

Université des Sciences et Technologies de Lille 1
Laboratoire de Mécanique de Lille
École doctorale Sciences pour l'Ingénieur Université Lille Nord de France

* * *

Thèse

Présentée en vue

D'obtenir le grade de

Docteur

en

Spécialité : Génie Civil

Titre de la thèse :

**Modélisation multi-échelles des shales.
Influence de la microstructure sur les propriétés
macroscopiques et le processus de fracturation.**

Par

Alexis Vallade

Soutenue le 7 Novembre 2016 devant le jury d'examen :

Pr. Fabrice GATUINGT	LMT, ENS Cachan	<i>Président du jury</i>
Pr. Albert GIRAUD	INPL - École de Géologie	<i>Rapporteur</i>
Pr. Thierry MASSART	Université Libre de Bruxelles	<i>Rapporteur</i>
Dr. Yun JIA	LML, Université de Lille 1	<i>Examinatrice</i>
Dr. Atef ONAISI	Total	<i>Examineur</i>
Dr. Hamid POURPAK	Total	<i>Examineur</i>
Pr. Jian Fu SHAO	LML, Université Lille 1	<i>Directeur de thèse</i>
Pr. Jean Baptiste COLLIAT	LML, Université Lille 1	<i>Directeur de thèse</i>

Laboratoire de mécanique de Lille

Boulevard Paul Langevin Cité Scientifique 59655 Villeneuve d'Ascq cedex

Abstract

This research study aims at developing tools and numerical methods to model the influence of the microstructure of shales on macroscopic properties and cracking process.

The first part of the document is dedicated to the description of a 3D Finite Elements model (E-FEM) developed to represent the microstructure and cracking phenomena. This model is part of the methods with discontinuities. Two cracking criteria are described, a mode I criteria (Rankine) and a mode II criteria (Mohr-Coulomb). These criteria will be used to characterize the influence of the microstructure of shales on the macroscopic properties in triaxial compression testing. More particularly, the impact of the presence of kerogen in the shale rock is considered.

The second part presents a domain decomposition method (mortar method) used to reduce computation time. This method has the advantage of allowing the use of non-conforming mesh, so a local mesh refinement is possible. This method has been integrated into a computing code using the component-oriented programming and more specifically the CTL middleware. The computing code solves both linear and nonlinear problems using the E-FEM model.

The last part concerns the study of the influence of mineralogy on the cracking process using the parallelized calculation code. A hydro-mechanical coupling model is then developed and applied to the calculation of crack to measure the impact of cracking on the shales permeability.

Résumé

Le travail présenté dans ce document consiste à la réalisation d'outils et de méthodes numériques pour modéliser l'influence de la microstructure des shales à la fois sur les propriétés macroscopiques ainsi que sur le processus de fissuration.

La première partie du document est dédiée à la description d'un modèle Éléments Finis 3D (E-FEM) développé pour représenter la microstructure ainsi que la fissuration. Ce modèle fait parti des méthodes à discontinuités. Deux critères de fissuration sont décrits, un en mode I (critère de Rankine) et un second en mode II (critère de Mohr-Coulomb). Ces critères seront utilisés pour caractériser l'influence de la microstructure des shales sur les propriétés macroscopiques à l'aide d'essai de compression triaxiale. Plus particulièrement l'impact de la présence de kérogène dans la roche de schiste sera étudié.

La seconde partie présente une méthode de décomposition de domaine (la méthode de mortier) utilisée pour réduire les temps de calcul. Cette méthode a pour avantage de permettre l'utilisation de maillage non conforme, ainsi un raffinement local du maillage est possible. Cette méthode a été intégrée à un code de calcul utilisant la programmation orientée composant et plus particulièrement à l'aide du middleware CTL. Le code de calcul permet de résoudre des problèmes linéaires et non linéaires en utilisant le modèle E-FEM.

La dernière partie concerne l'étude de l'influence de la minéralogie sur le processus de fissuration à l'aide du code de calcul parallélisé. Un modèle de couplage hydro-mécanique est ensuite développé et appliqué au calcul de fissuration pour mesurer l'impact de la fissuration sur la perméabilité des shales.

Remerciements

Je tiens, en premier lieu, à remercier le Professeur Fabrice Gatuingt de m'avoir fait l'honneur d'être le président de mon jury de thèse ainsi que les Professeurs Albert Giraud et Thierry Massart d'avoir accepté la lourde tâche d'être les rapporteurs du travail présenté dans ce manuscrit. Enfin je remercie Yun Jia d'avoir accepté d'être examinatrice.

Je remercie mes directeurs de thèse, les professeurs Jian Fu Shao et Jean Baptiste Colliat, pour m'avoir proposé ce sujet de thèse et permis de le mener à bien. Plus particulièrement, je remercie Jean Baptiste pour la bonne ambiance qu'il a su amener dans notre équipe de recherche, pour le cadre de travail qu'il m'a fourni ainsi que pour sa vision du monde de la recherche.

Mes travaux de thèse ont été réalisés avec l'accompagnement de Total. Je tiens à remercier Adeline Garnier qui a effectué le suivi de mes travaux lors de ma première année de thèse. Je remercie aussi MM. Atef Onaisi et Hamid Pourpak qui ont suivi mes travaux durant les dernières années de ma thèse et qui m'ont fait l'honneur d'être présents dans mon jury de thèse en tant qu'examineurs.

Pour finir, je tiens à remercier ma famille ainsi que mes proches. Un grand merci à ma mère, ma belle mère ainsi que ma femme pour avoir effectué la correction de ce manuscrit. Une grande pensée à toutes les personnes qui sont passées par le P3, Mahban, Hamid, Hanaa, Andry, Raphael, Paul et Mei, merci pour tout les bons moments passés ensemble. Encore une fois merci pour tout.

À mon fils, Nathan

Table des matières

Introduction générale	1
1 Modélisation de la fissuration des milieux hétérogènes dans un cadre EF	5
1.1 Introduction	7
1.2 Notion de maillage non adapté	8
1.3 Formulation variationnelle	9
1.3.1 Hu-Washizu [Washizu, 1982]	9
1.3.2 Modes incompatibles [Ibrahimbegovic and Wilson, 1991]	9
1.3.3 Discontinuités faibles	11
1.3.4 Discontinuités fortes	13
1.3.5 Application à un élément volumique	17
1.3.6 Méthode de résolution	19
1.4 Modèle discret sur la discontinuité	22
1.4.1 Critère de Rankine (mode I)	22
1.4.2 Critère de Mohr-Coulomb (mode II)	30
1.5 Étude de l'impact de la minéralogie	35
1.5.1 Génération des différentes morphologies	35
1.5.2 Résultats obtenus	38
1.5.3 Évolution de la fissuration	44
1.6 Étude de l'impact du kérogène	47
1.7 Conclusions et perspectives	55
2 Décomposition de domaine dans un cadre EF non linéaire	57
2.1 Introduction	58
2.2 Méthodes de décomposition de domaine	59
2.2.1 Formulation d'un problème de décomposition de domaine	60
2.2.2 Méthode primale	63
2.2.3 Méthode duale	64
2.2.4 Autres méthodes	65
2.3 Méthode de mortier appliquée à la mécanique	67
2.3.1 Éléments finis de mortier	67
2.3.2 Formulation de la méthode de mortier appliquée à la mécanique	68
2.3.3 Construction des matrices d'interpolation en 3D	70
2.4 Application de la méthode de mortier à un problème de mécanique	72
2.4.1 Algorithme de résolution linéaire	72

2.4.2	Algorithme de résolution non-linéaire	74
2.4.3	Programmation orientée composant	76
2.4.4	Cas d'une poutre en traction en élasticité	84
2.4.5	Cas d'une poutre en traction en non linéaire	86
2.5	Scalabilité du code de calcul	89
2.6	Conclusion	90
3	Modélisation par décomposition de domaine d'un essai de fissuration	92
3.1	Introduction	93
3.2	Essai de fissuration	93
3.2.1	Paramètres matériaux et géométriques	93
3.2.2	Conditions aux limites	94
3.2.3	Résultats mécaniques	95
3.3	Couplage Hydro-mécanique	97
3.3.1	Formulation du problème hydraulique avec un fluide incompressible dans le cadre de la méthode E-FEM	97
3.3.2	Couplage entre le calcul mécanique et le calcul hydrique	102
3.4	Calcul de fissuration avec le couplage mécanique-hydraulique faible	109
3.5	Conclusion	117
	Conclusion	118
	A Minéralogies étudiées	121
	B Composant mortier	123
	Bibliographie	128

Introduction générale

Le travail présenté dans ce manuscrit porte sur la réalisation d'outils et de méthodes numériques afin de modéliser l'influence de la microstructure des roches de schiste sur les propriétés macroscopiques ainsi que sur le processus de fracturation. Ce travail a été réalisé avec la collaboration du groupe TOTAL.

Les shales sont des roches ayant un aspect feuilleté. Ces dernières peuvent être des roches sédimentaires argileuses ou métamorphiques. Seul le premier type intéresse les ingénieurs pétroliers et gaziers. En effet, seules ces roches peuvent contenir une part significative d'hydrocarbures car la température et la pression y sont élevées lors de leur formation. Cette propriété entraîne la transformation des roches en molécules d'hydrocarbures. Ces gaz d'hydrocarbures sont appelés gaz de shale.

L'extraction des hydrocarbures contenus dans les roches dites "roches mères" est un enjeu important pour les industriels pétroliers. Ces gaz sont situés dans des réservoirs d'hydrocarbures non-conventionnels qui sont peu perméables. Les méthodes classiques d'extraction de gaz ne peuvent être utilisées dans ce contexte. En effet, il est nécessaire d'avoir recours à un forage horizontal afin d'augmenter la surface de contact avec le réservoir. De plus, pour augmenter artificiellement la perméabilité de la roche et permettre une extraction plus rapide du gaz, une fracturation hydraulique est réalisée. Cette fracturation consiste en l'injection d'une grande quantité de fluide sous pression qui est composé principalement d'eau et d'adjuvants chimiques permettant un contrôle de la viscosité de celui-ci. En fin d'injection des propants sont injectés en plus du liquide pour éviter la fermeture des fissures. Cette fracturation est réalisée à différents niveaux du puits horizontal dans le but d'obtenir la plus grande zone fracturée possible. La réalisation d'un forage horizontal à une grande profondeur ainsi que les multiples fracturations hydrauliques représentent un coût important. Il est nécessaire de bien connaître le comportement mécanique de la roche ainsi que son processus de fissuration afin de maximiser les performances du puits. Cette technique est résumée sur le schéma 1 où l'on peut observer la mise en place du forage horizontal, les différents points d'injections du fluide et la création de fissure permettent l'augmentation artificielle de la perméabilité de la roche.

Le travail présenté ici s'inscrit dans le cadre d'un projet de compréhension du comportement des shales à différentes échelles. L'étude a été divisée sur trois niveaux d'échelles. La première échelle, celle du réservoir, correspond à une échelle kilométrique. Cette étude est réalisée par le laboratoire "Sols, Solides, Structures - Risques" (3SR) de Grenoble. La seconde étude concerne la fissuration au niveau du joint et est effectuée à une échelle

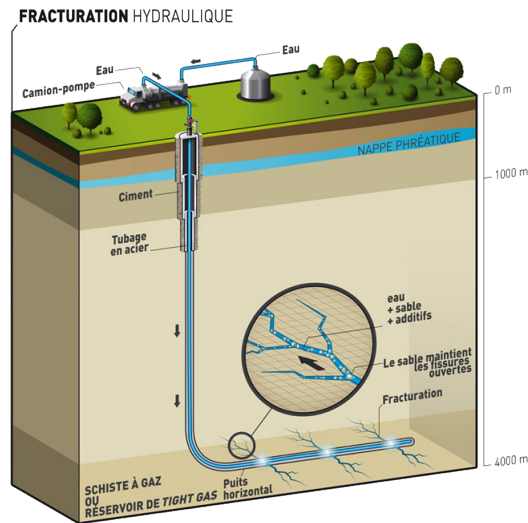


FIGURE 1 – Technique de production d’un réservoir non conventionnel, TOTAL

métrique. Le Laboratoire des Fluides Complexes et leurs Réservoirs (LFCR) situé à Pau effectue cette dernière. Enfin, la troisième échelle, la plus petite, correspond à l’étude de l’impact de la minéralogie sur la fissuration, soit une échelle micrométrique. Notre équipe, nommée Couplages thermo-hydomécanique et chimiques du Laboratoire de Mécanique de Lille, se charge d’étudier le comportement mécanique à l’échelle micrométrique.

L’objectif principal du projet est de créer une chaîne de modélisation entre les différentes échelles. Cette chaîne est de type séquencée, les informations microscopiques d’une échelle venant des résultats macroscopiques de l’échelle précédente. En ce qui concerne notre échelle les données microscopiques viennent de la connaissance du matériau du groupe TOTAL.

Dans le contexte de l’exploitation du gaz de schiste, la roche du réservoir est soumise à des sollicitations hydromécaniques et chimiques couplées. Afin d’étudier l’intégrité des puits et du réservoir puis d’améliorer les techniques d’exploitation, il est nécessaire de mener des modélisations numériques en tenant compte des différents mécanismes de fissuration et de rupture. Le phénomène de fissuration des matériaux quasi-fragiles est complexe, plusieurs méthodes permettent de modéliser ce dernier. Les approches continues comme les modèles d’endommagements ([Mazars and Pijaudier Cabot, 1989],[Pijaudier-Cabot and Bazant, 1987]) sont de types phénoménologiques à fissuration répartie. Ces modèles reposent sur des quantités macroscopiques telles que les contraintes ou les déformations, elles-mêmes définies au niveau d’un Volume Élémentaire Représentatif (V.E.R), pour lequel le matériau, par essence de nature hétérogène, est supposé homogène. Les lois utilisées découlent alors d’un cadre thermodynamique et permettent de rendre compte des phénomènes observés à l’échelle macroscopique, en laboratoire ou in situ.

Une approche plus récente, les modèles à fissuration discrète, consiste en l’utilisation de discontinuités fortes (discontinuités dans le champs de déplacements) afin de représenter les fissures.

Ces méthodes répondent à deux problématiques fondamentales en EF :

- le problème de la localisation numérique, bien connu pour les modèles à fissuration répartie en conjonction avec des lois adoucissantes [Richard et al., 2010] : en dissipant l'énergie sur la surface de rupture (et non dans le volume), les modèles à fissuration discrète permettent de conserver l'objectivité par rapport au maillage
- la question de l'ouverture ainsi que de l'orientation des fissures est fondamentale pour le couplage avec les transferts de masse.

Cette thèse porte en partie sur le développement de cette approche.

Concernant la modélisation du couplage fissuration/transfert, les modèles phénoménologiques permettent de réaliser des calculs de transfert dans le cas où la structure est peu endommagée. Dans ce cas il est possible de prendre comme hypothèse que les fissures sont isotropes et équi-réparties. Il reste alors à déterminer un lien entre la valeur de l'endommagement D et la valeur de la perméabilité K . Ce lien peut être établi de manière expérimentale [Jason, 2004] ou analytique [Kondo and Dormieux, 2003]. Cependant, dans le cas où la structure est fortement endommagée, lorsque une ou plusieurs macro-fissures apparaissent, les hypothèses d'équi-répartition et d'isotropie ne sont plus vraies. Il n'est donc plus possible de lier directement l'endommagement et la perméabilité. Par conséquent il est nécessaire d'utiliser une fissuration discrète pour pouvoir mener à bien le calcul de transfert.

Dans un contexte industriel, il est important de pouvoir effectuer les simulations dans un temps relativement court. Afin de répondre à ce problème, les méthodes de décomposition de domaine permettent, en découpant le domaine d'étude en plusieurs sous-domaines, d'effectuer une partie du calcul en parallèle et ainsi de diminuer le temps de celui-ci. Parmi ces méthodes, deux approches peuvent être distinguées. Les approches avec chevauchement des sous-domaines "overlapping decomposition methods" ([Le Tallec, 1994],[Farhat and Roux, 1991]) et les méthodes sans chevauchement "non-overlapping decomposition methods" ([Schwarz, 1869],[Rateau, 2003],[Lions, 1988]). Dans le cadre de l'étude présentée dans ce manuscrit, l'utilisation de maillage avec chevauchement n'a pas d'utilité, seul l'approche sans chevauchement sera étudiée. Parmi cette approche, de nombreuses méthodes sont apparues au fil du temps. La méthode employée, développée par Yvon Maday [Maday et al., 1989], est nommée méthode de mortier. Un des avantages majeurs de cette méthode est la possibilité d'utiliser des maillages non conformes aux interfaces. Cela permet de raffiner localement un sous-domaine et de faciliter la génération des maillages. De plus cette méthode permet de réaliser des calculs avec propagation de fissure, ce qui est un élément essentiel de notre étude.

Le premier chapitre de cette thèse sera composé d'une description du cadre théorique de la méthode de fissuration discrète dans le cadre d'éléments tétraédriques. Deux critères de fissuration (critère de Rankine en mode I et critère de Mohr-Coulom II) seront décrits ensuite. Des exemples simples montreront les limites du premier critère concernant la modélisation du comportement macroscopique et justifieront la création du second critère. Enfin, la dernière partie du chapitre portera sur l'étude de l'impact de la minéralogie sur le comportement mécanique à l'aide de la méthode à enrichissement cinématique. Dans cette partie le comportement macroscopique d'un VER sera étudié dans le cas où celui ci

est soumis à une sollicitation de compression triaxiale. L'étude de la réponse mécanique ainsi que le faciès de fissuration sera effectuée lors de cette sollicitation.

Le second chapitre décrira le cadre théorique des méthodes de décomposition de domaine et plus particulièrement de la méthode de mortier. L'algorithme de résolution sera ensuite présenté ainsi que son implémentation numérique à l'aide de la programmation orientée composant et plus précisément à l'aide du middleware CTL [Niekamp and Stein, 2002]. Ensuite deux exemples de calculs seront présentés dans le cas d'un calcul linéaire et d'un calcul non linéaire avec l'utilisation de la décomposition de domaine. Ces exemples permettront de valider le choix de la méthode. Enfin, une étude de la performance de l'outil crée sera effectuée afin de vérifier les capacités de l'outil.

Le dernier chapitre sera consacré à l'étude de la fissuration d'une plaque entaillée. La méthode de couplage hydro-mécanique sera ensuite présentée avec plusieurs exemples simples. Pour finir un exemple de couplage hydro-mécanique faible avec la méthode de décomposition de domaine sera proposé.

La réalisation du travail présenté dans ce manuscrit s'est effectuée en quatre étapes. La première étape consistait à la compréhension de la problématique et à une réflexion concernant les outils à développer pendant cette thèse, permettant de répondre au mieux à cette problématique. La seconde étape concernait la réalisation de la bibliographie dans le but de se familiariser avec les méthodes utiles lors de cette thèse et permettant de répondre aux différentes problématiques. Deux méthodes étaient nécessaires à cette étude. La première était la méthode des éléments finis avec enrichissement cinématique (E-FEM) et la seconde, la méthode de décomposition de domaine. La troisième étape avait pour objet la conception des outils numériques permettant de répondre aux problématiques. Elle représente la partie la plus importante de cette thèse aussi bien en terme de temps qu'en terme de difficultés rencontrées. La dernière étape mettait en œuvre l'exploitation des outils créés au cours de cette thèse.

Chapitre 1

Modélisation de la fissuration des milieux hétérogènes dans un cadre EF

Sommaire

1.1	Introduction	7
1.2	Notion de maillage non adapté	8
1.3	Formulation variationnelle	9
1.3.1	Hu-Washizu [Washizu, 1982]	9
1.3.2	Modes incompatibles [Ibrahimbegovic and Wilson, 1991]	9
1.3.3	Discontinuités faibles	11
1.3.3.1	Nombre d'inconnues	12
1.3.3.2	Forme du champ de déformation :	13
1.3.4	Discontinuités fortes	13
1.3.4.1	Nombre d'inconnues	14
1.3.4.2	Enrichissement du champs standard de déformation	15
1.3.4.3	Enrichissement du champs virtuel de déformation	16
1.3.5	Application à un élément volumique	17
1.3.6	Méthode de résolution	19
1.4	Modèle discret sur la discontinuité	22
1.4.1	Critère de Rankine (mode I)	22
1.4.1.1	Critère de localisation	22
1.4.1.2	Critère d'ouverture	22
1.4.1.3	Comportement d'un matériau à matrice argileuse avec un critère de Rankine	23
1.4.2	Critère de Mohr-Coulomb (mode II)	30
1.4.2.1	Critère de localisation	30
1.4.2.2	Critère d'ouverture	31
1.4.2.3	Comportement d'un matériau à matrice argileuse avec le critère de Mohr-Coulomb	32
1.5	Étude de l'impact de la minéralogie	35
1.5.1	Génération des différentes morphologies	35

1.5.1.1	Placement aléatoire de sphères polydisperses	35
1.5.1.2	Projection des sphères sur un maillage	37
1.5.2	Résultats obtenus	38
1.5.3	Évolution de la fissuration	44
1.6	Étude de l'impact du kérogène	47
1.7	Conclusions et perspectives	55

1.1 Introduction

Les hydrocarbures de roches mères sont présents au sein de formations géologiques argileuses. Ces formations sont peu perméables et riches en matière organique. Du fait de la faible perméabilité de leur lieu de stockage, ces hydrocarbures stockés, dits non conventionnels, ont longtemps été inexploités. Des méthodes plus complexes, telles que l'utilisation de forage horizontal ainsi que de la mise en place de la fracturation hydraulique, ont permis depuis quelques années d'exploiter ces hydrocarbures.

La fracturation hydraulique consiste, à l'aide d'une contrainte mécanique imposée par l'injection d'un fluide sous hautes pressions, à fissurer la roche et donc à augmenter la perméabilité de celle-ci. Une bonne connaissance des paramètres ayant une influence sur le processus de fissuration est primordiale.

Parmi les méthodes de modélisation du processus de fissuration nous avons choisi d'utiliser les méthodes à enrichissement cinématique. Ces méthodes ont l'avantage de représenter directement la fissuration à l'aide d'une discontinuité dans le champ de déplacement.

Il existe plusieurs méthodes utilisant l'enrichissement cinématique pour représenter la fissuration. On peut citer la méthode X-FEM où l'enrichissement est fait au niveau des nœuds des éléments coupés. Ceci a pour conséquence d'augmenter la taille globale du système et donc le temps de résolution et devient rapidement problématique pour les problèmes de multi-fissurations. De plus l'implémentation de ce type de méthode est fortement intrusive. La méthode E-FEM, qui est utilisée dans le cadre de cette thèse, a un enrichissement local. Ainsi le nombre de degrés de liberté globale est inchangé et l'implémentation de ce type de méthode n'est pas intrusive (la forme du système global à résoudre est la même). Cette méthode est donc indiquée pour les cas de multi-fissurations.

Dans un cadre éléments finis classique, les éléments du maillage ne prennent en compte qu'un seul matériau à la fois. Afin de modéliser la présence d'hétérogénéités celles-ci doivent être maillées directement. Le maillage se fait alors en deux étapes. Une première étape consiste à mailler les frontières des différentes hétérogénéités. Une fois cette étape réalisée, la seconde réside au maillage des domaines définis par ces frontières. La mise en place de ce type de maillage est lente et peut rencontrer des difficultés lorsque deux hétérogénéités sont trop proches (les éléments maillés entre les deux hétérogénéités sont alors beaucoup plus petits que les autres éléments du maillage). Ce dernier inconvénient peut poser des problèmes de conditionnement de la matrice. De plus, un changement même minime dans le placement des hétérogénéités impose un remaillage total de la structure. La méthodologie développée dans cette thèse consiste à ne pas mailler directement les hétérogénéités et à permettre aux éléments finis d'être bi-phasiques. La présence de deux matériaux dans un éléments finis impose une discontinuité de la déformation. En enrichissant nos éléments d'une discontinuité dans le champ de déformation (appelée discontinuité faible), nos éléments finis peuvent être bi-phasiques.

Du point de vue du champ de déplacement une fissure est vue comme une discontinuité. En enrichissant nos éléments d'une discontinuité dans le champ de déplacement (appelée discontinuité faible), ceux-ci peuvent représenter le phénomène de fissuration. La fissure est alors modélisée par un saut de déplacement le long d'une surface de discontinuité. La

valeur du saut de déplacement représente alors l'ouverture de la fissure. Lorsque la discontinuité faible est activée il est assumé que la surface de discontinuité forte sera égale à la surface de discontinuité faible ($\Gamma_u = \Gamma_\epsilon$).

Dans ce chapitre nous détaillerons dans un premier temps la formulation de la méthode E-FEM dans le cas d'un tétraèdre. La seconde partie consistera à la description de deux critères de rupture (critère de Rankine et critère de Mohr-Coulomb) ainsi qu'à la présentation de résultats obtenus. Enfin la dernière partie consistera en l'étude numérique de l'impact de la minéralogie sur le comportement en compression triaxiale avec la méthode E-FEM.

1.2 Notion de maillage non adapté

Pour modéliser des matériaux hétérogènes deux méthodes peuvent être utilisées. La première consiste tout d'abord à mailler les surfaces correspondantes aux interfaces des différentes phases puis à mailler les domaines homogènes inclus dans ces surfaces fermées. Un des problèmes de cette méthode est qu'il est difficile de prendre en compte le comportement de l'interface de manière simple, il faut en effet ajouter des éléments à surface nulle au niveau des surfaces des interfaces. De plus il faut recréer un nouveau maillage lorsque l'on change les données géométriques or cette méthode de maillage n'est pas rapide.

La méthode utilisée ici consiste à utiliser des maillages non adaptés. Un des avantages de cette méthode est que le positionnement des nœuds est indépendant de la position des interfaces. Il n'est donc pas nécessaire de recréer un nouveau maillage lorsque la position des interfaces change, ce qui peut être le cas lors d'analyses de l'impact de la minéralogie sur le comportement. La prise en compte de l'hétérogénéité du matériau est possible grâce aux éléments finis bi-phasiques qui peuvent présenter un saut fini au niveau de leur champ de déformation (discontinuité faible). Il suffit alors de déterminer quels éléments sont coupés par une interface (figure 1.1) et à quel niveau afin de pouvoir modéliser le caractère hétérogène.

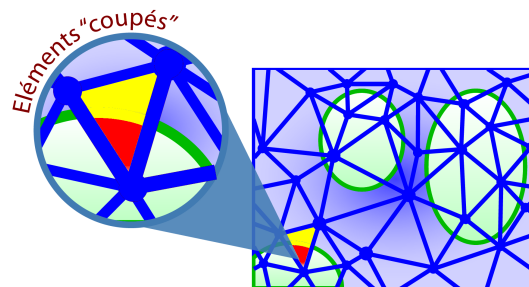


FIGURE 1.1 – Schéma de l'ajout d'inclusions dans une matrice : éléments surfaciques coupés par une interface physique [Colliat, 2010]

1.3 Formulation variationnelle

1.3.1 Hu-Washizu [Washizu, 1982]

L'idée principale de la formulation Hu-Washizu est que, contrairement à des formulations classiques de déplacement, les trois champs (u, ε, σ) , respectivement champ de déplacement, champ de déformation et champ de contrainte, sont indépendants. Correspondant à chacun d'eux, nous appelons (w, γ, τ) respectivement le champ de déplacement virtuel, le champ de déformation virtuel et le champ de contrainte virtuel. Elle conduit à la nécessité d'incorporer de nouvelles équations dans la formulation. Nous avons maintenant dans la formulation trois équations :

— **l'équation d'équilibre classique** (ou travail virtuel) (déplacement virtuel)

$$\text{HW}_u(u, \varepsilon, \sigma; w) = \int_{\Omega} \nabla^s w : \sigma \, d\Omega - \int_{\Omega} w \cdot b \, d\Omega - \int_{\Gamma_t} w \cdot \bar{t} \, d\Gamma = 0 \quad (1.3.1)$$

— **l'équation cinématique** (contrainte virtuel)

$$\text{HW}_{\sigma}(u, \varepsilon, \sigma; \tau) = \int_{\Omega} \tau : (\nabla^s u - \varepsilon) \, d\Omega = 0 \quad (1.3.2)$$

— et **l'équation de comportement** (ou modèle constitutif) (contrainte virtuelle)

$$\text{HW}_{\varepsilon}(u, \varepsilon, \sigma; \gamma) = \int_{\Omega} \gamma : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega = 0 \quad (1.3.3)$$

où $\hat{\sigma}$ est un champ de contrainte qui vérifie la loi de comportement.

Habituellement, dans une formulation classique de déplacement, nous avons $\begin{cases} \varepsilon \triangleq \nabla^s u \\ \sigma \triangleq \hat{\sigma}(\varepsilon) \end{cases}$

ce qui rend (1.3.2) et (1.3.3) non pertinents. b représente les forces volumiques et \bar{t} les forces de surface.

1.3.2 Modes incompatibles [Ibrahimbegovic and Wilson, 1991]

Le champ de déformation est décomposé en trois parties. Une pour le champ naturel $\underline{\underline{\varepsilon}}$ et deux pour les deux champs enrichis (faible $\underline{\underline{\tilde{\varepsilon}}}$ et fort $\underline{\underline{\hat{\varepsilon}}}$) :

$$\underline{\underline{\varepsilon}} = \underbrace{\underline{\underline{\nabla^s u}}}_{\text{compatible}} + \underbrace{\underline{\underline{\tilde{\varepsilon}}} + \underline{\underline{\hat{\varepsilon}}}}_{\text{incompatible}} \quad (1.3.4)$$

Le premier principe de la méthode des modes incompatibles est d'enrichir aussi le champ de déformation virtuelle :

$$\begin{aligned} \underline{\underline{\varepsilon}} &= \underline{\underline{\nabla^s u}} + \underline{\underline{\tilde{\varepsilon}}} + \underline{\underline{\hat{\varepsilon}}} \\ \underline{\underline{\gamma}} &= \underline{\underline{\nabla^s w}} + \underline{\underline{\tilde{\gamma}}} + \underline{\underline{\hat{\gamma}}} \end{aligned} \quad (1.3.5)$$

La formulation de Hu-Washizu devient alors :

$$\boxed{\begin{cases} \text{HW}_u(u, \varepsilon, \sigma; w) &= \int_{\Omega} \nabla^s w : \sigma \, d\Omega - \int_{\Omega} w \cdot b \, d\Omega - \int_{\Gamma_t} w \cdot \bar{t} \, d\Gamma &= 0 \\ \text{HW}_{\sigma}(u, \varepsilon, \sigma; \tau) &= \int_{\Omega} \tau : (\underline{\underline{\hat{\varepsilon}}} + \underline{\underline{\hat{\varepsilon}}}) \, d\Omega &= 0 \\ \text{HW}_{\varepsilon}(u, \varepsilon, \sigma; \gamma) &= \int_{\Omega} \underline{\underline{\nabla}}^s w : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega + \int_{\Omega} (\underline{\underline{\tilde{\gamma}}} + \underline{\underline{\hat{\gamma}}}) : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega &= 0 \end{cases}} \quad (1.3.6)$$

Les déformations enrichies étant indépendantes des déformations naturellement présentes dans l'élément enrichi, les variables $(\underline{\underline{\nabla}}^s w, \underline{\underline{\tilde{\gamma}}}, \underline{\underline{\hat{\gamma}}})$ sont donc indépendantes. On peut donc séparer les équations et réécrire (1.3.6) :

$$\boxed{\begin{cases} \int_{\Omega} \nabla^s w : \sigma(\varepsilon) \, d\Omega - \int_{\Omega} w \cdot b \, d\Omega - \int_{\Gamma_t} w \cdot \bar{t} \, d\Gamma &= 0 \\ \int_{\Omega} \tau : \underline{\underline{\hat{\varepsilon}}} \, d\Omega &= 0 \\ \int_{\Omega} \tau : \underline{\underline{\hat{\varepsilon}}} \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\nabla}}^s w : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\tilde{\gamma}}} : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\hat{\gamma}}} : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega &= 0 \end{cases}} \quad (1.3.7)$$

Afin de simplifier la formulation discrétisée on choisira des champs de contraintes et de déformations enrichies construits orthogonalement, nous avons alors les relations suivantes :

$$\begin{aligned} \int_{\Omega} \tau : \underline{\underline{\hat{\varepsilon}}} \, d\Omega &= 0 \\ \int_{\Omega} \tau : \underline{\underline{\hat{\varepsilon}}} \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\tilde{\gamma}}} : \sigma \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\hat{\gamma}}} : \sigma \, d\Omega &= 0 \end{aligned} \quad (1.3.8)$$

En injectant ces équations dans le système d'équations 1.3.7 le système se simplifie et devient :

$$\boxed{\begin{cases} \int_{\Omega} \nabla^s w : \sigma(\varepsilon) \, d\Omega - \int_{\Omega} w \cdot b \, d\Omega - \int_{\Gamma_t} w \cdot \bar{t} \, d\Gamma &= 0 \\ \int_{\Omega} \underline{\underline{\nabla}}^s w : (\hat{\sigma}(\varepsilon) - \sigma) \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\tilde{\gamma}}} : \hat{\sigma}(\varepsilon) \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\hat{\gamma}}} : \hat{\sigma}(\varepsilon) \, d\Omega &= 0 \end{cases}} \quad (1.3.9)$$

En supposant maintenant que le champ de contrainte vérifie la loi de comportement de façon forte :

$$\hat{\sigma}(\varepsilon) - \sigma \quad (1.3.10)$$

Le système 1.3.9 devient alors :

$$\boxed{\begin{cases} \int_{\Omega} \nabla^s w : \sigma(\varepsilon) \, d\Omega - \int_{\Omega} w \cdot b \, d\Omega - \int_{\Gamma_t} w \cdot \bar{t} \, d\Gamma &= 0 \\ \int_{\Omega} \underline{\underline{\tilde{\gamma}}} : \hat{\sigma}(\varepsilon) \, d\Omega &= 0 \\ \int_{\Omega} \underline{\underline{\hat{\gamma}}} : \hat{\sigma}(\varepsilon) \, d\Omega &= 0 \end{cases}} \quad (1.3.11)$$

Pour que la convergence de la méthode soit assurée, le patch test doit être vérifié. Pour cela on teste l'élément enrichi avec une contrainte constante σ_c , les déformations virtuelles enrichies doivent vérifier les relations suivantes :

$$\begin{aligned}\sigma_c \int_{\Omega} \underline{\underline{\hat{\gamma}}} d\Omega &= 0 \\ \sigma_c \int_{\Omega} \underline{\underline{\hat{\gamma}}} d\Omega &= 0\end{aligned}\tag{1.3.12}$$

Les déformations virtuelles enrichies doivent donc vérifier la condition de moyenne nulle pour que le patch test soit validé.

On discrétise ensuite le système avec la méthode des éléments finis, soit pour les déformations virtuelles et standards :

$$\begin{aligned}\varepsilon &= Bd + G_w a + G_s b \\ \gamma &= Bd + \hat{G}_w a + \hat{G}_s b\end{aligned}\tag{1.3.13}$$

Avec B la matrice standard d'interpolation, G_w la matrice des termes d'enrichissements faibles pour les déformations réelles, G_s la matrice des termes d'enrichissements forts pour les déformations réelles, \hat{G}_w la matrice des termes d'enrichissements faibles pour les déformations virtuelles et \hat{G}_s la matrice des termes d'enrichissements forts pour les déformations virtuelles. Les vecteurs contenant les degrés de liberté sont notés a et b .

En remplaçant ces valeurs dans 1.3.11 :

$$\left\{ \begin{array}{l} \int_{\Omega} B^T \hat{\sigma}(Bd + G_w a + G_s b) d\Omega = f_{ext} \\ \int_{\Omega} \hat{G}_w \hat{\sigma}(Bd + G_w a + G_s b) d\Omega = 0 \\ \int_{\Omega} \hat{G}_s \hat{\sigma}(Bd + G_w a + G_s b) d\Omega = 0 \end{array} \right.\tag{1.3.14}$$

En considérant maintenant la discrétisation complète avec la méthode éléments finis, la première ligne de l'équation 1.3.14 peut être réécrite sous une forme plus classique :

$$\begin{array}{l} A_{e=1}^{n_{elm}} [f_{int}^e - f_{ext}^e] = 0 \\ \int_{\Omega} \hat{G}_w \hat{\sigma}(Bd + G_w a + G_s b) d\Omega = 0 \\ \int_{\Omega} \hat{G}_s \hat{\sigma}(Bd + G_w a + G_s b) d\Omega = 0 \\ \text{ou} \\ f_{int}^e = \int_{\Omega} B^T \hat{\sigma}(Bd + G_w a + G_s b) d\Omega \end{array}\tag{1.3.15}$$

Ce système d'équation correspond à l'équation d'équilibre global classique pour la première équation et aux équations ajoutées par la présence des deux champs d'enrichissements pour les deux autres équations. Dans la partie suivante nous allons décrire le calcul des inconnues a et b pour les deux types d'enrichissements ainsi que la construction des matrices d'enrichissement dans le cadre de l'utilisation d'un élément volumique tétraédrique.

1.3.3 Discontinuités faibles

Dans les années soixante-dix les modèles développés pour simuler la propagation des fissures (reposant sur la mécanique non linéaire de la rupture ou sur la mécanique continue de l'endommagement) ont en général négligé l'impact que pouvait voir les hétérogénéités.

Les matériaux étudiés étaient alors homogénéisés. Ce type de modélisation ne permettant pas de caractériser de manière complète à une échelle fine le processus de fissuration dans les matériaux fortement hétérogènes comme les shales. C'est pourquoi différents modèles ont été développés plus récemment pour pouvoir modéliser cette échelle.

Nous avons choisi ici d'utiliser les modèles à discontinuité faible pour pouvoir prendre en compte les hétérogénéités à une échelle fine. L'utilisation de maillages non adaptés implique que les éléments puissent être coupés en deux et que ces deux parties aient des modules Young différents. Le fait d'avoir une discontinuité dans le champ de déformation permet d'avoir un élément avec deux modules Young différents. Sur la figure 1.2 on peut voir l'exemple d'un tétraèdre coupé en deux par une interface, sur la figure 1.2.b la partie haute du tétraèdre a une déformation plus importante due à un module Young plus faible dans la partie haute.

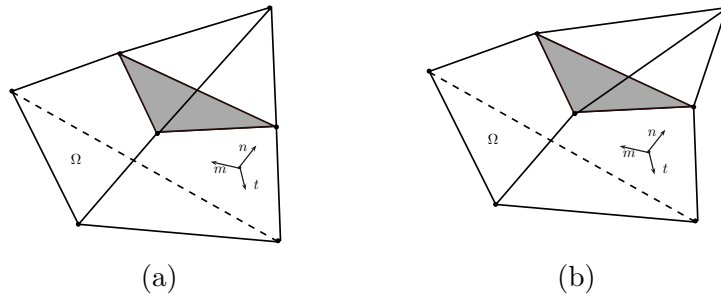


FIGURE 1.2 – Représentation d'une discontinuité faible dans un tétraèdre

1.3.3.1 Nombre d'inconnues

La condition que nous avons à l'interface est la suivante :

$$u_{\Gamma}^{\oplus} = u_{\Gamma}^{\ominus} \quad (1.3.16)$$

Nous avons une discontinuité du champ de déformation sur l'interface Γ en raison d'une discontinuité des paramètres matériaux.

$$\underline{\underline{\underline{\varepsilon}}}^{\oplus} - \underline{\underline{\underline{\varepsilon}}}^{\ominus} = \left[\left[\underline{\underline{\underline{\varepsilon}}} \right] \right] \quad (1.3.17)$$

Nous devons trouver la forme de $\left[\left[\underline{\underline{\underline{\varepsilon}}} \right] \right]$ (noté a précédemment). En notant n la normale à l'interface et m et t les vecteurs orthogonaux au vecteur n qui forme une base orthonormée on peut écrire :

$$\underline{\underline{\underline{\nabla}}}(u) \Big|_{\Gamma} = \left(\begin{array}{ccc} \frac{\partial u_1}{\partial n} & \frac{\partial u_1}{\partial m} & \frac{\partial u_1}{\partial t} \\ \frac{\partial u_2}{\partial n} & \frac{\partial u_2}{\partial m} & \frac{\partial u_2}{\partial t} \\ \frac{\partial u_3}{\partial n} & \frac{\partial u_3}{\partial m} & \frac{\partial u_3}{\partial t} \end{array} \right) \Big|_{(n,m,t)_{\Gamma}} \quad (1.3.18)$$

Si nous essayons d'exprimer $\left[\left[\underline{\underline{\underline{\varepsilon}}} \right] \right]$, nous pouvons voir que tous les termes en gras vont être égaux pour Ω_e^{\oplus} et Ω_e^{\ominus} , la discontinuité étant normale à l'interface. Il en découle

que, suivant les directions perpendiculaires à la direction de l'interface, il n'y a pas de discontinuité sur le champ de déformation.

$$[[\underline{\underline{\varepsilon}}]] = \underline{\underline{\varepsilon}}^{\oplus} - \underline{\underline{\varepsilon}}^{\ominus} = \underline{\underline{\nabla}}^s \underline{\underline{u}}^{\oplus} - \underline{\underline{\nabla}}^s \underline{\underline{u}}^{\ominus} = \begin{pmatrix} [\varepsilon]_n & [\varepsilon]_m & [\varepsilon]_t \\ [\varepsilon]_m & 0 & 0 \\ [\varepsilon]_t & 0 & 0 \end{pmatrix} \quad (1.3.19)$$

Notons :

$$[[\underline{\underline{\varepsilon}}]] = \begin{pmatrix} [\varepsilon]_n \\ [\varepsilon]_m \\ [\varepsilon]_t \end{pmatrix} \quad (1.3.20)$$

1.3.3.2 Forme du champ de déformation :

On sait que $\underline{\underline{\varepsilon}}$ a besoin de 3 inconnues pour être complètement représenté. Nous pouvons trouver sa forme en prenant un champ de déplacement enrichi de la forme :

$$\tilde{\underline{\underline{u}}} = M(\underline{\underline{x}}) ([\varepsilon]_n \underline{\underline{n}} + [\varepsilon]_m \underline{\underline{m}} + [\varepsilon]_t \underline{\underline{t}}) \quad (1.3.21)$$

Avec la fonction $M(\underline{\underline{x}})$ qui vaut :

$$M(\underline{\underline{x}}) = \Theta^{\oplus|\ominus} \underline{\underline{n}} \cdot (\underline{\underline{x}} - \underline{\underline{\xi}}) \quad (1.3.22)$$

$$\Theta^{\oplus|\ominus} \triangleq \begin{cases} \Theta^{\oplus} & \text{si } \underline{\underline{n}} \cdot (\underline{\underline{x}} - \underline{\underline{\xi}}) \geq 0, \\ \Theta^{\ominus} & \text{sinon.} \end{cases} \quad (1.3.23)$$

On obtient alors le champs de déformation enrichi faiblement :

$$\underline{\underline{\varepsilon}} = \underline{\underline{\nabla}}^s(\tilde{\underline{\underline{u}}}) = \Theta^{\oplus|\ominus} \left([\varepsilon]_n \underline{\underline{n}} \otimes \underline{\underline{n}} + \frac{[\varepsilon]_m}{2} (\underline{\underline{n}} \otimes \underline{\underline{m}} + \underline{\underline{m}} \otimes \underline{\underline{n}}) + \frac{[\varepsilon]_t}{2} (\underline{\underline{n}} \otimes \underline{\underline{t}} + \underline{\underline{t}} \otimes \underline{\underline{n}}) \right) \quad (1.3.24)$$

La fonction $\Theta^{\oplus|\ominus}$ doit être choisi tel que $\underline{\underline{\varepsilon}}$ respecte la condition de moyenne nulle, soit :

$$\int_{\Omega} \underline{\underline{\varepsilon}} dV = \underline{\underline{0}} \quad (1.3.25)$$

1.3.4 Discontinuités fortes

Une discontinuité forte, correspondant à une discontinuité locale du champ de déplacement sur une ligne (en deux dimensions) et sur une surface (en trois dimensions), permet par définition de représenter une fissuration. La fissure est alors représentée dans chaque élément par une valeur d'ouverture et une orientation.

Sur la figure 1.3 nous pouvons voir l'exemple d'un tétraèdre sans fissuration (figure 1.3.a), d'un tétraèdre fissurant en mode I (figure 1.3.b) et enfin d'une fissuration en mode II (figure 1.3.c). Pour les cas (1.3.b et 1.3.c) la fissure est définie par l'orientation du plan de fissuration ainsi que par l'ouverture $||u||$.

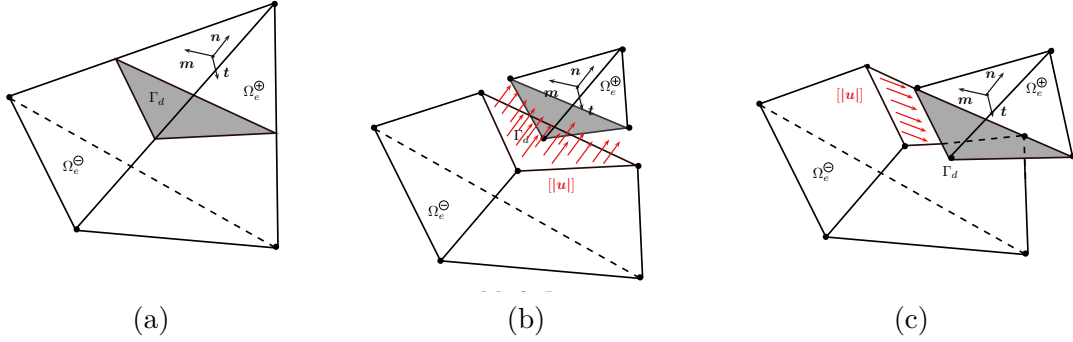


FIGURE 1.3 – Représentation d’une discontinuité forte dans un tétraèdre, (a) sans discontinuité (b) mode I et (c) mode II

1.3.4.1 Nombre d’inconnues

Nous avons ici une discontinuité dans le champ de déplacement. Elle conduit à un champ de déformation non borné $\hat{\underline{\underline{\epsilon}}}$ qui peut être représenté comme suit, avec une partie bornée $\hat{\underline{\underline{\epsilon}}}_b$ et une partie non bornée $\hat{\underline{\underline{\epsilon}}}_u$:

$$\hat{\underline{\underline{\epsilon}}} = \underbrace{\hat{\underline{\underline{\epsilon}}}_b}_{\text{régulière}} + \underbrace{\hat{\underline{\underline{\epsilon}}}_u}_{\text{singulière}} \quad (1.3.26)$$

Le champ de déplacement sous-jacent sera également coupé en deux parties :

$$\hat{\underline{u}} = \hat{\underline{u}}_b + \hat{\underline{u}}_u = \hat{\underline{u}}_b + \mathcal{H}_\Gamma [[\underline{u}]] \quad (1.3.27)$$

Où $[[\underline{u}]]$ représente le saut de déplacement au niveau de la discontinuité, le saut est défini par les trois composantes de son vecteur :

$$[[\underline{u}]] = \begin{pmatrix} [u]_n \\ [u]_m \\ [u]_t \end{pmatrix}$$

La fonction \mathcal{H} correspond à la fonction heaviside définie par :

$$\mathcal{H}_\Gamma = \begin{cases} 1 & \text{if } \underline{n} \cdot (\underline{x} - \underline{\xi}) \geq 0, \\ 0 & \text{else.} \end{cases} \quad (1.3.28)$$

On peut voir sur la figure 1.4 la représentation d’une discontinuité forte avec à gauche la décomposition des fonctions d’enrichissement et à droite la décomposition du champ de déplacement. La déformation due à la discontinuité est décomposée en une partie régulière (bornée) et une partie singulière (dite illimitée) :

$$\hat{\underline{\underline{\epsilon}}} = \underline{\nabla}^s \hat{\underline{u}}_b + \underline{\nabla}^s \hat{\underline{u}}_u = \underbrace{\underline{\nabla}^s \hat{\underline{u}}_b}_{\text{régulière}} + \underbrace{\delta_S ([[u]] \otimes \underline{n})^s}_{\text{singulière}} \quad (1.3.29)$$

Avec :

$$\delta_\Gamma = \begin{cases} \infty & \text{if } \underline{n} \cdot (\underline{x} - \underline{\xi}) = 0, \\ 0 & \text{else.} \end{cases} \quad (1.3.30)$$

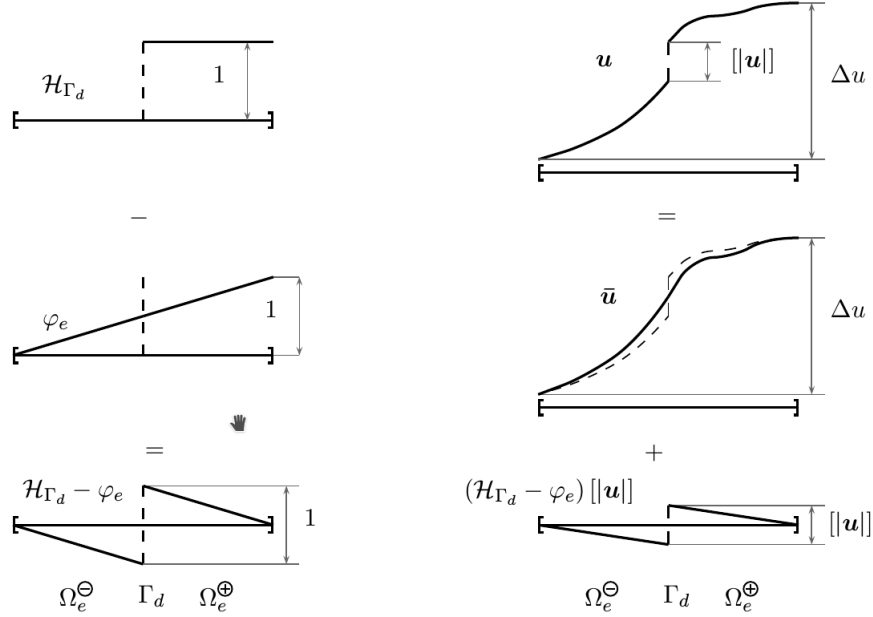


FIGURE 1.4 – Représentation 1D de la décomposition d’un champ de déplacement enrichi d’une discontinuité forte [Roubin, 2013],[Oliver, 1996]

La partie singulière peut être développée dans la base globale et dans la base de la fissure comme suit :

$$\begin{aligned}
 ([\underline{u}] \otimes \underline{n})^s &= \begin{pmatrix} [u]_x & \frac{1}{2}([u]_x n_y + [u]_y n_x) & \frac{1}{2}([u]_x n_z + [u]_z n_x) \\ \cdot & [u]_y & \frac{1}{2}([u]_y n_z + [u]_z n_y) \\ \cdot & \cdot & [u]_z \end{pmatrix}_{(X,Y,Z)} \\
 &= \begin{pmatrix} [u]_n & \frac{1}{2}[u]_m & \frac{1}{2}[u]_t \\ \cdot & 0 & 0 \\ \cdot & \cdot & 0 \end{pmatrix}_{(n,m,t)}
 \end{aligned} \tag{1.3.31}$$

La discrétisation des champs de déformations virtuels et standards ne se font pas dans la même base. Le champ standard est basé sur la décomposition KES (Kinematically Enhanced System) et le champ virtuel sur la décomposition EAS (Enhanced Assumed Strain). Le fait d’utiliser une formulation non symétrique permet d’avoir de meilleurs résultats numériques [Wells and Sluys, 2001] Nous allons maintenant voir ce qui différencie ces deux décompositions.

1.3.4.2 Enrichissement du champs standard de déformation

Nous devons chercher un enrichissement cinématique du champ de contraintes en remplaçant la déformation $\hat{\underline{\underline{\epsilon}}}_b$ par une nouvelle déformation $\hat{\underline{\underline{\epsilon}}}_b^*$. Nous avons besoin de repré-

senter le déplacement $[[[\underline{u}]]]$ sur l'interface sur les nœuds dans Ω_e^\oplus . Donc, en cédant un nouveau déplacement virtuel sous-jacent \hat{u}_b tel que :

$$\hat{u}_b(\underline{x}) = -\varphi(\underline{x}) [[[\underline{u}]]] \quad (1.3.32)$$

avec φ qui vaut 1 aux nœuds de Ω_e^\oplus et 0 aux autres nœuds.

On peut l'écrire comme suit :

$$\varphi(\underline{x}) = \sum_{i=1}^{n^\oplus} N_{p(i)}(\underline{x}) \quad (1.3.33)$$

Si n est le nombre de nœuds dans un élément alors $n^\oplus \leq n$ est le nombre de nœuds dans Ω_e^\oplus et $p : [1..n^\oplus] \mapsto [1..n]$ qui lie le numéro du nœud dans Ω_e^\oplus au numéro de nœuds dans Ω .

Ainsi nous avons :

$$\hat{\underline{\underline{\varepsilon}}}_b = - ([[[\underline{u}]]] \otimes \nabla^s \varphi)^s \quad (1.3.34)$$

Finalement pour l'enrichissement cinématique nous avons :

$$\hat{\underline{\underline{\varepsilon}}} = \underbrace{- ([[[\underline{u}]]] \otimes \nabla^s \varphi)^s}_{\hat{\underline{\underline{\varepsilon}}}_b \text{ régulière}} + \underbrace{\delta_\Gamma ([[[\underline{u}]]] \otimes \underline{n})^s}_{\hat{\underline{\underline{\varepsilon}}}_u \text{ singulière}} \quad (1.3.35)$$

Le terme $\hat{\underline{\underline{\varepsilon}}}_u$ est calculé sur $\Omega \setminus \Gamma_d$ en utilisant une loi de comportement au niveau de l'interface qui fournit un lien entre les contraintes et le saut de déplacement, cette méthode s'appelle "Discrete Strong Discontinuity approach", [Oliver, 2000].

1.3.4.3 Enrichissement du champ virtuel de déformation

Nous avons la même décomposition du champ virtuel et du champ standard avec une partie bornée et une partie illimitée :

$$\gamma = \underbrace{\gamma_b}_{\text{régulière}} + \underbrace{\gamma_u}_{\text{singulière}} \quad (1.3.36)$$

Nous devons déterminer la forme de γ_b afin qu'il respecte la condition d'orthogonalité entre le champ de contrainte et le champ de déformation virtuelle (point clé de la méthode EAS). Elle est équivalente à la continuité du vecteur de traction à l'interface.

$$\gamma = \gamma_b^* + \delta_S ([[[\underline{u}]]] \otimes \underline{n})$$

$$\int_\Omega \gamma : \delta \underline{\underline{\sigma}} \, d\Omega = \int_\Omega \delta \gamma : \underline{\underline{\sigma}} \, d\Omega \quad (1.3.37)$$

$$\int_\Omega \delta \underline{\underline{\sigma}} : \gamma \, d\Omega = \int_\Omega \delta \underline{\underline{\sigma}} : \gamma_b \, d\Omega + \int_\Omega \delta \underline{\underline{\sigma}} : \gamma_u \, d\Omega = \underline{\underline{0}} \quad (1.3.38)$$

$$\int_\Omega \delta \underline{\underline{\sigma}} : \gamma_b \, d\Omega + \int_\Gamma \delta \underline{\underline{\sigma}} : ([[[\underline{u}]]] \otimes \underline{n})^s \, d\partial\Omega = \underline{\underline{0}} \quad (1.3.39)$$

Avec l'hypothèse que si nous avons une interface plane ($\underline{n} = cst$) et un champ de contrainte constant (en utilisant un tétraèdre à quatre nœuds), nous avons, par l'intégration :

$$V \gamma_b + A ([[[\underline{u}]]] \otimes \underline{n})^s = \underline{\underline{0}} \quad (1.3.40)$$

$$\gamma_b = -\frac{A}{V} ([[\underline{u}]] \otimes \underline{n})^s \quad \text{EAS} \quad (1.3.41)$$

Enfin, nous avons pour le champ de contrainte enrichie (EAS) :

$$\gamma = \underbrace{-\frac{A}{V} ([[\underline{u}]] \otimes \underline{n})^s}_{\underline{\underline{\varepsilon}}_b^* \text{ régulière}} + \underbrace{\delta_\Gamma ([[\underline{u}]] \otimes \underline{n})^s}_{\underline{\underline{\varepsilon}}_u^* \text{ singulière}} \quad (1.3.42)$$

Nous pouvons remarquer ici que le champ de déformation virtuel de déformation vérifie le patch test [Simo and Rifai, 1990] :

$$\int_{\Omega} \gamma \, d\Omega = 0 \quad (1.3.43)$$

1.3.5 Application à un élément volumique

Nous allons maintenant présenter le calcul des différentes matrices d'enrichissement dans le cas de l'utilisation d'un élément volumique tétraédrique. Cette discrétisation volumique apporte plusieurs avantages par rapport à l'utilisation d'éléments poutres qui a été faite durant la thèse de Nathan Benkemoun [Benkemoun, 2009] qui a servi de base au développement de l'élément volumique présenté ici.

Tout d'abord, une représentation exacte peut être faite lors du maillage ce qui peut ne pas être le cas lors d'utilisation d'éléments cubiques. Cet élément conduit à représenter parfaitement un problème à contraintes constantes. De plus, une cinématique plus complète que dans le cas de l'utilisation d'un élément poutre peut désormais être représentée. Par conséquent, à la fois la construction géométrique et le comportement mécanique sont représentés avec plus de précisions par rapport à des éléments poutres. Malheureusement, cette modélisation plus complexe conduit à plusieurs inconvénients majeurs inhérents à la mise en place des enrichissements.

Lorsque l'on utilise des éléments d'interface (avec discontinuité faible), l'interface est construite avec les propriétés géométriques des inclusions. Elle conduit à un calcul prédictif de volumes (V^\oplus et V^\ominus) et l'orientation de l'interface (\underline{n}) indépendant de l'état de contrainte. Il est rappelé que la matrice d'interpolation G_w peut être décomposée en θ , une fonction constante dans chacun des sous-domaines et H_w une matrice qui contient les informations d'orientation de l'interface. Afin de respecter la condition d'orthogonalité entre les champs de contrainte et de déformation, le choix de la forme de θ est toujours le suivant :

$$\Theta^{\oplus|\ominus} \triangleq \begin{cases} \Theta^\oplus = \frac{V^\ominus}{V} & \text{if } \underline{n} \cdot (\underline{x} - \underline{\xi}) \geq 0, \\ \Theta^\ominus = -\frac{V^\oplus}{V} & \text{else.} \end{cases} \quad (1.3.44)$$

Cette formulation permet de vérifier la condition de moyenne nulle ce qui permet de vérifier le patch test (voir la partie concernant la méthode des modes incompatibles). En utilisant la notation de Voigt on obtient la matrice d'interpolation pour les discontinuités faibles suivantes :

$$\underline{\underline{G_w}}^{\oplus|\ominus} = \Theta^{\oplus|\ominus} \underline{\underline{H_w}} = \Theta^{\oplus|\ominus} \begin{pmatrix} n_x^2 & n_x m_x & n_x t_x \\ n_y^2 & n_y m_y & n_y t_y \\ n_z^2 & n_z m_z & n_z t_z \\ 2n_x n_y & n_x m_y + n_y m_x & n_x t_y + n_y t_x \\ 2n_y n_z & n_y m_z + n_z m_y & n_y t_z + n_z t_y \\ 2n_x n_z & n_x m_z + n_z m_x & n_x t_z + n_z t_x \end{pmatrix} \quad (1.3.45)$$

Cette matrice est valable aussi bien pour le champ standard que le champ virtuel.

Concernant les discontinuités fortes, les enrichissements du champ standard $\underline{\underline{G_s}}$ et du champ virtuel $\underline{\underline{G_s^*}}$ sont définis par les équations (1.3.46) et (1.3.47).

$$\underline{\underline{G_s}} = - \begin{pmatrix} \varphi_{e,x} & 0 & 0 \\ 0 & \varphi_{e,y} & 0 \\ 0 & 0 & \varphi_{e,z} \\ \varphi_{e,y} & \varphi_{e,x} & 0 \\ 0 & \varphi_{e,z} & \varphi_{e,y} \\ \varphi_{e,z} & 0 & \varphi_{e,x} \end{pmatrix} \quad (1.3.46)$$

$$\underline{\underline{G_s^*}} = -\frac{A}{V} \underline{\underline{H_s^*}} = -\frac{A}{V} \begin{pmatrix} n_x & 0 & 0 \\ 0 & n_y & 0 \\ 0 & 0 & n_z \\ n_y & n_x & 0 \\ 0 & n_z & n_y \\ n_z & 0 & n_x \end{pmatrix} \quad (1.3.47)$$

Finalement nous avons les champs de déformation enrichis suivants pour Ω_e^{\oplus} et Ω_e^{\ominus} :

$$\boxed{\begin{cases} \underline{\underline{\varepsilon}}^{\oplus} = \underline{\underline{B}}d + \underline{\underline{G_w}}^{\oplus} [|\underline{\underline{\varepsilon}}|] + \underline{\underline{G_s}} [|\underline{\underline{u}}|] = \underline{\underline{B}}d + \frac{V^{\ominus}}{V} \underline{\underline{H_w}} [|\underline{\underline{\varepsilon}}|] + \underline{\underline{G_s^*}} [|\underline{\underline{u}}|] \\ \underline{\underline{\varepsilon}}^{\ominus} = \underline{\underline{B}}d + \underline{\underline{G_w}}^{\ominus} [|\underline{\underline{\varepsilon}}|] + \underline{\underline{G_s}} [|\underline{\underline{u}}|] = \underline{\underline{B}}d - \frac{V^{\oplus}}{V} \underline{\underline{H_w}} [|\underline{\underline{\varepsilon}}|] + \underline{\underline{G_s^*}} [|\underline{\underline{u}}|] \end{cases}} \quad (1.3.48)$$

La relation élastique entre le champ de déformation et le champ de contrainte est :

$$\underline{\underline{\sigma}}^{\oplus|\ominus} = \underline{\underline{C}}^{\oplus|\ominus} \underline{\underline{\varepsilon}}^{\oplus|\ominus} \quad (1.3.49)$$

Où $\underline{\underline{C}}$ représente le tenseur de contrainte qui dans le cadre de la loi de Hooke avec un matériau isotrope vaut :

$$\underline{\underline{C}} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \cdot & 1-\nu & \nu & 0 & 0 & 0 \\ \cdot & \cdot & 1-\nu & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \frac{1-2\nu}{2} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \frac{1-2\nu}{2} & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \frac{1-2\nu}{2} \end{pmatrix} \quad (1.3.50)$$

On distingue le tenseur dans la partie Ω_e^\oplus et dans la partie Ω_e^\ominus soit :

$$\begin{cases} \text{Dans } \Omega_e^\oplus, \underline{\underline{C}} = \underline{\underline{C}}^\oplus \text{ avec } \nu = \nu^\oplus \text{ et } E = E^\oplus \\ \text{Dans } \Omega_e^\ominus, \underline{\underline{C}} = \underline{\underline{C}}^\ominus \text{ avec } \nu = \nu^\ominus \text{ et } E = E^\ominus \end{cases} \quad (1.3.51)$$

Maintenant que la construction des différentes matrices d'interpolations est effectuée nous allons voir la méthode de résolution utilisée dans le cas de la méthode E-FEM.

1.3.6 Méthode de résolution

En explicitant le terme $\hat{\sigma}(Bd + G_w \left[\left[\underline{\underline{\varepsilon}} \right] \right] + G_s [u])$ on obtient les linéarisations des trois équations suivantes :

Linéarisation de l'équation d'équilibre global

$$\begin{aligned} f_{int}^e = \int_{\Omega} B^T \hat{\sigma}(Bd + G_w \left[\left[\underline{\underline{\varepsilon}} \right] \right] + G_s [u]) &= \underbrace{B^T (V^\oplus \underline{\underline{C}}^\oplus + V^\ominus \underline{\underline{C}}^\ominus)}_{\underline{\underline{K}}_{bb}} B d \\ &+ \underbrace{\frac{V^\oplus V^\ominus}{V} B^T (\underline{\underline{C}}^\oplus - \underline{\underline{C}}^\ominus) \underline{\underline{H}}_w}_{\underline{\underline{K}}_{bw}} \left[\left[\underline{\underline{\varepsilon}} \right] \right] \\ &+ \underbrace{B^T (V^\oplus \underline{\underline{C}}^\oplus + V^\ominus \underline{\underline{C}}^\ominus)}_{\underline{\underline{K}}_{bs}} G_s n_p [u] \end{aligned} \quad (1.3.52)$$

Linéarisation de l'équation de discontinuité faible

$$\begin{aligned} \int_{\Omega} e \hat{G}_w \hat{\sigma}(Bd + G_w \left[\left[\underline{\underline{\varepsilon}} \right] \right] + G_s [u]) d\Omega &= \underbrace{\frac{V^\oplus V^\ominus}{V} \underline{\underline{H}}_w^T (\underline{\underline{C}}^\oplus - \underline{\underline{C}}^\ominus)}_{\underline{\underline{K}}_{wb}} B d \\ &+ \underbrace{\frac{V^\oplus V^\ominus}{V^2} \underline{\underline{H}}_w^T (V^\ominus \underline{\underline{C}}^\oplus - V^\oplus \underline{\underline{C}}^\ominus) \underline{\underline{H}}_w}_{\underline{\underline{K}}_{ww}} \left[\left[\underline{\underline{\varepsilon}} \right] \right] \\ &+ \underbrace{\frac{V^\oplus V^\ominus}{V} \underline{\underline{H}}_w^T (\underline{\underline{C}}^\oplus - \underline{\underline{C}}^\ominus)}_{\underline{\underline{K}}_{ws}} G_s n_p [u] \end{aligned} \quad (1.3.53)$$

Linéarisation de l'équation de discontinuité faible

$$\begin{aligned}
\int_{\Omega} e \hat{G}_s \hat{\sigma} (Bd + G_w \left[\left[\underline{\underline{\varepsilon}} \right] \right] + G_s [u]) d\Omega &= \underbrace{B^T (V^{\oplus} \underline{\underline{C}}^{\oplus} + V^{\ominus} \underline{\underline{C}}^{\ominus}) B d}_{K_{s^*b}} \\
&+ \underbrace{\frac{V^{\oplus} V^{\ominus}}{V} B^T (\underline{\underline{C}}^{\oplus} - \underline{\underline{C}}^{\ominus}) \underline{\underline{H}}_w \left[\left[\underline{\underline{\varepsilon}} \right] \right]}_{K_{sSw}} \\
&+ \underbrace{B^T (V^{\oplus} \underline{\underline{C}}^{\oplus} + V^{\ominus} \underline{\underline{C}}^{\ominus}) G_s n_p [u]}_{K_{sSs}}
\end{aligned} \tag{1.3.54}$$

Une fois la linéarisation des trois équations réalisée le système complet peut être mis sous la forme matricielle suivante :

$$\left[\begin{array}{ccc} \underline{\underline{K}}_{bb} & \underline{\underline{K}}_{bw} & \underline{\underline{K}}_{bs} \\ \underline{\underline{K}}_{wb} & \underline{\underline{K}}_{ww} & \underline{\underline{K}}_{ws} \\ \underline{\underline{K}}_{s^*b} & \underline{\underline{K}}_{s^*w} & \underline{\underline{K}}_{s^*s} + \underline{\underline{K}}_q \end{array} \right]_{n+1}^k \left\{ \begin{array}{c} \Delta d \\ \Delta \left[\left[\underline{\underline{\varepsilon}} \right] \right] \\ \Delta [u] \end{array} \right\}_{n+1}^{k+1} = \left\{ \begin{array}{c} -A_{e=1}^{nelm} [f_{int}^e - f_{ext}^e] \\ -h \\ -\Phi_0 \end{array} \right\}_{n+1}^k \tag{1.3.55}$$

La résolution de ce système linéaire est réalisée en deux parties, une partie globale (l'équation d'équilibre global) et une partie locale (les équations de discontinuités faibles et fortes). La résolution est ensuite faite en utilisant la méthode "operator split method". On calcule dans un premier temps les variables internes Δ et $\Delta [u]^{k+1}$ pour une valeur fixe de d :

$$h_{\left[\left[\underline{\underline{\varepsilon}} \right] \right]} = 0 \tag{1.3.56}$$

$$\Phi_0 = 0 \tag{1.3.57}$$

Après cette étape on réalise une condensation statique [Wilson, 1974] du système. Ceci revient à effectuer le calcul de l'équation d'équilibre global avec une matrice de rigidité modifiée K_m . Pour calculer cette nouvelle matrice de rigidité on part de l'équation globale :

$$\underline{\underline{K}}_{bb} \Delta d_{n+1}^{k+1} + \left[\underline{\underline{K}}_{bw} \quad \underline{\underline{K}}_{bs} \right] \left\{ \begin{array}{c} \Delta \left[\left[\underline{\underline{\varepsilon}} \right] \right] \\ \Delta [u] \end{array} \right\}_{n+1}^{k+1} = -A_{e=1}^{nelm} [f_{int}^e - f_{ext}^e] \tag{1.3.58}$$

On exprime ensuite les variables internes en fonction des déplacements :

$$\left[\begin{array}{cc} \underline{\underline{K}}_{ww} & \underline{\underline{K}}_{ws} \\ \underline{\underline{K}}_{s^*w} & \underline{\underline{K}}_{s^*s} + \underline{\underline{K}}_q \end{array} \right]_{n+1}^k \left\{ \begin{array}{c} \Delta \left[\left[\underline{\underline{\varepsilon}} \right] \right] \\ \Delta [u] \end{array} \right\}_{n+1}^{k+1} = - \left[\begin{array}{c} \underline{\underline{K}}_{wb} \\ \underline{\underline{K}}_{s^*b} \end{array} \right]_{n+1}^k \Delta d_{n+1}^{k+1} \tag{1.3.59}$$

Soit :

$$\left\{ \begin{array}{c} \Delta \left[\begin{array}{c} \underline{\underline{\varepsilon}} \\ \underline{\underline{u}} \end{array} \right] \\ \Delta \left[\begin{array}{c} \underline{\underline{\varepsilon}} \\ \underline{\underline{u}} \end{array} \right] \end{array} \right\}_{n+1}^{k+1} = - \left(\left[\begin{array}{cc} \underline{\underline{K}}_{ww} & \underline{\underline{K}}_{ws} \\ \underline{\underline{K}}_{s^*w} & \underline{\underline{K}}_{s^*s} + \underline{\underline{K}}_q \end{array} \right]_{n+1}^k \right)^{-1} \left[\begin{array}{c} \underline{\underline{K}}_{wb} \\ \underline{\underline{K}}_{s^*b} \end{array} \right]_{n+1}^k \Delta d_{n+1}^{k+1} \quad (1.3.60)$$

L'équation (1.3.58) devient alors :

$$\underbrace{\underline{\underline{K}}_{bb} - \left[\begin{array}{cc} \underline{\underline{K}}_{bw} & \underline{\underline{K}}_{bs} \end{array} \right]}_{K_m} \left(\left[\begin{array}{cc} \underline{\underline{K}}_{ww} & \underline{\underline{K}}_{ws} \\ \underline{\underline{K}}_{s^*w} & \underline{\underline{K}}_{s^*s} + \underline{\underline{K}}_q \end{array} \right]_{n+1}^k \right)^{-1} \left[\begin{array}{c} \underline{\underline{K}}_{wb} \\ \underline{\underline{K}}_{s^*b} \end{array} \right]_{n+1}^k \Delta d_{n+1}^{k+1} = -A_{e=1}^{n_{elm}} [f_{int}^e - f_{ext}^e] \quad (1.3.61)$$

La matrice K_m est une matrice de même taille que $\underline{\underline{K}}_{bb}$ (de la taille du système). On retrouve ainsi un problème éléments finis classique avec en plus une résolution locale pour les variables internes. Cette propriété de la méthode E-FEM permet la modélisation de la multi-fissuration sans augmentation de la taille globale du système à résoudre. Cette propriété ne se retrouve pas dans la méthode X-FEM, où les éléments fissurés contiennent des degrés de liberté supplémentaires [Moës et al., 1999]. Dans le cas de la multi-fissuration le nombre d'éléments fissurés étant important, la taille du système devient trop grande et la résolution du problème global demande alors un temps de calcul trop important.

Cette propriété nous conforte dans le choix que nous avons fait d'utiliser un modèle à discontinuité de type E-FEM car le nombre d'éléments fissurés lors des différents calculs est très important.

La méthode de résolution choisie ici pour le problème non linéaire fait partie des méthodes "quasi-Newton" et se nomme BFGS (Broyden-Fletcher-Goldfarb-Shanno) [Broyden, 1969],[Fletcher, 1970],[Goldfarb, 1970],[Shanno, 1970]. Un des points clés de la méthode est d'apporter une correction à la matrice de raideur $K_{(k+1)}$ à chaque itération en fonction de la matrice $K^{(k)}$:

$$K^{(k+1)} = K^{(k)} + \frac{\gamma\gamma^T}{\gamma^T\delta} - \frac{(K^{(k)}\delta)(K^{(k)}\delta)^T}{\delta^T K^{(k)}\delta} \quad (1.3.62)$$

La formulation de l'incrément de la matrice de raideur permet de garder le caractère symétrique de celle-ci. Le vecteur γ dépend de l'incrément de déplacement à l'itération k et le vecteur δ de la différence de résidu entre l'itération k et $k+1$.

Une méthode de type "line-search" est aussi utilisée dans le but d'accélérer la convergence du calcul. Cette méthode consiste à chercher un coefficient multiplicateur de l'incrément de déplacement trouvé. On cherche alors pour chaque itération un coefficient $s^{(k+1)}$ qui minimise l'énergie, ce coefficient est compris entre 0 et 1. Le déplacement est alors incrémenté comme suit :

$$u^{(k+1)} = u^{(k)} + s^{(k+1)}\Delta u^{(k+1)} \quad (1.3.63)$$

L'implémentation de l'algorithme BFGS avec la méthode "line-search" est décrite dans [Matthies and Strang, 1979]. Cette procédure est intégrée sur le composant éléments finis coFeap utilisé dans le cadre de cette thèse.

1.4 Modèle discret sur la discontinuité

Pour décrire le modèle de fissuration nous avons besoin de deux critères. Le premier critère est un critère de localisation. Ce critère nous indique le moment à partir duquel la discontinuité forte va être activée (et donc que la fissuration va avoir lieu).

Une fois ce critère activé nous avons besoin d'un critère d'ouverture qui lie l'ouverture de la fissure à la contrainte.

Dans cette partie nous allons présenter deux modèles suivants.

- Le premier modèle est un critère de type Rankine avec une ouverture en mode I. Ce modèle est basé sur un critère simple en traction et permet d'avoir une bonne modélisation du comportement des bétons (en traction et en compression). Ce modèle sera utilisé pour la modélisation de la fissuration d'une entaille.
- Le deuxième modèle est un modèle de type Mohr-Coulomb avec une ouverture en mode II. Ce critère permet d'avoir une bonne modélisation du comportement en compression triaxiale des shales.

1.4.1 Critère de Rankine (mode I)

1.4.1.1 Critère de localisation

Pour pouvoir définir le critère de localisation il est nécessaire de définir la direction d'orientation de la fissure. On doit ici de distinguer deux cas.

Le premier cas concerne un élément dont la discontinuité faible est activée (qui possède donc deux modules Young différents). Dans ce cas la direction est définie par la normale à l'interface de la discontinuité faible. On peut alors calculer le vecteur traction \mathcal{T} . Le critère de localisation s'écrit alors :

$$\Phi^{weak} = \sigma_{eq} - \sigma_y = n \cdot \mathcal{T} - \sigma_y \quad (1.4.1)$$

Dans le cas d'un élément où la discontinuité faible n'est pas activée nous allons utiliser la direction principale liée à la plus grande valeur propre du tenseur de contrainte σ_1 . On calcule ensuite le critère comme suit :

$$\Phi^{strong} = \sigma_{eq} - \sigma_y = \sigma_1 - \sigma_y \quad (1.4.2)$$

Dans les deux cas il est nécessaire de définir au préalable la valeur de σ_y correspondant à la limite en traction de l'élément.

1.4.1.2 Critère d'ouverture

Pour le critère d'ouverture le choix la fonction seuil s'écrit sous la forme :

$$\Phi_0 = n \cdot \mathcal{T} - (\sigma_y - q) \quad (1.4.3)$$

Cette fonction seuil est active uniquement si la localisation a eu lieu. Pour la modélisation des matériaux quasi-fragiles un adoucissement exponentiel est choisi :

$$q = \sigma_y(1 - \exp(-\frac{\sigma_y}{\mathcal{G}_f}[u])) \quad (1.4.4)$$

Cet adoucissement permet de définir l'énergie de fissuration \mathcal{G}_f en $J \cdot m^2$. Lorsque l'énergie de fissuration tend vers 0 le comportement est fragile et lorsque l'énergie tend vers ∞ le comportement devient ductile.

Ce critère amène à une équation non-linéaire en fonction de $[u]$, en effet σ_{eq} et q dépendent de $[u]$ non linéairement (pour q). L'équation peut être mise sous la forme (pour plus d'information voir [Roubin, 2013] et [Roubin et al., 2015]) :

$$\underbrace{\left(\frac{K_{s^*b}}{Te} - \frac{K_{s^*w}}{Te} \frac{K_{ww}^{-1} K_{wb}}{Te} \right) d}_{Te} + \underbrace{\left(\frac{K_{s^*s}}{M} - \frac{K_{s^*w}}{M} \frac{K_{ww}^{-1} K_{ws}}{M} \right) [u]}_M - \sigma_y \exp\left(-\frac{\sigma_y}{\mathcal{G}_f}[u]\right) = 0 \quad (1.4.5)$$

Soit :

$$Te + M[u] = \sigma_y \exp\left(-\frac{\sigma_y}{\mathcal{G}_f}[u]\right) \quad (1.4.6)$$

Une solution analytique existe si $Te > \sigma_y$ ce qui correspond à la condition de localisation. Cette solution est obtenue à l'aide de la branche principale de la fonction W de Lamber (W_0) :

$$[u]_{sol} = \frac{\mathcal{G}_f}{\sigma_y} \left(W_0 \left(\frac{\sigma_y^2 \exp\left(\frac{\sigma_y Te}{\mathcal{G}_f M}\right)}{\mathcal{G}_f M} \right) - \frac{\sigma_y Te}{\mathcal{G}_f M} \right) \quad (1.4.7)$$

1.4.1.3 Comportement d'un matériau à matrice argileuse avec un critère de Rankine

Dans cette partie nous allons montrer les résultats obtenus avec le modèle de type Rankine. Les résultats en traction, compression et compression triaxiale seront donnés dans un premier temps puis la deuxième partie traitera de l'anisotropie induite par un état de fissuration. Tous les résultats donnés sont obtenus pour un même matériau et un même maillage.

Paramètres de la modélisation

Le domaine utilisé, visible sur la figure 1.5 est un cube de $100\mu m$ possédant des inclusions sphériques. La génération des inclusions est décrite plus loin dans ce chapitre.

La matrice est de type argileuse et a pour module Young 20 GPa et pour coefficient de Poisson 0.3. Les inclusions ont un rayon de $5\mu m$ et représentent une fraction volumique de 10%. Leur module Young est de 100 GPa et leur coefficient de Poisson est de 0.35.

Seul la matrice et les éléments d'interfaces (ayant une discontinuité faible) sont enrichis par une discontinuité forte. La résistance en traction est imposée à 6 MPa et l'énergie de fissuration à $0.6J \cdot m^{-2}$.

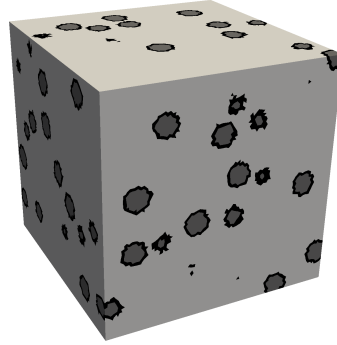


FIGURE 1.5 – Exemple de maillage obtenu pour une minéralogie

Résultats obtenus en traction et en compression

Le calcul en traction compression a été réalisé en imposant un déplacement suivant la direction Z de la face $z = 100\mu m$ et en bloquant les déplacements suivant Z de la face $z = 0\mu m$. Les pas de déplacement utilisés ici sont de $1.6nm$ pour la traction et $8.8nm$ pour la compression.

Le comportement obtenu en traction-compression est visible sur la figure 1.6. Bien que le critère utilisé soit simple et basé uniquement sur une rupture en traction en mode I, on obtient une rupture fragile à la fois en traction et en compression. On observe un rapport de 3 entre la résistance en traction et la résistance en compression. Ce type de comportement permet de bien modéliser le comportement de matériaux à matrice cimentaire avec peu de paramètre (σ_y et \mathcal{G}_f).

Résultats obtenus en compression triaxiale

La réalisation du calcul en compression triaxiale est effectuée en deux étapes. La première étape consiste à imposer la pression de confinement en réalisant une tri-compression. Cette étape est visible sur la figure 1.7.a. Une fois cette étape réalisée on récupère les déplacements obtenus sur les faces où l'on souhaite imposer la compression triaxiale (ici la face du haut et du bas) et l'on impose ce déplacement sur ses faces (on continue d'imposer la pression sur les quatre autres faces). Enfin on impose un déplacement sur la face du haut (en plus du déplacement dû à la tri-compression) pour réaliser la compression triaxiale, cette phase est observable sur la figure 1.7.b.

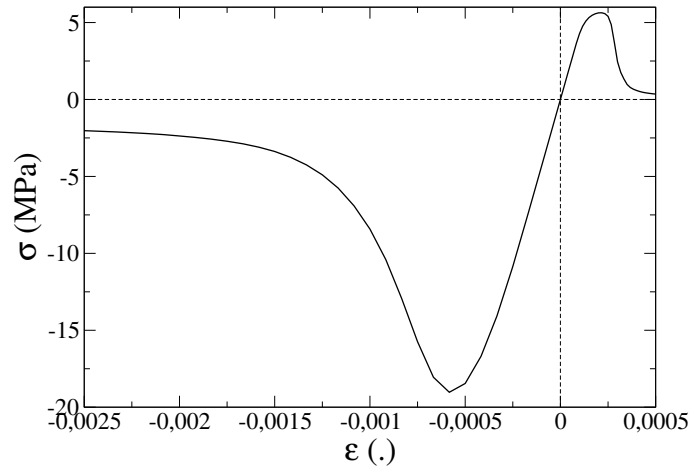
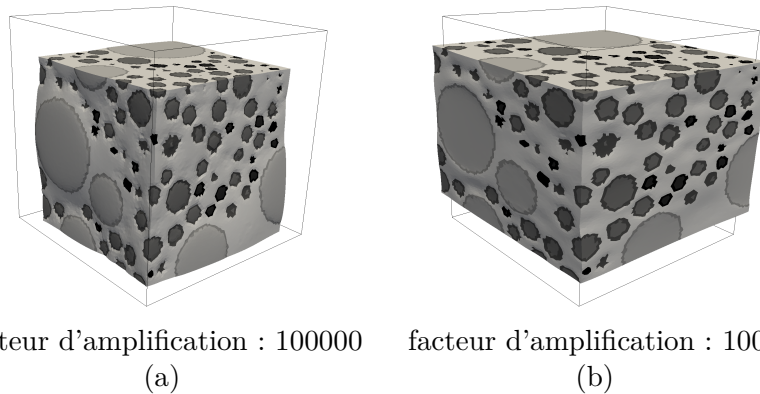


FIGURE 1.6 – Comportement mécanique en traction et en compression

Le calcul est réalisé à l'aide du logiciel cofeap. Chacune des étapes est réalisée à l'aide d'un composant cofeap, le client récupère les déplacements obtenus lors du premier calcul de confinement via le premier composant cofeap et l'impose au second composant cofeap qui réalise ensuite le calcul de compression triaxiale.



facteur d'amplification : 100000

(a)

facteur d'amplification : 1000

(b)

FIGURE 1.7 – Déformation obtenue en (a) tricompression (b) compression triaxiale

On réalise le même calcul avec plusieurs pressions de confinement pour obtenir les résultats visibles sur la figure 1.8. La résistance augmente lorsque l'on augmente le confinement. Le critère de Rankine montre ici ses limites. En effet avec uniquement un critère traction en mode I le comportement obtenu n'est pas celui attendu en compression triaxiale.

Si l'on regarde les résultats dans le repère de Mohr sur la figure 1.9, on remarque que le comportement n'est pas tout à fait de type Mohr-Coulomb, si l'on calcule quand même la cohésion et l'angle de frottement on trouve que la cohésion obtenue est très faible (de l'ordre de 5 MPa) et que l'angle de frottement est bien trop grand ($\tan(\phi) = 1.6$). Ces résultats nous ont poussés au développement d'un second critère, le critère de Mohr-

Coulomb en mode II, qui correspond mieux à la modélisation d'un matériau à matrice argileuse. Ce modèle est présenté plus loin dans le chapitre.

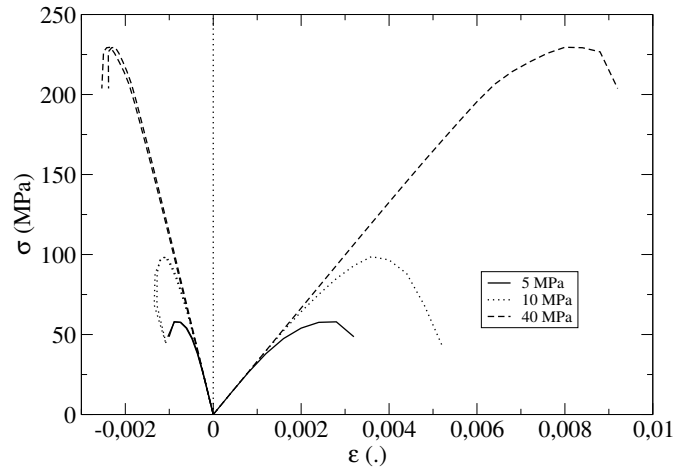


FIGURE 1.8 – Comportement mécanique obtenu en compression triaxiale

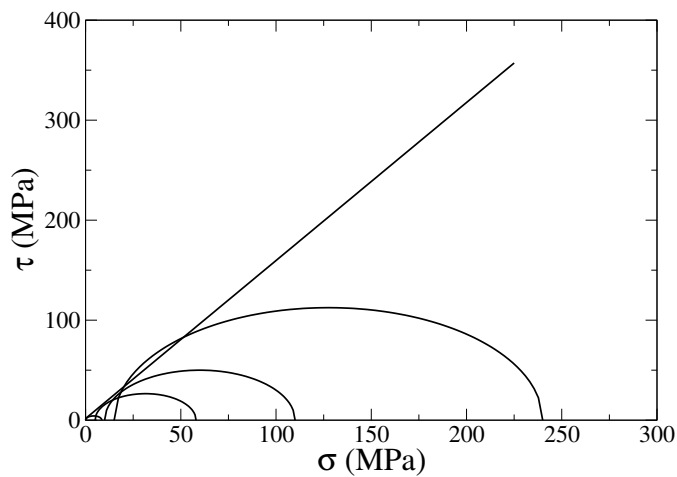


FIGURE 1.9 – Comportement mécanique obtenu en compression triaxiale dans le repère de Mohr

Calcul de l'anisotropie induite

La possibilité d'avoir accès aux directions et ouvertures des fissures nous a permis de calculer l'anisotropie induite par un état de fissuration. Nous allons dans cette partie étudier l'influence de la fissuration due à de la traction ou à de la compression sur le module Young ainsi que sur la résistance en traction du matériau.

Afin d'obtenir l'endommagement du matériau dans toutes les directions, nous avons séparé le calcul en deux étapes :

La première étape consiste à effectuer un calcul en traction ou en compression avec le modèle d'éléments finis présenté auparavant. Pour chaque itération dans le calcul non linéaire, nous enregistrons les valeurs d'ouverture des fissures et leur orientation. Ces valeurs sont utilisées pour tenir compte de la non-linéarité et seront utilisées pour le calcul d'endommagement dans les différentes directions.

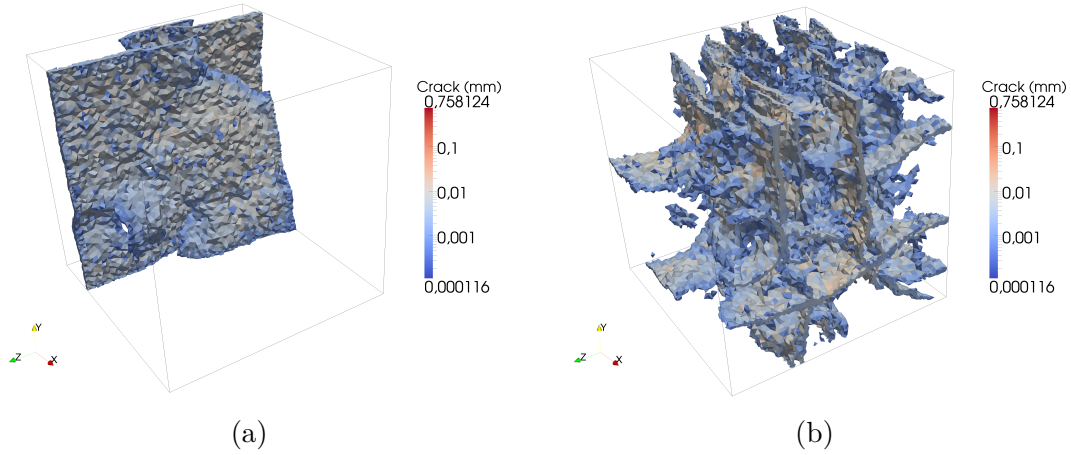


FIGURE 1.10 – Faciès de fissuration (a) en traction, (b) en compression

L'étape suivante consiste à imposer les ouvertures des fissures et les orientations. On doit ensuite calculer le module d'élasticité résiduel en testant dans chaque direction. L'endommagement est défini comme étant le rapport entre le module d'élasticité initial et le module d'élasticité résiduel. Afin d'obtenir l'endommagement dans différentes directions on utilise des conditions aux limites de type KUBC (Kinetic Uniform Boundary Conditions).

Afin d'obtenir l'anisotropie de résistance en traction, nous devons faire la même chose que pour l'endommagement, mais nous avons ici besoin de déterminer la résistance en traction et non plus le module d'élasticité. Les calculs permettant de déterminer la résistance en traction étant bien plus lents, nous calculons l'anisotropie de la résistance seulement suivant les axes principaux et nous n'utilisons plus les conditions aux limites de type KUBC.

Pour obtenir le module d'élasticité dans différentes directions nous imposons des conditions aux limites de type KUBC. Le tenseur de déformation que nous appliquons est décrit dans l'équation 1.4.8, ψ et θ sont les angles qui définissent la direction choisie.

$$\underline{\underline{\varepsilon}} = \varepsilon_0 * \begin{pmatrix} \cos(\psi) * \sin(\theta) & 0 & 0 \\ 0 & \sin(\psi) * \sin(\theta) & 0 \\ 0 & 0 & \cos(\theta) \end{pmatrix} \quad (1.4.8)$$

Ensuite nous imposons les déplacements aux bords suivants (avec $\underline{x} \in \delta\Omega$) :

$$d(\underline{x}) = \underline{\underline{\varepsilon}} \cdot \underline{x} \quad (1.4.9)$$

Pour commencer, nous allons présenter les résultats obtenus pour l'anisotropie d'élasticité dans les axes principaux. Sur la figure 1.11.a on peut voir l'endommagement dû à la traction dans les trois directions principales. La traction a été effectuée suivant l'axe X et on peut noter que l'endommagement le plus important a lieu dans cette direction ($d_{x,max} = 0.99$), les endommagements dans les deux autres directions sont plus faibles et du même ordre de grandeur ($d_{y,max} = 0.169$ et $d_{z,max} = 0.175$). Il semble logique que la traction endommage peu le matériau dans les axes normaux au chargement.

Dans le cas de la compression suivant l'axe x, qui peut être observé sur la figure 1.11.b, la conséquence obtenue sur l'endommagement est opposée à celle observée précédemment. En effet l'endommagement augmente plus rapidement suivant les axes perpendiculaires au chargement, mais les valeurs maximum atteintes sont proches ($d = 1$). À nouveau le comportement obtenu semble logique pour un chargement de compression.

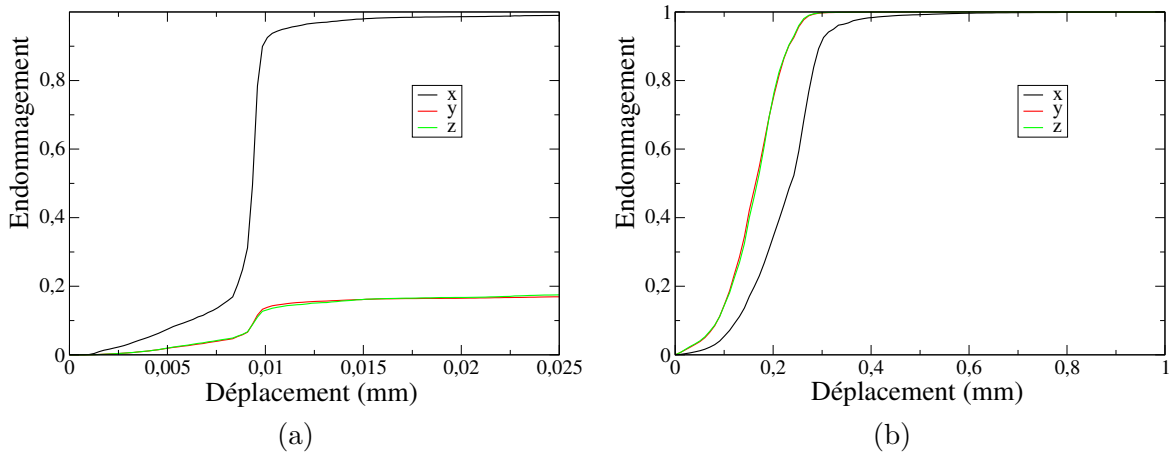


FIGURE 1.11 – Endommagement dans les directions principales (a) en traction, (b) en compression

Si l'on regarde sur la figure 1.12.a et 1.12.b on peut observer l'évolution de l'endommagement suivant toutes les directions à $F = F_{max}$ et après. La surface a une forme de cacahuète, elle est plus longue suivant l'axe X (l'axe de chargement) que suivant les deux autres directions (suivant Y et Z ou l'endommagement est plus faible). On remarque que la surface n'évolue après que le pic soit atteint.

Dans le cas de la compression au moment du pic, sur la figure 1.12.a, la surface est sphéroïdale. Le rayon suivant la direction de chargement est plus faible que suivant les deux autres axes. Lorsque l'on augmente le chargement, sur la figure 1.12.b, la surface prend une forme de sphère. On retrouve le même comportement que celui observé sur les trois axes principaux avec un retard de l'endommagement suivant l'axe de chargement.

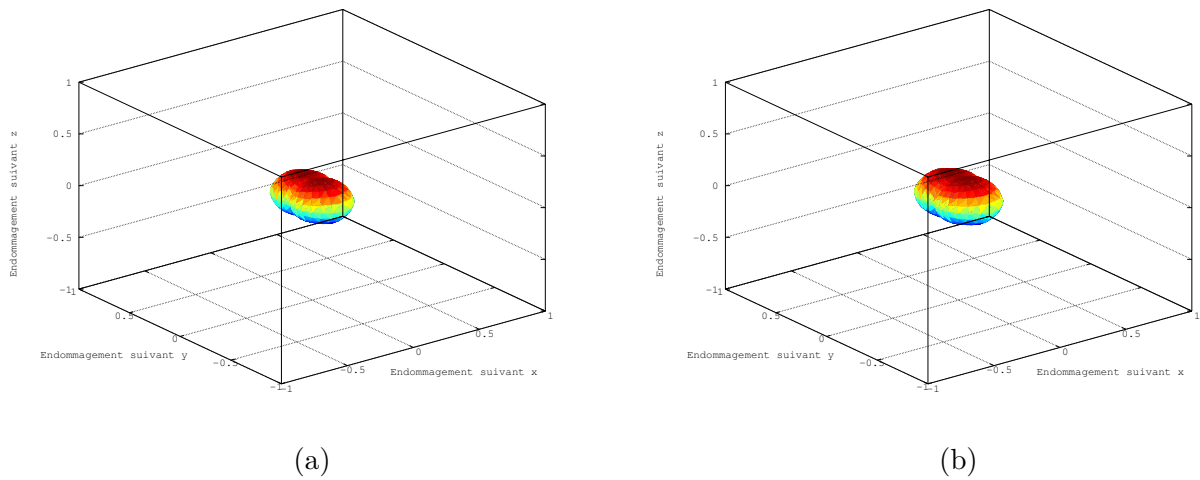


FIGURE 1.12 – Endommagement dû à la traction suivant toutes les directions (a) à $F = F_{max}$, (b) après F_{max}

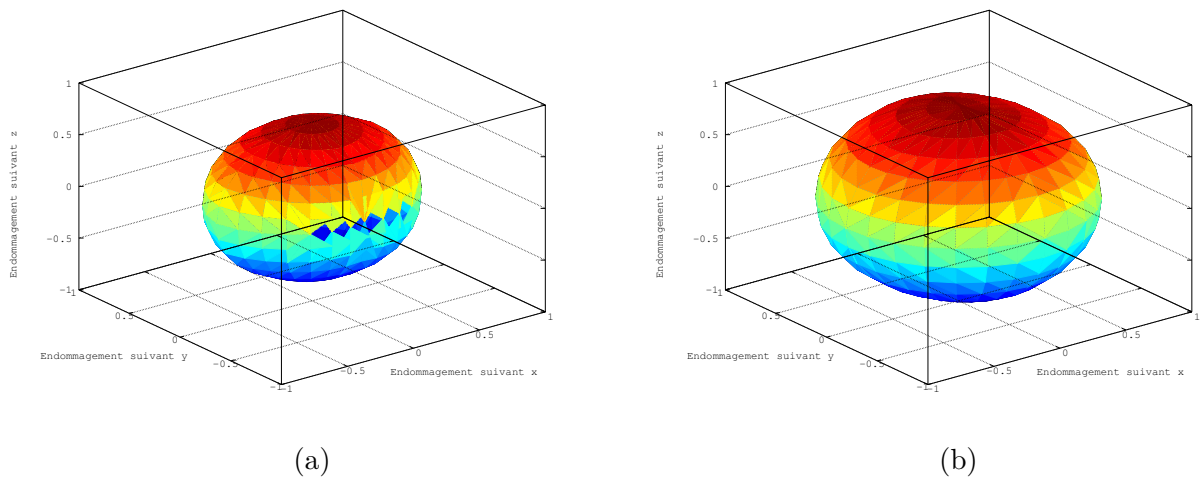


FIGURE 1.13 – Endommagement dû à la compression suivant toutes les directions (a) à $F = F_{max}$, (b) après F_{max}

Nous allons maintenant nous intéresser à l'anisotropie obtenue sur la résistance en traction. Les résultats obtenus sur la figure 1.14 sont obtenus après une traction suivant l'axe X. Pour différents niveaux de chargement de l'essai de traction initial nous récupérons la fissuration et nous mesurons la résistance en traction résiduelle suivant les trois directions principales. Nous calculons ensuite le ratio entre la résistance initiale et la résistance obtenue. On observe à nouveau que la traction a uniquement un effet sur l'axe de chargement, l'impact suivant les autres axes étant relativement faible.

Concernant la compression, sur la figure 1.14.b, on observe à nouveau un impact sur les trois directions avec un retard suivant la direction du chargement.

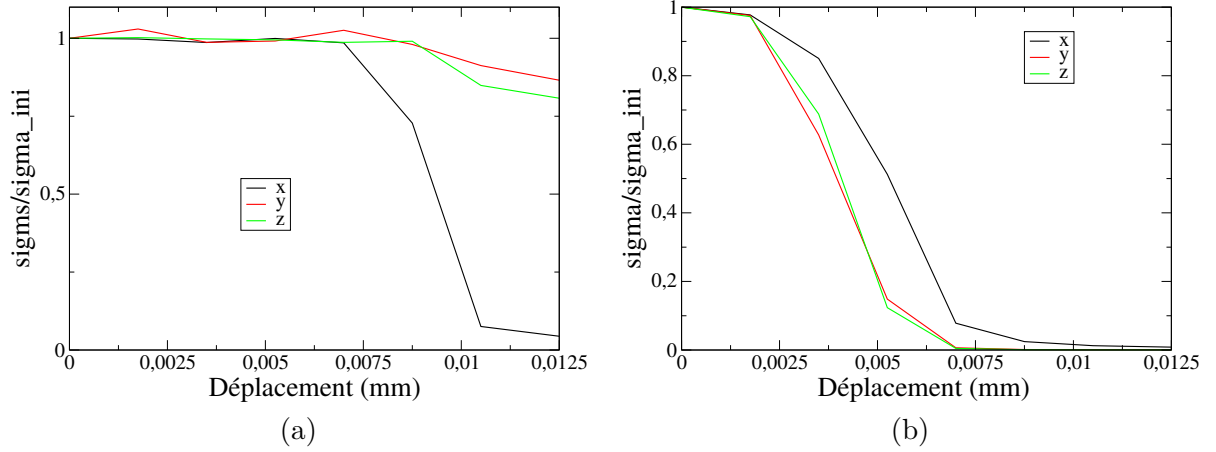


FIGURE 1.14 – Contraintes en fonction de l'endommagement (a) en traction, (b) en compression

1.4.2 Critère de Mohr-Coulomb (mode II)

Le critère de Mohr-Coulomb [Salençon, 2002] est un critère couramment utilisé dans le milieu de la mécanique des roches. Ce critère introduit deux paramètres, la cohésion C (en MPa) et l'angle de frottement φ . Nous allons décrire dans cette partie l'utilisation du critère de Mohr-Coulomb avec la méthode EFEM.

1.4.2.1 Critère de localisation

Comme pour le critère de Rankine nous avons à décrire deux cas selon si la discontinuité faible est activée ou pas.

Le critère de localisation Φ_l peut être noté en fonction du vecteur traction $T = \sigma \cdot n$:

$$\Phi_l = \sup_n (\|T_t\| + T_n \tan \varphi - C) \quad (1.4.10)$$

Avec $T_t = T - T_n \cdot n$ et $T_n = Tn$.

Dans le cas où la discontinuité faible est activée le vecteur n est connu et on peut donc calculer le critère à l'aide de cette formulation.

Lorsque la discontinuité n'est pas activée on calcule le critère à l'aide des contraintes principales $(\sigma_1, \sigma_2, \sigma_3)$ avec $\sigma_1 \geq \sigma_2 \geq \sigma_3$:

$$\Phi_l = \frac{1}{2 \cos \varphi} (\sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3) \sin \varphi - 2C \cos \varphi) \quad (1.4.11)$$

Les formulations des équations (1.4.10) et (1.4.11) donne les mêmes résultats, l'avantage de la deuxième formulation est de pouvoir calculer directement le critère lorsque l'on ne connaît pas n alors que pour la première formulation on doit chercher la normale qui

maximise le critère. Pour le critère de localisation la connaissance de la normale n'est pas importante mais celle ci le deviendra pour exprimer le critère d'ouverture. Il est donc nécessaire de connaître la normale. Celle ci peut être obtenue analytiquement [Bai, 2009] en fonction de la base propre des contraintes principales et de l'angle de frottement φ . Les coordonnées de n dans la base propre (ν_1, ν_2, ν_3) sont les suivantes :

$$\begin{cases} \nu_1^2 &= \frac{1}{1 + (\sqrt{1 + \varphi^2} - \varphi)^2} \\ \nu_2^2 &= 0 \\ \nu_3^2 &= \frac{1}{1 + (\sqrt{1 + \varphi^2} + \varphi)^2} \end{cases} \quad (1.4.12)$$

1.4.2.2 Critère d'ouverture

Une fois la localisation effectuée, si $\Phi \geq 0$ nous devons calculer la valeur d'ouverture de fissure. Dans le cas de l'utilisation du critère de Mohr-Coulomb nous allons utiliser une ouverture en mode II (en cisaillement). On repart de l'équation du critère d'ouverture :

$$\Phi = \sigma_{eq} - (\sigma_u - q) \quad (1.4.13)$$

Avec :

$$\begin{aligned} \sigma_{eq} &= \|T_t\| + T_n \tan \varphi \\ \sigma_y &= C \\ q &= C \left(1 - \exp\left(-\frac{C}{G_f} u\right)\right) \end{aligned}$$

On fait ici l'hypothèse que $\|T_t\| = T_t \cdot n_t$ avec n_t la direction de cisaillement. On peut alors décomposer T_t et T_n en fonction de d , $\left[\left[\underline{\underline{\varepsilon}}\right]\right]$ et $[u]$ en utilisant la valeur du vecteur traction T :

$$\begin{aligned} T_n = T \cdot n &= n^T \frac{1}{V} \underline{\underline{H}}_s^{*T} \left(V^\oplus \underline{\underline{C}}^\oplus + V^\ominus \underline{\underline{C}}^\ominus \right) B d \\ &+ n^T \frac{V^\oplus V^\ominus}{V} \underline{\underline{H}}_s^{*T} \left(\underline{\underline{C}}^\oplus - \underline{\underline{C}}^\ominus \right) \underline{\underline{H}}_w \left[\left[\underline{\underline{\varepsilon}} \right] \right] \\ &+ n^T \frac{1}{V} \underline{\underline{H}}_s^{*T} \left(V^\oplus \underline{\underline{C}}^\oplus + V^\ominus \underline{\underline{C}}^\ominus \right) \underline{\underline{G}}_s n_p |u| \\ &= n^T \underline{\underline{K}}_{s*b} d + n^T \underline{\underline{K}}_{s*w} \left[\left[\underline{\underline{\varepsilon}} \right] \right] + n^T \underline{\underline{K}}_{s*s} |u| \end{aligned} \quad (1.4.14)$$

Pour le calcul de T_t il faut noter que $n \cdot n_t = 0$ ce qui réduit l'expression à $\|T_t\| = T \cdot n_t$, on retrouve alors une expression similaire à T_n , seul le vecteur de projection change :

$$\begin{aligned}
\|T_t\| = T \cdot n_t &= \frac{1}{V} \underline{H_s}^{*T} \left(V^\oplus \underline{C}^\oplus + V^\ominus \underline{C}^\ominus \right) B d \cdot n_t \\
&+ \frac{V^\oplus V^\ominus}{V} \underline{H_s}^{*T} \left(\underline{C}^\oplus - \underline{C}^\ominus \right) \underline{H_w} \left[\underline{\underline{\varepsilon}} \right] \cdot n_t \\
&+ \frac{1}{V} \underline{H_s}^{*T} \left(V^\oplus \underline{C}^\oplus + V^\ominus \underline{C}^\ominus \right) \underline{G_s} n_p |u| \cdot n_t \\
&= \underline{K_{s^*b}} d \cdot n_t + \underline{K_{s^*w}} \left[\underline{\underline{\varepsilon}} \right] \cdot n_t + \underline{K_{s^*s}} |u| \cdot n_t
\end{aligned} \tag{1.4.15}$$

Comme pour le critère de Rankine, nous allons exprimer $\left[\underline{\underline{\varepsilon}} \right]$ en fonction de d et $|u|$:

$$\left[\underline{\underline{\varepsilon}} \right] = - \left(\underline{K_{ww}}^{-1} \underline{K_{wb}} d + \underline{K_{ww}}^{-1} \underline{K_{ws}} |u| \right) \tag{1.4.16}$$

On peut alors exprimer σ_{eq} en fonction de d et $|u|$ en remplaçant $\left[\underline{\underline{\varepsilon}} \right]$ dans les expressions de T_t et T_n :

$$\begin{aligned}
\sigma_{eq} &= \underline{K_{s^*b}} d \cdot n_t - \underline{K_{s^*w}} \left(\underline{K_{ww}}^{-1} \underline{K_{wb}} d + \underline{K_{ww}}^{-1} \underline{K_{ws}} |u| \right) \cdot n_t + \underline{K_{s^*s}} |u| \cdot n_t \\
&+ \left(n^T \underline{K_{s^*b}} d - n^T \underline{K_{s^*w}} \left(\underline{K_{ww}}^{-1} \underline{K_{wb}} d + \underline{K_{ww}}^{-1} \underline{K_{ws}} |u| \right) + n^T \underline{K_{s^*s}} |u| \right) \tan \varphi
\end{aligned} \tag{1.4.17}$$

Comme pour le critère de Rankine, on peut mettre l'équation sous la forme $\sigma_{eq} = T^e + M|u|$ avec :

$$\begin{aligned}
T^e &= \underline{K_{s^*b}} d \cdot n_t - \underline{K_{s^*w}} \underline{K_{ww}}^{-1} \underline{K_{wb}} d \cdot n_t \\
&+ \left(n^T \underline{K_{s^*b}} d - n^T \underline{K_{s^*w}} \underline{K_{ww}}^{-1} \underline{K_{wb}} d \right) \tan \varphi
\end{aligned} \tag{1.4.18}$$

et :

$$\begin{aligned}
M &= -\underline{K_{s^*w}} \underline{K_{ww}}^{-1} \underline{K_{ws}} |u| \cdot n_t + \underline{K_{s^*s}} |u| \cdot n_t \\
&+ \left(-n^T \underline{K_{s^*w}} \underline{K_{ww}}^{-1} \underline{K_{ws}} |u| + n^T \underline{K_{s^*s}} |u| \right) \tan \varphi
\end{aligned} \tag{1.4.19}$$

On utilise alors la même solution que pour le critère de Rankine ((1.4.7)), seules les expressions de T^e et M ont changé.

1.4.2.3 Comportement d'un matériau à matrice argileuse avec le critère de Mohr-Coulomb

Dans cette partie nous allons montrer les résultats obtenus avec le critère de type Mohr-Coulomb en traction-compression ainsi qu'en compression triaxiale. Une étude plus précise du comportement en compression triaxiale en fonction de la minéralogie sera effectuée dans la partie suivante.

Paramètres de la modélisation

Le domaine utilisé, visible sur la figure 1.15 est un cube de $200\mu m$ possédant des inclusions sphériques. La génération des inclusions est décrite plus loin dans ce chapitre. La matrice est de type argileuse et a pour module Young 20 GPa et pour coefficient de Poisson 0.3. Il y a trois types d'inclusions :

- Le quartz (module Young de 47 GPa, coefficient de Poisson de 0.12 et fraction volumique de 4.6%)
- La calcite (module Young de 55 GPa, coefficient de Poisson de 0.3 et fraction volumique de 13.5%)
- La pyrite (module Young de 85 GPa, coefficient de Poisson de 0.17 et fraction volumique de 0.4%)

À nouveau, seule la matrice peut fissurer, elle a pour paramètre $C=10$ MPa, $\tan(\phi) = 0.6$ et $\mathcal{G}_f = 0.6J \cdot m^2$.

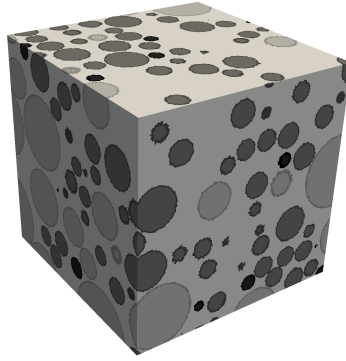


FIGURE 1.15 – Représentation du maillage d'une des minéralogies étudiées

Comportement en traction-compression

Le comportement en traction-compression est visible sur la figure 1.16. On observe un rapport de cinq entre la résistance en traction et la résistance en compression (3 MPa et 15 MPa). Le comportement est fragile

Comportement en compression triaxiale

Concernant le comportement en compression triaxiale, on observe sur la figure 1.17 à nouveau une augmentation de la résistance en fonction du confinement. Le comportement change en devenant plus ductile lorsque l'on augmente la pression de confinement.

Dans le repère de Mohr, figure 1.18, on remarque cette fois ci que le critère macro correspond à un critère de type Mohr-Coulomb malgré la présence d'inclusions élastiques (ici seule la matrice a un critère de type Mohr-Coulomb). La cohésion macro du matériau vaut alors 5 MPa au lieu de 10 MPa pour la cohésion micro. La valeur de $\tan(\phi)$ passe elle de 0.6 pour la valeur micro à 0.56 pour la valeur macro. La présence d'inclusions semble

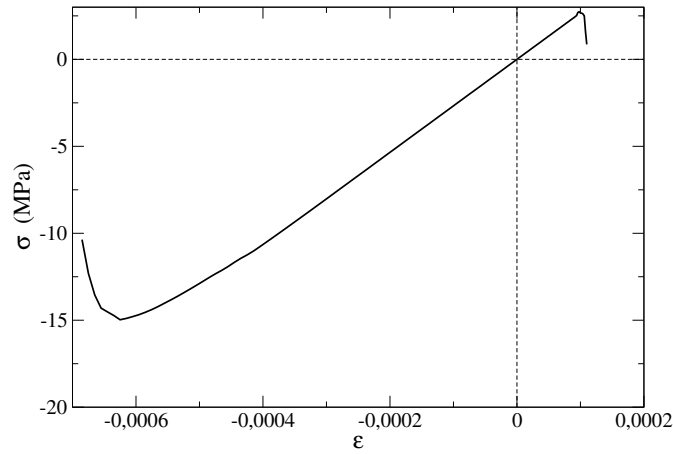


FIGURE 1.16 – Comportement obtenu pour un chargement de traction/compression

diminuer la valeur de la cohésion ainsi que de l'angle de frottement. Nous verrons plus en détail ce phénomène dans la partie suivante.

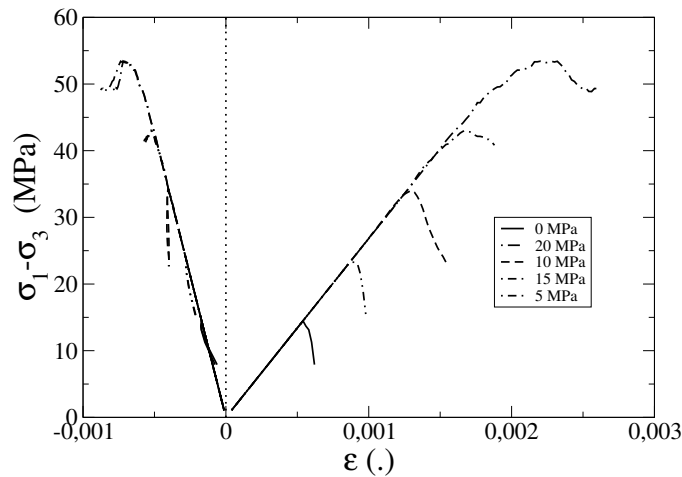


FIGURE 1.17 – Comportement obtenu pour un chargement compression triaxiale

Des calculs de compression triaxiale ont été réalisés avec des méthodes d'homogénéisation [Pastor et al., 2016], dans le cas d'inclusions plus rigides que la matrice et d'inclusions moins rigides que la matrice (matériaux poreux) avec une matrice ayant un comportement de type Mohr-Coulomb et des inclusions élastiques. Les résultats observés dans le cas d'inclusions plus rigides que la matrice correspondent aux résultats observés ici. Le comportement macro est alors de type Mohr-Coulomb et confirme les résultats obtenus avec la méthode E-FEM présentée ici.

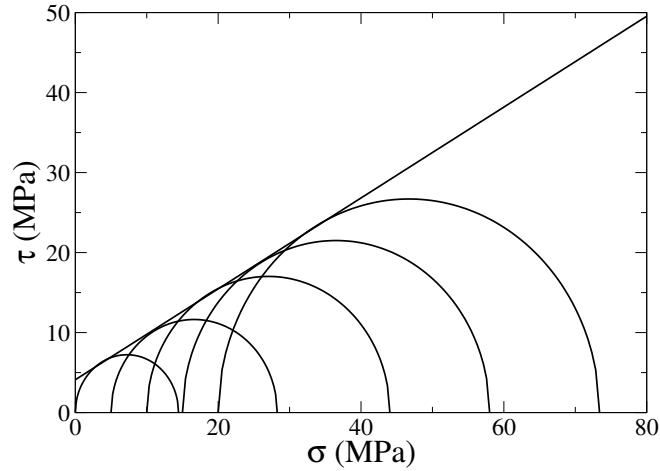


FIGURE 1.18 – Représentation du comportement obtenu dans le repère de Mohr

1.5 Étude de l'impact de la minéralogie

1.5.1 Génération des différentes morphologies

Dans cette section, nous allons comparer les résultats obtenus en faisant varier la minéralogie utilisée dans le cas de la compression dans un premier temps puis de la compression triaxiale dans un second temps. Nous avons utilisé un même maillage avec 72 minéralogies différentes. En effet l'utilisation des discontinuités faibles permet d'avoir un unique maillage et nous faisons ici seulement varier les fractions volumiques pour les calculs en compression et en compression triaxiale (seule une étape de projection des nouvelles inclusions est nécessaire). Ces fractions volumiques sont présentées dans le tableau A.1.

Les inclusions seront modélisées comme étant des sphères. Il aurait été possible d'utiliser la génération de champs aléatoires pour générer les inclusions, cependant le fait d'avoir plusieurs types d'inclusions et des fractions volumiques élevées peut être problématique pour cette méthode de génération d'inclusions.

Pour pouvoir générer différentes minéralogies nous avons réalisé un code de placement aléatoire des sphères. Ce code est décrit dans la partie suivante.

1.5.1.1 Placement aléatoire de sphères polydisperses

Nous avons ici besoin d'un algorithme permettant de placer aléatoirement des sphères polydisperses. Il existe dans la littérature un grand nombre de méthode de placement aléatoire.

- la méthode de sédimentation [Tory et al., 1973] : on place une à une les sphères en vérifiant à chaque placement que la sphère ne touche pas une autre sphère. Cette algorithme permet d'avoir des sphères de diamètres différents mais ne permet pas d'avoir des fractions volumiques élevées. De plus cette méthode s'avère lente et non parallélisable.
- la méthode de Jodrey-Tory [Jodrey and Tory, 1979] : cette méthode est utilisable uniquement pour des sphères de mêmes diamètres. Dans un premier temps un nuage

de points est créé, chaque point correspondant à une sphère. À chaque itération on sélectionne les deux sphères ayant le plus grand chevauchement et on annule celui-ci en déplaçant les deux sphères suivant l'axe défini par leur centre. Cette méthode demande un grand nombre d'itérations pour converger (plus de treize millions d'itérations sont nécessaires pour un placement de mille sphères).

- la méthode de réarrangement collectif des sphères : cette méthode fait partie de la famille des algorithmes basés sur des forces et peut être vue comme une amélioration de la méthode de Jodrey-Tory. Elle a été développée dans un premier temps pour des sphères de même diamètre [Mocinski and Bargiel, 1991] et a été améliorée [Bezrukov et al., 2002] afin de pouvoir avoir des sphères de diamètres différents. Cette méthode permet donc l'utilisation de sphères polydisperses, elle est rapide et peut être parallélisée. Nous avons choisi d'utiliser cette méthode que nous allons décrire plus précisément.

Réarrangement collectif des sphères

La première étape de l'algorithme est de générer aléatoirement la place des N sphères. Ces sphères sont placées à l'intérieur d'un domaine défini (un parallélépipède). On a donc à cette étape un groupe de N sphères $b(r_i, d_i)$ avec r_i les coordonnées du centre de la sphère i et d_i le diamètre de la sphère i . Les sphères se chevauchent pendant cette étape initiale.

L'étape suivante est une succession d'itération où l'on va déplacer les sphères jusqu'à ce qu'il n'y ait plus de chevauchement entre les sphères. Cette équation est une version permettant l'utilisation de différents rayons, une première version de l'algorithme présent dans [Mocinski and Bargiel, 1991] permet uniquement d'utiliser des rayons de même taille. Le déplacement d'une sphère est calculé comme suit :

$$\delta r_{i+} = \frac{r_j - r_i}{\|r_j - r_i\|_2} d_j \left(\frac{\|r_j - r_i\|_2^2}{(d_i + d_j)^2} - 1 \right) \quad (1.5.1)$$

Pour une sphère i on doit réaliser le calcul du déplacement avec chacune des sphères j . Ce calcul peut être réalisé en parallèle. Cette parallélisation a été réalisée à l'aide de la programmation orientée composant en utilisant CTL (programmation décrite plus loin). On met ensuite à jour la position de chacune des sphères :

$$r_{i+} = \delta r_i \quad (1.5.2)$$

On itère ensuite jusqu'à ce que les sphères ne se touchent plus, soit :

$$\max_{i,j} (d_i + d_j - \|r_i - r_j\|_2) \leq tol \quad (1.5.3)$$

Étant donné que la discontinuité faible présente dans les tétraèdres ne permet de prendre en compte que deux matériaux différents dans un tétraèdre, chaque tétraèdre ne doit être coupé que par une inclusion (une sphère). Comme on peut le voir sur la figure 1.19 la distance minimale entre les sphères doit être supérieure à la plus grande arête d'un tétraèdre. L'algorithme utilisé permet d'avoir un placement de sphères dont la distance minimale est égale à zéro. Pour pouvoir garantir une distance minimale non nulle au

début de la génération des sphères on augmente leurs rayons de $\frac{l_{max}}{2}$. On utilise ensuite le logarithme pour assurer un non chevauchement des sphères, une fois cette partie effectuée on réduit le rayon des sphères de $\frac{l_{max}}{2}$.

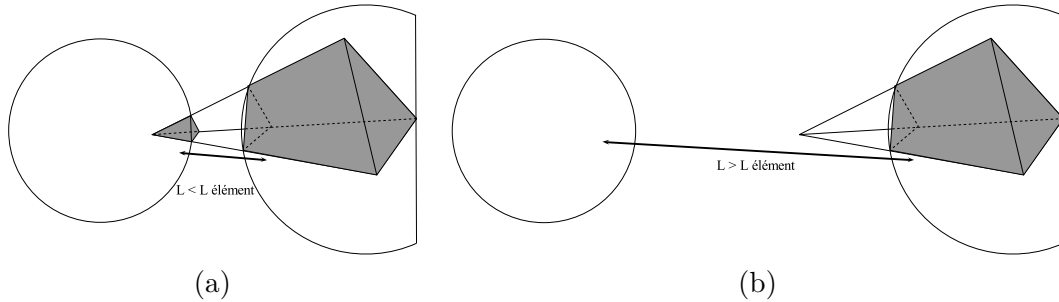


FIGURE 1.19 – Représentation d'un tétraèdre coupé par (a) deux sphères, (b) 1 sphère

Sur la figure 1.20 on peut voir deux configurations obtenues. Dans la première configuration les sphères ont le même rayon et la fraction volumique est élevée. On observe que la répartition des sphères se rapproche d'un motif cristallin. Dans le second cas le rayon des sphères varie, la compacité obtenue ici est plus importante et peut atteindre 70%.

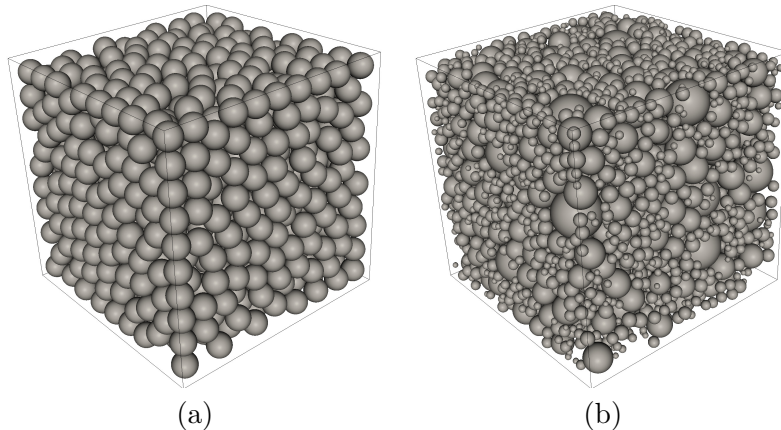


FIGURE 1.20 – Réarrangement de sphères (a) monodisperses, (b) polydisperses

1.5.1.2 Projection des sphères sur un maillage

Une fois l'étape de création des sphères terminée, la dernière étape consiste à projeter les sphères sur notre maillage tétraédrique (obtenu à l'aide de l'outil GMSH [Geuzaine and Remacle, 2009]). Comme nous avons pu le voir précédemment nous avons besoin de deux informations pour les discontinuités faibles. La première information est l'orientation de l'interface et la deuxième est le ratio volumique (entre l'une des deux phases et le volume total du tétraèdre).

Pour obtenir ces informations nous cherchons dans un premier temps si le tétraèdre est coupé par une sphère. Si le tétraèdre n'est pas coupé par une sphère il n'y a pas de

discontinuité faible (et donc pas d'informations à donner). Dans le cas où le tétraèdre est coupé on cherche dans un premier temps l'intersection de la sphère avec chacune des arêtes du tétraèdre. Deux cas peuvent apparaître :

- Si il y a trois arêtes coupées les trois points d'intersection entre la sphère et les arêtes forment un plan. Dans ce cas on peut récupérer la normale au plan qui correspond à l'orientation de l'interface. Pour calculer le ratio de volume on l'obtient en calculant le volume du tétraèdre coupé en deux (le tétraèdre initial T_{ini} est coupé en une pentaèdre P_{cut} et un tétraèdre T_{cut}). Le ratio vaut alors $ratio = \frac{V_{cut}}{V_{ini}}$ ou $ratio = 1 - \frac{V_{cut}}{V_{ini}}$ selon si le tétraèdre obtenu est dans l'inclusion ou dans la matrice (voir figure 1.21.a et 1.21.b).
- Si il y a quatre arêtes coupées, les quatre points d'intersection entre la sphère et les arêtes ne forment pas un plan, ce cas correspond à la figure 1.21.c. On calcule alors la normale à l'interface comme étant la normale au plan moyen. Pour le calcul du ratio de volume, le tétraèdre est coupé en deux pentaèdres. On utilise le logiciel qhull [Barber and Huhdanpaa, 1996] pour calculer le volume du pentaèdre dans l'inclusion et on en déduit le ratio.

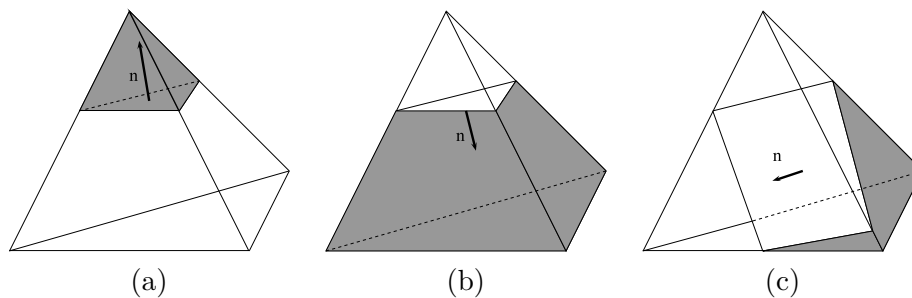


FIGURE 1.21 – Représentation dans le cas de (a) trois arêtes coupées avec le tétraèdre dans l'inclusion, (b) trois arêtes coupées avec le tétraèdre dans la matrice (c) quatre arêtes coupées

On réalise ensuite cette projection sur tous les tétraèdres. On peut voir sur la figure 1.22 un exemple de création de sphères (on a dans ce cas trois types d'inclusions, le quartz, la calcite et la pyrite, représentés par différentes couleurs) puis la projection de ces sphères sur un maillage. On dénombre dans ce cas sept types de tétraèdres :

- Les tétraèdres entièrement dans la matrice
- Les tétraèdres entièrement dans le quartz
- Les tétraèdres matrice/quartz
- Les tétraèdres entièrement dans la calcite
- Les tétraèdres matrice/calcite
- Les tétraèdres entièrement dans la pyrite
- Les tétraèdres matrice/pyrite

1.5.2 Résultats obtenus

Compression simple

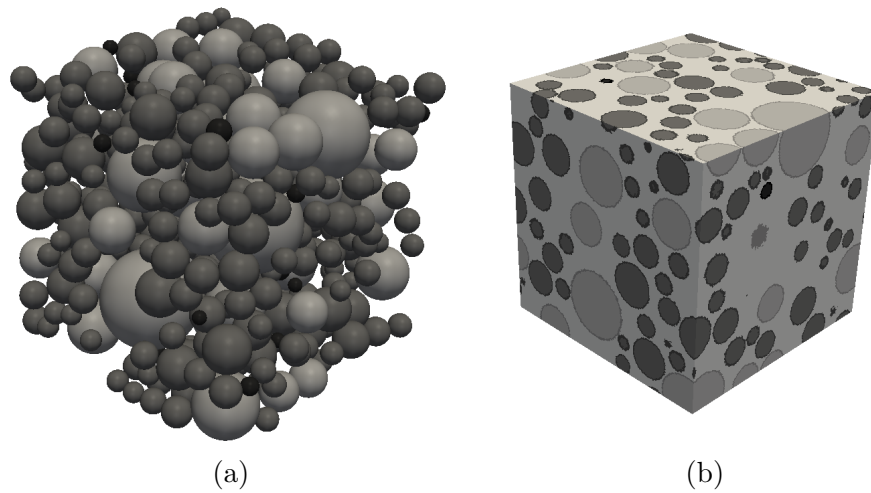


FIGURE 1.22 – Projection de sphères sur un maillage tétraédrique (a) sphères à projeter, (b) projection obtenue

Nous avons dans un premier temps effectué des calculs en compression simple. Les fractions volumiques vont de 0% à 48%, on peut dans un premier temps observer l'évolution du module Young en fonction des fractions volumiques sur la Figure 1.23. On note que pour les fractions de volume total allant de 0 % à 48% la variation du module de Young semble être une fonction linéaire de la fraction volumique totale.

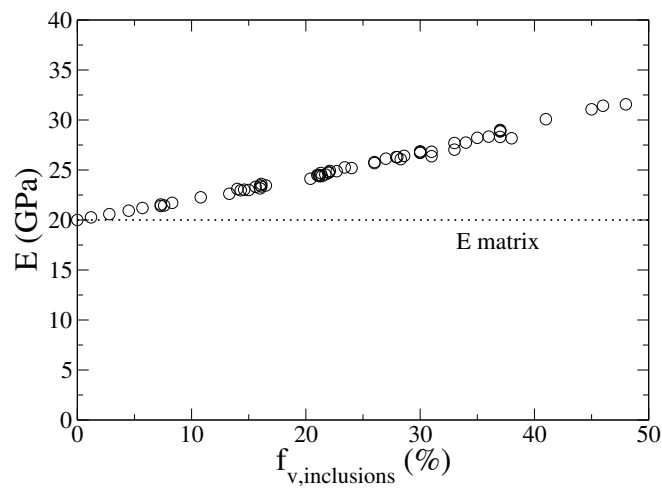


FIGURE 1.23 – Module Young en fonction de la fraction volumique totale d'inclusions

Des résultats en compression simple des différentes minéralogies peuvent être observés sur la figure 1.24. On remarque dans un premier temps que le comportement est le même pour toutes les minéralogies et est de type fragile. Si l'on regarde la résistance en compression on remarque une disparité en fonction de la minéralogie observée (allant de 15 MPa à 20 MPa).

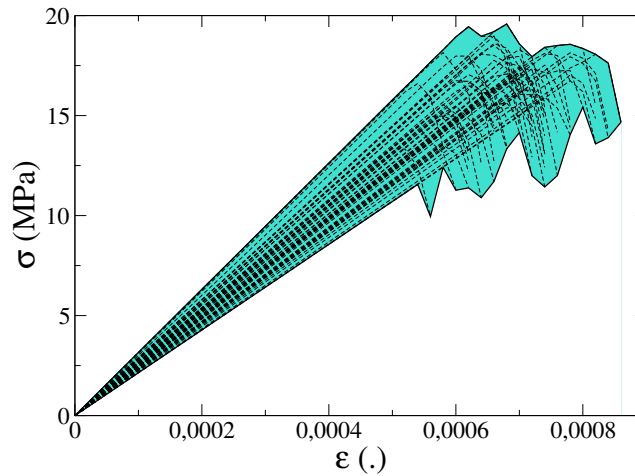


FIGURE 1.24 – Comportement en compression obtenu pour les différentes minéralogies

Compression triaxiale

Des calculs en compression triaxiale ont été réalisés avec les mêmes minéralogies que précédemment pour différentes pressions de confinements (0 MPa, 5 MPa, 10 MPa), des calculs avec une pression de confinement plus importante ont été réalisés uniquement avec certaines minéralogies. On peut voir les résultats obtenus sur la figure 1.25.

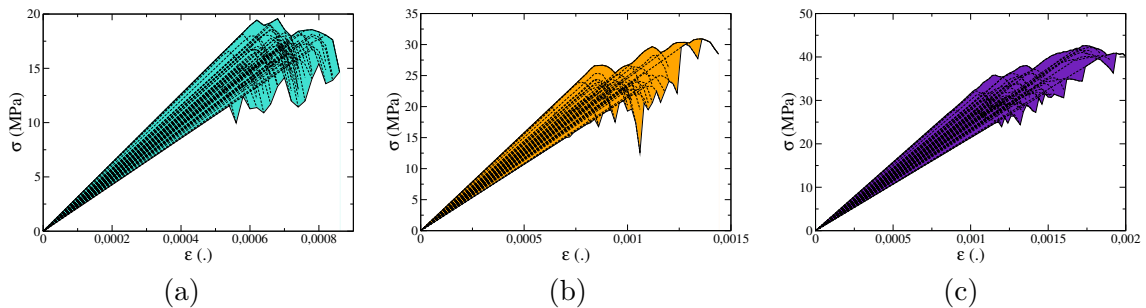


FIGURE 1.25 – Comportement en compression tri-axiale pour (a) 0 MPa, (b) 5 MPa, (c) 10 MPa

On observe dans un premier temps une grande variabilité du comportement obtenu en compression triaxiale en fonction des différentes minéralogies étudiées.

Le comportement fragile-ductile ne semble pas impacté par la différence de minéralogie. On observe en effet que pour un faible confinement le comportement est de type fragile alors que lorsque la pression de confinement augmente le comportement devient de plus en plus ductile.

En traçant les cercles de Mohr pour chacune des minéralogies on note que l'on retrouve à chaque fois un comportement macro de type Mohr Coulomb. On peut observer sur la figure 1.26 les cercles de Mohr obtenus pour trois minéralogies différentes. Il est intéressant de noter ici que les cohésions et les angles de frottements obtenus en traçant les cercles de

Mohr sont différents pour chacune des trois minéralogies.

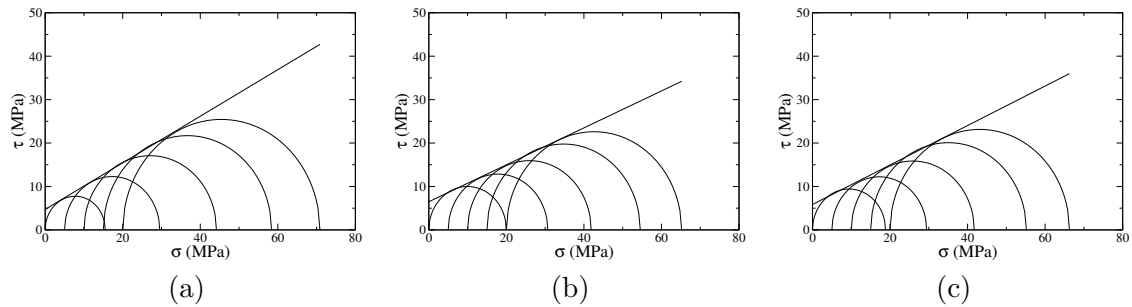


FIGURE 1.26 – Cercles de Mohr pour différentes minéralogies (a) C14_Q16_P1, (b) C20_Q15_P1 et (c) C25_Q11_P1

Nous pouvons maintenant déterminer les cohésions et les angles de friction macro en fonction des différentes minéralogies.

Sur la Figure 1.27 on a tracé l'évolution de la minéralogie en fonction des différentes fractions volumiques.

Dans un premier temps nous pouvons noter qu'il y a une importante variation dans la cohésion obtenue (de 4 MPa à 10 MPa). Il ne semble pas y avoir de corrélation direct entre la cohésion et chacune des fractions volumiques obtenues. Il semble difficile ici de pouvoir établir une relation entre la cohésion et la minéralogie.

Concernant la variation de l'angle de frottement, qui est tracée sur la Figure 1.28, les conclusions sont les mêmes. On observe à nouveau une grande disparité (de 0.6 à 0.38) et il semble difficile de trouver une relation entre la minéralogie et l'angle de frottement.

Dans le cas des courbes de la cohésion en fonction de la fraction volumique la variabilité peut être expliquée par la variation du ratio entre les différentes inclusions de chaque minéralogie. En effet pour une fraction volumique totale équivalente on peut avoir plusieurs minéralogies différentes.

Nous avons réalisé des calculs avec un seul type d'inclusions en faisant varier la fraction volumique correspondante. Les résultats de ce calcul sont disponibles sur la figure 1.29.

On observe le même comportement pour les deux types d'inclusions. La cohésion diminue rapidement lorsque la fraction volumique augmente (passant de 10 MPa à environ 5MPa). Ceci s'explique par le phénomène de concentration de contrainte qui peut apparaître autour des inclusions à cause d'une rigidité différente du matériau ainsi que d'un coefficient de Poisson différent.

Pour le quartz la cohésion change peu pour une fraction volumique comprise entre 5 et 25%, la cohésion ne descend pas en dessous de 5.5 MPa. Concernant la calcite la cohésion minimum obtenue est de 4.5 MPa et la cohésion varie plus rapidement que dans le cas de la calcite. Le comportement macro est plus impacté dans le cas où la calcite est présente. La calcite ayant un plus grand module Young (55 MPa contre 47 GPa pour le quartz) les concentrations de contraintes doivent être plus importantes. Le coefficient de Poisson plus

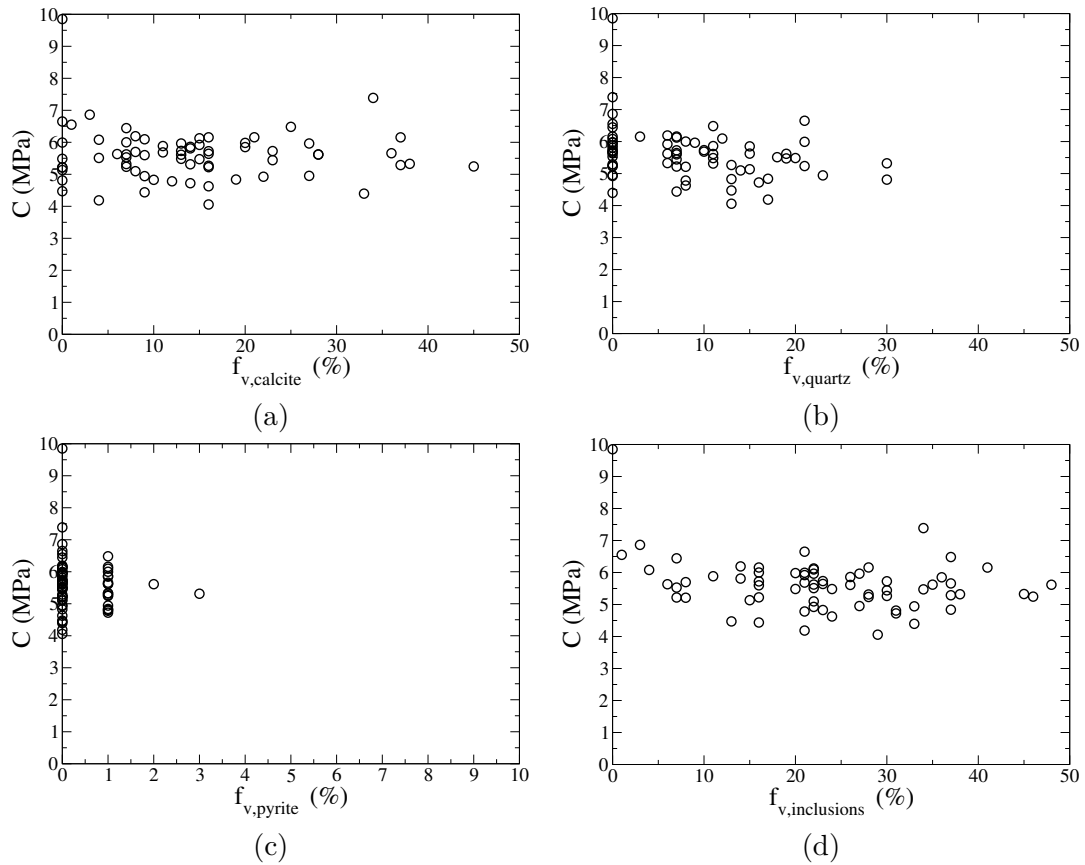


FIGURE 1.27 – Cohésion en fonction de la fraction volumique (a) de calcite, (b) de quartz (c) de pyrite (d) totale

faible du quartz peut aussi expliquer cette différence de résultat.

Dans le but de quantifier l'impact du ratio entre le quartz et la calcite à fraction volumique totale nous avons lancé une série de calculs en faisant varier ce ratio. Ces résultats sont visibles sur la Figure 1.30.

La cohésion macro obtenue est plus faible lorsqu'il y a la présence des deux inclusions (avec une part plus importante de calcite) La plus faible valeur est en effet obtenue lorsque la fraction de quartz est de 20% et celle de calcite de 80%.

Lorsque les deux inclusions sont présentes le champ de contrainte devient fortement hétérogène et d'importantes concentrations de contraintes peuvent alors apparaître et amener plus rapidement à la ruine du matériau.

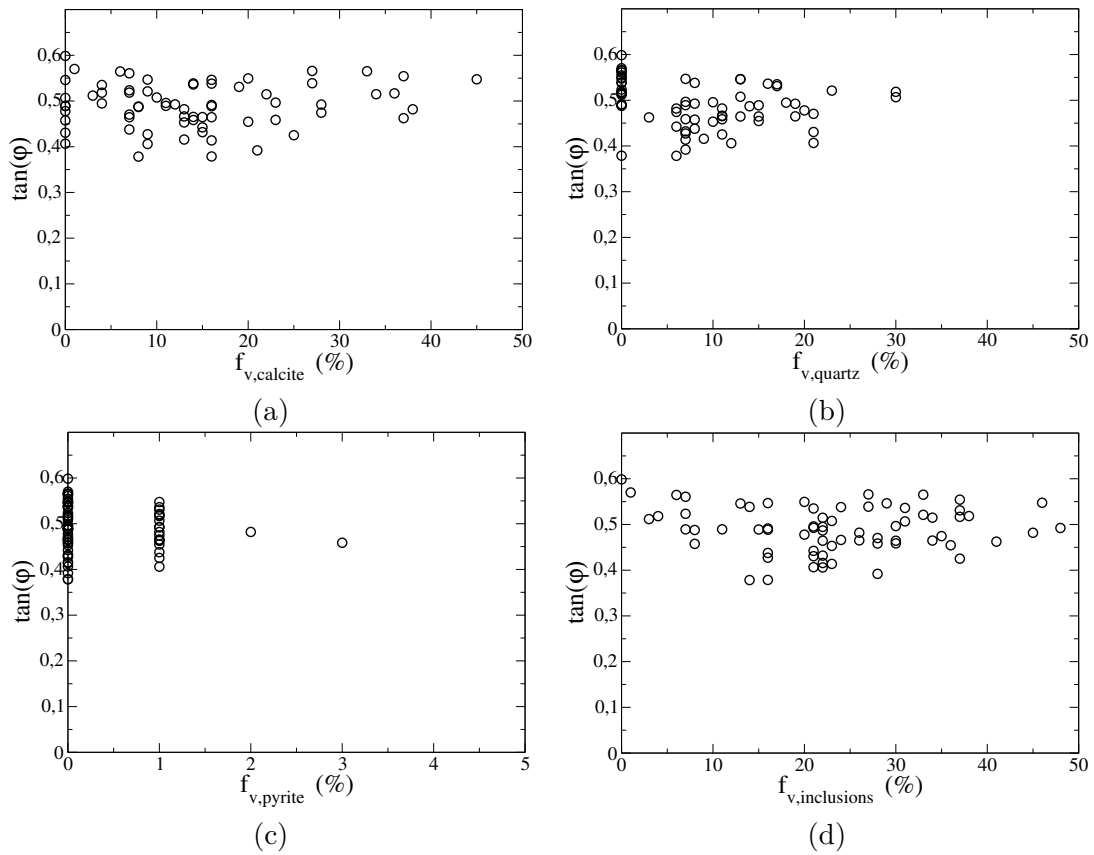


FIGURE 1.28 – $\tan \varphi$ en fonction de la fraction volumique (a) de calcite, (b) de quartz (c) de pyrite (d) totale

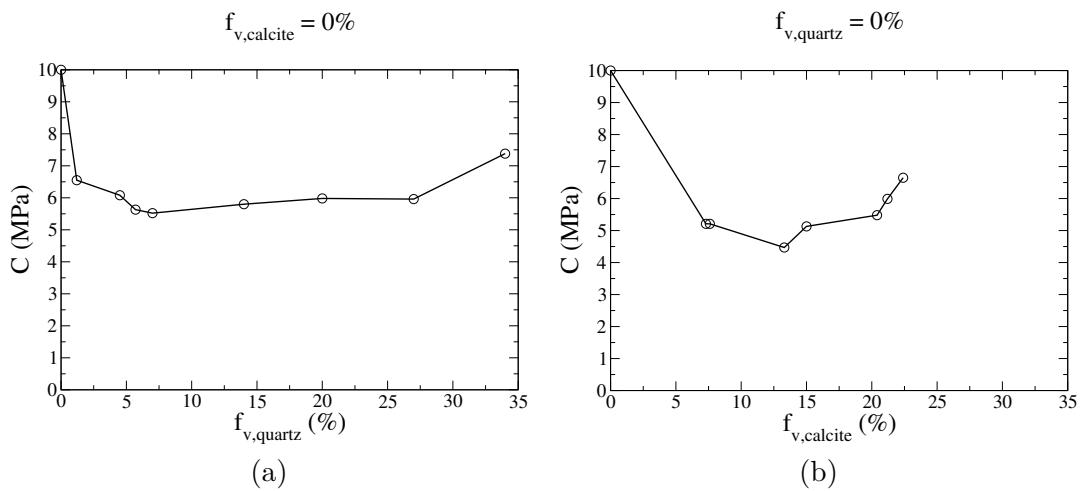


FIGURE 1.29 – Cohésion en fonction (a) de la fraction volumique de quartz sans calcite (b) de la fraction volumique de calcite sans quartz,

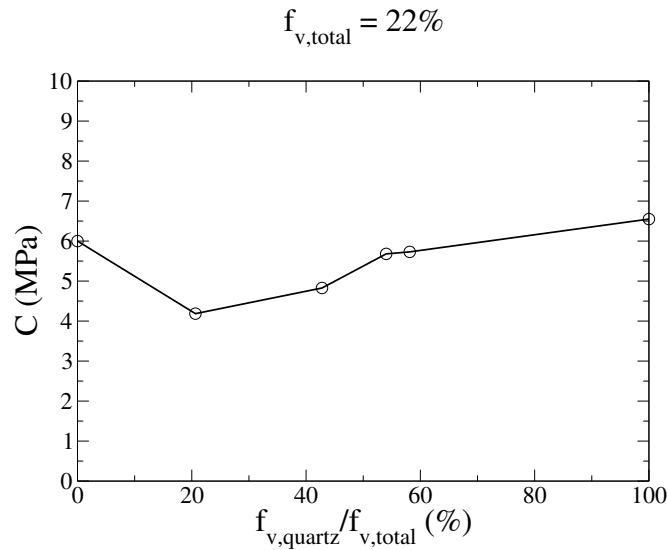


FIGURE 1.30 – Cohésion en fonction du ratio entre la fraction volumique de quartz et celle de calcite

Pour conclure, le comportement d'un matériau hétérogène dépend fortement de la minéralogie de celui-ci. Chaque type d'inclusion semble avoir de l'importance et il paraît compliqué de trouver une relation simple entre la minéralogie et le comportement macro. Un calcul numérique semble nécessaire pour chaque minéralogie dont on souhaite connaître le comportement macro.

1.5.3 Évolution de la fissuration

L'étude du faciès de fissuration est nécessaire à la bonne connaissance du comportement du matériau. Dans cette partie nous allons étudier l'apparition des fissures pour une minéralogie. La minéralogie choisie est composée de 20% de quartz, 15% de calcite et 1% de pyrite. Le maillage obtenu est visible sur la figure 1.31. La représentation du maillage est faite avec et sans la matrice. L'image sans la matrice permet de se rendre compte de la compacité du maillage ainsi que de la distance devant être imposée entre les sphères.

Sur la figure 1.32 on peut voir la réponse obtenue à une sollicitation de compression triaxiale. La résistance pour une pression de confinement nulle est de 15 MPa. On remarque que lorsque la pression de confinement augmente le comportement devient de plus en plus ductile, ce phénomène a déjà été observé sur les différentes minéralogies présentées dans la partie précédente.

Une représentation de ces résultats est visible sur la figure 1.33. On note à nouveau qu'avec une matrice non linéaire de type Mohr Coulomb et des inclusions élastiques plus rigides que la matrice, le comportement macro obtenu est à nouveau de type Mohr Coulomb. Pour cette minéralogie les paramètres macroscopiques obtenus sont de 5.7 MPa pour la cohésion et de 0.39 pour $\tan(\phi)$.

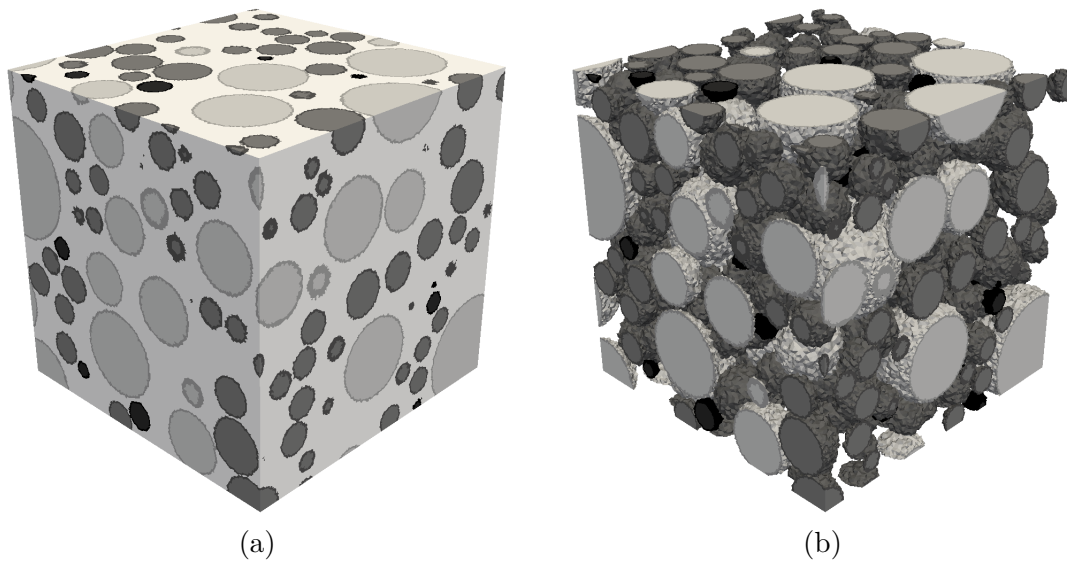


FIGURE 1.31 – Maillage de la minéralogie réalisée (a) maillage complet, (b) maillage sans la matrice

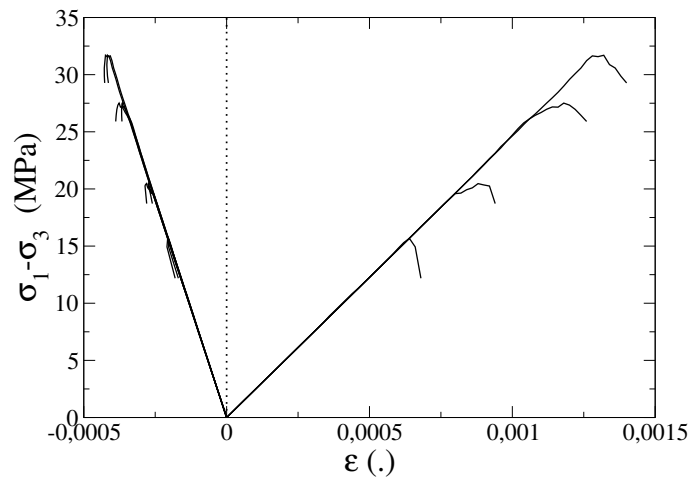


FIGURE 1.32 – Réponse à un chargement de compression triaxiale

La figure 1.34 représente l'évolution de la distance moyenne entre les nouvelles fissures et les anciennes fissures en fonction de la déformation. Ces courbes sont tracées pour différentes valeurs de confinement. On peut observer dans un premier temps que la distance moyenne est importante, ceci indique que les fissures apparaissent dans un premier temps dans les lieux où les concentrations de contraintes sont importantes. Dans un second temps on remarque que la distance diminue rapidement, il y a petit à petit coalescence des fissures. Pour le cas de plus fort confinement, soit 15 MPa, la distance moyenne est moins importante et semble ne pas beaucoup varier. On peut aussi noter que les fissures apparaissent dès le début de calcul, ceci indique que la valeur de confinement est proche de la valeur à

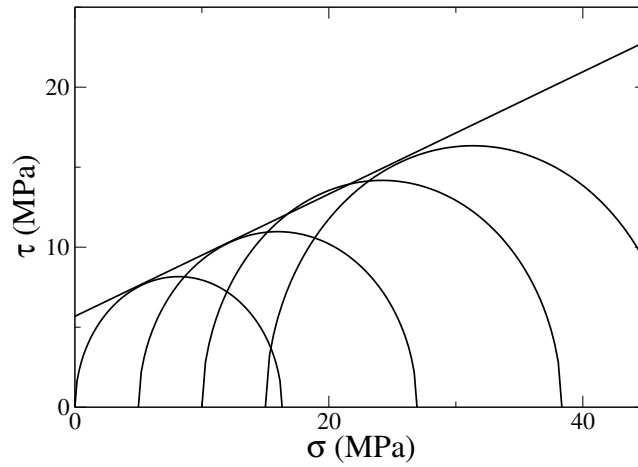


FIGURE 1.33 – Réponse à une sollicitation de compression triaxiale dans le repère de Mohr

rupture du matériau en tri-compression.

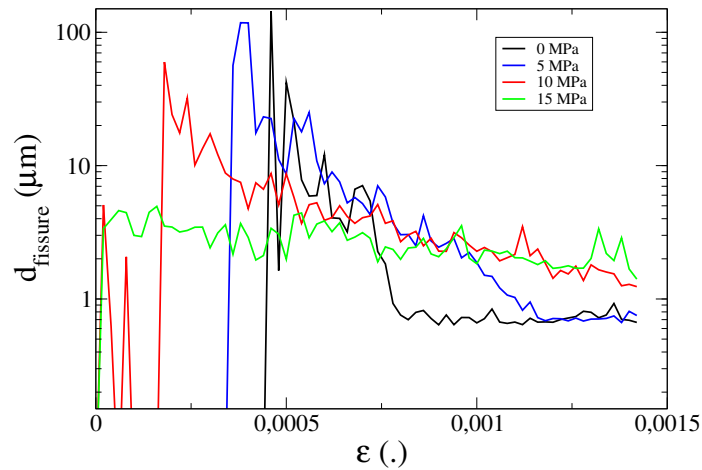


FIGURE 1.34 – Distance moyenne d'apparition des fissures par rapport aux anciennes fissures

La distance d'apparition des fissures par rapport aux inclusions est une donnée intéressante. Celle-ci est tracée sur la figure 1.35. Pour chaque déformation on calcule la moyenne des distances minimales entre les inclusions et les nouvelles fissures. Cette donnée permet de savoir si les fissures apparaissent dans un premier temps près des inclusions, et sont donc directement liées à la présence de celles-ci. On observe pour les différents pressions de confinement que la distance calculée est faible (inférieure à 1 micron) pour de faibles déformations. Cela signifie que les fissures apparaissent dans un premier temps aux bords des inclusions. Il faut noter que la distance est ici calculée entre les centres de gravités de tétraèdres et que la taille moyenne des tétraèdres est d'environ un micron, une distance d'un micron correspond alors à deux tétraèdres qui se touchent.

Pour chacun des confinements on note l'apparition d'un pic de distance pour différentes

valeurs de déformation. Sur le graphique les points en pointillés représente la déformation imposée lors de la rupture. Le pic correspond pour les trois premières pressions de confinement à la déformation à rupture. Pour le cas de la pression de confinement la rupture n'étant pas totalement atteinte et ne peut donc pas être tracée. Au moment de la rupture un grand nombre d'éléments fissurent. Ceci explique que la distance augmente temporairement, en effet il y a à ce moment du calcul une coalescence importante et l'apparition de macro fissures.

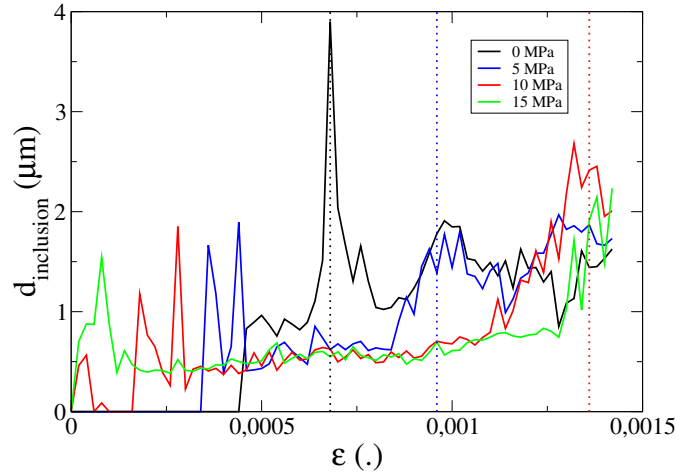


FIGURE 1.35 – Distance moyenne d'apparition des fissures par rapport aux inclusions

1.6 Étude de l'impact du kérogène

Le but de la fracturation hydraulique est d'extraire le kérogène inclus dans la matrice argileuse. La modélisation précédente ne prenait pas en compte la présence de kérogène dans le matériau, de plus la fraction volumique de la matrice argileuse était trop importante par rapport à celle pouvant être mesurée in situ (on observe entre 20% et 40% d'argile). Dans le travail présenté ici la matrice argileuse englobe tout ce qui n'est pas soit du quartz, de la calcite ou de la pyrite, la fraction volumique de la matrice argileuse visée sera donc un peu supérieure à celle observable in situ.

Dans l'étude précédente, les fractions volumiques minimales d'argile obtenues (environ 50%), représentent la fraction minimum de la matrice et donc la fraction maximum d'inclusions sphériques dans le cas où celles-ci ne se chevauchent pas. Pour atteindre des fractions volumiques plus importantes pour les inclusions nous allons autoriser le chevauchement des sphères. Cependant, afin de vérifier la présence au maximum de deux matériaux par élément tétraédrique, seuls les sphères de même matériau pourront se chevaucher.

Trois générations de morphologie sont représentées sur la figure 1.36. Une des difficultés de cette méthode de génération est que l'on a aucun contrôle sur les fractions volumiques obtenues après génération. Pour obtenir des fractions volumiques de matrice inférieures à 30%, il a fallu insérer un volume de sphères représentant trois fois le volume total, ceci n'a évidemment aucun sens physique et montre la limite de la méthode employée ici. Ce

manque de contrôle est un défaut majeur ici. L'étude se limitera donc à un petit nombre de morphologies.

La génération de morphologies avec un pourcentage important d'inclusions montre les limites de l'approche choisie dans le travail présenté dans ce document.

Le kérogène est modélisé par un groupement de sphères. Celui ci est considéré dans cette étude comme ayant un comportement élastique linéaire. Dans la littérature les propriétés du kérogène varient grandement, afin de connaître l'impact de cette variation sur les propriétés mécaniques nous avons réalisé des calculs avec différents paramètres mécaniques pour le kérogène. Les propriétés utilisées pour sa modélisation sont les suivantes [Bousige et al., 2016].

- Module Young : 1 GPa, 5 GPa et 10 GPa
- Coefficient de Poisson : 0.17 et 0.40
- Dimension : $10\mu m$
- Fraction volumique : 3%, 5% et 10%

La fraction volumique choisie pour le calcul correspond à une fraction massique de TOC (Total Organic Content) varie de 3% à 10%, ce qui est une valeur typique. Pour ce qui est de la dimension, la taille des kérogènes peut varier du nanomètre au millimètre. Étant donné la grande disparité de dimension, la taille retenue est celle permettant de générer au mieux nos inclusions.

Trois générations d'inclusions sont présentées sur la figure 1.36. La matrice argileuse est représentée par la couleur grise la plus claire. Puis, par ordre de niveaux de gris, on retrouve le quartz, la calcite, la pyrite et enfin le kérogène en noir. Les inclusions ont maintenant des formes complexes dont la taille peut être très éloignée de la taille des sphères générées.

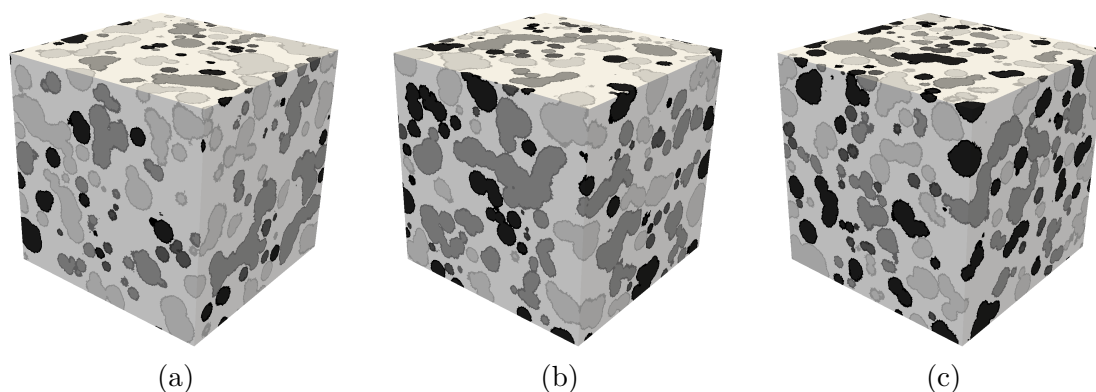


FIGURE 1.36 – Génération d'inclusions pour (a) 22% de quartz 33% de calcite 5% de pyrite et 3% de kérogène ,(b) 22% de quartz 32% de calcite 5% de pyrite et 5% de kérogène et (c) 10% de quartz 32% de calcite 8% de pyrite et 10% de kérogène

Pour la dernière morphologie présentée sur la figure 1.36, des vues séparées de la matrice et des inclusions ont été réalisées. Ces vues sont visibles sur la figure 1.37. En observant la figure 1.37.a, correspondant à la matrice seule, on remarque que la matrice semble totalement connectée. Ceci est dû à la distance minimale nécessaire entre deux inclusions de matériaux différents. Chaque inclusion étant entourée de matrice, celle-ci ne peut pas être percolée. Les inclusions sont visibles sur les figures 1.37.b-e, on peut noter qu'un espace est bien présent entre chaque inclusion de matériau différent. La méthode à enrichissement cinématique peut alors être utilisée.

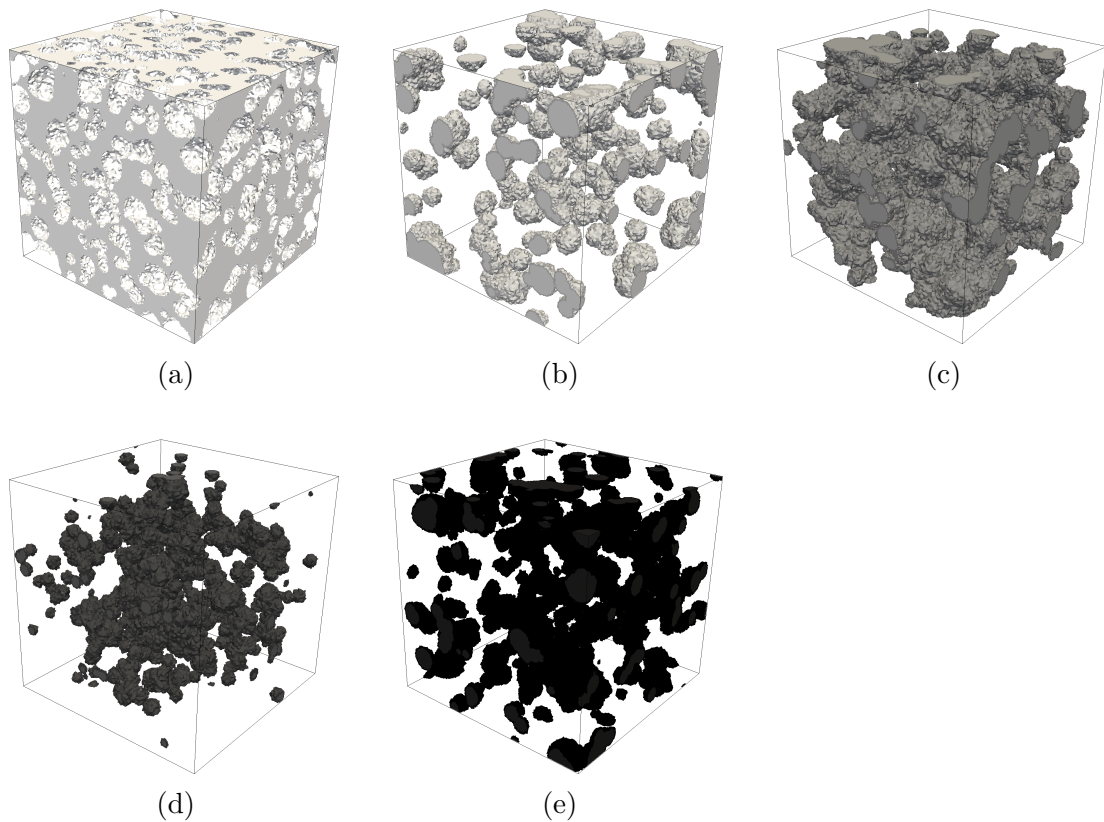


FIGURE 1.37 – Microstructure dans le cas d'une minéralogie avec 10% de quartz 32% de calcite 8% de pyrite et 10% de kérogène, (a) vue de la matrice seule, (b) vue du quartz, (c) vue de la calcite, (d) vue de la pyrite et (e) vue du kérogène

La première étude réalisée est celle de l'impact des paramètres microscopiques du kérogène sur le comportement linéaire macroscopique du VER. Sur la figure 1.38.a le module de compressibilité est tracé en fonction de la fraction volumique de kérogène. Le module varie de 19.2 MPa à 24.3 MPa. La plus grande variation est obtenue lorsque la fraction volumique de kérogène est la plus forte.

En traçant le module de compressibilité macroscopique en fonction du module de compressibilité de kérogène pour les trois fractions volumiques étudiées, visible sur la figure

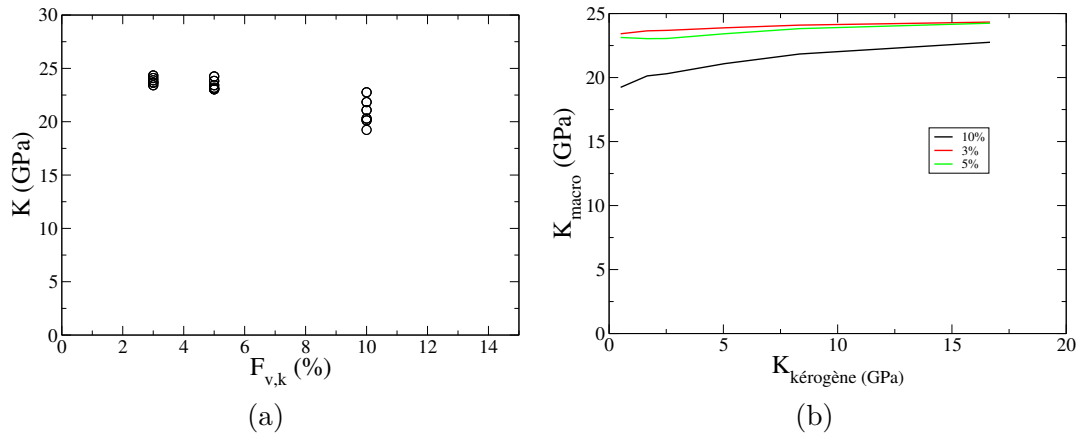


FIGURE 1.38 – Module de compressibilité macroscopique en fonction (a) de la fraction volumique de kérogène, (b) du module de compressibilité du kérogène

1.38.b, on note que plus le module du kérogène est élevé et plus le module macroscopique observée est important. À nouveau plus la fraction volumique de kérogène est élevée plus le phénomène est important.

La même étude peut être réalisée sur le module de cisaillement. Les résultats obtenus dans ce cas sont visibles sur les figure 1.41.a et 1.41.b. Les observations sont similaire à celle effectuées précédemment, cependant on peut noté une plus faible variation du module de cisaillement par rapport au module de compressibilité.

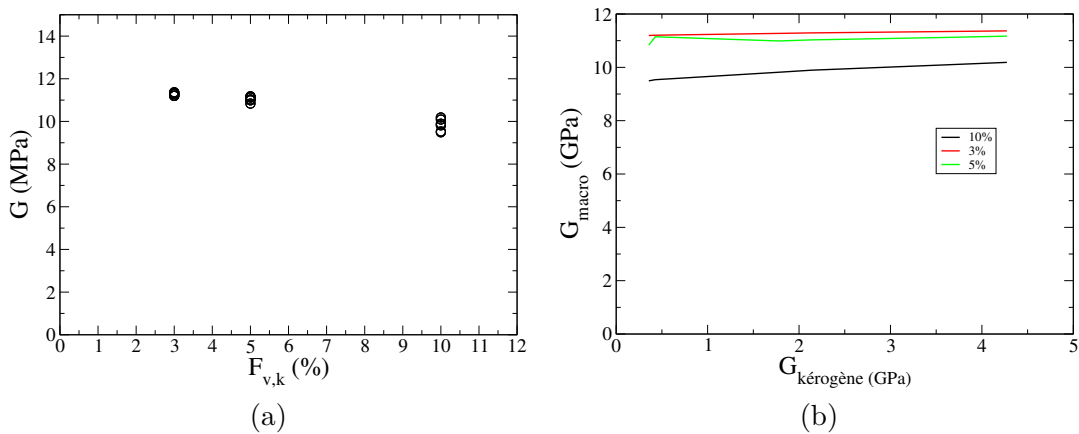


FIGURE 1.39 – Module de cisaillement macroscopique en fonction (a) de la fraction volumique de kérogène, (b) du module de cisaillement du kérogène

La seconde étude porte sur les paramètres non linéaire macroscopique (la cohésion et l'angle de frottement). La déformation obtenue pour les trois pressions de confinements imposées est visible sur la figure 1.40 dans le cas $E=10$ GPa et $\nu=0.17$. Une rupture par cisaillement est observée pour les faibles pressions de confinements. Le cisaillement devient très faible pour la pression de confinement la plus importante.

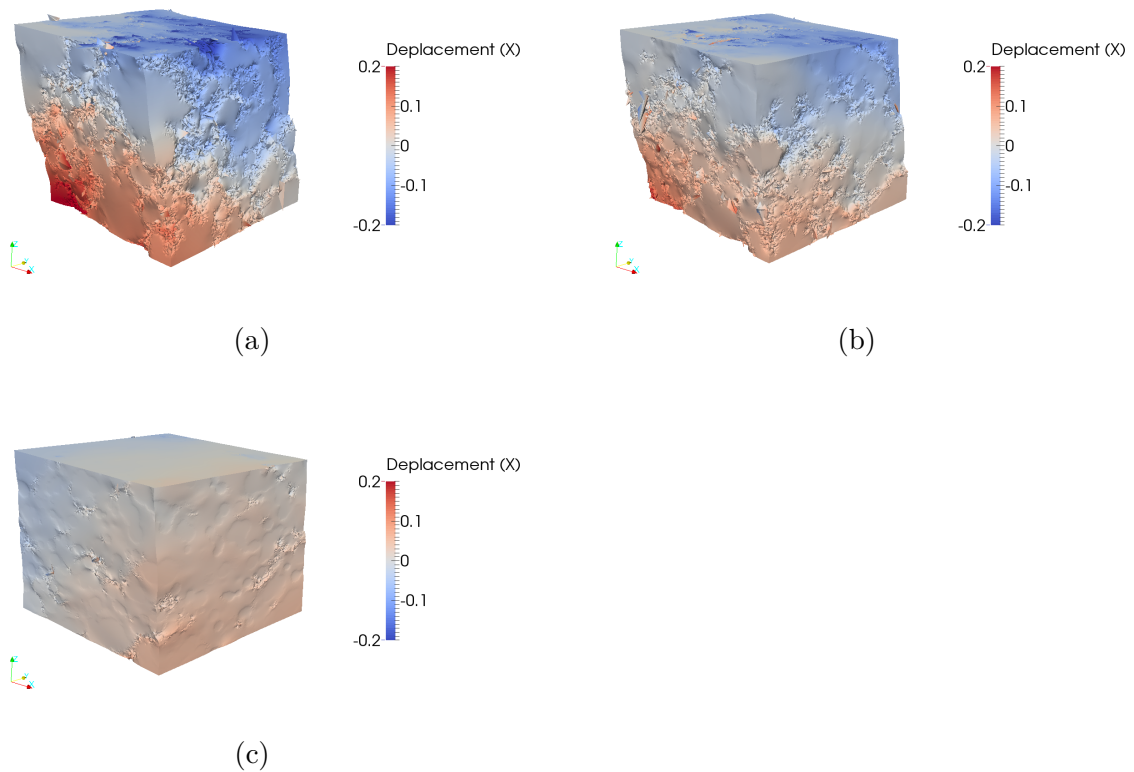


FIGURE 1.40 – Déformation obtenue avec un coefficient d’amplification de 100 pour une pression de confinement de (a) 0 MPa, (b) 5 MPa et (c) 10 MPa

Sur la figure 1.41 sont représenté les différents comportements obtenus pour les jeux de paramètres étudiés. Dans un premier temps on note la présence de peu de résultat pour le cas $E=1$ GPa et $\nu=0.17$, une partie importante des calculs n’a pas pu aboutir pour ce jeu de données, comme nous le verrons dans la suite une faible module Young du kérogène impose une résistance faible du matériau, de même pour le coefficient de Poisson dans une moindre mesure.

Dans un second temps on remarque à nouveau le comportement observé précédemment, plus la pression de confinement est élevée plus le comportement devient ductile. À nouveau, pour obtenir les paramètres macroscopiques tel que la cohésion et l’angle de frottement, ces résultats ont été transposés dans le repère de Mohr pour chaque jeu de paramètres ainsi que que pour chaque minéralogie.

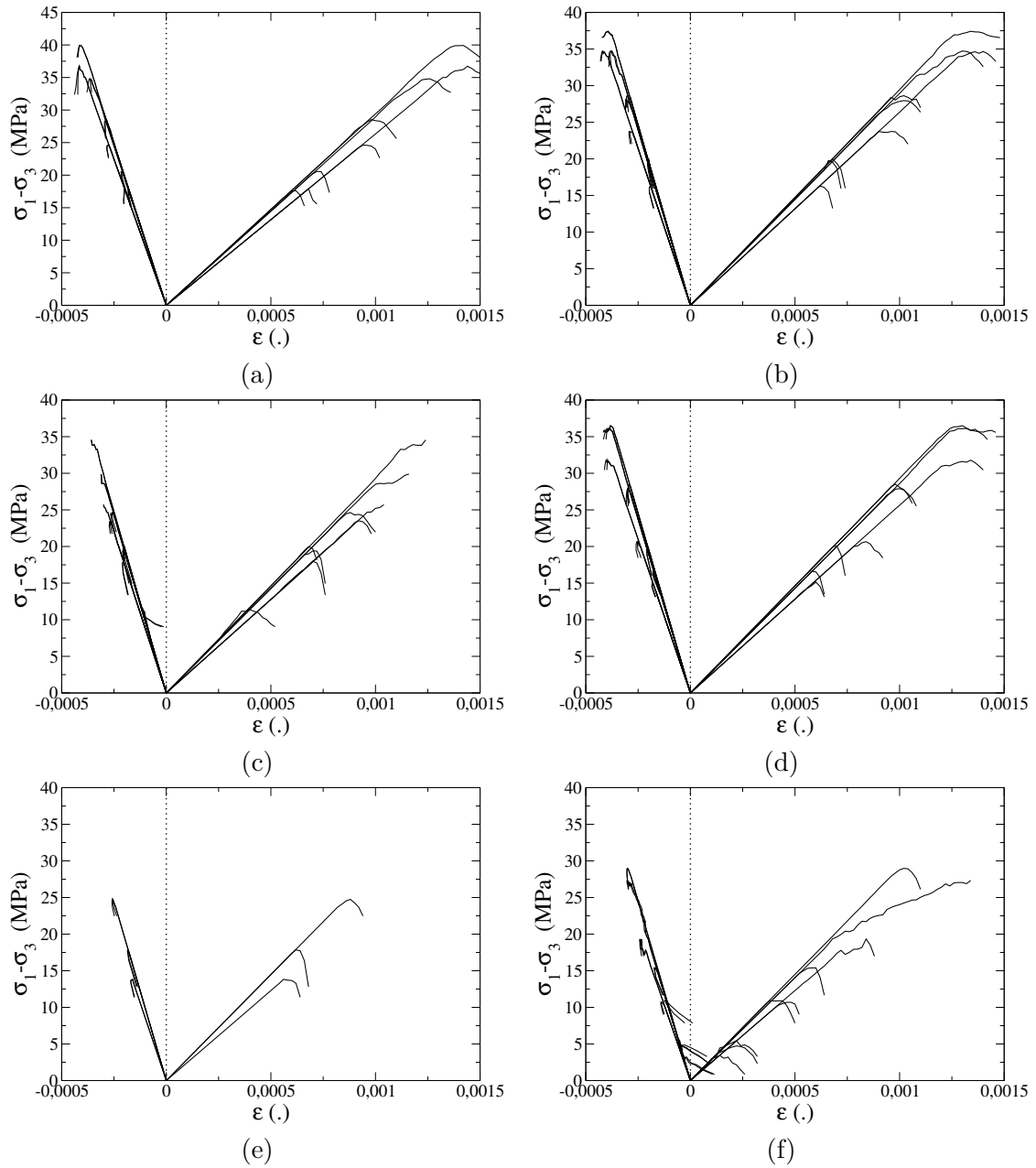


FIGURE 1.41 – Comportement obtenu en compression triaxiale pour (a) $E=10$ GPa et $\nu=0.17$, (b) $E=10$ GPa et $\nu=0.40$, (c) $E=5$ GPa et $\nu=0.17$, (d) $E=5$ GPa et $\nu=0.40$, (e) $E=1$ GPa et $\nu=0.17$, (f) $E=1$ GPa et $\nu=0.40$,

Le comportement obtenu dans le repère de Mohr pour le jeu de paramètre $E=10$ GPa et $\nu=0.40$ est visible sur la figure 1.42. À nouveau en partant d'une matrice ayant un comportement non linéaire de type Mohr-Coulomb avec des inclusions élastiques (plus rigide que la matrice pour la plus part) le comportement macroscopique obtenu est de type Mohr-Coulomb.

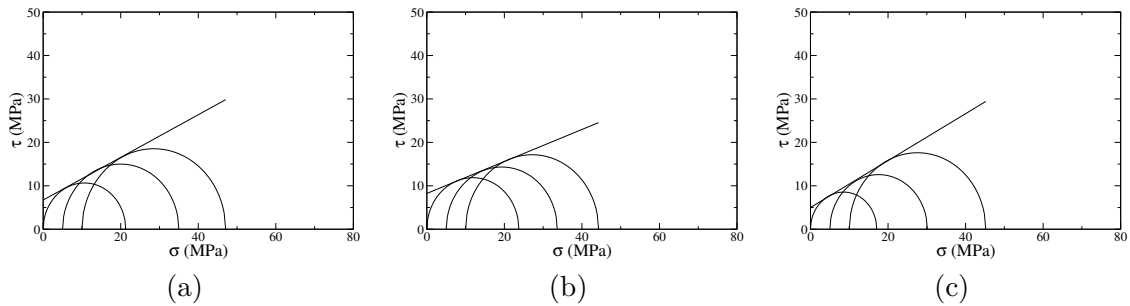


FIGURE 1.42 – Cercles de Mohr pour $E=10$ GPa et $\nu=0.40$ avec (a) 3% de kérogène, (b) 5% de kérogène et (c) 10% de kérogène

En traçant l'évolution de la cohésion en fonction du module de compressibilité du kérogène, visible sur la figure 1.43, on remarque que le lien entre les deux paramètres est complexe. En effet pour chaque module de perméabilité, le module Young ainsi que le module de Poisson correspondant ont été modifiés. En traçant la cohésion en fonction de ces deux paramètres (qui sont les paramètres d'entrées du jeu de données), l'impact est plus visible.

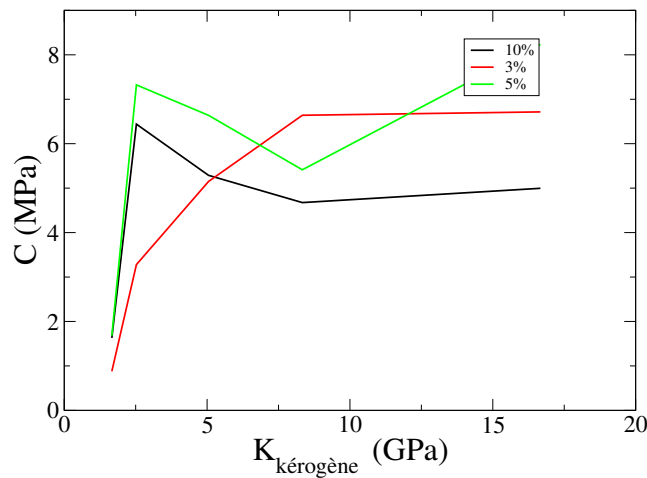


FIGURE 1.43 – Cohésion macroscopique en fonction du module de cisaillement du kérogène

Ces courbes sont visibles sur les figures 1.44.a et 1.44.b. La figure 1.44.a montre un lien entre le module Young du kérogène et la cohésion macroscopique. Plus le premier est important et plus le second l'est aussi. Cette évolution est surtout important entre 1 GPa et 5 GPa. Pour l'évolution en fonction du coefficient de Poisson et la cohésion les points avec un module Young de 1 GPa ont été retirés. On observe en moyenne que lorsque le coefficient de Poisson augmente (et se rapproche de 0.5, ce qui correspond à un matériau dit incompressible) la cohésion augment en moyenne.

L'impact des propriétés mécaniques sur l'angle de frottement semble moins important.

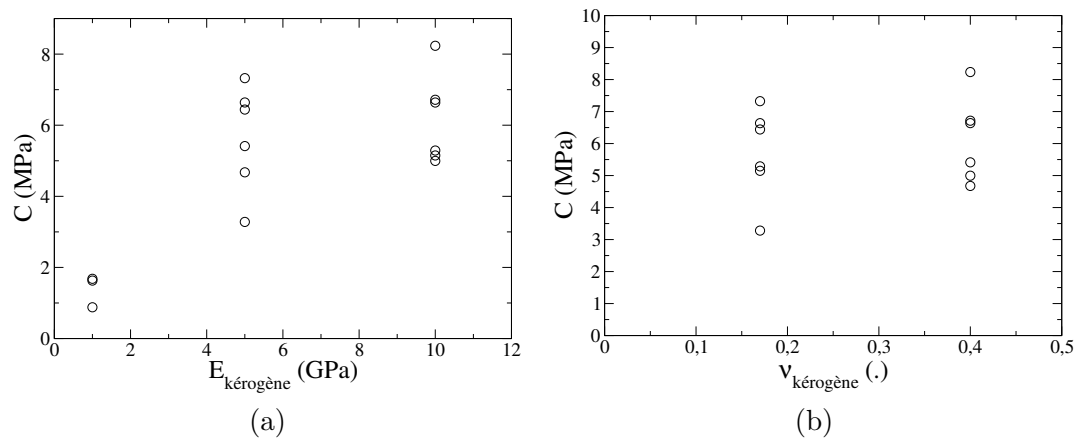


FIGURE 1.44 – Cohésion macroscopique en fonction (a) du module Young du kérogène et (b) du coefficient de Poisson du kérogène

Si l'on regarde la figure 1.45 on ne remarque pas de lien direct entre l'angle de frottement et les deux paramètres mécaniques. La variation de l'angle de frottement semble dépendre uniquement de la minéralogie étudiée.

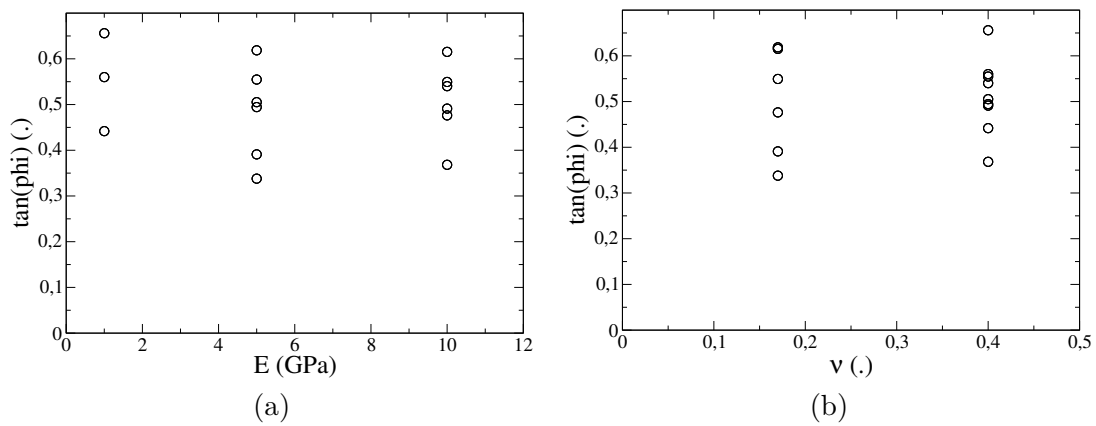


FIGURE 1.45 – $\tan(\phi)$ macroscopique en fonction (a) du module Young du kérogène et (b) du coefficient de Poisson du kérogène

1.7 Conclusions et perspectives

Dans ce chapitre deux modèles E-FEM ont été présentés. Les deux modèles présentent à la fois une discontinuité faible et une discontinuité forte. La différence entre les deux modèles présentés concerne le modèle discret sur la discontinuité.

Dans un premier temps, le modèle discret de type Rankine (mode I) a été présenté. Ce modèle permet de bien simuler le comportement de matériau comme le béton mais ne permet pas une bonne modélisation du comportement des matériaux de type argileux. Cet inconvénient nous a amené à développer le second critère de discontinuité fort, le critère de Mohr-Coulomb en mode II, pour pouvoir modéliser le comportement des shales.

L'utilisation des discontinuités faibles dans les deux modèles nous autorise à utiliser un même maillage et à seulement projeter des inclusions sur le maillage afin de prendre en compte celles ci. Pour générer aléatoirement des inclusions sphériques nous avons utilisé un algorithme de type réarrangement collectif. Cet algorithme nous a permis de réaliser des morphologies avec des fractions volumiques importantes (allant jusqu'à 50%). Cette fraction volumique correspond à un maximum atteint avec notre algorithme, cette limite est en partie due à la distance minimale nécessaire entre deux sphères. En effet les éléments tétraédriques ne disposent que d'une discontinuité faible dans chaque élément. Ainsi on ne peut modéliser que deux matériaux dans un tétraèdre. Si la distance entre deux inclusions était plus faible, les éléments auraient pu contenir plus de deux matériaux.

Afin de palier à ce problème notre algorithme a été modifié pour que des sphères de même matériau puissent se recouvrir. Cette modification a permis la génération de fractions volumiques plus importantes. Il faut cependant noter que nous n'avons plus le contrôle en amont de la fraction volumique finale.

L'étude de différentes minéralogies a permis d'observer la complexité du matériau, due aux nombreuses hétérogénéités le composant. Dans certains cas nous avons été en mesure d'observer un lien direct entre une fraction volumique (ou un ratio de fraction volumique) et le comportement macroscopique de ce matériau. Il nous apparaît nécessaire pour connaître le comportement macroscopique d'un matériau, en connaissant sa composition minéralogique, de réaliser un calcul éléments finis complet. Les résultats obtenus ne nous permettant pas de mettre en relation direct les paramètres d'entrés (la connaissance de la fraction volumique) et les paramètres de sorties (les valeurs macroscopiques).

L'impact de la variabilité des paramètres mécaniques du kérogène a pu être étudié. L'influence de ces paramètres sur les propriétés macroscopiques linéaires et non linéaires du VER a pu être observé. Les résultats obtenus semblent montrer l'importance des propriétés mécaniques du kérogène sur le comportement global du matériau lorsque la fraction volumique de celui ci n'est pas trop faible (supérieur à 5%).

Le premier avantage, de la modélisation réalisée ici, réside au faible nombre de paramètres utilisés. Les paramètres sont communs à chaque calcul réalisé et sont au nombre de deux pour la partie élastique de la modélisation (module de Young et coefficient de Poisson) et de trois pour la modélisation de la fissuration (cohésion, angle de frottement et énergie de fissuration). Un des points clés est que chacun de ces paramètres est un

paramètre physique. Il n'y a donc pas ici besoin de réaliser des calculs préliminaires pour caler de nombreux paramètres. Bien que n'utilisant peu de paramètre, l'utilisation de la méthode E-FEM a permis d'obtenir un comportement complexe du matériau.

Un autre avantage de notre choix d'utiliser la méthode E-FEM est d'avoir accès aux valeurs d'ouvertures ou de glissements des fissures ainsi qu'à leurs orientations. Cet avantage sera utilisé dans la dernière partie du manuscrit et permettra de calculer la perméabilité d'un matériau fissuré.

Chapitre 2

Décomposition de domaine dans un cadre EF non linéaire

Sommaire

2.1	Introduction	58
2.2	Méthodes de décomposition de domaine	59
2.2.1	Formulation d'un problème de décomposition de domaine	60
2.2.2	Méthode primale	63
2.2.3	Méthode duale	64
2.2.4	Autres méthodes	65
2.2.4.1	Méthode hybride	65
2.2.4.2	Méthode à trois champs	65
2.2.4.3	Méthode de mortier	66
2.3	Méthode de mortier appliquée à la mécanique	67
2.3.1	Éléments finis de mortier	67
2.3.2	Formulation de la méthode de mortier appliquée à la mécanique	68
2.3.3	Construction des matrices d'interpolation en 3D	70
2.4	Application de la méthode de mortier à un problème de mécanique	72
2.4.1	Algorithme de résolution linéaire	72
2.4.2	Algorithme de résolution non-linéaire	74
2.4.3	Programmation orientée composant	76
2.4.3.1	Le composant, une brique élémentaire	76
2.4.3.2	Un composant éléments finis, coFeap	76
2.4.3.3	Application à la décomposition de domaine	77
2.4.4	Cas d'une poutre en traction en élasticité	84
2.4.4.1	Présentation de la géométrie et des conditions aux limites	84
2.4.4.2	Résultats obtenus	85
2.4.5	Cas d'une poutre en traction en non linéaire	86
2.4.5.1	Présentation de la géométrie et des conditions aux limites	86
2.4.5.2	Résultats obtenus	87
2.5	Scalabilité du code de calcul	89
2.6	Conclusion	90

2.1 Introduction

Dans le cadre de la modélisation du comportement mécanique autour de la pointe de fissure nous réalisons un essai de fissuration. Ce calcul est effectué sur une plaque (d'environ 10 cm de côté) avec une entaille préexistante dont le matériau contient les mêmes hétérogénéités que celles présentées dans le chapitre précédent. Les inclusions contenues dans le matériau ayant un diamètre d'un ordre de grandeur de $10\mu\text{m}$ pour les plus petites inclusions, le modèle demande un grand nombre de degrés de liberté. Ce calcul n'est pas réalisable avec les moyens informatiques actuels dans un temps raisonnable (le nombre de nœuds pouvant être supérieur à 10^7).

Afin de permettre la réalisation un tel calcul nous utilisons la méthode de décomposition de domaine. Cette dernière permet de découper notre domaine en un certain nombre de sous-domaines, les sous-domaines ont moins de degrés de liberté (les calculs sont ainsi bien plus rapides) et sont informatiquement indépendants (la réalisation du calcul dans chaque sous-domaine peut être effectuée en simultanée). Cependant il faut assurer la continuité des déplacements et des forces aux interfaces de chaque sous-domaine. C'est pourquoi les sous-domaines sont uniquement indépendants informatiquement. Nous pouvons observer sur la figure 2.1 le schéma d'un domaine et de sa décomposition en sous-domaines.

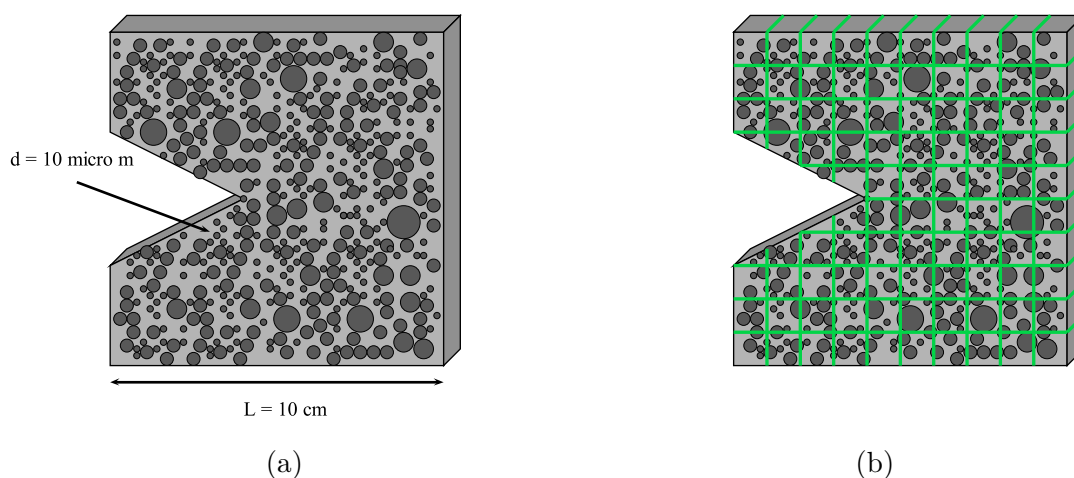


FIGURE 2.1 – Schéma de la décomposition d'un domaine en plusieurs sous-domaines

D'un point de vue général le problème de décomposition de domaine revient à assurer la continuité des forces et des déplacements aux interfaces. On peut dans un premier temps distinguer deux types de méthodes. La différence entre ces deux méthodes provient de la présence ou non des chevauchements des sous-domaines. Ces derniers se chevauchent dans le cas des "overlapping decomposition methods" ([Le Tallec, 1994],[Farhat and Roux, 1991]) à l'inverse dans le cas "non-overlapping decomposition methods" ([Schwarz, 1869],[Rateau, 2003],[Lions, 1988]) ils ne se chevauchent pas.

Dans le travail présenté dans ce manuscrit seul les méthodes sans chevauchement sont

utilisées et plus particulièrement la méthode de mortier [Maday et al., 1989].

Dans le cadre des méthodes sans chevauchement les sous-domaines voisins ont une interface commune (les nœuds du maillage éléments finis de ces sous-domaines peuvent ne pas correspondre).

Un résumé des différentes méthodes est effectué dans [Gosselet and Rey, 2006]. On peut distinguer les approches suivantes :

- Approche primale
- Approche duale
- Autres approches (hybride, à trois champs)

La figure 2.2 présente la décomposition d'un domaine Ω en deux sous-domaines nommés $\Omega^{(1)}$ et $\Omega^{(2)}$. Le travail consiste à assurer la continuité des déplacements (u_1^Γ et u_2^Γ) ainsi que la continuité des forces (λ_1 et λ_2 au niveau de l'interface. Des conditions de type Dirichlet sont appliquées sur une partie de $\Omega^{(1)}$ et des conditions de Neumann sur une partie de $\Omega^{(2)}$. Cet exemple nous servira de base pour expliquer les différentes méthodes existantes.

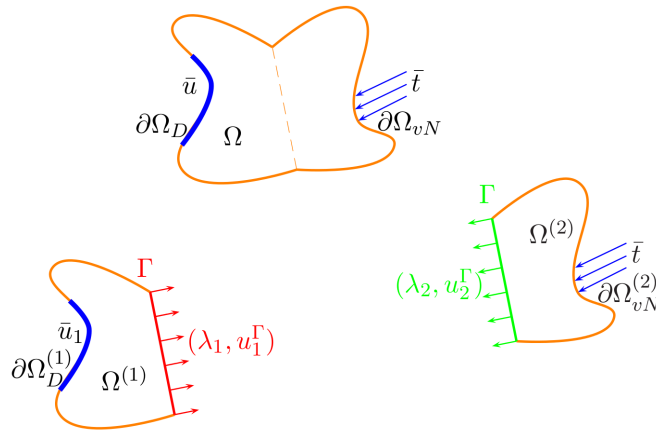


FIGURE 2.2 – Décomposition d'un domaine en deux sous-domaine [Hautefeuille, 2009]

2.2 Méthodes de décomposition de domaine

Dans cette partie les systèmes d'équations utilisés dans les différentes méthodes de décomposition de domaine sans chevauchement sont présentés. Les méthodes avec chevauchement (méthode Schwarz [Schwarz, 1869], méthode Arlequin [Rateau, 2003] par exemple) ne sont pas détaillées dans ce manuscrit. L'exemple de la figure est utilisé 2.2 pour expliquer chaque méthode. La généralisation des différentes méthodes, présentées ici avec seulement deux sous-domaines, consiste à assembler les différents groupes de sous-domaines.

2.2.1 Formulation d'un problème de décomposition de domaine

Dans un premier temps le cadre mathématique des méthodes de décomposition de domaine est présenté en donnant les équations générales communes à ces dernières.

L'énergie potentielle du système peut être divisée en deux parties, une partie Π^1 définie sur (1) et une partie Π^2 définie sur (2) . L'énergie totale étant calculée comme la somme des deux $\Pi^{tot} = \Pi^1 + \Pi^2$. Une force \bar{t} est appliquée sur le sous domaine 2 sur la surface de Neuman ($\delta\Omega_{vN}$) et un déplacement \bar{u} est imposé sur le sous domaine 1 sur la surface de Dirichlet ($\delta\Omega_D$). Les deux énergies potentielles valent :

$$\Pi^1 = \int_{\Omega^{(1)}} \Psi_1(\varepsilon_1) \, dS - \int_{\Gamma} \lambda_1 u_1^\Gamma \, dS \quad (2.2.1a)$$

$$\Pi^2 = \int_{\Omega^{(2)}} \Psi_2(\varepsilon_2) \, dS - \int_{\Gamma} \lambda_2 u_2^\Gamma \, dS - \int_{\delta\Omega_{vN}^{(2)}} \bar{t} u_2^\Gamma \, dS \quad (2.2.1b)$$

Une des méthodes de résolution d'un système couplé est l'utilisation d'un schéma de résolution de Newton. Ce dernier nécessite la connaissance du développement de Taylor de premier ordre du résidu (ici de l'énergie potentielle). La variation d'énergie potentielle est calculée dans chacun des sous-domaines comme suit :

$$\delta\Pi^1 = \int_{\Omega^{(1)}} \delta\Psi_1(\varepsilon_1) \, dS - \int_{\Gamma} \delta(\lambda_1 u_1^\Gamma) \, dS \quad (2.2.2a)$$

$$\delta\Pi^2 = \int_{\Omega^{(2)}} \delta\Psi_2(\varepsilon_2) \, dS - \int_{\Gamma} \delta(\lambda_2 u_2^\Gamma) \, dS - \int_{\delta\Omega_{vN}^{(2)}} \bar{t} \delta u_2^\Gamma \, dS \quad (2.2.2b)$$

La variation de la densité d'énergie de déformation peut être réécrite en fonction des contraintes de Cauchy σ_e et des déplacements u_e :

$$\delta\Psi_e = \sigma_e \delta u_e \quad (2.2.3)$$

Les champs de déplacement u_e doivent être définis dans l'espace de Sobolev ($H^1(\Omega^{(e)})$) et les forces cohésives λ_e doivent être prises dans un espace de Hilbert ($L^2(\Gamma)$). Nous pouvons désormais approximer les valeurs de u_e et λ_e en utilisant classiquement la méthode des éléments finis.

Les déplacements sont pris dans l'espace des polynômes de degrés p $\mathcal{P}_h^p(\Omega^{(e)})$ qui est un sous espace de $H^1(\Omega^{(e)})$ ce choix est classique en éléments finis.

Concernant les forces cohésives le choix est plus complexe. Dans le cas où les maillages sont conformes (c'est à dire que les nœuds aux interfaces correspondent) l'espace des fonctions delta $\mathcal{F}_h^\delta(\Gamma)$ peut être choisi, alors que dans le cas des maillages non conformes les approximations de mortier peuvent être utilisées. Ces dernières seront choisies et décrites plus précisément dans la suite du manuscrit.

Un des points clés de la méthode de décomposition de domaine est de séparer les déplacements en deux parties, une partie dans le domaine $u_{e\Gamma}^h$ et une partie sur l'interface $u_{e\Gamma}^h$.

Les déplacements et les forces cohésives sont décomposés comme suit :

$$\lambda_e^h = \sum_{i=1}^{n_\Gamma} N_\lambda^i \Delta_e^i, \quad \text{ou } :N_\lambda^i \in \mathcal{F}_h^\delta(\Gamma) \quad (2.2.4a)$$

$$u_{e\Gamma}^h = \sum_{i=1}^{n_\Gamma} N_\Gamma^i U_{e\Gamma}^i, \quad \text{ou } :N_\Gamma^i \in \mathcal{P}_h^p(\Gamma) \quad (2.2.4b)$$

$$u_{e\bar{\Gamma}}^h = \sum_{i=1}^{n_{e\bar{\Gamma}}} N_{e\bar{\Gamma}}^i U_{e\bar{\Gamma}}^i, \quad \text{ou } :N_{e\bar{\Gamma}}^i \in \mathcal{P}_h^p(\Omega^{(e)} \setminus \Gamma) \quad (2.2.4c)$$

Nous pouvons désormais réécrire les termes des équations (2.2.2) en utilisant les espaces décrit précédemment ainsi que la décomposition avec l'interface :

$$\int_{\Omega^{(e)}} \sigma_e \partial \delta u_e \, dS = \delta U_{e\Gamma}^T \underbrace{\int_{\Gamma} \partial N_\Gamma \sigma_e^h \, dS}_{f_{e\bar{\Gamma}}^{int}} + \delta U_{e\bar{\Gamma}}^T \underbrace{\int_{\Omega^{(e)} \setminus \Gamma} \partial N_{\bar{\Gamma}} \sigma_e^h \, dS}_{f_{e\bar{\Gamma}}^{int}} \quad (2.2.5a)$$

$$\int_{\Gamma} \delta(\lambda_e u_e^\Gamma) \, dS = \delta U_{e\Gamma}^T \underbrace{\int_{\Gamma} N_\Gamma N_\lambda^T \, dS}_E \Lambda_e + \delta \Lambda_e^T \underbrace{\int_{\Gamma} N_\lambda N_\Gamma^T \, dS}_{E^T} U_e^T \quad (2.2.5b)$$

$$\int_{\delta\Omega_{vN}^{(2)}} \bar{t} \delta u_2^\Gamma \, dS = \delta U_{2\bar{\Gamma}}^T \underbrace{\int_{\delta\Omega_{vN}^{(2)}} N_{2\bar{\Gamma}} \bar{t} \, dS}_{f_{2\bar{\Gamma}}^{ext}} \quad (2.2.5c)$$

Le système peut être réécrit sous forme de résidu :

Pour tout $\delta U_{e\bar{\Gamma}} \in \mathcal{P}_h^p(\Omega^{(e)} \setminus \Gamma)$ sur chaque sous-domaine e et pour chaque $\delta U_{e\Gamma} \in \mathcal{P}_h^p(\Gamma)$ et $\delta \Lambda_{e\Gamma} \in \mathcal{F}_h^\delta(\Gamma)$ sur l'interface Γ

Trouver $U_{e\bar{\Gamma}} \in \mathcal{P}_h^p(\Omega^{(e)} \setminus \Gamma)$ sur chaque sous domaine e et $U_\Gamma \in \mathcal{P}_h^p(e\Gamma)$ et $\Lambda_{e\Gamma} \in \mathcal{F}_h^\delta(\Gamma)$ sur l'interface Γ

Tel que :

$$0 = \begin{bmatrix} \delta U_{1\bar{\Gamma}} \\ \delta U_{2\bar{\Gamma}} \\ \delta U_{1\Gamma} \\ \delta U_{2\Gamma} \\ \delta \Lambda_1 \\ \delta \Lambda_2 \end{bmatrix}^T \cdot \begin{bmatrix} f_{1\bar{\Gamma}}^{int} \\ f_{2\bar{\Gamma}}^{int} - f_{1\bar{\Gamma}}^{ext} \\ f_{1\Gamma}^{int} - E \Lambda_1 \\ f_{2\Gamma}^{int} - E \Lambda_2 \\ E^T U_{1\Gamma} \\ E^T U_{2\Gamma} \end{bmatrix} \quad (2.2.6)$$

Cette équation peut être résolue en utilisant la méthode de Newton qui consiste à résoudre le système d'équation linéaire donné par le résidu de l'expansion de Taylor du premier ordre à chaque itération. Le résidu à l'itération $k + 1$ est calculé de la manière suivante :

$$\begin{aligned}
\begin{bmatrix} f_{1\bar{\Gamma}}^{int} \\ f_{2\bar{\Gamma}}^{int} - f_{1\bar{\Gamma}}^{ext} \\ f_{1\Gamma}^{int} - E\Lambda_1 \\ f_{2\Gamma}^{int} - E\Lambda_2 \\ E^T U_{1\Gamma} \\ E^T U_{2\Gamma} \end{bmatrix}^{(k+1)} &= \begin{bmatrix} f_{1\bar{\Gamma}}^{int} \\ f_{2\bar{\Gamma}}^{int} - f_{1\bar{\Gamma}}^{ext} \\ f_{1\Gamma}^{int} - E\Lambda_1 \\ f_{2\Gamma}^{int} - E\Lambda_2 \\ E^T U_{1\Gamma} \\ E^T U_{2\Gamma} \end{bmatrix}^{(k)} \\
&+ \begin{bmatrix} \hat{K}_{11}^{(k)} & 0 & K_{1\Gamma}^{(k)} & 0 & 0 & 0 \\ 0 & \hat{K}_{22}^{(k)} & 0 & K_{2\Gamma}^{(k)} & 0 & 0 \\ K_{\Gamma 1}^{(k)} & 0 & K_{\Gamma\Gamma}^{(k)} & 0 & -E & 0 \\ 0 & K_{\Gamma 2}^{(k)} & 0 & K_{\Gamma\Gamma}^{(k)} & 0 & -E \\ 0 & 0 & -E^T & 0 & 0 & 0 \\ 0 & 0 & 0 & -E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta U_{1\bar{\Gamma}} \\ \delta U_{2\bar{\Gamma}} \\ \delta U_{1\Gamma} \\ \delta U_{2\Gamma} \\ \delta \Lambda_1 \\ \delta \Lambda_2 \end{bmatrix}^{(k+1)} \quad (2.2.7)
\end{aligned}$$

Dans cette équation la matrice de rigidité est décomposée de la manière suivante :

$$K_{ee} = \begin{bmatrix} \hat{K}_{ee} & K_{e\Gamma} \\ K_{\Gamma e} & K_{\Gamma\Gamma} \end{bmatrix} \quad (2.2.8)$$

À chaque itération nous calculons l'incrément du vecteur d'inconnues :

$$\begin{bmatrix} \delta U_{1\bar{\Gamma}} \\ \delta U_{2\bar{\Gamma}} \\ \delta U_{1\Gamma} \\ \delta U_{2\Gamma} \\ \delta \Lambda_1 \\ \delta \Lambda_2 \end{bmatrix}^{(k+1)} = \begin{bmatrix} \hat{K}_{11}^{(k)} & 0 & K_{1\Gamma}^{(k)} & 0 & 0 & 0 \\ 0 & \hat{K}_{22}^{(k)} & 0 & K_{2\Gamma}^{(k)} & 0 & 0 \\ K_{\Gamma 1}^{(k)} & 0 & K_{\Gamma\Gamma}^{(k)} & 0 & -E & 0 \\ 0 & K_{\Gamma 2}^{(k)} & 0 & K_{\Gamma\Gamma}^{(k)} & 0 & -E \\ 0 & 0 & -E^T & 0 & 0 & 0 \\ 0 & 0 & 0 & -E^T & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} f_{1\bar{\Gamma}}^{int} \\ f_{2\bar{\Gamma}}^{int} - f_{1\bar{\Gamma}}^{ext} \\ f_{1\Gamma}^{int} - E\Lambda_1 \\ f_{2\Gamma}^{int} - E\Lambda_2 \\ E^T U_{1\Gamma} \\ E^T U_{2\Gamma} \end{bmatrix}^{(k)} \quad (2.2.9)$$

La valeur du vecteur d'inconnues est ensuite actualisée :

$$\begin{bmatrix} U_{1\bar{\Gamma}} \\ U_{2\bar{\Gamma}} \\ U_{1\Gamma} \\ U_{2\Gamma} \\ \Lambda_1 \\ \Lambda_2 \end{bmatrix}^{(k+1)} = \begin{bmatrix} U_{1\bar{\Gamma}} \\ U_{2\bar{\Gamma}} \\ U_{1\Gamma} \\ U_{2\Gamma} \\ \Lambda_1 \\ \Lambda_2 \end{bmatrix}^{(k)} + \begin{bmatrix} \delta U_{1\bar{\Gamma}} \\ \delta U_{2\bar{\Gamma}} \\ \delta U_{1\Gamma} \\ \delta U_{2\Gamma} \\ \delta \Lambda_1 \\ \delta \Lambda_2 \end{bmatrix}^{(k+1)} \quad (2.2.10)$$

La résolution consiste pour chaque itération à former la matrice tangente et le vecteur de résidu puis à calculer l'incrément du vecteur d'inconnues. Nous mettons ensuite à jour le vecteur d'inconnues en ajoutant l'incrément calculé puis la procédure est recommencée tant que la norme du vecteur résidu est supérieure à une tolérance.

Une des difficultés des méthodes de décomposition de domaine est l'impossibilité d'imposer simultanément un déplacement et une force à un même nœud afin que les équations

de continuité soient vérifiées. Ainsi, le système d'équation (2.2.9) ne peut être résolu sous cette forme.

Un choix doit être effectué afin de trouver un compromis soit en vérifiant l'égalité des déplacements (méthode primale) soit en vérifiant l'égalité des forces cohésives (méthode duale) ou en faisant un compromis entre les deux (méthode hybrides, à trois champs ou de mortier).

Nous allons maintenant détailler les particularités des différentes méthodes existantes.

2.2.2 Méthode primale

Le principe de la méthodes dite primale ([Le Tallec, 1994], [Mandel, 1993]) est d'imposer la discontinuité des déplacements entre les différents sous-domaines.

Une nouvelle variable U_Γ correspondant aux déplacements aux nœuds de l'interface Γ est introduite ici. Le principe étant d'imposer la continuité des déplacements, l'égalité suivante est vérifiée :

$$U_{1\Gamma} = U_{2\Gamma} = U_\Gamma \text{ on } \Gamma^h \quad (2.2.11)$$

La variable suivante pour les forces cohésives est alors définie :

$$\Lambda_\Sigma = \Lambda_1 + \Lambda_2 \quad (2.2.12)$$

Le système d'équation (2.2.9) devient alors :

$$\begin{bmatrix} \delta U_{1\bar{\Gamma}} \\ \delta U_{2\bar{\Gamma}} \\ \delta U_\Gamma \\ \delta \Lambda_\Sigma \end{bmatrix}^{(k+1)} = \begin{bmatrix} \hat{K}_{11}^{(k)} & 0 & K_{1\Gamma}^{(k)} & 0 \\ 0 & \hat{K}_{22}^{(k)} & K_{2\Gamma}^{(k)} & 0 \\ K_{\Gamma 1}^{(k)} & K_{\Gamma 2}^{(k)} & K_{\Gamma\Gamma}^{(k)} & -E \\ 0 & 0 & -E^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} f_{1\bar{\Gamma}}^{int} \\ f_{2\bar{\Gamma}}^{int} - f_{1\bar{\Gamma}}^{ext} \\ f_{1\Gamma}^{int} + f_{2\Gamma}^{int} - E\Lambda_\Sigma \\ E^T U_\Gamma \end{bmatrix}^{(k)} \quad (2.2.13)$$

Dans la littérature le problème est souvent réduit à un système où on ne s'occupe uniquement que des déplacements et où la partie sur les forces cohésives est omise. Le problème se réduit alors au problème suivant :

$$\begin{bmatrix} \delta U_{1\bar{\Gamma}} \\ \delta U_{2\bar{\Gamma}} \\ \delta U_\Gamma \end{bmatrix}^{(k+1)} = \begin{bmatrix} \hat{K}_{11}^{(k)} & 0 & K_{1\Gamma}^{(k)} \\ 0 & \hat{K}_{22}^{(k)} & K_{2\Gamma}^{(k)} \\ K_{\Gamma 1}^{(k)} & K_{\Gamma 2}^{(k)} & K_{\Gamma\Gamma}^{(k)} \end{bmatrix}^{-1} \begin{bmatrix} f_{1\bar{\Gamma}}^{int} \\ f_{2\bar{\Gamma}}^{int} - f_{1\bar{\Gamma}}^{ext} \\ f_{1\Gamma}^{int} + f_{2\Gamma}^{int} - E\Lambda_\Sigma \end{bmatrix}^{(k)} \quad (2.2.14)$$

La résolution d'un tel problème peut se faire à l'aide d'une condensation statique. La valeur de U_Γ est supposée connue dans un premier temps et les deux équations locales sont alors résolues :

$$\delta U_{e,\bar{\Gamma}}^{k+1,i} = \hat{K}_{ee}^{(k)-1} (f_{e\bar{\Gamma}}^{int} - f_{e\bar{\Gamma}}^{ext} - K_{e\Gamma} (f_{1\Gamma}^{int} + f_{2\Gamma}^{int})) \quad (2.2.15)$$

Cette résolution correspond à un problème classique de Dirichlet. Une fois cette étape réalisée sur chacun des sous-domaines, les résidus de la première et deuxième ligne sont nuls. On peut alors calculer la valeur de U_Γ à l'aide du complément de Schur :

$$\delta U_{\Gamma}^{k+1,i+1} = \left(K_{\Gamma\Gamma} + \sum_{e=1}^2 K_{\Gamma e} \hat{K}_{ee}^{-1} K_{e\Gamma} \right)^{-1} (f_{1\Gamma}^{int,k} + f_{2\Gamma}^{int,k}) \quad (2.2.16)$$

La taille de la matrice du complement de Schurr ainsi que la complexité de son calcul font partie des difficultés de la méthode primale.

2.2.3 Méthode duale

La seconde méthode consiste à imposer l'égalité des forces cohésives sur l'interface. Cette méthode est appelée méthode duale [Farhat and Roux, 1991]. On note comme pour la méthode primale la valeur commune à l'interface :

$$\Lambda_1 = -\Lambda_2 = \Lambda \text{ on } \Gamma \quad (2.2.17)$$

On notera la différence de déplacement au niveau de l'interface :

$$U_g = U_{1\Gamma} - U_{2\Gamma} \quad (2.2.18)$$

Dans le cadre de la méthode duale les conditions aux interfaces sont de types Neuman ce qui impose que la matrice de rigidité ne soit plus décomposée. Il est nécessaire ici d'introduire une fonction de projection de $\Omega^{(e),h}$ dans Γ^h :

$$P_e : \begin{array}{l} \Omega^{(e),h} \rightarrow \Gamma^h \\ U_e \rightarrow U_{e\Gamma} \end{array} \quad (2.2.19)$$

Le système d'équation est simplifié comme pour le cas de la méthode primale et devient alors :

$$\begin{bmatrix} \delta U_1 \\ \delta U_2 \\ \delta \Lambda \\ \delta U_g \end{bmatrix}^{(k+1)} = \begin{bmatrix} K_{11}^{(k)} & 0 & P_1^T E & 0 \\ 0 & K_{22}^{(k)} & P_2^T E & 0 \\ E^T P_1 & E^T P_2 & 0 & E^T \\ 0 & 0 & E & 0 \end{bmatrix}^{-1} \begin{bmatrix} f_1^{int} \\ f_2^{int} - f_2^{ext} \\ E^T U_g \\ -E\Lambda \end{bmatrix}^{(k)} \quad (2.2.20)$$

La dernière ligne est a dernière colonne correspondent à la différence de déplacement et sont omise comme pour la méthode primale. Le système devient alors :

$$\begin{bmatrix} \delta U_1 \\ \delta U_2 \\ \delta \Lambda \\ \delta U_g \end{bmatrix}^{(k+1)} = \begin{bmatrix} K_{11}^{(k)} & 0 & P_1^T E \\ 0 & K_{22}^{(k)} & P_2^T E \\ E^T P_1 & E^T P_2 & 0 \end{bmatrix}^{-1} \begin{bmatrix} f_1^{int} \\ f_2^{int} - f_2^{ext} \\ E^T U_g \\ -E\Lambda \end{bmatrix}^{(k)} \quad (2.2.21)$$

La méthode de résolution est similaire à celle de la méthode primale. On résout indépendamment le calcul dans chaque sous-domaine. Cette partie du calcul est ici plus compliquée car les problèmes locaux peuvent être mal posés au sens de la mécanique. En effet les sous-domaines peuvent n'avoir aucune condition aux limites de type Dirichlet, leur

déplacement est alors défini à un mouvement de corps rigide près. Il est possible d'utiliser les inverses généralisés mais qui ne donnent pas accès à une solution unique. Une fois les solutions trouvées on utilise à nouveau le complément de Schur pour la résolution du problème :

$$\delta\Lambda^{(k,i+1)} = \left(\sum_{e=1}^2 E^T P_e K_{ee} P_e^T E \right)^{-1} E^T U_g \quad (2.2.22)$$

La difficulté de cette méthode réside à la définition au mouvement de corps rigide près du déplacement lorsqu'il n'y a aucune condition aux limites de type Dirichlet imposé sur un sous-domaine.

2.2.4 Autres méthodes

2.2.4.1 Méthode hybride

Les méthodes hybrides [Farhat et al., 2001] sont un mélange des méthodes primale et duale. Dans ces méthode l'interface est coupé en deux parties Γ^D et Γ^{vN} qui correspondent aux parties primale et duale.

$$U_{1\Gamma} = U_{2\Gamma} = U_\Gamma \text{ sur } \Gamma^D \quad (2.2.23)$$

$$\Lambda_1 = -\Lambda_2 = \Lambda \text{ sur } \Gamma^{vN} \quad (2.2.24)$$

2.2.4.2 Méthode à trois champs

Dans cette approche ([Hautefeuille, 2009], [Park et al., 1997]) l'idée est d'utiliser une interface fictive Γ_s . Les forces cohésives ne sont pas appliquées ici entre les deux sous-domaines mais entre chaque sous domaine et l'interface fictive. On assure l'égalité des déplacements de manière faible à l'interface.

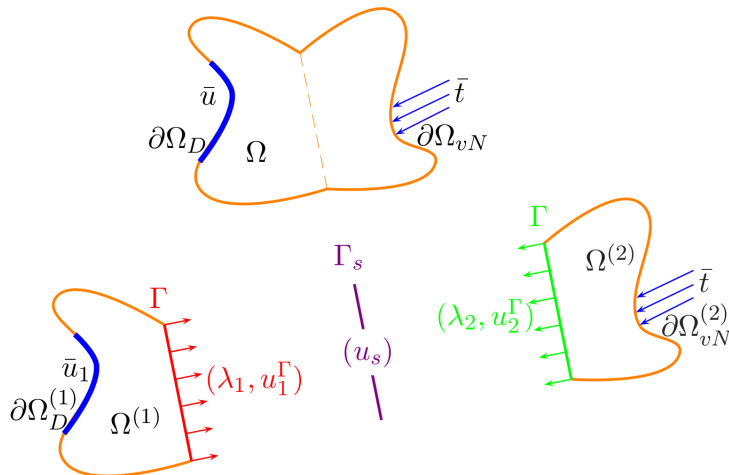


FIGURE 2.3 – Décomposition de deux sous-domaines avec une interface fictive

Ces méthodes permettent l'utilisation de maillages non conformes aux interfaces.

2.2.4.3 Méthode de mortier

Le principe des méthodes de mortier ([Maday et al., 1989], [Bernardi et al., 1990], [Stefanica, 1999], [Bernardi et al., 2005], [Series et al., 2003]) est de différencier pour chacune des interfaces les deux côtés. On définit arbitrairement un des côtés comme étant mortier et l'autre comme esclave. Sur la figure 2.5 on peut voir un exemple de repartition de mortier (en bleu) et esclave en noir.

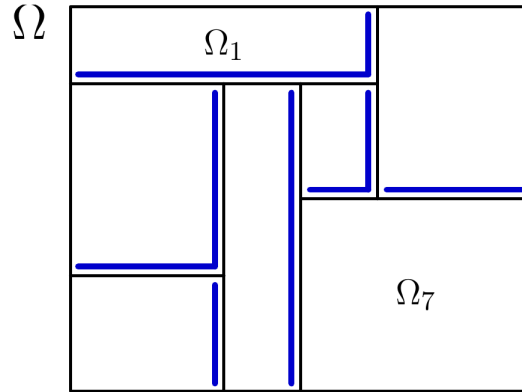


FIGURE 2.4 – Partitionnement en sous-domaines de mortier (côté mortier en bleu)

Les déplacements à l'interface côté mortier sont projetés sur le côté non mortier. On vérifie ensuite l'équation suivante qui doit garantir la continuité des déplacements de manière faible [Danek and Kutakova,] :

$$\int_{\delta l} (u_l - \pi_{q_1, q_2}(u_l)) \psi \, ds = 0; \quad \forall \psi \in \Psi^h(\delta l) \quad (2.2.25)$$

Dans cette équation u_l est le déplacement côté esclave, $\pi_{q_1, q_2}(u_l)$ est la projection (appelé projection de mortier) du déplacement côté mortier sur le côté esclave et ψ représente les fonctions de mortier. Cette méthode permet l'utilisation des maillages non conformes aux interfaces.

Pour pouvoir raffiner localement le maillage il est nécessaire de pouvoir utiliser une méthode qui permet l'utilisation de maillage non conforme aux interfaces. Comme nous avons pu le voir précédemment il existe seulement deux approches qui autorisent l'utilisation de maillage non conforme : les méthodes de mortier et les approches à trois champs. Le choix a été fait ici d'utiliser la méthode de mortier.

Ce choix a été motivé par le besoin d'utiliser la décomposition de domaine dans un cadre non linéaire (fissuration). Pour l'approche à trois champs on impose un champ de déplacement aux interfaces (via l'interface fictive), le fait d'imposer un déplacement aux bords ne permet pas d'avoir une fissuration convenable. Les limites de cette approche dans un cadre de fissuration par ajout de discontinuité forte ont été observées dans le cadre de la thèse de Martin Hautefeuille [Hautefeuille, 2009].

Dans la partie suivante nous allons voir la formulation détaillée de la méthode de mortier appliquée à la mécanique.

2.3 Méthode de mortier appliquée à la mécanique

Une des spécificités de la méthode de mortier est de différencier les deux côtés de l'interface avec un côté mortier et un côté non mortier. Les déplacements du côté mortier sont projetés du côté esclave. Pour effectuer cette projection il est nécessaire de définir dans un premier temps l'espace des éléments finis de mortier.

Une fois cette définition effectuée nous donnerons une définition plus complète de la formulation d'un problème de mécanique à l'aide de la méthode de mortier.

Comme pour les différentes méthodes de décomposition de domaine, la formulation du problème fait apparaître des matrices d'interpolations. Dans le cas général présenté précédemment où les maillages étaient conformes et où l'utilisation de fonction delta pour les matrices d'interpolation, la création des matrices est immédiate. Dans le cas de l'utilisation des éléments finis de mortier, la construction des matrices d'interpolation est plus complexe et sera décrite à la fin de cette section.

2.3.1 Éléments finis de mortier

Nous allons ici prendre l'exemple d'un domaine Ω dans \mathbb{R}^d , $d = 2$ or 3 . Ce domaine est décomposé en K sous-domaines sans chevauchement. Nous avons alors les définitions suivantes :

$$\overline{\Omega} = \bigcup_{k=1}^K \overline{\Omega}_k \quad \text{et} \quad \Omega_k \cap \Omega_l = \emptyset, \quad 1 < k < l \leq K \quad (2.3.1)$$

Nous définissons ensuite l'interface Γ entre les sous-domaines Ω_k .

$$\Gamma = \bigcup_{k=1}^P \overline{(\delta\Omega_k \setminus \delta\Omega)} \quad (2.3.2)$$

On introduit aussi une notation pour l'interface entre deux sous-domaines donnés :

$$\Gamma_{kl} = \overline{\Omega}_k \cap \overline{\Omega}_l \quad (2.3.3)$$

On note l'espace des éléments finis de mortier défini sur Ω :

$$V^h(\Omega) = V_1^h(\Omega_1) \times V_2^h(\Omega_2) \times V_3^h(\Omega_3) \times \cdots \times V_K^h(\Omega_K) \quad (2.3.4)$$

Où $V_k^h(\Omega_k)$ est l'espace d'éléments finis de mortier dans un sous domaine.

Pour chaque interface Γ_{kl} entre deux sous-domaines k et l on choisit arbitrairement un côté mortier et un côté esclave. Nous utiliserons les notations suivantes pour différencier les deux côtés :

- γ pour le côté mortier
- δ pour le côté esclave

Pour une interface d'un sous domaine Ω_i on note γ_k si le côté est un côté de mortier δ_k si le côté est un côté esclave. Le nombre k représente le numéro de l'interface.

Nous avons alors la relation suivante :

$$\begin{aligned} \Gamma &= \bigcup_{k=1}^K \overline{\delta_k} \\ \Gamma &= \bigcup_{k=1}^L \overline{\gamma_k} \end{aligned} \quad (2.3.5)$$

Ici K est le nombre de côté esclave et L le nombre de côté mortier. Dans le cas d'une décomposition géométrique conforme nous avons la relation $K = L$.

Pour chaque côté esclave δ_k nous avons un côté de mortier γ_k associé et on définit un espace de fonction test $\psi^h(\gamma_k)$ qui est un sous espace de restriction de la fonction $V_k^h(\Omega_k)$ sur γ_k . L'espace choisi est l'espace des fonctions linéaires, avec comme condition que les fonctions linéaires soient constantes sur le bord. On peut voir un exemple de ces fonctions linéaires appelées fonction de mortier sur la figure 2.5 pour un problème 2D.

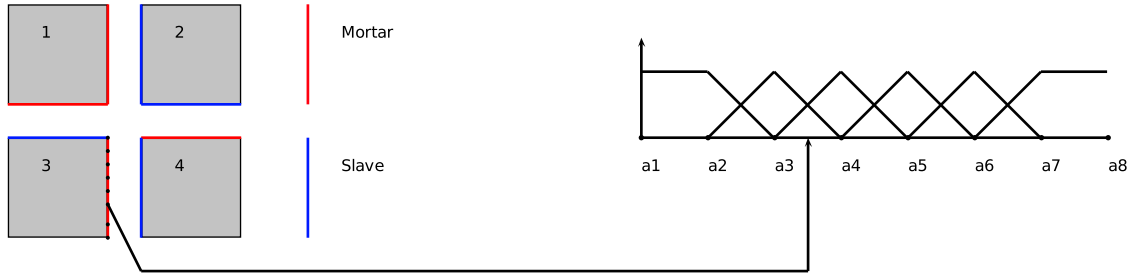


FIGURE 2.5 – Définition des fonctions de mortier en 2D

2.3.2 Formulation de la méthode de mortier appliquée à la mécanique

Dans un premier temps nous allons considérer le problème général d'une équation de Laplace. Nous considérons à nouveau un domaine Ω avec des conditions aux limites homogènes de type Dirichlet sur la frontière $\partial\Omega$.

$$\begin{cases} -\Delta u = f & \text{dans } \Omega \\ u = 0 & \text{sur } \delta\Omega \end{cases} \quad (2.3.6)$$

Où $f \in L^2(\Omega)$. La formulation variationnelle du problème est la suivante :

$$\int_{\Omega} \nabla u(x) \cdot \nabla v(x) \, dx = \int_{\Omega} f v(x) \, dx \quad \forall v \in H_0^1(\Omega) \quad (2.3.7)$$

Où $H_g^1(\Omega)$ représente l'espace $H_g^1(\Omega) = \{u \in H^1(\Omega), u = g \text{ on } \delta\Omega\}$.

Comme précédemment nous décomposons le domaine Ω en sous-domaines Ω_k . À nouveau les sous-domaines ne se chevauchent pas. Nous pouvons introduire la définition de multiplicateur de Lagrange dans l'équation d'Euler [Belgacem, 1999]. Les multiplicateurs de Lagrange sont définis sur un sous espace M de $\Pi_{1 \leq k \leq l \leq K} H^{-1/2}(\Gamma_{k,l})$. On note X^* l'union des sous espaces X_k . Le problème variationnel peut être écrit de la manière suivante :

Trouver $(u^*, \lambda) \in V^* \times M$ tel que :

$$\forall v^* \in V^*, \sum_{k=1}^K \int_{\Omega^k} \nabla u_k \nabla v_k \, dx + \sum_{1 \leq k \leq l \leq K} \int_{\Gamma_{k,l}} (v_k|_{\Gamma_{k,l}} - v_l|_{\Gamma_{k,l}}) \lambda \, dS = \sum_{k=1}^K \int_{\Omega^k} f_k v_k \, dx \quad (2.3.8)$$

$$\forall \mu \in M, \quad \sum_{1 \leq k \leq l \leq K} \int_{\Gamma_{k,l}} (u_k|_{\Gamma_{k,l}} - u_l|_{\Gamma_{k,l}}) \mu \, dS = 0 \quad (2.3.9)$$

Avec a_h la forme bilinéaire symétrique et b_h la forme bilinéaire :

$$\begin{aligned} a_h(u_h, v_h) &= \sum_{k=1}^K \int_{\Omega_k} \nabla u_{h,k} \nabla v_{h,k} \, dx \\ b_h(v_h, \mu_h) &= \sum_{1 \leq k \leq l \leq K} \int_{\Gamma_{k,l}} (v_k|_{\Gamma_{k,l}} - v_l|_{\Gamma_{k,l}}) \mu \, dS \end{aligned} \quad (2.3.10)$$

Nous choisissons ensuite l'espace des éléments finis pour l'espace V^* et l'espace des éléments finis de mortier défini précédemment pour l'espace M . On peut alors réécrire les variables ainsi que les formes linéaires et bilinéaires :

$$u_h = \sum_{i=0}^{n_{FEM}} u_i N_i \quad (2.3.11)$$

$$f_f = \sum_{i=0}^{n_{FEM}} f_i N_i \quad (2.3.12)$$

$$A_{h,ij} = a(N_i, N_j) \quad (2.3.13)$$

$$B_{h,ij} = b(N_i, \varphi_j) \quad (2.3.14)$$

Avec N_i les fonctions de formes conventionnelles en éléments finis et φ_i les éléments finis de mortier décrits précédemment. La matrice A_h est donnée par le code de calcul éléments finis pour chaque sous domaine et la matrice B_h est calculée numériquement dans notre code.

Les équations (2.3.8) et (2.3.9) [Lacour and Maday, 1997] peuvent être réécrites à l'aide des opérateurs linéaires définis précédemment :

Trouver $(u^*, \lambda) \in V^* \times M$ tel que :

$$A_h u_h + B_h^t \lambda_h = f_h \quad (2.3.15)$$

$$B_h u_h = 0 \quad (2.3.16)$$

Le système peut être écrit sous forme matricielle :

$$\begin{pmatrix} A_h & B_h \\ B_h^T & 0 \end{pmatrix} \begin{pmatrix} u_h \\ \lambda_h \end{pmatrix} = \begin{pmatrix} f_h \\ 0 \end{pmatrix} \quad (2.3.17)$$

Maintenant que la formulation de la méthode de mortier est décrite pour un problème général nous allons passer à un problème de mécanique. Dans le cas de la mécanique des milieux continus la matrice A_h correspond à la matrice de raideur K classique. Le problème dans le cas d'un domaine Ω coupé en deux sous-domaines Ω_1 et Ω_2 amène à l'équation matricielle suivante :

$$\begin{pmatrix} K_{11} & K_{1\delta} & 0 & 0 & 0 \\ K_{\delta 1} & K_{\delta\delta} & 0 & 0 & B_\delta^T \\ 0 & 0 & K_{22} & K_{2\gamma} & 0 \\ 0 & 0 & K_{\gamma 2} & K_{\gamma\gamma} & -B_\gamma^T \\ 0 & B_\delta & 0 & B_\gamma & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_\delta \\ u_2 \\ u_\gamma \\ \lambda_\delta \end{pmatrix} = \begin{pmatrix} f_1 \\ f_\delta \\ f_2 \\ f_\gamma \\ 0 \end{pmatrix} \quad (2.3.18)$$

Les déplacements sont décomposés avec une partie en dehors de l'interface et une partie sur l'interface. Les matrices B_δ et B_γ représentent respectivement la matrice d'interpolation côté mortier et la matrice d'interpolation côté esclave.

Dans la partie suivante nous allons voir la méthode utilisée dans cette thèse pour construire les matrices d'interpolation en 3D dans le cas d'utilisation de maillages non conformes.

2.3.3 Construction des matrices d'interpolation en 3D

La méthode de mortier nous permettant d'avoir des maillages non conformes aux interfaces, c'est à dire que les nœuds du maillage d'un sous-domaine au niveau de l'interface ne correspondent pas forcément aux nœuds du maillage voisin, les matrices d'interpolation $B_{h,i,j}$ doivent être calculées numériquement.

Pour décrire le calcul numérique nous allons ici nous placer dans le cas d'un domaine Ω coupé en deux sous-domaines Ω_1 et Ω_2 . Le découpage étant effectué en parallélépipède l'interface sera toujours de forme rectangulaire. Dans notre exemple l'interface est donc un rectangle de longueur L et de largeur l . Notre maillage étant un maillage tétraédrique les deux interfaces notées Γ_1 et Γ_2 seront donc composées de triangles notés t_{Ω_1} et t_{Ω_2} . L'interface Γ_1 correspondra à l'interface de mortier et l'interface Γ_2 à l'interface esclave.

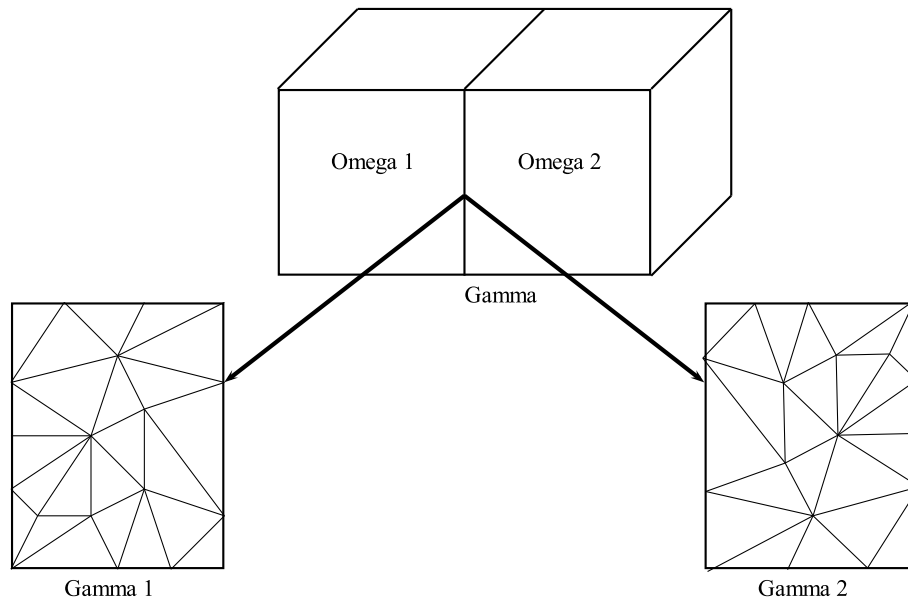


FIGURE 2.6 – Schéma de la décomposition du domaine Ω

La première étape consiste à discrétiser l'interface en rectangle de longueur $\frac{L}{n_d}$ et de largeur $\frac{l}{n_d}$ avec n_d le nombre de discrétisation. Cette discrétisation sera commune aux deux interfaces Γ_1 et Γ_2 .

Les nœuds du maillage de Γ_1 seront notés i , ceux du maillage de Γ_2 seront notés j et enfin ceux de la discrétisation seront notés k . Les fonctions de mortier étant définies uniquement sur l'interface de mortier, elles sont donc uniquement définies sur les nœuds i . Le calcul va se dérouler en plusieurs étapes. Pour chaque point k nous allons réaliser les

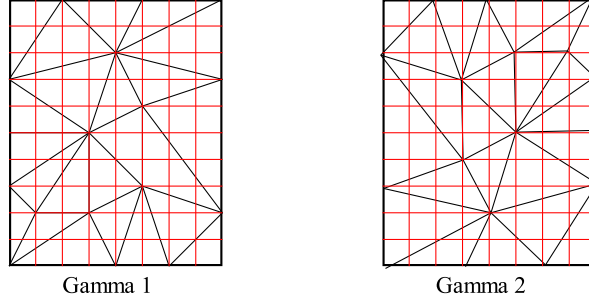


FIGURE 2.7 – Discretisation des interfaces Γ_1 et Γ_2

étapes suivantes :

- On cherche tous les triangles t_{Ω_1} et t_{Ω_2} qui contiennent ce point.
- Pour chacun des points i et j de ces triangles on va calculer $N_i(k)$, $N_j(k)$ et $\mu_I(K)$, N étant les fonctions de formes éléments finis classiques et μ étant les fonctions de mortier définies précédemment.
- On calcul ensuite les matrices d'interpolation de mortier côté mortier (B_1) et côté esclave (B_2) :

$$\begin{aligned} B_1(3 * i + l, 3 * i + l) &+ = N_i(k) * \mu_i(k) * S(k) && \text{avec } l \in [0 - 2] \\ B_2(3 * i + l, 3 * j + l) &+ = N_j(k) * \mu_i(k) * S(k) && \text{avec } l \in [0 - 2] \end{aligned} \quad (2.3.19)$$

Avec $S(k)$ la surface associée au nœud k :

- Si k est dans Γ : $S(k) = \frac{L}{n_d} * \frac{l}{n_d}$
- Si k est sur un bord Γ : $S(k) = \frac{1}{2} \frac{L}{n_d} * \frac{l}{n_d}$
- Si k est dans un coin de Γ : $S(k) = \frac{1}{4} \frac{L}{n_d} * \frac{l}{n_d}$

Le nombre de degrés de libertés par nœuds étant de trois la matrice B_1 sera de taille $3N_I \times 3N_I$ et la matrice B_2 de taille $3N_I \times 3N_J$ avec N_I le nombre de nœuds côté esclave et N_J celui côté mortier. Les valeurs des matrices d'interpolation sont les mêmes suivants les trois degrés de libertés.

Le problème final dans le cas de deux sous-domaines peut se mettre sous la forme :

$$\begin{bmatrix} K_1 & 0 & B_1^T \\ 0 & K_2 & -B_2^T \\ B_1 & -B_2 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ 0 \end{bmatrix} \quad (2.3.20)$$

Avec K_1 et K_2 les matrices de rigidités du problème éléments finis classique, B_1 et B_2 les matrices d'interpolations, u_1 et u_2 les déplacements aux nœuds des deux sous-domaines, f_1 et f_2 les forces extérieures et λ les multiplicateurs de Lagrange.

2.4 Application de la méthode de mortier à un problème de mécanique

Dans cette partie nous allons voir la méthode utilisée pour résoudre le système d'équation obtenu dans le cadre de la méthode de mortier appliquée aux éléments finis. Nous allons voir dans un premier temps l'algorithme développé pour la résolution du système d'équation puis nous détaillerons l'architecture logicielle utilisée et la mise en place informatique de l'algorithme à l'aide de nos outils informatiques. Enfin nous verrons un exemple de calcul mécanique utilisant notre cadre ainsi qu'une étude des performances de celui ci.

2.4.1 Algorithme de résolution linéaire

Notre algorithme consiste à résoudre localement l'équation (2.3.15) de manière distribuée sur un serveur, à l'aide d'un solveur éléments finis classique, pour différentes pseudo forces. Ces pseudo forces sont calculées de manière à diminuer le résidu de l'équation (2.3.16). A chaque itération on calcule les pseudo forces et on résout les systèmes éléments finis locaux jusqu'à ce que le résidu de l'équation (2.3.16) soit inférieur à une tolérance. Les pseudo forces de l'itération i sont calculées comme suit :

$$\lambda_i = B_h u_i \quad (2.4.1)$$

Ces pseudo forces vont permettre la résolution de l'équation (2.3.16) qui assure la continuité des déplacements entre les différents sous-domaines.

Si les pseudo forces ne sont pas équilibrées dans les sous-domaines il est nécessaire de calculer les mouvements de corps rigide qui les équilibrent sans quoi la résolution éléments finis n'est pas possible. On calcule alors les mouvements de corps rigides :

$$u_{RBM,i} = -R^t (R^t B^t B R)^{-1} \lambda_i \quad (2.4.2)$$

Avec R la matrice des mouvements de corps rigide qui est de taille $6 * nddl$ avec $nddl$ le nombre de degrés de libertés. Les six colonnes de la matrice correspondent aux six mouvements de corps rigide possible (trois translations et trois rotations). Pour un nœud i la matrice R_i est calculée de la façon suivante :

$$R_i = \begin{bmatrix} 1 & 0 & 0 & 0 & -(z_i - z_c) & (y_i - y_c) \\ 0 & 1 & 0 & (z_i - z_c) & 0 & -(x_i - x_c) \\ 0 & 0 & 1 & -(y_i - y_c) & (x_i - x_c) & 0 \end{bmatrix} \quad (2.4.3)$$

Avec x_c , y_c et z_c les coordonnées du centre du domaine. On assemble ensuite toutes les matrices R_i .

Nous pouvons recalculer les pseudos forces qui sont maintenant équilibrées :

$$\lambda_i = B(u_i + u_{RBM,i}) \quad (2.4.4)$$

On résout localement le système éléments finis :

$$u_{s,i,k} = K_k^{-1} \lambda_i \quad (2.4.5)$$

Avec i le numéro de l'itération et k le numéro du sous-domaine. On assemble ensuite les solutions des différents sous-domaines.

Une fois la résolution des problèmes éléments finis locaux effectuée on cherche un coefficient amplificateur α_i (méthode line-search) qui minimise le résidu de l'équation (2.3.16) .

$$r_{2,i} = \|B(\alpha_i u_{s,i} + u_{RBM,i})\| \quad (2.4.6)$$

Si le résidu $r_{2,i}$ obtenu est trop important on effectue une nouvelle itération en actualisant la valeur du déplacement u_{i+1} :

$$u_{i+1} = u_{s,i} + U_{RBM,i} \quad (2.4.7)$$

L'algorithme complet de résolution est décrit dans l'algorithme 1. Cette algorithme présente la résolution d'un problème mécanique en partant des conditions au limites jusqu'à la solution finale. Celui ci est implémenté dans notre code de calcul, cette implémentation sera décrite dans la partie suivante.

Pour réaliser la parallélisation des calculs (à la fois des calculs éléments finis locaux et la multiplication des matrices d'interpolations locales par un vecteur) nous utilisons la programmation orientée composants à l'aide du middleware CTL [Niekamp, 2005]. Avec ce middleware nous pouvons utiliser la multi-instanciation et le contrôle à distance de nos composants.

Cette programmation permet de lancer des appels asynchrones en utilisant les systèmes de dialogue TCP/thread/MPI. En utilisant un composant pour chaque sous domaine nous pouvons ainsi lancer de façon asynchrone la résolution éléments finis sur chacun des sous-domaines.

Pour assembler les matrices K locales et pour effectuer la résolution du problème éléments finis nous utilisons un composant éléments finis appelé coFeap [Kassiotis and Hautefeuille, 2008]. Ce composant est l'assemblage du logiciel éléments finis FEAP [Taylor, 1977] et du middleware CTL.

Chaque sous domaine est associé à un composant appelé composant mortier qui encapsule un composant coFeap. Ainsi chaque sous domaine est totalement indépendant informatique des autres pour la résolution et la communication avec le client principal. Les composants mortier peuvent dialoguer entre eux pour échanger des données lorsque cela est nécessaire.

Le client principal permet de contrôler tout les composants mortier pour leurs indiquer quelle tâche effectuer. Dans la partie suivante nous allons décrire plus précisément la programmation orientée composant et l'utilisation de celle ci dans le cas de la décomposition de domaine dans le cadre éléments finis.

Algorithm 1 Algorithmme steepest descent + line search

```
1:  $d\lambda = F_{conditions\ aux\ limites}$ 
2:  $F = R \cdot d\lambda$ 
3: if  $\exists F_{local} \neq 0$  then
4:    $RBM = (R^t B^t B R)^{-1} \cdot F$ 
5:    $d\lambda+ = B^t \cdot B \cdot R^t \cdot RBM$ 
6:    $dU_{RBM} = R^t \cdot RBM$ 
7:    $U_{RBM} = dU_{RBM}$ 
8: end if
9: for chaque sous-domaine  $i$  do
10:   $dU_i = K_i^{-1} \cdot d\lambda_i$ 
11: end for
12:  $U = dU + dU_{RBM}$ 
13:  $\lambda = d\lambda$ 
14: while  $\|B \cdot U\| > tol$  do
15:   $d\lambda = B^t B \cdot U$ 
16:   $F = R \cdot d\lambda$ 
17:  if  $\exists F_{local} \neq 0$  then
18:     $RBM = (R^t B^t B R)^{-1} \cdot F$ 
19:     $d\lambda+ = B^t \cdot B \cdot R^t \cdot RBM$ 
20:     $dU_{RBM} = R^t \cdot RBM$ 
21:     $U_{RBM}+ = dU_{RBM}$ 
22:  end if
23:  for chaque sous-domaine  $i$  do
24:     $dU_i = K_i^{-1} \cdot d\lambda_i$ 
25:  end for
26:   $U+ = dU_{RBM}$ 
27:  Trouver  $\alpha$  qui minimise :  $B(U + \alpha dU)$ 
28:   $U+ = \alpha dU$ 
29:   $\lambda+ = \alpha d\lambda$ 
30: end while
```

2.4.2 Algorithme de résolution non-linéaire

Afin de résoudre le problème mécanique non-linéaire nous utilisons la méthode de Newton. Pour un chargement donné une succession de calcul linéaire est réalisée. Ces calculs linéaires sont réalisés à l'aide de la méthode de mortier présentée précédemment. Une fois le calcul linéaire terminé, les matrices de rigidités locales sont actualisées en suivant la méthode présentée dans le premier chapitre. La procédure est réalisée jusqu'à ce que le calcul converge.

Algorithm 2 Algorithmme steepest descent + line search + méthode de newton

```
1:  $d\lambda = F_{conditions\ aux\ limites}$ 
2:  $F = R \cdot d\lambda$ 
3: if  $\exists F_{local} \neq 0$  then
4:    $RBM = (R^t B^t B R)^{-1} \cdot F$ 
5:    $d\lambda_+ = B^t \cdot B \cdot R^t \cdot RBM$ 
6:    $dU_{RBM} = R^t \cdot RBM$ 
7:    $U_{RBM} = dU_{RBM}$ 
8: end if
9: for chaque sous-domaine  $i$  do
10:   $dU_i = K_i^{-1} \cdot d\lambda_i$ 
11: end for
12:  $U = dU + dU_{RBM}$ 
13:  $\lambda = d\lambda$ 
14:  $iter = 0$ 
15: while do
16:  while  $\|B \cdot U\| > tol$  do
17:     $d\lambda = B^t B \cdot U$ 
18:     $F = R \cdot d\lambda$ 
19:    if  $\exists F_{local} \neq 0$  then
20:       $RBM = (R^t B^t B R)^{-1} \cdot F$ 
21:       $d\lambda_+ = B^t \cdot B \cdot R^t \cdot RBM$ 
22:       $dU_{RBM} = R^t \cdot RBM$ 
23:       $U_{RBM} = dU_{RBM}$ 
24:    end if
25:    for chaque sous-domaine  $i$  do
26:       $dU_i = K_{i,iter}^{-1} \cdot d\lambda_i$ 
27:    end for
28:     $U_+ = dU_{RBM}$ 
29:    Trouver  $\alpha$  qui minimise :  $B(U + \alpha dU)$ 
30:     $U_+ = \alpha dU$ 
31:     $\lambda_+ = \alpha d\lambda$ 
32:  end while
33:   $sortie = true; iter ++$ 
34:  for chaque sous-domaine  $i$  do
35:    if  $K_{i,iter} U_i \neq \lambda_i$  then
36:       $sortie = false$ 
37:    break
38:  end if
39: end for
40: if  $sortie$  then
41:  break
42: end if
43: end while
```

2.4.3 Programmation orientée composant

Dans cette partie nous allons expliquer les principes de la programmation orientée composant, de son application à un code éléments finis puis la réalisation d'un code de calcul éléments finis par décomposition de domaine à l'aide de celle-ci.

2.4.3.1 Le composant, une brique élémentaire

La notion de composant [Szyperski, 1998] est une extension du concept d'objet présent dans la programmation orientée objet. Afin d'être reconnu comme composant celui-ci doit vérifier les cinq critères suivants :

La multi-instanciation Plusieurs composants peuvent être instanciés en même temps, c'est ce qui permet le calcul en parallèle.

Mise en œuvre sans contexte spécifique Les composants ayant la même interface peuvent être échangés. Par exemple, on peut échanger un composant éléments finis par un autre sans qu'il n'y ait de modification à apporter au code.

Composabilité Le composant doit pouvoir communiquer avec d'autres composants et peut éventuellement inclure d'autres composants dans lui même.

Encapsulation Via l'interface du composant on ne peut pas accéder à sa structure interne.

Indépendance à la version du compilateur et au langage de programmation Lorsqu'un composant a été compilé avec succès, si la machine sur laquelle il se trouve est disponible alors ce composant sera toujours accessible même si un autre compilateur est utilisé ou si le langage de programmation a changé.

Pour réaliser un composant il est nécessaire d'utiliser un middleware. Il existe un grand nombre de middleware tel que CORBA, Microsoft .NET, Java RMI. Cependant excepté CORBA ces codes ne correspondent pas à un usage scientifique où des performances élevées sont nécessaires. D'autres codes ont été réalisés par des scientifiques et répondent mieux aux attentes d'un usage scientifique. Parmi eux le middleware CTL [Niekamp, 1995] développé à Institut für Wissenschaftliches Rechnen (TU Braunschweig) a été choisi.

Ce middleware est une librairie C++ qui permet de créer des composants et autorise la communication via un grand nombre de moyen (thread, TCP, MPI, openMPI...).

2.4.3.2 Un composant éléments finis, coFeap

Le composant éléments finis que nous utilisons est basé sur le logiciel éléments finis FEAP [Taylor, 2008] et sur le middleware CTL. Le composant coFeap a été créé à la base à l'université de Braunschweig [Niekamp and Stein, 2002] puis amélioré dans le cadre des thèses de Martin Hautefeuille et Christophe Kassiotis [Hautefeuille, 2009], [Kassiotis, 2009]. Il est basé sur une interface générique qui peut être commune à un grand nombre de simulateur (éléments finis, volumes finis...) appelé ici simu.ci.

Dans cette interface on peut distinguer les trois types de méthodes suivantes :

Les méthodes de pré-traitement Ces méthodes correspondent à la mise en place du problème. Elles peuvent par exemple permettre de mettre en place les conditions

aux limites. Un exemple simple de cette méthode est la méthode `set_load` qui sert à imposer les forces initiales. Cette méthode a besoin d'un vecteur contenant les chargements en chaque nœud du maillage, la définition de la méthode est la suivante :

```
# define CTL_Method6 void, set_load, (const array<scalar1> /*load*/), 1
```

item[] **Les méthodes de traitement** Ces méthodes consistent principalement à mettre en place le système matriciel et à le résoudre. Un exemple évident des ces méthodes est la méthode `solve` qui est une méthode void (elle n'a besoin d'aucune donnée pour être lancée) :

```
# define CTL_Method10 int4, solve, (), 0
```

Les méthodes de post-traitement Ces méthodes permettent de récupérer des données une fois la résolution effectuée. On retrouve ici les méthodes permettant d'obtenir les variables primale, duale. Un exemple de méthode est la méthode `reaction` qui permet d'avoir les réactions nodales :

```
# define CTL_Method21 void, reaction, (array<scalar1> /*r*/) const, 1
```

Un des intérêts d'avoir un composant et d'utiliser une interface générique pour le simulateur est que celui-ci peut être échangeable. Si l'on souhaite changer le code éléments finis FEAP par un autre code éléments finis, tous les programmes utilisant le composant `coFeap` n'auront pas à être modifiés. Le composant est ici une brique interchangeable.

2.4.3.3 Application à la décomposition de domaine

Le choix de garder le composant `coFeap` comme étant un composant générique et donc non dépendant du contexte nous a amené à créer un second composant, appelé mortier, pour pouvoir réaliser notre code de calcul par décomposition de domaine. Le but de ce composant est de limiter au maximum les échanges de données avec le client principal afin d'améliorer la parallélisation du code.

Le composant mortier encapsule un composant `coFeap` pour la partie résolution du système éléments finis. On peut voir la structure du code dans la figure 2.8. Comme on peut le voir sur la figure, le composant `coFeap` étant encapsulé dans le composant mortier, celui-ci n'est pas accessible directement par le client. Certaines méthodes du composant mortier seront présentes pour que le client puisse utiliser les méthodes des composants `coFeap` (pour la résolution des problèmes éléments finis dans chaque sous-domaine par exemple).

Pour expliquer les différentes méthodes nécessaires dans le composant mortier nous allons utiliser l'exemple d'un domaine coupé en 4 sous-domaines en traction. Cet exemple est visible sur la figure 2.9. Pour cet exemple nous avons le client principal, quatre composants mortier contenant chacun un composant `coFeap`.

Initialisation du programme

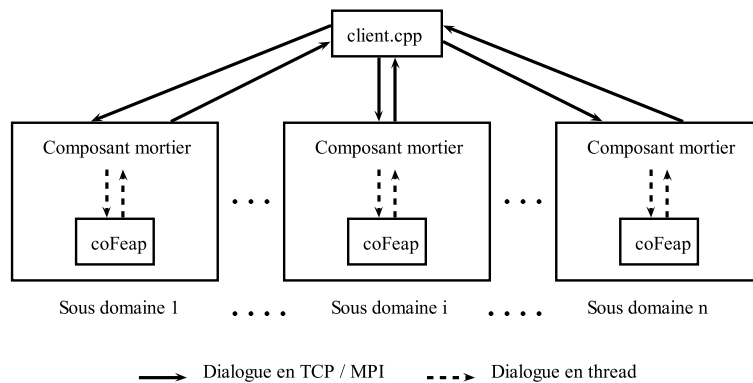


FIGURE 2.8 – Décomposition du programme en briques élémentaires

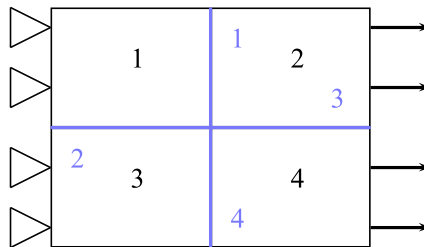


FIGURE 2.9 – Exemple d'un domaine découpé en quatre sous-domaines

Dans cette partie nous allons décrire les différentes étapes nécessaires avant d'effectuer la résolution du problème de décomposition de domaine.

Dans un premier temps nous devons instancier les quatre composants mortier. Les arguments de l'instanciation sont les liens menant aux exécutables du composant mortier et du composant coFeap :

```
mortier( const ctl::string& path_mortier , const ctl::string& path_coFeap ) ;
```

Une fois cette étape réalisée le client va récupérer les informations sur la géométrie des différents sous-domaines. Pour cela il va utiliser une méthode appelée `get_domain` qui a pour argument un vecteur de double qui contiendra les limites du sous-domaine. Le composant mortier va ensuite utiliser une méthode du composant coFeap pour avoir accès aux coordonnées des nœuds du maillage et ainsi pouvoir fournir au client les limites du sous-domaine.

```
void mortier::get_domain(std::vector<double>&){
    std::vector<double> nodes;
    int dim;
    _cofeap.get_nodes(nodes,dim);
    ...
}
```

Afin de pouvoir lancer en parallèle les calculs tout en respectant un nombre de processeur maximum utilisable par nœuds du serveur de calcul, nous utilisons l'algorithme de distribution de calculs présenté dans l'algorithme 4 dans le cas où chaque composant doit lancer la même méthode une fois. L'utilisation du double point d'exclamation devant le composant (`!!composant[i]`) permet de récupérer l'état du composant (renvoi `true` si le composant est disponible et `false` si le composant est déjà en train de tourner).

Algorithm 3 Distribution de calculs sur un serveur

```

1: int nb_proc_utilisé;
2: int nb_calcul;
3: std::vector<int> job (composant.size(),1);
4: while nb_calcul < composant.size() do
5:     for unsigned i = 0; i < composant.size();i++) do
6:         if (job[i]>0)&&!!composant[i]&&(nb_proc_utilisé < nb_proc_utilisable)
7:             then
8:                 composant[i].methode(arguments);
9:                 nb_proc_utilisé++;
10:                job[i]=0;
11:            end if
12:        end for
13:    end while

```

Le client va ensuite déterminer les différentes interfaces (ici les 4 interfaces). Pour chacune des interfaces le client va récupérer les nœuds de chacun des sous-domaines sur cette interface ainsi que les faces des tétraèdres. Le client va faire appel aux deux méthodes du composant mortier qui permettent de calculer les matrices d'interpolation de mortier (côté mortier et côté esclave). Le composant a besoin des données géométriques pour calculer les matrices. Ces matrices sont uniquement stockées dans les composants.

```

void compute_B1(std::vector<int> node_ms,std::vector<double> domain,
std::vector<std::vector<int>> tria_1,std::vector<std::vector<double>> coor_1);
void compute_B2(std::vector<int> node_ms,std::vector<double> domain,
std::vector<std::vector<int>> tria_1,std::vector<std::vector<double>>
coor_1,std::vector<std::vector<int>> tria_2,
std::vector<std::vector<double>> coor_2);

```

En fonction du nombre de processeurs disponibles, les interfaces ne seront pas calculées dans le même ordre. En effet si deux processeurs sont disponibles, les interfaces seront calculées dans l'ordre 1,2,3 et 4 (une interface à la fois car il faut deux processeurs par interfaces). Dans le cas de 4 processeurs, les interfaces seront calculées dans l'ordre 1 et 4 puis 2 et 3. Il est donc nécessaire pour chaque composant de stocker le numéro des interfaces. En effet dans le cas de deux processeurs, le composant 4 stocke l'interface 3 puis l'interface 4 alors que dans le cas où quatre processeurs sont utilisés, ce même composant stocke l'interface 4 puis l'interface 3. Si l'on souhaite pouvoir sauvegarder les matrices

d'interpolation, il semble nécessaire de stocker aussi d'autres informations comme l'ordre de stockage.

Une fois les matrices d'interpolation calculées, le client va devoir calculer la matrice des mouvements de corps rigides (RBM). Pour le calcul de cette matrice, nous avons besoin de la matrice des forces globales R et d'une succession de produits matrices vecteurs.

Cette matrice permet de vérifier l'équilibre globale des forces externes pour un sous-domaine n donné :

$$R_n^T \begin{bmatrix} T \\ M \end{bmatrix} = 0 \quad (2.4.8)$$

On vérifie cette équation pour chacun des nœuds i de la structure :

$$\sum_i R_{n,i}^T \begin{bmatrix} T_i \\ M_i \end{bmatrix} = 0 \quad (2.4.9)$$

Dans le cas d'un problème 3D la matrice $R_{n,i}$ vaut :

$$R_{n,i} = \begin{bmatrix} 1 & 0 & 0 & 0 & -(z_i - z_c) & (y_i - y_c) \\ 0 & 1 & 0 & (z_i - z_c) & 0 & -(x_i - x_c) \\ 0 & 0 & 1 & -(y_i - y_c) & (x_i - x_c) & 0 \end{bmatrix} \quad (2.4.10)$$

Avec i le nœud où la matrice est calculée et c le centre du domaine, la matrice R_n , calculée pour le sous domaine n , est construite de la manière suivante :

$$R_n^T = [R_{n,1}^T | R_{n,2}^T | \dots | R_{n,k}^T] \quad (2.4.11)$$

La matrice R est composée des matrice R_n de chaque sous-domaine :

$$R = [R_1 | R_2 | \dots | R_n] \quad (2.4.12)$$

Nous devons maintenant calculer la matrice RBM comme étant la matrice suivante :

$$RBM = (R^T B^T B R)^{-1} \quad (2.4.13)$$

Nous calculons dans un premier temps la matrice BR . Pour effectuer ce calcul nous allons réaliser le produit de la matrice B par chacune des colonnes, notés Rc_i de la matrice R . Le calcul de B_i , et plus généralement le calcul du produit de la matrice B par un vecteur, se fait en deux étapes.

La première étape consiste pour le sous-domaine concerné par la colonne Rc_i (et plus généralement pour chacun des sous-domaines) à multiplier la colonne par la matrice B . Pour ce faire le client va utiliser la méthode `compute_B_action` du composant mortier, cette méthode n'a besoin d'aucun argument. En effet elle multiplie la matrice B par un vecteur donné qui est déjà stocké dans le composant. Nous devons préalablement définir ce vecteur. Dans le cas de notre programme la matrice B peut être multipliée par trois vecteurs différents au cours du code :

- Le vecteur `temp_URBM`, ce vecteur correspond à l'une des colonnes `Rci`, il suffit juste de préciser au composant la colonne qui doit être utilisée. Ce vecteur est utilisé seulement pour le calcul de la matrice des mouvements de corps rigides.
- Le vecteur `U`, ce vecteur est utilisé pour calculer les pseudos force au début de chaque itération.
- Le vecteur `U+alpha*dU`, ce vecteur permet de calculer le coefficient α qui minimise la discontinuité. On doit ici fournir la valeur de α pour la tester.

Nous avons donc au final trois possibilités auxquelles nous avons associé à chacune une méthode différente (`compute_B_action1`, `compute_B_action2` et `compute_B_action3`).

```
void compute_B_action1();
void compute_B_action2();
void compute_B_action3(const double alpha);
```

Pour le calcul de B_i nous utiliserons donc la première méthode en précisant au préalable le numéro de la colonne. La matrice B étant une matrice creuse, nous utilisons un stockage de type IJV. Pour chaque valeur non nulle de la matrice, nous stockons les indices de la matrice dans les vecteurs `I` et `J` et la valeur dans le vecteur `V`. Les trois vecteurs ont la même taille qui correspond au nombre de valeurs non nulles dans la matrice B .

Cette méthode permet de réduire la taille des données stockées. De plus, lors du calcul du produit de la matrice B par un vecteur, le nombre de calcul est réduit, au lieu d'avoir $n * n$ opérations (n correspondant à la taille de la matrice B) nous avons k opérations (k correspondant au nombre de valeurs non nulles). Les méthodes `compute_B_action` s'écrivent ainsi en peu de ligne :

```
void actionmatrice::compute_B_action(){
for(unsigned i = 0;i < result.size();i++){
for(unsigned j = 0; j < result[i].size();j++){
result[i][j] = 0.0;
}
for(unsigned j=0;j<BI[i].size();j++){
result[i][3*BJ[i][j]+0]+=U[3*node_comp[i][BI[i][j]]+0]*BV[i][j];//X
result[i][3*BJ[i][j]+1]+=U[3*node_comp[i][BI[i][j]]+1]*BV[i][j];//Y
result[i][3*BJ[i][j]+2]+=U[3*node_comp[i][BI[i][j]]+2]*BV[i][j];//Z
}
}
return;
}
```

Une fois cette opération faite, nous n'avons en réalité sur chaque composant qu'une partie du produit de la matrice B et du vecteur. En effet si l'on regarde la matrice B en détail, nous pouvons voir que pour chaque interface le calcul est effectué sur les sous-domaines des deux côtés de l'interface et donc sur deux composants mortier différents et indépendants :

$$B = \begin{bmatrix} B_{1,m} + B_{2,m} & -B_{1,s} & -B_{2,s} & 0 \\ 0 & B_{3,m} & 0 & -B_{3,s} \\ 0 & 0 & B_{4,m} & -B_{4,s} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.4.14)$$

La deuxième étape va consister à échanger les données. Pour une interface donnée, il est nécessaire que le produit de la matrice B et du vecteur soit stocké sur les deux composants mortier des sous-domaine correspondant à l'interface. Nous allons donc utiliser une méthode qui permet d'échanger ces données entre les deux composants. Le client va utiliser cette méthode sur le composant mortier côté mortier de l'interface, cette méthode a pour argument l'autre composant mortier ainsi que deux indices :

```
void set_mortar(ActionmatriceCI& component,const int indice1,const int indice2);
```

Les indices permettent de préciser le numéro de l'interface à chacun des composants, un composant pouvant avoir plusieurs interfaces. Le vecteur $B_{i,m} - B_{i,s}$ est ainsi calculé et stocké dans les deux composant mortier de l'interface i . Le fait que la méthode ait pour argument un autre composant permet de ne pas échanger de données avec le client mais seulement entre deux composant ce qui rend l'échange parallélisable. Cet échange est réalisé en parallèle à l'aide d'une version modifiée de l'algorithme de distribution de calcul :

Algorithm 4 Distribution de calcul pour l'échange de donné entre deux composants

```
1: int nb_proc_utilisé;
2: int nb_calcul;
3: std::vector<int> job (nombre_d'interface,1);
4: while nb_calcul < composant.size() do
5:     for unsigned i = 0; i < nombre_d'interface;i++) do
6:         if (job[i]>0)&&!composant[numéro_composant_mortier[i]] &&!com-
           posant[numéro_composant_esclave[i]] &&(nb_proc_utilisé <
           nb_proc_utilisable) then
7:             composant[numéro_composant_mortier[i]].set_mortar (nu-
               méro_composant_esclave[i],k,l);
8:             nb_proc_utilisé++;
9:             nb_proc_utilisé++;
10:            job[i]=0;
11:        end if
12:    end for
13: end while
```

Une fois BRc_i calculé nous pouvons calculer $B^T BRc_i$. Pour effectuer le produit de la matrice B^T nous avons comme pour le calcul du produit de la matrice B trois méthodes en fonction du besoin. Pour ce calcul nous n'aurons pas besoin d'échanger les données entre les sous-domaines.

La dernière étape va consister à calculer le produit de la matrice R_T avec le vecteur $B^T B R c_i$. Pour chaque sous-domaine nous allons utiliser la méthode `get_floc` qui a pour argument un vecteur où seront stockées les valeurs du de $R^T B^T B R c_i$. En 3D, ce vecteur sera composé de 6 valeurs par sous-domaine (correspondant aux trois forces et aux trois moments).

```
void get_Floc(std::vector<double>& F_loc);
```

Le client va uniquement stocker les vecteurs $R^T B^T B R c_i$ qui une fois assemblés donneront la matrice $R^T B^T B R$. Cette dernière aura pour taille six fois le nombre de sous-domaines. Les lignes et les colonnes correspondant aux degrés de libertés bloqués sont supprimées. La matrice ayant une petite taille nous pouvons utiliser la méthode du pivot de Gauss pour l'inverser et la stocker dans le client.

Résolution du problème

La partie initialisation étant terminée, nous allons maintenant décrire l'utilisation des composants pour la résolution du problème de décomposition de domaine.

Les composants doivent dans un premier temps récupérer les forces externes pour chacun des sous-domaines et les stocker dans un vecteur `dlambda`. Ce vecteur est stocké dans chaque composant mortier et n'est donc pas accessible par le client principal. en effet, seul les composants mortier et `coFeap` auront besoin de ce vecteur, le fait de stocker le vecteur dans le client aurait ralenti le code.

L'étape suivante consiste à calculer les mouvements de corps rigide. Le client va récupérer les forces globales de chaque sous-domaine. Si les forces d'un des sous-domaines non bloqués ne sont pas équilibrées nous devons calculer les mouvement de corps rigide. Le client va récupérer les forces globales à l'aide de la méthode `get_Floc` qui à déjà été utilisée.

La matrice des mouvements de corps étant stockée dans le client, celui ci peut, si cela est nécessaire, calculer les mouvements de corps rigide et les renvoyer aux composants mortier. La méthode `set_state` permet d'envoyer un vecteur à un composant mortier que celui ci va pouvoir stocker. La variable `name` permet d'indiquer au composant où stocker le vecteur.

```
void set_state(const std::string name, std::vector<double>& mortar);
```

Les forces étant maintenant équilibrées dans tous les sous-domaines, nous pouvons passer à la résolution du problème éléments finis. Pour cette étape, le client va utiliser la méthode `solve`. Lorsque l'on fait appel à cette méthode, le composant mortier va vérifier si il y a des forces appliquées au sous-domaine et va ensuite utiliser la méthode de résolution du composant `coFeap`. Le fait d'avoir utilisé une interface générale nous permet ici de changer le composant éléments finis sans avoir à modifier le composant mortier.

La dernière étape consiste à actualiser les valeurs du déplacement U et du vecteur de force Λ . Il est nécessaire pour cela de calculer le facteur d'amplification α qui diminue le résidu. La méthode `update` permet de mettre à jour un vecteur. La variable `name` indique quelle variable mettre à jour.

```
void update(const std::string name,const double alpha);
```

Si le résidu n'est pas suffisamment faible, on itère la procédure en calculant les nouvelles pseudo-forces `dlambda`.

Afin de montrer la capacité de notre code de calcul avec décomposition de domaine, deux exemples vont être étudiés. Un premier pour vérifier les capacités de prise en compte des hétérogénéités et un second pour valider le comportement dans le cas non-linéaire.

2.4.4 Cas d'une poutre en traction en élasticité

2.4.4.1 Présentation de la géométrie et des conditions aux limites

Le domaine étudié ici est une poutre de 30 cm de long avec une section carrée de 10cm de côté. La poutre est coupée en trois sous-domaines carrés, sur la figure 2.10 nous pouvons observer la décomposition de la poutre en trois sous-domaines. Chaque sous-domaine contient 20000 nœuds pour un total de 60000 nœuds. Nous allons ici étudier un cas homogène et un cas hétérogène. Les hétérogénéités sont ici des inclusions sphériques de 3 cm de diamètre.

La matrice a un module Young de 20 GPa et un coefficient de Poisson de 0.3 alors que les inclusions ont un module Young de 60 GPa et le même coefficient de Poisson.

Nous bloquons la face de gauche en déplacement et nous imposons un déplacement de 0.3 mm sur la face opposée.

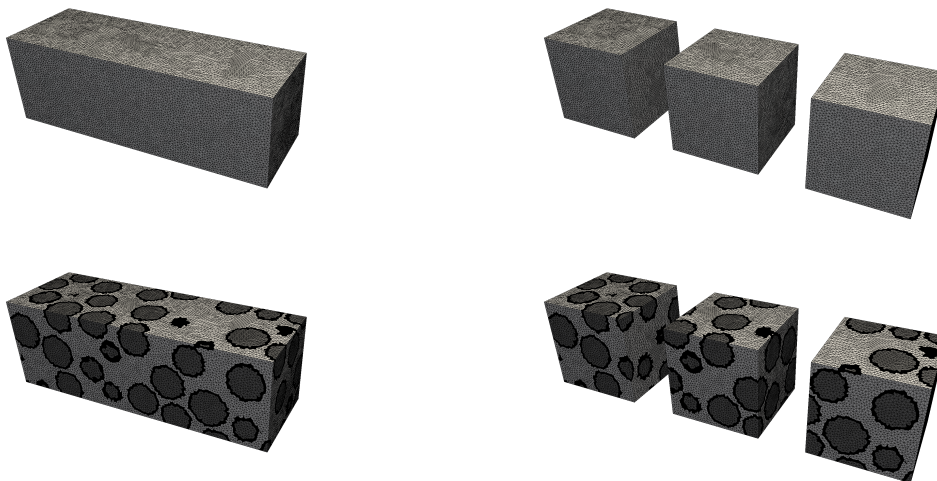


FIGURE 2.10 – Vue du domaine et des sous-domaines en homogène et en hétérogène

2.4.4.2 Résultats obtenus

Sur la figure 2.11 nous pouvons voir les déformations obtenues dans les deux cas. Nous pouvons noter dans les deux cas que la déformation obtenue est celle attendue et que la discontinuité de déplacement est bien vérifiée dans les deux cas.

Dans le cas hétérogène, nous observons bien l'impact des hétérogénéités sur la déformation. En zoomant sur l'interface, sur la figure 2.12 et en amplifiant la déformation, nous observons pour le sous-domaine du milieu que l'interface suit bien la déformation due aux inclusions.

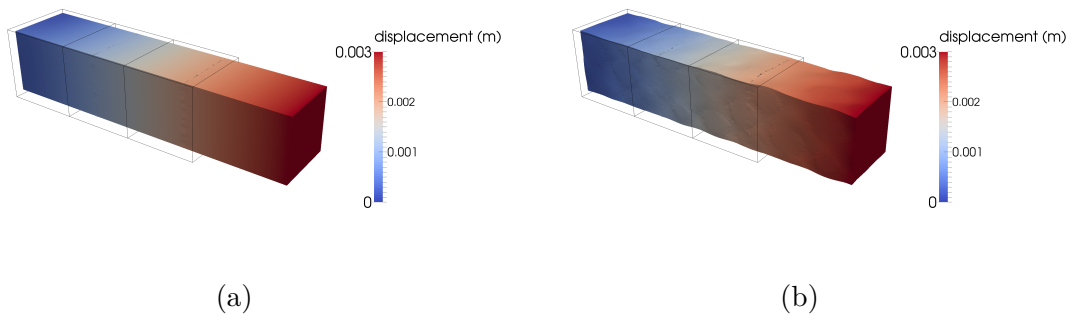


FIGURE 2.11 – Déformations obtenues pour (a) le cas homogène et (b) le cas hétérogène (facteur d'amplification : 400)

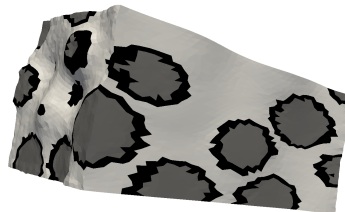


FIGURE 2.12 – Déformation obtenue sur le sous-domaine du milieu (facteur d'amplification : 800)

Si l'on regarde les contraintes obtenues (ici σ_{xx} , x étant la direction de chargement) dans le cas hétérogène sur la figure 2.13.a, nous notons que la contrainte est constante et vaut 20 MPa ce qui correspond à la valeur attendue.

Pour le cas hétérogène, la contrainte n'est plus constante ce qui s'explique par la présence d'hétérogénéités, les contraintes obtenues sont plus importantes dans les inclusions.

La méthode de mortier nous permet ici d'assurer une continuité du déplacement et une bonne représentation des contraintes malgré une non conformité au niveau des interfaces (les nœuds ne correspondent pas au niveau de l'interface).

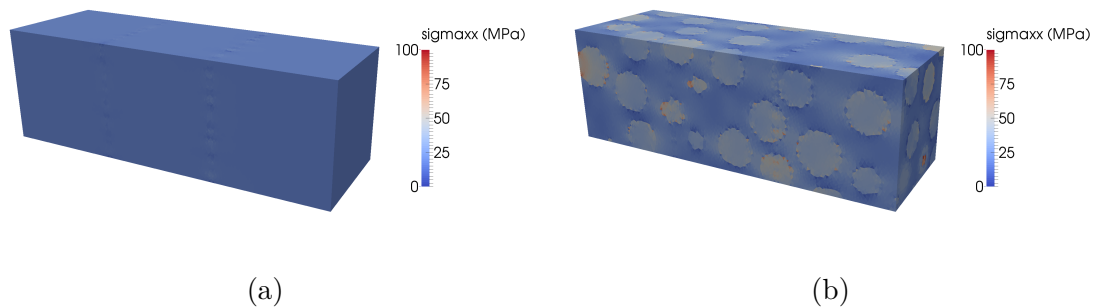


FIGURE 2.13 – σ_{xx} pour (a) le cas homogène et (b) le cas hétérogène

Si l'on regarde l'évolution du critère de convergence en fonction du nombre d'itérations sur la figure 2.14 on remarque que la présence d'inclusions n'a pas d'impact sur la vitesse de convergence.

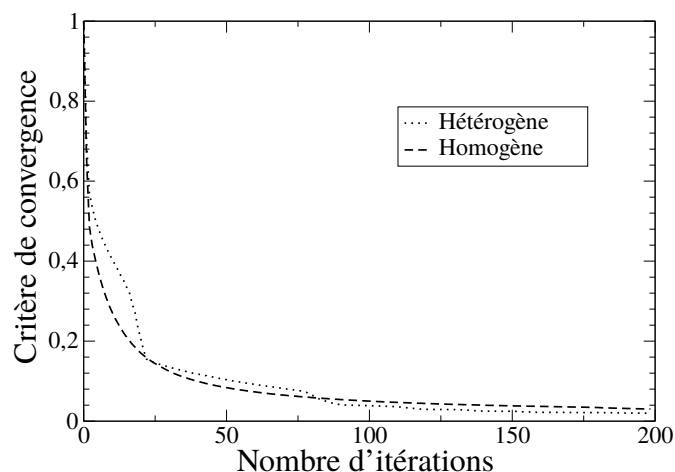


FIGURE 2.14 – Critère de convergence en fonction du nombre d'itérations (dans le cas homogène et dans le cas hétérogène)

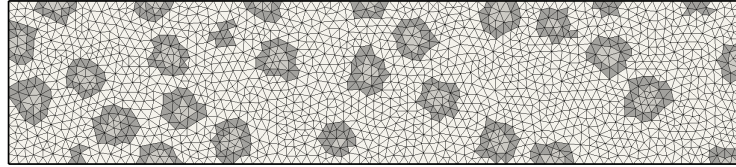
2.4.5 Cas d'une poutre en traction en non linéaire

2.4.5.1 Présentation de la géométrie et des conditions aux limites

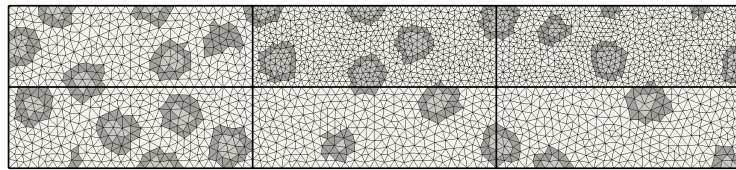
Nous avons utilisé ici deux maillages, un avec décomposition de domaine et un sans, afin de vérifier si la décomposition de domaine n'influe pas sur le comportement global de la structure. Pour ce cas, la structure est une poutre de 90 cm long, 20 cm de haut et 10 cm de large. La structure est hétérogène, les inclusions sont sphériques et ont un rayon de 3 cm de diamètre. Les inclusions sont les mêmes pour les deux cas.

Le maillage sans décomposition de domaine est composé de dix milles nœuds. Pour le maillage avec décomposition de domaine, quatre des sous-domaines ont deux milles nœuds, les deux autres sous-domaines ont cinq milles nœuds. Les deux maillages peuvent

être observés sur la figure 2.15.



(a)



(b)

FIGURE 2.15 – Maillage utilisé (a) sans décomposition de domaine et (b) avec décomposition de domaine

La poutre est bloquée en déplacement à gauche et on impose un déplacement horizontal à droite. La valeur du déplacement varie jusqu'à 0.3 mm (soit une déformation de 0.033

La matrice a un module Young de 20 GPa et les inclusions ont un module Young de 100 GPa. Les coefficients de Poisson sont les mêmes et valent 0.3.

Seule la matrice a un comportement non linéaire. On utilise ici les discontinuités fortes présentées dans le 1er chapitre et plus particulièrement le 1er modèle. La limite en traction est imposée à 4 MPa.

2.4.5.2 Résultats obtenus

Le comportement mécanique obtenu dans les deux cas est visible sur la figure 2.16. On observe dans un premier temps que le comportement est similaire dans les deux cas. Sur la partie linéaire les deux courbes sont superposées. Comme on a pu le voir dans la partie précédente, la décomposition de domaine nous permet d'avoir le bon comportement linéaire.

La valeur au pic est la même et vaut ici 3 MPa. Dans les deux cas le comportement observé est fragile et semble être le même. Pour le calcul sans décomposition de domaine la contrainte semble osciller, ceci s'explique par un faible nombre d'itération pour le calcul non linéaire (15 itérations par pas de déplacement) alors que dans le calcul avec décomposition de domaine le nombre d'itération était de 100, ainsi le résultat semble plus stable.

Si l'on regarde maintenant le faciès de fissuration obtenu après le pic, sur la figure 2.17, on remarque dans les deux cas la présence d'une macro fissure. On observe que les deux macro-fissures ne se situent pas aux mêmes endroits. Ceci peut être expliqué par la

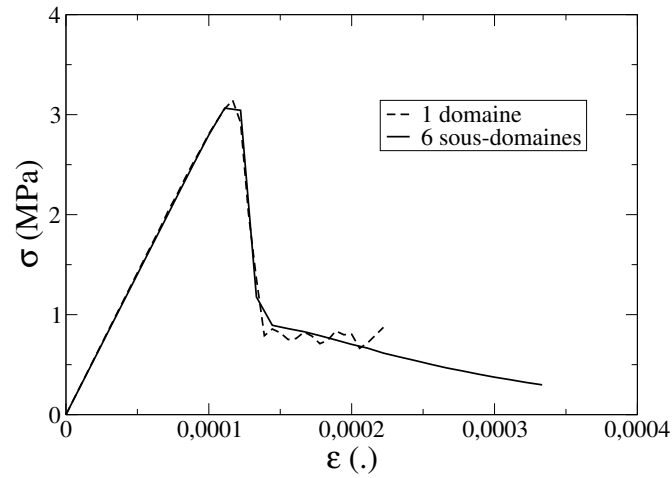


FIGURE 2.16 – Comportement global obtenu avec et sans décomposition de domaine

différence des deux maillages (nombre de nœuds, position des nœuds, raffinement d’une zone dans le cas de la décomposition de domaine...) ce qui a pu avoir un impact sur la micro-fissuration et sur la nucléation des fissures amenant à la macro-fissure.

Il est important de noter ici que malgré le fait qu’il n’y ai pas de dialogue entre les sous-domaines au niveau de la fissuration, la macro fissure traverse parfaitement deux sous-domaines. Ce résultat est intéressant et était nécessaire à la réalisation d’un essai de fissuration.

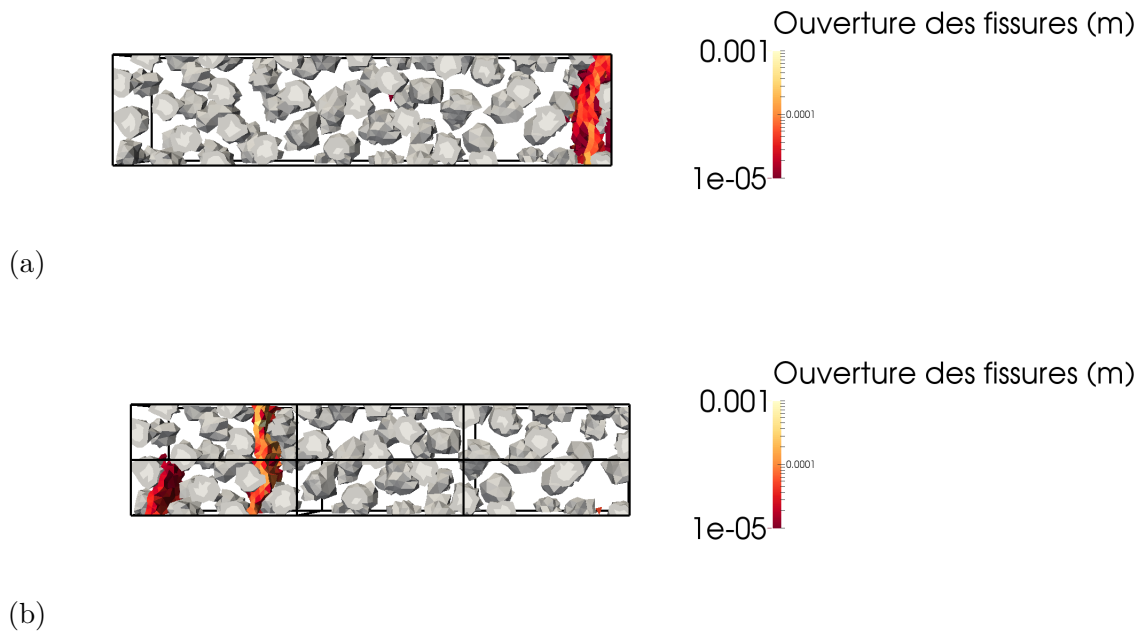


FIGURE 2.17 – Faciès de fissuration obtenu (a) sans décomposition de domaine et (b) avec décomposition de domaine

2.5 Scalabilité du code de calcul

Dans cette partie nous allons étudier la calculabilité de notre code de calcul. L'exemple utilisé ici comprend 944 sous-domaines d'environ 4000 nœuds chacun. Le nombre total de nœuds est de 4000000.

Pour étudier plus précisément l'impact du nombre de processeur sur la rapidité du calcul, nous allons étudier la scalabilité de celui ci. La scalabilité est défini ici comme le rapport entre le temps de calcul avec un processeur et le temps de calcul avec n processeurs multiplié par le nombre de processeurs soit :

$$S = \frac{t_1}{t_n * n} \quad (2.5.1)$$

La scalabilité est parfaite lorsque s vaut 1.

Dans un premier temps on peut observer le temps de calcul en fonction du nombre de processeur sur la figure 2.18.a ainsi que la scalabilité 2.18 . On note dans un premier temps que le temps de calcul varie de plus de 400 minutes à moins de 100 minutes avec un nombre de processeurs variant de 24 à 320.

Le temps de calcul diminue rapidement jusqu'à l'utilisation de 192 processeurs. Au delà le temps de calcul semble stagner à environ 100 minutes. Si l'on regarde maintenant la variable de scalabilité en fonction du nombre de processeurs, on remarque que celle ci est égale ou proche de 1 jusqu'à l'utilisation de 64 processeurs. Au delà de cette valeur la scalabilité diminue.

La diminution de la scalabilité s'explique par la quantité de plus en plus importante de données à transférer entre les différents nœuds du calcul. On remarque un pic aux alentours des 256 processeurs. Ce pic qui correspond à une amélioration du fonctionnement du code uniquement pour un nombre de processeurs donné ne devrait pas être présent. Celui ci peut s'expliquer par une utilisation moins importante du serveur au moment du calcul et ainsi par une diminution du temps de calcul due à une amélioration de la vitesse de transfert entre les différents nœuds du serveur.

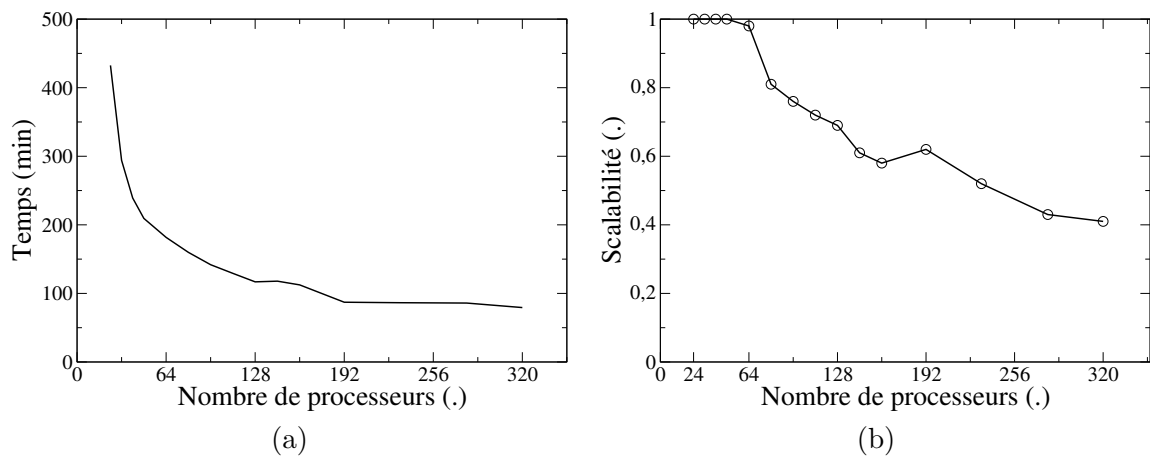


FIGURE 2.18 – Temps de calcul et scalabilité en fonction du nombre de processeur

Sur les graphiques suivants, on s'intéresse à nouveau au temps de calcul 2.19.a et à la scalabilité 2.19.b mais cette fois ci, en fonction du nombre de sous-domaines calculé sur chaque processeur.

Concernant le temps de calcul, on remarque qu'en dessous de 5 sous-domaines par processeur le temps de calcul stagne. En dessous de cette valeur, le temps gagné par une résolution plus rapide des calculs éléments finis locaux est moins important que le temps engendré par le transfert des données à l'échelle globale. On peut se fixer cette valeur comme valeur seuil à ne pas dépasser pour que l'utilisation du serveur soit optimale. Si l'on regarde la valeur de la scalabilité, on observe une diminution rapide en dessous de 5 sous-domaines par processeur ce qui correspond à une moindre efficacité de la parallélisation.

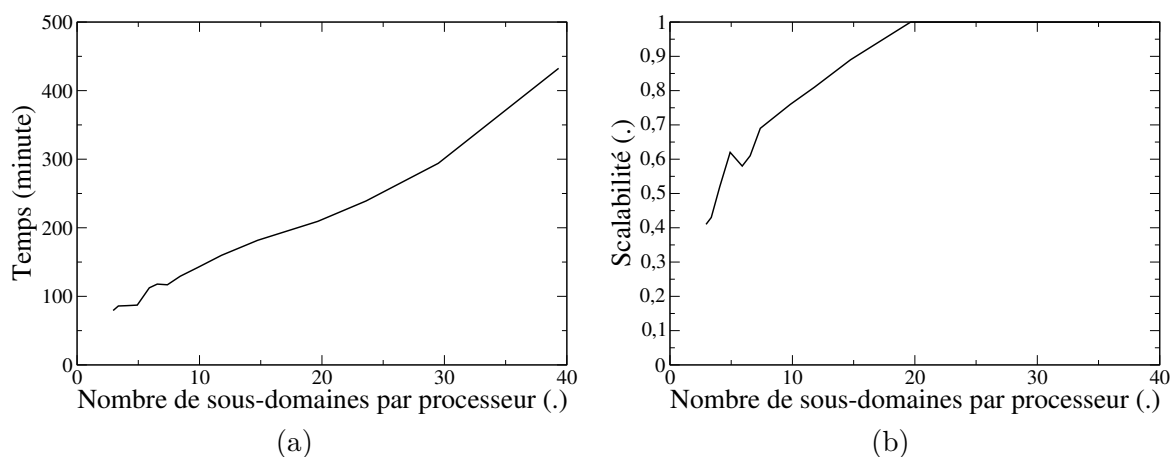


FIGURE 2.19 – Temps de calcul et scalabilité en fonction du nombre de sous-domaines par processeur

2.6 Conclusion

Les problèmes de grandes tailles demandent un temps de calcul important. Pour parler à ce problème, les méthodes de décomposition de domaine peuvent être utilisées. Ces méthodes consistent à découper le domaine en sous-domaines. Chaque sous-domaine est indépendant informatiquement et le problème mécanique associé peut être résolu indépendamment. Les sous-domaines comportant moins de nœuds, le calcul mécanique y est plus rapide. Il reste néanmoins à assurer la continuité des variables entre les eux.

Parmi les méthodes existantes, la méthode de mortier à été choisie puis intégrée à un code de calcul maison. Un des avantages de cette méthode est qu'elle autorise l'utilisation de maillages non conformes aux interfaces, c'est à dire dont les points ne sont pas en vis à vis au niveau de l'interface. Cette particularité autorise l'utilisation de maillage raffiné par zone et de générer plus simplement le maillage éléments finis.

Pour assurer l'indépendance informatique entre les sous-domaines, nous avons employé la programmation orientée composant et plus précisément le middleware CTL. À l'aide de celui ci, un composant mortier à été créé. Ce composant englobe un autre composant

éléments finis coFeap qui contient le maillage éléments finis d'un sous-domaine.

Le composant mortier peut être vu comme une brique élémentaire. Le client peut l'appeler à travers différentes méthodes pour lui affecter des tâches. Le rôle du client est alors de résoudre le problème global à l'aide d'un algorithme maison en faisant appel aux différentes méthodes briques locales.

Le code de calcul ainsi créé permet de résoudre des problèmes mécaniques. La méthode E-FEM, présentée dans le premier chapitre, a été intégrée à ce code de calcul. Cette intégration a permis de résoudre, à l'aide de la décomposition de domaine, un calcul non linéaire avec prise en compte d'hétérogénéités dans un temps plus court que si l'on avait pas utilisé la décomposition de domaine.

À travers un premier exemple simple, il a été montré que la présence d'hétérogénéités ne posait pas de problème lors de la résolution. Un second exemple, comparant le comportement non linéaire avec et sans décomposition d'une même minéralogie, a mis en avant que le comportement mécanique n'était pas influencé par l'utilisation de la décomposition de domaine. L'outil développé ici est donc apte à résoudre des problèmes de fissuration de grande taille.

Enfin nous avons vérifié la capacité de notre code de calcul à utiliser pleinement les ressources informatiques. Pour cela nous avons lancé un même calcul en utilisant un nombre différent de processeurs afin d'évaluer la performance du code de calcul. Nous avons noté une dégradation des performances au delà de cinq sous-domaines par processeur.

Un travail sur le composant mortier pourra être effectué afin d'accélérer ou de limiter les échanges de données entre les composants, point faible de la programmation utilisée ici. De plus afin d'améliorer les performances globales du code de calcul, l'algorithme utilisé peut être amélioré dans le but de limiter le nombre d'itérations nécessaires à la convergence du calcul et ainsi à diminuer un peu plus le temps de calcul.

Chapitre 3

Modélisation par décomposition de domaine d'un essai de fissuration

Sommaire

3.1	Introduction	93
3.2	Essai de fissuration	93
3.2.1	Paramètres matériaux et géométriques	93
3.2.2	Conditions aux limites	94
3.2.3	Résultats mécaniques	95
3.3	Couplage Hydro-mécanique	97
3.3.1	Formulation du problème hydraulique avec un fluide incompressible dans le cadre de la méthode E-FEM	97
3.3.1.1	Formulation forte d'un problème hydraulique	97
3.3.1.2	Formulation faible d'un problème hydraulique	98
3.3.1.3	Méthode E-FEM appliquée à un problème hydraulique	101
3.3.2	Couplage entre le calcul mécanique et le calcul hydrique	102
3.3.2.1	Débit à travers une fissure	102
3.3.2.2	Passage aux éléments finis	102
3.3.2.3	Perméabilité d'un cube coupé par une fissure plane	103
3.3.2.4	Perméabilité d'un matériau hétérogène lors d'un essai de traction	105
3.4	Calcul de fissuration avec le couplage mécanique-hydraulique faible	109
3.5	Conclusion	117

3.1 Introduction

La fracturation hydraulique, mise en place dans un premier temps pour les gisements d'hydrocarbures conventionnels est une étape nécessaire à l'exploitation des hydrocarbures non-conventionnels. Celle ci consiste, en imposant une contrainte mécanique à l'aide d'un fluide, à augmenter la perméabilité de la roche en la fracturant. Cette technique est utilisée, dans le cas des hydrocarbures non-conventionnels avec la technique de forage horizontal. Le forage horizontal permet de maximiser la zone stimulée du réservoir.

Une fois le forage horizontal effectué, des perforations à travers le coffrage sont effectuées au bout du forage. Ces trous servent de points d'entrées au fluide injecté sous pression. Les contraintes mécaniques imposées par le fluide vont fissurer la roche en partant de ces perforations.

La connaissance du processus de fissuration au niveau de cette perforation est une donnée importante pour les industriels. L'étude proposée ici consiste à modéliser ce processus. Afin de répondre à cette problématique nous réalisons un essai mécanique correspondant aux contraintes imposées en pointe de fissure.

Les modèles à enrichissement cinématique permettent d'avoir accès aux ouvertures ainsi qu'aux orientations des fissures. Ces données sont essentielles pour la réalisation d'un calcul hydraulique en prenant en compte la présence de fissures. En modélisant les fissures par deux plans parallèles il est possible d'utiliser une loi de Poiseuille et ainsi de connaître le débit à travers la fissure. Un tel modèle sera décrit dans cette partie et utilisé pour calculer la perméabilité induite par une fissuration issue d'un calcul mécanique.

3.2 Essai de fissuration

Le calcul idéal est représenté sur la figure 3.1. Par symétrie nous avons simplifié le problème en gardant uniquement la pression de droite

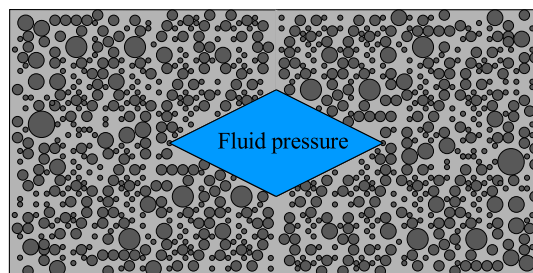


FIGURE 3.1 – Modélisation de la fracturation hydraulique

La pression de fluide sera modélisée par une contrainte appliquée à l'intérieur de l'entaille.

3.2.1 Paramètres matériaux et géométriques

Les calculs sont réalisés avec un seul maillage. Celui ci est composé de 148 sous-domaines ayant entre 5000 et 8000 nœuds. Le nombre total de ddl est de 3.10^6 . La

décomposition est visible sur la figure 3.2.

Le domaine est composé d'une plaque entaillée de 9.6 mm de large par 4 mm de hauteur et 0.5 mm de profondeur. Les tétraèdres ont une dimension de l'ordre de grandeur de 15 micromètres.

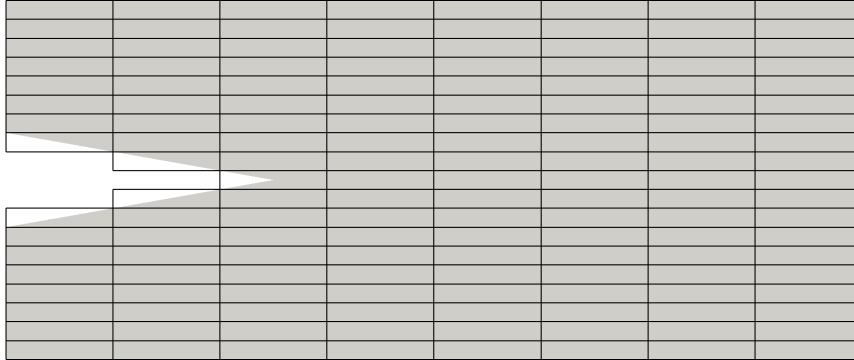


FIGURE 3.2 – Répartition des sous-domaines sur une plaque entaillée

Les différentes minéralogies sont réalisées en combinant des inclusions de quartz de calcite et de kérogène. Les propriétés mécaniques des différents matériaux sont les suivantes :

- Matrice
 - Module Young : 20 GPa
 - Coefficient de Poisson : 0.35
 - Résistance à la traction : 4 MPa
- Quartz
 - Module Young : 55 GPa
 - Coefficient de Poisson : 0.3
- Calcite
 - Module Young : 47 GPa
 - Coefficient de Poisson : 0.12
- Kérogène
 - Module Young : 10 GPa
 - Coefficient de Poisson : 0.17

3.2.2 Conditions aux limites

Les conditions de symétrie sont appliquées sur le domaine. Sur la partie gauche du maillage (en $x=0$) les déplacements sont bloqués suivant la direction x . Pour le sous-domaine en bas à gauche (en $x=0$ et $y=0$) on impose en déplacement nul suivant y . Cette condition permet d'éviter les mouvements de corps rigides. Les déplacements en $z=0$ sont bloqués suivant z . Les autres faces ne sont pas bloquées.

Une pression est imposée à l'intérieur de l'entaille et varie de 0 à 6 MPa. Celle ci est visible sur le schéma 3.3.

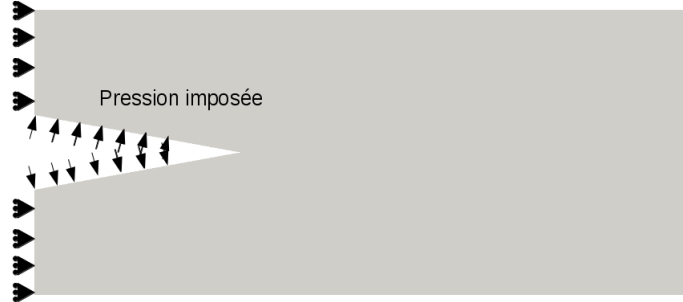


FIGURE 3.3 – Conditions aux limites et contraintes imposées pour le calcul numérique

Le calcul est réalisé sur les serveurs de grid5000. Ces serveurs sont stockés dans différentes villes (Lille, Rennes, Nantes ...) et sont accessibles à la recherche. Une des contraintes de l'utilisation de ces serveurs est que les calculs de production doivent être effectués entre 19h et 8h (soit 14h continu).

Les calculs ont été lancés sur le site de Lille. 12 nœuds de 8 processeurs (soit 96 processeurs) ont été utilisés pour chaque calcul permettant de respecter une plage horaire de 14 pour un calcul non-linéaire complet de 3.10^6 ddls. Une telle performance ne serait pas possible sans l'utilisation de la décomposition de domaine.

3.2.3 Résultats mécaniques

Sur la figure 3.4 on peut observer des exemples de faciès de fissuration obtenus pour une pression imposée de 0.6 MPa (soit la pression maximale). La fissure semble fortement dépendre du placement des inclusions. Celles-ci étant considérées comme purement élastiques, la fissure doit les contourner. Sur la première figure on peut observer que la fissure est globalement horizontale alors que pour les deux autres faciès elle part avec un angle plus ou moins important. Cette bifurcation peut être due à la présence d'une grosse inclusion près de la pointe d'entaille.

La longueur des fissures mesurée en fonction de la pression imposée dans l'entaille est visible sur la figure 3.6. On note la présence de paliers correspondant au passage de la frontière d'un sous-domaine. Bien que les calculs aient convergés, il semble nécessaire d'ajouter des itérations supplémentaires. Cependant on remarque une différence entre les différentes minéralogies.

Le facteur d'intensité des contraintes $K_{1,c}$ peut être exprimé en fonction de la pression imposée, de la longueur de fissure par la formule suivante [Adachi and Detournay,] :

$$K_{1,c} = p\sqrt{\pi l} \quad (3.2.1)$$

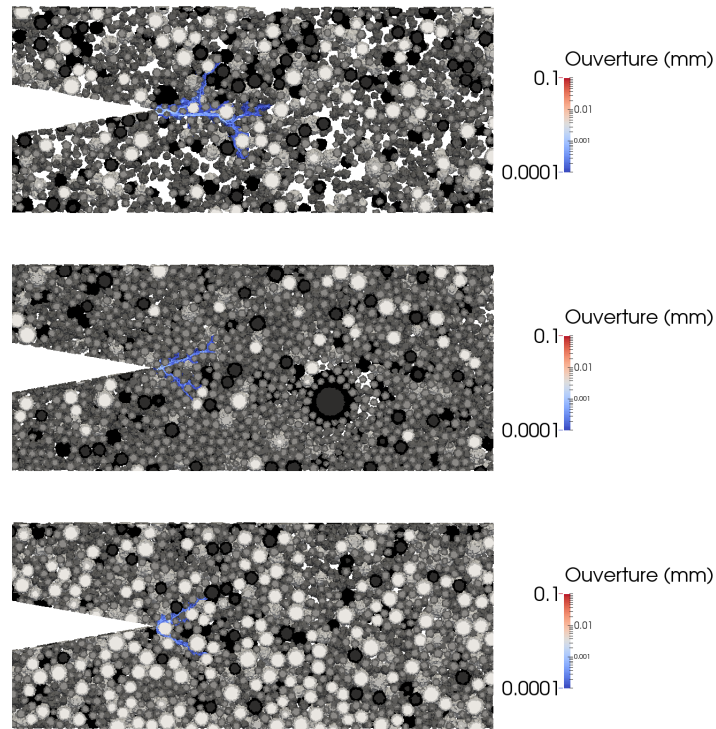


FIGURE 3.4 – Exemple de faciès de fissuration à 6 MPa pour différentes minéralogies

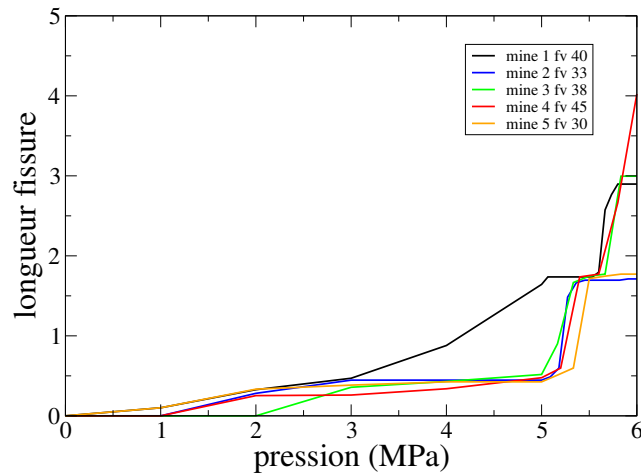


FIGURE 3.5 – Longueur de fissure mesurée pour les différentes minéralogies

En utilisant cette formule ainsi que les données calculées on peut tracer la ténacité en fonction de la fraction volumique total d'inclusion :

Dans un premier temps, on remarque que globalement la ténacité augmente lorsque la fraction volumique des inclusions augmente. La ténacité varie alors de 0.4 à $0.7 \text{ MPa}\cdot\text{m}^{1/2}$, ces valeurs sont légèrement plus faibles que celles mesurées par micro-indentation ([Xie et al., 2012], [Su et al.,]) pour les puits de Vaca Muerta (1.4 à $3.4 \text{ MPa}\cdot\text{m}^{1/2}$). Cette

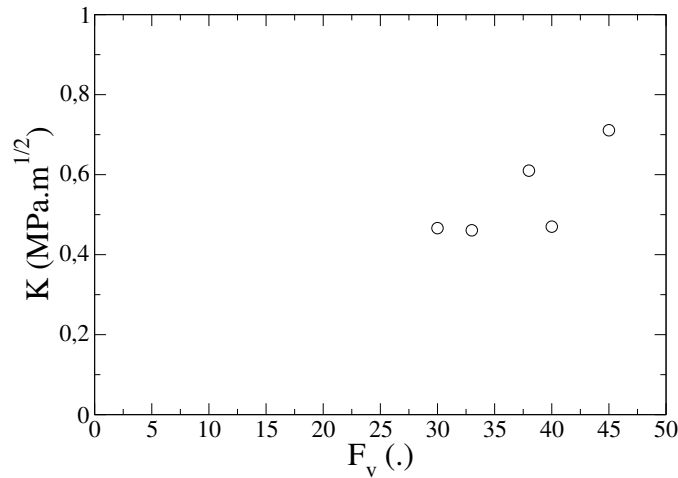


FIGURE 3.6 – Ténacité obtenue en fonction de la fraction volumique des inclusions

différence peut s'expliquer par la plus grande fraction volumique de matrice utilisée lors des calcul (55 à 70 %) comparé au 15-30% observé dans le milieu. La ténacité semblant augmentée lorsque la fraction volumique de matrice diminue il semble logique d'obtenir des valeurs plus faibles que celles observées numériquement.

La faible valeur de ténacité pour une fraction volumique de 40 peut s'expliquer par le caractère très local de l'essai de fissuration. En effet à la différence de l'essai de compression triaxiale pour lequel toutes les inclusions ont un impact, ici seul les inclusions proches de l'entaille ont un impact. Le placement de celle ci joue un rôle important dans le résultat. De plus, comme nous avons pu le voir précédemment, la décomposition de domaine semble avoir un impact sur la longueur des fissures et peut ainsi ajouter un biais dans nos résultats.

Pour palier à ce problème il serait intéressant de raffiner fortement le maillage en pointe d'entaille pour pouvoir y inclure des sphères plus petites et ainsi diminuer l'impact de leur placement.

3.3 Couplage Hydro-mécanique

Dans cette partie nous allons présenter le modèle hydraulique avec discontinuité faible ainsi que le couplage effectué entre le calcul mécanique et le calcul numérique. Un tel couplage a été réalisé dans le cadre de la thèse de Xavier Jourdain au LMT ([[Jourdain, 2014](#)], [[Jourdain et al., 2014](#)]) dans le cas d'un treillis. Les travaux, présentés dans cette section, en reprennent les grandes idées en les appliquant aux tétraèdres.

3.3.1 Formulation du problème hydraulique avec un fluide incompressible dans le cadre de la méthode E-FEM

3.3.1.1 Formulation forte d'un problème hydraulique

Le calcul hydraulique est réalisé dans le cadre des hypothèses suivantes :

- Le milieu étudié ici est considéré comme saturé en fluide.
- Le fluide est incompressible et son écoulement est laminaire.
- L'effet de la gravité est négligé.

L'équation d'équilibre de problème étudié correspond à la forme conservative de la première équation de Navier-Stokes. Celle ci provient du fait que la somme des débits massiques entrants est égale à la somme des débits massiques sortants :

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \underline{v}) = 0 \quad (3.3.1)$$

Où ρ est la masse volumique du fluide et \underline{v} est le vecteur vitesse au sens de Darcy.

Une des hypothèses du calcul est que celui ci est stationnaire, le terme $\frac{\partial \rho}{\partial t}$ s'annule, l'équation devient alors :

$$\text{div}(\rho \underline{v}) = 0 \quad (3.3.2)$$

Le calcul est effectué dans le cas d'un fluide incompressible, la masse volumique étant homogène le terme ρ peut être sorti de l'équation :

$$\text{div}(\underline{v}) = 0 \quad (3.3.3)$$

Il est nécessaire pour résoudre cette équation d'introduire une relation de comportement entre la vitesse \underline{v} et la pression p . La loi de Darcy est utilisée pour exprimer la vitesse débitante \underline{v} en fonction de la pression p :

$$\underline{v} = \frac{k}{\mu} \cdot \underline{\text{grad}}(p) \quad (3.3.4)$$

Où k correspond à la perméabilité du matériau et μ la viscosité dynamique du fluide.

La solution du problème étant complexe à obtenir, il est possible d'obtenir une solution approchée en utilisant la méthode des éléments finis. La formulation doit être écrite sous une forme faible puis projeté sur le maillage discrétisé.

3.3.1.2 Formulation faible d'un problème hydraulique

La mise en place de la formulation faible débute par la définition des espaces de travail : $S(\Omega)$ est l'espace des champs de pression admissible :

$$S(\Omega) = \{p \in H^1(\Omega) | p(\delta_\Omega) = p_0\} \quad (3.3.5)$$

Où $H^1(\Omega) = \{p \in L^2(\Omega) | p \in L^2(\Omega)\}$ est l'espace de Sobolev, $L^2(\Omega) = \{p | \int_\Omega p^2 d\Omega < \infty\}$ l'espace des fonctions de carré intégrables et $V\Omega$ l'espace des champs de pression admissible à zéro.

En multipliant par une pression virtuelle p^* et en intégrant l'équation sur le domaine Ω , le problème peut dans un premier temps se mettre sous la forme :

Trouver $p \in S(\Omega) | \forall p^* \in V(\Omega)$ tel que :

$$\int_{\Omega} p^* \cdot \text{div}(\underline{q}(p)) \, \mathbf{d}\Omega = 0 \quad (3.3.6)$$

En intégrant cette équation par partie on obtient :

$$\forall p^*, \int_{\Omega} \text{div}(p^* \cdot \underline{q}(p)) \, \mathbf{d}\Omega - \int_{\Omega} \underline{\text{grad}}(p^*) \cdot \underline{q}(p) \, \mathbf{d}\Omega = 0 \quad (3.3.7)$$

En utilisant le théorème de Green-Ostrogradski l'équation devient :

$$\forall p^*, \int_{\delta\Omega} p^* \cdot \underline{q}(p) \, \mathbf{d}S - \int_{\Omega} \underline{\text{grad}}(p^*) \cdot \underline{q}(p) \, \mathbf{d}\Omega = 0 \quad (3.3.8)$$

p^* appartenant au champ de pression admissible à zéro (et en utilisant la loi de comportement) on peut mettre l'équation sous la forme :

$$\forall p^*, \int_{\delta\Omega_q} p^* \cdot \underline{q}_0 \, \mathbf{d}S - \int_{\Omega} \underline{\text{grad}}(p^*) \cdot \frac{-k}{\mu} \underline{\text{grad}}(p) \, \mathbf{d}\Omega = 0 \quad (3.3.9)$$

Cette formulation correspond à la formulation faible du problème. La prochaine étape consiste à projeter cette formulation sur une maillage éléments finis.

La résolution par la méthode des éléments finis permet d'obtenir une solution " \tilde{p} " de la solution exacte " p " issue de la formulation forte. Le domaine Ω est découpé en plusieurs éléments " e ". Il suffit ensuite de poser que la pression " \tilde{p} " est le produit scalaire entre la valeur de " \underline{p}^e " au nœud et les fonctions de forme de chacun des éléments définissant le domaine \underline{N} . Soit :

$$\tilde{p} = \underline{N} \cdot \underline{\tilde{p}}^e \quad (3.3.10)$$

Pour \underline{p}^* on obtient de la même façon :

$$\tilde{p}^* = \underline{N} \cdot \underline{\tilde{p}}^{e*} \quad (3.3.11)$$

On définit le gradient de pression par :

$$\underline{\text{grad}}(\underline{p}^e) = \underline{\underline{B}} \cdot \underline{p}^e \quad (3.3.12)$$

La matrice $\underline{\underline{B}}$ sera définie plus tard et dépend des fonctions de forme \underline{N} .

La formulation faible peut maintenant s'écrire :

$$A_e \int_{\delta\Omega_q} \underline{N} \cdot \underline{\tilde{p}}^{e*} \cdot \underline{q}_0 \, \mathbf{d}S - A_e \int_{\Omega} \underline{\underline{B}}^T \underline{\tilde{p}}^{e*} \cdot \frac{-k}{\mu} \underline{p}^e \cdot \underline{\underline{B}} \, \mathbf{d}\Omega = 0 \quad (3.3.13)$$

L'équation doit être vérifiée pour tout \underline{p}^{e*} et en particulier lorsque ses composantes sont constantes. En prenant $\underline{p}^{e*} = \underline{I}$ l'équation devient :

$$A_e \int_{\delta\Omega_q} \underline{N} \cdot \underline{q}_0 \, \mathbf{d}S - A_e \int_{\Omega} \underline{\underline{B}}^T \cdot \frac{-k}{\mu} \underline{p}^e \cdot \underline{\underline{B}} \, \mathbf{d}\Omega = 0 \quad (3.3.14)$$

On pose alors :

$$\underline{f}^{ext} = A_e \int_{\delta\Omega_q} \underline{N} \cdot \underline{q}_0 dS \quad (3.3.15)$$

$$\underline{f}^{int} = A_e \int_{\Omega} \underline{B}^T \cdot \frac{-k}{\mu} \underline{p}^e \cdot \underline{B} d\Omega \quad (3.3.16)$$

Pour résoudre le problème il faut minimiser le résidu \underline{r}

$$\underline{r} = \underline{f}^{ext} - \underline{f}^{int} \quad (3.3.17)$$

Pour les tétraèdres utilisés dans le premier chapitre de cette thèse la matrice \underline{B} peut s'écrire sous la forme :

$$\underline{B} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} & \frac{\partial N_4}{\partial z} \end{pmatrix} \quad (3.3.18)$$

Avec N_1, N_2, N_3 et N_4 les fonctions de formes associées aux nœuds du tétraèdre.

Le système obtenu pouvant être complexifié ultérieurement lors du couplage hydro-mécanique, il est nécessaire d'effectuer une linéarisation cohérente du problème. Pour obtenir une solution correcte au pas de temps $t + 1$ il sera nécessaire d'effectuer des itérations sur des pas de temps fictifs k en partant de la solution au pas de temps réel t .

Le problème linéarisé est obtenu en tronquant la série de Taylor après les deux premiers termes. Au pas de temps virtuel k la linéarisation au pas de temps $t + 1$ est la suivante :

$$L(\underline{r}_{t+1}^{k+1}) = \underline{r}_{t+1}^k + \left. \frac{\delta \underline{r}}{\delta p} \right|_{t+1}^k \Delta \underline{p}_{t+1}^{k+1} = 0 \quad (3.3.19)$$

Le terme $\Delta \underline{p}_{t+1}^{k+1}$ représente un incrément du vecteur de pression. Pour trouver ce terme il suffit de résoudre l'équation :

$$\Delta \underline{p}_{t+1}^{k+1} = \left(\left. \frac{\delta \underline{r}}{\delta p} \right|_{t+1}^k \right)^{-1} \underline{r}_{t+1}^k \quad (3.3.20)$$

Il reste ensuite à déterminer la forme de $\left. \frac{\delta \underline{r}}{\delta p} \right|_{t+1}^k$:

$$\left. \frac{\delta \underline{r}}{\delta p} \right|_{t+1}^k = \left. \frac{\delta f^{int}}{\delta p} \right|_{t+1}^k = A_e \int_{\Omega} \underline{B}^T \cdot \frac{\delta v}{\delta p} d\Omega \quad (3.3.21)$$

La relation de comportement, ici la loi de Darcy, est ensuite utilisée pour exprimer v^e en fonction de p^e (pour chaque élément e du maillage) :

$$\underline{v}^e = -\frac{1}{\mu} \underline{k}^e \cdot \underline{\text{grad}}(p^e) \quad (3.3.22)$$

Cette relation nous permet d'exprimer $\frac{\delta v^e}{\delta p^e}$:

$$\frac{\delta v^e}{\delta p^e} = -\frac{1}{\mu} \underline{k}^e \underline{B}^e \quad (3.3.23)$$

En introduisant cette relation dans l'équation 3.3.21 on obtient :

$$\frac{\delta \underline{r}}{\delta p} \Big|_{t+1}^k = \frac{1}{\mu} A_e \int_{\Omega^e} \underline{B}^{eT} \underline{k}^e \underline{B}^e d\Omega \quad (3.3.24)$$

Au final, la pression au pas de temps $t + 1$ et au pas de temps fictif $k + 1$ est donnée par la formule :

$$\underline{p}_{t+1}^{k+1} = \underline{p}_{t+1}^k + \Delta \underline{p}_{t+1}^{k+1} \quad (3.3.25)$$

L'incrément de pression est obtenu par l'équation suivante :

$$\Delta \underline{p}_{t+1}^{k+1} = \left(\frac{1}{\mu} A_e \int_{\Omega^e} \underline{B}^{eT} \underline{k}^e \underline{B}^e d\Omega \right)^{-1} \underline{r}_{t+1}^k \quad (3.3.26)$$

3.3.1.3 Méthode E-FEM appliquée à un problème hydraulique

Les principes ainsi que la mise en place de discontinuité faible ayant été évoqués dans le chapitre 1 dans le cadre de la mécanique, la mise en place de celle ci pour un problème hydraulique sera évoquée succinctement ici, pour plus de détail voir la partie concernée dans le chapitre 1.

La discontinuité faible est présente dans le cas d'un calcul hydraulique dans le champ de gradient de pression (correspondant au champ de déformation en mécanique). la discontinuité faible du champ de gradient de pression est notée $[[\text{grad}(p)]]$. Les équations (1.3.52) et (1.3.53) sont réutilisées avec $G_s = 0$ car il n'y a pas de discontinuité forte et les tenseurs de contraintes \underline{C}^\oplus et \underline{C}^\ominus sont remplacés par les tenseurs de perméabilités $\frac{1}{\mu} \underline{k}^\oplus$ et $\frac{1}{\mu} \underline{k}^\ominus$.

Linéarisation de l'équation d'équilibre global

$$\begin{aligned} f_{int}^e = \int_{\Omega} B^T \hat{q}(Bd + G_w [[\text{grad}(p)]])) &= \underbrace{B^T \left(V^\oplus \frac{1}{\mu} \underline{k}^\oplus + V^\ominus \frac{1}{\mu} \underline{k}^\ominus \right)}_{Kblabla} B d \\ &+ \underbrace{\frac{V^\oplus V^\ominus}{V} B^T \left(\frac{1}{\mu} \underline{k}^\oplus - \frac{1}{\mu} \underline{k}^\ominus \right) \underline{H}_w}_{Kblabla} [[\text{grad}(p)]] \end{aligned} \quad (3.3.27)$$

Linéarisation de l'équation de discontinuité faible

$$\begin{aligned} \int_{\Omega} \epsilon \hat{G}_w \hat{q}(Bd + G_w [[\text{grad}(p)]])) d\Omega &= \frac{V^\oplus V^\ominus}{V} \underline{H}_w^T \left(\frac{1}{\mu} \underline{k}^\oplus - \frac{1}{\mu} \underline{k}^\ominus \right) B d \\ &+ \frac{V^\oplus V^\ominus}{V^2} \underline{H}_w^T \left(V^\ominus \frac{1}{\mu} \underline{k}^\oplus - V^\oplus \frac{1}{\mu} \underline{k}^\ominus \right) \underline{H}_w [[\text{grad}(p)]] \end{aligned} \quad (3.3.28)$$

Le système complet peut alors se mettre sous la forme matricielle suivante :

$$\begin{bmatrix} \underline{\underline{K_{H,bb}}} & \underline{\underline{K_{H,bw}}} \\ \underline{\underline{K_{H,wb}}} & \underline{\underline{K_{H,ww}}} \end{bmatrix}_{n+1}^k \left\{ \begin{array}{c} \Delta d \\ \Delta[\|\text{grad}(p)\|] \end{array} \right\}_{n+1}^{k+1} = \left\{ \begin{array}{c} -A_{e=1}^{n_{elm}} [f_{int}^e - f_{ext}^e] \\ h \end{array} \right\}_{n+1}^k \quad (3.3.29)$$

La procédure de résolution est la même que celle présentée dans le premier chapitre (en utilisant la condensation statique et un solveur de type BFGS + line-search). Cette procédure est intégrée dans le logiciel éléments finis coFeap.

3.3.2 Couplage entre le calcul mécanique et le calcul hydrique

3.3.2.1 Débit à travers une fissure

Nous allons considérer ici que la fissure peut être représentée par deux plans parallèles distants de la valeur de l'ouverture de fissure. Dans le cas d'un écoulement entre deux plans parallèles nous pouvons utiliser un écoulement de type Poiseuille. Le débit massique ($kg.s^{-1}$) est donc donné par la formule :

$$\underline{Q}_m = \rho \frac{b\|u\|^3}{12\mu} (\underline{\underline{1}} - \underline{n} \otimes \underline{n}) \cdot \underline{\underline{grad}}(p) \quad (3.3.30)$$

Avec b la longueur de la fissure, \underline{n} la normale à la fissure et $\|u\|$ la valeur de l'ouverture de fissure.

3.3.2.2 Passage aux éléments finis

Le débit massique ($kg.s^{-1}$) est noté dans le cadre des éléments finis :

$$\underline{q}_{aniso,m} = \rho \frac{b\|u\|^3}{12\mu} \cdot (\underline{\underline{B}} \cdot \underline{p}^e \cdot (\underline{\underline{1}} - \underline{n} \otimes \underline{n})) \quad (3.3.31)$$

Nous devons ensuite utiliser le débit massique surfacique. La surface va être prise comme étant égale à $S = b * h$. Avec b la largeur de la fissure et L la largeur de la fissure. Ne connaissant pas la forme exacte du plan de fissuration et celle ci étant de forme non rectangulaire la largeur de la fissure doit être approximée. Le choix a été fait d'utiliser la hauteur moyenne du tétraèdre pour approximer la largeur de la fissure. Les résultats obtenus avec cette approximation conforteront ce choix. La hauteur moyenne du tétraèdre est calculée avec la formule suivante :

$$h_m = \frac{\sqrt{6}}{3} * l_m \quad (3.3.32)$$

Avec l_m la longueur moyenne des arrêtes du tétraèdre.

Le débit surfacique s'écrit alors à l'aide de la surface décrite précédemment :

$$\underline{q}_{aniso,s} = \rho \frac{\|u\|^3}{12\mu h_m} \cdot ((\underline{\underline{1}} - \underline{n} \otimes \underline{n}) \cdot \underline{\underline{B}} \cdot \underline{p}^e) \quad (3.3.33)$$

La valeur de matrice $\underline{\underline{K}}_{ani}$ peut être déduite de ce débit surfacique. La perméabilité est non nulle dans les directions perpendiculaires au plan de la fissure et est définie par la formule suivante :

$$\underline{\underline{K}}_{ani} = \frac{\|u\|^3}{12\mu h_m} \cdot ((\underline{\underline{1}} - \underline{\underline{n}} \otimes \underline{\underline{n}}) \cdot \underline{\underline{B}} \cdot \underline{\underline{p}}^e) \quad (3.3.34)$$

Dans le cas d'un élément fissuré les forces internes sont calculées dans le cadre des éléments finis à l'aide la formule suivante :

$$\underline{\underline{f}}^{int} = -A_e \int_{\Omega} \underline{\underline{B}}^T \cdot \left(\underbrace{\frac{1}{\mu} \underline{\underline{k}}}_{K_{iso}} + \underbrace{\frac{\|u\|^3}{12\mu h_m} (\underline{\underline{1}} - \underline{\underline{n}} \otimes \underline{\underline{n}})}_{K_{ani}} \right) \underline{\underline{p}}^e \cdot \underline{\underline{B}} \, d\Omega \quad (3.3.35)$$

Pour un matériau sain (où $\|u\| = 0$) seule la perméabilité intrinsèque du matériau est non nulle. Lorsqu'une fissure apparaît, la valeur de la perméabilité anisotrope devient non nulle et pour des valeurs importantes de l'ouverture de fissure, la perméabilité intrinsèque devient négligeable.

3.3.2.3 Perméabilité d'un cube coupé par une fissure plane

Afin de vérifier la capacité de notre élément hydraulique à représenter des fissures, le cas d'un cube contenant une fissure plane est testé. Des conditions aux limites de type Dyrichlet sont imposées aux bords du domaine (gradient de pression δP) et la perméabilité macroscopique est ensuite calculée.

Le cube est composé de deux parties, une partie non fissurée de volume $V_{sain} = (L - [u]) * L^2$ et une partie fissurée de volume $V_{fiss} = [u] * L^2$ avec L la largeur du cube et $[u]$ la valeur de l'ouverture de fissure.

La perméabilité macroscopique peut être calculée analytiquement :

$$\begin{aligned} Q_M &= Q_{sain} & + & Q_{fiss} \\ Q_M &= -\frac{k_{iso}}{\mu} L(L - [u]) \Delta P & - & L [u] \left(\frac{[u]^3}{12\mu [u]} + \frac{k_{iso}}{\mu} \right) \Delta P \\ K_M &= k_{iso} & + & \frac{[u]^3}{12L} \end{aligned} \quad (3.3.36)$$

On observe que la perméabilité macroscopique dépend de l'ouverture de fissure et de la taille du domaine. Nous allons maintenant comparer les résultats obtenus numériquement aux résultats analytiques. Pour cela nous allons chercher dans le maillage tous les éléments qui sont coupés par le plan de fissure et nous appliquons ensuite la valeur d'ouverture de fissure $[u]$ et l'orientation de la fissure $\underline{\underline{n}}$ à ces éléments.

Sur la figure 3.7 on peut observer les éléments fissurés pour trois finesses de maillages différentes, la fissure est identique pour les trois maillages. Les éléments coupés par la fissure sont représentés en bleu. On note que dans le cas avec 300 nœuds, la fissure paraît grossière mais elle permet néanmoins de trouver de bons résultats.

Dans un premier temps nous comparons la perméabilité obtenue pour trois ouvertures de fissures différentes en fonction du nombre de nœuds pour vérifier si nous n'avons pas de

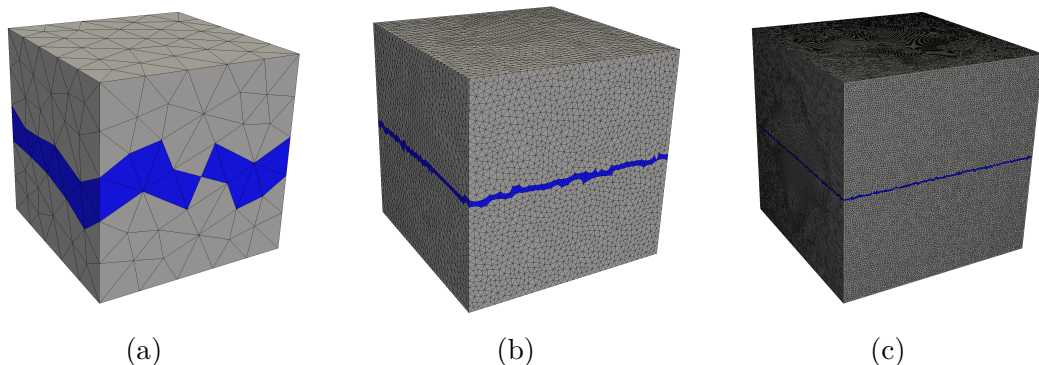


FIGURE 3.7 – Maillage avec une fissure plane (éléments fissurés en bleu) pour (a) 300 nœuds (b) 20000 nœuds et (c) 400000 nœuds.

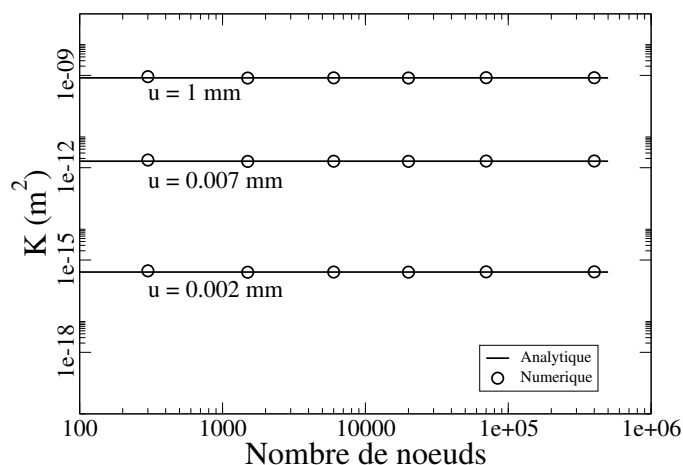


FIGURE 3.8 – Perméabilité macro en fonction du nombre de nœuds

dépendance au maillage. Le domaine sera ici une cube de 100 mm de côté. Ces résultats sont observables sur la figure 3.8.

On remarque pour les trois ouvertures de fissure que la perméabilité obtenue n'est pas influencée par le nombre de nœuds du maillage. Si l'on regarde plus précisément la valeur de la perméabilité pour une fissure de 1 mm sur la figure 3.9 on remarque une légère erreur commise, cependant ici le résultat qui nous intéresse le plus est l'ordre de grandeur obtenue (la perméabilité variant de plusieurs ordres de grandeurs). L'erreur semble venir de l'approximation faite pour le calcul de la hauteur du tétraèdre. Les résultats restent cependant concluants.

Dans un second temps nous allons vérifier que la perméabilité macroscopique obtenue en fonction de l'ouverture de fissure pour un maillage de 20000 nœuds et de 100 mm de côté correspond à celle attendue. Ces résultats sont donnés dans la figure 3.10.

La perméabilité macroscopique obtenue dans le cas numérique suit parfaitement celle obtenue analytiquement. Ceci nous conforte dans le choix fait pour la hauteur du tétraèdre (en utilisant la hauteur moyenne h_m).

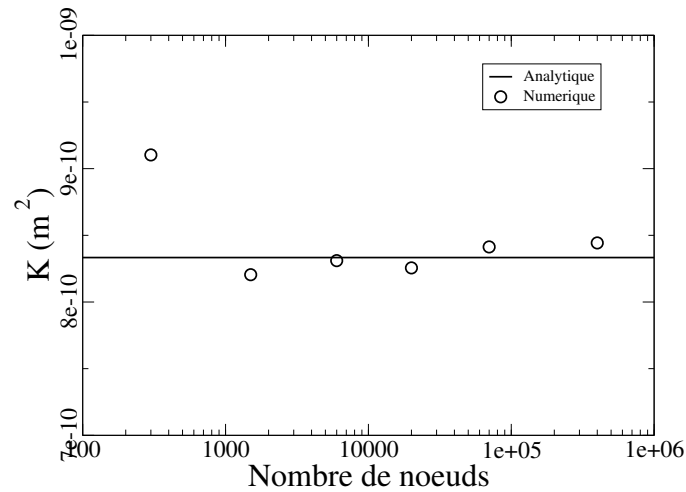


FIGURE 3.9 – Perméabilité macro en fonction du nombre de noeuds pour une fissure de 1 mm

Comme nous avons pu le voir précédemment, la perméabilité dépend de la taille du domaine, nous avons donc testé avec différentes tailles de domaine si cette propriété est vérifiée. Nous pouvons voir ces résultats dans la figure 3.11.

À nouveau les résultats observés correspondent aux valeurs analytiques. Le modèle développé permet ici de représenter précisément le comportement d'une fissure parfaite dans une cube.

3.3.2.4 Perméabilité d'un matériau hétérogène lors d'un essai de traction

Dans cette partie nous allons étudier l'influence du taux de chargement d'un cube hétérogène sur la perméabilité de celui ci. Ce test permet de voir le comportement du modèle pour une fissure due à un chargement mécanique, la modélisation de la fissure "réelle" étant plus complexe que la modélisation d'une fissure parfaite. De plus le matériau contient des inclusions et permet de tester l'ajout de discontinuité faible au modèle.

Le domaine étudié ici est un cube de 10 cm de côté. Ce cube est composé d'inclusions sphériques élastiques ayant un rayon variant entre 1 et 2 cm.

La matrice a un module Young de 32 GPa et un coefficient de Poisson de 0.3, elle a un comportement fragile avec une rupture en traction à 3.5 MPa (utilisation du modèle avec discontinuités fortes présenté dans le premier chapitre) et une perméabilité de $10^{-17}m^2$. Les inclusions ont un module Young de 100 GPa et un coefficient de 0.3, elles sont purement élastiques et ont une perméabilité de $10^{-21}m^2$.

Le maillage ainsi que les inclusions sont visibles sur la figure 3.13. Le maillage est composé de 20000 noeuds.

Le comportement mécanique obtenu est de type fragile. Bien que les inclusions soient grandes vis-à-vis de la taille du domaine et donc que le domaine ne soit pas un volume élémentaire représentatif le comportement observé dans les trois directions est proche. On

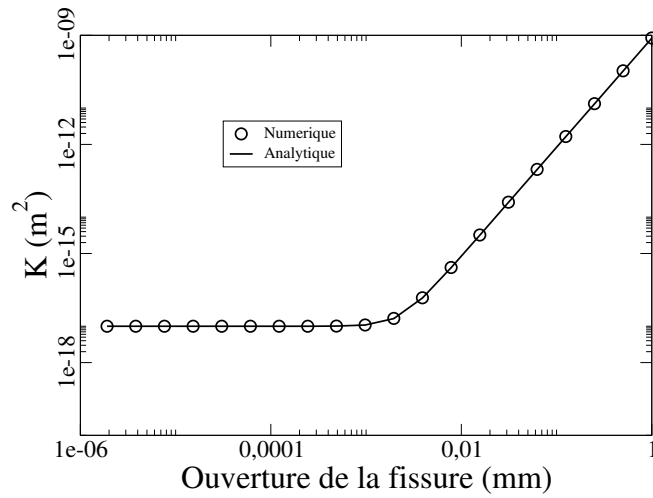


FIGURE 3.10 – Perméabilité macro en fonction de l’ouverture de fissure

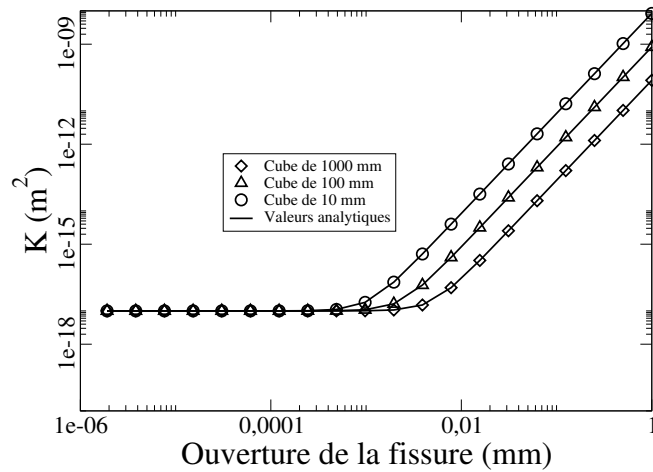


FIGURE 3.11 – Perméabilité macro en fonction de la taille du domaine

peut tout de même noter que suivant la direction X la contrainte redescend plus rapidement.

A cause de la présence de grosses inclusions le plan de fissure n’est pas parfaitement traversant et les contraintes ne descendent donc pas rapidement à 0. De plus l’énergie de fissuration définie dans le modèle de Rankine est non nulle, il y a donc une contrainte résiduelle.

L’évolution de la perméabilité en fonction de la déformation pour les trois directions de traction est visible sur la figure 3.14.

Dans un premier temps nous pouvons noter que suivant la direction du chargement la perméabilité semble ne pas être impactée, ce qui semble logique étant donné que l’ouverture de la fissure se fait dans la direction du chargement. La perméabilité due à l’ouverture

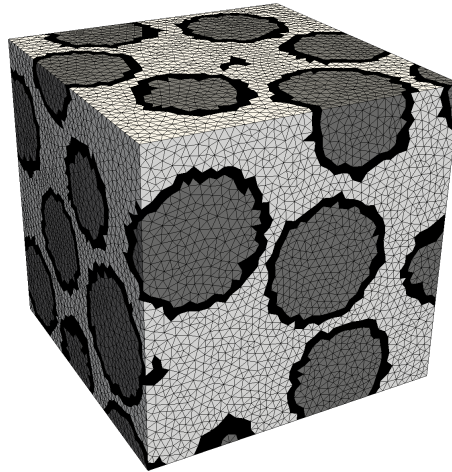


FIGURE 3.12 – Maillage utilisé pour le couplage mécanique \rightarrow hydrique

d'une fissure se faisant dans une direction perpendiculaire au plan de fissure, il est normal que cette direction ne soit pas impactée.

Dans un second temps on note que dans les directions perpendiculaires au chargement, la perméabilité augmente de plusieurs ordres de grandeur lorsque la fissuration apparaît (pour une déformation d'environ $4e - 5$), l'augmentation de la perméabilité correspond à l'apparition d'une microfissure.

On note que dans le cas de la traction suivant la direction x la perméabilité suivant la direction z met plus de temps à augmenter que suivant la direction y . Ceci s'explique par le fait que la fissure n'est pas totalement traversante suivant cette direction lors de l'apparition de la macro-fissure et qu'il est nécessaire d'avoir une déformation plus importante pour que la fissure soit traversante dans cette direction. On peut voir sur la figure 3.15 l'évolution de la macro-fissure pour une déformation de $4e - 5$ et $6e - 5$. Il apparaît que la fissure n'est pas traversante suivant la direction y à droite du cube pour un déformation de $4e - 5$ alors qu'elle le devient pour une déformation de $6e - 5$ ce qui explique le décalage de la courbe.

Pour le cas des tractions suivant y et z on observe pas ou peu (pour le cas suivant z où il y a un léger décalage) le phénomène. Ici seule la répartition spatiale des inclusions a impacté le trajet de fissure et donc la perméabilité. En effet les inclusions étant ici trop grandes par rapport à la taille du domaine pour que l'on puisse considérer celui ci comme un volume élémentaire représentatif.

La modélisation par discontinuité forte nous permet ici de modéliser une interaction entre le comportement mécanique et le comportement hydrique de manière complexe. En effet, le fait d'avoir accès aux ouvertures des fissures ainsi qu'à leur direction nous permet de modéliser efficacement l'impact de la fissuration sur la perméabilité d'un solide.

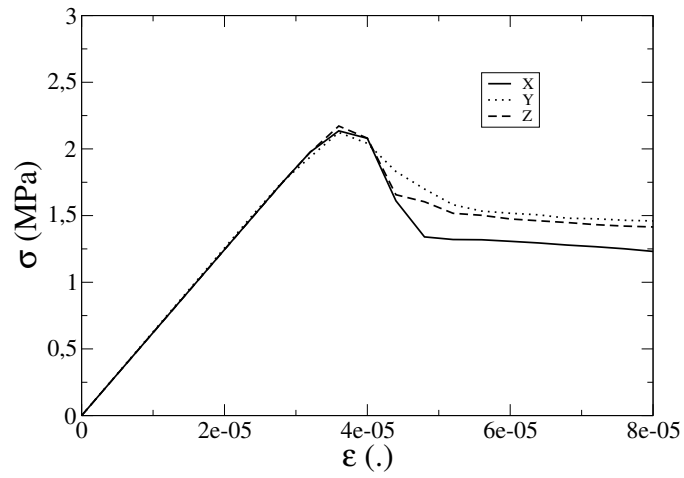


FIGURE 3.13 – Réponse mécanique obtenue dans les trois directions

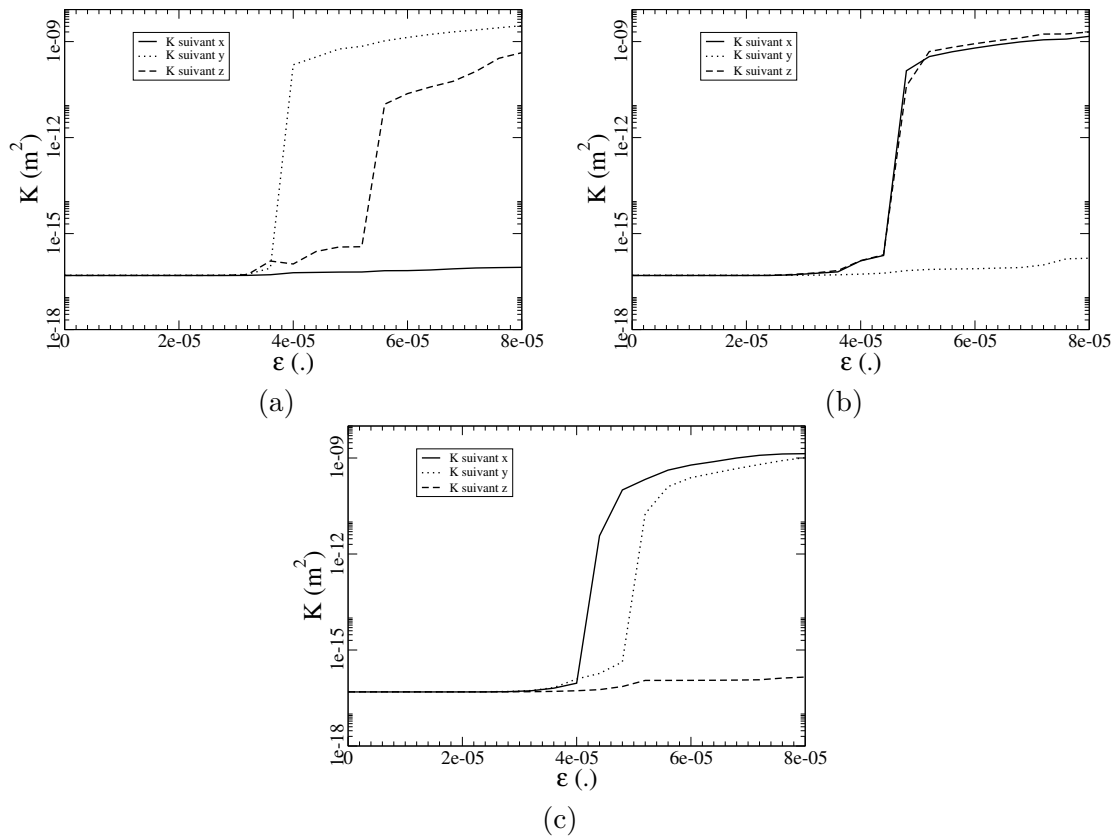


FIGURE 3.14 – Perméabilité en fonction de la déformation pour une traction suivant (a) x, (b) y et (c) z

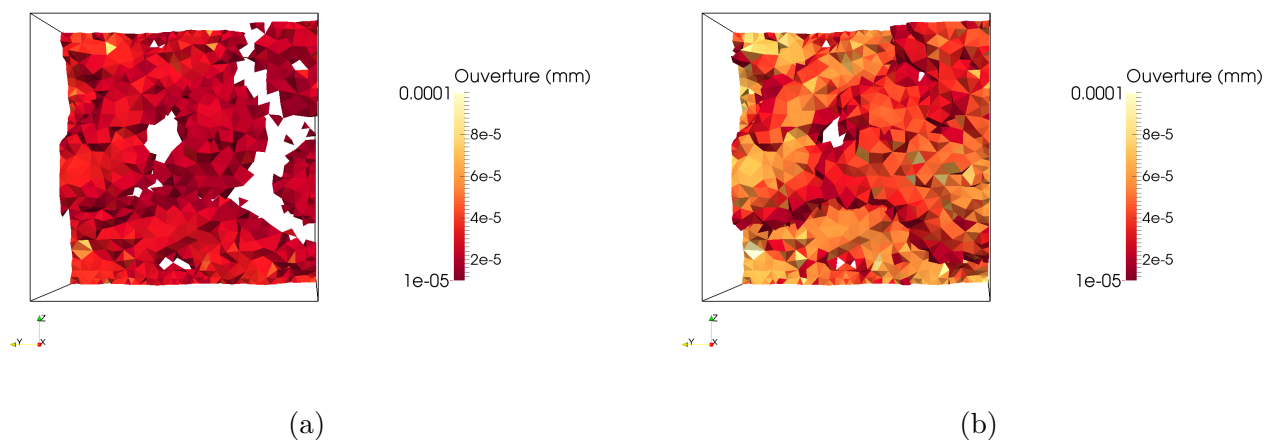


FIGURE 3.15 – Faciès de fissuration avec une déformation de (a) $4e-5$ (b) $6e-5$

3.4 Calcul de fissuration avec le couplage mécanique-hydraulique faible

Le composant mortier est enrichi d'un second composant coFeap (hydrique), cette intégration est visible sur la figure 3.16. Lorsqu'une itération du calcul est terminée, il est possible de récupérer les ouvertures et les orientations de chacune des fissures pour chacun des sous-domaines. Il est alors possible de calculer localement, dans tous les sous-domaines, la perméabilité induite par la fissuration.

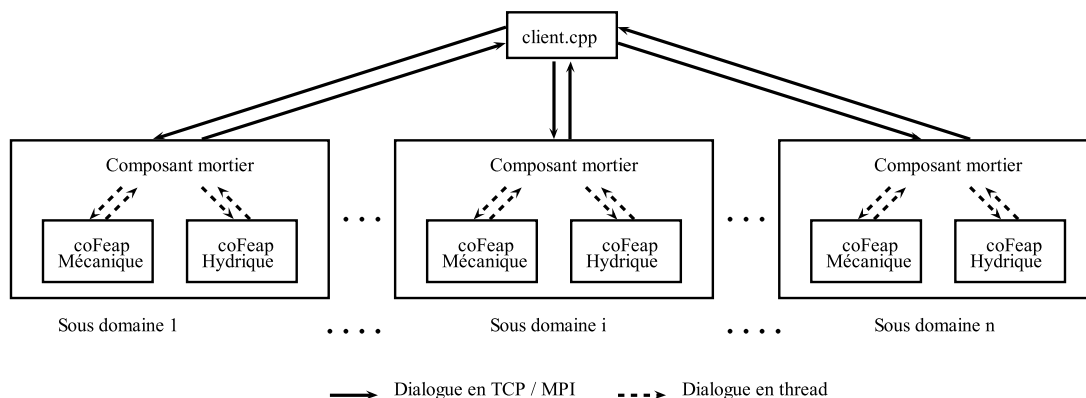


FIGURE 3.16 – Décomposition du programme en brique élémentaire avec présence d'un composant hydrique

Le composant mortier a pour rôle de récupérer les fissures du composant mécanique pour les envoyer comme données d'entrées au composant hydrique. En imposant un gradient de pression dans la direction dont on souhaite connaître la perméabilité et en mesurant le débit traversant le sous-domaine, il est possible de calculer la perméabilité macroscopique du sous-domaine.

Le composant mortier est enrichi de deux méthodes. Une méthode pour imposer l'état de fissuration, et une méthode pour calculer la perméabilité dans les trois directions principales.

```
CTL_Method(25,int,actionmatrice::set_crack,() ,0);  
CTL_Method(29,int,actionmatrice::get_permea,() ,0);
```

FIGURE 3.17 – Méthodes utiles au couplage faible

Le maillage utilisé pour chaque sous-domaine est le même pour la partie mécanique et la partie hydrique. Le nombre de degrés de libertés du calcul d'un sous-domaine hydrique est donc faible (inférieur à 10^4). Le calcul de la valeur de perméabilité est alors très rapide pour chaque sous-domaine. De plus, ces calculs étant effectués en parallèle, le temps de calcul dû à la perméabilité est négligeable devant le temps de calcul de la partie mécanique. Cette propriété est intéressante car elle permet de fournir des données importantes du calcul en ne demandant pas de ressources supplémentaire. Pour un calcul non linéaire de 12h avec 20 pas de chargements, le temps de calcul dû à la partie hydrique est inférieur à une minute.

L'exemple utilisé pour le calcul comporte 18% de quartz, 21% de calcite et 10% de kérogène. La génération des inclusions est visible sur la figure 3.18.

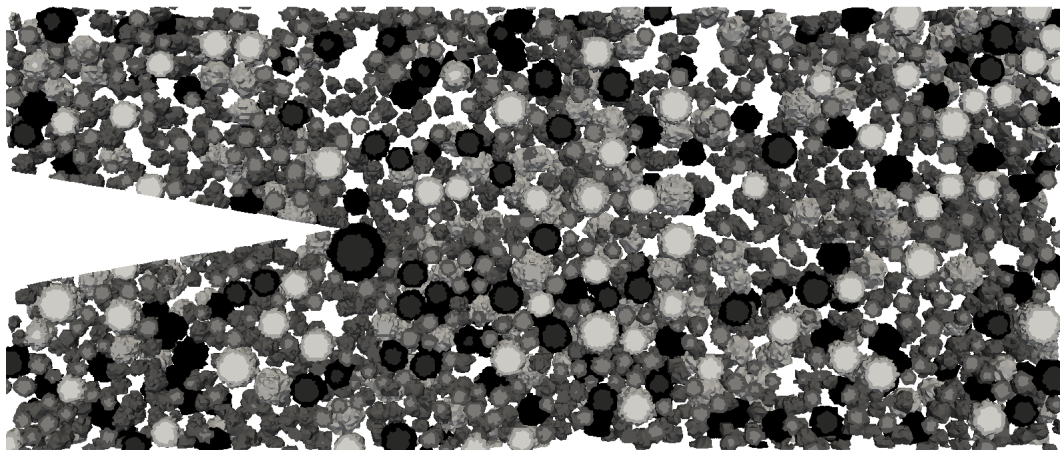


FIGURE 3.18 – Placement des inclusions pour une minéralogie de 18% de quartz, 21% de calcite et 10% de kérogène

Pour une pression de 0.6 MPa, soit la pression maximale imposée dans le calcul, le champ de déplacement suivant Y présente une forte discontinuité au niveau de l'entaille. Ce champ est visible sur la figure 3.19. Il est intéressant de noter que la discontinuité du

champ n'est pas perturbée par l'utilisation de la décomposition de domaine.

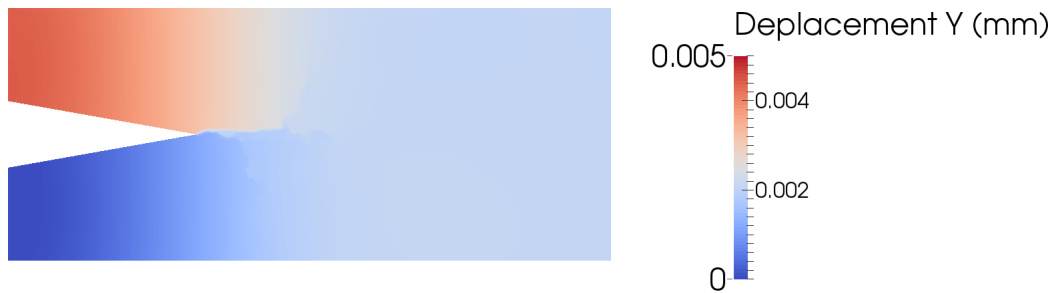


FIGURE 3.19 – Placement des inclusions pour une minéralogie de 18% de quartz, 21% de calcite et 10% de kérogène

En imposant une pression de 0.1 MPa la fissuration est quasi inexistante (seule une dizaine de tétraèdres ont fissuré). Le faciès de fissuration est visible sur la figure 3.20 ainsi que les valeurs de perméabilités obtenues. Comme indiqué la fissuration n'est pas visible sur la figure 3.20.a, les tétraèdres étant trop petits. Sur les trois figures suivantes la perméabilité de chaque sous-domaine est indiquée. On remarque une variabilité de la perméabilité entre les sous-domaines. Cette variabilité est due à la petite taille des sous-domaines par rapport aux inclusions. Les sous-domaines ne peuvent ainsi pas être considérés comme des sous-domaines et la variabilité spatiale des inclusions explique la variabilité de perméabilité observée. Un rapport de 40 est observé entre la plus grande et la plus faible perméabilité.

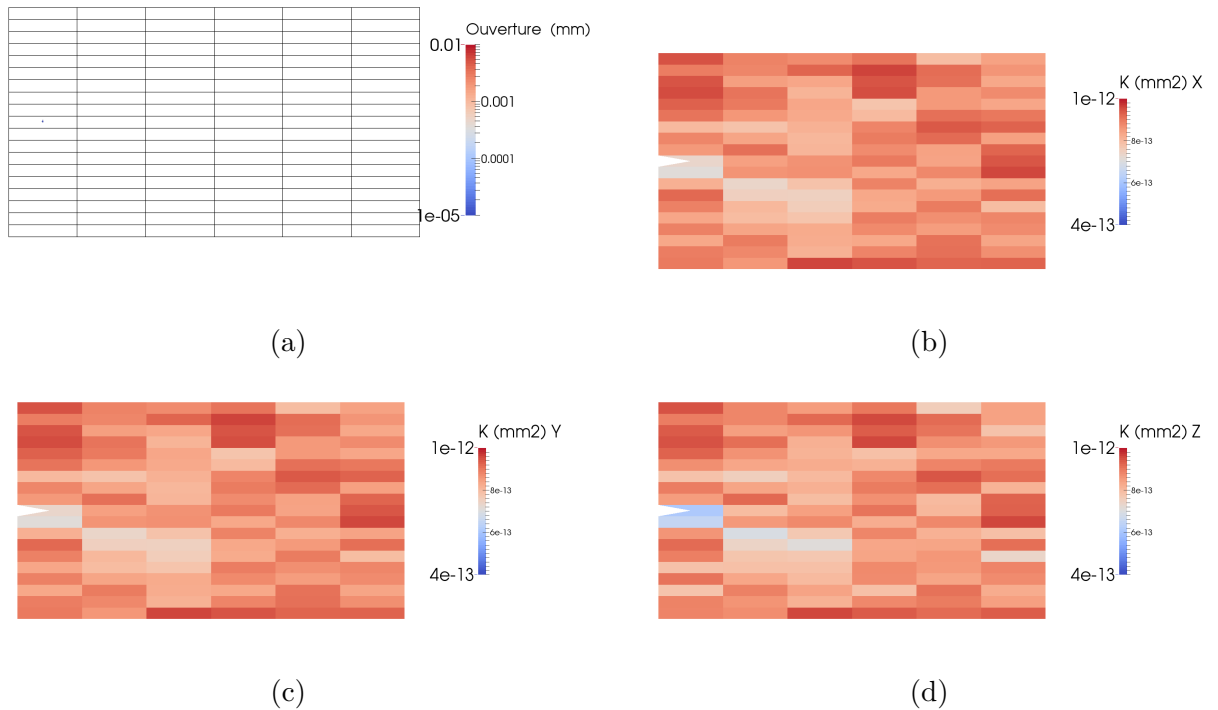


FIGURE 3.20 – Résultats obtenus pour 0.1 MPa (a) faciès de fissuration, (b) perméabilité suivante X, (c) perméabilité suivante Y et (d) perméabilité suivante Z

Lorsque la pression augmente et arrive à 0.5 MPa le nombre d'élément fissurés devient plus important. Comme pour le cas 0.1 MPa on peut observer la fissuration ainsi que la perméabilité sur la figure 3.21. La fissuration dépasse le sous-domaine de l'entaille et son trajet est dicté par la pression imposée ainsi que la présence des inclusions. Celles-ci ne pouvant pas fissurer la fissuration doit les contourner pour se propager. Ainsi il est intéressant de noter que la fissure n'est pas uniquement horizontale et peut avoir des parties verticales.

L'impact de la fissure sur la perméabilité est visible pour les sous-domaines concernés. Pour le sous-domaine en pointe d'entaille on note une augmentation plus importante suivant les directions X et Z. La fissure ayant globalement pour normal le plan Y il semble logique que la perméabilité suivant cette direction soit moins impactée. Cependant, à cause de la bifurcation due à la présence des inclusions, la perméabilité suivant Y est impactée (elle est augmentée d'un peu plus d'un ordre de grandeur contre deux voir trois suivant les deux autres directions).

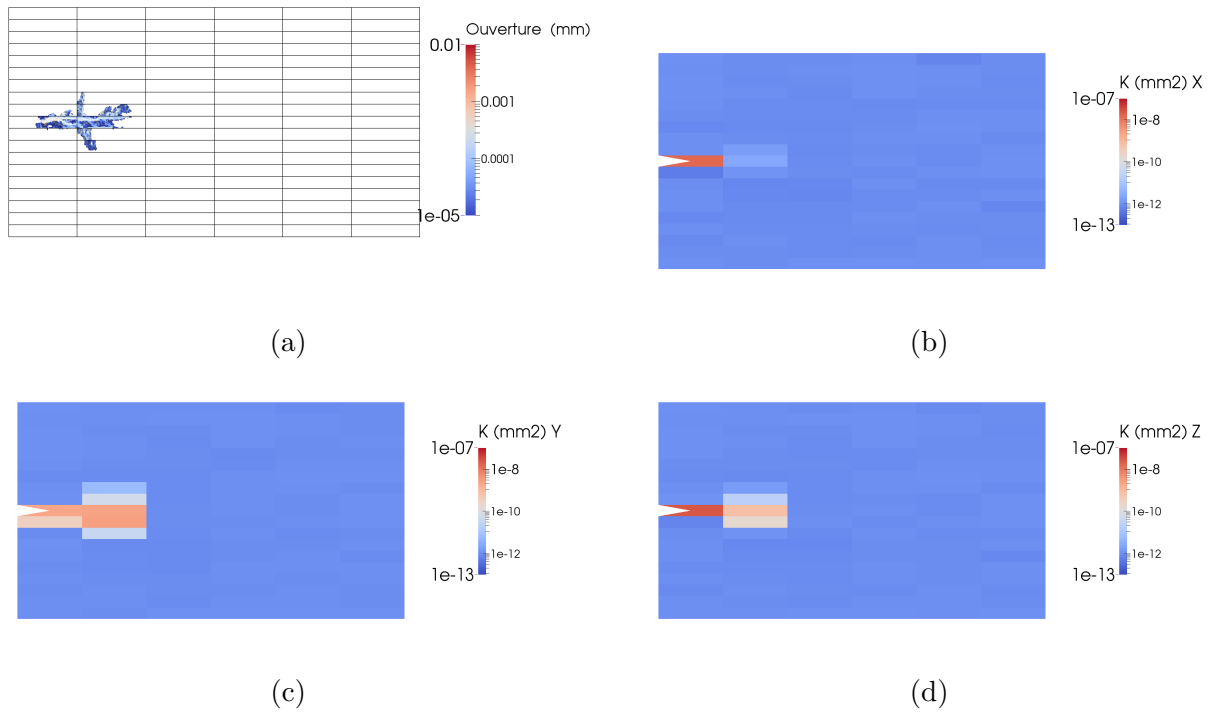


FIGURE 3.21 – Résultats obtenus pour 0.5 MPa (a) faciès de fissuration, (b) perméabilité suivante X, (c) perméabilité suivante Y et (d) perméabilité suivante Z

Une fois que la pression maximale imposée est atteinte la fissuration atteint sa taille maximale. À nouveau la figure 3.22 représente les différentes données nous intéressant. Par rapport au cas précédent la fissuration croit dans différentes directions. Le nombre de sous-domaines fissurés a augmenté. Ceci est nettement visible sur les images représentant la perméabilité. La valeur de celle ci a été augmentée de 9 ordres de grandeur. À ce niveau de perméabilité l'extraction des hydrocarbures est possible.

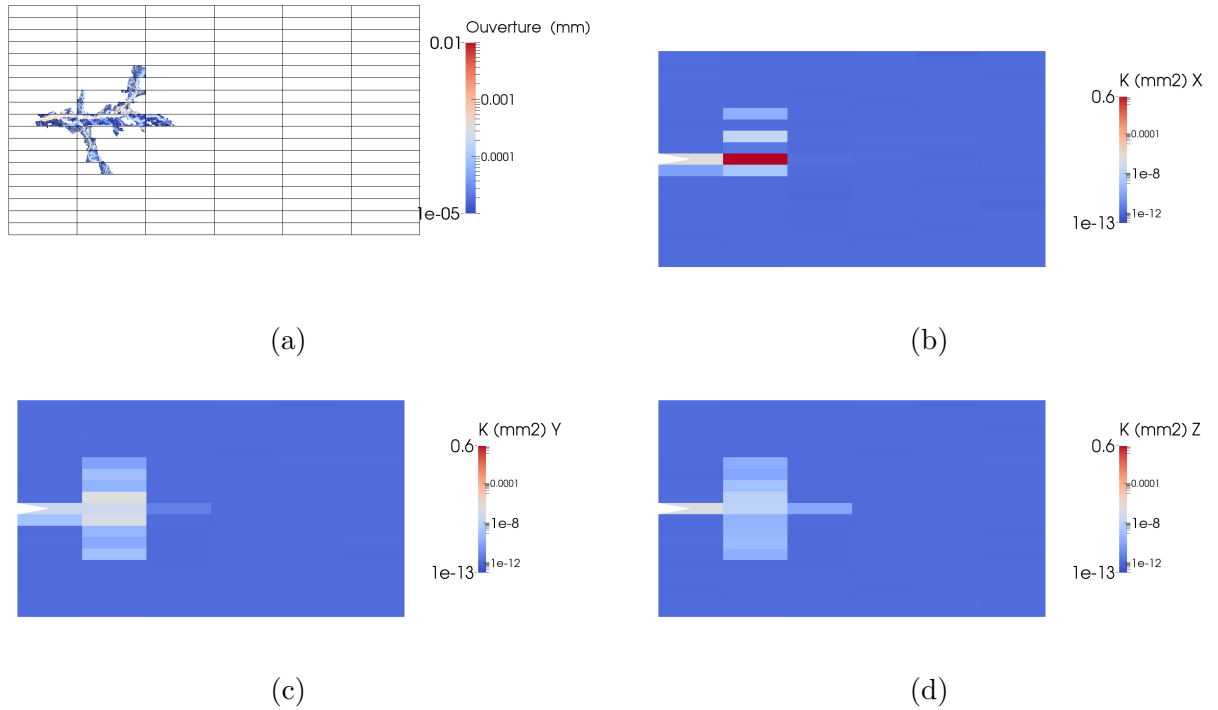


FIGURE 3.22 – Résultats obtenus pour 0.6 MPa (a) faciès de fissuration, (b) perméabilité suivante X, (c) perméabilité suivante Y et (d) perméabilité suivante Z

L'impact de la fissuration sur la perméabilité étant plus importante auprès de la pointe d'entaille il est intéressant d'étudier l'évolution de la perméabilité en fonction de la pression pour ces sous-domaines. Sur la figure 3.23 les trois sous-domaines pour lesquels nous allons nous intéresser sont indiqués.

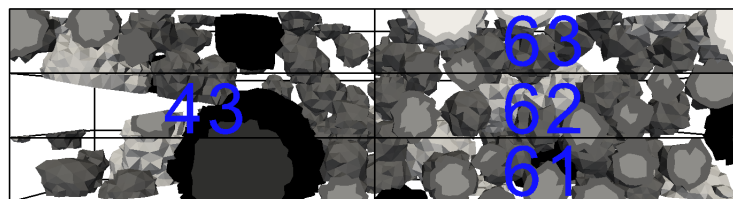


FIGURE 3.23 – Numérotation des sous-domaines en pointre d'entaille

Le sous domaine numéro 43 est celui qui contient la pointe d'entaille. La perméabilité, visible sur la figure 3.24.a, augmente dans un premier temps plus rapidement suivant la direction Z (normale à la plaque entaillée). La perméabilité suivant X augmente ensuite ainsi que celle suivant Y dans une moindre mesure. Avec un calcul sans inclusions, la perméabilité suivant Y serait nulle et serait similaire suivant les deux autres directions.

Comme expliqué précédemment, la présence d'inclusions a un impact non négligeable sur la fissuration et donc la perméabilité induite.

Le faciès de fissuration, visible sur la figure 3.24.b englobe la quasi totalité des tétraèdres à droite de la pointe d'entaille. La présence d'un grand nombre de fissure et la bifurcation due aux inclusions expliquent bien l'augmentation de la perméabilité observée.

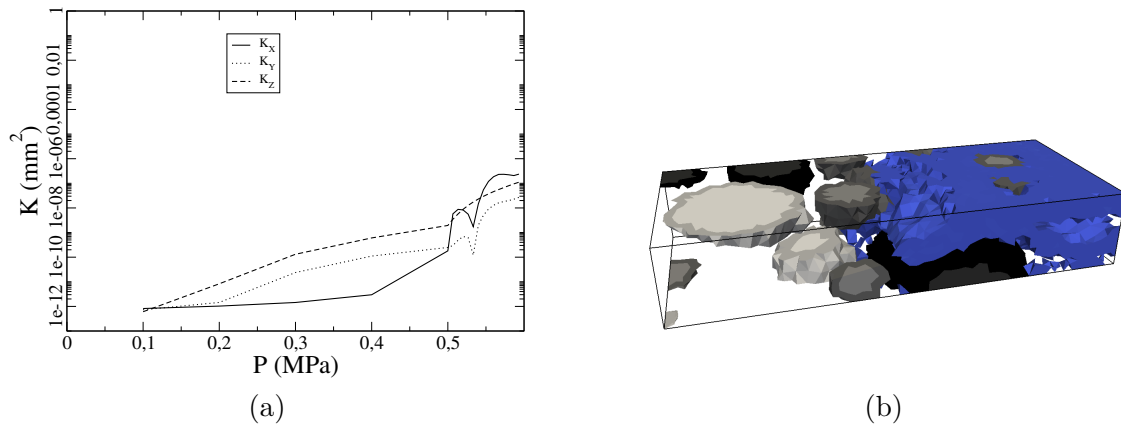


FIGURE 3.24 – (a) Perméabilité et (b) faciès de fissuration du sous-domaine 43

En ce qui concerne le sous-domaine 62 (situé à droite du sous-domaine précédent), la perméabilité suivant X augmente beaucoup plus rapidement (voir figure 3.25.a) En regardant le faciès de fissuration de la figure 3.25, on note la présence d'une macro-fissure ayant pour normale Y. La faible augmentation de la perméabilité suivant Z peut s'expliquer par la présence d'inclusions bloquant le transfert de masse.

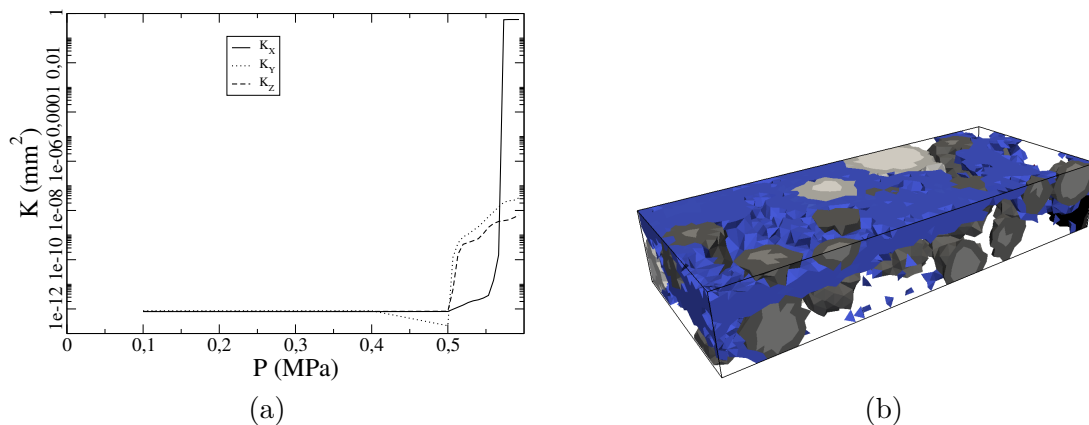
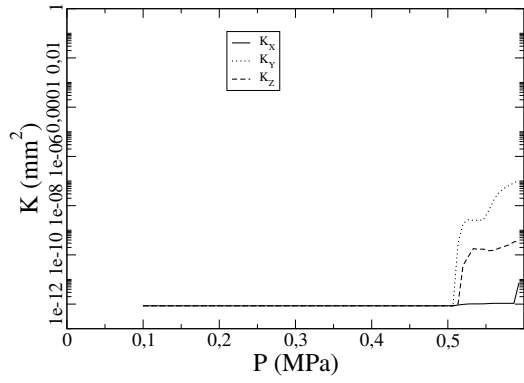
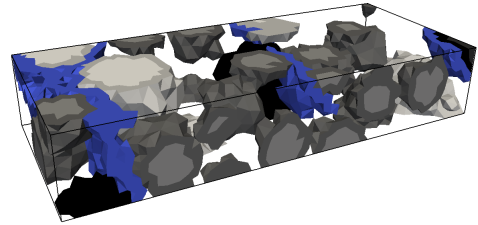


FIGURE 3.25 – (a) Perméabilité et (b) faciès de fissuration du sous-domaine 62

Les sous-domaines 61 et 63 ont le même comportement. Leur perméabilité suivant l'axe Y (visible sur les figures 3.26.a et 3.27.a) augmente plus rapidement et celle suivant X reste presque inchangée. En regardant les deux faciès de fissuration (figure 3.26.b et 3.27.b) on remarque que les fissures ont toutes pour normale X. L'évolution de la perméabilité obtenue semble correspondre au faciès de fissuration.

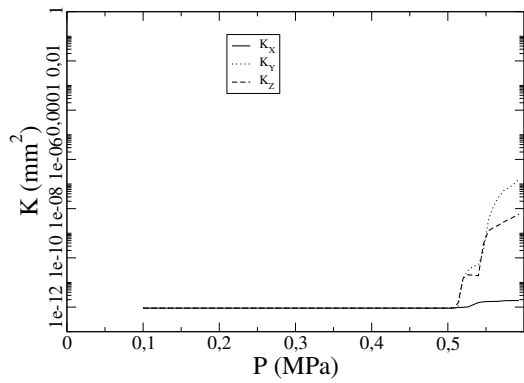


(a)

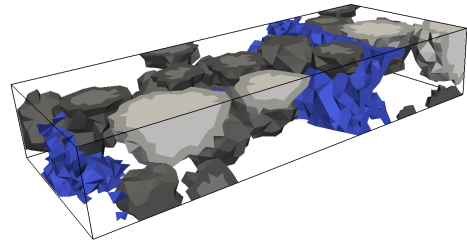


(b)

FIGURE 3.26 – (a) Perméabilité et (b) faciès de fissuration du sous-domaine 61



(a)



(b)

FIGURE 3.27 – (a) Perméabilité et (b) faciès de fissuration du sous-domaine 63

3.5 Conclusion

Le modèle de décomposition de domaine a permis de modéliser l'impact de la minéralogie sur la ténacité du matériau. Les résultats obtenus lors de cette modélisation sont du même ordre de grandeur que ceux obtenus expérimentalement. Grâce au code développé, le temps de calcul nécessaire pour la réalisation d'un essai de fissuration numérique avec plus de 3.10^6 degrés de libertés est descendu sous les 14 heures contre plus de 200 heures pour un calcul sans décomposition de domaine.

Le système de couplage présenté ici est une première approche. Un couplage fort entre la mécanique et l'hydraulique aurait dû être mis en place. Un problème de convergence de notre code de calcul pour la partie hydraulique lorsque la perméabilité d'un tétraèdre était très différente d'un tétraèdre voisin au niveau de l'interface a été mis en évidence. Bien que tout soit mis en place pour le couplage complet, ce problème de convergence empêche la réalisation d'un tel couplage.

Une version améliorée de l'algorithme ou une système de résolution du problème de décomposition de domaine devront être mis en place afin de régler ce problème et ainsi pouvoir effectuer un couplage fort.

Le modèle hydraulique développé dans cette partie permet de prendre en compte des éléments bi-biphasiques et donc d'utiliser des maillages non adaptés. Une loi de type Poiseuille a été ajoutée afin de prendre en compte la présence de fissure au sein d'un élément lors du calcul hydraulique. Ce modèle a été testé avec une fissure parfaite et semble bien se comporter.

Un composant hydraulique a été incorporé au composant mortier développé dans la partie précédente. Ce composant permet, à chaque itération du calcul mécanique, de récupérer les fissures et de calculer la perméabilité des différents sous-domaines. À l'aide de ce composant il a été possible de réaliser une carte de la perméabilité à chaque itération d'un calcul de fissuration. Ces résultats montrent les capacités d'un tel modèle et nous encourageant à poursuivre nos efforts dans la réalisation d'un couplage hydro-mécanique fort.

Conclusion générale

La modélisation numérique avec la méthode des éléments finis est la base des travaux présentés ici. Le travail a consisté dans un premier temps à développer une méthode élément fini permettant de modéliser la fissuration. Les méthodes à discontinuité, et plus particulièrement la méthode E-FEM, ont été choisies dans le cadre de cette thèse. Ces méthodes consistent à modéliser la fissuration à l'aide d'un enrichissement cinématique. Plus précisément c'est l'ajout de discontinuité dans le champ de déplacement (appelé discontinuité forte) qui permet de modéliser le phénomène de fissuration. L'ajout d'un second type d'enrichissement, les discontinuités dans le champ de déformation, permet de prendre en compte des éléments bi-phasiques. Cette seconde discontinuité est intéressante car elle permet d'utiliser des maillages non adaptés. Plus précisément, pour modéliser des inclusions il n'est pas nécessaire de mailler celles-ci. Un même maillage peut être utilisé pour les différentes minéralogies, il suffit alors de projeter les inclusions sur le maillage. La génération de maillage adapté demande beaucoup de temps de calcul alors que la projection d'inclusion sur un maillage est très rapide. L'utilisation de maillage non adapté semble indispensable à l'étude d'un grand nombre de morphologies.

Les méthodes d'enrichissements cinématiques ont l'avantage de représenter véritablement la fissure. En comparaison aux méthodes à fissurations réparties où il n'y a pas d'informations précises sur la fissuration. Les méthodes d'enrichissement fournissent des informations sur la fissure comme l'ouverture de celle-ci ainsi que son orientation. Cet avantage permet de coupler simplement un calcul hydraulique à un calcul mécanique. En effet, en modélisant le débit à travers la fissure à l'aide d'une loi de Poiseuille à l'aide des informations d'ouverture et d'orientation, il est possible de calculer la perméabilité anisotrope due à la fissure et ainsi de calculer une perméabilité macroscopique due à une fissuration issue d'un calcul mécanique.

Pour réaliser les calculs numériques plus rapidement, une partie importante du travail a été de développer un code de calcul utilisant la décomposition de domaine. Cette méthode permet, en découpant un domaine en sous-domaines, de réaliser une partie du calcul en parallèle et ainsi d'améliorer le temps nécessaire à la résolution du problème. Au sein des méthodes de décomposition de domaine, nous avons décidé d'utiliser la méthode de mortier. Un des avantages de cette méthode est de pouvoir réaliser des calculs avec des maillages non conformes et ainsi de pouvoir raffiner localement le maillage si besoin.

Le code de calcul a été réalisé à l'aide de la programmation orientée composant et plus particulièrement à l'aide du middleware CTL. Le principe de la programmation orientée

composant est de créer des composants, qui peuvent être vus comme une brique élémentaire. Il est possible d'avoir plusieurs de ces briques et celles ci sont indépendantes. Ces briques disposent de méthodes qui peuvent être lancées indépendamment. Une brique élémentaire, appelée coFeap, contenant le logiciel éléments finis Feap était déjà existante avant les travaux présentés ici. Le travail a consisté à créer un composant, nommé mortier, qui encapsule le composant coFeap et qui dispose de méthode propre à la décomposition de domaine (création des matrices d'interpolation par exemple). Un algorithme a ensuite été développé pour résoudre le problème de décomposition de domaine en linéaire et en non linéaire. Finalement, le code de calcul créé combine l'algorithme ainsi que le composant mortier et a pour finalité la résolution de problème éléments finis à l'aide de la décomposition de mortier.

L'outil développé a été utilisé pour la simulation numérique d'un essai de fissuration. Un premier test a été réalisé sur un exemple simple d'une poutre en traction pour vérifier que le comportement mécanique n'était pas modifié par l'utilisation de la méthode de décomposition de domaine. Deux calculs ont pour cela été menés, un en utilisant un seul domaine et une résolution éléments finis classique et un second en décomposant le domaine en six sous-domaines dont deux plus raffinés. Cet exemple a mis en avant un comportement mécanique similaire, validant ainsi notre choix d'utiliser la méthode de mortier. Des essais numériques de fissuration ont été réalisés avec différentes minéralogies sur un maillage commun. Ces calculs ont mis en avant la capacité de notre modèle à obtenir des paramètres macroscopiques, utiles pour des calculs à une plus grande échelle, en fonction de la minéralogie du matériau. Un des points clés de notre approche est le faible temps de calcul demandé. Pour un maillage de plus d'un million de degrés de liberté, le temps de calcul est resté inférieur à 12h. De tels calculs sont donc réalisables en une nuit ce qui était à la fois un impératif issu du serveur que nous utilisons ainsi qu'un avantage considérable pour une utilisation comme outil d'ingénierie dans une entreprise. Sans décomposition de domaine de tel calcul aurait demandé environ une semaine.

La méthode E-FEM ayant pour avantage l'accès aux valeurs d'ouvertures ainsi que l'orientation de celle ci un modèle hydraulique a été développé. Ce modèle permet, à l'aide d'une loi de Poiseuille, de prendre en compte l'impact d'une fissure sur la perméabilité du matériau. Pour utiliser les mêmes maillages que pour la partie mécanique, c'est à dire des maillages non adaptés, la méthode des enrichissements cinématique a de nouveau été utilisée. Seule une discontinuité dans le champ de déformation a été ajoutée pour avoir des éléments tétraédriques biphasiques.

Un couplage hydro-mécanique total n'a pas pu être mise en place en utilisant la méthode de décomposition de domaine. En effet un problème de convergence, apparaissant lorsque la différence de perméabilité entre deux tétraèdres est trop importante (supérieure à 10^3). Une des perspectives est d'améliorer l'algorithme de résolution dans le but de ne plus avoir ce problème de convergence. Le couplage entre la partie hydraulique et la partie mécanique pourra alors se faire en prenant en compte la pression la pression hydraulique lors du calcul de la contrainte élémentaire. La contrainte appliquée à chaque tétraèdre du maillage, qui est noté $\underline{\underline{\sigma_t}}$ serait composée de la somme de la contrainte effective, notée $\underline{\underline{\sigma_{eff}}}$, et de la contrainte due à la pression de fluide, notée $\underline{\underline{\sigma_f}}$. Ce couplage à déjà été

intégré à notre code de calcul, un bon comportement de celui ci a pu être observé. Le couplage complet entre la partie mécanique et la partie hydraulique pourra être réalisé rapidement une fois le problème de convergence résolu, tous les éléments étant déjà en place.

Annexe A

Minéralogies étudiées

Name	% of Calcite	% of Quartz	% of Pyrite	% of all inclusions
C14_Q11_P1	14	11	1	26
C27_Q0_P0	27	0	0	27
C28_Q6_P1	28	6	1	35
C28_Q19_P1	28	19	1	48
C38_Q6_P1	38	6	1	45
C45_Q0_P1	45	0	1	46
C37_Q0_P0	37	0	0	37
C0_Q30_P1	0	30	1	31
C12_Q8_P1	12	8	1	21
C9_Q23_P1	9	23	1	33
C7_Q8_P1	7	8	1	16
C19_Q17_P1	19	17	1	37
C14_Q16_P1	14	16	1	31
C37_Q3_P1	37	3	1	41
C7_Q30_P1	7	30	1	38
C16_Q13_P1	16	13	1	30
C13_Q11_P0	13	11	0	24
C25_Q11_P1	25	11	1	37
C9_Q12_P1	9	12	1	22
C14_Q11_P3	14	11	3	28
C13_Q11_P2	13	11	2	26
C20_Q15_P1	20	15	1	36
C33_Q0_P0	33	0	0	33
C15_Q19_P0	15	19	0	34
C15_Q6_P0	15	6	0	21
C15_Q7_P0	15	7	0	22
C16_Q13_P0	16	13	0	29
C23_Q7_P0	23	7	0	30
C9_Q7_P0	9	7	0	16
C9_Q7_P0	9	7	0	16

Name	% of Calcite	% of Quartz	% of Pyrite	% of all inclusions
C23_Q7_P0	23	7	0	30
C21_Q7_P0	21	7	0	28
C16_Q8_P0	16	8	0	24
C10_Q7_P0	10	7	0	17
C16_Q7_P0	16	7	0	23
C22_Q0_P0	22	0	0	22
C26_Q0_P0	26	0	0	26
C8_Q0_P0	8	0	0	8
C16_Q0_P0	16	0	0	16
C8_Q0_P0	8	0	0	8
C16_Q0_P0	16	0	0	16
C23_Q0_P0	23	0	0	23
C16_Q0_P0	16	0	0	16
C11_Q0_P0	11	0	0	11
C7_Q0_P0	7	0	0	11
C6_Q0_P0	6	0	0	11
C5_Q0_P0	5	0	0	11
C3_Q0_P0	3	0	0	11
C1_Q0_P0	1	0	0	11
C13_Q10_P0	13	10	0	23
C11_Q10_P0	11	10	0	21
C4_Q18_P0	4	18	0	22
C5_Q17_P0	5	17	0	22
C10_Q13_P0	10	13	0	23
C13_Q9_P0	13	9	0	22
C4_Q17_P0	4	17	0	21
C0_Q7_P0	0	7	0	7
C0_Q21_P0	0	21	0	21
C0_Q8_P0	0	8	0	8
C0_Q13_P0	0	13	0	13
C0_Q20_P0	0	20	0	20
C0_Q6_P0	0	6	0	6
C0_Q15_P0	0	15	0	15
C0_Q21_P0	0	21	0	21
C8_Q14_P0	8	14	0	22
C7_Q15_P0	7	15	0	22
C8_Q6_P0	8	6	0	14
C7_Q20_P0	7	20	0	27
C7_Q21_P0	7	21	0	28
C7_Q20_P0	7	20	0	27
C8_Q7_P0	8	7	0	15
C0_Q0_P0	0	0	0	0

TABLE A.1 – Distribution volume fractions

Annexe B

Composant mortier

```
#ifndef __actionmatriceCI
#define __actionmatriceCI
#include <vector>
#include <string>
#include <simu.ci>
#include "action_matrice.ci"

class actionmatrice {

public :
    actionmatrice();
    /*! Constructor that instantiate a simu as an attribute of the class.
    actionmatrice(const ctl::string& path , const ctl::string& path_C,
                  const ctl::string& path_P ) ;
    /*! Destructor
    ~actionmatrice();
    void get_domain(std::vector<double>& domain, std::vector<int>& facex0,
                   std::vector<int>& facex1, std::vector<int>& facey0,
                   std::vector<int>& facey1);
    void get_ms_data(std::vector<std::vector<double> > geo,
                    std::vector<std::vector<int> > facex0,
                    std::vector<std::vector<int> > facex1,
                    std::vector<std::vector<int> > facey0,
                    std::vector<std::vector<int> > facey1,
                    std::vector<std::vector<int> >& master_slave,
                    std::vector<double>& value_ms,
                    std::vector<std::vector<std::vector<int> > >& node_ms );
```

FIGURE B.1 – Fichier composant_mortier.hpp

```

void compute_B1(std::vector<int> node_ms, std::vector<double> domain,
               std::vector<std::vector<int> > tria_1,
               std::vector<std::vector<double> > coor_1);
void compute_B2(std::vector<int> node_ms, std::vector<double> domain,
               std::vector<std::vector<int> > tria_1,
               std::vector<std::vector<double> > coor_1,
               std::vector<std::vector<int> > tria_2,
               std::vector<std::vector<double> > coor_2);
void get_nodes_connectivity(std::vector<int>& connect, std::vector<double>& nodes);
int compute_B_action();
int compute_B_action_p();
int compute_B_action2(const double alpha);
int compute_B_action2_p(const double alpha);
int compute_B_action3();
int compute_B_action3_p();
int compute_Bt_action();
int compute_Bt_action_p();
void set_mortar(ActionmatriceCI& component, const int indice1, const int indice2);
void set_mortar_p(ActionmatriceCI& component, const int indice1, const int indice2);
void set_state(const std::string name, std::vector<double>& mortar);
void set_state2(const std::string name);
void get_state(const std::string name, std::vector<double>& mortar, const int indice);
void get_Floc(std::vector<double>& F_loc);

void update(const std::string name, const double alpha);
void save(const std::string name, const int iter, const int nolin, const int time);
void load(const std::string name);
void plot(const std::string name);
int solve();
int solve_p();
double get_residual();
double get_norme();
double get_norme_p();
double lanorme;

```

FIGURE B.1 – Fichier composant_mortier.hpp

```

private :

SimuCI<double> _cofeap;
SimuCI<double> _cofeap_p;
int count_solve;
int count_solve_p;
std::vector<double> RBM;
std::vector<double> U;
std::vector<double> U_p;
std::vector<double> load_ini;
std::vector<double> load_ini_p;
std::vector<double> load_ini_2;
std::vector<double> U_nolin;
std::vector<double> dU;
std::vector<double> dU_p;
std::vector<double> URBM;
std::vector<double> URBM_p;
std::vector<double> dURBM;
std::vector<double> dURBM_p;
std::vector<double> lambda;
std::vector<double> lambda_p;
std::vector<double> dlambd;
std::vector<double> dlambd_p;
std::vector<double> temp_dURBM;
std::vector<double> temp_dURBM_p;
std::vector<std::vector<double> > R;
std::vector<std::vector<double> > temp_mortar;
std::vector<std::vector<double> > temp_lambda;
int dim ;

```

FIGURE B.1 – Fichier composant_mortier.hpp

```

std::vector<std::vector<int> > BI;
std::vector<std::vector<int> > BJ;
std::vector<std::vector<double> > BV;
std::vector<std::vector<int> > node_comp;
std::vector<std::vector<double> > mortar_function;
std::vector<std::vector<double> > mortar_function_p;
std::vector<std::vector<double> > transp_function;
std::vector<std::vector<double> > transp_function_p;
inline void matrice_vecteur(std::vector<std::vector<double> > matrice,
                           std::vector<double> in, std::vector<double>& out);
inline void matrice_vecteur_t(std::vector<std::vector<double> > matrice,
                              std::vector<double> in, std::vector<double>& out);
inline void IJV_vecteur(std::vector<int> I, std::vector<int> J,
                       std::vector<double> V, std::vector<double> in,
                       std::vector<double>& out);
inline void IJV_vecteur_t(std::vector<int> I, std::vector<int> J,
                          std::vector<double> V, std::vector<double> in,
                          std::vector<double>& out);
inline double norme(std::vector<double> Vec);
inline double norme(std::vector<std::vector<double> > Mat);
};

#endif

```

FIGURE B.1 – Fichier composant_mortier.hpp

```

#ifndef __ActionmatriceCI
#define __ActionmatriceCI
#include <ctl.h>
#include <simu.ci>

#define CTL_Class ActionmatriceCI
#include CTL_ClassBegin
#define CTL_Constructor1( const string, const string, const string ), 3
#define CTL_Method1 void, get_domain,(array<real8>,array<int4>,array<int4>,
array<int4>,array<int4> ), 5
#define CTL_Method2 void, get_ms_data,(array<array<real8> >,array<array<int4> >,
array<array<int4> >,array<array<int4> >,
array<array<int4> >,array<array<int4> >,
array<real8>,array<array<array<int4> > > ),8
#define CTL_Method3 void, compute_B1,(array<int4>,array<real8>,array<array<int4> >,
array<array<real8> > ), 4
#define CTL_Method4 void, compute_B2,(array<int4>,array<real8>,array<array<int4> >,
array<array<real8> >,array<array<int4> >,
array<array<real8> > ), 6

#define CTL_Method5 void, get_nodes_connectivity,(array<int4>,array<real8>),2
#define CTL_Method6 int4, compute_B_action,() ,0
#define CTL_Method7 int4, compute_B_action2,(const real8) ,1
#define CTL_Method8 int4, compute_B_action3,() ,0
#define CTL_Method9 int4, compute_Bt_action,() ,0
#define CTL_Method10 void, set_mortar,(ActionmatriceCI , const int4,const int4) ,3
#define CTL_Method11 void, set_state,(const string,array<real8>) ,2
#define CTL_Method12 void, get_state,(const string,array<real8>,const int4) ,3
#define CTL_Method13 void, get_Floc,(array<real8>),1
#define CTL_Method14 void, update,(const string,const real8),2
#define CTL_Method15 void, plot,(const string),1
#define CTL_Method16 int4, solve,(),0
#define CTL_Method17 real8, get_norme,(),0
#define CTL_Method18 void, set_state2,(const string) ,1
#define CTL_Method19 void, save,(const string,const int4,const int4,const int4),4
#define CTL_Method20 void, load,(const string),1
#define CTL_Method21 real8, get_residual,(),0
#define CTL_Method22 int4, compute_B_action_p,() ,0
#define CTL_Method23 int4, compute_Bt_action_p,() ,0
#define CTL_Method24 int4, solve_p,(),0
#define CTL_Method25 int4, compute_B_action3_p,() ,0
#define CTL_Method26 void, set_mortar_p,(ActionmatriceCI , const int4,const int4) ,3
#define CTL_Method27 real8, get_norme_p,(),0
#define CTL_Method28 int4, compute_B_action2_p,(const real8) ,1
#include CTL_ClassEnd
#endif //__ActionmatriceCI

```

FIGURE B.2 – Fichier composant_mortier.ci

Bibliographie

- [Adachi and Detournay,] Adachi, J. and Detournay, E. Plane strain propagation of a hydraulic fracture in a permeable rock. *Engineering Fracture Mechanics, Volume = 75, issue = 16, Pages = 4666-4694, Year = 2008.*
- [Bai, 2009] Bai, Y. and Wierzbicki, T. (2009). Application of extended mohr-coulomb criterion to ductile fracture. *International Journal of Fracture*, 161 :1–20.
- [Barber and Huhdanpaa, 1996] Barber, C.B., D. D. and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *Mathematical Software*, 22 :469–483.
- [Belgacem, 1999] Belgacem, F. B. (1999). The mortar finite element method with lagrange multipliers. *Numerische Mathematik*, 84 :173–197.
- [Benkemoun, 2009] Benkemoun, N. (2009). *Contribution aux approches multi-échelles séquentielles pour la modélisation numérique des matériaux à matrice cimentaire.* PhD thesis, UTC.
- [Bernardi et al., 1990] Bernardi, C., Maday, Y., and Patera, A. T. (1990). A new nonconforming approach to domain decomposition : the mortar element method. In *Collège de France Seminar, volume XI.*
- [Bernardi et al., 2005] Bernardi, C., Maday, Y., and Rapetti, F. (2005). Basics and some element method applications of the mortar.
- [Bezrukov et al., 2002] Bezrukov, A., Bargiel, M., and Stoyan, D. (2002). Statistical analysis of simulated random packings of spheres. *Particle Particle Systems Characterization*, 19 :111–118.
- [Bousige et al., 2016] Bousige, C., Ghimbeu, C. M., Vix-Guterl, C., Pomerantz, A. E., Suleimenova, A., Vaughan, G., Garbarino, G., Feygenson, M., Wildgruber, C., Ulm, F.-J., Pellenq, R. J. M., and Coasne, B. (2016). Realistic molecular model of kerogen/'s nanostructure. *Nat Mater*, 15(5) :576–582. Article.
- [Broyden, 1969] Broyden, C. G. (1969). A new double-rank minimization algorithm. *Notices American Mathematical Society.*
- [Colliat, 2010] Colliat, J. B. (2010). *Modélisation numérique des matériaux hétérogènes : applications aux matériaux à matrice cimentaire.* HDR.
- [Danek and Kutakova,] Danek, J. and Kutakova, H. The mortar finite element method in 2d : implementation in matlab.
- [Farhat et al., 2001] Farhat, C., Lesoinne, M., and Le Tallec, P. (2001). FETI-DP : a dual-primal unified FETI method. *International Journal for Numerical Methods in Engineering*, 50 :1523–1544.

- [Farhat and Roux, 1991] Farhat, C. and Roux, F.-X. (1991). A method of finite tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32 :1205–1227.
- [Fletcher, 1970] Fletcher, R. (1970). A new approach to variable metric algorithms. *Ibid.*
- [Geuzaine and Remacle, 2009] Geuzaine, C. and Remacle, J.-F. (2009). Gmsh : A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11) :1309–1331.
- [Goldfarb, 1970] Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematical Computation*.
- [Gosselet and Rey, 2006] Gosselet, P. and Rey, C. (2006). Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13 :512–572.
- [Hautefeuille, 2009] Hautefeuille, M. (2009). *Numerical Modeling Strategy for Heterogeneous Materials : A FE Multi-scale and Component-based Approach*. PhD thesis, UTC.
- [Ibrahimbegovic and Wilson, 1991] Ibrahimbegovic, A. and Wilson, E. L. (1991). A modified method of incompatible modes. *Communication in applied numerical methods*, 7.
- [Jason, 2004] Jason, L. (2004). *Relation endommagement perméabilité pour les bétons application aux calculs de structures*. PhD thesis, École Centrale de Nantes.
- [Jodrey and Tory, 1979] Jodrey, W. S. and Tory, E. M. (1979). Simulation of random packing of spheres. *Simulation*, 32 :1–12.
- [Jourdain, 2014] Jourdain, X. (2014). *Étude numérique méso-macro des propriétés de transfert des bétons fissurés*. PhD thesis. Thèse de doctorat dirigée par Gatuingt, Fabrice Génie civil Cachan, Ecole normale supérieure 2014.
- [Jourdain et al., 2014] Jourdain, X., Colliat, J.-B., DeSa, C., Benboudjema, F., and Gatuingt, F. (2014). Upscaling permeability for fractured concrete : meso–macro numerical approach coupled to strong discontinuities. *International Journal for Numerical and Analytical Methods in Geomechanics*, 38(5) :536–550.
- [Kassiotis, 2009] Kassiotis, C. (2009). *Nonlinear fluid structure interaction : a partitioned approach and its application through component technology*. PhD thesis, ENS Cachan.
- [Kassiotis and Hautefeuille, 2008] Kassiotis, C. and Hautefeuille, M. (2008). *coFeap’s Manual*.
- [Kondo and Dormieux, 2003] Kondo, D. and Dormieux, L. (2003). Comportement macroscopique d’un milieu à fissures frottantes. *C. R. Mécanique*.
- [Lacour and Maday, 1997] Lacour, C. and Maday, Y. (1997). Two different approaches for matching nonconforming grids : the mortar element method and the feti method. *BIT numerical mathematics*, 37 :720–738.
- [Le Tallec, 1994] Le Tallec, P. (1994). Domain decomposition methods in computational mechanics. *Computational Mechanics Advances*.
- [Lions, 1988] Lions, P. L. (1988). On the schwartz alternating method, i. first international symposium on domain decomposition methods for partial differential equations.

- [Maday et al., 1989] Maday, Y., Mavriplis, C., and Patera, A. (1989). Nonconforming mortar element methods : Application to spectral discretizations. *Domain Decomposition Methods*.
- [Mandel, 1993] Mandel, J. (1993). Balanced domain decomposition. *Communication in applied numerical methods*, 9 :233–241.
- [Matthies and Strang, 1979] Matthies, H. G. and Strang, G. (1979). Solution of non-linear finite-element equations. *International Journal for Numerical Methods in Engineering*, (14) :1613–1626.
- [Mazars and Pijaudier Cabot, 1989] Mazars, J. and Pijaudier Cabot, G. (1989). Continuum damage theory-application to concrete. *Journal of Engineering Mechanics*, 115 :345–365.
- [Mocinski and Bargiel, 1991] Mocinski, J. and Bargiel, M. (1991). C-language program for the irregular dose packing of hard spheres. *Computer Physics Communications*, 64 :183–192.
- [Moscinski and Bargiel, 1991] Moscinski, J. and Bargiel, M. (1991). C-language program for the irregular dose packing of hard spheres. *Computer Physic*, 64 :183 – 192.
- [Moës et al., 1999] Moës, N., Dolbow, J., and Belytschko, T. (1999). A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46 :131 – 150.
- [Niekamp, 1995] Niekamp, R. (1995). *CTL home page*. <http://www.wire.tu-bs.de/forschung/projekte/ctl/>.
- [Niekamp, 2005] Niekamp, R. (2005). *CTL Manual for Linux/Unix for the usage with C++*. Institut für Wissenschaftliches Rechnen, TU Braunschweig.
- [Niekamp and Stein, 2002] Niekamp, R. and Stein, E. (2002). An object-oriented approach for parallel two- and three-dimensional adaptive finite element computations. *Computers and structures*, 80 :318 – 328.
- [Oliver, 1996] Oliver, J. (1996). Modeling strong discontinuities in solid mechanics via strain softening constitutive equations. part 1 : Fundamentals. part 2 : Numerical simulation. *International Journal Numerical Methodes Engineering*, 39.
- [Oliver, 2000] Oliver, J. (2000). On the discrete constitutive models induced by strong discontinuity kinematics and continuum constitutive equations. *International Journal of Solids and Structures*, 37(48-50) :7207 – 7229.
- [Park et al., 1997] Park, K. C., Justino, M. R., and Felippa, C. A. (1997). An algebraically partitioned FETI method for parallel structural analysis : algorithm description. *International Journal for Numerical Methods in Engineering*, 40 :2717–2737.
- [Pastor et al., 2016] Pastor, F., Anoukou, K., Pastor, J., and Kondo, D. (2016). Limit analysis and homogenization of porous materials with mohr–coulomb matrix. part ii : Numerical bounds and assessment of the theoretical model. *Journal of the Mechanics and Physics of Solids*, 91 :14 – 27.
- [Pijaudier-Cabot and Bazant, 1987] Pijaudier-Cabot, G. and Bazant, Z. (1987). Nonlocal damage theory. *Journal of Engineering Mechanics*, 113 :1512–1533.

- [Rateau, 2003] Rateau, G. (2003). *Méthodes Arlequin pour les problèmes mécaniques multi-échelles, Application à des problèmes de jonction et de fissuration de structures élancées*. PhD thesis, École Centrale Paris.
- [Richard et al., 2010] Richard, B., Ragueneau, F., Cremona, C., and Adelaide, L. (2010). Isotropic continuum damage mechanics for concrete under cyclic loading : Stiffness recovery, inelastic strains and frictional sliding. *Engineering fraction mechanics*.
- [Roubin, 2013] Roubin, E. (2013). *Modélisation EF et morphologique de milieux hétérogènes à l'échelle mésoscopique : applications aux matériaux à matrice cimentaire*. PhD thesis, ENS Cachan.
- [Roubin et al., 2015] Roubin, E., Vallade, A., Benkemoun, N., and Colliat, J.-B. (2015). Multi-scale failure of heterogeneous materials : A double kinematics enhancement for embedded finite element method. *International Journal of Solids and Structures*, 52 :180–196.
- [Salençon, 2002] Salençon, J. (2002). *De l'Élasto-plasticité au calcul de la rupture*.
- [Schwarz, 1869] Schwarz, H. A. (1869). Über einige abbildungsaufgaben. *Gesammelte Mathematische Abhandlungen*.
- [Series et al., 2003] Series, L., Feyel, F., and Roux, F.-X. (2003). Une méthode de décomposition de domaine à deux multiplicateurs de lagrange. *Actes de sixième colloque national en calcul de structures*, 3.
- [Shanno, 1970] Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematical Computation*.
- [Simo and Rifai, 1990] Simo, J. C. and Rifai, M. (1990). A class of mixed assumed strain methods and the methods of incompatible modes. *International Journal of Numerical Methods in Engineering*, 46 :1595 – 1638.
- [Stefanica, 1999] Stefanica, D. (1999). *Domain Decomposition Methods for Mortar Finite Elements*. PhD thesis, Courant Institute of Mathematical Sciences New York University.
- [Su et al.,] Su, K., Onaisi, A., and Garnier, A. A comprehensive methodology of evaluation of the fracability of a shale gas play.
- [Szyperski, 1998] Szyperski, C. (1998). *Component Software Beyond Object-Oriented Programming*. Addison-Wesley and ACM Press, Massachusetts.
- [Taylor, 1977] Taylor, R. (1977). Computer procedures for finite element analysis. *McGraw-Hill Book Co.*, pages 677–773.
- [Taylor, 2008] Taylor, R. L. (2008). *FEAP A Finite Element Analysis Program Version 8.2 User Manual*. Departement of Civil and Environmental Engineering University of California at Berkeley., URL = <http://www.ce.berkeley.edu/~rlt/feap/manual.pdf>.
- [Tory et al., 1973] Tory, E. M., Church, B. H., Tarm, M. K., and Ratner, M. (1973). Simulated random packing of equal spheres. *The Canadian Journal of Chemical Engineering*, 51 :484–493.
- [Washizu, 1982] Washizu, K. (1982). *Variational methods in elasticity and plasticity*. Pergamon Press, New York, 3 edition.
- [Wells and Sluys, 2001] Wells, G. N. and Sluys, L. J. (2001). Three-dimensional embedded discontinuity model for brittle fracture. *International Journal of Solids and Structures*, 38 :897–913.

- [Wilson, 1974] Wilson, E. L. (1974). The static condensation algorithm. *International Journal for Numerical Methods in Engineering*, 8.
- [Xie et al., 2012] Xie, Y. J., Hu, X. Z., Chen, J., and Lee, K. (2012). Micro-indentation fracture from flat-ended cylindrical indenter. *Fatigue Fracture of Engineering Materials Structures*, 35(1) :45–55.