Table des matières

Table des figures

Liste des tableaux

PRÉFACE

Ce stage a pour objet la simulation numérique des déformations dans un foie humain soumis à des déplacements imposés sur sa frontière. De telles simulations ont déja été effectuées mais le modèle considéré était un modèle linéaire qui ne donne plus de solutions valables lorsqu'on est en grands déplacements. On se propose d'effectuer des calculs avec un modèle non linéaire. Il sera alors possible de comparer les résultats et donc de déterminer le domaine de validité du modèle linéaire.

On divise le rapport en trois parties. Dans la première partie, on rappèlera quelques résultats théoriques de mécanique des milieux continus et on fixera les notations utilisées par la suite. On passera également en revue les outils utilisés pour traiter l'incompressibilité et la non linéarité du problème. On utilisera une méthode itérative pour résoudre le problème non linéaire. La méthode iterative la plus simple est celle de Newton mais il est possible d'en utiliser d'autres plus élaborées. On a par exemple la méthode de Newton avec incrémentation du chargement ou encore la méthode de continuation.

Dans la seconde partie, on montre les résultats de plusieurs calculs. On commence tout d'abord par tester le programme d'élasticité non linéaire sur un exemple simple dont on connait la solution analytique. On utilise dans ce test deux types d'éléments finis. Le premier est un hexaèdre Q2P1[1] et le second un tétraèdre P2P0[2]. Lors de la modèlisation du foie humain, on utilisera l'élément fini tétraédrique car on a alors des mailleurs automatiques qui permettent de mailler le volume d'un objet à partir d'un maillage sur la peau de cet objet. On passe ensuite aux tests sur le foie proprement dit. Pour obtenir des déformations en temps raisonnable (moins d'une semaine !!!), on utilisera un algorithme de calcul en décomposition de domaines.

Dans la troisième et dernière partie, on donne un aperçu d'une méthode de calcul très rapide permettant d'avoir des déformations en temps réel sur le modèle du foie. Cette méthode, appellée méthode d'analyse modale, permet de trouver rapidement des déplacements pour le modèle linéaire. On n'obtient malheureusement pas de solution exacte en déplacement mais seulement une

1. Q2 représente l'espace d'approximation de chacune des composantes en déplacement et P1 l'espace de projection du gradient cf section 1.4

2. P2 représente l'espace d'approximation de chacune des composantes en déplacement et P0 l'espace de projection du gradient cf section 1.4

approximation de cette solution. On espère cependant obtenir une solution suffisamment bonne pour rester dans un cadre réaliste et donc utiliser cette méthode avec succés pour la simulation en temps réel.

Chapitre 1

Préliminaires théoriques

1.1 Introduction

On donne ici quelques résultats généraux de mécanique des milieux continus et on fixe les notations qui seront utilisées par la suite. On étudie également les outils utilisés pour traiter numériquement l'incompressibilité et la non linéarité du problème à résoudre.

1.2 Résultats généraux

1.2.1 Le champ de déformations

On veut étudier les déformations dans un solide soumis à différentes forces ou déplacements imposés. On assimile donc le solide étudié à un ouvert borné de $\mathbb{R}^3$ qu'on nomme Ω. On définit un champ de déformations φ sur $\overline{\Omega}$ donnant une correspondance entre un point quelconque du solide dans sa configuration naturelle et l'emplacement du même point matériel dans sa configuration déformée :

$$\begin{array}{rccc} \varphi : & \overline{\Omega} & \to & \mathbb{R}^3 \\ & x & \to & \varphi(x) \end{array}$$

Ce champ vectoriel possède plusieurs propriétés naturelles que l'on traduit en termes mathématiques par les hypothèses suivantes :

$$\begin{cases} \varphi \in C^1(\overline{\Omega};\mathbb{R}^3) \\ \\ \varphi\; injective\; sur\; \Omega \\ \\ det(\nabla\varphi) > 0\; sur\; \overline{\Omega} \\ \\ (\varphi\; ne\; modifie\; pas\; l'orientation.) \end{cases}$$

On pose alors $F = \nabla\varphi$ et $J = det(F)$.

1.2.2 Les équations d'équilibre

On se place dans le cadre de l'élasticité non linéaire. On a donc à résoudre les équations aux derivées partielles suivantes :

$$(1.1) \qquad \begin{cases} -divT(x) = f(x), x \in \Omega \\ \\ T(x)\nabla\varphi(x)^T = \nabla\varphi(x)T(x)^T \\ \\ T(a).n = g(a)\; sur\; \Gamma_N \end{cases}$$

T s'appelle le premier tenseur de Piola Kirchoff. T n'est pas symétrique. En effet :

$$T(x) = (det\nabla\varphi(x))T^{\varphi}(\varphi(x))[\nabla\varphi(x)]^{-T}$$

mais

$$T^T(x) = (det\nabla\varphi(x))[\nabla\varphi(x)]^{-1}T^{\varphi}(\varphi(x))$$

On préfére donc souvent résoudre le système équivalent suivant :

$$(1.2) \qquad \begin{cases} -div\nabla\varphi(x)\Sigma(x) = f(x), x \in \Omega \\ \\ \Sigma(x) = \Sigma(x)^T \\ \\ \nabla\varphi(x)\Sigma(x).n = g(a)\; sur\; \Gamma_N \end{cases}$$

avec :

$$\Sigma(x) = [\nabla\varphi(x)]^{-1}T(x)$$

Ce tenseur est alors symétrique et s'appelle le second tenseur de Piola Kirchoff. On a utilisé ici les notation de [1] où tout ceci est expliqué avec précision.

1.2.3 Lois de comportement

Une loi de comportement permet, d'une manière générale, de relier les contraintes aux déplacements. Lorsque le matériau étudié possède des propriétées particulières, on peut alors exprimer les contraintes avec moins d'informations que l'ensemble des déplacements. On considérera toujours par la suite des matériaux hyperélastiques isotropes. On donne alors la définition de l'hyperélasticité et les proprietés de tels matériaux.

Définition : Un matériau est dit hyperélastique si $\exists \hat{W} : \overline{\Omega} \times M^3_+ \rightarrow \mathbb{R}$ tel que :

$$\hat{T}(x, F) = \frac{\partial \hat{W}}{\partial F}(x, F)$$

Lorsque le matériau est hyperélastique, on a alors le

Théorème 1 *Pour un materiau hyperélastique isotrope, le potentiel élastique est de la forme* $\hat{W}(x, F) = \tilde{W}(x, i_{F^TF})$.

où i_{F^TF} désigne les invariants de $C = F^T F$. On definit alors les invariants de C de la manière suivante :

$$\begin{aligned}
I_1(C) &= Tr(C) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\
I_2(C) &= \tfrac{1}{2}\{(Tr(C))^2 - Tr(C^2)\} \\
&= Tr(cof(C) = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \\
I_3(C) &= det(C) = (det F)^2 = \lambda_1^2\lambda_2^2\lambda_3^2 = J^2
\end{aligned}$$

1.3 Les méthodes itératives pour la résolution de $F(x) = 0$

1.3.1 La méthode de Newton

La méthode de Newton-Raphson remplace l'équation non lineaire $F(x) = 0$ par son développement au premier ordre autour du point X^n :

$$F(X^n) - \frac{DF(X^n)}{Dx}(X^{n+1} - X^n) = 0$$

Sous certaines conditions on a alors :

$$X^n \to X$$

avec X vérifiant F(X)=0. C'est l'objet du théorème suivant :

Théorème 2 *Soit X une solution de $F(x) = 0$, avec $\frac{DF(x)}{Dx}$ inversible et lipschitzienne dans un voisinage de X. Alors, si X^0 est suffisamment proche de X, nous avons*

$$\|X^{n+1} - X\| \leq \beta \|X^n - X\|^2$$

On mesure alors bien l'importance du choix de X^0. Dans de nombreux cas, X^0 est trop éloigné de X pour que l'on puisse rentrer dans le cadre des hypothèses du théorème. Il n'y a alors plus rien qui garantisse la convergence de la méthode. Suivant les cas, on a alors soit une divergence de la méthode de Newton soit une convergence fortuite. On donnera par la suite des exemples pour mettre en évidence ces situations.

1.3.2 La méthode de Newton avec chargement incrémental

La méthode de Newton avec chargement incrémental est une amélioration très simple de la méthode de Newton. Le principe de la méthode est de choisir un paramètre $\lambda_0 \in (0, 1]$ et de résoudre le problème avec :

- $u_{\lambda_0}(x) = \lambda_0 u(x)\ \forall x \in \Gamma_{Dirichlet}$, les conditions de Dirichlet sur le bord.
- $f_{\lambda_0}(x) = \lambda_0 f(x)\ \forall x \in \Omega$, les forces volumiques.
- $g_{\lambda_0}(x) = \lambda_0 g(x)\ \forall x \in \Gamma_{Neumann}$, les forces surfaciques.

On recommence ensuite la même opération en choisissant λ_1 tel que $\lambda_0 < \lambda_1 \leq 1$ et en partant de la configuration déformée obtenue à l'etape 1. On réitére ensuite le processus n fois jusqu'a ce que $\lambda_n = 1$ ce qui donne alors une approximation de la solution du problème étudié.

1.3.3 La méthode de continuation

La méthode de Newton avec chargement incrémental, bien que d'usage courant, possède quelques inconvenients. On peut donc améliorer cette méthode en prenant $\Delta\lambda$ qui vérifie

$$\|\Delta U\|^2 + \|\Delta P\|^2 + \|\Delta\lambda\|^2 = constante$$

On obtient alors une méthode robuste, appelé méthode de continuation, qui calcule automatiquement le chargement incrémental. Pour plus de details sur ces méthodes, on peut se référer à [2].

1.4 Approximation de problèmes incompressibles

On considére un matériau dont le potentiel élastique est de la forme :

$$\begin{aligned} W(F) &= W_{inc}(x, (detF)^{-1/3}F) + \tfrac{1}{2\epsilon}(detF - 1)^2 \\ &= W_0(x, F) + \tfrac{1}{2\epsilon}(detF - 1)^2 \end{aligned}$$

Pour tout $\epsilon > 0$, le materiau considèré est compressible. Sa formulation variationelle s'ecrit alors :

$$\left\{ \begin{array}{c} \text{Trouver } u \in V + u_0 \text{ tel que :} \\ \\ \int_\Omega \frac{\partial W_0}{\partial F}(x, Id + \nabla u) : \nabla v dx \\ \\ + \int_\Omega \frac{1}{\epsilon}[det(Id + \nabla u) - 1]\frac{\partial det}{\partial F}(x, Id + \nabla u) : \nabla v dx = \\ \\ \int_\Omega f.v dx + \int_{\Gamma_1} g.v da, \forall v \in V \end{array} \right.$$

On ne peut pas résoudre le problème uniquement en remplacant V par un espace d'approximation de dimension fini.

On introduit alors V_h et P_h vérifiant la condition suivante (qui provient du théorème de Ladyzenskaya-Babuska-Brezzi) :

$$\inf_{q_h \in \mathbb{P}_h, \|q_h\|=1} \left\{ \sup_{v_h \in \mathbb{V}_h, \|v_h\|=1} \left\{ \int_\Omega q_h \frac{\partial det}{\partial F}(Id + \nabla u_h) : \nabla v_h dx \right\} \right\} = C_0(h) > 0$$

On pose alors :

$$\nabla_h v = \text{ la projection de } \nabla v \text{ sur } (P_h)^{N\times N}$$

et on definit le problème approché comme suit :

$$\begin{cases} \text{Trouver } u_h \in V_h + u_0 \text{ tel que :} \\ \int_\Omega \frac{\partial W_0}{\partial F}(x, Id + \nabla u_h) : \nabla v_h dx \\ + \int_\Omega \frac{1}{\epsilon}[det(Id + \nabla_h u_h) - 1]\frac{\partial det}{\partial F}(x, Id + \nabla_h u_h) : \nabla_h v_h dx = \\ \int_\Omega f.v_h dx + \int_{\Gamma_1} g.v_h da, \forall v_h \in V_h \end{cases}$$

Cette formulation variationnelle permet alors de résoudre le problème et c'est cette formulation qui est utilisée dans le programme.

En pratique, on prendra :

$$V_h = \{v; v \in C^0(\overline{\Omega}); v_{|\tau_i} \in \mathbb{Q}_2\}$$

et

$$P_h = \{v; v_{|\tau_i} \in \mathbb{P}_1\}$$

si τ_i une hexaédrisation de Ω ou bien encore

$$V_h = \{v; v \in C^0(\overline{\Omega}); v_{|\tau_i} \in \mathbb{P}_2\}$$

et

$$P_h = \{v; v_{|\tau_i} \in \mathbb{P}_0\}$$

si τ_i une tetraédrisation de Ω.

Pour des approfondissements sur cette méthode, on peut se référer à [2]. Pour un exposé sur les formulations variationnelles mixtes qui est une autre approche qui permet de résoudre des problèmes similaires, se référer à [3].

Chapitre 2

Résultats numériques

2.1 Validation du programme d'élasticité non linéaire sur un cas simple : le cube de Rivlin

2.1.1 Étude théorique du cube de Rivlin

Introduction

L'étude des contraintes et des déplacements dans un solide tridimentionnel en élasticité non linéaire laisse supposer des calculs longs et nécessitant beaucoup de mémoire. On teste donc le programme d'élasticité non linéaire sur un exemple simple dont on connait la solution analytique. On présente tout d'abord les lois de comportement utilisées dans le programme. On introduira ensuite le problème du cube de Rivlin.

Lois de comportement

Le programme permet le choix de deux lois de comportement pour le solide qui sont données par deux potentiels élastiques. On donne ci dessous les potentiels élastiques de ces deux lois.

Matériau d'Odgen Le premier type de matériau a une loi qui a été introduite par Ogden. L'expression de son potentiel élastique associé est :

$$\hat{W}(F) = C_1(I_1 - 3) + C_2(I_2 - 3) + a(J^2 - 1) - (2C_1 + 4C_2 + 2a)\ln J$$

C'est un matériau compressible hyperélastique isotrope.

Matériau de Mooney-Rivlin Le second type de matériau sera celui utilisé lors de la modélisation du foie humain. C'est un matériau de type Mooney-Rivlin dont le potentiel élastique est donné par :

$$\hat{W}(F) = C_1(I_1 - 3) + C_2(I_2 - 3)$$

C'est un matériau incompressible hyperélastique isotrope. Numériquement, on traite le problème par pénalisation. On rajoute le terme $a(J-1)^2$ à la partie incompressible et on résoud le problème avec la loi suivante :

$$\hat{W}_\epsilon(F) = \hat{W}(F) + a(J-1)^2$$

C'est cette expression du potentiel élastique qui est utilisé dans le programme et qui fait alors intervenir trois coefficients. Pour obtenir un matériau incompressible, on donne une valeur très grande à a.

Cas test suivant la loi de comportement

Positionnement du problème Soit $\Omega = \{-1 < x < 1, -1 < y < 1, -1 < z < 1\}$. On considère un milieu continu occupant l'ouvert Ω. On impose des forces surfaciques (conditions aux limites de Neumann) sur la totalité de la surface de l'ouvert. On impose les forces de manière à avoir des solutions analytiques pour les déplacements. Par raison de symétrie, le problème se ramène à un problème posé sur un huitième du cube. Pour certaines lois de comportement et pour certaines forces bien choisies, on à alors des solutions analytiques en déplacements très simples. C'est l'objet des deux paragraphes suivants.

Le cas compressible On considère le problème du cube de Rivlin exposé au paragraphe précédent. On suppose qu'on a un matériau d'Ogden.

On pose :

$$b = 2C_1 + 4C_2 + 2a$$

et

$$C_3 = 2a - \frac{b}{(\lambda_1+1)^2(\lambda_2+1)^2(\lambda_3+1)^2}$$

On impose alors les forces surfaciques suivantes :

- Sur la face $x = 1$, $g(x, y, z) = \begin{pmatrix} \sigma_{11} \\ 0 \\ 0 \end{pmatrix}$

avec :

$$\sigma_{11} = 2C_1(\lambda_1 + 1) + 2C_2(\lambda_1 + 1)[(\lambda_2 + 1)^2 + (\lambda_3 + 1)^2] + 2C_3(\lambda_2 + 1)^2(\lambda_3 + 1)^2$$

- Sur la face $y = 1$, $g(x, y, z) = \begin{pmatrix} 0 \\ \sigma_{22} \\ 0 \end{pmatrix}$

avec :

$$\sigma_{33} = 2C_1(\lambda_2 + 1) + 2C_2(\lambda_2 + 1)[(\lambda_3 + 1)^2 + (\lambda_1 + 1)^2] + 2C_3(\lambda_3 + 1)^2(\lambda_1 + 1)^2$$

- Sur la face $z = 1$, $g(x, y, z) = \begin{pmatrix} 0 \\ 0 \\ \sigma_{33} \end{pmatrix}$

avec :

$$\sigma_{33} = 2C_1(\lambda_3 + 1) + 2C_2(\lambda_3 + 1)[(\lambda_1 + 1)^2 + (\lambda_2 + 1)^2] + 2C_3(\lambda_1 + 1)^2(\lambda_2 + 1)^2$$

On a alors le champ de déplacement suivant :

$$u(x, y, z) = \begin{pmatrix} \lambda_1 x \\ \lambda_2 y \\ \lambda_3 z \end{pmatrix}$$

Le cas incompressible On considére toujours le problème du cube de Rivlin mais constitué d'un matériau de type Mooney-Rivlin.

On donne les forces surfaciques suivantes :

- Sur la face $x = 1$, $g(x, y, z) = \begin{pmatrix} \sigma_{11} \\ 0 \\ 0 \end{pmatrix}$

avec :

$$\sigma_{11} = 2C_1(\lambda_1 + 1) + 2C_2(\lambda_1 + 1)[(\lambda_2 + 1)^2 + (\lambda_3 + 1)^2] - p(\lambda_2 + 1)^2(\lambda_3 + 1)^2$$

- Sur la face $y = 1$, $g(x, y, z) = \begin{pmatrix} 0 \\ \sigma_{22} \\ 0 \end{pmatrix}$

avec :

$$\sigma_{33} = 2C_1(\lambda_2+1) + 2C_2(\lambda_2+1)[(\lambda_3+1)^2 + (\lambda_1+1)^2] - p(\lambda_3+1)^2(\lambda_1+1)^2$$

- Sur la face $z = 1$, $g(x, y, z) = \begin{pmatrix} 0 \\ 0 \\ \sigma_{33} \end{pmatrix}$

avec :

$$\sigma_{33} = 2C_1(\lambda_3+1) + 2C_2(\lambda_3+1)[(\lambda_1+1)^2 + (\lambda_2+1)^2] - p(\lambda_1+1)^2(\lambda_2+1)^2$$

Si on prend $p = 1$, et si $(1+\lambda_1)(1+\lambda_2)(1+\lambda_3) = 1$, on obtient alors le champ de déplacement suivant :

$$u(x, y, z) = \begin{pmatrix} \lambda_1 x \\ \lambda_2 y \\ \lambda_3 z \end{pmatrix}$$

2.1.2 Résultats numériques

Description et données du problème

On utilise les résultats du chapitre précédent pour effectuer des tests sur le cube de Rivlin. On ne montre dans cette partie que les résultats obtenus lorsqu'on considère un matériau de Mooney-Rivlin. Comme on l'a expliqué au paragraphe précédent, pour des raisons de symétrie, on ne maille qu'un huitième du cube unité. On maille donc le domaine $D = \{0 < x < 1; 0 < y < 1; 0 < z < 1\}$ de deux manières différentes :

- Avec un élément fini hexaédrique nommé HEXA 3QCC dans le programme qui est un élément fini Q2P1. On maille alors le domaine avec un seul élément de ce type.

- Avec un élément fini tétraédrique nommé TETR 3P20 dans le programme qui est un élément fini P2P0. Le domaine est alors maillé avec six éléments de ce type.

Le matériau dont on modélise le comportement est un matériau de type Mooney-Rivlin. On utilise en pratique la formule suivante pour modéliser un matériau de ce type :

$$\hat{W}_\epsilon(F) = C_1(I_1 - 3) + C_2(I_2 - 3) + a(J-1)^2$$

ou a est le terme de pénalisation. On donne certaines valeurs aux coefficients C_1, C_2 et a suivant les caractéristiques du matériau considéré et la valeur du terme de pénalisation :

$$C_1 = 0.5 \; C_2 = 0.0056 \; a = C_1/\epsilon$$

ou ϵ reprèsente un réel compris entre 1 et 10^{-9}.

On a également introduit les paramètres p, λ_1, λ_2 et λ_3 à la section précédante, on les fixe maintenant :

$$p = 1, \lambda_1 = .1, \lambda_2 = .2, \lambda_3 = -.242424$$

On verifie bien que $(1+\lambda_1)(1+\lambda_2)(1+\lambda_3) = 1$. En fixant ces paramètres, on a alors :

$$u(x,y,z) = \begin{pmatrix} x/10 \\ y/5 \\ -0.242424z \end{pmatrix}$$

On utilise ici une méthode de pénalisation pour traiter le problème. On va observer ce qui se passe lorsqu'on résoud le problème du cube de Rivlin avec une valeur de la pénalisation fixée. On regardera ensuite l'influence qu'a cette valeur sur le déroulement du calcul (en terme de nombre d'itérations et de précision du résultat) en la faisant varier. On considérera suivant les cas soit le maillage avec un hexaèdre soit le maillage avec six tétraèdres.

Essai avec une valeur de la pénalisation fixe.

On donne ci-dessous le listing de l'exécution du programme avec le domaine maillé en tétraèdres et avec les forces surfaciques données lors de l'étude théorique. On fixe ici la valeur de ϵ a 10^{-5}. On obtient alors l'éxécution suivante.

****

```
  ntdl  81
  fin de newt0
 uini  .0
 bfix avant c.l  .3502395400772638
```

```
bfix  .3502395400772638
      ernorr  .3502395400772638
      enors    1.15998544370659
 ITERATION :  1  erreur  .6425834294891524
      ernorr  2814.682109210903
      enors    .1430613010781453
 ITERATION :  2  erreur  2678.857355475351
      ernorr  217.9407231124705
      enors    9.326298925343790E-03
 ITERATION :  3  erreur  207.4233147168479
      ernorr  1.01298816819507
      enors    .2552713025389071
 ITERATION :  4  erreur  1.03744555547418
      ernorr  155.3275535713755
      enors    7.611029111790192E-03
 ITERATION :  5  erreur  147.8320157387349
      ernorr  .558647955427861
      enors    .229854610413705
 ITERATION :  6  erreur  .5977327929516656
      ernorr  130.9218939747092
      enors    5.543148141473436E-03
 ITERATION :  7  erreur  124.6038331490538
      ernorr  .367992089628649
      enors    .1013105756760768
 ITERATION :  8  erreur  .3793415597216164
      ernorr  25.11880760258317
      enors    2.041203467433407E-03
 ITERATION :  9  erreur  23.90689898180423
      ernorr  1.392738855541414E-02
      enors    2.089980848600118E-02
 ITERATION :  10  erreur  1.926087523511999E-02
      ernorr  1.07391220226303
      enors    9.799221237393880E-05
 ITERATION :  11  erreur  1.02210217731612
      ernorr  2.513224671878437E-05
      enors    6.829508511113281E-05
 ITERATION :  12  erreur  4.354437355175138E-05
      ernorr  1.146124329600029E-05
```

```
      enors    6.800977430947205E-10
 ITERATION :  13  erreur  1.090819979795743E-05
      ernorr  1.131788149686315E-11
      enors    5.108099226919569E-15
 ITERATION :  14  erreur  1.077303056205317E-11
LE SYSTEME CONVERGE APRES   14 ITERATIONS, RESIDU :  .1077D-10
FIN DU MODULE NEWTON

*****************************************************************
```

On a ici :

- ernorr $= \|A^n X^{n+1} - A^n X^n\|$
- enors $= \|X^{n+1} - X^n\|$

On veut que ces deux valeurs soient petites pour décider de la convergence de la méthode. On introduit donc une autre variable qui tient compte de cette remarque. La variable erreur est une combinaison des deux variables ernorr et enors. C'est cette variable qui décide de la convergence de l'algorithme.

L'impression de l'exécution montre que le résidu n'est pas strictement décroissant. Or la vitesse de convergence de la méthode de Newton est quadratique lorsqu'on est proche de la solution. Ceci montre que la convergence vers la solution est fortuite car les forces exercées font qu'on ne commence pas dans un voisinage de la solution.

Pour ce test, on obtient alors la déformée du cube de Rivlin affichée figure 2.1.

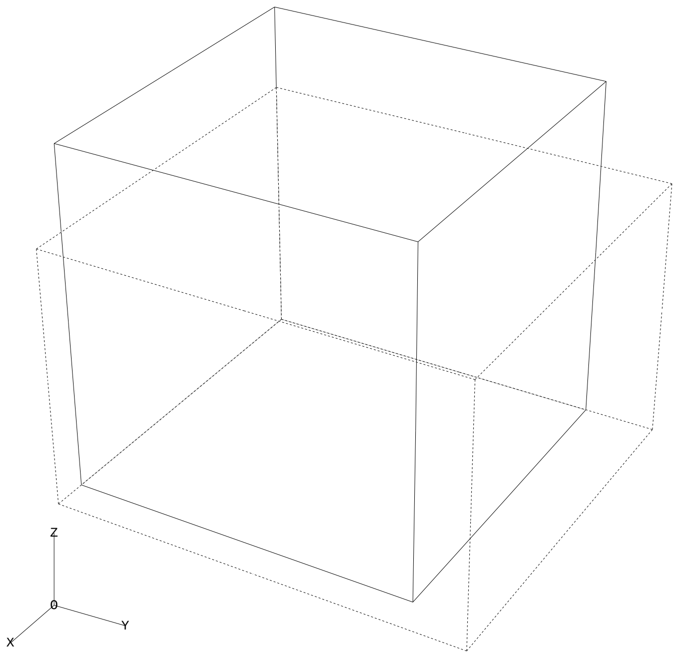

FIG. 2.1 – *Tracé de la déformée du cube de Rivlin*

Étude de l'influence de la valeur de la pénalisation sur la vitesse de convergence

On fait maintenant varier la valeur de la pénalisation. On choisit alors des valeurs pour ϵ comprises entre 1 et 10^{-9}. Il est important de faire remarquer qu'une valeur de ϵ très petite rend la résolution du systéme linéaire difficile car la matrice à inverser a alors un mauvais conditionnement. Expérimentalement, on observe ce phenomène lorsqu'on prend $\epsilon = 10^{-10}$ ou des valeurs encore plus petites. On a alors non convergence de l'algorithme de Newton pour une initialisation à zero.

On donne dans le tableau 2.1 le nombre d'itérations nécessaires à la convergence de l'algorithme de Newton en fonction de le valeur de la pénalisation pour les 2 types d'éléments finis :

Valeur de ϵ	10^{-0}	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
HEXA 3QCC	4	5	6	15	8	14	18	13	14	14
TETR 3P20	4	5	6	15	8	14	$+\infty$	13	14	14

TAB. 2.1 – *Nombre d'itérations en fonction de la valeur de ϵ pour les deux types d'éléments finis HEXA 3QCC et TETR 3P20*

La non convergence lorsqu'on utilise l'élément fini TETR 3P20 et lorsque $\epsilon = 10^{-6}$ corrobore les conclusions du paragraphe précédant sur la convergence de l'algorithme de Newton lorsqu'on est pas dans un voisinage de la solution. De même que précédemment, on impose des forces trop importantes pour être dans un voisinage de la solution. Cette fois-ci on n'a alors pas convergence de l'algorithme de Newton. On doit alors utiliser une méthode de Newton avec chargement incrémental ou encore une méthode de continuation si on veut converger. Expérimentalement, on constate que si on utilise une méthode de continuation en choisissant le λ initial (cf 1.3.2) à 0.4, on a alors besoin de 23 itérations pour converger.

Étude de l'influence de la valeur de la pénalisation sur la précision du résultat du calcul

On donne la courbe donnant la norme l^2 relative de l'erreur en fonction de la valeur de ϵ.

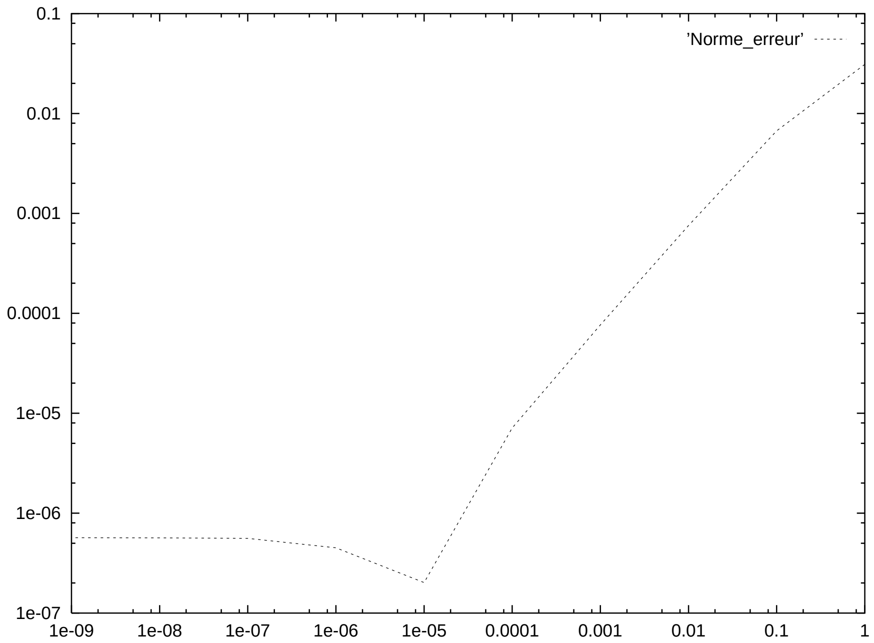

FIG. 2.2 – *Norme l^2 relative de l'erreur en fonction de ϵ*

Cette courbe fait apparaitre une très forte décroissance de la norme quand la valeur de ϵ tend vers zero mais cette décroissance stoppe quand $\epsilon = 10^{-5}$. On voit qu'on obtient une très bonne approximation d'un matériau incompressible lorsque on choisit $\epsilon = 10^{-5}$ ou une valeur encore plus petite.

2.2 Étude du foie soumis à des déplacements imposés

2.2.1 Présentation générale

Le foie en général

Le foie est un organe situé au dessous du diaphragme qui le sépare des poumons et du coeur. Son poid est de 1500 grammes sur un cadavre auquel on ajoute 800 a 900 grammes de sang chez les êtres vivants. Autour du foie se trouve la capsule de glisson qui est beaucoup plus rigide que le foie.

La fonction principale du foie est de maintenir constante la quantité de glucose dans le sang (1 a 1,3 grammes par litres).

Intérêt de la simulation numérique des déformations dans un foie

La formation d'un chirurgien passe par de très nombreuses opérations chirurgicales sous la direction d'un chirurgien plus expérimenté. L'éléve ne possède au début que des connaissances théoriques et doit donc acquérir de l'expérience pratique en effectuant ces opérations. L'intervention virtuelle permet au chirurgien d'acquérir les gestes opératoires adéquats ; l'opération étant effectuée par commande manuelle à distance de l'organe cible et sous contrôle visuel sur écran. Si on veut que cette simulation lui soit bénéfique, on doit bien entendu faire en sorte que la simulation soit la plus proche possible de la réalité. Pour cela on doit verifier les deux conditions suivantes :

- Obtenir des déformations visuellement réalistes et des retours de force convenables.
- Avoir ces déformations et ces retours de force en temps réel ce qui suppose des ordinateurs très puissants et des algorithmes adaptés à ce genre de problèmes.

Si on arrive à réunir ces deux conditions, on obtient alors un outil de simulation et d'apprentissage très puissant. Dans les paragraphes qui vont suivre, on s'attache à présenter deux résultats de calculs de déformation réalistes. Ce genre de résultat demande environ quatre jour de temps de calcul lorsqu'on utilise un algorithme en décomposition de domaine. On est donc très loin d'obtenir un résultat en temps réel. Cependant, dans la dernière partie du rapport, on donne une méthode permettant d'obtenir rapidement une approximation du champ de déformations ce qui permet d'avoir des calculs de déformées en temps réel.

Le maillage du foie

On montre, figure 2.3 le maillage initial du foie. Ce maillage approche bien la géométrie d'un foie humain mais certains des triangles de la peau de l'objet sont de mauvaise qualité ce qui va influer sur la qualité du maillage volumique et donc des résultats. Il semble alors intéressant de remailler l'objet de manière à obtenir une meilleure qualité des triangles. On peut voir le résultat de cette opèration figure 2.4. Tous les calculs effectués par la suite le seront sur un maillage obtenu à partir de ces données traitées.

On cherche à effectuer une comparaison entre les résultats obtenus avec un modèle linéaire et ceux obtenus avec un modèle non linéaire. On va donc essayer d'imposer des conditions aux limites proches de celles utilisées avec le modèle linéaire. Etant donné que les conditions à imposer sont toutes de type Dirichlet, on cherche à fixer les déplacements aux mêmes endroits. Comme on utilise un nouveau maillage du foie, on cherche à faire correspondre certains nœuds du nouveau maillage avec ceux de l'ancien.

Une fois que l'on sait où imposer les conditions aux limites, on effectue les calculs proprement dits. Pour cela on peut utiliser un programme séquentiel ou un programme parallèle. Des considérations de temps de calcul feront que l'utilisation du programme parallèle s'imposera de lui même.

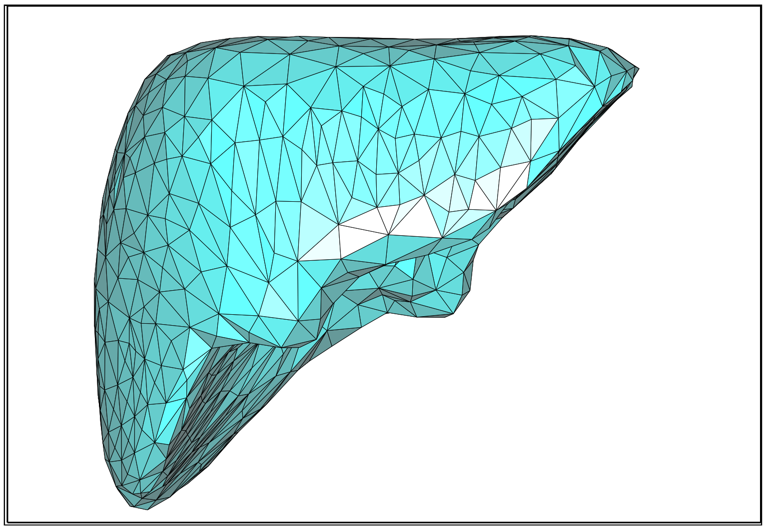

FIG. 2.3 – *Maillage brut du foie*

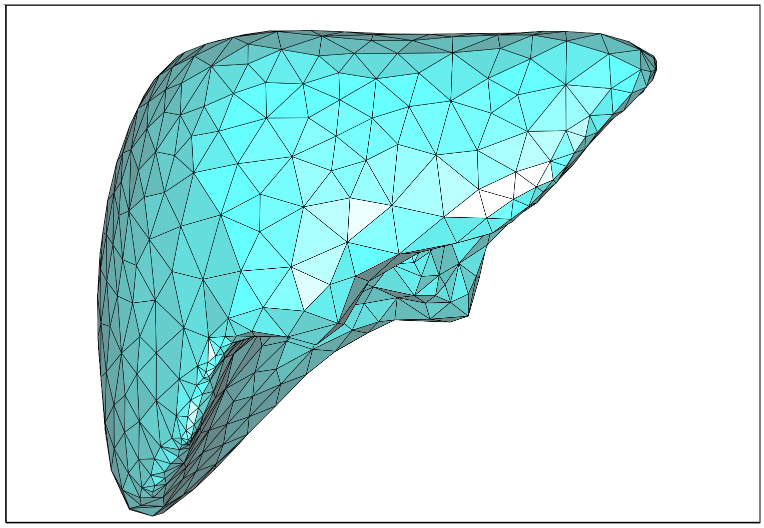

Fig. 2.4 – *Maillage du foie traité*

2.2.2 Prise en compte des conditions aux limites

Modéliser le comportement d'un foie humain est très compliqué. En effet, tous les organes du corps humain sont en interraction mutuelle. Il est donc très difficile de modéliser un d'entre eux sans tenir compte des autres. De plus, comme il a été dit en introduction, on ne peut considérer le foie que comme une approximation d'un matériau homogène isotrope puisqu'il existe une pellicule formée d'un matériau rigide sur sa surface qu'on appelle capsule de glisson. C'est pourquoi, plusieurs approches sont possibles dans la manière dont on impose des déformations. On donne ici deux approches possibles.

On utilisera souvent le mot 'Référence' dans les paragraphes qui vont suivre. Ce terme provient de la terminologie utilisée avec le logiciel d'éléments finis MODULEF. Une Référence permet de distinguer une partie de la frontière de l'objet. Ici, les nœuds qui seront à une même référence auront tous des conditions aux limites identiques.

Première méthode pour le choix des conditions aux limites

Cette première méthode consiste à reproduire les mêmes conditions aux limites que pour les tests avec le modèle linéaire. On écrit un programme qui donne une même référence pour un nœud du nouveau maillage et pour un nœud de l'ancien si la différence de leurs coordonnées est en norme plus petit que ϵ. Par la suite, on effectura tous les calculs en utilisant les références que l'on a ainsi obtenu. On affiche les résultats obtenus par cette méthode figure 2.5.

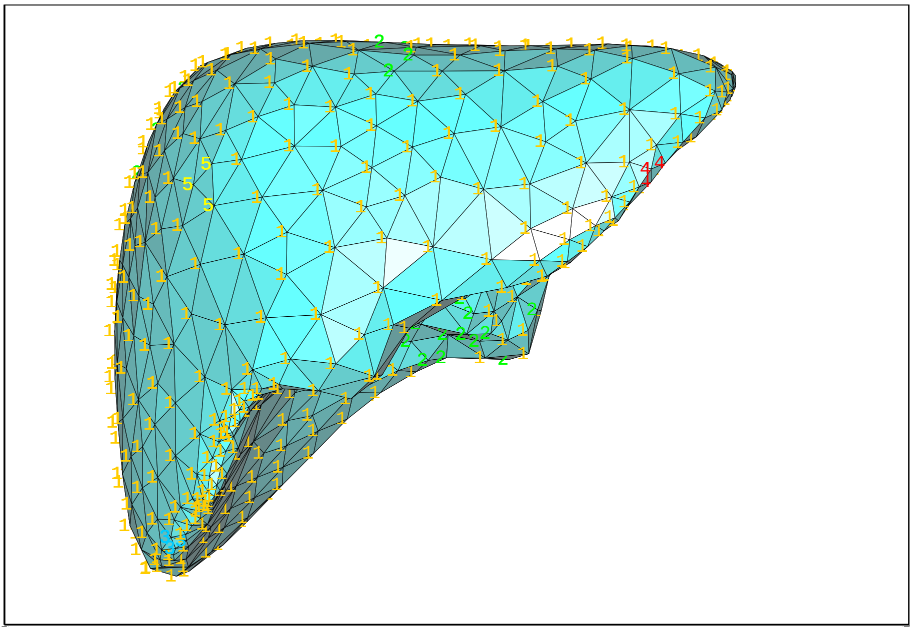

FIG. 2.5 – *Première méthode pour le choix des conditions aux limites*

Seconde méthode pour le choix des conditions aux limites

Une autre approche possible est de diviser le foie en quatres zones. Ce choix des conditions aux limites résulte des conseils d'un médecin travaillant à l'INRIA. Pour chacune des zones on affecte une référence et on lui impose les conditions suivantes :

- Pour la référence 1, les nœuds à cette référence sont libres (les conditions aux limites sont de type Neumann homogène).
- Pour la référence 2, les nœuds à cette référence sont encastrés (les conditions aux limites sont de type Dirichlet homogène).
- Pour la référence 3, les conditions aux limites sont de type non pénétration : $u.n = 0$ et $\sigma n \wedge n = 0$

– Pour la référence 4, les conditions aux limites sont de type déplacement imposé (Dirichlet non homogène).

On affiche les résultats obtenus par cette méthode figure 2.6.

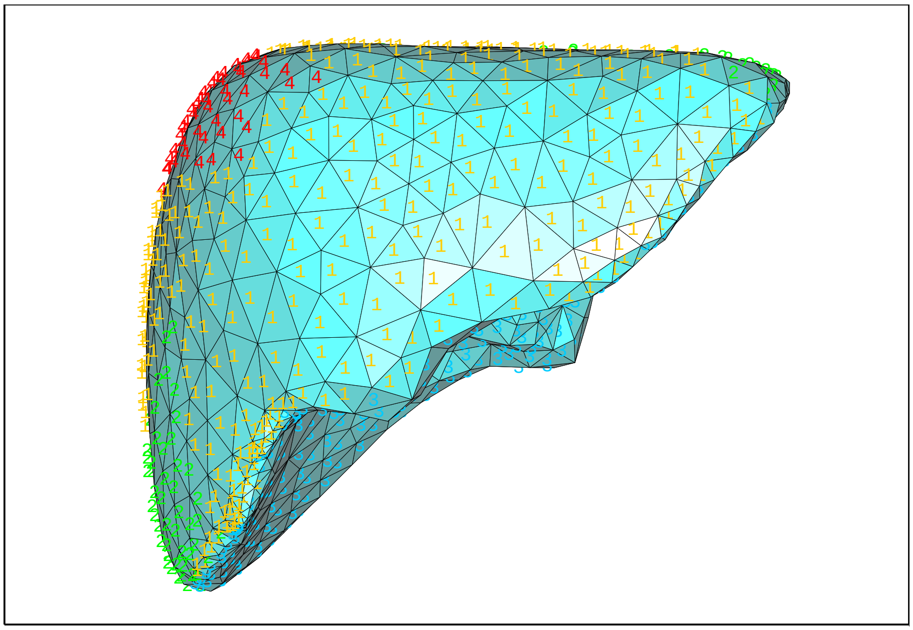

FIG. 2.6 – *Deuxiéme méthode pour le choix des conditions aux limites*

2.2.3 Calculs avec l'algorithme séquentiel

On commence par utiliser le même programme pour calculer les déformations dans le foie que lors du test sur le cube de Rivlin. On impose des déplacements nul sur de nombreux nœuds et, pour simuler l'action qu'un scalpel exerce sur le foie, on impose un déplacement non nul sur un seul nœud. Le nœud où on impose des déplacements non nuls est supposé être le point de contact entre le foie et le scalpel. Comme on suppose que le matériau et de type Mooney-Rivlin, on donne les valeurs des constantes C_1, C_2 caracteristiques du matériau considéré et a le terme de pénalisation aux valeurs suivantes :

$$C_1 = 19,7, C_2 = 1,97, a = 197 \times 10^3$$

On effectue trois tests avec ce programme. Pour chacun de ces tests, on impose des déplacements nuls aux mêmes nœuds. On impose également un déplacement non nul pour un seul nœud qui sera le même pour les trois tests mais avec des valeurs très differentes :

- Le premier cas test se veut être dans un cadre proche de l'élasticité linéaire puisqu'on impose un très petit déplacement (de l'ordre de 10^{-20}) pour le nœud considéré.
- Le second test est dans le cadre de l'élasticité non linéaire mais reste dans un voisinage de la solution. On converge donc vers la solution en quelques itérations. On montre figure 2.7 le résultat d'un calcul de ce type. Comme les déformation restent très petites on les amplifie d'un facteur 1200 pour observer quelque chose visuellement.
- Le dernier test (cf figure 2.8) est en dehors d'un voisinage de la solution et donc diverge. Le déplacement du nœud considéré n'est pourtant, en norme, que de l'ordre de l'unité. Pour arriver à trouver le champ de déplacement, il faudrait alors utiliser une méthode de Newton améliorée. On affiche le résultat du calcul.

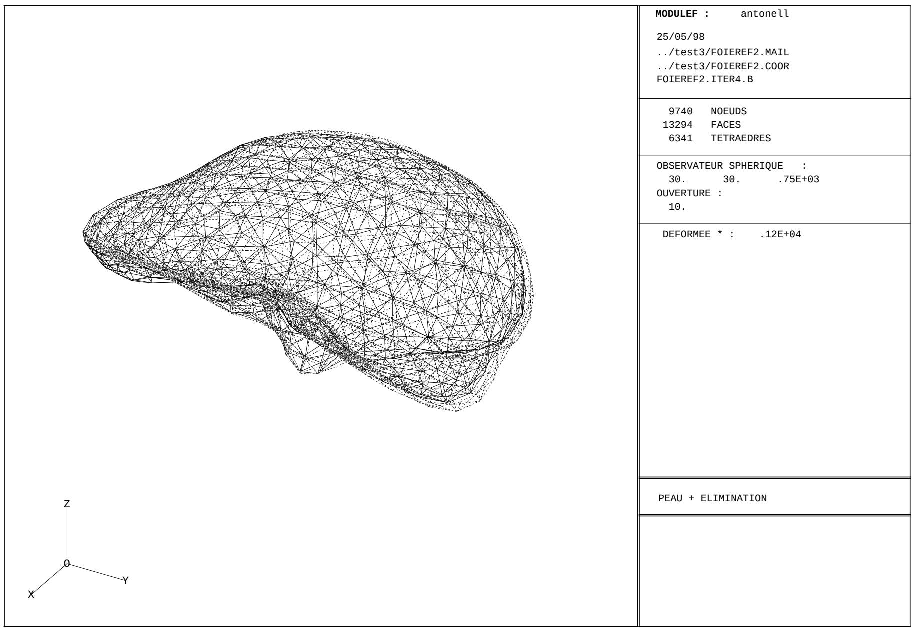

Fig. 2.7 – *Déformation obtenue lors du second cas test*

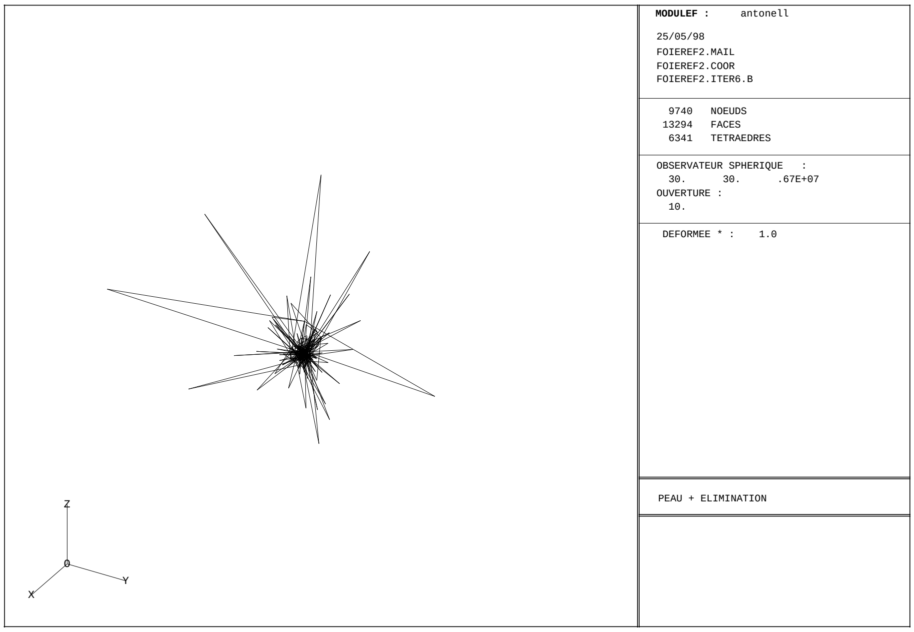

Fig. 2.8 – *Exemple de non convergence de l'algorithme de Newton*

On voit donc que, très rapidement, si on veut imposer de grandes déformations, on doit utiliser un algorithme de Newton avec pas incrémental ou une méthode de continuation.

Cependant, le système à résoudre comporte près de 35000 degrés de liberté (9740 pour chaque composante du déplacement) et est par conséquent assez lent à résoudre. Expérimentalement, on observe qu'il faut entre huit et dix heures pour le calcul d'une itération. On verra par la suite que le nombre d'itérations necessaires au calcul d'une déformée dans un cas non linéaire est de l'ordre de la centaine lorsqu'on utilise par exemple une méthode de Newton avec pas incrémental.

La taille du système à résoudre étant trop imposante pour traiter un problème non linéaire avec ce type de programme, on utilisera par la suite un programme parallèle pour le calcul des déformations dans ce genre de situation.

2.2.4 Utilisation de l'algorithme parallèle

On montre figure 2.9 le maillage d'un foie décomposé en 20 sous domaines.

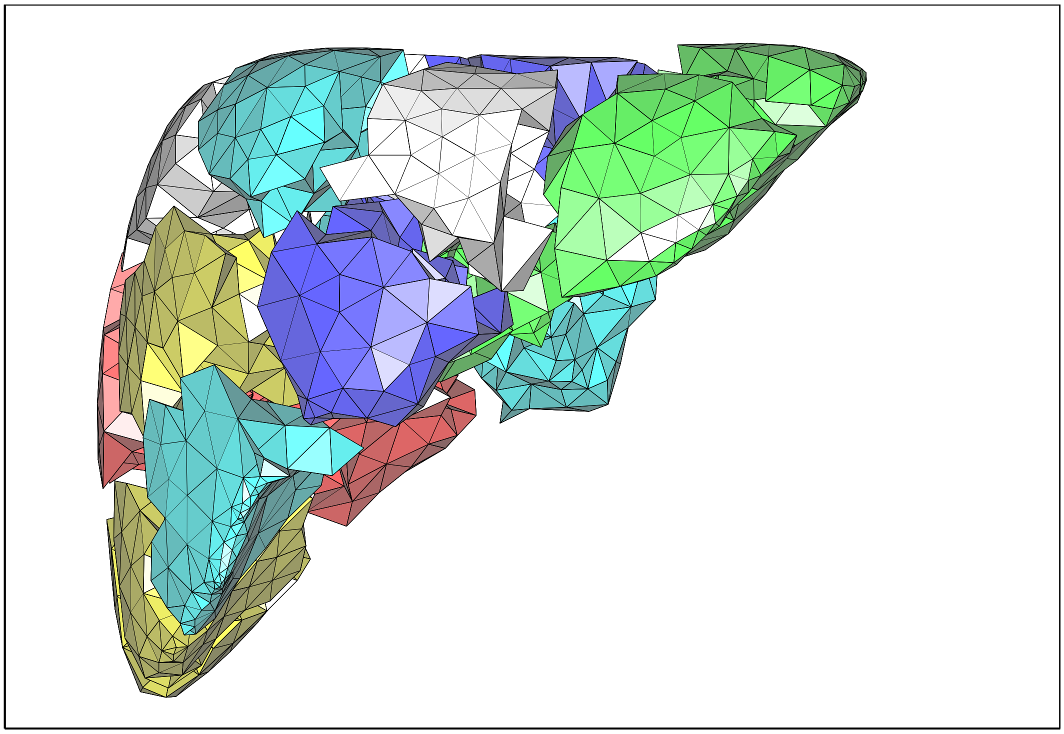

Fig. 2.9 – *Affichage des domaines du foie*

Cette décomposition de l'objet en sous domaines permet de diminuer drastiquement le temps de calcul par itération. En effet, on passe de 9 heures de calcul à moins de 15 minutes par itération. Il faut cependant préciser que l'algorithme utilisé dans la programmation parallèle n'est pas le même que celui utilisé pour l'algorithme séquentiel ce qui explique un tel gain de temps. On propose deux calculs de déformées dans un cadre non linéaire. Pour ces deux calculs, on utilise une méthode de Newton avec pas incrémental. Pour les deux cas test, on part des numéros de référence affichés figure 2.5. Pour ces deux tests, on fixe les constantes C_1, C_2 et a aux valeurs suivantes :

$$C_1 = 1520, C_2 = 152, a = 1520 \times 10^{+4}$$

Résultat du premier test

Pour ce cas test, on modifie légèrement le fichier donnant les références en chaque nœud. On cherche à avoir une singularité très localisée et donc ne tirer que sur un seul nœud. On va donc supprimer par exemple deux des nœuds qui ont une référence 5 et n'en garder qu'un seul à cette référence. On imposera alors à ce nœud un déplacement dans la direction normale ce qui correspond à ce qui se passe souvent lors d'une opération chirurgicale.

On veut imposer un déplacement d'environ 10% de la longueur du foie dans la direction de la normale extérieure. Le foie a une longueur de 100 unités environ, on décide donc d'imposer un déplacement de 10 fois la normale extérieure unitaire.

Comme on utilise une méthode de Newton avec pas incremental, on effectue de très nombreuses exécutions du programme de Newton classique. Le nombre des exécutions est d'environ 60. La résolution de chacun de ces problèmes nécessite entre 5 et 8 itérations et donc nécessite au total environ 100 heures de temps de calcul. Malgré cela, le déplacement reste visuellement relativement faible. On affiche cependant les résultats sans amplification figure 2.10. On voit que la déformation reste très localisée. Par la suite, on cherchera à imposer des déplacements sur toute une face au lieu d'un seul point.

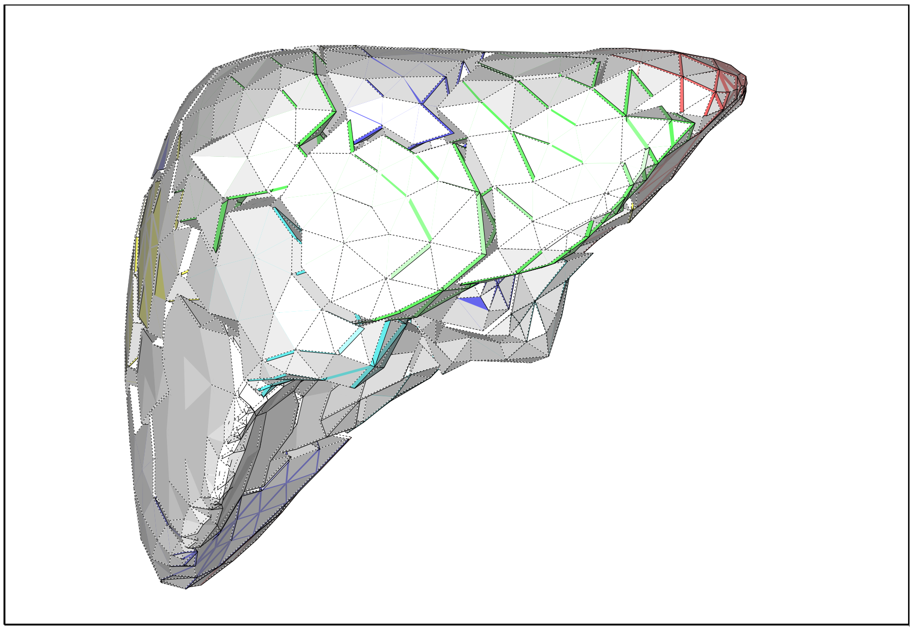

FIG. 2.10 – *Résultat du premier test d'élasticité*

Résultat du second test

Cette fois, on ne fait pas de modifications dans les numéros de référence. On va imposer un déplacement normal sur toute une face ce qui reviendra à translater cette face le long de la normale. Ici encore on tire en direction de l'extérieur du foie ce qui est physiquement plus réaliste que de tirer dans une direction tangente par exemple. Cette fois on veut obtenir un déplacement final de 6 fois la normale extérieure unitaire.

Le nombre des exécutions est d'environ 50. La résolution de chacun de ces problèmes nécessite entre 5 et 8 itérations et donc nécessite au total environ 80 heures de temps de calcul. On affiche les résultats de ce test figure 2.11 avec une amplification de 10.

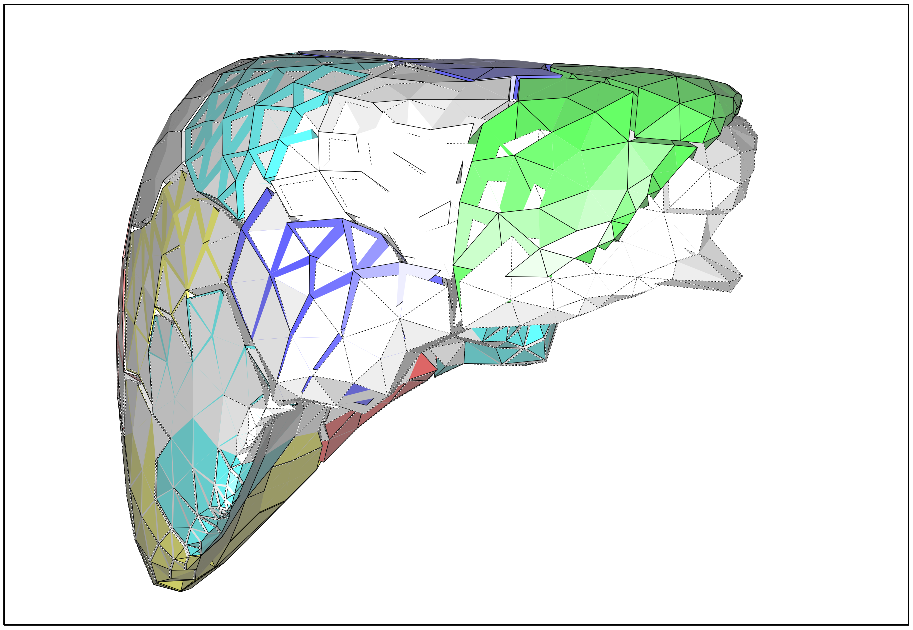

FIG. 2.11 – *Résultat amplifié du second test d'élasticité*

Non linéarité des problèmes

Pour les deux tests effectués avec l'algorithme parallèle, on est en élasticité non linéaire. On affiche figure 2.12 la valeur de la norme du déplacement d'un nœud du foie lors du premier test en fonction de λ (on rappelle que λ est un paramètre intervenant dans la méthode de Newton avec incrémentation du chargement cf chapitre 1) ainsi que la valeur qu'aurait cette norme si on se placait en élasticité linéaire. Dans le cas de l'élasticité linéaire, on a linéarité par rapport à λ. Ici, on n'a plus la linéarité par rapport à λ comme le montre le graphique affiché figure 2.12.

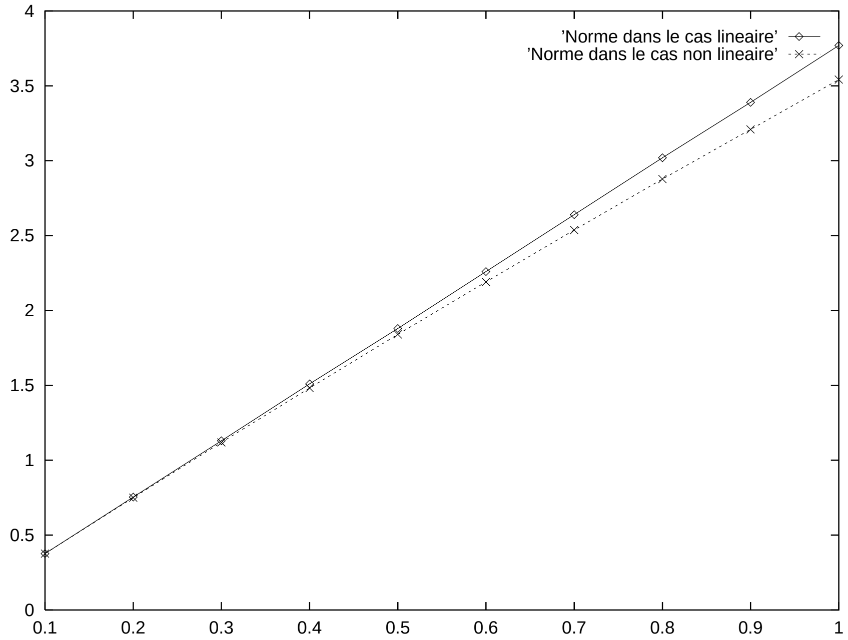

FIG. 2.12 – *Norme d'un point en fonction de* λ

La courbe avec des cercles est obtenue en multipliant la valeur de la norme obtenue lorsque $\lambda = 0.1$ par $10 \times \lambda$. Celle avec des croix étant quand à elle la norme du déplacement effectivement obtenu à l'aide du programme d'élasticité non linéaire. Les deux courbes sont quasiment confondues tant que $\lambda \leq 0.3$. Lorsque λ ne dépasse pas 0.3, les déplacements données par le programme non linéaire semblent donc proches de ceux qu'on obtiendrait en multipliant les déplacements obtenus avec une très petite valeur de λ.

Chapitre 3

Principe de l'analyse modale et application à la biomécanique.

3.1 Principe Général

3.1.1 Introduction

On donne dans ce chapitre une méthode très rapide de résolution approchée d'un système linéaire. Cette méthode pourra être appliquée à la biomécanique pour donner des déformations en temps réel. On considére ici le cadre général de cette technique et on montre ensuite comment appliquer cette méthode à la biomécanique. Des traveaux ultérieurs pouront se baser sur cette méthode pour la similation en temps réel d'opérations chirurgicales.

3.1.2 Résolution exacte

Soit A une matrice $N \times N$, à coefficients réel, symétrique et définie positive. On cherche à obtenir une approximation de la solution du problème $Ax = b$. On utilise pour cela une méthode de type modale dont on va décrire le principe général dans cette section.

On pose $\lambda_1, ..., \lambda_N$ les N valeurs propres strictement positives de la matrice A répétées avec leurs ordre de multiplicité. Soit $v_1, ..., v_N$, les N vecteurs propres associés aux N valeurs propres. On sait alors, d'après un résultat très classique d'algèbre linéaire, que ces N vecteurs propres, correctement normalisés, forment une base orthonormale de $\mathbb{R}^N$. On peut donc décomposer b sur cette base et on a alors :

$$b = \sum_{i=1}^{N} < b, v_i > v_i$$

En posant :

$$\bar{x} = \sum_{i=1}^{N} \frac{< b, v_i >}{\lambda_i} v_i \tag{3.1}$$

On vérifie aisément que :

$$A\bar{x} = \sum_{i=1}^{N} < b, v_i > v_i = b$$

et donc que $\bar{x}$ est solution de $Ax = b$.

3.1.3 Approximation non linéaire

Une maniére de trouver une bonne approximation de $\bar{x}$ sans nécessiter trop de calculs est de résoudre le problème approché $A\tilde{x} = \tilde{b}$ ou

$$\tilde{b} = \sum_{i \in P} < b, v_i > v_i$$

avec P un sous ensemble de $\{1, ..., N\}$.
On a alors :

$$\tilde{x} = \sum_{i \in P} \frac{< b, v_i >}{\lambda_i} v_i$$

Si on veut minimiser l'erreur $\|\bar{x} - \tilde{x}\|$, on a alors intêret à mettre dans P les indices i tel que $\frac{<b, v_i>}{\lambda_i}$ soit grand en module.

3.1.4 Approximation linéaire

Lorsqu'on fait une approximation linéaire, on prend alors $P = \{1, ..., m\}$ avec $m \leq N$. Ce choix est guidé par le fait qu'on a alors de bonnes chances que les coefficients $\frac{<b, v_i>}{\lambda_i}, i \in P$ soient parmi les plus grands en module et donc que $\|\bar{x} - \tilde{x}\|$ soit petit.

3.2 Application à la biomécanique

Modéliser le comportement d'un foie humain nécessite de très long calculs. Ceci est due au fait que la géométrie de l'objet à mailler est très compliquée ce qui fait qu'on doit faire les calculs avec un maillage très fin de l'objet et surtout que ces calculs se font en élasticité non linéaire incompressible. Si on veut obtenir un comportement en temps réel de l'objet, on doit simplifier le problème et utiliser des algorithmes rapides. C'est pour cela qu'on impose un comportement linéaire au matériau. De plus, on ne calcule pas un déplacement exact mais une approximation du déplacement qu'on espère très proche de la solution exacte.

La méthode qui a été retenue est donc d'utiliser une méthode modale d'approximation linéaire qui a été exposée au chapitre précédent. Plus exactement, on va utiliser une variante de cette méthode dans l'espoir d'obtenir une meilleure approximation de la solution. En effet, on a dit au chapitre précédent que le principe de l'approximation linéaire est de résoudre le systéme linéaire avec un second membre $\tilde{b}$, ou $\tilde{b}$ est la projection de b sur les m vecteurs propres associés aux m valeurs propres de plus petit module. Cependant, il est possible que les sollicitations imposées par le chirurgien sur le foie donnent des seconds membres très éloignées de l'espace engendré par les m vecteurs selectionnés. Pour combler ce vide, on doit ajouter à l'espace d'approximation des vecteurs qui approximent correctement le type de sollicitation courante qu'un chirurgien peut imposer sur le foie.

Au final, l'espace d'approximation sera engendré par l'union de deux familles de vecteurs :

- Les vecteurs associés aux valeurs propres de plus petit module.
- Les vecteurs issus de sollicitations type que le chirurgien est suceptible d'imposer.

Bibliographie

[1] Ph. G. Ciarlet. *Mathematical Elasticity Tome 1.* North-Holland, Amsterdam, New-York, 1988.

[2] P. Le Tallec. *Numerical methods for nonlinear three-dimensional elasticity.*

[3] E. Stein U. Brink. A-posteriori error estimation in large-strain elasticity using equilibrated local neumann problems. *Computer Methods in Applied Mechanics and Engineering*, 1997.