : $F^M_{EE'} = P_{EE'} f_{EE'}$ où $P_{EE'}$ est le projecteur contenant l'ensemble des vecteurs de base et $f_{EE'}$ un vecteur contenant les paramètres macros des torseurs impulsions. Par simplicité on choisit : $V^M_{EE'} = P_{EE'} w_{EE'}$ de manière à conserver les mêmes vecteurs de base. Les paramètres macro peuvent être choisis en fonction du problème à modéliser ; ici, nous choisissons les moyennes généralisées des champs $F_{EE'}$ et $V_{EE'}$ sur $\Gamma_{EE'}$, d'ordre 1.

Un premier projecteur macro en moment et résultante, en termes de vitesses, pourrait être un extracteur des mouvements de corps rigide de l'interface. Les vitesses macroscopiques représentent la cinématique de corps rigides des interfaces, et les impulsions, la résultante et le moment duaux d'une telle cinématique. L'enrichissement de cette base revient, du point de vue de la cinématique macroscopique, à considérer l'interface comme un milieu déformable, au lieu d'un simple corps rigide.

Soient les centres de masse des grains appartenant à l'interface, de coordonnées x_i, x_G les coordonnées du centre de masse de l'interface et (n_1, n_2, n_3) une base associée à cette interface (avec n_1, n_2 dans le plan de l'interface et n_3 le vecteur normal).

Si on fait travailler l'ensemble des torseurs des impulsions aux différents nœuds, dans un torseur cinématique représentant un champ virtuel de vitesse constante, on obtient,

$$\sum_i \left\{ \begin{array}{c} F_i \\ C_i \end{array} \right\}_{x_i} \cdot \left\{ \begin{array}{c} 0 \\ V^* \end{array} \right\}_{.} = \underbrace{\left(\sum_i F_i \right)}_{F^M} V^*, \tag{3.3}$$

où F^M contient les trois composantes de la résultante moyenne généralisée de $F_{EE'}$.

Notons qu'ici, les couples C_i interviennent. C'est une particularité du milieu granulaire où chaque demi-grain agit sur sa moitié par une résultante, mais aussi par un moment en son centre de masse. Ce n'est pas le cas d'un système de tenségrité

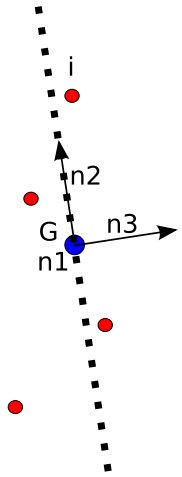


FIG. 3.1 – Interface avec 4 grains i et le centre de masse G.

où un noeud, point, ne peut transmettre que la résultante et pour la même raison ces micro-couples n'apparaissent pas non plus dans le cas de milieux continus discrétisés.

F^M est alors la résultante duale du mouvement de translation de solide rigide correspondant à V^* et les trois fonctions de base correspondant à la projection sur l'espace des translations sont :

$$e_k^M = n_k \quad k = 1, 2, 3 \quad \text{appliqué aux résultantes.} \tag{3.4}$$

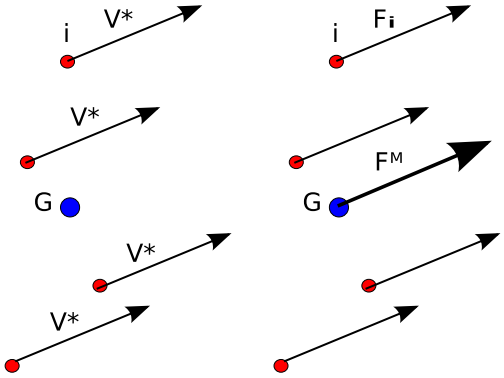


FIG. 3.2 – Champ de vitesses virtuelles V^* constant / Efforts duaux et généralisés.

Pour exprimer le mouvement de solide rigide de l'interface il manque les degrés de liberté en rotation. Pour les obtenir, on fait travailler l'ensemble des torseurs des

efforts des différents noeuds, par un torseur représentant un champ virtuel de vitesse de rotation uniforme autour de G,

$$\sum_i \left\{ \begin{array}{c} F_i \\ C_i \end{array} \right\}_{x_i} \cdot \left\{ \begin{array}{c} \Omega^* \\ 0 \end{array} \right\}_{x_G} = \underbrace{\sum_i \left(C_i + F_i \wedge x_i x_G\right) \Omega^*}_{C^M}, \tag{3.5}$$

où C^M contient les trois composantes du moment moyen généralisé de $F_{EE'}$.

C^M est alors le moment dual du mouvement de rotation de solide rigide correspondant à Ω^* représenté Figure 3.3 et les trois fonctions de base correspondant à la projection sur l'espace des rotations uniformes sont :

$$e_k^M = x_i x_G \wedge n_k \text{ appliqué aux résultantes} + n_k \text{ appliqué aux moments} \quad k = 1, 2, 3 \tag{3.6}$$

se qui s'écrit

$$e_k^M = \left\{ \begin{array}{c} x_i x_G \wedge n_k \\ n_k \end{array} \right\} \tag{3.7}$$

tel que,

$$C^M = \sum_i e_i^M \left\{ \begin{array}{c} F_i \\ C_i \end{array} \right\}. \tag{3.8}$$

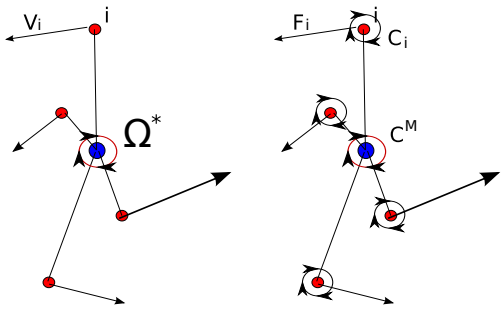


FIG. 3.3 – Champ de vitesses virtuelles de rotation Ω^* constante / Efforts duaux et généralisés.

De manière à enrichir la cinématique des interfaces, on fait le choix d'utiliser un projecteur dit d'extraction de la "partie linéaire" des champs. Pour cela on utilise un champ virtuel de vitesse traduisant l'allongement dans deux directions et la

déformation de cisaillement. Ces allongements dépendent linéairement de la distance du point considéré, i, au centre de masse G.

Le champ de vitesse caractéristique d'un allongement dans les deux directions du plan de l'interface et d'un cisaillement dans ce même plan est :

$$\begin{array}{rcl} V_i^* & = & n_1 n_1^T (x_i - x_G)\, \alpha_1^* \\ & + & n_2 n_2^T (x_i - x_G)\, \alpha_2^* \qquad .. \\ & + & \left(n_1 n_2^T + n_2 n_1^T\right)(x_i - x_G)\, \alpha_3^* \end{array} \tag{3.9}$$

En faisant travailler ce champ de vitesses,

$$\begin{array}{c} \sum_i \left\{ \begin{array}{c} F_i \\ C_i \end{array} \right\}_{x_i} \cdot \left\{ \begin{array}{c} 0 \\ V_i^* \end{array} \right\}_{x_i} = \sum_i (F_i . V_i^*) \\ = \sum_i \left(n_1^T (x_i - x_G)\, n_1^T F_i \alpha_1^* + n_2^T (x_i - x_G)\, n_2^T F_i \alpha_2^* \right. \\ \left. + F_i^T \left(n_1 n_2^T + n_2 n_1^T\right)(x_i - x_G)\, \alpha_3^* \right) \end{array} \tag{3.10}$$

On obtient les deux moyennes des deux tensions et la moyenne du cisaillement dans le plan de l'interface.

Les trois fonctions de base correspondant à la projection sur l'espace des mouvements d'allongement et de cisaillement sont :

$$\begin{array}{lll} e_k^M = n_k n_k^T (x_i - x_G) & k = 1, 2 & \text{appliqué aux résultantes} \\ e_3^M = \left(n_1 n_2^T + n_2 n_1^T\right)(x_i - x_G) & & \text{appliqué aux résultantes} \end{array} . \tag{3.11}$$

Ainsi $f_{EE'}$ comprend les neufs paramètres que l'on vient de définir : les trois résultantes des forces, les trois moments, les deux tensions et le cisaillement dans le plan de l'interface. Pour sa part, w'_{EE} comprend les quantités duales qui sont les trois translations, les trois rotations, les deux allongements et le cisaillement définies virtuellement ci-dessus.

On remarque que 9 paramètres sont nécessaires par interface pour les champs de forces macros généralisés (et 9 paramètres duaux pour les champs de vitesses macros généralisées). Par rapport au cas de la tenségrité développé dans [50] le nombre et le type de champs macros est inchangé mais il a fallut prendre en compte les micros couples possibles aux interfaces dans l'expression de ces champs.
Comme simplification supplémentaire, la base $P_{EE'}$ est orthonormée. Les fonctions de base, que l'on vient de définir, que ce soit pour les moments ou les allongements ne sont pas celles qui sont exactement ajoutées à la base, car elles ne sont ni orthogonales aux autres fonctions de la base (au sens de la forme bilinéaire travail $(.,.)$), ni normées. Les fonctions ajoutées à la base sont déduites de celles-ci par le procédé d'orthogonalisation de Schmidt,

$$P_{EE'}^T P_{EE'} = I_d, \tag{3.12}$$

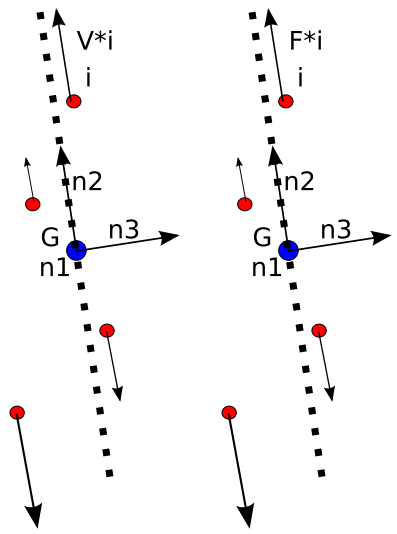


FIG. 3.4 – Champ de vitesses virtuelles pour la composante d'allongement sur n_2 / Efforts duaux et généralisés.

$$(F^M_{EE'})^T V^M_{EE'} = f^T_{EE'} P^T_{EE'} P_{EE'} w_{EE'} = f^T_{EE'} w_{EE'}. \tag{3.13}$$

Avec l'expression de l'énergie et l'orthogonalité,

$$w_{EE'} = {P_{EE'}}^T V_{EE'} \tag{3.14}$$

$$f_{EE'} = {P_{EE'}}^T F_{EE'} \tag{3.15}$$

$$P^T_{EE'} V^m_{EE'} = 0 = P^T_{EE'} F^m_{EE'}. \tag{3.16}$$

Admissibilité macro. Pour traduire le fait qu'en moyenne on doit avoir la continuité des vitesses et le saut d'effort nul aux interfaces ces relations sont imposées aux champs Macro,

$$\left\{ \begin{array}{l} F^M_{EE'} + F^M_{E'E} = 0 \\ V^M_{EE'} = V^M_{E'E} \end{array} \right. \quad \text{ou} \quad \left\{ \begin{array}{l} f_{EE'} + f_{E'E} = 0 \\ w_{EE'} = w_{E'E} \end{array} \right. . \tag{3.17}$$

Introduire l'admissibilité macro dès la formulation simplifie grandement la présentation de l'algorithme multiéchelle qui suit par rapport au travail précédent sur la tenségrité où cette admissibilité macro intervenait a posteriori [[50],Chap 6.3, p128].

3.2 Algorithme de type LATIN multiéchelle

Le solveur décrit ici est fortement inspiré de [50], cependant il est réécrit avec le formalisme du milieu granulaire. Ceci va permettre de mettre en place un premier algorithme mais aussi de commenter les résultats qui lui sont liés dans un cas de tenségrité. La méthode LATIN [50] est utilisée ici de façon incrémentale comme un solveur sur un pas de temps pour trouver la solution du problème de référence (2.26). Le principe étant ici de séparer les équations à résoudre en deux groupes, en introduisant des équations linéaires et locales comme directions de recherche. Les notations désignant les deux groupes d'équations sont conservées. L'étape consistant à construire $\widehat{s}$ connaissant s (l'étape "locale") nécessite des paramètres de direction de recherche positifs d^+ et l^+,

$$(\widehat{r}_E - r_E) - l^+ (\widehat{v}_E - v_E) = 0 \tag{3.18}$$

$$\left(\widehat{F}_{EE'} - F_{EE'}\right) + d^+ \left(\widehat{V}_{EE'} - V_{EE'}\right) = 0. \tag{3.19}$$

L'étape consistant à construire s connaissant $\widehat{s}$ (l'étape "linéaire") nécessite les paramètres d^- et l^-,

$$(r_E - \widehat{r}_E) + l^- (v_E - \widehat{v}_E) = 0 \tag{3.20}$$

$$\left(F_{EE'} - \widehat{F}_{EE'}\right) - d^- \left(V_{EE'} - \widehat{V}_{EE'}\right) = 0. \tag{3.21}$$

On a ainsi introduit 4 paramètres numériques l^+, l^-, d^+ et d^-. Le choix de ces paramètres permet d'agir sur les taux de convergence mais la solution à convergence n'en dépend pas. Le choix de paramètres optimaux et leur influence sera étudié dans la suite.

3.2.1 Étape locale.

Comme mentionné auparavant, cette étape consiste à trouver $\widehat{s}$ connaissant s calculé à l'étape précédente. Elle mène à des problèmes locaux et linéaires sur chaque interface parfaite $\Gamma_{EE'}$, et à des problèmes de complémentarité linéaire (LCP) diffus sur chaque sous structure E.

Par interface. On vérifie l'équilibre des impulsions et la continuité des vitesses,

$$\begin{cases} \widehat{F}_{EE'} + \widehat{F}_{E'E} = 0 \\ \widehat{V}_{EE'} = \widehat{V}_{E'E} \end{cases} . \tag{3.22}$$

On ajoute la direction de recherche,

$$\left(\widehat{F}_{EE'} - F_{EE'}\right) + d^{+}\left(\widehat{V}_{EE'} - V_{EE'}\right) = 0 \ . \tag{3.23}$$

Finalement, on obtient explicitement les champs $\hat{.}$,

$$\left\{\begin{array}{l} \widehat{F}_{E'E} = -\widehat{F}_{EE'} = \frac{1}{2}\left[\left(F_{E'E} - F_{EE'}\right) + d^{+}\left(V_{E'E} - V_{EE'}\right)\right] \\ \widehat{V}_{E'E} = \widehat{V}_{EE'} = \frac{1}{2}\left[\left(V_{E'E} + V_{EE'}\right) + \frac{1}{d^{+}}^{-1}\left(F_{E'E} + F_{EE'}\right)\right] \end{array}\right. \tag{3.24}$$

Par contact. Chaque contact doit vérifier la relation de comportement (2.33),

$$\left\{\begin{array}{lll} \text{Si} & u_n^p > 0 & \widehat{r}_n = 0 \\ \text{Si} & u_n^p \leq 0 & 0 \leq \widehat{v}_n \perp \widehat{r}_n \geq 0 \end{array}\right. \ . \tag{3.25}$$

3.2.2 Étape linéaire.

A partir de l'estimation de la solution $\widehat{s}$ obtenue à l'étape précédente, l'étape linéaire consiste à trouver s qui satisfait la dynamique, la compatibilité des vitesses par sous structures et l'admissibilité macro (2.32),(2.30) et (3.17).

Description Macro et direction de recherche. L'introduction de notations supplémentaires est nécessaire :

- l'indice $(E\Gamma)$ se rapporte toujours à l'ensemble des quantités (EE') liées à une même sous structure E.
- pour manipuler l'admissibilité des impulsions macro, on utilise un champ macro f_Γ défini sur l'interface globale Γ. $f_{E\Gamma}$ peut donc s'écrire $f_{E\Gamma} = B_E f_\Gamma$ où B_E est une matrice booléenne signée.
- La quantité duale de f_Γ est $w_\Gamma = \sum_{E=1}^{n_{sd}} B_E^T w_{E\Gamma}$, celle-ci représente le saut des quantités cinématiques macros généralisées sur les interfaces. L'admissibilité macro impose que $w_\Gamma = 0$.

Formulation de l'étape linéaire.

La direction de recherche micro s'écrit,

$$\left\{\begin{array}{l} \left(r_E - \widehat{r}_E\right) + l^{-}\left(v_E - \widehat{v}_E\right) = 0 \\ \left(F_{E\Gamma}^m - \widehat{F}_{E\Gamma}^m\right) - d^{-}\left(V_{E\Gamma}^m - \widehat{V}_{E\Gamma}^m\right) = 0 \end{array}\right. \ . \tag{3.26}$$

soit, en utilisant la compatibilité des vitesses,

$$\begin{cases} r_E = \widehat{r}_E + l^- \widehat{v}_E - l^- H_E^T V_E \\ F_{E\Gamma}^m = \left(\widehat{F}_{E\Gamma}^m - d^- \widehat{V}_{E\Gamma}^m\right) + d^- \left(V_{E\Gamma} - P_{E\Gamma} w_{E\Gamma}\right) \end{cases} . \tag{3.27}$$

car $V_{E\Gamma}^m = V_{E\Gamma} - V_{E\Gamma}^M = V_{E\Gamma} - P_{E\Gamma} w_{E\Gamma}$.
L'équation de la dynamique s'écrit,

$$M_E V_E = R_E - F_E + R_E^d. \tag{3.28}$$

Avec la direction de recherche,

$$M_E V_E - H_E \left[\widehat{r}_E + l^- \widehat{v}_E - l^- H_E^T V_E\right] + F_E^m + F_E^M = R_E^d \tag{3.29}$$

soit,

$$\begin{gathered} M_E V_E - H_E \left[\widehat{r}_E + l^- \widehat{v}_E - l^- H_E^T V_E\right] \\ + C_{E\Gamma}^T \left[\left(\widehat{F}_{E\Gamma}^m - d^- \widehat{V}_{E\Gamma}^m\right) + d^- \left(V_{E\Gamma} - P_{E\Gamma} w_{E\Gamma}\right)\right] + C_{E\Gamma}^T P_{E\Gamma} f_{E\Gamma} \\ = R_E^d \end{gathered} \tag{3.30}$$

soit,

$$\begin{gathered} \left[M_E + H_E l^- H_E^T + C_{E\Gamma}^T d^- C_{E\Gamma}\right] V_E + C_{E\Gamma}^T d^- P_{E\Gamma} \left(d^{-^{-1}} f_{E\Gamma} - w_{E\Gamma}\right) = \\ R_E^d + H_E \left(\widehat{r}_E + l^- \widehat{v}_E\right) - C_{E\Gamma}^T \left(\widehat{F}_{E\Gamma}^m - d^- \widehat{V}_{E\Gamma}^m\right) \end{gathered} \tag{3.31}$$

soit enfin,

$$\widetilde{M_E} V_E - \widetilde{Q_E} w_{E\Gamma}{}^* = \widehat{R}_E^d \tag{3.32}$$

où

$$\widetilde{M_E} = M_E + H_E l^- H_E^T + C_{E\Gamma}^T d^- C_{E\Gamma}, \tag{3.33}$$

$$\widetilde{Q_E} = C_{E\Gamma}^T d^- P_{E\Gamma}, \tag{3.34}$$

$$w_{E\Gamma}{}^* = w_{E\Gamma} - \frac{1}{d^-} f_{E\Gamma} \tag{3.35}$$

et

$$\widehat{R}_E^d = R_E^d - C_{E\Gamma}^T \left(\widehat{F}_{E\Gamma}^m - d^- \widehat{V}_{E\Gamma}^m\right) + H_E \left(\widehat{r}_E + l^- \widehat{v}_E\right) \tag{3.36}$$

C'est un problème couplé en (V_E,$w_{E\Gamma}^*$). Les équations d'équilibre s'expriment grâce au couple d'inconnues (V_E,$w_{E\Gamma}$) sous la condition qu'elles soient liées par :

$$\begin{gathered} w_{E\Gamma} = P_{E\Gamma}^T C_{E\Gamma} V_E \\ \Leftrightarrow P_{E\Gamma}^T d^- C_{E\Gamma} V_E = d^- \left(w_{E\Gamma} - \tfrac{1}{d^-} f_{E\Gamma}\right) + f_{E\Gamma} \ . \\ \Leftrightarrow \widetilde{Q}_E^T V_E - d^- \left(w_{E\Gamma} - \tfrac{1}{d^-} f_{E\Gamma}\right) = + f_{E\Gamma} \end{gathered} \tag{3.37}$$

Reformulation de l'étape linéaire.

En combinant (3.32) et (3.37),

$$\begin{bmatrix} \widetilde{M_E} & -\widetilde{Q}_E \\ -\widetilde{Q}_E^T & d^- I_d \end{bmatrix} \begin{bmatrix} V_E \\ w^*_{E\Gamma} \end{bmatrix} = \begin{bmatrix} \widehat{R}^d_E \\ -f_{E\Gamma} \end{bmatrix}. \tag{3.38}$$

Où l'on a trois inconnues : $f_{E\Gamma}, w^*_{E\Gamma}, V_E$. Pour fermer le problème on ajoute l'admissibilité macro,

$$\sum_{E=1}^{n_{sd}} B_E^T w_{E\Gamma} = 0. \tag{3.39}$$

Problème macro homogénéisé.

La partie homogénéisée du comportement de la sous structure E est obtenue numériquement par la condensation du système précédent sur les degrés de liberté grossiers $w^*_{E\Gamma}$,

$$A_E w^*_{E\Gamma} = -f_{E\Gamma} + g_E \tag{3.40}$$

avec

$$A_E = d^- I_{dE} - \widetilde{Q_E}^T \widetilde{M_E}^{-1} \widetilde{Q_E} \tag{3.41}$$

et où

$$g_E = \widetilde{Q_E}^T \widetilde{M_E}^{-1} \widehat{R}^d_E. \tag{3.42}$$

$\widetilde{M_E}$ est inversible, tout comme A_E^T, ce qui donne,

$$w^*_{E\Gamma} = A_E^{-1} \left[-f_{E\Gamma} + g_E \right] \tag{3.43}$$

Le problème macro à résoudre est constitué des équations suivantes,

$$\begin{array}{c} w^*_{E\Gamma} = A_E^{-1} \left[-f_{E\Gamma} + g_E \right] \\ \sum_{E=1}^{n_{sd}} B_E^T w_{E\Gamma} = 0 \\ F_{E\Gamma} = P_{E\Gamma} f_{E\Gamma} = P_{E\Gamma} B_E f_\Gamma \end{array}. \tag{3.44}$$

Il s'écrit finalement aussi,

$$w^*_{E\Gamma} = A_E^{-1} \left[f_{E\Gamma} + g_E \right] = w_{E\Gamma} - (d^-)^{-1} f_{E\Gamma} \tag{3.45}$$

soit,

$$w_{E\Gamma} = \left(-A_E^{-1} + (d^-)^{-1} I_d \right) B_E f_\Gamma + A_E^{-1} g_E \tag{3.46}$$

soit le système,

$$\sum_{E=1}^{n_{sd}} \left[B_E^T \underbrace{\left(-A_E^{-1} + (d^-)^{-1} I_d\right)}_{L_E} B_E \right] f_\Gamma = -\sum_{E=1}^{n_{sd}} B_E^T A_E^{-1} g_E \tag{3.47}$$

Relocalisation.

Une fois le problème macro résolu, $f_{E\Gamma}$ est connu pour chaque sous structure E, et $w_{E\Gamma}$ peut être post traité avec (3.46). En remontant par substitution (3.38), on arrive à l'expression,

$$\widetilde{M}_E V_E = Q_E w_{E\Gamma}^* + \widehat{R}_E^d \tag{3.48}$$

qui est résolue grâce à la méthode de Cholesky. La partie micro additionnelle est $V_{E\Gamma}^m = V_{E\Gamma} - P_{E\Gamma} w_{E\Gamma}$ et $F_{E\Gamma}^m$ est calculé grâce à la direction de recherche (3.26).b. La variation de distance entre les centres des grains est $v_E = H^T V_E$ et r_E est calculé grâce à la direction de recherche (3.26).a.

3.3 Algorithme de type GSNL multiéchelle

A partir du champ macroscopique définis précédemment, on impose un raccord cinématique supplémentaire sur les parties macro des champs d'interfaces, que l'on écrira,

$$\sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} V_E = 0. \tag{3.49}$$

ce qui correspond à la partie continuité des vitesses de l'équation (3.17).

Ce raccord supplémentaire macro (3.49) étant satisfait dès que le raccord standard (2.55) l'est, le nouveau problème est sur-contraint, autrement dit le multiplicateur de Lagrange f_Γ associé au raccord macro est indéterminé. Cette approche est similaire à l'augmentation des algorithmes FETI-DP [24] qui ont permis de retrouver l'extensibilité pour les problèmes élastiques du 2nd ordre 3D. Le multiplicateur $F_\Gamma + P_\Gamma f_\Gamma$ satisfait la même équation que F_Γ (2.56),

$$W_E r_E - v_E - \hat{A}_{E\Gamma}^T (F_\Gamma + P_\Gamma f_\Gamma) = -v_E^d, \quad E = 1, n_{SD} \tag{3.50}$$

On résume donc le problème dans le système d'équations suivant,

$$\left\{\begin{array}{c} \left.\begin{array}{c} W_E r_E - v_E - \hat{A}_{E\Gamma}^T \overbrace{(F_\Gamma + P_\Gamma f_\Gamma)}^{\hat{F}_\Gamma} = -v_E^d \\ \mathcal{R}\left(v_E, r_E\right) = 0 \end{array}\right\} E = 1, n_{SD} \\ \sum_{E=1}^{n_{sd}} A_{E\Gamma} V_E = 0 \\ \sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} V_E = 0 \end{array}\right. \tag{3.51}$$

ou encore,

$$\left\{\begin{array}{c} \left.\begin{array}{c} W_E r_E - v_E - \hat{A}_{E\Gamma}^T \hat{F}_\Gamma = -v_E^d \\ \mathcal{R}\left(v_E, r_E\right) = 0 \end{array}\right\} E = 1, n_{SD} \\ X \hat{F}_\Gamma - \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} R_E = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} R_E^d \\ X^M f_\Gamma - \sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} M_E^{-1} (R_E - A_{E\Gamma}^T F_\Gamma) = \sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} M_E^{-1} R_E^d \end{array}\right. \tag{3.52}$$

avec $X^M = \sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} M_E^{-1} A_{E\Gamma}^T P_\Gamma$. Cet opérateur est maintenant de taille réduite et le système linéaire macro associé peut donc être résolu directement sans décomposition de X^M. Un algorithme de type Gauss-Seidel multiéchelle comporte alors trois étapes. La première est similaire au cas monoéchelle avec la résolution en parallèle par sous domaine et avec la méthode GSNL standard des équations (3.52)a et (3.52)b en fixant les efforts (multiplicateurs) aux interfaces F_Γ ou $\hat{F}_\Gamma$. La deuxième étape consiste à déterminer une valeur provisoire de F_Γ aux interfaces en résolvant (3.52)c, par une itération de Gauss-Seidel linéaire (par exemple). On corrige dans la troisième étape la valeur de ce multiplicateur par la résolution du problème macro d'interface (3.52)d. Le caractère séquentiel et "Gauss-Seidel" est appliqué non seulement localement dans la résolution du problème non régulier dans chaque sous domaine mais également globalement dans les trois étapes où l'on détermine chaque inconnue en utilisant les valeurs actualisées des autres. Plus concrètement, connaissant $\tilde{r}_E$, $\tilde{v}_E$ et $\tilde{F}_\Gamma$ on détermine r_E et v_E, pour $E = 1, n_{SD}$, par

$$\left\{\begin{array}{c} W_E^D r_E - v_E = \hat{A}_{E\Gamma}^T \tilde{F}_\Gamma - v_E^d - (W_E - W_E^D)\tilde{r}_E \\ \mathcal{R}\left(v_E, r_E\right) = 0 \end{array}\right. \tag{3.53}$$

puis on prédit une première estimation de F_Γ par

$$X^D F_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} (R_E + R_E^d) - (X - X^D)\tilde{F}_\Gamma \tag{3.54}$$

enfin on corrige F_Γ par le calcul d'une correction macro f_Γ via

$$X^M f_\Gamma = \sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} M_E^{-1} \{R_E - A_{E\Gamma}^T F_\Gamma + R_E^d\} \tag{3.55}$$

En pratique, pour passer de (3.53) à (3.54), on aura intérêt à actualiser les torseurs cinématiques des grains dans chaque sous domaine via,

$$M_E \bar{V}_E = R_E^d + H_E r_E - A_{E\Gamma}^T \tilde{F}_\Gamma \tag{3.56}$$

puis déterminer $F_\Gamma - \tilde{F}_\Gamma$ via,

$$X^D (F_\Gamma - \tilde{F}_\Gamma) = \sum_E A_{E\Gamma} \bar{V}_E \tag{3.57}$$

La correction macro prend alors également l'expression suivante,

$$X^M f_\Gamma = \sum_{E=1}^{n_{sd}} P_\Gamma^T A_{E\Gamma} \{\bar{V}_E - M_E^{-1} A_{E\Gamma}^T (F_\Gamma - \tilde{F}_\Gamma)\} \tag{3.58}$$

3.4 Faible efficacité de l'échelle macroscopique

Comme avec l'algorithme LATIN en lançant de nouveau les calculs pour les cas tests précédents, du chapitre 2.4.1 on obtient des résultats très similaires en terme de convergence avec l'addition de l'échelle macro, à la fois pour le cas linéaire sur les courbes 2.8 et 2.9 et pour le granulaire sans frottement sur la courbe 2.13, obtenues sans approche multiéchelle. Les courbes avec et sans échelle macro étant quasiment superposées on ne les réaffiche pas ici. Ceci est dû au faible couplage entre interfaces d'un même sous domaine lié à la matrice X qui s'avère diagonale, car construite par projection des matrices de masse M_E diagonales. Le champ macro construit comme une réduction de cette matrice (3.58) ne permet donc pas de transmettre d'information à grande longueur d'onde, c'est à dire d'une sous structure à une autre.

Cependant l'aspect macro pourrait permettre de ne pas effectuer les calculs micro sur certaines sous structures si leur état n'évolue pas énormément, et si la solution fine dans celles-ci n'est pas une quantité d'intérêt de la simulation. Comme mentionné dans l'introduction de ce chapitre, ceci permettrait de réalouer les ressources de calcul. En effet, si un indicateur macro permettait de détecter si une sous structure n'a pas besoin d'être mise à jour, par exemple dans le cas où une partie du domaine à son chargement variant très faiblement vis à vis du reste de la structure, il est possible de ne pas calculer de nouveau cette partie du domaine et de profiter ainsi des moyens de calcul disponibles pour le reste du domaine.

C'est dans cette optique que la partie suivante vise à enrichir le problème, de manière à retrouver un opérateur aux interfaces non diagonal, basé non plus seulement sur une matrice de masse, mais aussi sur l'utilisation combinée du treillis sous jacent au maillage, ayant pour noeuds les grains, et pour éléments les contacts potentiels (voire avérés) et de la notion de direction de recherche de la LATIN, qui en tant que paramètre numérique pourra être utilisé comme "rigidité" de ce maillage fictif.

Chapitre 4

Comment enrichir un algorithme GSNL multidomaine ?

Sommaire

4.1 Analyse de convergence de GSNL multidomaine 89
4.2 Une tentative d'enrichissement 93
4.3 Analyse asymptotique et non régulière 99
4.4 Premières validations numériques 111

La formulation GSNL multidomaine est basée sur une approche de type FETI où les sous domaines sont “collés” par un multiplicateur de Lagrange qui correspond aux forces d’interaction entre les sous domaines. Ce choix est bien en accord avec l’approche NSCD où les impulsions sont les variables privilégiées. La méthode de résolution générique est constituée de deux étapes d’un algorithme de type Gauss-Seidel. Une telle stratégie utilise l’algorithme générique “Gauss Seidel Non Linéaire” (GSNL) associé à la formulation NSCD ; pour quelques résultats de convergence, voir [35].

Les développements suivants ont trois objectifs principaux. Le premier est de proposer un algorithme amélioré combinant à la fois les avantages de la LATIN et de GSNL-DD (GSNL version décomposition de domaine) précédemment comparés au chapitre 2.3.3. L’amélioration est motivée par la nature du problème d’interface dérivé de l’approche FETI. L’opérateur associé s’est révélé être quasiment diagonal de telle façon qu’aucune interaction à longue distance n’est générée et aucune quantité macro à grande longueur d’influence ne peut être extraite de ce problème d’interface (cf chapitre 3.4). Le second objectif est donc d’enrichir le problème d’interface en vue d’accélérer la convergence et de définir un problème macro homogénéisé. Enfin, en troisième objectif, le rôle des mouvements de corps rigides, par sous domaine, sur la convergence de l’algorithme doit être clarifié. En effet, l’insertion de ceux-ci au sein du problème grossier est essentielle pour l’extensibilité de problèmes élastostatiques, ce qui a motivé le développement de la méthode FETI [27].

Ce dernier chapitre sera plus théorique en vue essentiellement de proposer un enrichissement de l’algorithme standard. L’analyse théorique permettra de justifier les choix d’enrichissement que l’on illustrera par des exemples de petite taille. Une première partie permettra de mettre en place la méthodologie et de tester la convergence sur un problème linéaire élastodynamique des algorithmes standards ou enrichis. Suivra alors une analyse asymptotique dédiée à l’étude de ces algorithmes dans un cas non régulier pour une collection de corps rigides avec des interactions de contact unilatéral sans frottement. On finira sur des premières validations numériques de l’enrichissement proposé, en étudiant quels sont les gains obtenus et comment les optimiser.

4.1 Analyse de convergence de GSNL multidomaine

4.1.1 Méthodologie

L'algorithme 1 obtenu précédemment au chapitre 2.3.2 est basé sur les équations suivantes :

– Dynamique :

$$M_E V_E - R_E + A_{E\Gamma}^T F_\Gamma = R_E^d \tag{Dyn}$$

– Admissibilité :

$$\begin{cases} v_E = H_E^T V_E \\ R_E = H_E r_E \end{cases} \tag{Adm}$$

– Comportement :

$$\mathcal{R}\left(v_E, r_E\right) = 0 \tag{Compt}$$

– Continuité :

$$\sum_E A_{E\Gamma} V_E = 0 \tag{Conti}$$

C'est l'agencement de ces équations qui fournit l'algorithme générique des chapitres précédents. On choisit de répartir ces équations entre deux étapes intervenant à chaque itération ; on part d'une solution (r_E, v_E, F_Γ) pour arriver à un itéré suivant $(\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma))$,

– Étape 1 : (Dyn) + (Adm) + (Compt) $\Rightarrow \begin{cases} W_E \bar{r}_E - \bar{v}_E = H_E^T M_E^{-1}(A_{E\Gamma}^T F_\Gamma - R_E^d) \\ \mathcal{R}\left(\bar{v}_E, \bar{r}_E\right) = 0 \end{cases}$

– Étape 2 : (Dyn) + (Conti) + (Adm) $\Rightarrow X\ \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1}(\bar{R}_E + R_E^d)$

Cet agencement donne l'algorithme 1 mis en place au hapitre 2.3.2 et répété ici pour faciliter la lecture.

En éliminant les variables locales $(\bar{r}_E, \bar{v}_E)$ dans la première étape de l'algorithme 1, on peut définir un algorithme de point fixe appliqué au multiplicateur d'interface F_Γ. Pour cela on introduit l'application T_Γ qui permet de passer d'un itéré de F_Γ au suivant $\bar{F}_\Gamma$,

$$T_\Gamma : \quad F_\Gamma \mapsto \bar{F}_\Gamma. \tag{4.1}$$

La deuxième étape permet de mettre à jour les impulsions d'interface $\bar{F}_\Gamma$ à partir des impulsions de contact $\bar{r}_E$ mises à jour dans l'étape 1,

$$X\bar{F}_\Gamma = \hat{f} + \sum_{E=1}^{n_{sd}} \widehat{A}_{E\Gamma} \bar{r}_E. \tag{4.2}$$

Algorithm 1 (Version générique)

for $i = 1, \ldots, N$ **do** {Boucle sur les pas de temps}
  initialisation des inconnues (r_E, v_E, F_Γ) au temps t_i.
  while $\zeta < \zeta_{\max}$ (critère de convergence) **do**
    (r_E, v_E, F_Γ) sont connus, calcul de $(\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma)$:
    Étape 1 : Calcul de $\bar{r}_E, \bar{v}_E$ en parallèle pour $E = 1, \ldots, n_{sd}$
    avec n_{GS} itérations de Gauss-Seidel sur :

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E = -v_E^d + \hat{A}_{E\Gamma}^T F_\Gamma \\ \mathcal{R}(\bar{v}_E, \bar{r}_E) = 0 \end{cases}$$

    Étape 2 : Calcul de $\bar{F}_\Gamma$:

$$X \bar{F}_\Gamma = \hat{f} + \sum_{E=1}^{n_{sd}} \hat{A}_{E\Gamma} \bar{r}_E$$

    for $E = 1, \ldots, n_{sd}$ **do**
      $(r_E, v_E, F_\Gamma) \leftarrow (\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma)$
    end for
  end while
end for

La première étape est écrite sous la forme de l'application $T_{\Gamma E}$ permettant de passer des grandeurs d'interface F_Γ aux impulsions par sous domaine $\bar{r}_E$. Cet opérateur est a priori très non linéaire,

$$T_{\Gamma E} : \quad F_\Gamma \mapsto \bar{r}_E. \tag{4.3}$$

L'opérateur T_Γ s'écrit alors,

$$T_\Gamma(F_\Gamma) = \bar{F}_\Gamma = X^{-1}(\hat{f} + \sum_{E=1}^{n_{sd}} \widehat{A}_{E\Gamma} T_{\Gamma E}(F_\Gamma)) \tag{4.4}$$

et T_Γ est contractante si $X^{-1} \sum_{E=1}^{n_{sd}} \widehat{A}_{E\Gamma} T_{\Gamma E}(F_\Gamma)$ l'est. Du fait de la non linéarité, il n'est pas évident d'étudier les caractéristiques de cet opérateur dans un cas général. Pour étudier le caractère contractant de cette application, on revient au cas linéaire décrit au chapitre 2.4.1. Pour la suite il est pratique de mettre en place un paramètre linéaire ϵ correspondant à l'inverse du paramètre linéaire k_E, supposé identique pour toutes les interactions. La relation de comportement s'écrit alors,

$$\mathcal{R}(\bar{v}_E, \bar{r}_E) = 0 \Leftrightarrow \bar{r}_E = -k_E \; \bar{v}_E = -\frac{1}{\epsilon} \; \bar{v}_E. \tag{4.5}$$

Ce qui permet de réécrire la première étape sous la forme d'unique inconnue $\bar{r}_E$ (I_d, matrice identité),

$$(W_E + \epsilon I_d) \, \bar{r}_E = -v_E^d + \widehat{A}_{E\Gamma}^T F_\Gamma. \tag{4.6}$$

L'opérateur $T_{\Gamma E}$ peut alors être mis en évidence,

$$\bar{r}_E = (W_E + \epsilon I_d)^{-1} \left(-v_E^d + \widehat{A}_{E\Gamma}^T F_\Gamma \right) \tag{4.7}$$

et en posant $W_{E,\epsilon} = (W_E + \epsilon Id)$ on obtient,

$$T_{\Gamma E}(F_\Gamma) = W_{E,\epsilon}^{-1} \widehat{A}_{E\Gamma}^T F_\Gamma + \hat{f}_E^d, \tag{4.8}$$

où $\hat{f}_E^d = W_{E,\epsilon}^{-1} \left(-v_E^d \right)$. L'application T_Γ caractéristique de l'application de point fixe s'écrit alors,

$$T_\Gamma(F_\Gamma) = X^{-1} \left(\hat{f} + \sum_{E=1}^{n_{sd}} \widehat{A}_{E\Gamma} \left(W_{E,\epsilon}^{-1} \widehat{A}_{E\Gamma}^T F_\Gamma + \hat{f}_E^d \right) \right). \tag{4.9}$$

En posant $\hat{f}^d = X^{-1} \left(\hat{f} + \sum_{E=1}^{n_{sd}} \widehat{A}_{E\Gamma} \hat{f}_E^d \right)$ on obtient,

$$T_\Gamma(F_\Gamma) = X^{-1} Y_\epsilon F_\Gamma + \hat{f}^d, \tag{4.10}$$

avec

$$Y_\epsilon = \sum_{E=1}^{n_{sd}} \widehat{A}_{E\Gamma} W_{E,\epsilon}^{-1} \widehat{A}_{E\Gamma}^T, \tag{4.11}$$

ou en écriture moins condensée,

$$Y_\epsilon = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_E^{-1} H_E (W_E + \epsilon I_d)^{-1} H_E^T M_E^{-1} A_{E\Gamma}^T. \tag{4.12}$$

La matrice de contraction B_ϵ dépend alors du paramètre de flexibilité ϵ,

$$B_\epsilon = X^{-1} Y_\epsilon, \tag{4.13}$$

et l'application est contractante si le rayon spectral de B_ϵ est inférieur à 1,

$$\rho(B_\epsilon) < 1. \tag{4.14}$$

4.1.2 Analyse de convergence dans le cas linéaire

Une estimation théorique du rayon spectral de cette matrice n'est pas aisément accessible, mais une évaluation numérique peut être réalisée pour des exemples simples, en vue de comparer les performances de cet algorithme avec celui à venir. Le premier exemple simple consiste en un essai 2D sans frottement de problème granulaire avec 4 disques et potentiellement 5 contacts, ces derniers représentés par des lignes continues sur la Figure 4.1. Une solution est dessinée sur la même figure pour une vitesse imposée sur un des disques ; certaines tensions sont positives ou nulles (en gris), car la perte de contact n'est pas prise en compte dans cet exemple. Ce système est divisé en deux sous domaines, chacun contenant un disque et deux demi-disques. Deux exemples plus gros sont représentés dans les Figures 4.2 et 4.3. Le rayon spectral de la matrice de contraction est donné en fonction de ϵ sur la Figure 4.4 ; deux valeurs particulières sont distinguées : $\epsilon = 1$ et le cas limite $\epsilon = 0$.

Dans le cas $\epsilon = 0$ le rayon spectral est à son maximum. On est dans un cas limite rigide, donc la convergence est plus délicate ; toutefois les valeurs obtenues restent inférieures à 1 : l'application reste contractante. Les valeurs obtenues pour $\epsilon = 1$ sont indiquées car elles sont une référence pour la suite. Les masses des grains étant unitaires ici, ce cas correspond au choix fait pour le paramètre régularisant dans le chapitre 2.2.2. Plus ϵ est grand plus le problème est régularisé, le rayon spectral diminue en conséquence. Enfin plus on a de disques, plus les liens sont nombreux, et moins l'influence de la partie régularisante est grande.

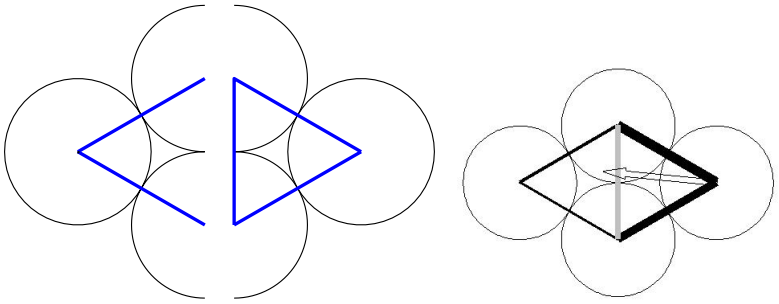


FIG. 4.1 – Un premier exemple avec 4 disques et 5 liens découpés en deux sous domaines et un réseau de tensions résultant (noir pour la compression et gris pour la tension).

4.2 Une tentative d'enrichissement

4.2.1 Algorithme enrichi

Pour un système granulaire dense, les équations de la dynamique seules ne suffisent pas pour reproduire le comportement collectif (global) de l'exemple. L'opérateur global précédent X ne mettait en jeu que la dynamique via les matrices d'inertie M_E ; cet opérateur est diagonal et ne peut ainsi fournir aucun transfert d'information entre les grains à une longue distance. L'introduction d'un comportement collectif doit donc aussi prendre en compte les interactions entre grains. Pour cela, nous proposons d'introduire une direction de recherche additionnelle dans l'algorithme précédent 1 qui, en gardant le problème de référence inchangé, peut s'interpréter comme un comportement numérique tangent entre les grains. Cela introduit alors un paramètre additionnel dont il faudra étudier la détermination optimale. Un comportement tangent est donc ajouté aux équations utilisées pour l'algorithme standard du chapitre 4.1.1 sous la forme,

$$(\widetilde{r}_E - \bar{r}_E) + l_E(\widetilde{v}_E - \bar{v}_E) = 0. \tag{Dir}$$

Le seul changement dans la répartition des équations consiste à ajouter cette direction de recherche à l'étape 2 :

- Étape 1 : (Dyn) + (Adm) + (Compt)
- Étape 2 : (Dyn) + (Conti) + (Adm) + (Dir)

L'algorithme précédent est donc modifié comme suit. La première étape consiste toujours à trouver indépendamment sur chaque sous domaine $E = 1, \ldots, n_{sd}$, les

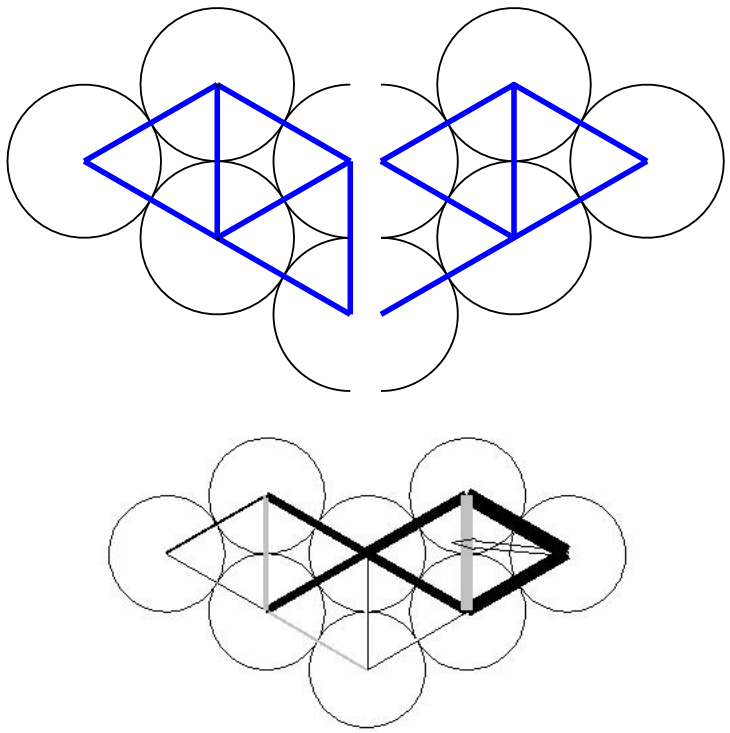


FIG. 4.2 – Un second exemple avec 8 disques et 13 liens découpés en deux sous domaines et un réseau de tensions résultant (noir pour la compression et gris pour la tension).

quantités $(\bar{v}_E, \bar{r}_E)$ satisfaisant,

$$\left\{ \begin{array}{rcl} W_E \bar{r}_E - \bar{v}_E & = & \hat{A}^T_{E\Gamma} F_\Gamma - v^d_E \\ \mathcal{R}(\bar{v}_E, \bar{r}_E) & = & 0 \end{array} \right. \tag{4.15}$$

La deuxième étape est modifiée. Elle nécessite de trouver $(\tilde{v}_E, \tilde{r}_E)$ et $\bar{F}_\Gamma$, une fois que $(\bar{v}_E, \bar{r}_E)$ sont connus de l'étape précédente, satisfaisant :

- les équations de la dynamique (Dyn),
- l'admissibilité (Adm),
- la contrainte de continuité à l'interface (Conti),
- la nouvelle direction de recherche $(\tilde{r}_E - \bar{r}_E) + l_E(\tilde{v}_E - \bar{v}_E) = 0$ (Dir).

l_E est un paramètre numérique additionnel dans l'algorithme. Comme pour k_E, il est commode pour l'analyse à venir d'introduire un paramètre identique pour toutes les interactions, $\eta = \frac{1}{l_E}$.

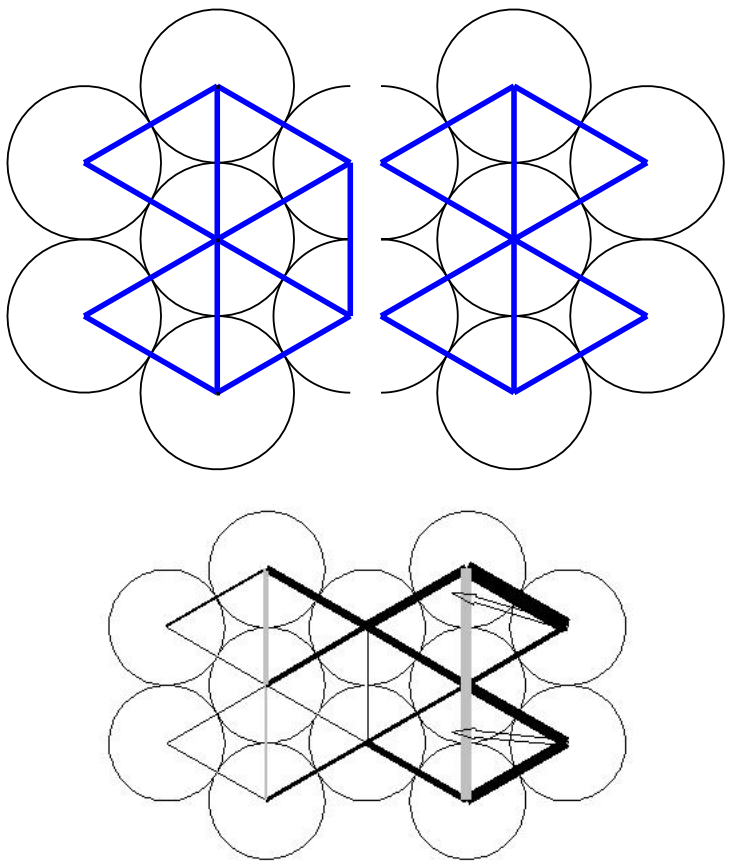


FIG. 4.3 – Un troisième exemple avec 12 grains et 21 liens découpés en deux sous domaines et un réseau de tensions résultant (noir pour la compression et gris pour la tension).

En introduisant maintenant le champ de vitesse des grains $\tilde{V}_E$, la dynamique s'écrit,

$$M_E\tilde{V}_E + A_{E\Gamma}^T\bar{F}_\Gamma = R_E^d + H_E\tilde{r}_E. \tag{4.16}$$

Substituer $\tilde{r}_E$ avec la nouvelle direction de recherche, en utilisant l'admissibilité cinématique $\tilde{v}_E = H_E^T\tilde{V}_E$, donne

$$M_{\eta,E}\tilde{V}_E = R_E^d + H_E(\bar{r}_E + \frac{1}{\eta}\bar{v}_E) - A_{E\Gamma}^T\bar{F}_\Gamma \tag{4.17}$$

avec $M_{\eta,E} = M_E + \frac{1}{\eta}K_E$ et $K_E = H_E H_E^T$ qui contiennent des informations sur le réseau des interactions grâce à la matrice de connectivité H_E. Comme précédemment,

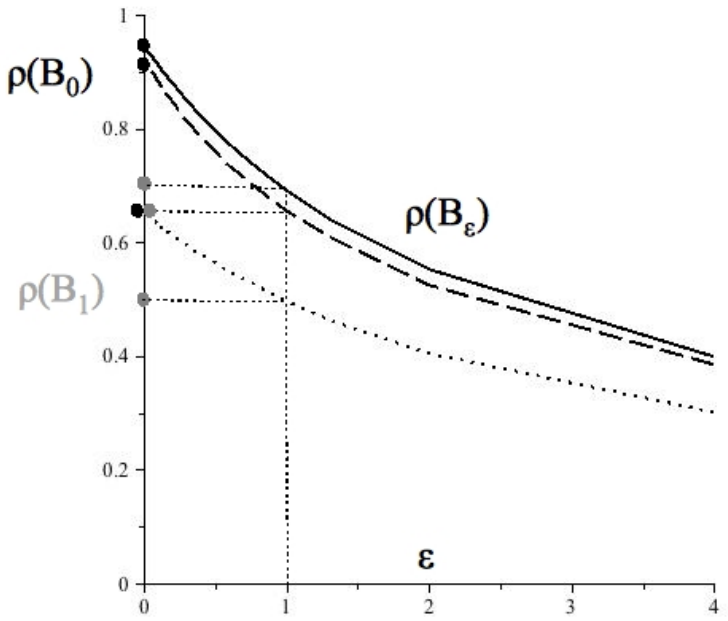


FIG. 4.4 – Évolution du rayon spectral de l'algorithme générique en fonction de ϵ pour les 3 exemples : 4 disques (point), 8 disques (tiret), 12 disques (continus).

les équations de la dynamique sont condensées sur les impulsions d'interface, et la condition de continuité à l'interface donne le nouveau comportement global avec $\bar{F}_\Gamma$ comme inconnue,

$$X_\eta \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{\eta,E}^{-1} [R_E^d + H_E(\bar{r}_E + \frac{1}{\eta}\bar{v}_E)], \tag{4.18}$$

avec le nouvel opérateur global sur les impulsions d'interface,

$$X_\eta = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{\eta,E}^{-1} A_{E\Gamma}^T. \tag{4.19}$$

Ceci constitue la deuxième étape modifiée de l'algorithme. L'algorithme enrichi est écrit dans Algorithme 2.

4.2.2 Analyse de convergence dans le cas linéaire

Dans un tel cas, la première étape se simplifie en : trouver $\bar{r}_E$, pour chaque sous domaine indépendamment, vérifiant (4.6). Pour la seconde étape, le membre de

Algorithm 2 (Version enrichie)

for $i = 1, \ldots, N$ **do** {Boucle sur les pas de temps}
  initialisation des inconnues (r_E, v_E, F_Γ) au temps t_i
  while $\zeta < \zeta_{\max}$ (critère de convergence) **do**
    (r_E, v_E, F_Γ) sont connus, calcul de $(\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma)$.
    Étape 1 : Calcul de $\bar{r}_E, \bar{v}_E$ en parallèle pour $E = 1, \ldots, n_{sd}$
    avec n_{GS} itérations de Gauss-Seidel sur :

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E \;=\; H_E^T M_E^{-1} (A_{E\Gamma}^T F_\Gamma - R_E^d) \\ \mathcal{R}(\bar{v}_E, \bar{r}_E) = 0 \end{cases}$$

    Étape 2 : Calcul de $\bar{F}_\Gamma$:

$$X_\eta \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{\eta,E}^{-1} [R_E^d + H_E(\bar{r}_E + \frac{1}{\eta}\bar{v}_E)]$$

    for $E = 1, \ldots, n_{sd}$ **do**
      $(r_E, v_E, F_\Gamma) \leftarrow (\bar{r}_E, \bar{v}_E, \bar{F}_\Gamma)$
    end for
  end while
end for

droite se simplifie,

$$\bar{r}_E + l_E \bar{v}_E = (1 - \frac{\epsilon}{\eta})\bar{r}_E. \tag{4.20}$$

Comme pour la méthode générique appliquée dans 4.1.1 au cas linéaire, les variables locales peuvent être éliminées de manière à obtenir un algorithme de point fixe sur l'inconnue F_Γ. L'expression de T_Γ correspondante à (4.4) s'écrit, à partir de l'étape 2, pour cet algorithme,

$$T_\Gamma(F_\Gamma) = \bar{F}_\Gamma = X_\eta^{-1} \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} \left[R_E^d + H_E \left(1 - \frac{\epsilon}{\eta}\right) T_{\Gamma E}\left(F_\Gamma\right) \right] \tag{4.21}$$

et puisque l'étape 1 n'est pas modifiée on conserve l'expression de $T_{\Gamma E}$ de (4.8).

On obtient la matrice de contraction qui est assez complexe et dépend à la fois du paramètre physique de flexibilité ϵ et du paramètre additionnel numérique η :

$$B_{\epsilon,\eta} = X_\eta^{-1} Y_{\epsilon,\eta} \tag{4.22}$$

où

$$\begin{array}{rcl} X_\eta & = & \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} A_{E\Gamma}^T \\ Y_{\epsilon,\eta} & = & \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} H_E \left(1 - \frac{\epsilon}{\eta}\right) W_{E,\epsilon}^{-1} \hat{A}_{E\Gamma}^T \\ & = & \left(1 - \frac{\epsilon}{\eta}\right) \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} H_E W_{E,\epsilon}^{-1} H_E^T M_E^{-1} A_{E\Gamma}^T \\ M_{E,\eta} & = & M_E + \frac{1}{\eta} K_E \\ K_E & = & H_E H_E^T \end{array} \tag{4.23}$$

Quand $\eta = \epsilon$ la matrice $Y_{\epsilon,\eta}$ est évidemment nulle et le rayon spectral de $B_{\epsilon,\eta}$ est nul aussi. Ceci signifie qu'il converge en une seule itération, parce que les deux étapes sont totalement découplées : la deuxième étape est le problème condensé sur l'interface du problème sous-structuré. Ce résultat ne fait que souligner la consistance et l'enrichissement fort de cette seconde version de la méthode pour un paramètre ϵ non nul. Le rayon spectral peut être évalué sur les exemples pour chaque valeur de η, ϵ étant fixé à 1 ou 0 sur la Figure 4.5. Cependant pour le cas asymptotique, $\epsilon = 0$, la valeur optimale zéro du rayon spectral ne semble pas être atteinte, spécialement quand la taille du système augmente de l'exemple 1 à l'exemple 3. Les traits horizontaux de la Figure 4.5 correspondent à la version générique, celle-ci ne variant pas en fonction de η, et dont les valeurs sont mises en évidence Figure 4.4.

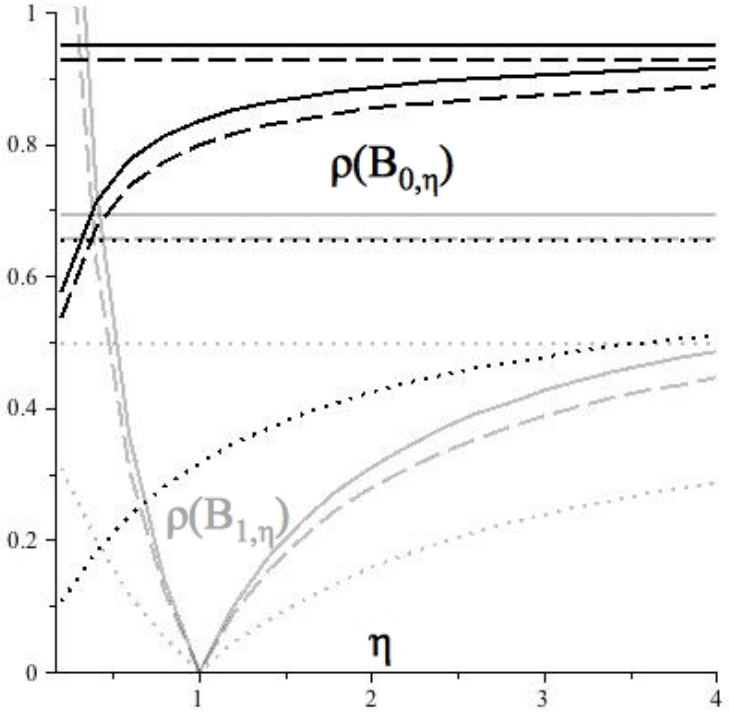


FIG. 4.5 – Rayon spectral de l'algorithme enrichi en fonction de η, pour $\epsilon = 1$ et pour $\epsilon = 0$ et les trois exemples : 4 disques (point), 8 disques (tiret), 12 disques (continus).

4.3 Analyse asymptotique et non régulière

Une telle analyse concerne deux cas limites. Le second, le cas non régulier, consiste à déterminer les impulsions de contact et les vitesses induites à un temps donné dans un granulat dense impacté par des particules. On se limite ici à l'étude du contact sans frottement (ce cas est appelé non régulier dans la suite).

Le premier cas n'est pas relié à un cas physique réaliste, mais est un problème intermédiaire vu comme le cas limite du problème linéaire régulier où le paramètre de flexibilité ϵ tend vers zéro. On a alors à déterminer dans ce problème linéaire une distribution d'impulsions entre corps rigides sans perte de contact possible. Ceci est similaire à la détermination de tensions dans un réseau hyperstatique de barres rigides.

4.3.1 Cas limite rigide avec GSNL standard

Quand ϵ tend vers zéro, la matrice de contraction limite est alors égale à :

$$B_0 = X^{-1}Y_0 \qquad \text{où} \qquad Y_0 = \sum_{E=1}^{n_{sd}} \hat{A}_{E\Gamma} W_E^{-1} \hat{A}_{E\Gamma}^T. \tag{4.24}$$

Il n'est pas surprenant que le rayon spectral augmente quand le paramètre ϵ tend vers zéro, comme montré sur la Figure 4.4. De plus cette matrice ne peut pas toujours être évaluée si l'opérateur de Delassus W_E est non inversible, c'est à dire quand le réseau des contacts est hyperstatique. C'est une situation commune dans les granulats denses, générant la non unicité de la distribution des impulsions de contact. Cependant une telle difficulté n'est pas rencontrée dans les exemples traités ici.

4.3.2 Cas limite rigide avec GSNL enrichi

Si l'on trace le rayon spectral de la matrice de contraction $\rho(B_{0,\eta})$ en fonction de η, comme sur la Figure 4.5, on obtient une valeur décroissante du rayon spectral quand η tend vers zéro, mais un rayon spectral nul ne semble pas pouvoir être atteint. Pour vérifier directement la cohérence du problème limite, $\eta = 0$ pour $\epsilon = 0$, nous avons à réaliser une approximation de la matrice $M_{E,\eta}^{-1}$ quand η tend vers zéro. Dans l'algorithme enrichi (4.17) la matrice $M_{E,\eta}$ s'écrit,

$$M_{E,\eta} = \left(M_E + \frac{1}{\eta} K_E \right). \tag{4.25}$$

M_E est symétrique et définie positive et même diagonale, on peut donc extraire sa racine carré,

$$M_E^{1/2} \quad \text{tel que} \quad M_E = M_E^{1/2} M_E^{1/2}. \tag{4.26}$$

On pourrait à partir d'ici utiliser le théorème de diagonalisation simultanée de M_E et K_E, mais on va plutôt détailler la démarche pour faire apparaître la pseudo-inverse de Moore-Penrose. Soit donc $K_{EM} = M_E^{-1/2} K_E M_E^{-1/2}$ symétrique et définie positive. Par le théorème de diagonalisation il existe Q_E orthogonale ($Q_E^T Q_E = I_E$) telle que,

$$K_{EM} = Q_E^T D_E Q_E \quad \text{avec} \quad d_{ii} \geq 0, \tag{4.27}$$

où D_E est une matrice diagonale, dont les entrées d_{ii} sont positives.

On a alors,

$$K_E = \underbrace{M_E^{1/2} Q_E^T}_{O_E^T} D_E \underbrace{Q_E M_E^{1/2}}_{O_E}, \tag{4.28}$$

$$O_E^T O_E = M_E^{1/2} Q_E^T Q_E M_E^{1/2} = M_E. \tag{4.29}$$

On a donc l'expression suivante de $M_{E,\eta}$,

$$M_{E,\eta} = O_E^T \left(\underbrace{I_E + \frac{1}{\eta} D_E}_{\widetilde{D}_E} \right) O_E \tag{4.30}$$

où $\widetilde{D}_E$ est inversible car $\widetilde{d}_{ii} = 1 + \frac{d_{ii}}{\eta} > 0$ (avec $\widetilde{d}_{ii}$ les éléments de la matrice $\widetilde{D}_E$, l'indice E étant omis pour simplifier l'écriture) et

$$M_{E,\eta}^{-1} = O_E^{-1} \widetilde{D}_E^{-1} O_E^{-T}. \tag{4.31}$$

Détaillons $\widetilde{D}_E^{-1}$ et son développement au premier ordre,

$$\widetilde{d}_{ii}^{-1} = \frac{\eta}{\eta + d_{ii}} \sim \eta \frac{1}{d_{ii}} \quad \text{si} \quad d_{ii} > 0 \quad \text{et} \quad \eta \quad \text{petit}. \tag{4.32}$$

Définissons alors D_E^∞ et D_E^+ de la manière suivante :

$$d_{ii}^\infty = \begin{cases} 0 & \text{si} \quad d_{ii} \neq 0 \\ 1 & \text{si} \quad d_{ii} = 0 \end{cases} \quad \text{et} \quad d_{ii}^+ = \begin{cases} \frac{1}{d_{ii}} & \text{si} \quad d_{ii} \neq 0 \\ 0 & \text{si} \quad d_{ii} = 0 \end{cases}. \tag{4.33}$$

Il est clair alors que,

$$\widetilde{D}_E^{-1} \sim D_E^\infty + \eta D_E^+ \tag{4.34}$$

et, en posant $M_E^\infty = O_E^{-1} D_E^\infty O_E^{-T}$ et $K_E^+ = O_E^{-1} D_E^+ O_E^{-T}$,

$$M_{E,\eta}^{-1} \sim M_E^\infty + \eta K_E^+. \tag{4.35}$$

Pour faire apparaître la pseudo-inverse de Moore-Penrose on passe par l'étude des propriétés de M_E^∞ et K_E^+.

Propriété 1 : $\forall x$ vecteur, $M_E^\infty x$ n'est non nul que sur le noyau de K_E. Or H_E engendre $Im(K_E)$. Comme $\mathrm{Ker}(K_E) \perp \mathrm{Im}(K_E)$ on a donc,

$$M_E^\infty H_E = 0 \tag{4.36}$$

Propriété 2 : K_E^+ peut être interprété comme la pseudo-inverse de Moore-Penrose modulo M_E. En effet la pseudo-inverse de Moore-Penrose, indicée MP, s'écrit normalement,

$$K_E^{MP} = U_E^T \Lambda_E^+ U_E \tag{4.37}$$

où $K_E = U_E^T \Lambda_E U_E$ avec $U_E^T U_E = I_E$ et $\Lambda_{ii}^+ = \begin{cases} \frac{1}{\Lambda_{ii}} & \text{si} \quad \Lambda_{ii} \neq 0 \\ 0 & \text{si} \quad \Lambda_{ii} = 0 \end{cases}$

On a ici,

$$K_E = O_E^T D_E O_E = M_E^{1/2} Q_E^T D_E Q_E M_E^{1/2} \tag{4.38}$$

et

$$K_E^+ = O_E^{-1} D_E^+ O_E^{-T} = M_E^{-1/2} Q_E^T D_E^+ Q_E M_E^{-1/2}. \tag{4.39}$$

En posant $K_{EM} = M_E^{-1/2} K_E M_E^{-1/2}$, de (4.38) on tire,

$$K_{EM} = Q_E^T D_E Q_E \tag{4.40}$$

et de (4.39),

$$K_E^+ = M_E^{-1/2} K_{EM}^{MP} M_E^{-1/2}. \tag{4.41}$$

Limite du second membre de l'étape 2. l' approximation au premier ordre du second membre de l'étape 2 s'écrit,

$$\sum_{E=1}^{n_{sd}} A_{E\Gamma} \left[\left(M_E^\infty + \eta K_E^+ \right) F_E^d + \left(M_E^\infty + \eta K_E^+ \right) H_E (\bar{r}_E + \frac{1}{\eta} \bar{v}_E) \right], \tag{4.42}$$

et avec (4.36) il reste,

$$\sum_{E=1}^{n_{sd}} A_{E\Gamma} \left[\left(M_E^\infty + \eta K_E^+ \right) F_E^d + \eta K_E^+ H_E \bar{r}_E + K_E^+ H_E \bar{v}_E \right]. \tag{4.43}$$

Sa limite quand η tend vers zéro a alors une expression simple et l'étape 2 s'écrit alors,

$$XF_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} \left[M_E^\infty F_E^d + K_E^+ H_E \bar{v}_E \right]. \tag{4.44}$$

Convergence de l'algorithme pour le problème limite rigide. Dans le cas limite rigide $\epsilon = 0$, l'étape 1 s'écrit,

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E = \hat{A}^T_{E\Gamma} F_\Gamma - v^d_E \\ \bar{v}_E = 0 \end{cases}$$

soit $W_E \bar{r}_E = \hat{A}^T_{E\Gamma} F_\Gamma - v^d_E$ de seule inconnue $\bar{r}_E$. On peut réécrire la deuxième étape de l'algorithme enrichi quand η tend vers zéro, et quelque soit la loi locale de comportement (même pour des lois non linéaires ou non régulières) :

$$X^\infty \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma}[M^\infty_E R^d_E + K^+_E H_E \bar{v}_E] \qquad \text{avec} \qquad X^\infty = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M^\infty_E A^T_{E\Gamma} \quad (4.45)$$

Cette étape est découplée de l'étape 1. On retrouve le cas limite $\epsilon = \eta$ du cas régulier du chapitre 4.2.2. Le cas limite rigide linéaire est caractérisé par un champ de variables cinématiques $\bar{v}_E$ (vitesses relatives entre grains pour un système granulaire) égal à zéro, de sorte que les étapes de l'algorithme soient découplées, et la convergence nécessite alors une seule itération comme pour le cas linéaire régulier. Il est intéressant de noter que la matrice M^∞_E extrait les mouvements de corps rigide du sous domaine E, souvent utilisés comme pré-conditionneurs dans les problèmes élastostatiques et même élastodynamiques de milieux continus.

4.3.3 Cas non régulier avec GSNL standard

Un premier essai pour analyser la convergence d'un cas non régulier consiste à considérer une condition de complémentarité entre r_E et v_E (caractéristique du contact unilatéral sans frottement entre les grains d'un granulat). Une telle condition peut être interprétée comme une relation linéaire par morceaux. La convergence globale est difficile à analyser [1, 35] mais on peut étudier la convergence locale telle que réalisée par exemple dans [12].

La solution d'un problème de complémentarité est caractérisée mathématiquement par le domaine de linéarité à laquelle elle appartient, et mécaniquement, si on se réfère à un problème de contact, à la liste des contacts réellement actifs à convergence. On introduit alors l'opérateur de compatibilité, noté ici G_E, restriction de H_E à cette liste de contacts.

On illustre la différence entre G_E et H_E sur les trois exemples précédents des Figures 4.1, 4.2 et 4.3. Parmi l'ensemble des contacts de H_E, on distingue ceux de G_E en trait continu, ceux restant (différence entre H_E et G_E) étant représentés en pointillés sur les Figures 4.6, 4.7 et 4.8. A cette restriction des contacts réellement

actifs sont associées les trois nouvelles solutions correspondantes sur ces mêmes figures.

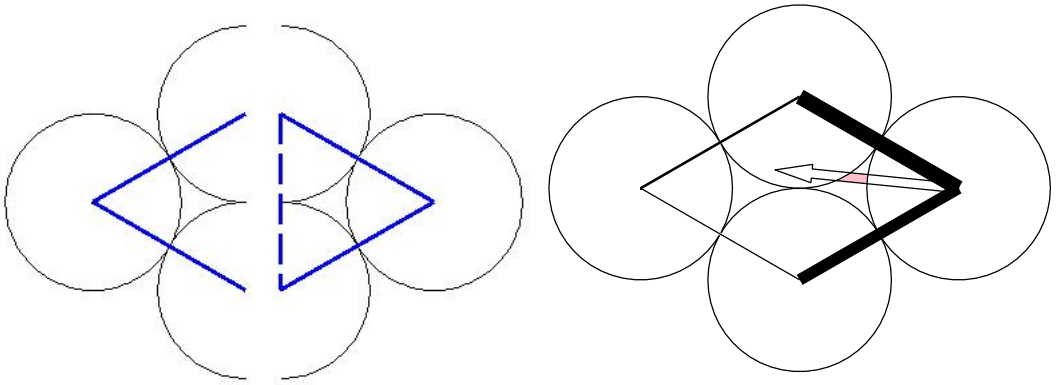


FIG. 4.6 – Le premier exemple avec 4 disques et 5 liens découpés en deux sous domaines et un réseau de compression résultant des seuls liens actifs (traits continus).

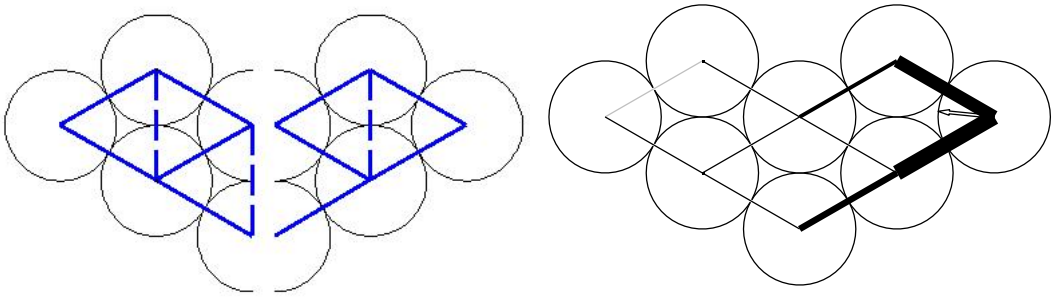


FIG. 4.7 – Le second exemple avec 8 disques et 13 liens découpés en deux sous domaines et un réseau de compression résultant des seuls liens actifs (traits continus).

La solution respecte donc la “loi de comportement”,

$$\bar{v}_E = G_E V_E = 0 \tag{4.46}$$

qui stipule que tous les contacts de G_E sont actifs. On retrouve donc le cas limite rigide associé à l'opérateur G_E. Les impulsions sur les grains sont reliées aux seules impulsions de contact actives par G_E et à toutes les impulsions de contact potentiel par H_E,

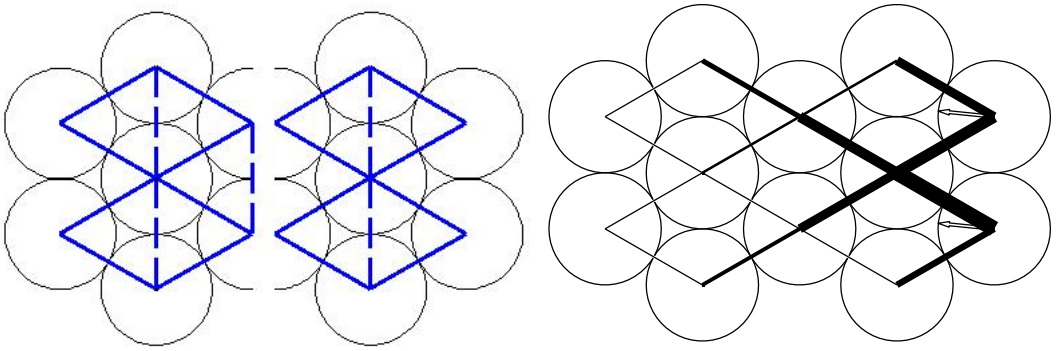


FIG. 4.8 – Le troisième exemple avec 12 grains et 21 liens découpés en deux sous domaines et un réseau de compression résultant des seuls liens actifs (traits continus).

$$R_E = H_E r_E = G_E \bar{r}_E \tag{4.47}$$

alors, $r_E = C_E \bar{r}_E$ et $H_E C_E = G_E$.

Lors des dernières itérations de l'Algorithme 2, les contacts actifs ayant été déterminé, la première étape de l'algorithme générique 1 peut être écrite comme suit : G_E remplace H_E dans W_E, $\hat{A}_{E\Gamma}$ et $\hat{f}_\Gamma$ pour définir $\bar{W}_E$, $\bar{A}_{E\Gamma}$ et $\bar{f}_\Gamma$ ($\bar{A}^T_{E\Gamma} = G^T_E M^{-1}_E A^T_{E\Gamma}$, $\bar{W}_E = G^T_E M^{-1}_E G_E$, on rappelle que dans ce cas $\bar{v}_E$ est égal à zéro). On ne résout le problème local que sur les contacts actifs,

$$\text{Étape 1 :} \quad \bar{W}_E \bar{r}_E = -\bar{v}^d_E + \bar{A}^T_{E\Gamma} \bar{F}_\Gamma \tag{4.48}$$

La deuxième étape est aussi réécrite. On prend les $\bar{r}_E$ calculés sur les contacts actifs et 0 sur les autres de manière à travailler avec des vecteurs de même taille qu'auparavant en prenant tous les contacts en compte, de sorte que $\bar{r}_E$ est remplacé par $C_E \bar{r}_E$,

$$\text{Étape 2 :} \quad X \bar{F}_\Gamma = \hat{f} + \sum_{E=1}^{n_{sd}} \hat{A}_{E\Gamma} C_E \bar{r}_E = \hat{f} + \sum_{E=1}^{n_{sd}} \bar{A}_{E\Gamma} \bar{r}_E. \tag{4.49}$$

La matrice de contraction limite correspond à celle associée au cas limite linéaire rigide, utilisant G_E à la place de H_E. Pour les trois exemples, le rayon spectral est représenté sur la Figure 4.9 avec une ligne horizontale, indépendante de η,

$$\bar{B} = X^{-1} \bar{Y} \qquad \text{où} \qquad \bar{Y} = \sum_{E=1}^{n_{sd}} \bar{A}_{E\Gamma} \bar{W}^{-1}_E \bar{A}^T_{E\Gamma} \tag{4.50}$$

4.3.4 Cas non régulier avec GSNL enrichi

L'analyse locale de la convergence de l'algorithme enrichi nécessite de réécrire l'expression de la nouvelle direction de recherche car $\tilde{v}_E$ et $\tilde{r}_E$ sont définis à partir de H_E tandis que $\bar{v}_E$ et $\bar{r}_E$ sont définis avec G_E. $\tilde{r}_E$ et $\bar{r}_E$ n'ont pas la même dimension, le premier assemble toutes les impulsions de tous les contacts potentiels, avec de nombreuses valeurs nulles, tandis que le second n'assemble que les impulsions des contacts actifs.

$$(\tilde{r}_E - \bar{r}_E) + \frac{1}{\eta}(\tilde{v}_E - \bar{v}_E) = 0 \quad \Longrightarrow \quad (\tilde{r}_E - C_E\bar{r}_E) + \frac{1}{\eta}(\tilde{v}_E - H_E^T\bar{V}_E) = 0 \tag{4.51}$$

En suivant la méthode précédente,

$$\widetilde{R}_E = H_E\widetilde{r}_E = H_E C_E\bar{r}_E - \frac{1}{\eta}H_E H_E^T\widetilde{V}_E + \frac{1}{\eta}H_E H_E^T\bar{V}_E, \tag{4.52}$$

et l'écriture de la dynamique (4.17) donne

$$\widetilde{V}_E = M_{E,\eta}^{-1}\left[R_E^d + H_E C_E\bar{r}_E + \frac{1}{\eta}H_E H_E^T\bar{V}_E - A_{E\Gamma}^T\bar{F}_\Gamma\right]. \tag{4.53}$$

En conséquence, seule la deuxième étape de l'algorithme est modifiée comme suit :

$$\text{Étape 2 :} \quad X_\eta\bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma}M_{E,\eta}^{-1}\left[F_E^d + H_E C_E\bar{r}_E + \frac{1}{\eta}K_E\bar{V}_E\right] \tag{4.54}$$

où $X_\eta = \sum_{E=1}^{n_{sd}} A_{E\Gamma}M_{E,\eta}^{-1}A_{E\Gamma}^T$.

Analyse de la convergence. Il faut calculer $\bar{r}_E$ et $\bar{V}_E$ en fonction de F_Γ sachant que $\bar{v}_E = 0$. $\bar{r}_E$ satisfait la dynamique réduite,

$$\bar{r}_E = \bar{W}_E^{-1}\left(-f_E^d + \bar{A}_{E\Gamma}^T F_\Gamma\right), \tag{4.55}$$

et $\bar{V}_E$ satisfait la dynamique,

$$M_E\bar{V}_E - \bar{R}_E + A_{E\Gamma}^T F_\Gamma = R_E^d, \tag{4.56}$$

or $\bar{R}_E = G_E\bar{r}_E$ donc $\bar{V}_E$ s'écrit,

$$\bar{V}_E = M_E^{-1}\left[G_E\bar{W}_E^{-1}\bar{A}_{E\Gamma}^T - A_{E\Gamma}^T\right]F_\Gamma + M_E^{-1}\left[R_E^d + G_E\bar{W}_E^{-1}v_E^d\right]. \tag{4.57}$$

L'étape 2 s'écrit alors,

$$\begin{array}{rcl} X\eta F_\Gamma & = & \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} H_E \times \\ & & \left[C_E \bar{W}_E^{-1} \bar{A}_{E\Gamma}^T + \frac{1}{\eta} H_E^T M_E^{-1} \left(G_E \bar{W}_E^{-1} G_E^T M_E^{-1} - I_E\right) A_{E\Gamma}^T\right] F_\Gamma \\ & + & \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} \left[\left(I_E + \frac{1}{\eta} H_E H_E^T M_E^{-1}\right) R_E^d \right. \\ & + & \left. \left(-H_E C_E \bar{W}_E^{-1} + \frac{1}{\eta} H_E H_E^T M_E^{-1} G_E \bar{W}_E^{-1}\right) f_E^d\right] \end{array} \tag{4.58}$$

La matrice de contraction dépend du paramètre numérique de flexibilité η,

$$\bar{B}_\eta = X_\eta^{-1} \bar{Y}_\eta \tag{4.59}$$

où

$$\begin{array}{rcl} \bar{Y}_\eta & = & \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} H_E Z_{E,\eta} A_{E\Gamma}^T \\ Z_{E,\eta} & = & C_E \bar{W}_E^{-1} G_E^T M_E^{-1} + \dfrac{1}{\eta}\left(H_E^T M_E^{-1} G_E \bar{W}_E^{-1} G_E^T M_E^{-1} - H_E^T M_E^{-1}\right) \end{array} \tag{4.60}$$

Un tel opérateur est assez difficile à analyser d'un point de vue théorique. Cependant, il peut être testé sur les exemples, en déterminant le rayon spectral en fonction de η.

On remarque sur la Figure 4.9 que le rayon spectral est généralement plus petit que celui de l'algorithme générique, se rapprochant de la valeur du générique quand η augmente, mais devenant plus grand que 1 quand η tend vers zéro. Toutefois, plus le nombre de disques est grand, plus la valeur optimale de η est proche de zéro. Un tel comportement semble exclure l'utilisation de la version asymptotique ($\eta = 0$) pour résoudre un système dynamique non régulier comme un milieu granulaire. Cependant, une telle analyse asymptotique illustre la consistance de l'approche précédente et ouvre des perspectives numériques.

De la même façon que dans (4.25) $M_{E,\eta}^{-1}$ est développé au premier ordre, ce qui permet d'obtenir l'expression de la seconde étape,

$$X^\infty \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} \left[M_E^\infty R_E^d + K_E^+ K_E \bar{V}_E\right] \tag{4.61}$$

Analyse de la convergence. En procédant comme précédemment on obtient,

$$\begin{array}{rcl} X^\infty \bar{F}_\Gamma & = & \sum_{E=1}^{n_{sd}} A_{E\Gamma} K_E^+ H_E \left(H_E^T M_E^{-1} G_E \bar{W}_E^{-1} G_E^T M_E^{-1} - H_E^T M_E^{-1}\right) A_{E\Gamma}^T F_\Gamma \\ & + & \sum_{E=1}^{n_{sd}} A_{E\Gamma} \left[\left(M_E^\infty + K_E^+ H_E H_E^T M_E^{-1}\right) R_E^d \right. \\ & + & \left. K_E^+ H_E H_E^T M_E^{-1} G_E \bar{W}_E^{-1} f_E^d\right] \end{array} . \tag{4.62}$$

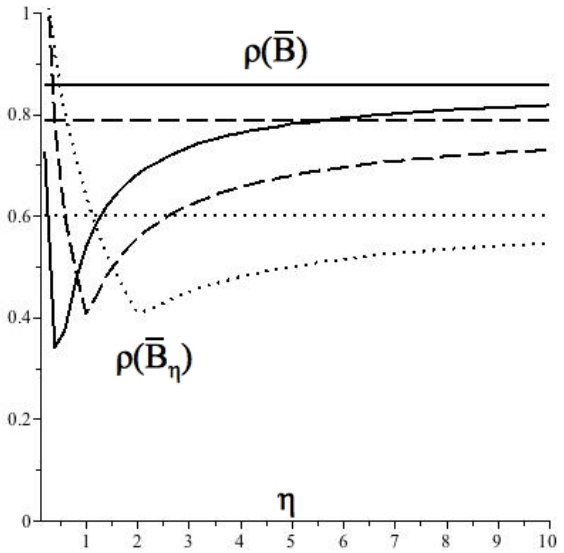


FIG. 4.9 – Rayon spectral asymptotique de l'algorithme enrichi en fonction de η pour les 3 exemples : 4 disques (point), 8 disques (tiret), 12 disques (continus).

Ainsi, quand η tend vers zéro, la matrice de contraction prend la forme suivante,

$$\bar{B}^\infty = X^\infty \bar{Y}^\infty \qquad \begin{array}{ll} \text{où} & \bar{Y}^\infty = \sum_{E=1}^{n_{sd}} A_{E\Gamma} K_E^+ H_E Z^\infty A_{E\Gamma}^T \\ \text{et} & Z^\infty = H_E^T M_E^{-1} G_E \bar{W}_E^{-1} G_E M_E^{-1} - H_E^T M_E^{-1} \end{array} . \tag{4.63}$$

L'opérateur Z^∞ s'annule quand $G_E = H_E$ en accord avec les résultats de 4.3.2 :

$$Z^\infty = \underbrace{H_E^T M_E^{-1} G_E}_{\bar{W}_E \text{ si } G_E = H_E} \bar{W}_E^{-1} G_E M_E^{-1} - H_E^T M_E^{-1} = 0 \text{ si } G_E = H_E \tag{4.64}$$

Les deux étapes sont découplées : la seconde calcule le multiplicateur à l'interface F_Γ et la première détermine les variables internes v_E et r_E. Un tel résultat peut être étendu à un cas plus général où le rang de H_E est plus petit que celui de G_E. Si l'on se réfère à un problème de type granulaire, ceci signifie que les contacts potentiels considérés à l'étape globale sont moins nombreux que les contacts actifs. Dans ce cas, les deux étapes sont toujours découplés mais la solution ne vérifie pas toutes les équations, car la deuxième étape (globale) ne prend pas en compte tous les liens actifs (contacts) représentés par la matrice de passage G_E.

Cas où $Im(H_E) \subset Im(G_E)$. Ceci correspond au cas où l'on prendrait pour construire l'opérateur H_E, supposé basé sur les contacts potentiels, des contacts

dont on serait sûr a priori qu'ils seraient actifs à convergence. L'opérateur Z_E^∞ agit sur les torseurs des grains du sous domaine E, $\mathcal{T}_E$. Tout élément z_E de $\mathcal{T}_E$ peut être décomposé en deux contributions,

$$z_E = G_E x_E + y_E, \tag{4.65}$$

l'une appartenant à $Im(G_E)$ et de sorte que son complémentaire y_E soit M_E^{-1}-orthogonal à $Im(G_E)$, soit, $x_E^T G_E^T M_E^{-1} y_E = 0 \quad \forall x_E$, donc $G_E^T M_E^{-1} y_E = 0$. Donc $\forall z_E \in \mathcal{T}_E$,

$$Z_E^\infty z_E = H_E^T \left[M_E^{-1} G_E \left(G_E^T M_E^{-1} G_E \right)^{-1} G_E^T M_E^{-1} - M_E^{-1} \right] (G_E x + y_E) \tag{4.66}$$

avec

$$Z_E^\infty G_E x_E = \underbrace{H_E^T \left[M_E^{-1} G_E \underbrace{\left(G_E^T M_E^{-1} G_E \right)^{-1} G_E^T M_E^{-1} G_E}_{I_E} - M_E^{-1} G_E \right]}_{=0} x_E \tag{4.67}$$

et

$$Z_E^\infty y_E = H_E^T M_E^- 1 G_E \left(G_E^T M_E^{-1} G_E \right)^{-1} G_E^T M_E^{-1} y_E - H_E^T M_E^{-1} y_E. \tag{4.68}$$

Or par définition $G_E^T M_E^{-1} y = 0$ et puisque $Im(H_E) \subset Im(G_E)$ alors $H_E^T M_E^{-1} y = 0$. Au final dans le cas $Im(H_E) \subset Im(G_E)$ on a,

$$Z_E^\infty = 0. \tag{4.69}$$

Les deux étapes de l'algorithme sont découplées. Si l'on considère un nombre de contacts pour H_E inférieur au nombre de contacts actifs, il suffit de résoudre l'étape 2 pour connaître F_Γ puis la dynamique par sous domaine.

L'algorithme GSNL enrichi dans le cas limite $\eta = 0$, est écrit dans l'algorithme 3.

Algorithm 3 (Version enrichie pour le cas limite $\eta = 0$).

for $i = 1, \ldots, N$ **do** {Boucle sur les pas de temps}
  initialisation des inconnues (r_E, v_E, F_Γ) au temps t_i.
  (r_E, V_E, F_Γ) sont connus, calcul de $(\bar{r}_E, \bar{V}_E, \bar{F}_\Gamma)$:
  while $\zeta < \zeta_{\max}$ (critère de convergence) **do**
    Étape 1 : Calcul de $\bar{r}_E$ en parallèle pour $E = 1, \ldots, n_{sd}$
    avec n_{GS} itérations de Gauss-Seidel sur :

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E \;=\; H_E^T M_E^{-1} (A_{E\Gamma}^T F_\Gamma - R_E^d) \\ \mathcal{R}(\bar{v}_E, \bar{r}_E) = 0 \end{cases}$$

    Étape 2 : Calcul de $\bar{F}_\Gamma$:

$$X^\infty \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} \left[M_E^\infty R_E^d + K_E^+ K_E \bar{V}_E \right]$$

    for $E = 1, \ldots, n_{sd}$ **do**
      $(r_E, V_E, F_\Gamma) \leftarrow (\bar{r}_E, \bar{V}_E, \bar{F}_\Gamma)$
    end for
  end while
end for

4.4 Premières validations numériques

4.4.1 Comparaison des versions générique et enrichie

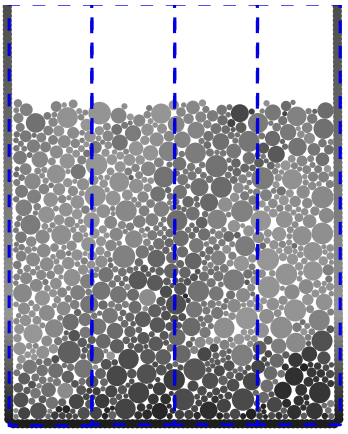


FIG. 4.10 – Découpage en 4 sous domaines d'un échantillon bidimensionnel à 1 000 disques.

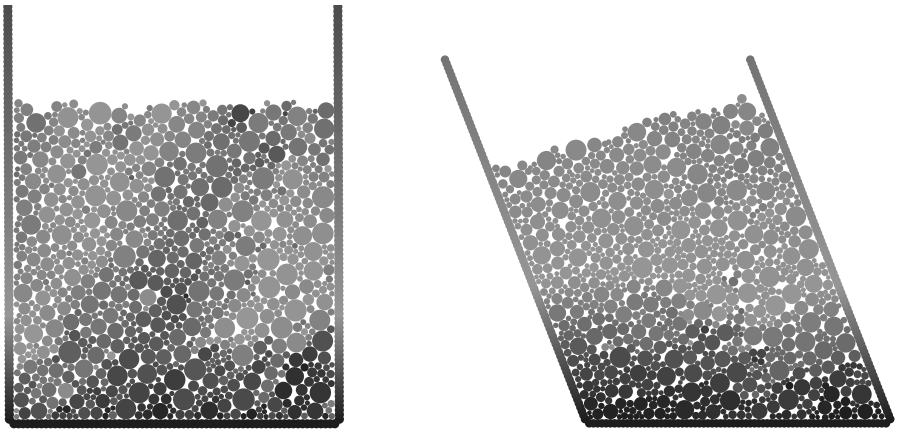


FIG. 4.11 – Essai de cisaillement bidimensionnel sur 1 000 disques.

Les versions générique et enrichie sont testées sur un exemple de cisaillement de faible dimension, avec quatre sous domaines dont le découpage est illustré Figure 4.10. Les configurations initiale et finale de ce test sont illustrées sur la Figure 4.11. Dans la version enrichie le paramètre η a été choisi $\eta = \frac{1}{\text{moyenne}((W^D)^{-1})}$

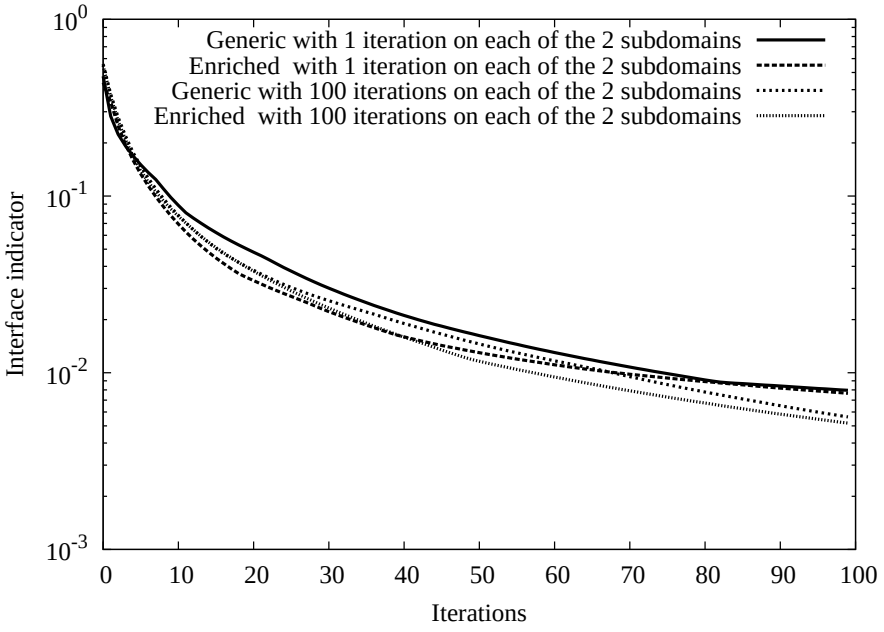


FIG. 4.12 – Comparaison des algorithmes classique et enrichi sur l'essai de cisaillement à 1 000 disques.

en accord avec le choix de la direction de recherche optimum réalisé au chapitre 2.3.3. Pour chaque version, sur un pas de temps on réalise 100 itérations (que l'on considère suffisante pour garantir la convergence) des algorithmes générique et enrichi. Pour chacune de ces itérations on réalise un certain nombre d'itérations de GSNL (n_{GS}) entre chaque mise à jour d'interface. Deux nombres d'itérations de Gauss-Seidel par sous domaine sont testés, $n_{\mathrm{GS}} = 1$ et $n_{\mathrm{GS}} = 100$, voir Figure 4.12.

De manière synthétique, les taux de convergence sont similaires, avec une faible amélioration pour l'algorithme enrichi.

On interprète que que pour $n_{\mathrm{GS}} = 1$, le gain est plus prononcé pendant les premières itérations dans la version enrichie, et disparaît quand la majorité des corrections sont locales, dans les sous domaines, et concernent principalement le réseau des forces faibles [52].

Pour $n_{\mathrm{GS}} = 100$, la version enrichie donne moins d'amélioration au départ mais le taux de convergence asymptotique est le même que la version générique est, bien sûr, meilleur que celui de $n_{\mathrm{GS}} = 1$. Toutefois la différence entre les deux est faible en rapport avec la différence de coût en nombre d'opérations.

4.4.2 Stratégies de résolutions incomplètes

Il apparaît Figure 4.12 qu'il n'est pas du tout évident de devoir faire converger chaque sous domaine avant de faire la mise à jour des interfaces pour passer à l'itéré suivant. Lors de l'écriture de la version enrichie de Gauss Seidel appliquée au cas granulaire, la résolution des efforts d'interface à chaque itération s'écrit,

$$X_\eta \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} \left[R_E^d + H_E(\bar{r}_E + \frac{1}{\eta}\bar{v}_E) \right]. \tag{4.70}$$

La résolution de cette équation est basée sur le X_η qui a été enrichi de manière à coupler les interfaces. La dimension de cette matrice est fonction du nombre total de grains d'interface. Cette résolution est par conséquent chère en temps de calcul. Pour diminuer ce temps de calcul un découpage de X_η est réalisé, en extrayant X_η^D, partie diagonale de X_η. Un nouveau paramètre est introduit : n_{Inter}, le nombre de sous itérations du problème d'interface. Une sous itération d'interface est régie par l'équation (4.71) réalisant un passage de $\hat{F}_\Gamma$ à $\tilde{F}_\Gamma$.

$$X_\eta^D \tilde{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{E,\eta}^{-1} \left[R_E^d + H_E(\bar{r}_E + \frac{1}{\eta}\bar{v}_E) \right] - (X_\eta - X_\eta^D)\hat{F}_\Gamma. \tag{4.71}$$

Ces valeurs d'interfaces sont des intermédiaires au cours des sous itérations entre F_Γ à $\bar{F}_\Gamma$, valeur du multiplicateur à l'interface au début et à la fin d'une itération. Une itération complète comprenant désormais n_{GS} sous itérations du problème local par sous domaine et n_{Inter} sous itérations du problème d'interface.

Au sein d'un pas de temps on a un certain nombre d'itérations à faire. Pour chacune de ces itérations on peut sous itérer le problème local qui donne la résolution de l'ensemble des contacts dans chaque sous domaine, on ira donc plus ou moins loin dans la résolution sur chaque sous domaine en fonction du nombre de sous itérations n_{GS}. Maintenant on peut aussi faire un certain nombre de sous itérations d'interface. Avec cette décomposition de X_η l'algorithme enrichi à résolution partielle est écrit dans l'algorithme 4.

Plusieurs options sont alors possibles en terme d'algorithme, en allant de la plus à la moins coûteuse. La première consiste à faire suffisamment de sous itérations d'interface n_{Inter} pour avoir convergé vers la solution du problème initial donnant les efforts d'interface, et ce à chacune des itérations au sein d'un pas de temps. Cette solution revient au même que la résolution directe du problème d'interface et est aussi coûteuse. Dans le cadre de l'étude qui suit ce paramètre est de 100 sous itérations d'interfaces pour garantir la convergence du problème d'interface. Il a été pris par défaut égal au nombre maximum d'itérations fait sur un pas de

Algorithm 4 (Version enrichie avec résolution itérative du problème d'interface).

for $i = 1, \ldots, N$ **do** {Boucle sur les pas de temps}
  initialisation des inconnues (r_E, v_E, F_Γ) au temps t_i.
  quand (r_E, V_E, F_Γ) sont connus, calcul de $(\bar{r}_E, \bar{V}_E, \bar{F}_\Gamma)$:
  while $\zeta < \zeta_{\max}$ (critère de convergence) **do**
    Étape 1 : Calcul de $\bar{r}_E$ en parallèle pour $E = 1, \ldots, n_{sd}$
    avec n_{GS} itérations de Gauss-Seidel sur :

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E \;=\; H_E^T M_E^{-1} (A_{E\Gamma}^T F_\Gamma - R_E^d) \\ \mathcal{R}(\bar{v}_E, \bar{r}_E) = 0 \end{cases}$$

    Étape 2 : Calcul de $\bar{F}_\Gamma$:
    for $j = 1, \ldots, n_{\mathrm{Inter}}$ **do**
      $X_\eta^D \bar{F}_\Gamma = \sum_{E=1}^{n_{sd}} A_{E\Gamma} M_{\eta,E}^{-1} [R_E^d + H_E(\bar{r}_E + \frac{1}{\eta}\bar{v}_E)] - (X_\eta - X_\eta^D) F_\Gamma$
      $(F_\Gamma) \leftarrow (\bar{F}_\Gamma)$
    end for
    for $E = 1, \ldots, n_{sd}$ **do**
      $(r_E, V_E, F_\Gamma) \leftarrow (\bar{r}_E, \bar{V}_E, \bar{F}_\Gamma)$
    end for
  end while
end for

temps. Ensuite, on peut tenter de diminuer ce nombre de sous itérations d'interface jusqu'à une seule sous itération. Dans la suite, trois valeurs de ce paramètre sont testées : 100, le maximum garantissant une résolution complète de l'interface, 2 sous itérations représentant une potentielle diminution du coût de calcul et une résolution moins fine sur l'interface, et 1 sous itération qui serait l'optimum en terme de coût mais qui semble ne pas fonctionner et qui n'est pas affichée sur les courbes suivantes, l'algorithme divergeant.

Dans l'étude qui suit on va chercher à déterminer les valeurs optimums des deux nombres d'itérations n_{Inter} et n_{GS}.

Cas linéaire.

On se place dans le cas linéaire identique à celui du chapitre 2.4.1, le paramètre linéaire $k = \text{moyenne}\left(W^D\right)^{-1}$ et $\eta = \frac{1}{k}$.

On commence par réaliser entre chaque mise à jour d'interface une seule sous itération du problème local soit : $n_{\text{GS}} = 1$. La Figure 4.13 donne pour deux cas de sous itérations d'interface ($n_{\text{Inter}} = 2$ ou 100) deux indicateurs au cours des itérations (chaque itération comportant alors $n_{\text{GS}} = 1$ sous itération dans le sous domaine et suivant le cas $n_{\text{Inter}} = 2$ ou 100 sous itérations à l'interface). Le premier indicateur est un indicateur en volume décrit par l'équation (2.66) et le second l'indicateur d'interface par (2.67).

Avec deux sous itérations d'interface, on a une erreur très proche de celle calculée avec 100. Pour le cas $n_{\text{Inter}} = 100$ les valeurs aux interfaces sont justes dès la première itération, en revanche il faut une quarantaine d'itérations pour converger à l'intérieur des sous domaines. On est avec le cas $\eta = \frac{1}{k}$ dans une configuration où la résolution de l'interface est indépendante des valeurs obtenues par sous domaines (cf chapitre 4.2.2) et l'on voit que le nombre d'itérations pour obtenir la convergence sur l'erreur en volume reste lui à 40 malgré cela. Pour le cas $n_{\text{Inter}} = 2$ la résolution à l'interface prend 40 itérations et au final l'indicateur en volume ne prend pas de retard.

Il n'apparaît pas si intéressant de résoudre complètement l'interface si la résolution par sous domaine est trop incomplète ($n_{GS} = 1$).

On réalise, ensuite, pour la Figure 4.14, entre chaque mise à jour d'interface 100 sous itérations du problème local soit : $n_{\text{GS}} = 100$. Avec ce critère on estime qu'on résout complètement le problème sur les sous domaines à chaque itération.

Les deux indicateurs sont représentés Figure 4.14. En ce qui concerne l'erreur en volume dans le cas $n_{\text{Inter}} = 100$ on a bien convergence en une seule itération. Cette itération comprend $n_{\text{Inter}} = 100$ sous itérations d'interface qui donnent avec $\eta = 1/k$

la valeur solution des efforts d'interface et $n_{\text{GS}} = 100$ sous itérations du problème par sous domaine qui donne la solution puisque les valeurs d'interface sont les bonnes. Pour le cas $n_{\text{Inter}} = 2$ on a au final besoin d'autant d'itérations que quand on avait $n_{\text{GS}} = 1$ sous itération du problème local (Figure 4.13). Dans le cas précédent on avait un véritable parallèle entre interface et sous domaines. Si on ne résout pas l'interface complètement il n'est pas utile de faire une résolution fine complète dans les sous domaines.

On n'a donc pas besoin de beaucoup sous itérer sur l'interface et en parallèle il faut peu itérer sur le sous domaine.

Non Linéaire.

On revient au problème non linéaire, contact unilatéral sans frottement, de la Figure 4.10. On réalise, pour la Figure 4.16, entre chaque mise à jour d'interface, 1 sous itération du problème local soit : $n_{\text{GS}} = 1$ mais ici dans le cas non linéaire. η est toujours égal à $\frac{1}{\text{moyenne}((W^D)^{-1})}$.

Sur la Figure 4.15, pour les deux indicateurs, les courbes sont parfaitement superposées. Les cas $n_{\text{Inter}} = 2$ et $n_{\text{Inter}} = 100$ continuent à fonctionner. La convergence au niveau de l'interface est très lente. Le cas $n_{\text{Inter}} = 2$ est tout à fait aussi efficace que le $n_{\text{Inter}} = 100$. Il semble donc inutile, comme pour le cas linéaire, de réaliser complètement le calcul de l'interface à chaque itération lorsque la résolution dans les sous domaines n'est effectuée que partiellement (La deuxième étape dans le cas non linéaire dépendant des valeurs des impulsions locales issues de la première étape).

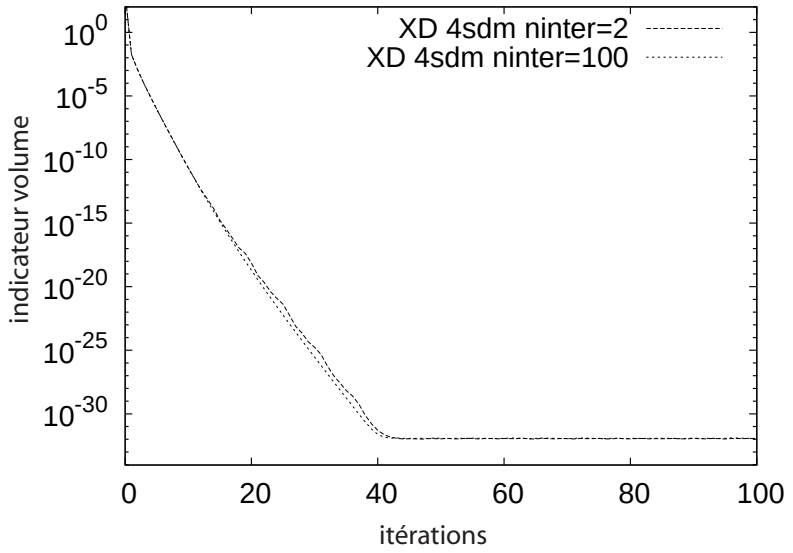
On réalise, pour la Figure 4.16, entre chaque mise à jour d'interface 100 sous itérations du problème local soit : $n_{\text{GS}} = 100$. Avec ce critère on estime qu'on résout complètement le problème sur les sous domaines à chaque itération.

Sur la Figure 4.16 et pour les deux indicateurs, les courbes sont à nouveau superposées parfaitement. Les cas $n_{\text{Inter}} = 2$ et $n_{\text{Inter}} = 100$ continuent à fonctionner et à donner des résultats similaires. Il semble donc à nouveau inutile de réaliser complètement le calcul de l'interface à chaque itération, et ceci même en résolvant finement les sous domaines (même si ici le calcul de l'interface est dépendant de la résolution des sous domaines à l'inverse du cas linéaire). En conclusion aucune des deux étapes n'a besoin d'être résolue finement. Le calcul de l'un dépendant fortement de la mise à jour de l'autre, les deux étapes étant liées.

Pour la comparaison entre la résolution de 100 itérations sur le sous domaine et une seule, on a vu Figure 4.12 qu'il est peu intéressant même avec une résolution complète de l'interface de sous itérer sur les sous domaines. Ce résultat est confirmé par les faibles différences entre les Figures 4.15 et 4.16, sur lesquelles des repères

ont été ajoutés, donnant pour une valeur d'erreur donnée les nombres d'itérations correspondants

Au final le choix optimum ici est d'utiliser à chaque itération, une seule sous itération dans les sous domaines et 2 aux interfaces (1 seule ne semblant pas fonctionner), ceci afin de véritablement résoudre de front les deux étapes.



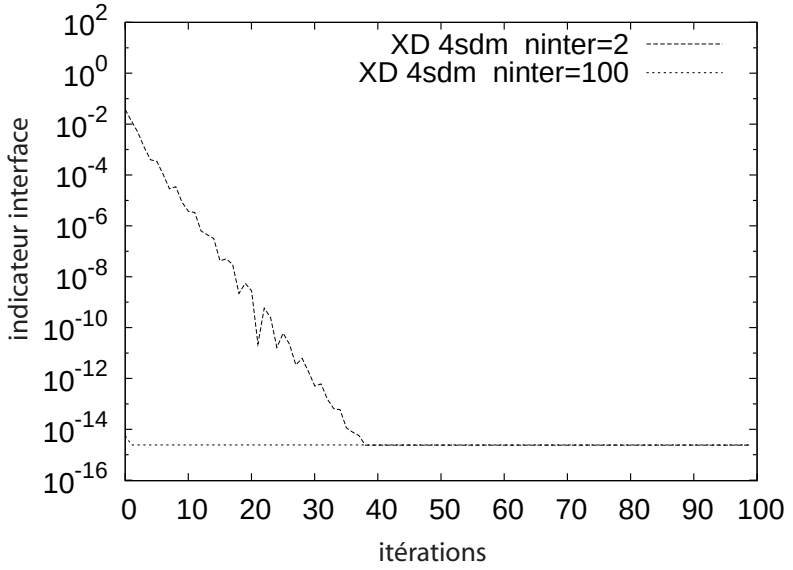
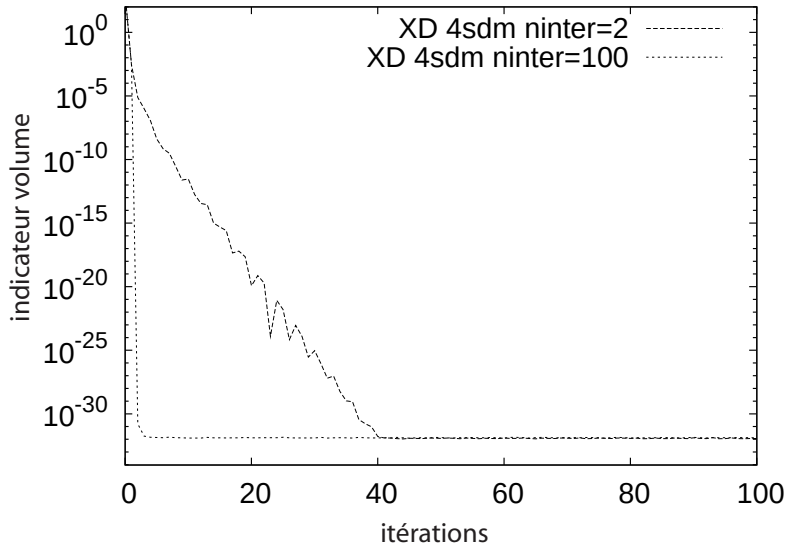


FIG. 4.13 – Erreur en volume et indicateur à l'interface dans le cas linéaire avec $\eta = \frac{1}{k}$ et $n_{\mathrm{GS}} = 1$.



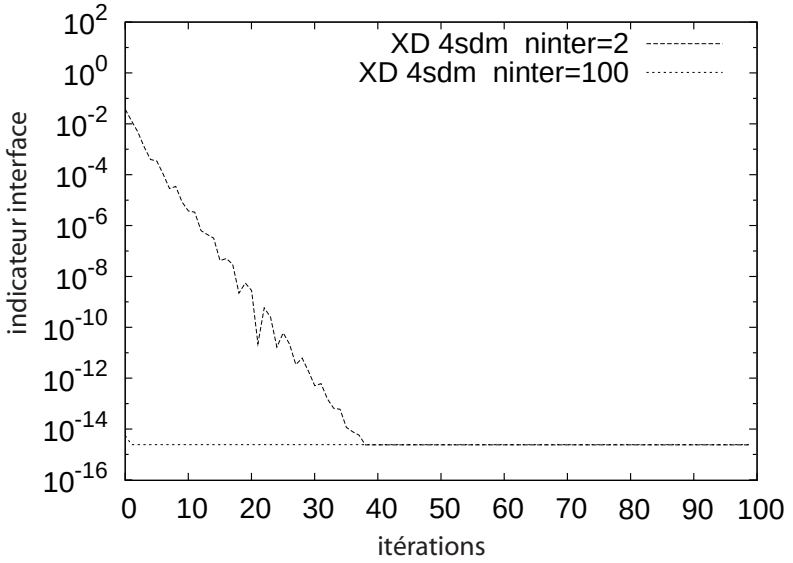
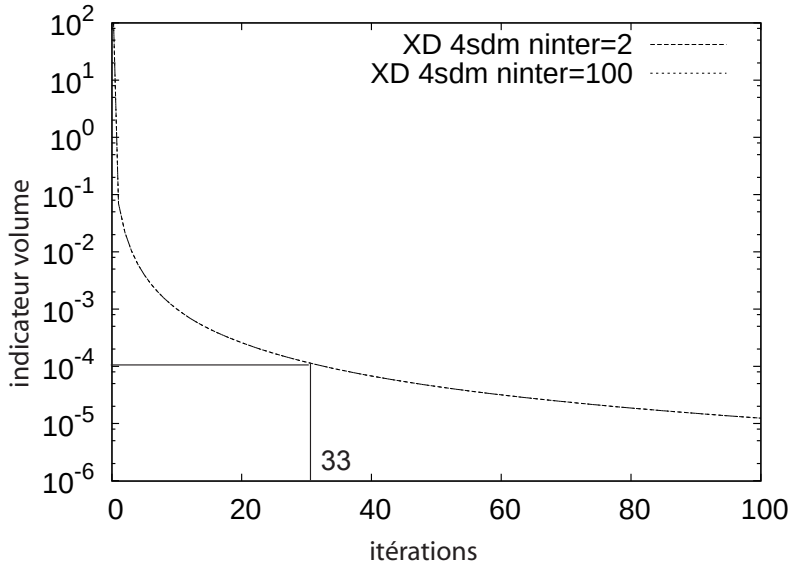


FIG. 4.14 – Erreur en volume et indicateur à l'interface dans le cas linéaire avec $\eta = \frac{1}{k}$ et $n_{\mathrm{GS}} = 100$.



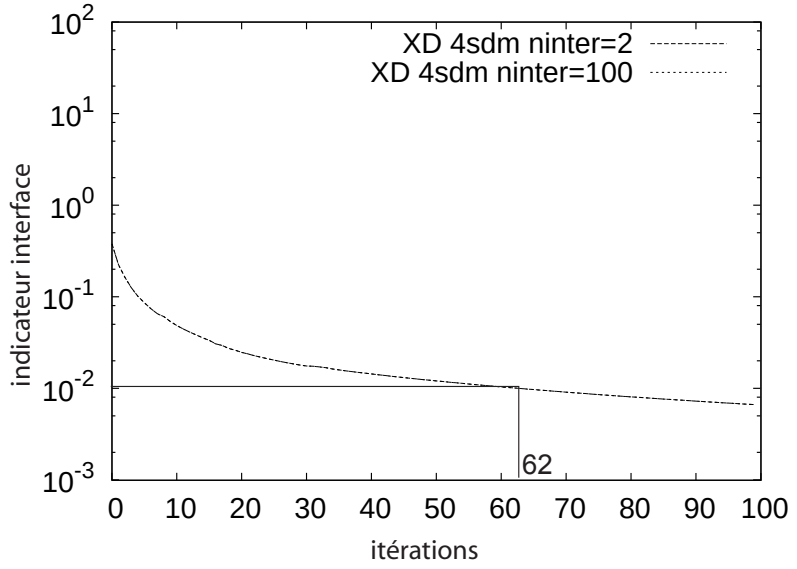
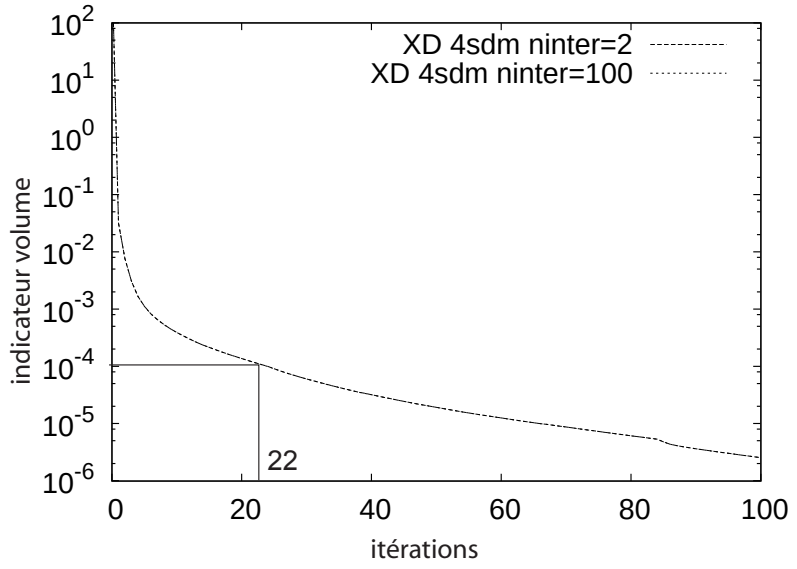


FIG. 4.15 – Erreur en volume et indicateur à l'interface dans le cas non linéaire $n_{\mathrm{GS}} = 1$.



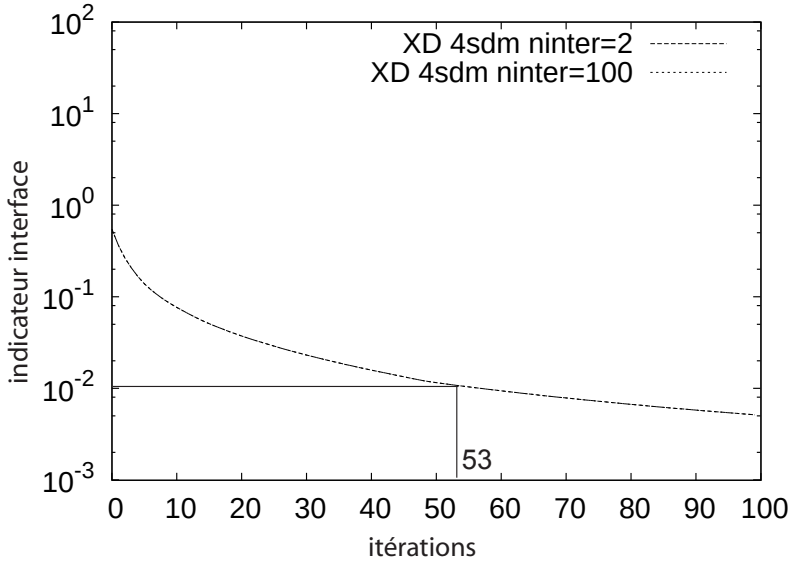


FIG. 4.16 – Erreur en volume et indicateur à l'interface dans le cas non linéaire ($n_{\mathrm{GS}} = 100$).

Conclusions et perspectives

Afin de répondre au besoin du LMGC de disposer d'un logiciel permettant le calcul de grands systèmes granulaires, des développements importants de LMGC90, outil de calcul pour éléments discrets en contact, ont été réalisés. Ces développements se sont appuyés sur les méthodes de décomposition de domaine, initialement mises en place pour la résolution de problèmes de grande dimension sur des milieux continus discrétisés. Ainsi, les objectifs de cette thèse consistaient à mettre en place des algorithmes en décomposition de domaine, adaptés à un système de type granulaire. Ces algorithmes devaient être utilisés pour paralléliser la plateforme de calcul existante, et devaient également être optimisés de manière à améliorer leurs vitesses de convergence. Deux stratégies d'optimisation numérique ont été testées : soit en recourant à des méthodes classiques comme l'introduction de problèmes macroscopiques, soit en envisageant des stratégies plus spécifiques au milieu considéré.

Notre but n'est pas simplement de disposer d'une version parallèle du code, ou des parties de celui-ci (génération des corps, détection des contacts, solveur). Le solveur GSNL bénéficie déjà d'une version OpenMP [56] exigeant de stocker dans une mémoire partagée tout le problème. Il s'agit aussi de préparer l'exploitation des possibilités de la décomposition de domaine dans le domaine du dialogue discret/continu. En effet, si un milieu granulaire est intrinsèquement discret, la définition de grandeurs macroscopiques sur un sous-domaine, et de relations les liant, peut être en mesure de nous fournir le comportement du milieu continu équivalent.

A l'issue d'une analyse des méthodes de décomposition de domaine et des spécificités de la dynamique, certaines orientations ont été définies. Ainsi la nature diffuse des interactions non régulières ne permet pas d'isoler les non régularités uniquement aux interfaces entre sous-domaines. Le choix en ce qui concerne le découpage est donc fait de manière à obtenir un comportement d'interface parfait : les non régularités se retrouvent confinées au sein des sous-domaines. On retrouve une telle configuration dans [2] pour les milieux cellulaires. Cependant le rapport nombre de contacts sur nombre de degrés de libertés y était faible et justifiait l'utilisation d'un algorithme de Newton généralisé.

La décomposition de domaine était alors utilisée pour résoudre le problème linéarisé à chaque itération de Newton. La valeur élevée du rapport défini précédemment dans un milieu granulaire semble exclure ici cette approche. En conséquence la méthode de décomposition de domaine sera directement appliquée au problème non linéaire (non régulier) et non pas à un problème, ou une succession de problèmes, linéarisés.

À partir de ce découpage, deux algorithmes ont été mis en oeuvre et comparés. Les liens entre ces algorithmes ont mis en évidence l'influence du choix des paramètres numériques internes à chacune de ces méthodes, nommées directions de recherche. L'implantation de ces algorithmes dans la plateforme de calcul à été et reste une des difficultés de cette thèse. La version actuelle est basée sur des échanges MPI permettant une résolution multiprocesseur et a été testée en parallèle avec plus de 16 processeurs sur des exemples 2D et 3D avec frottement. Cette implantation a permis de mettre en évidence l'extensibilité numérique des deux stratégies proposées. En effet, sans ajout d'une échelle macroscopique, l'augmentation du nombre de sous-domaines n'affecte que très faiblement le nombre d'itérations à réaliser par pas de temps. La partie bloquante pour réaliser des simulations à plus grande échelle ne concerne pas les algorithmes, mais l'implantation elle-même qui nécessite une optimisation du code en terme de mémoire et d'architecture logicielle.

Toutefois, une échelle macroscopique a été imaginée et mise en place dans cette thèse au sein des deux premiers solveurs, afin d'en créer deux nouvelles versions. La prise en compte d'une échelle macroscopique sur les variables d'interface, n'améliore pas la convergence des schémas numériques. Ceci est imputable à la "pauvreté" du problème d'interface lui-même où la matrice X s'avère diagonale dans GSNL. Ainsi les variables d'interface d'un même sous-domaine ne sont pas couplées par cette matrice. La projection sur un espace macroscopique n'induit aucun couplage supplémentaire. La stratégie n'est donc pas capable de transmettre des informations à grande longueur d'onde, autrement dit d'une interface à l'autre d'un même sous-domaine.

L'enrichissement du problème d'interface est basé sur l'adjonction d'une direction de recherche impliquant un paramètre numérique non physique. La convergence du schéma enrichi, est, à l'issue d'une première étude, faiblement améliorée. De plus la détermination du paramètre numérique optimal reste une question ouverte. D'autant que la valeur limite nulle semble exclue par notre analyse. Enfin le surcoût de résolution du problème d'interface enrichi est important, puisque, contrairement à la version générique, ce problème requiert lui-même un solveur de type méthode de

décomposition de domaine. Une manière de pallier les inconvénients ci-dessus est de procéder à des résolutions incomplètes des deux étapes du schéma numérique. Une première étude montre l'intérêt de cette procédure et doit être poursuivie.

Les perspectives ouvertes par ce travail sont de différentes natures. La version générique de l'algorithme GSNL en décomposition de domaine étant extensible, sa mise en exploitation exige maintenant une optimisation du code, minimisant les allocations mémoires à différents niveaux et le recours au langage interprété Python.

La version enrichie laisse, quand à elle, de nombreuses questions ouvertes et donc des perspectives nouvelles en terme d'études théoriques et numériques. Ainsi il est nécessaire d'analyser la pertinence d'un problème macroscopique sur la version enrichie, sa signification et sa dépendance vis-à-vis d'un paramètre numérique. Une fois cette question éclaircie, il sera alors envisageable de substituer, sous certains critères, le problème homogénéisé au problème par sous domaine. L'implantation optimale de cette version requerra le branchement au code LMGC90 d'une méthode de décomposition de domaine pour résoudre le problème d'interface. Enfin un nouvel algorithme pourrait être dérivé de la version enrichie consistant en un schéma incomplet de résolution sur un pas de temps, mais résolvant correctement le problème d'évolution d'un granulat lors d'un processus. A ce titre le réseau fort des contacts [52], relativement stable dans le temps serait un candidat à la définition d'un problème d'interface grossier...

Bibliographie

[1] P. Alart, M. Barboteu, and F. Lebon. Solution of frictional contact problems by an EBE preconditioner. *Computational Mechanics*, 20 :370–378, 1997.

[2] P. Alart, M. Barboteu, and M. Renouf. Parallel computational strategies for multicontact problems : Application to cellular and granular media. *International Journal for Multiscale Computational Engineering*, 1 :419–430, 2003.

[3] P. Alart and D. Dureisseix. A scalable multiscale LATIN method adapted to nonsmooth discrete media. *Computer Methods in Applied Mechanics and Engineering*, 197(5) :319–331, 2008.

[4] P. Avery and C. Farhat. The FETI family of domain decomposition methods for inequality-constrained quadratic programming : Application to contact problems with conforming and nonconforming interfaces. *Computer Methods in Applied Mechanics and Engineering*, 2009. In Press, Accepted Manuscript.

[5] P. Avery, G. Rebel, M. Lesoinne, and C. Farhat. A numerically scalable dual-primal substructuring method for the solution of contact problems – Part I : The frictionless case. *Computer Methods in Applied Mechanics and Engineering*, 193 :2403–2426, 2004.

[6] M. Barboteu, P. Alart, and M. Vidrascu. A domain decomposition strategy for nonclassical frictional multi-contact problems. *Computer Methods in Applied Mechanics and Engineering*, 190 :4785–4803, 2001.

[7] P. E. Bjorstad and O. B. Widlund. Iterative methods for the solution of elliptic problems on regions partitionned into substructures. *SIAM Journal on Numerical Analysis*, 23 :1097–1120, 1986.

[8] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Mathematics of Computation*, 47(175) :103–134, 1986.

[9] L. Champaney, J.-Y. Cognard, D. Dureisseix, and P. Ladevèze. Numerical experimentations of parallel strategies in structural non-linear analysis. *Calculateurs Parallèles*, 8(2) :245–249, 1996.

[10] L. Champaney, J.-Y. Cognard, and P. Ladevèze. Modular analysis of assemblages of three-dimensional structures with unilateral contact conditions. *Computers and Structures*, 73 :249–266, 1999.

[11] T. F. Chan and D. Goovaerts. On the relationship between overlapping and nonoverlapping domain decomposition methods. *SIAM Journal on Matrix Analysis and Applications*, 13 :663–670, 1992.

[12] P. W. Christensen, A. Klarbring, J. S. Pang, and N. Strömberg. Formulation and comparison of algorithms for frictional contact problems. *International Journal for Numerical Methods in Engineering*, 42 :145–173, 1998.

[13] P. A. Cundall and O. D. L. Stack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1) :47–65, 1979.

[14] Y.-H. De Roeck, P. Le Tallec, and M. Vidrascu. A domain decomposed solver for nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering*, 99 :187–207, 1992.

[15] G. Dilintas, P. Laurent-Gengoux, and D. Trystram. A conjugate projected gradient method with preconditioning for unilateral contact problems. *Computers and Structures*, 29(4) :675–680, 1988.

[16] Z. Dostál, A. Friedlander, and S. Santos. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemporary Mathematics*, 218 :82–93, 1998.

[17] Z. Dostál, F. A. M. Gomes Neto, and S. A. Santos. Solution of contact problems by FETI domain decomposition with natural coarse space projection. *Computer Methods in Applied Mechanics and Engineering*, 190(13-14) :1611–1627, 2000.

[18] Z. Dostál, D. Horák, R. Kucera, V. Vondrák, J. Haslinger, J. Dobiás, and S. Pták. FETI based algorithms for contact problems : scalability, large displacements and 3d coulomb friction. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5) :395–409, 2005.

[19] D. Dureisseix. *Vers des stratégies de calcul performantes pour les problèmes multiphysiques et le passage par le multiéchelle.* Mémoire d'habilitation à diriger les recherches, Université Paris 6, 2001.

[20] D. Dureisseix, P. Alart, and S. Nineb. Une méthode de décomposition de domaine multiéchelle pour les structures discrètes en mécanique non régulière. In *Congrès SMAI 2007*, Praz sur Arly, juin 2007. symposium Décomposition de domaines en mécanique des structures.

[21] D. Dureisseix and C. Farhat. A numerically scalable domain decomposition method for the solution of frictionless contact problems. *International Journal for Numerical Methods in Engineering*, 50(12) :2643–2666, 2001.

[22] Y. Escaig, G. Touzot, and M. Vayssade. Parallelization of a multilevel domain decomposition method. *Computing Systems in Engineering*, 5(3) :253–263, 1994.

[23] C. Farhat, P. S. Chen, and J. Mandel. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *International Journal for Numerical Methods in Engineering*, 38(22) :3831–3854, 1995.

[24] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen. FETI-DP : A dual-primal unified FETI method – Part I : A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7) :1523–1544, 2001.

[25] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numerical Linear Algebra with Applications*, 7 :687–714, 2000.

[26] Ch. Farhat. A Lagrange multiplier based divide and conquer finite element algorithm. *Journal of Computing Systems in Engineering*, 2 :149–156, 1991.

[27] Ch. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6) :1205–1227, 1991.

[28] Ch. Farhat and F.-X. Roux. An unconventional domain decomposition method for an efficient parallel solution of large–scale finite element systems. *SIAM Journal on Scientific and Statistical Computing*, 13 :379–396, 1992.

[29] Ch. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 2. North-Holland, 1994.

[30] C. Glocker and F. Pfeiffer. Multiple impacts with friction in rigid multibody systems. *Nonlinear Dynamics, Transactions of A.S.M.E.*, 7(4) :471–497, 1995.

[31] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13(4) :515–572, 2005.

[32] J. Haslinger, Z. Dostál, and R. Kucera. On a splitting type algorithm for the numerical realization of contact problems with Coulomb friction. *Computer Methods in Applied Mechanics and Engineering*, 191(21-22) :2261–2281, 2002.

[33] T. J. R. Hughes and T. Belytschko. *Nonlinear Finite Element Analysis*. Zace services Ltd – ICE Division, 1995.

[34] M. Jean. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177 :235–257, 1999.

[35] F. Jourdan, P. Alart, and M. Jean. A Gauss-Seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 155(1-2) :31–47, 1998.

[36] jr. R. H. Dodds and L. A. Lopez. Substructuring in linear and non linear analysis. *International Journal for Numerical Methods in Engineering*, 15 :583–597, 1980.

[37] P. Ladevèze. Sur une famille d'algorithmes en mécanique des structures. *Comptes-Rendus de l'Académie des Sciences*, 300(2) :41–44, 1985.

[38] P. Ladevèze. *Nonlinear computational structural mechanics — New approaches and non-incremental methods of calculation.* Springer Verlag, 1999.

[39] P. Ladevèze, A. Nouy, and O. Loiseau. A multiscale computational approach for contact problems. *Computer Methods in Applied Mechanics and Engineering*, 191 :4869–4891, 2002.

[40] P. Le Tallec. Domain-decomposition methods in computational mechanics. *Computational Mechanics Advances*, 1(2) :121–220, 1994. North-Holland.

[41] P. Le Tallec and M. Vidrascu. Méthode de décomposition de domaines en calcul de structures. *Actes du premier colloque national en calcul des structures*, I :33–49, 1993.

[42] C. Liu, Z. Zhao, and B. Brogliato. Frictionless multiple impacts in multibody systems. I. Theoretical framework. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science*, 464(2100) :3193–3211, 2008.

[43] C. Liu, Z. Zhao, and B. Brogliato. Frictionless multiple impacts in multibody systems. II. Numerical algorithm and simulation results. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Science*, 465(2101) :1–23, 2009.

[44] J. Mandel. Balancing domain decomposition. *Communications in Applied Numerical Methods*, 9 :233–241, 1993.

[45] J. Mandel and C. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical Linear Algebra with Applications*, 10 :639–659, 2003.

[46] J. Mandel, R. Tezaur, and Ch. Farhat. A scalable substructuring method by Lagrange multipliers for plate bending problems. *SIAM Journal on Numerical Analysis*, 36(5) :1370–1391, 1999.

[47] H.-O. May. The conjugate gradient method for unilateral problems. *Computers and Structures*, 12(4) :595–598, 1986.

[48] A. Mobasher Amini, D. Dureisseix, and P. Cartraud. Multi-scale domain decomposition method for large scale structural analysis with a zooming technique : Application to plate assembly. *International Journal for Numerical Methods in Engineering*, 79(4) :417–443, 2009.

[49] J. J. Moreau. Numerical aspects of sweeping process. *Computer Methods in Applied Mechanics and Engineering*, 177 :329–349, 1999.

[50] S. Nineb. *Modélisations et outils numériques en mécanique non réguliére et applications à la tenségrité*. PhD thesis, Université Montpellier 2, 2007.

[51] S. Nineb, P. Alart, and D. Dureisseix. Domain decomposition approach for nonsmooth discrete problems, example of a tensegrity structure. *Computers and Structures*, 85(9) :499–511, 2007.

[52] F. Radjai, D. E. Wolf, M. Jean, and J. J. Moreau. Bimodal character of stress transmission in granular packings. *Physical Review Letters*, 80(1) :61–64, 1998.

[53] A. Rafiee, M. Vinches, and C. Bohatier. Modelling and analysis of the Nîmes arena and the Arles aqueduct subjected to a seismic loading, using the Non-Smooth Contact Dynamics method. *Engineering Structures*, 30(12) :3457–3467, 2008.

[54] G. Rebel, C. Farhat, M. Lesoinne, and P. Avery. A scalable dual-primal domain decomposition method for the solution of contact problems with friction. In *7th U.S. National Congress on Computational Mechanics — USNCCM7*, 2003.

[55] M. Renouf and P. Alart. Conjugate gradient type algorithms for frictional multi-contact problems : Applications to granular materials. *Computer Methods in Applied Mechanics and Engineering*, 194 :2019–2041, 2004.

[56] M. Renouf, F. Dubois, and P. Alart. A parallel version of non smooth contact dynamics algorithm applied to simulation of granular medium. *Journal of Computational and Applied Mathematics*, 168 :375–382, 2004.

[57] G. Saussine. *Contribution à la modélisation de granulats tridimensionnels : application au ballast*. PhD thesis, Universtité Montpellier 2, 2004.

[58] G. Saussine, C. Cholet, P.-E. Gautier, F. Dubois, C. Bohatier, and J. J. Moreau. Modelling ballast behaviour under dynamic loading, part 1 : A 2D polygonal discrete element method approach. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22) :2841–2859, 2005.

[59] H. A. Schwartz. über einige abbilldungsaufgaben. *Gesammelte Mathematishe Abhandlungen*, 11 :65–83, 1869.

[60] Marc Snir, editor. *MPI : The Complete Reference, second edition*. MIT Press, 1998.