
Calcul multi-échelles pour des composites à renfort Interlock sur la base FFT et FEM

ZHANG Xunjie

Rapport du stage fin d'étude

INSA - LaMCoS

Encadré par: M.NAOUAR Naïm, M.BLAL Nawfal

Tuteur pédagogique: M.FROSSARD Lionel

19 octobre 2018

Table des matières

Remerciements	4
Notations	5
Résumé	6
1 Introduction	7
1.1 Présentation du LaMCoS	7
1.1.1 Histoire	7
1.1.2 L'équipe MULTIMAP	8
1.1.3 Projets clefs du laboratoire	8
1.2 Introduction aux composites à microstructure périodique	9
2 Analyse Théorique	11
2.1 Mathématique	11
2.1.1 Fonction continue	11
2.1.2 Transformé de Fourier discret	11
2.2 Application	12
3 FFT : Méthode basée sur la transformée de Fourier	13
3.1 Loi de comportement	13
3.2 Opérateur de Green périodique	14
3.2.1 EXpression de C^0	14
3.2.2 Expressioin de Γ^0	14
3.3 Méthode FFT en alogorithme	15
3.3.1 Géométrie	15
3.3.2 Niveau de gris	15
3.3.3 Coefficients des milieux	16
3.4 Algorithme Discret	17
3.4.1 Traitement de figure d'entrée	17
3.4.2 Alogorithme	18
3.4.3 Convergence d'erreur	18
3.5 Exemple d'un carré 4x4	19
3.5.1 Espace réel et espace Fourier	19
3.6 Résultat	21
3.6.1 Erreur de convergence	21
3.6.2 Déformation moyenne	21
3.6.3 Matrice élasticité d'homogénéisation	23

4 FEM : Méthode basée sur l'éléments finis	25
4.1 Abaqus	25
4.2 Géométrie	26
4.3 Load	26
4.3.1 Points de référence	26
4.3.2 Conditions aux limites	27
4.4 Traitement du fichier <i>.inp</i>	28
4.4.1 Equation dans le fichier <i>inp</i>	28
4.4.2 Condition aux limites appliquées	29
4.5 Code abaqus	29
4.6 Résultat	29
5 Comparaison de deux méthodes de mesure	33
6 Conclusion du stage	34
Bibliographie	34
A Simulations et programmations	36
A.1 Code Matlab	36
A.2 Code abaqus	36
A.3 Code Python	36

Table des figures

1.1	Présentation général du laboratoire, histoire et structure	7
1.2	L'équipe multimap et sa fonction	8
1.3	Composite à renfort Interlock	9
1.4	Fibres circulaires en réseaux rectangle en 2D	9
1.5	Des choix de la cellule élémentaire	10
2.1	Composite à renfort interlock dans l'espace réel	12
2.2	Transformé de Fourier du composite à renfort interlock	12
3.1	4x4 pixels	15
3.2	16x16 pixels	16
3.3	32x32 pixels	16
3.4	64x64 pixels	16
3.5	256x256 pixels	16
3.6	La convergence de la première itération	20
3.7	L'erreur de convergence en fonction des itérations	21
3.8	ε_{11} pour la première hypothèse de déformation	22
3.9	ε_{11} moyenne en fonction de nombre du pixel	22
3.10	Déformation moyenne et contrainte moyenne pour hypothèse 1	23
3.11	Déformation moyenne et contrainte moyenne pour hypothèse 3	23
3.12	Déformation moyenne et contrainte moyenne pour hypothèse 3	24
4.1	Interface simple du Abaqus	25
4.2	Maillage de la pièce d'aluminium et de la fibre	26
4.3	Deux points de référence	26
4.4	Déplacement magnitude de la pièce dans le forme déformé	30
4.5	Déplacement magnitude de la pièce dans le forme qui n'est pas déformé	30
4.6	ε_{11} dans la forme déformé sous le premier condition	30
4.7	ε_{12} dans la forme déformé sous le premier condition	31
4.8	Contrainte $E\varepsilon_{22}$ dans la forme déformé sous le premier condition	31
4.9	Déformation moyenne pour première hypothèse sur Abaqus	31
4.10	S.Mises dans le forme déformé	32

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage.

Tout d'abord, je tiens à remercier M.Naïm Naouar, le chargé de recherche de CNRS, qui m'a proposé ce sujet du stage et m'a beaucoup aidé à comprendre le sujet.

Je tiens également à remercier M.Nawfal Blal, mon tuteur de stage et chef de Projet, pour sa patience et sa disponibilité pour répondre à mes nombreuses questions. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mon sujet.

Je tiens à remercier M.Frédéric Alizard et M.Frossard Lionel, mes tuteurs à l'université Lyon 1, qui m'ont permis de trouver mon stage et également pour leurs suivis lors de mon mission de stagiaire.

Enfin, je tiens à remercier mes camarades, Marion Albinet, Soukaina Raid, Matthieu Bara et Chen Shuai, qui m'ont beaucoup aidé sur le sujet et la correction du rapport dans mon stage.

Chacune de ces personnes ont rendu mon stage passionnant et m'ont permis de développer mes compétences. Je leur en suis particulièrement reconnaissante puisque grâce à elles, mon avenir est des plus prometteurs.

Notations

- E_m Module d'young d'aluminium
- E_f Module d'young de fibre
- ν_m Coefficient d'aluminium
- ν_f Coefficient de fibre
- ε Déformation dans l'espace réel
- $\hat{\varepsilon}$ Déformation dans l'espace Fourier
- Γ Opérateur Green
- A_{ijkh} i,j,k,h sont les dimensions varié 1,2,3
- x_1, x_2 Coordonates dans l'espace réel
- ξ_1, ξ_2 Fréquences dans l'espace Fourier
- $\langle A \rangle$ Moyenne de A
- $||A||$ Module de A
- λ, μ Coefficients du Lamé
- FFT Transformation de Fourier rapide
- FTD Transformation de Fourier Discret
- FEM Méthode de éléments finis
- fft2 FFT en 2D dans le logiciel Matlab
- ifft2 FFT inverse

Résumé

Du 02 Avril au 29 Septembre 2018, j'ai effectué un stage de fin d'étude d'une durée de six mois au Laboratoire de Mécanique des Contacts et des Structures - INSA de Lyon. Au cours de mon stage au LaMCoS, j'ai pu m'intéresser au sujet de Calculs multi-échelles pour des composites à renfortes interlock.

Le Laboratoire de Mécanique des Contacts et des Structures (LaMCoS) est une unité mixte de recherche de l'INSA de Lyon et du CNRS. Le laboratoire est structuré en 5 équipes de recherche thématiques. Chaque équipe possède la double compétence en expérimentation ainsi qu'en modélisation et simulation. J'ai fait mon stage dans l'équipe MULTIMAP.

Encadré par mes tuteurs M.Naïm Naouar et M.Nawfal Blal, j'ai fait les calculs des composites en utilisant une méthode Fast Fourier Transformation (FFT) dans le logiciel Matlab. Avec l'entrée d'une figure des milieux composites, j'ai fait l'analyse des données et exécution du programmation. Ensuite, j'ai fait la simulation de même pièce dans une autre logiciel Abaqus en utilisant une autre méthode commune Finite Element Method (FEM). Ensuite, je compare les deux résultats obtenus. L'avantage de la méthode FFT est le temps de calculer. Pour une matière composites inconnue, si on utilise la méthode FEM, il peut-être prend beaucoup plus de temps de simuler que la méthode FFT.

Il sûrement existe des difficultés dans ce stage. Pour la méthode FFT, on ne trouve pas la solution exacte en comparaison avec la solution numérique. J'ai essayé plusieurs fois de changer et exécuter le code dans le logiciel Matlab. Un autre défaut est la limitation de simulation. Je n'ai pas d'assez de temps de simuler la pièce en 3D. Le rapport du stage se développe principalement sur les simulation en 2D.

Plus largement, ce stage a été l'opportunité pour moi d'appréhender une méthode de simulation FFT pour des milieux composites. Il aussi m'aide d'améliorer la connaissance de programmation. Ce stage m'a permis de comprendre mieux dans la domaine de simulation et les caractérisation des milieux composites.

En vue de rendre le rapport du stage de 6 mois passés au LaMCoS, il apparaît logique de présenter à titre préalable le sujet du stage et remerciements, à savoir l'introduction du laboratoire et du sujet, puis d'envisager les deux méthodes de calcul et de simulation. Ensuite, il sera précisé les résultats obtenus par deux méthodes et il aura une conclusion du stage. Enfin, le rapport se termine avec une annexe, une référence et une bibliographie.

Chapitre 1

Introduction

1.1 Présentation du LaMCoS

1.1.1 Histoire

Le Laboratoire de Mécanique des Contacts et des Structures (LaMCoS) est une unité mixte de recherche (UMR5259) de l'INSA de Lyon et du CNRS (INSIS, Institut des sciences de l'ingénierie et des systèmes).

Les recherches menées au LaMCoS concernent la prédiction et l'analyse des performances et l'intégrité des systèmes mécaniques ou vivants, en conditions statiques, cycliques, dynamiques ou extrêmes, depuis le composant ou le constituant jusqu'au système complet.

Pluridisciplinaire dans les domaines de la Mécanique des Contacts et des Solides et de la dynamique des Structures, le LaMCoS offre un large champ de compétences en tribologie, dynamique rapide, vibratoire, contrôle, systèmes de transmission, machines tournantes, composites, procédés innovants, biomécanique, récupération d'énergie et physique des matériaux.

Le laboratoire est structuré en 5 équipes de recherche thématiques. Chaque équipe possède la double compétence en expérimentation ainsi qu'en modélisation et simulation.

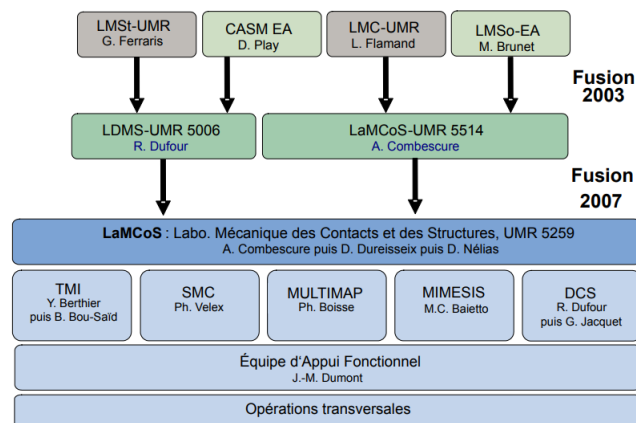


FIGURE 1.1 – Présentation général du laboratoire, histoire et structure

1.1.2 L'équipe MULTIMAP

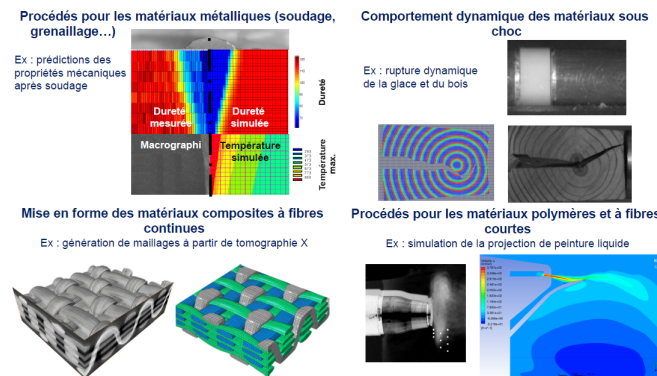


FIGURE 1.2 – L'équipe multimap et sa fonction

1.1.3 Projets clefs du laboratoire

LaMCoS développe ses projet avec beaucoup d'entreprises [1].

Chaires Industrielles

- Avreva-Safran : Life extension dans manufacturing processes
- SKF : Lubricated Interfaces for the Future
- Safran : Innovative mechanical transmissions for aeronautics
- Volov : Solutions for the Future of Urban Transport
- Michelin : Approches Multi-Echelles et Matériaux Innovants au service des Performances du Pneumatique

Equipex

- Durasol : Etude du vieillissement accéléré des composants et systèmes solaires photo-voltaïques et thermiques
- PHARE : Plate-forme machines tournantes pour la mÂîtrise des Risques Environnementaux

LabCom

- Drillab(DrillScan) : Laboratoire de simulation du forage pétrolier ou géothermique
- AD VITAM : AVNIR Engineering, Advanced vibrations test for the analysis of rotating machines
- Openlab PSA
- TRANSMECA sur les Transmissions Mécaniques(CETIM)
- CIRTRANS(Renaults trucks, Safran HE, Alstom Transport, GIMA, Texelis, Reel, ECAM, INSA, ECL)

1.2 Introduction aux composites à microstructure périodique

Aujourd'hui, il y a beaucoup de matériaux composites qui ont des microstructures périodiques. ça nous donne une raison de s'intéresser à l'étude et à la recherche de nature des matériaux. Il y a trois méthodes pour traiter le calcul des matériaux composites, ce stage me permet d'utiliser la deuxième méthode pour analyser et calculer les matériaux. C'est une méthode numérique utilisant directement les images des microstructures.

Pourquoi on s'intéresse aux composites à microstructure périodique ? Parceque la périodicité se rencontre plusieurs fois dans les composites artificiel, il va nous aider pour étudier les certaines composites naturels. Une seconde raison est que les composites périodique se prêtent bien aux calculs numériques dans la domaine élément finis avec des conditions aux limites bien posées. Dans mon stage, je vous présenter la pièce composite périodique :

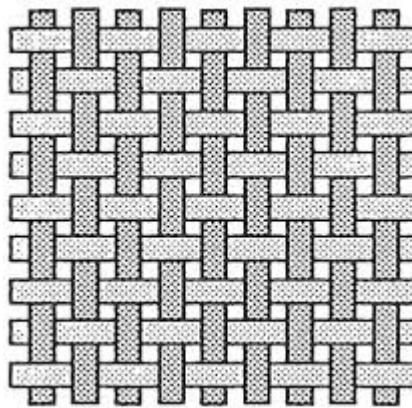


FIGURE 1.3 – Composite à renfort Interlock

C'est une pièce de la chaise d'avion. On utilise la technologie holographique pour construire cette pièce en 3D, mon sujet est calculer la matrice d'élasticité homogénéisée.

Une autre raison importante est que les milieux périodiques permettent de définir avec rigueur la notion de comportement homogénéisé. Pour mon stage, on définit un milieu périodique qui se présente au-dessous :

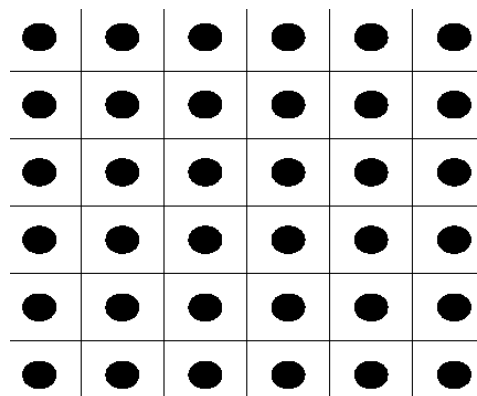


FIGURE 1.4 – Fibres circulaires en réseaux rectangle en 2D

Pour simplifier la pièce, on peut chercher une cellule caractéristique. Il y a de différents choix possible pour déterminer la cellule de base :

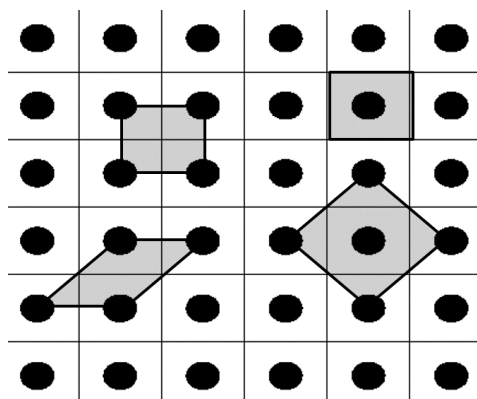


FIGURE 1.5 – Des choix de la cellule élémentaire

Dans mon stage, on choisit la cellule carré avec un cercle da la fibre au centre. Parce que cette cellule est plus simple que les autres. Et pour conduire cette cellule dans le logiciel de Matlab, il est plus facile de changer la fraction volumique.

Chapitre 2

Analyse Théorique

On commence par un rappel sur la transformée de Fourier en mathématique :

2.1 Mathématique

2.1.1 Fonction continue

Pour toutes les fonctions continues $f(x)$, on cherche sa transformée de Fourier $F(\xi)$ par :

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx \quad (2.1)$$

On va déduire la transformée de Fourier en 2D :

$$\hat{f}(\xi_1, \xi_2) = \int \int_{-\infty}^{\infty} f(x_1, x_2) e^{-2\pi i (\xi_1 x_1 + \xi_2 x_2)} dx_1 dx_2 \quad (2.2)$$

On peut aussi définir l'inverse de Fourier :

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{i2\pi x \xi} d\xi \quad (2.3)$$

et en 2D

$$f(x, y) = \int \int_{-\infty}^{\infty} \hat{f}(\xi_1, \xi_2) e^{i2\pi (\xi_1 x_1 + \xi_2 x_2)} d\xi_1 d\xi_2 \quad (2.4)$$

2.1.2 Transformée de Fourier discret

Néanmoins, quand on traite le problème en informatique ou mécanique, on utilise la transformée de Fourier discret :

$$\begin{aligned} \hat{f}(\xi) &= \sum_{x=0}^{N-1} f(x) e^{-i2\pi \frac{\xi x}{N}} \\ \hat{f}(\xi_1, \xi_2) &= \sum_{x_1=0}^{N_1-1} \sum_{x_2=0}^{N_2-1} f(x_1, x_2) e^{-i2\pi (\frac{\xi_1 x_1}{N_1} + \frac{\xi_2 x_2}{N_2})} \end{aligned} \quad (2.5)$$

et ses fonctions inversées :

$$\begin{aligned} f(x) &= \frac{1}{N} \sum_{\xi=0}^{N-1} \hat{f}(\xi) e^{i2\pi \frac{\xi x}{N}} \\ f(x_1, x_2) &= \frac{1}{N} \sum_{\xi_1=0}^{N_1-1} \sum_{\xi_2=0}^{N_2-1} \hat{f}(\xi_1, \xi_2) e^{i2\pi (\frac{\xi_1 x_1}{N_1} + \frac{\xi_2 x_2}{N_2})} \end{aligned} \quad (2.6)$$

On a les relations entre l'espace Réel et l'espace Fourier [2]

2.2 Application

On utilise cette transformé de Fourier en informatique et mécanique. Par exemple, on considère la figure d'une pièce de renforte interlock au-dessous :

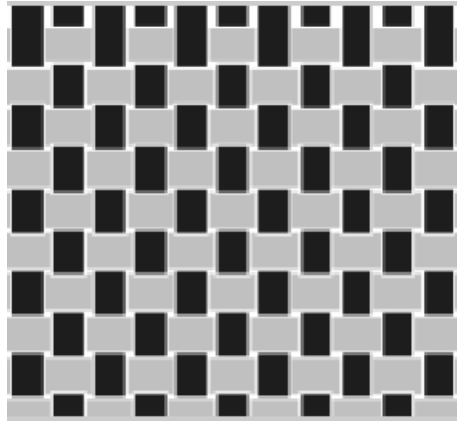


FIGURE 2.1 – Composite à renforte interlock dans l'espace réel

Dans la figure 2.1, les bandes noirs sont milieu 1, les bandes grises sont milieu 2, les deux milieux sont tissés. On applique la formule de transformé de Fourier dans ce milieu et on trouve la figure d'après de transformé :

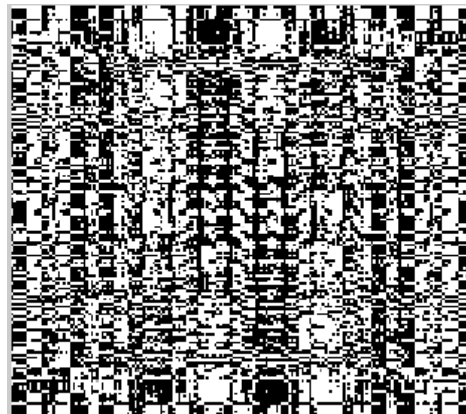


FIGURE 2.2 – Transformé de Fourier du composite à renforte interlock

Dans l'espace réel, les coordonnées sont les position du milieu, dans l'espace de Fourier, les coordonnées sont les fréquences.

Chapitre 3

FFT : Méthode basée sur la transformée de Fourier

La méthode FFT est une méthode numérique en utilisant directement les images des micro-structure. Soit un milieu composite quelconque, si on étudie ce milieu en utilisant la méthode FEM, on simule ce milieu à l'échelle fine, on a besoin d'une discrétisation avec plusieurs éléments, on trouvera les caractéristiques après simulation, mais il nous prend beaucoup de temps. C'est pourquoi on utilise la méthode FFT, parce que cette méthode prend moins de temps à cause de la transformée de Fourier. Dans mon stage, cette méthode est la base de l'homogénéisation en mécanique des matériaux [3].

3.1 Loi de comportement

La loi de comportement macroscopique obtenue a été ensuite implémentée dans un code de calcul par éléments finis. D'ici, on utilise des lois de comportement élastique linéaire :

$$\begin{aligned}\sigma &= C : \varepsilon \\ \sigma_{ij} &= C_{ijkl} \varepsilon_{kl}\end{aligned}\tag{3.1}$$

où C est la rigidité du matériau. ε est la déformation qui dépend de la déformation :

$$\varepsilon = \frac{1}{2}(\nabla u + \nabla^T u)\tag{3.2}$$

où u est la déformation causée par la dilatation thermique, la plasticité ou tout autre phénomène physique. Si le milieu est un composite, C dépend de la position x :

$$C = C(x)\tag{3.3}$$

L'objectif de l'homogénéisation est de trouver un comportement équivalent qui relie les champs moyens :

$$\langle \sigma \rangle = C^{hom} : \langle \varepsilon \rangle\tag{3.4}$$

Cela consiste à imposer $\langle \varepsilon \rangle$ sur la cellule et calculer $\langle \sigma \rangle$ par FFT et donc trouver le C^{hom} . J'ai seulement fait la simulation en 2D, on pose la condition de la déformation en 2D qui est :

$$\langle \varepsilon \rangle = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\tag{3.5}$$

La tableau C_{ijkh} est une matrice 3*3, pour trouver le résultat de la matrice élasticité homogénéisé, il faut aussi définir deux autres hypothèses :

$$\langle \varepsilon \rangle = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \langle \varepsilon \rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.6)$$

3.2 Opérateur de Green périodique

L'expression détaillée de l'opérateur de Green périodique Γ^0 pour les milieux isotropes homogènes dépend des coefficients de Lamé λ^0 et μ^0 d'un milieu de référence.

3.2.1 Expression de C^0

On prend un milieu de référence isotrope :

$$C_{ijkh}^0 = \lambda^0 \delta_{ij} \delta_{kh} + \mu^0 (\delta_{ik} \delta_{jh} + \delta_{ih} \delta_{jk}) \quad (3.7)$$

D'où, λ^0 et μ^0 sont les deux coefficients de Lamé.

3.2.2 Expression de Γ^0

On introduit la série de Fourier :

$$f(x) = \sum_k \hat{f}(\xi^k) e^{i\xi^k x} \quad (3.8)$$

où f est la force volumique pour une cellule avec volume V , et la transformée Fourier d'équation (3.8) est au-dessous :

$$\hat{f}(\xi^k) = \frac{1}{V} \int_V f(x) e^{i\xi^k x} dx \quad (3.9)$$

Dans l'espace réel, on a l'équation d'équilibre :

$$C_{ijkh} u_{k,hj} + f_i = 0 \quad (3.10)$$

où f_i est une force volumique. Pour les milieux Green Périodique, l'équation (3.10) devient :

$$\begin{aligned} C_{ijkh} g_{mk,hj} + \delta_{mi} \delta(x) &= 0 \\ \Gamma_{ijkh}(x) + g_{jk,ih}(x) &= 0 \end{aligned} \quad (3.11)$$

En même temps, on fait la transformée de Fourier d'équation (3.11) :

$$\begin{aligned} -C_{ijkh} \hat{g}_{mk} \xi_h \xi_j + \delta_{mi} &= 0 \\ \hat{\Gamma}_{ijkh}(\xi) - \xi_i \xi_h \hat{g}_{jk}(\xi) &= 0 \end{aligned} \quad (3.12)$$

On cherche les solutions générales d'équation (3.10) dans l'espace réel sur l'intervalle, on a :

$$\begin{aligned} u(x) &= -h * \tau + g * f \\ \varepsilon(x) &= -\Gamma * \tau + h * f \end{aligned} \quad (3.13)$$

où $*$ est convolution, et Γ est dans l'expression au-dessous :

$$\begin{aligned}\Gamma_{ijkh} &= \frac{1}{2} \left[\frac{\partial h_{ikh}}{\partial x_j}(x) + \frac{\partial h_{ikj}}{\partial x_h}(x) \right] \\ &= -\frac{1}{4} \left[\frac{\partial^2 g_{ih}}{\partial x_j \partial x_k}(x) + \frac{\partial^2 g_{ik}}{\partial x_j \partial x_h}(x) + \frac{\partial^2 g_{jh}}{\partial x_i \partial x_k}(x) + \frac{\partial^2 g_{jk}}{\partial x_i \partial x_h}(x) \right]\end{aligned}\quad (3.14)$$

Dans l'espace de Fourier, on a :

$$\begin{aligned}\hat{u}(\xi) &= -\hat{h} : \hat{\tau} + \hat{g} \cdot \hat{f} \\ \hat{\varepsilon}(\xi) &= -\hat{\Gamma} : \hat{\tau} + \hat{h} \cdot \hat{f}\end{aligned}\quad (3.15)$$

où $\hat{\Gamma}(\xi)$ est égal à :

$$\hat{\Gamma}_{ijkh}(\xi) = \frac{1}{4\mu^0|\xi|} \left(\delta_{ki}\xi_h\xi_j + \delta_{hi}\xi_k\xi_j + \delta_{kj}\xi_h\xi_i + \delta_{hj}\xi_k\xi_i \right) - \frac{\lambda^0 + \mu^0}{\mu^0(\lambda^0 + 2\mu^0)} \frac{\xi_i\xi_j\xi_k\xi_h}{|\xi|^4} \quad (3.16)$$

3.3 Méthode FFT en algorithme

3.3.1 Géométrie

Au début, on introduit volume fraction. Dans un mélange, la fraction volumique V_i est définie par :

$$V_i = \frac{v_i}{\sum_{j=1}^l v_j} \quad (3.17)$$

le volume V_i du composant i divisé par la somme des volumes de tous les composants utilisés pour fabriquer ce mélange. La somme de volume fraction égale 1.

3.3.2 Niveau de gris

Sous Matlab, on peut utiliser les commandes pour analyser les figures. On a une pièce composite d'une chaise d'avion pour développer cette méthode. En utilisant l'holographie, on peut obtenir la figure de la pièce en 2D, ensuite on utilise Matlab pour traiter la figure. Par défaut, on va simuler une pièce rectangle créé dans le logiciel Matlab.

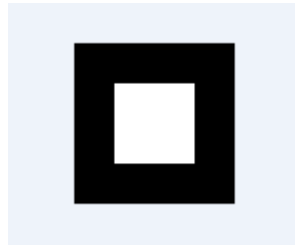


FIGURE 3.1 – 4x4 pixels

À l'avenir, on cherchera à appliquer la méthode à une image en couleur provenant de l'holographie mais pour l'instant, simplifions le problème avec une image créée directement en noir et blanc. En général, on utilise les niveaux de gris entre 0 et 255 pour définir la couleur noir ou blanc. Un pixel est un petit rectangle dans la figure de la pièce. On peut voir dans la figure

3.1, il y a $4 \times 4 = 16$ pixels dans l'espace réel de longueur 1×1 . Les pixels autour du centre sont noirs, on peut les définir comme les pixels appartenant au milieu 1 avec un Module d'Young de 70 GPa et un coefficient de poisson 0.3, Au centre, les 4 pixels sont blancs. On définit les pixels comme appartenant au deuxième milieu avec un Module d'Young 42 GPa et un coefficient de poisson 0.2.

On refait la simulation avec plus de pixels pour savoir les résultats fouuillées. Quand la fraction volume est fixée, on augement la nombre des pixels :

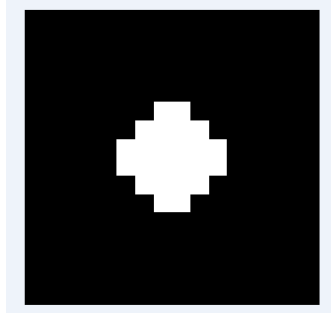


FIGURE 3.2 – 16x16 pixels

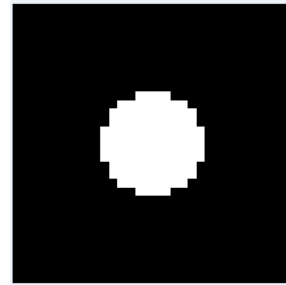


FIGURE 3.3 – 32x32 pixels

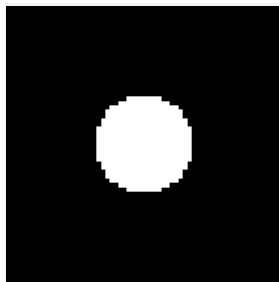


FIGURE 3.4 – 64x64 pixels

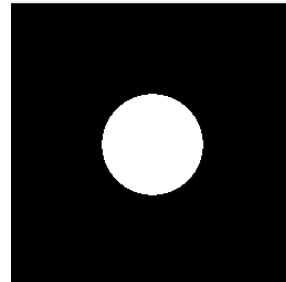


FIGURE 3.5 – 256x256 pixels

On voit que quand la nombre du pixel augmente, la forme de la fibre est plus proche d'une cercle. La simulation est réalisée sur un carré donc le nombre de pixel doit être multiplié par un multiple de 4.

3.3.3 Coefficients des milieux

Dans ce méthode, on définit les caractéristiques du milieu [4]

Module d'Young et Poisson

$$E_1 = 70 \text{ GPa}$$

$$E_2 = 420 \text{ GPa}$$

$$\nu_1 = 0.3$$

$$\nu_2 = 0.2$$

Coefficients de Lamé

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

$\lambda_1, \lambda_2, \mu_1, \mu_2, \lambda_0$, et μ_0 sont calculés par les coefficients de module d'Young et Poisson, ici, λ^0 et μ^0 sont les coefficients homogènes

3.4 Algorithme Discret

3.4.1 Traitement de figure d'entrée

Espace Réel

$$x_d(i_1, i_2) = \left((i_1 - 1) \cdot \frac{T_1}{N_1}, (i_2 - 1) \cdot \frac{T_2}{N_2} \right)$$

$$i_1 = 1, 2, 3, \dots, N_1$$

$$i_2 = 1, 2, 3, \dots, N_2$$

Espace Fourier

ξ_d est la fréquence correspondante à la coordonnée de l'espace réel

si N_j est pair,

$$\xi_j = \left(-\frac{N_j}{2} + 1\right) \frac{1}{T_j}, \left(-\frac{N_j}{2} + 2\right) \frac{1}{T_j}, \dots, -\frac{1}{T_j}, 0, \frac{1}{T_j}, \dots, \left(\frac{N_j}{2} - 1\right) \frac{1}{T_j}, \frac{N_j}{2} \frac{1}{T_j}$$

si N_j est impaire,

$$\xi_j = -\frac{N_j - 1}{2} \frac{1}{T_j}, \dots, -\frac{1}{T_j}, 0, \frac{1}{T_j}, \dots, \frac{N_j - 1}{2} \frac{1}{T_j}$$

3.4.2 Algorithme

Initialisation

$$\begin{aligned}\varepsilon^0(x_d) &= E \\ \sigma^0(x_d) &= c(x_d) : \varepsilon^0(x_d)\end{aligned}$$

Itération

ε^i et σ^i sont connus pour tous les pixels

$$\widehat{\sigma}^i = FFF(\sigma^i)$$

convergence test

$$\widehat{\varepsilon}^{i+1}(\xi_d) = \widehat{\varepsilon}^i(\xi_d) - \widehat{\Gamma}^0(\xi_d) : \widehat{\sigma}^i(\xi_d) \quad \forall \xi_d \neq 0$$

$$\widehat{\varepsilon}^{i+1}(0) = E$$

$$\varepsilon^{i+1} = iFFF(\widehat{\varepsilon}^{i+1})$$

$$\sigma^{i+1}(x_d) = c(x_d) : \widehat{\varepsilon}^{i+1}(x_d) \quad \forall x_d \in V$$

3.4.3 Convergence d'erreur

Dans l'espace réel ou dans l'espace Fourier, on peut calculer l'erreur en même temps. Sur le plan numérique, l'utilisation de la transformée de Fourier rapide permet d'accélérer considérablement les calculs. La convergence est atteinte lorsque σ^{i+1} est en équilibre, typiquement lorsque :

$$e^i = \frac{\sqrt{\langle ||\text{Div}(\sigma^i)||^2 \rangle}}{||\langle \sigma^i \rangle||} = \frac{\sqrt{\langle ||\xi \cdot \widehat{\sigma}^i(\xi)||^2 \rangle}}{||\widehat{\sigma}^i(0)||} \quad (3.18)$$

est inférieur à une valeur donnée (10^{-4} dans nos calculs). Parce que les calculs des matrices sont très compliqués dans l'espace réel à cause de divergence, donc on va calculer l'erreur dans l'espace Fourier. La vitesse de convergence de l'algorithme dépend du choix de milieu de référence. Le choix optimal pour un biphasé isotrope est un milieu isotrope de constantes de Lamé $\frac{1}{2}(\lambda^1 + \lambda^2)$ et $\frac{1}{2}(\mu^1 + \mu^2)$. Avec ce choix, le nombre d'itération à convergence est une fonction linéaire du contraste entre les phases. La méthode ne converge pas dans le cas d'un contraste infini (vides ou inclusions rigides).

3.5 Exemple d'un carré 4x4

Dans cette section, on donne un exemple de simulation, on réalise ce calcul sur un carré 1x1 avec 16 pixels.

3.5.1 Espace réel et espace Fourier

On fait l'application numérique pour ce carré [5]. Les coordonnées des pixels dans l'espace réel sont :

$$\begin{bmatrix} (0,0) & (0.25,0) & (0.5,0) & (0.75,0) \\ (0,0.25) & (0.25,0.25) & (0.5,0.25) & (0.75,0.25) \\ (0,0.5) & (0.25,0.5) & (0.5,0.5) & (0.75,0.5) \\ (0,0.75) & (0.25,0.75) & (0.5,0.75) & (0.75,0.75) \end{bmatrix} \quad (3.19)$$

Ces pixels sont rangés 4x4, donc pour avoir les fréquences dans l'espace Fourier, on utilise la formule ci-dessous :

$$\xi_j = \left(-\frac{N_j}{2} + 1\right) \frac{1}{T_j}, \left(-\frac{N_j}{2} + 2\right) \frac{1}{T_j}, \dots, -\frac{1}{T_j}, 0, \frac{1}{T_j}, \dots, \left(\frac{N_j}{2} - 1\right) \frac{1}{T_j}, \frac{N_j}{2} \frac{1}{T_j} \quad (3.20)$$

On peut trouver les fréquences dans l'espace de Fourier :

$$\begin{bmatrix} (-1,-1) & (0,-1) & (1,-1) & (2,-1) \\ (-1,0) & (0,0) & (1,0) & (2,0) \\ (-1,1) & (0,1) & (1,1) & (2,1) \\ (-1,2) & (0,2) & (1,2) & (2,2) \end{bmatrix} \quad (3.21)$$

On calcule le tableau d'élasticité qui est un tableau de 2*2*2*2 (en 2D). Pour milieu aluminium, par la formule (3.7), on a :

$$\begin{aligned} C_1(:, :, 1, 1) &= 1.0 * e^{10} * \begin{bmatrix} 9.4231 & 0 \\ 0 & 4.0385 \end{bmatrix} \\ C_1(:, :, 2, 1) &= 1.0 * e^{10} * \begin{bmatrix} 0 & 2.6923 \\ 2.6923 & 0 \end{bmatrix} \\ C_1(:, :, 1, 2) &= 1.0 * e^{10} * \begin{bmatrix} 0 & 2.6923 \\ 2.6923 & 0 \end{bmatrix} \\ C_1(:, :, 2, 2) &= 1.0 * e^{10} * \begin{bmatrix} 4.0385 & 0 \\ 0 & 9.4231 \end{bmatrix} \end{aligned} \quad (3.22)$$

Et pour le milieu fibre on a :

$$\begin{aligned} C_1(:, :, 1, 1) &= 1.0e^{11} * \begin{bmatrix} 4.6667 & 0 \\ 0 & 1.1667 \end{bmatrix} \\ C_1(:, :, 2, 1) &= 1.0e^{11} * \begin{bmatrix} 0 & 1.7500 \\ 1.7500 & 0 \end{bmatrix} \\ C_1(:, :, 1, 2) &= 1.0e^{11} * \begin{bmatrix} 0 & 1.7500 \\ 1.7500 & 0 \end{bmatrix} \\ C_1(:, :, 2, 2) &= 1.0e^{11} * \begin{bmatrix} 1.1667 & 0 \\ 0 & 4.6667 \end{bmatrix} \end{aligned} \quad (3.23)$$

Ensuite, on calcul la scalaire entre la matrice de déformation et la tableau de élasticité pour chercher la contrainte, au début, on utilise le première hypothèse de déformation 3.5, le résultat est dans une tableau de $4*4*2*2$ (il y a une tableau $2*2$ dans chaque pixel), on donnera la matrice au-dessous :

$$\begin{aligned}
 \sigma^0(:, :, 1, 1) &= 1.0e^{11} * \begin{bmatrix} 0.9423 & 0.9423 & 0.9423 & 0.9423 \\ 0.9423 & 0.4667 & 0.4667 & 0.9423 \\ 0.9423 & 0.4667 & 0.4667 & 0.9423 \\ 0.9423 & 0.9423 & 0.9423 & 0.9423 \end{bmatrix} \\
 \sigma^0(:, :, 2, 1) &= 1.0e^{11} * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \sigma^0(:, :, 1, 2) &= 1.0e^{11} * \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \sigma^0(:, :, 2, 2) &= 1.0e^{11} * \begin{bmatrix} 0.4308 & 0.4308 & 0.4308 & 0.4308 \\ 0.4308 & 1.1667 & 1.1667 & 0.4308 \\ 0.4308 & 1.1667 & 1.1667 & 0.4308 \\ 0.4308 & 0.4308 & 0.4308 & 0.4308 \end{bmatrix}
 \end{aligned} \tag{3.24}$$

Maintenant, toutes les entrées sont prêts, on entrer la boucle. On va calculer la tableau de $\hat{\sigma}$ dans l'espace de Fourier. Dans le code matlab on peut utiliser *fft2* directement, ou on utilise la formule (2.5). Si on utilise le code matlab *fft2*, on donnera la matrice du premier pixel :

$$\hat{\sigma}^1(1, 1, :, :) = 1.0e^{11} * \begin{bmatrix} 2.994 + 0.000i & 0 + 0.00i \\ 0 + 0.00i & 0.95128 + 0.00i \end{bmatrix} \tag{3.25}$$

Si on utilise la formule (2.5), on trouve :

$$\hat{\sigma}^1(1, 1, :, :) = 1.0e^{13} * \begin{bmatrix} 0.1022 + 0.2468i & 0 + 0.00i \\ 0 + 0.00i & 0.03154 + 0.07613i \end{bmatrix} \tag{3.26}$$

Maintenant, il y deux formules différents pour la transformé de Fourier, J'ai déjà essayé les deux formules à chercher les résultats, les codes marches mais les deux résultats de moyenne déformation sont loin de 1. On fait une hypothèse sur la déformation (3.5), donc le résultat de déformation moyenne exact est 1 d'après de calculs. C'est une défaut pour le méthode FFT. On continue les calculs avec le résultat (3.26). Ensuite, on examine la convergence à partir de la formule (3.18). Évidemment, il y a deux formules de vérifier l'erreur de convergence. Soit dans l'espace de réel, soit dans l'espace de Fourier. En fait, c'est difficile pour moi à réaliser la divergence dans l'espace réel sur le logiciel matlab, donc on utilise le deuxième formule dans l'espace de Fourier. On calcule la convergence au-dessous :

```

>> erreur = convergence_test(X,Y,Sigma_i)

erreur =

    1.0062e+14

```

FIGURE 3.6 – La convergenece de la première itération

L'erreur de convergence est plus grande que 10^{-4} , on continue notre boucle. On fait une remarque ici, lorsque l'opérateur green (3.16) n'est pas approprié pour les fréquences qui sont nulles. Donc pour des pixels qui ont de fréquence nulle, on change la déformation des pixels par la déformation initial :

$$\widehat{\varepsilon}^{i+1}(0) = \langle \varepsilon \rangle \quad (3.27)$$

On calcul la scalaire entre opérateur green et la matrice de contrainte dans l'espace de Fourier et on trouve $\widehat{\varepsilon}^2(\xi)$:

$$\widehat{\varepsilon}^2(1, 1, :, :) = 1.0e2 * \begin{bmatrix} 0.0208 + 0.0503i & -1.1923 - 2.8784i \\ -1.1923 - 2.8784i & 0.5579 + 1.3468i \end{bmatrix} \quad (3.28)$$

On utilise la Transformée de Fourier inversée pour trouver la tableau de déformation et le tableau de contrainte dans l'espace réel. Ici, on ne donne pas le résultat des deux tableaux. On finis la première itération, et on recommence la boucle jusqu'à l'erreur de convergence est plus petite que 10^{-4} . On arrête la boucle et on calcule la moyenne de déformation et contrainte.

3.6 Résultat

3.6.1 Erreur de convergence

On plot l'erreur de convergence en fonction de itération.

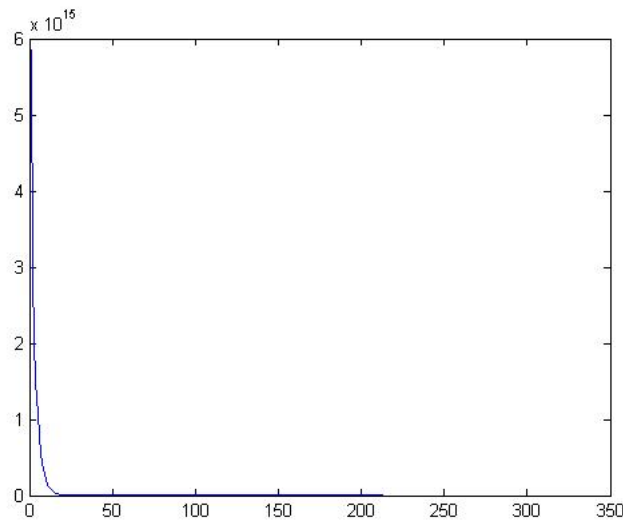
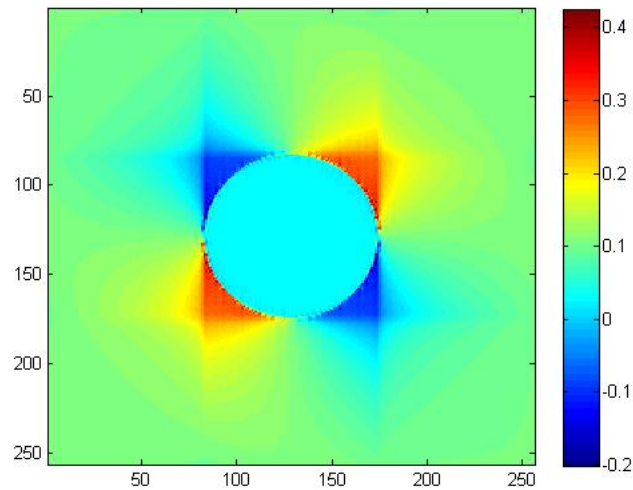


FIGURE 3.7 – L'erreur de convergence en fonction des itérations

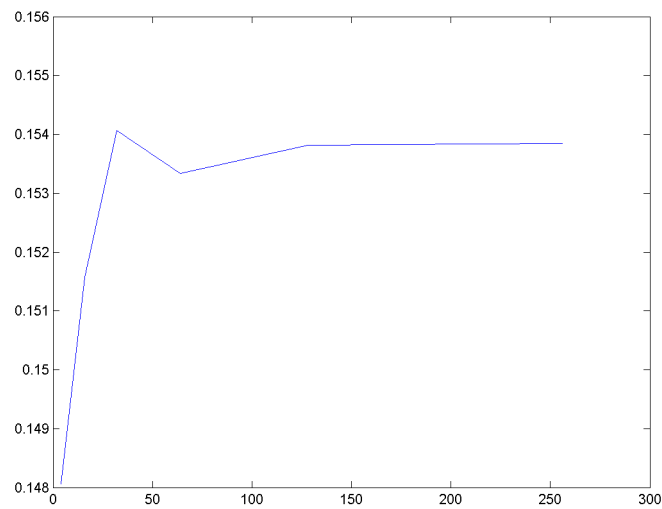
On voit que l'erreur diminue quand l'itération augmente. On arrête la boucle quand l'erreur de convergence est plus petite que 10^{-4} .

3.6.2 Déformation moyenne

Pour hypothèse de déformation (3.5), on calcule les contraintes ε_{11} pour tous les pixels, et on trace :

FIGURE 3.8 – ε_{11} pour la première hypothèse de déformation

Maintenant, on augmente le nombre du pixel par : 4, 16, 32, 64, 128, 256, et on plot E_{11} moyenne pour tous les pixels en fonction de nombre de pixels. Et on a :

FIGURE 3.9 – ε_{11} moyenne en fonction de nombre du pixel

On trouve la moyenne de contrainte E_{11} augment quand le nombre du pixel augment, mais il converge vers 0.155. À partir d'équation (3.1), on a $\varepsilon_{11} = 0.155$. On transforme la matrice (3.5) en un vecteur, qui est plus général dans l'étude :

$$\langle \varepsilon_1 \rangle = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \Rightarrow \langle \varepsilon_1 \rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (3.29)$$

Ee on trouve la déformation moyenne au dessous :


```

d_moy =
    0.1480 + 0.0097i  -0.3520 + 0.0029i   0.1480 - 0.0039i

c_moy =
    1.0e+10 *

    2.5583 - 0.0239i  -2.5583 + 0.0239i   2.5583 - 0.0239i

```

FIGURE 3.10 – Déformation moyenne et contrainte moyenne pour hypothèse 1

On calcule les déformations sans complexe et contrainte moyenne , on récrit ici :

$$\langle \varepsilon_1^{moy} \rangle = \begin{bmatrix} 0.1480 \\ -0.3520 \\ 0.1480 \end{bmatrix} \quad \langle \sigma_1^{moy} \rangle = 1.0e^{10} * \begin{bmatrix} 2.5583 \\ -2.5583 \\ 2.5583 \end{bmatrix} \quad (3.30)$$

3.6.3 Matrice élasticité d'homogénéisation

Pour trouver la matrice d'élasticité d'homogénéisation, on utilise la formule :

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{12} \\ \sigma_{22} \end{bmatrix} = \begin{bmatrix} C_{1111}^{hom} & C_{1211}^{hom} & C_{2211}^{hom} \\ C_{1112}^{hom} & C_{1212}^{hom} & C_{2212}^{hom} \\ C_{1122}^{hom} & C_{1222}^{hom} & C_{2222}^{hom} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{22} \end{bmatrix} \quad (3.31)$$

Parce que $\varepsilon_1 = [100]^t$, et $\langle \sigma_1^{moy} \rangle$ est déjà calculée, on peut trouver la première colonne :

$$\begin{bmatrix} C_{1111}^{hom} & \dots & \dots \\ C_{1112}^{hom} & \dots & \dots \\ C_{1122}^{hom} & \dots & \dots \end{bmatrix} = 1.0e^{10} * \begin{bmatrix} 2.5583 & \dots & \dots \\ -2.5583 & \dots & \dots \\ 2.5583 & \dots & \dots \end{bmatrix} \quad (3.32)$$

On utilise les autre deux hypothèse de déformation (3.6), on trouve les autre éléments de la matrice de matrice C^{hom} . Les résultats sont :

```

d_moy =
    -0.1480 + 0.0012i   0.3520 + 0.0012i  -0.1480 + 0.0012i

c_moy =
    1.0e+10 *

    -2.5583 + 0.0217i   2.5583 - 0.0217i  -2.5583 + 0.0217i

```

FIGURE 3.11 – Déformation moyenne et contrainte moyenne pour hypothèse 3

```

d_moy =
    0.1480 - 0.0039i  -0.3520 + 0.0029i   0.1480 + 0.0097i

c_moy =
    1.0e+10 *
    2.5583 - 0.0239i  -2.5583 + 0.0239i   2.5583 - 0.0239i

```

FIGURE 3.12 – Déformation moyenne et contrainte moyenne pour hypothèse 3

On trouve la matrice d'élasticité d'homogénéisation au-dessous :

$$C^{hom} = 1.0e^{10} * \begin{bmatrix} 2.5583 & -2.5583 & 2.5583 \\ -2.5583 & 2.5583 & -2.5583 \\ 2.5583 & -2.5583 & 2.5583 \end{bmatrix} \quad (3.33)$$

Chapitre 4

FEM : Méthode basée sur l'éléments finis

En analyse numérique, la méthode des éléments finis [6] est utilisée pour résoudre numériquement des équations aux dérivées partielles. Celles-ci peuvent par exemple représenter analytiquement le comportement dynamique de certains systèmes mécaniques.

On utilise la méthode élément fini pour analyser le composite, dans le stage on utilise le logiciel Abaqus pour faire cette simulation. Comme les autres logiciels, Nastran, Patran, Absqus a de même fonction de simuler. On commence par la création de géométrie

4.1 Abaqus

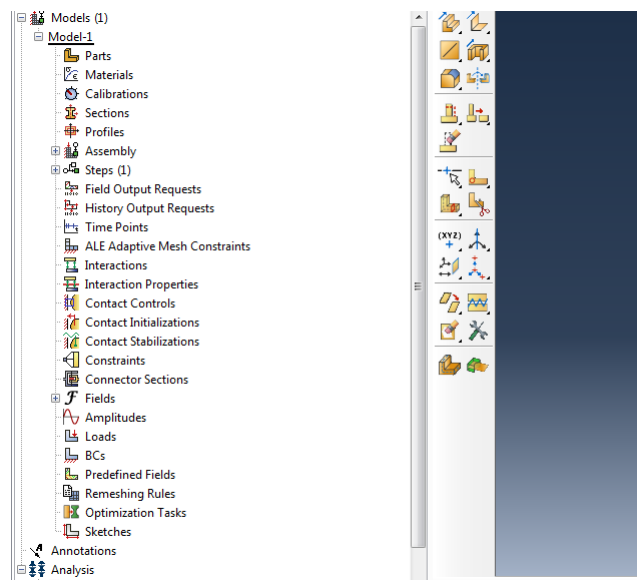


FIGURE 4.1 – Interface simple du Abaqus

4.2 Géométrie

On crée un carré de taille 1×1 situé à $(-0.5, -0.5)$, dans le centre du carré, on trace un cercle, son rayon est calculé par la fraction volumique :

$$r = \sqrt{\frac{l^2 * vf}{\pi}} = 0.18 \quad (4.1)$$

On simule ce carré par les éléments type de 'CPS3', la distance minimum de l'élément est 0.01,

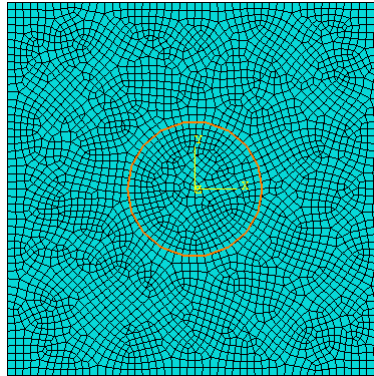


FIGURE 4.2 – Maillage de la pièce d'aluminium et de la fibre

4.3 Load

4.3.1 Points de référence

Sur Abaqus, on crée deux points qui sont indépendants du carré pour appliquer les conditions aux limites.

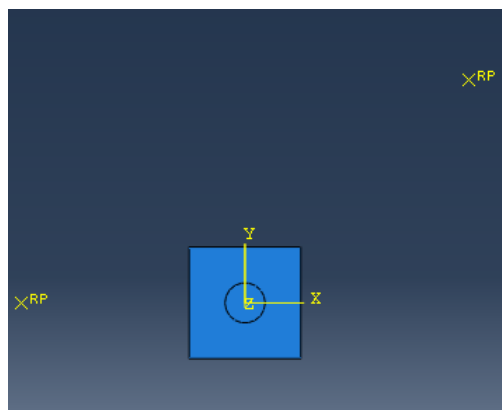


FIGURE 4.3 – Deux points de référence

4.3.2 Conditions aux limites

Déformation hypothèse 1

On définit les quatres frontièrespour ajouter les conditions aux limites en 2D, avec :

$$< \varepsilon_1 > = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.2)$$

et

$$\Delta u = \begin{pmatrix} \Delta u_x \\ \Delta u_y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \Delta x \\ 0 \end{pmatrix} = \begin{pmatrix} x^+ - x^- \\ y^+ - y^- \end{pmatrix} \quad (4.3)$$

Donc pour chaque point de référence de référence, on a une équation entre la frontière gauche et la frontière droite, et une autre equation entre la frontière du haut et la frontière du bas :

$$\begin{cases} u_x^+ - u_x^- = 1 \\ u_y^+ - u_y^- = 0 \end{cases} \quad \begin{cases} u_x^+ - u_x^- = 0 \\ u_y^+ - u_y^- = 0 \end{cases} \quad (4.4)$$

Déformation hypothèse 2

La deuximème condition au limite est :

$$< \varepsilon_2 > = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (4.5)$$

et

$$\Delta u = \begin{pmatrix} \Delta u_x \\ \Delta u_y \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} 0 \\ \Delta y \end{pmatrix} = \begin{pmatrix} x^+ - x^- \\ y^+ - y^- \end{pmatrix} \quad (4.6)$$

Comme l'hypothèse 1 dans (4.4), on a une équation entre la frontière gauche et la frontière droite, et une autre equation entre la frontière du haut et la frontière du bas :

$$\begin{cases} u_x^+ - u_x^- = 0 \\ u_y^+ - u_y^- = 0 \end{cases} \quad \begin{cases} u_x^+ - u_x^- = 0 \\ u_y^+ - u_y^- = 1 \end{cases} \quad (4.7)$$

Déformation hypothèse 3

La troisième condition au limite est :

$$< \varepsilon_3 > = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (4.8)$$

et

$$\Delta u = \begin{pmatrix} \Delta u_x \\ \Delta u_y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \Delta y \\ \Delta x \end{pmatrix} = \begin{pmatrix} x^+ - x^- \\ y^+ - y^- \end{pmatrix} \quad (4.9)$$

Donc pour chaque point de référence de référence on a :

$$\begin{cases} u_x^+ - u_x^- = 0 \\ u_y^+ - u_y^- = 1 \end{cases} \quad \begin{cases} u_x^+ - u_x^- = 1 \\ u_y^+ - u_y^- = 0 \end{cases} \quad (4.10)$$

4.4 Traitement du fichier *.inp*

Comme les autres logiciels, Abaqus édite un fichier *.inp* pour faire le calcul. Donc il faut créer ce fichier, il y a deux façons de rédiger ce fichier, soit par le logiciel Abaqus, soit par écrire. Dans mon stage, on crée ce fichier en utilisant Abaqus, pour les conditions aux limites, on va modifier les codes dans le fichier *.inp*, c'est parce que il est impossible d'ajouter les conditions aux limites périodiques entre les frontières et un point de référence sur le logiciel. La documentation [7] aide pour cette étape.

4.4.1 Equation dans le fichier *inp*

Pour ces lignes de conditions aux limites, on explique un peu au dessous :

$$A_1 u_i^P + A_2 u_j^Q + \dots + A_N u_k^R = 0 \quad (4.11)$$

L'équation 4.11 s'applique à la contrainte des multi-noeuds. D'où u_i^P est une variable nodal correspondante au noeud P de dimension i , et A_1 est un coefficient qui dépend du mouvement du noeud correspondante. N est le nombre total du noeud. Par exemple :

$$u_3^5 - u_1^6 + u_3^{1000} = 0 \quad (4.12)$$

D'où $N = 3$, $P = 5$, $i = 3$, $A_1 = 1.0$, $Q = 6$, $j = 1$, $A_2 = -1.0$, $R = 1000$, $k = 3$ et $A_3 = 1.0$. Il présente que la somme entre noeud 5 en dimension 3, noeud 6 en dimension 1, et noeud 1000 en dimension 3 égal 0. Mais dans le fichier *.inp*, on récrit sous la forme :

*Equation

N

$P, i, A_1, Q, j, A_2, \dots, R, k, A_N$

(4.13)

Donc sur le fichier *.inp*, l'équation 4.12 devient :

*Equation

3

5,3,1.0,6,1,-1.0,1000,3,1.0

(4.14)

Pourquoi est-ce qu'on ne définit pas l'équation dans le logiciel ? Parce que le logiciel applique l'équation (4.11) sur des points. Si une condition au limite se passe entre une point et deux frontières, on écrit les équations dans le fichier *.inp* directement. On définit quatre sets pour résoudre le problème de condition limite, ils sont "left", "right", "top" et "bot". Chaque set contient tous les noeuds dans chaque frontière. On donne la relation entre la frontière gauche, la frontière droit et un point de référence numéro 1000. Une remarque dans le fichier *.inp*, chaque ligne qui commence par "*" est une commentaire.

*Equation

3

** trois termes dans l'equation

Left, 1, 1

** tous les noeuds set dans la frontière gauche avec dimension = 1, coefficient = 1.0 (4.15)

Right, 1, -1

** tous les noeuds set dans la frontière droit avec dimension = 1, coefficient = -1.0

1000, 1, 1

** point de référence dimension = 1, coefficient = 1.0

C'est la relation entre la frontière du haut, la frontière du bas et le point de référence numéro 2000 au-dessous :

*** Equation**

3

** trois termes dans l'équation

Top, 2, 1

** tous les noeuds set dans la frontière du haut avec dimension = 2, coefficient = 1.0 (4.16)

Bot, 2, -1

** tous les noeuds set dans la frontière du bas avec dimension = 2, coefficient = -1.0

2000, 2, 1

** point de référence dimension = 2, coefficient = 1.0

4.4.2 Condition aux limites appliquées

Maintenant, les équations de appliquer conditions limites sont définis, on utilise la forme au-dessous pour réaliser les conditions limites :

*** Boundary**

1000, 1, 1, 1

** on décrit les conditions limites pour le noeud 1000 à l'aide des équations (4.15) et (4.16)
(4.17)

et les conditions aux limites entre la frontière du haut, la frontière du bas et le point de référence au-dessous :

*** Boundary**

2000, 2, 2, 0

(4.18)

Dans l'équation (4.17), dimension 1 est l'axe x, on peut retrouver que le déplacement des noeuds gauches moins le déplacement des noeuds droites est égal 1, qui est sûrement correspondant à l'équation (4.4) : $u_x^+ - u_x^- = 1$. On a $u_y^+ - u_y^- = 0$ dans l'équation (4.4) sur y, cette équation est retrouvée dans l'équation (4.18).

4.5 Code abaqus

On donne un exemple de la programmation ici. On peut retrouver les codes simplifiés dans l'annexe ci-joint. Parce ce-que le code est très long, j'ai importé le code ici [8], vous pouvez trouver le code complet.

4.6 Résultat

On seulement présent les résultats dans le premier condition (4.2). Pour la présentation du résultat dans le logiciel abaqus, au début, on compare le mesh déformé et le mesh qui n'est pas déformé. On met la figure de déplacement magnitude déformé au-dessous :

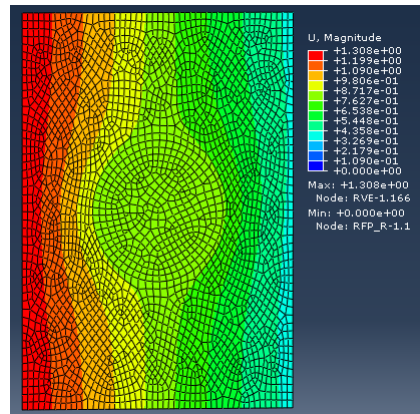


FIGURE 4.4 – Déplacement magnitude de la pièce dans le forme déformé

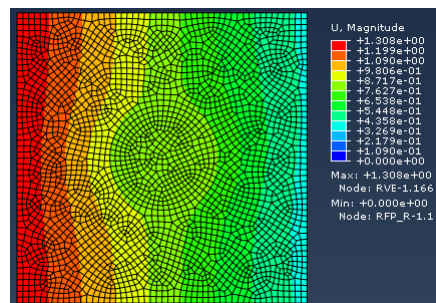
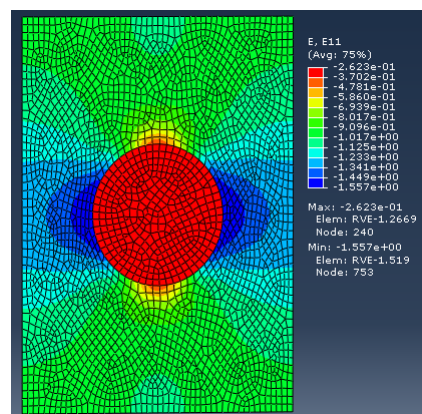
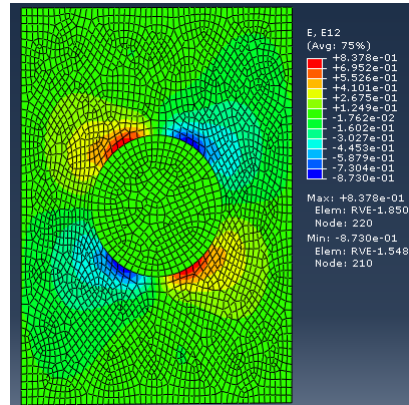
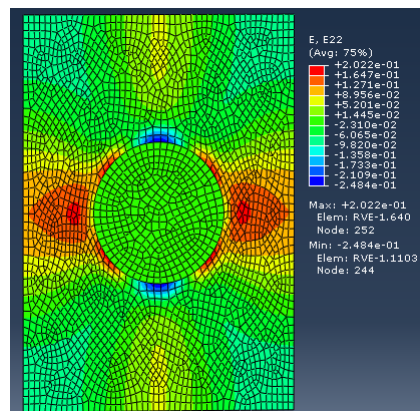


FIGURE 4.5 – Déplacement magnitude de la pièce dans le forme qui n'est pas déformé

Dans la figure (4.4) et la figure (4.5), on trouve que le déplacement maximal est 1.38, qui est dans la zone rouge. On trouve que le déplacement maximal est situé dans la frontière gauche. Ensuite on met les figure de la déformation ε_{11} , ε_{12} , ε_{22} au-dessous :

FIGURE 4.6 – ε_{11} dans la forme déformé sous le premier condition

FIGURE 4.7 – ε_{12} dans la forme déformé sous le premier conditionFIGURE 4.8 – Contrainte $E2\varepsilon_{22}$ dans la forme déformé sous le premier condition

Maintenant, on a trouver contrainte E11 pour tous les rectangles du maillage, On sort une fichier «deformation.rpt» de la deformation ey une autre fichier «volume.rpt», et on écrit une autre code de python [9] pour calculer le moyenne de déformation pour tous les pixels pour la première hypothèse et on trouve :

```
In [11]:
runfile('C:/Users/xzhang/Stage_LAMCOS/Abaqus_methode1/def_vol
ume.py', wdir='C:/Users/xzhang/Stage_LAMCOS/Abaqus_methode1')
-2.31941207406e-07 2.52567298642e-11 -2.31941207406e-07
```

FIGURE 4.9 – Déformation moyenne pour première hypothèse sur Abaqus

On aussi met la figure de S.Mise pour afficher les résultats :

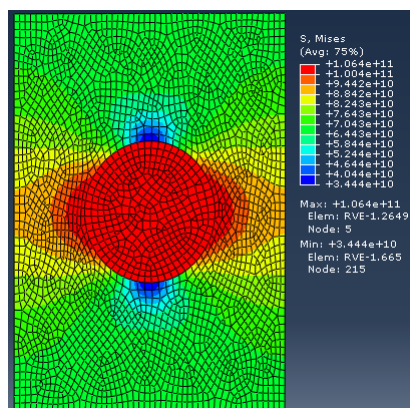


FIGURE 4.10 – S.Mises dans le forme déformé

Chapitre 5

Comparaison de deux méthodes de mesure

Dans ce chapitre, on va comparer les résultats obtenus par les deux méthodes.

Pour la première méthode FFT, je trouve le résultat, mais la déformation n'est pas correcte comparée avec la solution numérique, le code du FFT a besoin d'être amélioré. Mais c'est une bonne méthode pour traiter les milieux composites. Pour la deuxième méthode FEM, on fait la simulation sur le logiciel Abaqus, on modifie le fichier *.inp pour ajouter les conditions aux limites. Mais malheureusement, je ne trouve pas un bon résultat.

L'avantage de FFT n'est pas affiché dans ce stage, parce qu'il y a seulement deux matériaux, l'aluminium et la fibre, dans le milieu composite de stage. Et les milieux sont bien dans une forme géométrique. Donc, pour les deux méthodes, le temps de calcul n'est pas très long. Si il y a un milieu composite qui a beaucoup de milieux différents et les milieux sont dérangés, je pense la méthode de FFT va prendre moins de temps pour calculer les quantités de la matière.

La méthode FEM peut présenter les résultats de simulation visuellement. C'est aussi une raison pour laquelle on utilise la méthode des éléments généralement. Une autre raison, FFT est limitée à traiter les milieux composites périodiques.

Chapitre 6

Conclusion du stage

J'ai effectué mon stage de fin d'étude de Master 2 en modélisation et application en mécanique au sein du laboratoire de mécanique des contacts et des structures. Lors de ce stage de 6 mois, j'ai pu mettre en pratique mes connaissances théoriques acquises durant ma formation, de plus, ce stage a été l'opportunité pour moi d'appréhender milieux composites.

Mon stage au LaMCoS est situé à l'INSA-Lyon à côté de la campus du Lyon1. Mon tuteur de stage étant chercheur CNRS et maître de conférence INSA, j'ai pu apprendre le sujet de stage. Ce stage est consacrée au développement d'outils numériques basés sur la base approche FFT en vue de la détermination des propriétés effectives des structures périodiques.

Pour le sujet, il existe de défaut à améliorer. Je n'ai pas trouvé la solution du code exact, donc le résultat du déformation homogénéisation n'est pas l'égal de 1. Et je n'ai pas assez de temps de faire la simulation en 3D. J'enlève tous les codes de stage dans les annexes.

Finalement, c'est une bonne expérience de stage, je vais bénéficier beaucoup de ce stage quand je travail ou continue ma étude de doctorat à l'avenir.

Bibliographie

- [1] LaMCoS. Présentation. http://lamcos.insa-lyon.fr/files/docs/diaporama_lamcos_2018_fr.pdf.
- [2] I.M.A.C. TF2D. http://tf2d.free.fr/index.php?cours=021_Introduction.
- [3] Thierry Bretheau et Pierre Gilormini Michel Bornert. Homohénéisation en mécanique des matériaux 1. *Hermès Science Publications*, pages 74–78, 2001.
- [4] Thierry Bretheau et Pierre Gilormini Michel Bornert. Homohénéisation en mécanique des matériaux 1. *Hermès Science Publications*, pages 49–50, 2001.
- [5] Wiki. DTF. https://fr.wikipedia.org/wiki/Transformation_de_Fourier_discr%C3%A8te.
- [6] Wiki. FEM. https://fr.wikipedia.org/wiki/M%C3%A9thode_des_%C3%A9l%C3%A9ments_finis.
- [7] DS simulia. AbaqusDoc. <http://abaqusdoc.insa-lyon.fr:2080/v6.14/index.html>.
- [8] Xunjie. Github. <https://github.com/ZHANGxunjie/cours/blob/master/Job-2308.inp>.
- [9] Xunjie. Github. <https://github.com/ZHANGxunjie/cours/tree/master/stage%20LaMCoS>.

Annexe A

Simulations et programmations

A.1 Code Matlab

Interdit

A.2 Code abaqus

Interdit

A.3 Code Python

Code calcul E11 moyenne avec les fichiers *.rpt

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Jul 25 14:24:26 2018
4
5  @author: xzhang
6  """
7  import numpy as np
8  f = open("C:/Temp/strain.rpt", 'r')
9  for i in range(19):
10     a = f.readline()
11     E11 = []
12     a = f.readline().split()
13
14     while a != []:
15         b = list(map(eval, a))
16         E11.append(b[3])
17         a = f.readline().split()
18
19     for i in range(15):
20         a = f.readline()
21
22     a = f.readline().split()
23
24     while a != []:
25         b = list(map(eval, a))
26         E11.append(b[3])
27         a = f.readline().split()
28
29     f.close()
30     f = open("C:/Temp/strain.rpt", 'r')
31     for i in range(19):
```

```
32     a = f.readline()
33     E12 = []
34     a = f.readline().split()
35
36     while a != []:
37         b = list(map(eval,a))
38         E12.append(b[2])
39         a = f.readline().split()
40
41     for i in range(15):
42         a = f.readline()
43
44     a = f.readline().split()
45
46     while a != []:
47         b = list(map(eval,a))
48         E12.append(b[2])
49         a = f.readline().split()
50
51     f.close()
52     f = open("C:/Temp/strain.rpt",'r')
53     for i in range(19):
54         a = f.readline()
55         E22 = []
56         a = f.readline().split()
57
58         while a != []:
59             b = list(map(eval,a))
60             E22.append(b[3])
61             a = f.readline().split()
62
63         for i in range(15):
64             a = f.readline()
65
66         a = f.readline().split()
67
68         while a != []:
69             b = list(map(eval,a))
70             E22.append(b[3])
71             a = f.readline().split()
72
73     f.close()
74     #E11 = np.asarray(E11)
75
76     f = open("C:/Temp/volume.rpt",'r')
77     for i in range(19):
78         a = f.readline()
79         volume = []
80         a = f.readline().split()
81
82         while a != []:
83             b = list(map(eval,a))
84             volume.append(b[1])
85             a = f.readline().split()
86
87         for i in range(15):
88             a = f.readline()
89
90         a = f.readline().split()
91
92         while a != []:
93             b = list(map(eval,a))
94             volume.append(b[1])
95             a = f.readline().split()
96
97     f.close()
98     somme1 = np.dot(volume,E11)
```

```
99 somme2 = np.dot(volume, E12)
100 somme3 = np.dot(volume, E22)
101 E11_moy = somme/np.linalg.norm(volume)
102 print(somme1, somme2, somme3)
```