



Stage ingénieur de recherche de 3^{ème} année, effectué au CMAP de l'Ecole Polytechnique, en vue de l'obtention du diplôme d'ingénieur dans la spécialité mathématiques appliquées et calcul scientifique (MACS) de Sup Galilée et du Master 2 Mathématiques et Informatique (MMI) de Paris 13

Conception optimale de structures

Optimisation géométrique de formes en conduction et élasticité linéarisé - Utilisation du critère de l'énergie de déformation - Calcul de solutions explicites sur cas simples

NICOLAS Dimitri

Maîtres de stage : G. Allaire, J-F. Babadjian



<dimi.nicolas@gmail.com, gregoire.allaire@polytechnique.edu,
jean-francois.babadjian@polytechnique.edu>
Ecole polytechnique - Sup Galilée

Table des matières

1	Introduction	1
2	Présentation du CMAP	2
3	L'optimisation de forme	4
3.1	Généralités	4
3.2	Optimisation paramétrique	5
3.2.1	Analyse du problème	5
3.2.2	Application numérique	5
4	Optimisation géométrique	6
4.1	Théorie sous jacente	6
4.1.1	Contre-exemple de non existence	7
4.1.2	Existence sous condition de régularité	7
4.1.3	Une notion de variation de domaine	8
4.1.4	Différentiation par rapport au domaine	8
4.1.5	Dérivation rapide : methode du lagrangien	10
4.2	Cas de la conduction/membrane	11
4.2.1	Cas de la compliance	11
4.2.2	Cas de l'énergie de déformation	14
4.2.3	Anneau et forces suiveuses	16
4.2.4	Anneau et forces volumiques	20
4.3	Cas de l'élasticite linéarisée	22
4.3.1	Le problème aux limites	22
4.3.2	Cas de la compliance	22
4.3.3	Cas de l'énergie de déformation	24
4.4	Algorithme et numérique	25
4.4.1	La méthode du gradient	25
4.4.2	Gestion de la contrainte	26
4.4.3	Critère d'arrêt	27
4.4.4	Algorithme utilisé	27
4.5	Cas tests effectués	28
4.5.1	Conduction et forces volumiques	28
4.5.2	Conduction et variation du Neumann : rectangle	36
4.5.3	Conduction et variation du Neumann : carré troué	41
4.5.4	Conduction et forces volumiques : carré troué	45
4.5.5	Conduction L	46
4.5.6	Conduction M	52
4.5.7	La console en élasticité linéarisée	59
4.6	Contrainte géométrique	62

4.7	Optimisation du code	64
4.7.1	Direction de descente	64
4.7.2	Retournement de triangle	64
4.7.3	Contrôle du pas	65
4.7.4	Irrégularités aux coins	65
4.7.5	Eviter les oscillations	66
4.7.6	Extension de la normale	66
5	Optimisation par lignes de niveaux	67
5.1	Cadre théorique	67
5.2	Algorithme	67
6	Conclusion	69

Je remercie Grégoire Allaire pour m'avoir permis d'effectuer mon stage au sein du CMAP de l'Ecole Polytechnique : il m'a fait confiance et continue à travers la thèse qu'il m'a proposée et que je démarre bientôt. Merci aussi à Jean-François Babadjian pour son temps, sa patience et son écoute. Merci aussi à leur collègue Olivier Pantz qui m'a aussi aidé avec grande gentillesse.

Merci à Olivier Lafitte qui, passionné, essaye de faire monter la MACS au plus haut.

Notations et abréviations

Ω : Ouvert de \mathbb{R}^n qui représente la forme de la structure au repos

$\partial\Omega$: bord de Ω

Γ_N : bord Neumann

Γ_D : bord Dirichlet

Γ : bord libre de force

J : le critère à minimiser ou maximiser (appelé objectif ou fonction coût)

\mathcal{L} : lagrangien du problème

u : solution de l'équation d'état

p : solution de l'équation adjointe

f : force volumique de l'EDP

g : force surfacique de l'EDP

l : multiplicateur de lagrange

α_l : coefficient permettant de contrôler la précision sur la contrainte volumique

θ : direction de descente dans les différentiations

A , λ et μ : loi de Hooke et coefficients de Lamé

H : courbure, divergence de la normale

C_{opt} : condition d'optimalité

Code couleur FreeFem++ : le violet est la valeur la plus élevée et positive, puis le bleu, le vert et pour finir le jaune (proche de la valeur nulle) et l'orange pour les premières valeurs négatives. Exemple :



Résumé

L'optimisation de forme va permettre, de façon automatique, de calculer des meilleures formes pour un certain critère donné. Ce qui autrefois relevait de considérations techniques et intuitives, les mathématiciens d'aujourd'hui utilisent l'optimisation de forme pour toutes sortes de problèmes de l'industrie comme par exemple l'optimisation de la forme d'un pont ou la forme du fuselage d'un avion. Nous allons passer en revue une méthode d'optimisation de forme : l'optimisation dite géométrique, en discuter tant sur le plan théorique que numérique. Des résultats théoriques seront notamment obtenus dans le cadre de l'optimisation d'un anneau : solutions radiales.

Chapitre 1

Introduction

L'optimisation de forme, également appelé conception optimale de structures, concerne la recherche de la meilleure forme pour un problème donné. Celle-ci est conditionnée en particulier par la donnée du critère que l'on souhaite suivre dans l'optimisation, et des contraintes de poids, de volume, d'encombrement, ou encore de coût de réalisation. Cette optimisation de forme requiert l'utilisation de méthodes mathématiques et d'algorithmes qui permettront d'automatiser l'optimisation. En effet, on peut optimiser une forme, sans utiliser d'algorithme, par des considérations qui relèvent de l'intuition et de l'expérience : l'ingénieur est capable de rechercher une forme optimale en procédant par essais-erreurs, en construisant des prototypes, ce qui se révélera lent et coûteux et ne permettra pas forcément de trouver la forme optimale.

Historiquement, l'optimisation de forme est apparue la première fois, d'après la légende, en 814 av J.C. lorsque la reine Didon, fuyant les Assyriens, accosta sur les rives de l'actuelle Tunisie. Souhaitant s'y installer et fonder une ville (la future Carthage), Didon demanda au chef de la tribu qui occupait les lieux l'autorisation de disposer d'un territoire. Celui-ci lui tendit alors une peau de boeuf en lui disant d'un air goguenard : Le territoire que vous arriverez à couvrir avec cette peau est à vous ! Didon découpa la peau en une très fine lanière et se trouva alors confrontée au problème suivant : disposant d'une lanière de longueur donnée, comment enclore un territoire de surface maximale ?

On peut décomposer les méthodes d'optimisation de forme en trois grandes classes : l'optimisation paramétrique, qui permet d'optimiser une forme à travers l'optimisation d'un nombre limité de paramètres, l'optimisation géométrique de forme autorisant de modifier la frontière de la forme, et l'optimisation topologique lorsque non seulement la forme varie mais aussi la topologie.

Ce rapport présentera une méthode d'optimisation de forme par une méthode déterministe (par opposition aux méthodes stochastiques telles que les algorithmes évolutionnaires). L'objectif est de passer en revue la méthode géométrique, de comprendre son fonctionnement, aussi bien du point de vue théorique (espaces de Sobolev et optimisation comme base de réflexion) que pratique/numérique (implémentation par la méthode des éléments finis et méthodes de gradient et autres en FreeFem++), et d'appréhender ses atouts et faiblesses. Enfin, il conviendra de se placer dans le cas de la mécanique en élasticité linéarisée afin d'obtenir des formes optimales dans divers contextes issus de la mécanique. Précisons que beaucoup de calculs et notations seront plus formels qu'autre chose dans un but de clarté et de compréhension : le lecteur pourra se référer à [2] (d'où sont tirés un certain nombre de formules et résultats) pour plus de précision.

Chapitre 2

Présentation du CMAP

Depuis sa création en 1974, le Centre de Mathématiques Appliquées (CMAP) s'est fixé comme objectif le développement et l'exploration des mathématiques en liaison avec les applications. L'ouverture du CMAP vers d'autres disciplines se manifeste par la variété et la complexité des thèmes de recherche qui sont abordés. Son fonctionnement permet à ses équipes d'explorer et d'initier des thématiques nouvelles. Les domaines de recherche du CMAP sont en liaison étroite avec des problématiques posées en physique, mécanique, chimie, finance et biologie mais aussi dans les domaines socio-économiques ou des technologies de l'information. L'essence des recherches au CMAP obéit au cycle : modélisation, analyse mathématique, simulations numériques, visualisation puis raffinement de la modélisation. Chaque étape de ce cycle utilise les compétences des membres du laboratoire.

Le CMAP est organisé en équipes couvrant ses domaines de recherche. La politique scientifique du centre a permis un essaimage régulier de ses compétences et de son personnel vers d'autres laboratoires.

Le CMAP travaille en étroite liaison avec le Département d'Enseignement-Recherche en Mathématiques Appliquées de l'Ecole Polytechnique dans la définition de la politique d'enseignement et de recherche, du recrutement et de la participation du personnel CNRS aux enseignements de mathématiques appliquées tant en cycle ingénieur qu'en Master.

Le personnel enseignant chercheur du CMAP est pour l'essentiel composé de professeurs, professeurs chargés de cours et maîtres de conférence recrutés à temps plein ou partiel par le Département d'Enseignement-Recherche en Mathématiques Appliquées, et d'enseignants venant d'autres universités comme l'Université de Paris 10 ou l'Université d'Evry dans lesquelles des unités mixtes du CNRS n'existent pas. Ces derniers exercent au CMAP leurs recherches à mi-temps. Trois nouvelles équipes projets communes au CMAP et à l'INRIA Futurs ont rejoint le CMAP. Le personnel ITA du laboratoire est recruté soit par l'Ecole Polytechnique soit par le CNRS et récemment par INRIA Futurs (depuis septembre 2007). Ce personnel est affecté à des équipes du laboratoire ou à la gestion du système informatique ou au secrétariat du laboratoire.

Les effectifs du CMAP (enseignants chercheurs, chercheurs, doctorants et ITA) se montent à environ 70 personnes dont 25 doctorants. Les enseignants-chercheurs du CMAP employés à temps plein par l'École Polytechnique sont actuellement : 5 Professeurs, 1 Directeur de Recherche, 4 Professeurs Chargés de Cours (dont un poste vacant en 2007), 1 Maître de Conférences. Notons que 2 Professeurs Chargés de Cours ont quitté le CMAP dans la période 2004-2007 (1 PR universitaire et 1 DR INRIA).

Le CNRS fournit encore la plus grande partie des chercheurs permanents et reste un partenaire important du CMAP : 4 Directeurs de Recherche dont un à 1/3 temps depuis mars 2007, 5 Chargés de Recherche dont un vient de rejoindre par mutation le centre en septembre 2007 (notons aussi que 2 autres CR affectés au CMAP sont en détachement). Le rapprochement du CMAP avec INRIA Futurs a permis l'accueil d'un DR INRIA en septembre 2006 et de 2 CR INRIA en septembre 2007 dans le cadre d'équipes projets communes. Cette politique de rapprochement avec l'INRIA va être renforcée. La grande majorité des chercheurs permanents du CMAP sont également enseignants à mi-temps du Département de Mathématiques Appliquées.

Le CMAP compte aussi 3 professeurs des universités qui y effectuent une partie de leur recherche, leurs universités ne possédant pas encore d'UMR de mathématiques. Notons que les équipes-projet INRIA comprennent aussi officiellement (du point de vue de l'INRIA) des chercheurs extérieurs au CMAP.

Trois ingénieurs de recherche assurent la maintenance du réseau et des systèmes informatiques, l'assistance informatique, l'encadrement des doctorants au niveau logiciel, le développement des logiciels et la gestion des contrats. Le secrétariat du CMAP, auquel il faut adjoindre le secrétariat du département de Mathématiques Appliquées et le secrétariat du Master est assuré par 3 personnes.

Le CMAP est structurée en deux composantes principales, correspondant à deux domaines d'expertise forts, qui sont "Aléatoire" et "Analyse et Calcul Scientifique". Toutefois, cette séparation, en partie historique, ne représente pas une scission en deux parties indépendantes et presque toutes les équipes de recherche sont 'a cheval sur ces deux thématiques. Une grande originalité et l'attractivité du CMAP résident en ce fait que toutes les grandes thématiques des mathématiques appliquées sont représentées et qu'elles interagissent 'a travers des problématiques communes.

Chaque composante est formée de plusieurs équipes couvrant les thématiques de recherche du CMAP. Chaque chercheur ou enseignant chercheur, chaque doctorant et post-doctorant développe ses recherches dans une ou deux équipes. Pour plus d'information sur les équipes voir le site du CMAP.

Source : site web du CMAP¹

¹<http://www.cmap.polytechnique.fr>

Chapitre 3

L'optimisation de forme

3.1 Généralités

Un problème d'optimisation de forme est défini par trois données :

- *Un modèle* i.e. une équation aux dérivées partielles muni de composantes physiques permettant de correctement refléter le problème.
- *Un ou plusieurs critères* que l'on cherche à minimiser ou maximiser. (par exemple on veut souvent maximiser la rigidité d'une structure)
- *Un ensemble admissible* de variables d'optimisations tenant compte éventuellement de contraintes imposées aux variables.

Différentes questions se posent alors, étant donné un problème d'optimisation de forme :

- *Existence et unicité* d'une meilleure forme (dite forme optimale) ;
- *Conditions d'optimalités* (nécessaires et/ou suffisantes) qui caractérisent les solutions optimales ;
- *Calcul numérique* de forme optimales approchées.

Un problème d'optimisation de forme se composera donc : d'une EDP, d'une fonction coût et de contraintes (contraindre une forme à avoir un certain volume par exemple). Il sera par la suite indispensable de maîtriser la notion de différentiation par rapport au domaine, qui nous permettra d'établir explicitement la dérivée de forme de la fonction coût. Cette dernière permettra de calculer le champ de vecteur nécessaire pour la modification de la forme dans une direction donnée, à une itération donnée.

3.2 Optimisation paramétrique

3.2.1 Analyse du problème

Posons nous le problème du calcul de l'épaisseur optimale d'une membrane élastique fixée sur son contour déformée par une force extérieure f (voir la partie sur l'optimisation paramétrique de [2] pour un schéma et plus d'explications).

La variation de la forme ici est prise en compte à travers la variation d'un paramètre d'épaisseur, la forme Ω ne varie pas. Au repos, la membrane occupe un ouvert borné Ω de \mathbb{R}^n . On va faire varier l'épaisseur entre deux valeurs extrêmes $h_{max} > h_{min} > 0$. Le coefficient d'élasticité de la membrane est proportionnel à cette épaisseur h . Ainsi le comportement de la membrane est modélisé par :

$$\begin{cases} -\operatorname{div}(h\nabla u) = f & \text{dans } \Omega \\ u = 0 & \text{dans } \Gamma \end{cases} \quad (3.1)$$

où u est le déplacement vertical de la membrane.

On définit une fonction coût $J(h) = \int_{\Omega} j(u)dx$ où u est la solution de l'équation précédente (qui dépend de h). On prendra $j(u) = fu$, pour travailler avec la compliance.

Remarquons que nous n'avons toujours pas explicité le cadre fonctionnel, ni encore parler de l'existence et l'unicité d'une éventuelle solution à ce problème. Définissons \mathcal{U}_{ad} l'ensemble des épaisseurs h admissibles :

$$\mathcal{U}_{ad} = \left\{ h \in L^{\infty}(\Omega), \quad h_{max} \geq h(x) > 0 \text{ p.p. dans } \Omega, \quad \int_{\Omega} h(x)dx = h_0|\Omega| \right\} \quad (3.2)$$

Si $f \in L^2(\Omega)$ et si l'on discrétise l'espace (3.2) ou bien pose une contrainte de régularité sur l'épaisseur, alors la solution $u(h)$ de (3.1) est dans $H_0^1(\Omega)$ et le problème d'optimisation

$$\inf_{h \in \mathcal{U}_{ad}} J(h)$$

admet au moins une solution optimale.

Le problème est ainsi bien posé et il ne reste plus qu'à calculer le gradient de la fonction coût, d'écrire l'algorithme à utiliser et de le faire tourner.

Encore une fois, ceci n'est qu'une courte introduction pour présenter l'aspect de ce type de problème. Pour plus de détails, se reporter à [2].

3.2.2 Application numérique

J'ai utilisé quelques codes créés par G. Allaire sur l'optimisation paramétrique et joué avec les données. Pour des résultats et codes numériques, voir la section FreeFem de la page personnelle de G. Allaire :

www.cmap.polytechnique.fr/~allaire/freefem.html

Chapitre 4

Optimisation géométrique

Dans cette partie, une certaine compréhension des espaces de Sobolev est préférable pour comprendre la théorie, ainsi que quelques connaissances en optimisation sous contraintes. Le lecteur pourra se référer aux nombreux ouvrages sur les sujets. Une connaissance des éléments finis est aussi requise pour ceux souhaitant comprendre l'implémentation des codes numériques : voir la partie 6.

La théorie ci-dessous est tiré en grande partie de [1] et [2]. De nombreux autres auteurs ont travaillé sur le sujet.

4.1 Théorie sous jacente

Posons tout d'abord un certain cadre théorique (dû principalement à F. Murat et J. Simon [3,4]). Ce cadre va nous permettre d'affirmer l'existence d'une forme optimale au problème que nous allons décrire sous une condition de régularité.

On se donne un domaine de référence Ω_0 que l'on suppose être un ouvert borné et de classe C^1 de \mathbb{R}^n . On suppose que le bord $\partial\Omega_0$ se divise en trois parties disjointes non vides

$$\partial\Omega_0 = \Gamma \cup \Gamma_N \cup \Gamma_D$$

où l'on note Γ_D le bord Dirichlet, Γ_N le bord Neumann et Γ le bord libre de force.

Choisissons un modèle pour se fixer les idées, le modèle de membrane :

$$\left\{ \begin{array}{ll} -\Delta u = f & \text{dans } \Omega \\ u = 0 & \text{sur } \Gamma_D \\ \frac{\partial u}{\partial n} = g & \text{sur } \Gamma_N \\ \frac{\partial u}{\partial n} = 0 & \text{sur } \Gamma \end{array} \right. \quad (4.1)$$

D'une façon générale, les trois bords peuvent être considérés comme variables, i.e. que l'on va les modifier légèrement pour minimiser notre fonction coût. Mais de façon pratique, seul la partie Γ sera variable, les bords Dirichlet et Neumann n'ayant dans la réalité des problèmes que l'on se pose aucune raison de varier. (On fixe donc Γ_D et Γ_N , voir section 4.5.2 le cas non fixé de Γ_N).

Le problème de l'existence d'une solution optimale se pose alors. Existe-il une forme optimale dans tous les cas ? Quelles sont les conditions pour avoir l'existence ? Pouvons nous trouver un contre exemple de non existence pour motiver la recherche de conditions d'existence ? Y a t-il unicité ? Toutes ces questions seront abordées dans ce qui suit. Voyons tout d'abord un contre-exemple de non existence d'une forme optimale et ensuite une condition pour avoir l'existence. La suite sera réservée au calcul de la dérivée de forme par deux moyens différents.

4.1.1 Contre-exemple de non existence

Expliquons formellement comment il peut ne pas y avoir de solution optimale à partir d'un exemple tiré de [2].

En tirant uniformément sur les cotés d'un domaine Ω rectangulaire, les contraintes auront tendances à être alignées horizontalement. Il sera donc préférable de faire des trous qui seront allongés eux aussi horizontalement de manière à ne pas trop perturber ce champ de contraintes. Le problème c'est qu'à volume constant, il est préférable d'avoir plusieurs petits trous que quelques gros trous. Ainsi en réitérant à l'infini, on arrive à faire diminuer notre fonction coût à chaque fois. Il n'y a donc pas de solution optimale.

4.1.2 Existence sous condition de régularité

L'idée de base de la condition de régularité qui nous permettra d'obtenir l'existence d'une forme optimale est de définir un ensemble de forme admissible \mathcal{U}_{ad} dont tous les éléments s'obtiennent par application d'un difféomorphisme régulier au domaine de référence Ω_0 . De plus le difféomorphisme sera pris proche de l'identité. On modifie aussi très peu la forme de référence mais assez pour la faire varier sur un ensemble suffisamment grand de sorte à assurer l'existence de la forme)

Prenons l'espace de Banach des fonctions Lipschitziennes ϕ de \mathbb{R}^n dans \mathbb{R}^n muni de la norme

$$\|\phi\|_{W^{1,\infty}(\Omega,\mathbb{R}^n)} = \sup_{x \in \Omega} (|\phi(x)| + |\nabla \phi(x)|)$$

On définit un espace de formes admissibles obtenues par déformation de Ω_0 :

$$\mathcal{D}(\Omega_0) = \{\Omega \subset \mathbb{R}^n, \quad \exists T \in \mathcal{T}, \quad \Omega = T(\Omega_0)\} \quad (4.2)$$

où \mathcal{T} est l'espace des difféomorphismes sur \mathbb{R}^n définis par :

$$\mathcal{T} = \{T : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad (T - Id) \in W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n), \quad (T^{-1} - Id) \in W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n)\} \quad (4.3)$$

Chaque forme admissible est donc représentée par un difféomorphisme $T \in \mathcal{T}$ (cette représentation est non unique). Etant donné que les fonctions de $W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$ sont continues alors les applications de \mathcal{T} sont aussi des homéomorphismes (bijections continues d'inverses continues) et ils préservent donc la topologie des domaines auxquels ils s'appliquent. Par conséquent, toutes les formes de $\mathcal{D}(\Omega_0)$ auront la même topologie que Ω_0 .

Introduisons une pseudo distance sur $\mathcal{D}(\Omega_0)$:

$$d(\Omega_1, \Omega_2) = \inf_{T \in \mathcal{T} | T(\Omega_1) = \Omega_2} (\|T - Id\|_{W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n)} + \|T^{-1} - Id\|_{W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n)}) \quad (4.4)$$

On peut maintenant définir une condition de régularité des formes admissibles en se limitant à des ouverts Ω proches de Ω_0 au sens de cette pseudo distance d . On se fixe une constance $C > 0$ et on pose :

$$\mathcal{U}_{ad} = \mathcal{U}_{ad}(C) = \{\Omega \in \mathcal{D}(\Omega_0), \quad d(\Omega, \Omega_0) \leq C, \quad |\Omega| = V\} \quad (4.5)$$

Un résultat est alors possible si notre fonction coût est soit la compliance, soit un critère de moindre carré :

$$J(\Omega) = \int_{\Gamma_N} g u ds \quad \text{ou} \quad J(\Omega) = \int_{\Omega} |u - u_0|^2$$

où $u_0 \in H^1(\mathbb{R}^n)$ et g est la trace sur Γ_N d'une fonction dans $H^1(\mathbb{R}^n)$ (Voir [3]).

Théorème 1. *On suppose que \mathcal{U}_{ad} est défini par (4.5). Pour les fonctions objectifs ci dessus (compliance et critère moindre carré), le problème d'optimisation de formes*

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega) \quad (4.6)$$

admet au moins une forme optimale.

Remarquons que le théorème donne un résultat intéressant du point de vue théorique mais qu'il est difficile de vérifier numériquement la contrainte de régularité que doivent satisfaire les formes admissibles. Cependant l'idée de représenter les formes admissibles par des difféomorphismes est utile pour définir une notion de dérivation par rapport au domaine.

Remarquons que l'on peut aussi prouver l'existence d'une forme optimale sous une condition géométrique (Voir [2] pour l'explication complète). On pourrait aussi utiliser une contrainte sur le volume ou le périmètre.

4.1.3 Une notion de variation de domaine

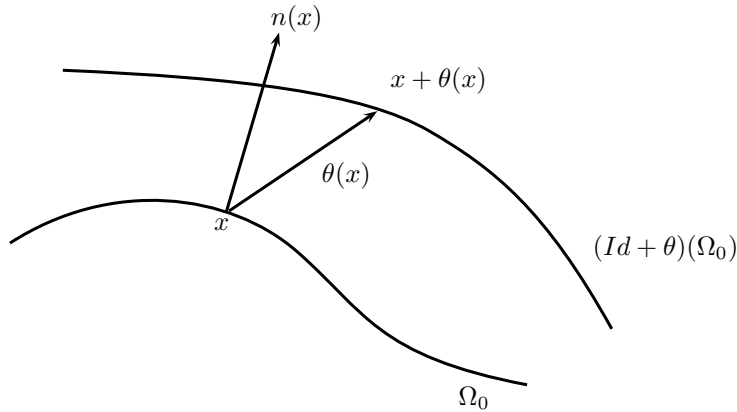
Puisque l'on veut optimiser la forme d'un objet selon un certain critère, nous avons besoin de définir une notion pour pouvoir faire varier la forme. Ici il s'agit d'une notion de variation de frontière. Pour ce faire nous considérerons seulement les difféomorphismes proches de l'identité, i.e. :

$$T = Id + \theta$$

avec $\theta \in W^{1,\infty}(\mathbb{R}^n, \mathbb{R}^n)$ Ainsi, l'ensemble $\Omega = (Id + \theta)(\Omega_0)$ est défini par :

$$\Omega = \{x + \theta(x) | x \in \Omega_0\}$$

On peut interpréter $\theta(x)$ comme un champ de vecteur qui transporte le domaine de référence Ω_0 . Autrement dit, on représente chaque forme admissible par un champ de vecteurs $\theta(x)$ de \mathbb{R}^n dans \mathbb{R}^n .



On peut alors définir une notion de différentiabilité en Ω_0 en utilisant la dérivation par rapport à θ . Le lecteur pourra consulter [2] pour la preuve du lemme qui assure que si θ est assez petit, alors $T = Id + \theta$ est bien un difféomorphisme et appartient à l'ensemble \mathcal{T} .

4.1.4 Différentiation par rapport au domaine

Nous allons introduire dans cette partie une méthode de variation de frontière qui remonte à Hadamard (1904) en se basant sur [3,4]. (voir également [5,6]). Dès lors que l'on saura différencier par rapport au domaine, on pourra écrire des conditions d'optimalités pour caractériser les formes optimales et calculer les gradients pour mettre en œuvre une méthode numérique.

Lemme 1. Soient Ω_0 un ouvert de \mathbb{R}^n , $T \in \mathcal{T}$ un difféomorphisme de \mathbb{R}^n et $1 \leq p \leq \infty$. Alors $f \in L^p(T(\Omega_0))$ si et seulement si $f \circ T \in L^p(\Omega_0)$, et on a alors :

$$\int_{T(\Omega_0)} f(x) dx = \int_{\Omega_0} (f \circ T)(x) |det \nabla T(x)| dx \quad (4.7)$$

Lemme 2. Soient Ω_0 un ouvert borné de classe \mathcal{C}^1 de \mathbb{R}^n et $T \in \mathcal{T}$ un \mathcal{C}^1 difféomorphisme de \mathbb{R}^n . Si $f \in L^1(\partial T(\Omega_0))$, alors $f \circ T \in L^1(\partial\Omega_0)$ et on a :

$$\int_{\partial T(\Omega_0)} f ds = \int_{\partial\Omega_0} (f \circ T) |det \nabla T| |((\nabla T)^{-1})^* n|_{\mathbb{R}^n} ds \quad (4.8)$$

Définition 1. Soit $J(\Omega) : \mathcal{D}(\Omega_0) \rightarrow \mathbb{R}$. On dit que J est différentiable par rapport au domaine en Ω_0 si la fonction

$$\theta \rightarrow J((Id + \theta)(\Omega_0))$$

est différentiable en 0 dans $W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$

Notons que la différentiabilité au sens de Fréchet s'écrit :

$$J((Id + \theta)(\Omega)) = J(\Omega) + J'(\Omega)(\theta) + o(\theta) \quad \text{avec} \quad \lim_{\|\theta\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)} \rightarrow 0} \frac{|o(\theta)|}{\|\theta\|_{W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)}} = 0 \quad (4.9)$$

où $J'(\Omega)$ est une forme lineaire continue sur $W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$.

Un résultat classique sur la dérivée directionnelle affirme que $J'(\Omega)$ ne depend que de la trace normale $\theta.n$ sur le bord $\partial\Omega$. En effet,

Lemme 3. Soit Ω un ouvert borné régulier et $J(\Omega)$ une fonction différentiable en Ω . Alors :

$$J'(\Omega)(\theta_1) = J'(\Omega)(\theta_2)$$

si $\theta_1, \theta_2 \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$ sont telles que $\theta_1 - \theta_2 \in \mathcal{C}^1(\mathbb{R}^n; \mathbb{R}^n)$ et $\theta_1.n = \theta_2.n$ sur $\partial\Omega$.

Voici deux exemples de dérivées de forme qui se révéleront très utiles par la suite (cf [2]).

Proposition 1. Soit Ω_0 un ouvert borné de classe \mathcal{C}^1 de \mathbb{R}^n . Soit $f \in W^{1,1}(\mathbb{R}^n)$ et J l'application de $\mathcal{D}(\Omega_0)$ dans \mathbb{R} définie par :

$$J(\Omega) = \int_{\Omega} f(x) dx$$

Alors J est différentiable en Ω_0 et on a :

$$J'(\Omega_0)(\theta) = \int_{\Omega_0} \text{div}(\theta(x)f(x)) dx = \int_{\partial\Omega_0} \theta(x).n(x)f(x) ds \quad (4.10)$$

Proposition 2. Soit Ω_0 un ouvert borné de classe \mathcal{C}^1 de \mathbb{R}^n . Soit $f \in W^{2,1}(\mathbb{R}^n)$ et J l'application de $\mathcal{D}(\Omega_0)$ dans \mathbb{R} définie par :

$$J(\Omega) = \int_{\partial\Omega} f ds$$

Alors J est différentiable en Ω_0 et on a :

$$J'(\Omega_0)(\theta) = \int_{\partial\Omega_0} (\nabla f . \theta + f(\text{div} \theta - \nabla \theta n . n)) ds \quad (4.11)$$

pour tout $\theta \in W^{1,\infty}(\mathbb{R}^n; \mathbb{R}^n)$. De plus si Ω_0 est de classe \mathcal{C}^2 , alors on a :

$$J'(\Omega_0)(\theta) = \int_{\partial\Omega_0} \theta.n \left(\frac{\partial f}{\partial n} + Hf \right) ds \quad (4.12)$$

où $H := \text{div}(n)$ est la courbure moyenne de $\partial\Omega_0$

Cas particulier :

On veut calculer la dérivée de forme d'une contrainte de volume $V(\Omega) = \int_{\Omega} dx = C$. En appliquant ce qui précède on a :

$$V'(\Omega)(\theta) = \int_{\partial\Omega} \theta.n ds$$

4.1.5 Dérivation rapide : methode du lagrangien

Cette méthode (appelée méthode de dérivation rapide, développée par J. Cea) est une méthode formelle dont la justification reste sujette à interprétation car les étapes intermédiaires peuvent être mal justifiées du fait que l'on ne sait pas si le lagrangien est différentiable par rapport à la forme. Cependant cette méthode est particulièrement efficace et reste donc une méthode plus pratique qu'autre chose.

Nous allons donc utiliser cette méthode pour tous les calculs qui vont suivre en supposant donc que le lagrangien est différentiable par rapport à la forme. Elle impose donc d'écrire le lagrangien du problème que l'on dérivera par rapport à toutes ses variables, qu'elles soient des paramètres ou une forme. En posant que toutes ses dérivées doivent être nulles pour la stationnarité du lagrangien, on en déduira les différents éléments permettant d'obtenir l'équation d'état, l'équation adjointe et la dérivée de forme.

Voir les calculs des sections suivantes sur des cas concrets pour une compréhension complète de la méthode.

Remarque : La méthode d'utilisation des dérivée eulerienne et lagrangienne est une méthode mathématiquement plus rigoureuse : voir [2] pour plus de détail.

4.2 Cas de la conduction/membrane

On reprend le cas de la membrane de l'optimisation paramétrique, mais cette fois ci on ne va pas chercher à optimiser l'épaisseur de la membrane (que l'on supposera constante dans notre problème), mais la forme décrite par le bord $\partial\Omega_0 = \Gamma_N \cup \Gamma_D \cup \Gamma$ de la membrane.

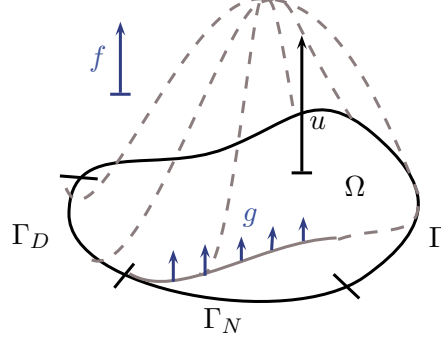


FIG. 4.1 – Schéma d'une membrane avec trois bords différents : dirichlet, Neumann et libre

La condition de Dirichlet sur Γ_D traduira la fixation de la membrane sur cette partie du bord, alors que la condition de Neumann sur Γ_N représentera la partie du contour de la membrane sur laquelle les force surfaciques sont exercées. Enfin la partie libre Γ sera exempte de toute force. On suppose par ailleurs que la membrane est soumise à une force volumique f qui peut représenter la force de gravité et le champ u qui représente le déplacement de la membrane en chaque point. Ce même problème peut aussi se décrire en conduction thermique où la condition Dirichlet représentera la mise à une certaine température du bord, le condition Neumann a un chauffage du bord (flux de chaleur traversant le bord), et la force volumique à un chauffage volumique. Le champ u représenterait alors la température en un point.

On rappelle l'équation aux dérivées partielles qui caractérise ce problème :

$$\begin{cases} -\Delta u = f & \text{dans } \Omega_0 \\ u = 0 & \text{dans } \Gamma_D \\ \frac{\partial u}{\partial n} = g & \text{dans } \Gamma_N \\ \frac{\partial u}{\partial n} = 0 & \text{dans } \Gamma \end{cases} \quad (4.13)$$

Etant donné que la forme de Ω va varier durant l'optimisation, il est nécessaire de connaître f et g sur un plus grand domaine contenant Ω . Notons D ce domaine de travail, ouvert borné de \mathbb{R}^n qui contient toutes les formes admissibles Ω .

Prenons $f \in L^2(D)$ et $g \in H^1(D)$ et supposons que $\Gamma_D \neq \emptyset$ pour ne pas à avoir à imposer une condition supplémentaire sur f et g pour l'unicité de u . Ainsi on est sûr que (4.13) admet une unique solution dans $H^1(\Omega)$.

4.2.1 Cas de la compliance

Si l'on définit notre fonction coût comme étant la compliance (travail extérieur fourni par le chargement) on a :

$$J(\Omega) = \int_{\Omega} f u dx + \int_{\Gamma_N} g u ds \quad (4.14)$$

Prenons notre ensemble admissible défini précédemment sous une condition de régularité. On fixe une constante $C > 0$ et on définit :

$$\mathcal{U}_{ad} = \mathcal{U}_{ad}(C) = \{ \Omega \in \mathcal{D}(\Omega_0), \quad d(\Omega, \Omega_0) \leq C, \quad |\Omega| = V \}$$

et notre problème d'optimisation de forme admettant, d'après le Théorème 1, au moins une forme optimale s'écrit donc :

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega) \quad (4.15)$$

En s'inspirant du lagrangien défini dans [1] (eq 11 p368), on introduit le lagrangien pour le problème de la membrane défini pour tout $(v, q) \in (H^1(\mathbb{R}^n; \mathbb{R}^n))^2$ par :

$$\begin{aligned} \mathcal{L}(\Omega, v, q) = & \int_{\Omega} f v dx + \int_{\Gamma_N} g v ds + \int_{\Omega} \nabla v \nabla q dx - \int_{\Omega} q f dx \\ & - \int_{\Gamma_N} q g ds - \int_{\Gamma_D} (q \frac{\partial v}{\partial n} + v \frac{\partial q}{\partial n}) ds \end{aligned} \quad (4.16)$$

Notons qu'ici, q est le multiplicateur de lagrange associé à l'équation d'état. On suppose que v et q ne dépendent pas de Ω (par la suite lorsque on évaluera v et q au point d'optimum avec u et p , alors u et p dépendront évidemment de Ω). On utilise la stationnarité du lagrangien (on note (u, p) ce point de stationnarité pour une forme Ω_0 donnée) pour trouver les conditions d'optimalités du problème de minimisation (4.1).

La dérivée partielle du lagrangien par rapport à q dans la direction $\phi \in H^1(\mathbb{R}^n, \mathbb{R}^n)$ est :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial q}(\Omega_0, u, p), \phi \rangle = & \int_{\Omega_0} \nabla u \nabla \phi dx - \int_{\Omega_0} \phi f dx - \int_{\Gamma_N} \phi g ds \\ & - \int_{\Gamma_D} (\phi \frac{\partial u}{\partial n} + u \frac{\partial \phi}{\partial n}) ds \end{aligned} \quad (4.17)$$

En intégrant par partie le premier terme du second membre on obtient :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial q}(\Omega_0, u, p), \phi \rangle = & - \int_{\Omega_0} (\Delta u + f) \phi dx + \int_{\Gamma_N} \phi (\frac{\partial u}{\partial n} - g) ds + \int_{\Gamma} \phi \frac{\partial u}{\partial n} ds \\ & - \int_{\Gamma_D} u \frac{\partial \phi}{\partial n} ds \\ = & 0 \end{aligned} \quad (4.18)$$

En prenant $\phi \in \mathcal{C}_c^\infty(\Omega_0)$ on retrouve l'équation d'état $-\Delta u = f$ dans Ω_0 . En faisant varier la trace de la fonction ϕ sur Γ_N , on retrouve la condition de Neumann de l'équation d'état $\frac{\partial u}{\partial n} = g$ sur Γ_N . De même en faisant varier $\frac{\partial \phi}{\partial n}$ sur Γ on retrouve la condition libre $\frac{\partial u}{\partial n} = 0$ sur Γ . Puis en faisant varier $\frac{\partial \phi}{\partial n}$ sur Γ_D on retrouve la condition de Dirichlet $u = 0$ sur Γ_D . On a donc retrouvé l'équation d'état dans son intégralité.

Pour trouver l'équation adjointe et ses conditions aux limites, il suffit de dériver le lagrangien par rapport à v :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial v}(\Omega_0, u, p), \phi \rangle = & \int_{\Omega_0} f \phi dx + \int_{\Gamma_N} g \phi ds + \int_{\Omega_0} \nabla \phi \nabla p dx \\ & - \int_{\Gamma_D} (p \frac{\partial \phi}{\partial n} + \phi \frac{\partial p}{\partial n}) \end{aligned}$$

En intégrant par partie le troisième terme du second membre on trouve :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial v}(\Omega_0, u, p), \phi \rangle = & \int_{\Omega_0} (f - \Delta p) \phi dx + \int_{\Gamma_N} \phi (\frac{\partial p}{\partial n} + g) ds + \int_{\Gamma} \phi \frac{\partial p}{\partial n} ds \\ & - \int_{\Gamma_D} (p \frac{\partial \phi}{\partial n}) ds \\ = & 0 \end{aligned} \quad (4.19)$$

En prenant ϕ à support compact dans Ω_0 on trouve l'équation adjointe :

$$-\Delta p = -f \quad \text{dans } \Omega_0$$

En faisant varier la trace de ϕ sur Γ_N on trouve la condition de Neumann adjointe :

$$\frac{\partial p}{\partial n} = -g \quad \text{dans } \Gamma_N$$

De même sur Γ on trouve :

$$\frac{\partial p}{\partial n} = 0 \quad \text{dans } \Gamma$$

Et en faisant varier $\frac{\partial \phi}{\partial n}$ sur Γ_D on a :

$$p = 0 \quad \text{dans } \Gamma_D$$

On a donc trouvé un problème aux limites bien posé pour l'état adjoint p :

$$\begin{cases} -\Delta p = -f & \text{dans } \Omega_0 \\ p = 0 & \text{dans } \Gamma_D \\ \frac{\partial p}{\partial n} = -g & \text{dans } \Gamma_N \\ \frac{\partial p}{\partial n} = 0 & \text{dans } \Gamma \end{cases} \quad (4.20)$$

On a donc $p = -u$. Le problème est dit auto-adjoint dans le cas de la compliance.

Lemme 4. Soient J et \mathcal{L} la fonction coût et le lagrangien définis précédemment. En supposant que J est dérivable en Ω_0 dans la direction θ on a :

$$J'(\Omega_0)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, p)(\theta)$$

Preuve :

Le résultat que l'on va prouver va nous permettre de trouver la dérivée de forme.

Nous allons différentier par rapport à une direction que l'on nomme $\theta \in W^{1,\infty}(\mathbb{R}, \mathbb{R})$ l'égalité suivante :

$$J(\Omega) = \mathcal{L}(\Omega, u(\Omega), q)$$

Cette égalité est due à la construction même du lagrangien de départ. En effet on a d'après l'équation (4.18) que :

$$\int_{\Omega_0} \nabla u \nabla q dx - \int_{\Omega_0} q f dx - \int_{\Gamma_N} q g ds - \int_{\Gamma_D} (q \frac{\partial u}{\partial n} + u \frac{\partial q}{\partial n}) ds = 0$$

ce qui donne l'égalité recherchée.

Dérivons cette égalité dans la direction θ en Ω_0 . A gauche on obtient $J'(\Omega_0)(\theta)$ par notation de la dérivée de forme dans une direction et à droite de l'égalité on obtient avec la règle de dérivation composée (en supposant que u et q sont différentiables par rapport à Ω) :

$$J'(\Omega_0)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, q)(\theta) + \left\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega_0, u, q), u'(\Omega_0)(\theta) \right\rangle + \left\langle \frac{\partial \mathcal{L}}{\partial q}(\Omega_0, u, q), q'(\Omega_0)(\theta) \right\rangle \quad (4.21)$$

Les deux derniers termes s'annulent car $\frac{\partial \mathcal{L}}{\partial q}(\Omega_0, u, q) = 0$ et $\frac{\partial \mathcal{L}}{\partial v}(\Omega_0, u, q) = 0$ d'après (4.18) et (4.19) qui ne dépendent pas de q . Il reste donc :

$$J'(\Omega_0)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, p)(\theta) \quad (4.22)$$

■

Nous sommes à présent en mesure d'explicitier la dérivée de forme de J et de déterminer une condition d'optimalité.

Regardons donc la dérivée du lagrangien par rapport à Ω dans la direction θ prise en (Ω_0, u, p) . En utilisant les Propositions 4.10 et 4.11 on obtient :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, p)(\theta) &= \int_{\partial \Omega_0} \theta.n(fu + \nabla u \nabla p - pf)ds + \int_{\Gamma_N} \theta.n\left(\frac{\partial(gu)}{\partial n} + Hgu\right)ds \\ &\quad - \int_{\Gamma_N} \theta.n\left(\frac{\partial(gp)}{\partial n} + Hgp\right)ds - \int_{\Gamma_D} \theta.n\left(\frac{\partial h}{\partial n} + Hh\right)ds \end{aligned} \quad (4.23)$$

avec $h = u\frac{\partial p}{\partial n} + p\frac{\partial u}{\partial n}$. Comme par ailleurs $p = -u$ et sachant que $u = 0$ sur Γ_D , on obtient :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, -u)(\theta) &= \int_{\partial \Omega_0} \theta.n(2fu - |\nabla u|^2)ds + 2 \int_{\Gamma_N} \theta.n\left(\frac{\partial(gu)}{\partial n} + Hgu\right)ds \\ &\quad - \int_{\Gamma_D} \theta.n\left(\frac{\partial h}{\partial n}\right)ds \end{aligned} \quad (4.24)$$

avec $h = -2u\frac{\partial u}{\partial n}$ et $\frac{\partial h}{\partial n} = -2|\nabla u|^2$ car $u = 0$ sur Γ_D et donc $\frac{\partial u}{\partial n} = 0$ sur Γ_D .

Finalement on en déduit l'expression un peu plus simplifiée suivante :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, -u)(\theta) &= \int_{\Gamma} \theta.n(2fu - |\nabla u|^2)ds + \int_{\Gamma_N} \theta.n\left(2fu - |\nabla u|^2 + 2\frac{\partial(gu)}{\partial n} + 2Hgu\right)ds \\ &\quad + \int_{\Gamma_D} \theta.n(|\nabla u|^2)ds \end{aligned} \quad (4.25)$$

On souhaite ne faire varier que le bord Γ . On pose donc $\theta.n = 0$ sur $\Gamma_D \cup \Gamma_N$ et il reste :

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, -u)(\theta) &= \int_{\Gamma} \theta.n(2fu - |\nabla u|^2)ds \\ &= 0 \end{aligned} \quad (4.26)$$

On en déduit la condition d'optimalité suivante (condition nécessaire seulement pour avoir l'optimum réalisé) :

$$2fu - |\nabla u|^2 = 0 \quad \text{sur } \Gamma \quad (4.27)$$

Remarque : Si l'on souhaite que seul Γ_N varie, on aurait par le même raisonnement la condition d'optimalité suivante :

$$2fu - |\nabla u|^2 + 2\frac{\partial(gu)}{\partial n} + 2Hgu = 0 \quad \text{sur } \Gamma_N \quad (4.28)$$

4.2.2 Cas de l'énergie de déformation

De la même manière que dans le cas de la compliance, il va falloir définir notre fonction coût, le lagrangien associé au problème de minimisation, et calculer la dérivée de forme et la condition d'optimalité.

Posons notre fonction coût comme étant :

$$J(\Omega) = \int_{\Omega} k(x)|\nabla u|^2 dx \quad (4.29)$$

où k permet de localiser. Vouloir minimiser cette fonction coût peut être vu comme vouloir uniformiser u dans la region $\{x \in \Omega, \quad k(x) \neq 0\}$. On suppose l'existence d'une forme optimale pour notre problème et ceux qui vont suivre.

Remarquons, étant toujours sur le problème de membrane, qu'en intégrant par partie J et en se servant de l'équation d'état ainsi qu'en supposant $k = 1$ on a :

$$J(\Omega) = \int_{\Omega} |\nabla u|^2 dx = \int_{\Omega} f u dx + \int_{\Gamma_N} g u ds \quad (4.30)$$

En disposant du même ensemble admissible (4.5) et du même problème de minimisation (4.15), on introduit le lagrangien, défini pour tout $(v, q) \in (H^1(\mathbb{R}^n; \mathbb{R}^n))^2$, par :

$$\begin{aligned} \mathcal{L}(\Omega, v, q) = & \frac{1}{2} \int_{\Omega} k(x) |\nabla v|^2 dx + \int_{\Omega} \nabla v \nabla q dx - \int_{\Omega} q f dx \\ & - \int_{\Gamma_N} q g ds - \int_{\Gamma_D} (q \frac{\partial v}{\partial n} + v \frac{\partial q}{\partial n}) ds \end{aligned} \quad (4.31)$$

Pour simplifier les choses on peut prendre k constante par morceaux, ce qui nous laisse tout de même le champ assez libre.

La dérivée du lagrangien (4.31) par rapport à q (dans la direction ϕ) est la même que dans le cas de la compliance, on retrouve donc similairement la même équation d'état. Par contre la dérivée par rapport à v (dans la direction ϕ) s'avère différente. On aura donc a priori un état adjoint différent du cas de la compliance (sauf si $k = 1$ bien évidemment, car le problème peut être vu du point de vue compliance).

Ainsi on a :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial v}(\Omega_0, u, p), \phi \rangle &= \int_{\Omega_0} k \nabla u \nabla \phi dx + \int_{\Omega_0} \nabla \phi \nabla p dx \\ &\quad - \int_{\Gamma_D} (p \frac{\partial \phi}{\partial n} + \phi \frac{\partial p}{\partial n}) \\ &= - \int_{\Omega_0} \text{div}(k(x) \nabla u) \phi dx + \int_{\partial \Omega_0} k(x) \frac{\partial u}{\partial n} \phi ds \\ &\quad - \int_{\Omega_0} (\Delta p) \phi dx + \int_{\partial \Omega_0} \frac{\partial p}{\partial n} \phi ds - \int_{\Gamma_D} (p \frac{\partial \phi}{\partial n} + q \frac{\partial p}{\partial n} ds) \\ &= - \int_{\Omega_0} \text{div}(k(x) \nabla u) \phi dx + \int_{\partial \Omega_0} k(x) \frac{\partial u}{\partial n} \phi ds \\ &\quad - \int_{\Omega_0} (\Delta p) \phi dx + \int_{\Gamma_N \cup \Gamma} \frac{\partial p}{\partial n} \phi ds - \int_{\Gamma_D} p \frac{\partial \phi}{\partial n} ds \end{aligned}$$

En prenant $\phi \in \mathcal{C}_c^\infty(\Omega_0) \cap H^1(\Omega_0)$, on obtient

$$-\Delta p = \text{div}(k(x) \nabla u) \quad \text{dans } \Omega_0$$

En prenant $\phi \in H^1(\Omega_0)$ nulle sur Γ_D avec $\frac{\partial \phi}{\partial n}$ quelconque on déduit que $p = 0$ sur Γ_D et en faisant varier ϕ sur $\Gamma_N \cup \Gamma$ on déduit que $\frac{\partial p}{\partial n} = -k \frac{\partial u}{\partial n} = -kg$ sur Γ_N et $\frac{\partial p}{\partial n} = 0$ sur Γ . Le problème adjoint se résume ainsi :

$$\begin{cases} -\Delta p = \text{div}(k(x) \nabla u) & \text{dans } \Omega_0 \\ p = 0 & \text{sur } \Gamma_D \\ \frac{\partial p}{\partial n} = -kg & \text{sur } \Gamma_N \\ \frac{\partial p}{\partial n} = 0 & \text{sur } \Gamma \end{cases} \quad (4.32)$$

En partant de (4.22), on calcul la dérivée de forme :

$$\begin{aligned}
J'(\Omega_0)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, p)(\theta) &= \int_{\partial \Omega_0} \theta.n \left(\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - pf \right) ds \\
&\quad - \int_{\Gamma_N} \theta.n \left(\frac{\partial(gp)}{\partial n} + H(gp) \right) ds - \int_{\Gamma_D} \theta.n \left(\frac{\partial h}{\partial n} + Hh \right) ds \quad (4.33)
\end{aligned}$$

où $h = p \frac{\partial u}{\partial n} + u \frac{\partial p}{\partial n}$. Or $u = p = 0$ sur Γ_D et donc $h = 0$ et $\frac{\partial h}{\partial n} = 2 \frac{\partial u}{\partial n} \frac{\partial p}{\partial n}$ sur Γ_D ce qui donne :

$$\begin{aligned}
J'(\Omega_0)(\theta) &= \int_{\partial \Omega_0} \theta.n \left(\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - pf \right) ds - \int_{\Gamma_N} \theta.n \left(\frac{\partial(gp)}{\partial n} + H(gp) \right) ds \\
&\quad - 2 \int_{\Gamma_D} \theta.n \left(\frac{\partial u}{\partial n} \frac{\partial p}{\partial n} \right) ds \\
&= \int_{\Gamma} \theta.n \left(\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - pf \right) ds + \int_{\Gamma_N} \theta.n \left(\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - pf \right. \\
&\quad \left. - \frac{\partial(gp)}{\partial n} - H(gp) \right) ds + \int_{\Gamma_D} \theta.n \left(\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - 2 \frac{\partial u}{\partial n} \frac{\partial p}{\partial n} \right) ds \quad (4.34)
\end{aligned}$$

En ne faisant varier que Γ (mise à zéro de $\theta.n$ sur $\Gamma_N \cup \Gamma_D$) on a finalement :

$$\frac{\partial \mathcal{L}}{\partial \Omega}(\Omega_0, u, p)(\theta) = J'(\Omega_0)(\theta) = \int_{\Gamma} \theta.n \left(\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - pf \right) ds \quad (4.35)$$

Donc notre condition d'optimalité est :

$$\frac{1}{2} k |\nabla u|^2 + \nabla u \nabla p - pf = 0 \quad \text{sur } \Gamma \quad (4.36)$$

4.2.3 Anneau et forces suiveuses

Toujours dans le cas de la conduction, on dispose d'un bord Neumann pouvant varier (dit à force suiveuse si l'on parle d'un problème de membrane) et d'un bord Dirichlet : schéma de la figure ci dessous 4.2.

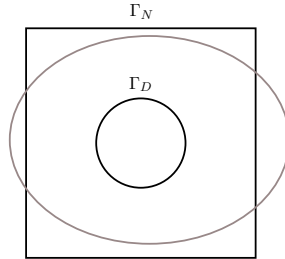


FIG. 4.2 – Schéma de la conduction sur un carré troué

On prend $f = 0$, on chauffe le bord Neumann en imposant $g = 1$ sur Γ_N et on maintient la température à 0 sur Γ_D . Le problème est alors le suivant :

$$\begin{cases} -\Delta u = 0 & \text{dans } \Omega_0 \\ u = 0 & \text{sur } \Gamma_D \\ \frac{\partial u}{\partial n} = 1 & \text{sur } \Gamma_N \end{cases}$$

La forme varie par la variation du bord extérieur. La fonction coût que l'on minimise est la compliance $J(\Omega) = \int_{\Gamma_N} u$. La forme aura donc tendance à diminuer de volume et éventuellement former un cercle pour traduire la symétrie par rotation de Γ_D .

Cas Neumann à l'extérieur

En supposant alors (dans toute cette section) que notre forme optimale de volume V_0 est un anneau (chose qui se confirmera par les résultats des cas tests dans la section 4.5.3), on peut alors calculer la solution exacte analytiquement en se plaçant en coordonnées polaire : $u = u(r)$. On a alors $\Omega_0 = \mathcal{B}(O, r_2)$, $0 < r_1 < r_2$. L'équation d'état se résume alors à :

$$\begin{cases} u'' + \frac{1}{r}u' = 0 & \text{dans } \Omega_0 \\ u(r_1) = 0 \\ u'(r_2) = 1 \end{cases} \quad (4.37)$$

En posant $v = u'$ on obtient $v = \frac{C_1}{r}$ donc $u = C_1 \ln(r) + C_2$. La condition en $[r = r_2]$ donne $C_1 = r_2$ et la condition en r_1 donne $C_2 = -r_2 \ln(r_1)$. D'où au final :

$$u = r_2 \ln\left(\frac{r}{r_1}\right)$$

avec $0 < r_1 \leq r \leq r_2$. La dérivée de la fonction coût s'écrit (cf (4.28)) :

$$J'(\Omega)(\theta) = \int_{\Gamma} \theta \cdot n(-|\nabla u|^2 + 2\frac{\partial u}{\partial n} + 2Hu)$$

or $H = \frac{1}{r_2}$ et $|\nabla u|^2 = 1$ (puisque $|\nabla u|^2 = |\frac{\partial u}{\partial n}|^2 + |\frac{\partial u}{\partial t}|^2 = |\frac{\partial u}{\partial n}|^2$ vu que $u = u(r)$). La condition d'optimalité en tenant compte de la contrainte volumique (voir la partie 4.4.2 pour cette contrainte) s'écrit alors :

$$-1 + 2 + \frac{2u}{r_2} + l = 0$$

ce qui donne

$$1 + l + 2\ln\left(\frac{r_2}{r_1}\right) = 0$$

ce qui nous donne la valeur exacte que le multiplicateur de lagrange doit prendre pour que la condition d'optimalité soit satisfaite : $l = -1 - 2\ln\left(\frac{r_2}{r_1}\right)$ (pour r_1 et r_2 tel que la contrainte volumique soit satisfaite).

De plus si on regarde directement la fonction coût $J = \int_{\Gamma} u$, et qu'on l'intègre on obtient :

$$J(r_2) = \int_{\{r=r_2\}} u(r_2) d\theta = 2\pi r_2^2 \ln\left(\frac{r_2}{r_1}\right)$$

En cherchant le minimum de cette fonction sans restriction sur l'ensemble des r_2 admissibles on trouve $r_2 = r_1 e^{-\frac{1}{2}} < r_1$. Or $r_2 > r_1$. En traçant la fonction J en fonction de r_2 pour $r_1 = 2$ sur $[2, 3]$ on obtient la figure (4.3).

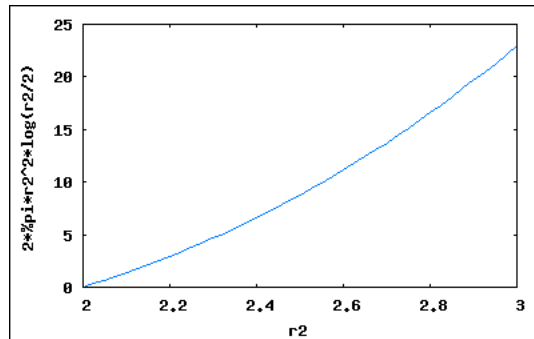


FIG. 4.3 – Tracé de la fonction coût avec $r_1 = 2$ sur $[2, 3]$ en fonction de r_2

On observe donc dans l'intervalle $r_2 \in [2, 3]$ (l'intervalle $r_2 \in [0, 2]$ est étudié après) une diminution de la fonction coût lorsque r_2 diminue jusqu'au seuil de $r_2 = r_1$. Remarquons que la dérivation de la

fonction coût par rapport à r_2 donne :

$$\frac{d}{dr}[J + lV]_{|r=r_2} = 2\pi r_2 [2\ln(\frac{r_2}{r_1}) + 1]$$

et la dérivation de la contrainte volumique donne :

$$V'(r_2) = (\pi r_2^2 - \pi r_1^2)' = 2\pi r_2$$

et l'égalité $J' + lV' = 0$ nous redonne la condition d'optimalité trouvée précédemment.

Il faut donc regarder le cas $r_2 < r_1$ pour voir si un minimum autre que $r_2 = r_1$ existe.

Cas Neumann à l'intérieur

En plaçant le bord Neumann à l'intérieur ($r_2 < r_1$) et en résolvant la même équation d'état on obtient la fonction $u = -r_2 \ln(\frac{r}{r_1})$ (car dans le calcul intervient $\frac{\partial u}{\partial n} = -\frac{\partial u}{\partial r}$). Comme $r < r_1$ on a bien $u > 0$. La fonction coût s'écrit alors $J(r_2) = -2\pi r_2^2 \ln(\frac{r_2}{r_1})$. Traçons donc l'évolution de la valeur absolue de la fonction coût lorsque $r_2 \in [0, 3]$ sur la figure 4.4 (ce qui représentera les valeurs de la fonction coût que r_2 soit plus petit que r_1 ou plus grand sans restriction).

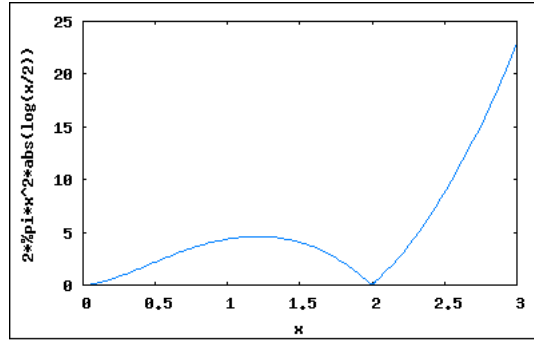


FIG. 4.4 – Tracé de la fonction coût avec $r_2 \in [0, 3]$ et $r_1 = 2$

Il n'y a pas de minimum autre que sur les extrémités $r_2 = 0$ et $r_2 = r_1$. La valeur $r_2 = r_1 e^{-\frac{1}{2}}$ est ici un maximum. Ainsi on a donc trouvé, si la forme optimale est un anneau, l'anneau optimal pour notre fonction coût compliance : il suffit pour un volume donné de prendre l'anneau correspondant si l'on dispose d'une contrainte volumique (en effet si l'on souhaite imposer un volume à la forme alors, étant donné que r_1 est fixé, le volume ne dépend plus que de r_2 : un volume fixe signifie donc un r_2 fixe et le problème de recherche d'une meilleure forme à volume fixé devient trivial). Il faut alors étendre le problème et une solution est de ne pas tenir compte de la topologie (voir la section suivante). Sans cette contrainte de volume, l'anneau optimal est soit réduit à un anneau ultra fin, soit à un disque plein avec un trou minuscule en son centre.

Existence d'une meilleure forme sans tenir compte de la topologie

On est toujours sur le même problème de la conduction dans un anneau muni de deux bords : un Dirichlet et un Neumann qui peut varier (avec $g = 1$). Toujours avec la compliance, on va explicitement calculer la meilleure forme pour se problème sans tenir compte de la topologie : i.e. que le bord Dirichlet soit à l'intérieur de l'anneau, ou à l'extérieur.

Proposition 3. Soit Ω un ouvert borné en forme d'anneau (de dimension 2). On suppose que notre problème est radial ($u=u(r)$) et sans force volumique ($f=0$) : cas du problème (4.37). On suppose de plus que g est constant. La contrainte volumique étant donnée par $V(\Omega) = V_0$.

Alors le problème (4.37) admet une forme optimale (dans la classe des solutions radiales) qui est un anneau dont le bord Dirichlet est intérieur à l'anneau si $g > 0$, extérieur si $g < 0$.

Preuve :

Soit $g(r) = g$ constante, $f = 0$ et u radial. Remarquons que u radial nous donne que $u \in H^2(\Omega)$ avec (4.37). Le problème (4.37) est alors bien posé. Il va donc falloir calculer les fonctions coût associées à ces deux formes de même volume et voir quelle forme est la meilleure.

Ainsi, si l'on fixe r_1 , le volume est donc $|\pi r_1^2 - \pi r_2^2|$. Fixons le volume cible à $V_0 = |\pi r_1^2 - \pi r_2^2| > 0$. Soient $r_2 < r_3$ les deux solutions de $V_0 = |\pi r_1^2 - \pi r^2|$. Dans le cas où dirichlet est extérieur, $r_2 < r_1$ et on a :

$$J(r_2) = \int_{r=r_2} g(r_2) u(r_2) d\theta = -2\pi r_2^2 \ln\left(\frac{r_2}{r_1}\right) g$$

Cherchons r_3 tel que $\pi r_3^2 - \pi r_1^2 = \pi r_1^2 - \pi r_2^2$. Cela donne $r_3 = \sqrt{2r_1^2 - r_2^2}$. Et le calcul de $J(r_3)$ donne :

$$J(r_3) = 2\pi(2r_1^2 - r_2^2) \left(\frac{1}{2} \ln(2r_1^2 - r_2^2) - \ln(r_1) \right) g$$

Regardons le signe de la différence $\text{Diff} = \frac{J(r_3) - J(r_2)}{2\pi}$. Posons

$$\text{Diff}(r) = (2r_1^2 - r_2^2) \left(\frac{1}{2} \ln(2r_1^2 - r_2^2) - \ln(r_1) \right) g + r^2 \ln\left(\frac{r}{r_1}\right) g$$

En dérivant par rapport à r on obtient :

$$\text{Diff}'(r) = r \ln\left(\frac{r^2}{2r_1^2 - r^2}\right)$$

Sachant que $0 < r < r_1$, on a $\frac{r^2}{2r_1^2 - r^2} < 1$. donc $\text{Diff}' < 0$ si $g > 0$ donc $\text{Diff}(r) \geq \text{Diff}(r_1) = 0$

Donc si $g > 0$ alors $\text{Diff} > 0$ et la forme avec le bord dirichlet à l'extérieur est meilleure à volume et r_1 fixés sinon c'est celle avec le bord dirichlet à l'intérieur qui est meilleure.

■

Contre exemple de non unicité

On se demande ici s'il y a unicité d'une meilleure solution, toutes topologies confondues. Pour ce faire on va regarder s'il existe une fonction g tel qu'il existe deux formes différentes (par leurs topologies) minimisant globalement cette fonction coût.

Proposition 4. *Soit Ω un ouvert borné en forme d'anneau dans le cas du problème (4.37). On suppose toujours que notre problème est radial et sans force volumique. On suppose de plus que g est radiale et constante par morceaux et que la contrainte volumique est donné par $V(\Omega) = V_0$.*

Alors on peut choisir g tel qu'il y ait pas unicité du problème de minimisation de la compliance : i.e. il existe deux formes différentes de même volume tel que la compliance est minimum.

Preuve :

Cas de la dimension 2 :

On suppose $r_2 < r_3$. Avec les notations précédentes, sur les deux formes de topologies différentes du paragraphe précédent, on a soit $g = g(r_2)$ soit $g = g(r_3)$ et on a aussi :

$$J(r_2) = -2\pi r_2^2 \ln\left(\frac{r_2}{r_1}\right) g(r_2)$$

et

$$J(r_3) = 2\pi r_3^2 \ln\left(\frac{r_3}{r_1}\right) g(r_3)$$

Ainsi, si l'on veut $J = J(r_2) = J(r_3)$, étant donné qu'à volume fixé r_2 et r_3 sont fixés, il reste à trouver g vérifiant cette égalité. Il suffit de prendre :

$$g(r) = \frac{J}{-2\pi r_2^2 \ln\left(\frac{r_2}{r_1}\right)}, \quad r \in [0, r_1]$$

$$g(r) = \frac{J}{2\pi r_3^2 \ln(\frac{r_3}{r_1})}, \quad r \in [r_1, r_3]$$

Avec un tel g , valant $g(r_2)$ en r_2 et $g(r_3)$ en r_3 , on obtient deux formes différentes (car de topologies différentes), réalisant le même minimum d'une même fonction coût sur le même problème. Il n'y a donc pas unicité de la forme optimale pour tous les problèmes. Remarquons qu'il existe même une infinité de couples $(g(r_2), g(r_3))$ réalisant l'égalité recherchée.

Cas de la dimension n :

Si on se place en dimension n sur le même problème de conduction (par exemple en dimension 3 on va être confronté au problème de conduction radiale entre deux sphères) on a la fonction coût qui vaut lorsque le bord Dirichlet est à l'intérieur :

$$J(\Omega) = \int_{\Gamma_D} g u_1 ds = g(r_2) \int_{\Gamma_D} u_1 ds$$

et quand il est à l'extérieur :

$$J(\Omega) = \int_{\Gamma_D} g u_2 ds = g(r_3) \int_{\Gamma_D} u_2 ds$$

Ainsi si l'on veut l'égalité des fonctions coût dans ces deux formes on prend

$$g(r_2) = \alpha \int_{\Gamma_D} u_2$$

et

$$g(r_3) = \alpha \int_{\Gamma_D} u_1$$

■

4.2.4 Anneau et forces volumiques

On dispose d'un bord libre de toute force qui variera seul, et d'un bord Dirichlet. Le bord Dirichlet sera d'abord intérieur puis extérieur. On reprend le schéma 4.2 de la section précédente.

Cas $f = 1$

On va donc essayer de trouver la solution explicite du problème sur un anneau avec $f = 1$ et d'en déduire la forme optimale. Le problème étant dans ce cas symétrique par rotation autour du centre de l'anneau, la fonction u s'écrit ne dépend que de r . Ainsi le problème s'écrit en coordonnées radiales :

$$\begin{cases} -u'' - \frac{1}{r}u' = 1 & \text{dans } \Omega_0 \\ u(r_1) = 0 \\ u'(r_2) = 0 \end{cases} \quad (4.38)$$

La solution de ce problème s'écrit :

$$u(r) = -\frac{r_2^2}{2} \ln\left(\frac{r_1}{r}\right) - \frac{1}{4}(r^2 - r_1^2)$$

Ainsi on peut calculer

$$J(r_2) = \int_{\Omega} f u dx = \int_{\Omega} u dx = \int_{\theta \in [0, 2\pi]} \int_{r \in [r_1, r_2]} u(r) r dr d\theta$$

ce qui donne à r_1 fixé :

$$J(r_2) = \pi \frac{r_2^4}{2} \ln(r_2) - \pi \frac{r_1^2 r_2^2}{2} \ln(r_1) - \pi \frac{r_2^2}{4} (r_2^2 - r_1^2) - \pi \frac{r_2^2}{2} (r_2^2 - r_1^2) \ln(r_1) - \frac{\pi}{8} (r_2^4 - r_1^4) + \pi \frac{r_1^2}{4} (r_2^2 - r_1^2)$$

ou plus simplement :

$$J(r_2) = 2\pi \left[-\frac{\ln(\frac{r_1}{r_2})r_2^4}{4} - \frac{3r_2^4}{16} + \frac{2r_1^2r_2^2 - r_1^4}{16} + \frac{r_1^2r_2^2}{8} \right]$$

Lorsqu'on trace cette fonction avec $r_1 = 1.5$ et $r_1 < r_2 < 3$ (si on trace de même $r_2 < r_1$, il faut prendre l'opposé de la fonction coût : car si l'on considère ce cas là, la solution d'état vérifie la même équation d'état, et est donc égale à la précédente mais la fonction coût s'intègre de r_2 à r_1 et non l'inverse, ce qui change le signe). On trace donc la valeur absolue de J et on obtient la figure 4.5 :

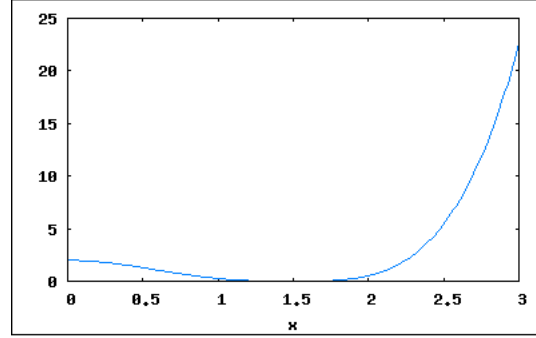


FIG. 4.5 – Tracé de la fonction coût sur $[0, 3]$ avec $r_1 = 1.5$

Ainsi la forme aura tendance, que $r_2 > r_1$ ou $r_2 < r_1$, à converger vers un anneau ultra fin si la contrainte sur le volume est faible.

4.3 Cas de l'élasticité linéarisée

4.3.1 Le problème aux limites

On s'inspire grandement dans cette partie de [1]. Soit $\Omega \subset \mathbb{R}^n$ ($n = 2$ ou 3) un ouvert borné occupé par un matériau élastique linéaire isotrope de loi de Hooke A . On rappelle que l'on a \forall matrice symétrique ξ :

$$A\xi = 2\mu\xi + \lambda(\text{Tr}\xi)Id \quad (4.39)$$

où μ et λ sont les coefficients de Lamé.

Le bord est fait de trois parties disjointes $\partial\Omega = \Gamma \cup \Gamma_N \cup \Gamma_D$. On suppose tout d'abord que toutes les frontières sont autorisées à varier. Il sera explicitement indiqué lorsque cela ne sera pas le cas. On note f la fonction vectorielle des forces volumiques et g le vecteur des forces surfaciques. Le champ déplacement u dans Ω est la solution du système de l'élasticité linéarisé suivant :

$$\begin{cases} -\text{div}(Ae(u)) = f & \text{dans } \Omega \\ u = 0 & \text{sur } \Gamma_D \\ (Ae(u)).n = g & \text{sur } \Gamma_N \\ (Ae(u)).n = 0 & \text{sur } \Gamma \end{cases} \quad (4.40)$$

De même qu'en conduction, étant donné que la forme de Ω va varier durant l'optimisation, il est nécessaire de connaître f et g sur un plus grand domaine contenant Ω . Notons D ce domaine de travail, ouvert borné de \mathbb{R}^n qui contient toutes les formes admissibles Ω .

Prenons $f \in L^2(D; \mathbb{R}^n)$ et $g \in H^1(D; \mathbb{R}^n)$ et supposons que $\Gamma_D \neq \emptyset$ pour ne pas à avoir à imposer une condition supplémentaire sur f et g pour l'unicité de u . Ainsi on est sûr que (4.40) admet une unique solution u dans $H^1(\Omega; \mathbb{R}^n)$ toujours si l'on se place soit dans un espace discrétisé ou bien si l'on ajoute une condition de régularité.

4.3.2 Cas de la compliance

Dans le cas de la compliance, la fonction coût s'écrit bien évidemment

$$J(\Omega) = \int_{\Omega} f \cdot u dx + \int_{\Gamma_N} g \cdot u ds$$

Remarquons que là aussi on peut écrire par simple intégration par partie et en utilisant (4.40) que

$$J(\Omega) = \int_{\Omega} Ae(u) : e(u) dx$$

Définissons l'ensemble des formes admissibles par

$$\mathcal{U}_{ad} = \{\Omega \text{ ouvert}, \Omega \subset D, |\Omega| = V\} \quad (4.41)$$

que l'on muni de contraintes de régularités ou géométriques ou topologique (comme vu précédemment) pour assurer que le problème de minimisation suivant est bien posé :

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega)$$

Pour d'autres résultats sur l'établissement de problème bien posé, le lecteur pourra consulter [2].

On introduit le lagrangien (de la même manière que dans le cas de la conduction) pour tout $(v, q) \in (H^1(\mathbb{R}^n; \mathbb{R}^n))^2$ par :

$$\begin{aligned} \mathcal{L}(\Omega, v, q) = & \int_{\Omega} f \cdot v dx + \int_{\Gamma_N} g \cdot v ds + \int_{\Omega} Ae(v) : e(q) dx - \int_{\Omega} q \cdot f dx \\ & - \int_{\Gamma_N} q \cdot g ds - \int_{\Gamma_D} (q \cdot (Ae(v)n) + v \cdot (Ae(q)n)) ds \end{aligned} \quad (4.42)$$

La dérivée partielle du lagrangien par rapport à q dans la direction $\phi \in H^1(\mathbb{R}^n, \mathbb{R}^n)$ en (Ω, u, p) , après intégration par partie est :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial q}(\Omega, u, p), \phi \rangle &= - \int_{\Omega} \phi \cdot (\text{div}(Ae(u)) + f) dx + \int_{\Gamma_N} \phi \cdot (Ae(u)n - g) ds + \int_{\Gamma} \phi \cdot (Ae(u)n) ds \\ &\quad - \int_{\Gamma_D} u \cdot (Ae(\phi)n) ds \\ &= 0 \end{aligned} \quad (4.43)$$

En choisissant convenablement ϕ (de la même manière que lors de la conduction), on retrouve l'équation d'état.

La dérivée partielle du lagrangien par rapport à v dans la direction $\phi \in H^1(\mathbb{R}^n, \mathbb{R}^n)$ en (Ω, u, p) , après intégration par partie est :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, u, p), \phi \rangle &= \int_{\Omega} (f - \text{div}(Ae(p))) \cdot \phi dx + \int_{\Gamma_N} \phi \cdot (Ae(p)n + g) ds + \int_{\Gamma_N} \phi \cdot Ae(p)n ds \\ &\quad + \int_{\Gamma_D} (\phi \cdot g - p \cdot Ae(\phi)n) ds \\ &= 0 \end{aligned} \quad (4.44)$$

En choisissant de même ϕ convenablement, on obtient l'équation de l'état adjoint :

$$\begin{cases} -\text{div}(Ae(p)) = -f & \text{dans } \Omega \\ p = 0 & \text{sur } \Gamma_D \\ Ae(p)n = -g & \text{sur } \Gamma_N \\ Ae(p)n = 0 & \text{sur } \Gamma \end{cases} \quad (4.45)$$

Notons que dans ce cas nous avons aussi $p = -u$: problème auto-adjoint.

Dérivons maintenant par rapport au domaine pour obtenir la dérivée de forme :

$$\begin{aligned} \langle \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u, p), \phi \rangle &= \int_{\partial \Omega} \theta \cdot n (fu + Ae(u) : e(p) - pf) ds + \int_{\Gamma_N} \theta \cdot n (\frac{\partial(g \cdot u)}{\partial n} + Hg \cdot u) ds \\ &\quad - \int_{\Gamma_N} \theta \cdot n (\frac{\partial(g \cdot p)}{\partial n} + Hg \cdot p) ds - \int_{\Gamma_D} (\theta \cdot n (\frac{\partial h}{\partial n} + Hh)) ds \\ &= 0 \end{aligned} \quad (4.46)$$

avec $h = u \cdot Ae(p)n + p \cdot Ae(u)n$. En prenant en compte que $u = p = 0$ sur Γ_D on a $h = 0$ et $\frac{\partial u}{\partial \tau} = \frac{\partial p}{\partial \tau} = 0$ sur Γ_D ce qui donne :

$$Ae(u) \cdot e(p) = \mu \frac{\partial u}{\partial n} \cdot \frac{\partial p}{\partial n} + (\mu + \lambda) (\frac{\partial u}{\partial n} \cdot n) (\frac{\partial p}{\partial n} \cdot n) \quad \text{sur } \Gamma_D \quad (4.47)$$

Donc on a :

$$\frac{\partial h}{\partial n} = 2Ae(u) \cdot e(p) \quad \text{sur } \Gamma_D \quad (4.48)$$

d'où :

$$\begin{aligned}
\langle \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u, p), \phi \rangle &= \int_{\Gamma} \theta.n(f.u + Ae(u).e(p) - pf)ds + \int_{\Gamma_N} \theta.n(f.u + Ae(u) : e(p) - pf \\
&\quad - \frac{\partial g.(p-u)}{\partial n} - Hg.(p-u))ds \\
&\quad + \int_{\Gamma_D} \theta.n(f.u - Ae(u) : e(p) + \frac{\partial g.u}{\partial n})ds
\end{aligned} \tag{4.49}$$

$$\begin{aligned}
&= \int_{\Gamma} \theta.n(2f.u + Ae(u) : e(u))ds + \int_{\Gamma_N} \theta.n(2f.u + Ae(u) : e(u) \\
&\quad + 2(\frac{\partial g.u}{\partial n} - Hg.u))ds \\
&\quad + \int_{\Gamma_D} \theta.n(f.u - Ae(u) : e(u) + \frac{\partial g.u}{\partial n})ds
\end{aligned} \tag{4.50}$$

En prenant $\theta.n$ nul sur $\Gamma_D \cup \Gamma_N$ (pour les forcer à ne pas varier au cours de l'optimisation de forme) on obtient :

$$J'(\Omega)(\theta) = \int_{\Gamma} \theta.n(2f.u + Ae(u) : e(u))ds \tag{4.51}$$

La condition d'optimalité s'écrit donc :

$$2f.u + Ae(u) : e(u) = 0 \quad \text{sur } \Gamma \tag{4.52}$$

4.3.3 Cas de l'énergie de déformation

On pose notre fonction coût comme

$$J(\Omega) = \int_{\Omega} k(x)Ae(u) : e(u)dx \tag{4.53}$$

représentant l'énergie élastique de déformation de la forme.

L'ensemble des formes admissibles est

$$\mathcal{U}_{ad} = \{\Omega \text{ ouvert}, \Omega \subset D, |\Omega| = V\} \tag{4.54}$$

que l'on muni de contraintes de régularités ou géométriques ou topologique (comme vu précédemment) pour assurer que le problème de minimisation suivant est bien posé :

$$\inf_{\Omega \in \mathcal{U}_{ad}} J(\Omega)$$

Le lagrangien est construit de la même façon pour tout $(v, q) \in (H^1(\mathbb{R}^n; \mathbb{R}^n))^2$ par :

$$\begin{aligned}
\mathcal{L}(\Omega, v, q) &= \frac{1}{2} \int_{\Omega} k(x)Ae(v) : e(v)dx + \int_{\Omega} Ae(v) : e(q)dx - \int_{\Omega} q.fdx \\
&\quad - \int_{\Gamma_N} q.gds - \int_{\Gamma_D} (q.(Ae(v)n) + v.(Ae(q)n))ds
\end{aligned} \tag{4.55}$$

En dérivant par rapport à q on obtient l'équation d'état (pour les calculs similaires voir (4.43)). En dérivant par rapport à v dans la direction $\phi \in H^1(\mathbb{R}^n, \mathbb{R}^n)$ en (Ω, u, p) on obtient :

$$\begin{aligned}
\langle \frac{\partial \mathcal{L}}{\partial v}(\Omega, u, p), \phi \rangle &= \int_{\Omega} k(x)Ae(u) : e(\phi)dx + \int_{\Omega} Ae(\phi) : e(p)dx \\
&\quad - \int_{\Gamma_D} (p.Ae(\phi)n + \phi.Ae(p)n)ds \\
&= - \int_{\Omega} \text{div}(k(x)Ae(u)).\phi dx + \int_{\partial \Omega} k(x)(Ae(u)n).\phi ds \\
&\quad - \int_{\Omega} \text{div}(Ae(p)).\phi dx + \int_{\Gamma_N \cup \Gamma} (Ae(p)n).\phi ds - \int_{\Gamma_D} p.(Ae(\phi)n)ds
\end{aligned} \tag{4.56}$$

En effet, étant donné que l'opérateur A est symétrique, on a :

$$\begin{aligned}
J(u+h) &= \int_{\Omega} kAe(u+h) : e(u+h) dx \\
&= \int_{\Omega} kAe(u) : e(u) dx + \int_{\Omega} kAe(u) : e(h) dx + \int_{\Omega} kAe(h) : e(u) dx + o(\|h\|_{H^1}^2) \\
&= \int_{\Omega} kAe(u) : e(u) dx + 2 \int_{\Omega} kAe(u) : e(h) dx + o(\|h\|_{H^1}^2)
\end{aligned}$$

Ensuite, toujours en prenant ϕ adéquatement, i.e. $\phi \in \mathcal{C}_c^\infty$ on a $-\text{div}(Ae(p)) = \text{div}(k(x)Ae(u))$ dans Ω . En prenant ensuite $\phi \in H^1(\Omega)^2$ tel que $\phi = 0$ sur Γ_D alors on a $Ae(p)n = -kAe(u)n = -kg$ sur Γ_N et $Ae(p)n = 0$ sur Γ . Et enfin, en prenant $\phi = 0$ sur Γ_D mais pas $Ae(\phi)n$ on obtient que $p = 0$ sur Γ_D . L'équation adjointe s'écrit finalement :

$$\left\{ \begin{array}{ll} -\text{div}(Ae(p)) = \text{div}(k(x)Ae(u)) & \text{dans } \Omega_0 \\ p = 0 & \text{sur } \Gamma_D \\ Ae(p)n = -kg & \text{sur } \Gamma_N \\ Ae(p)n = 0 & \text{sur } \Gamma \end{array} \right. \quad (4.57)$$

Dérivons ensuite par rapport au domaine pour obtenir la dérivée de forme :

$$\begin{aligned}
\langle \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u, p), \phi \rangle &= \int_{\partial \Omega} \theta.n(\frac{1}{2}k(x)Ae(u) : e(u) + Ae(u) : e(p) - pf) ds - \int_{\Gamma_N} \theta.n(\frac{\partial g.p}{\partial n} + Hg.p) ds \\
&\quad - \int_{\Gamma_D} (\theta.n(\frac{\partial h}{\partial n} + Hh) ds
\end{aligned} \quad (4.58)$$

$$\quad (4.59)$$

avec $h = u.Ae(p)n + p.Ae(u)n$. En simplifiant de la même manière que pour la compliance on obtient la dérivée de forme suivante :

$$\begin{aligned}
\langle \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u, p), \phi \rangle &= \int_{\Gamma} \theta.n(\frac{1}{2}k(x)Ae(u) : e(u) + Ae(u) : e(p) - pf) ds + \int_{\Gamma_N} \theta.n(k(x)|\nabla u|^2 + Ae(u) : e(p) \\
&\quad - pf - \frac{\partial g.p}{\partial n} - Hg.p) ds \\
&\quad + \int_{\Gamma_D} \theta.n(k(x)Ae(u) : e(u) - Ae(u) : e(p)) ds
\end{aligned} \quad (4.60)$$

En prenant $\theta.n$ nul sur $\Gamma_D \cup \Gamma_N$, il vient :

$$\langle \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, u, p), \phi \rangle = \int_{\Gamma} \theta.n(\frac{1}{2}k(x)Ae(u) : e(u) + Ae(u) : e(p) - pf) ds \quad (4.61)$$

En annulant cette dérivée de forme par stationnarité du lagrangien on obtient la condition d'optimalité :

$$\frac{1}{2}k(x)Ae(u) : e(u) + Ae(u) : e(p) - pf = 0 \quad \text{sur } \Gamma \quad (4.62)$$

4.4 Algorithmes et numérique

4.4.1 La méthode du gradient

La méthode du gradient est un outil assez efficace à utiliser en optimisation mais ici on souhaite l'appliquer aux variations de formes en optimisation de forme. (Tiré de [7])

Soit F une fonction d'un espace de Hilbert H dans \mathbb{R} . La méthode du gradient consiste en la construction d'une suite $(x_n)_{n \geq 0} \subset H$ avec :

$$x_{n+1} = x_n - h_n d_n$$

ou h_n est un reel positif petit et d_n est la direction de descente définie par :

$$(d_n, y)_H = \langle F'(x_n), y \rangle_{H^*, H}$$

pour tout $y \in H$

On identifie généralement le dual H^* à son espace H et donc d_n n'est rien d'autre que le gradient de F note $F'(x_n)$. Si $F'(x_n)$ n'est pas nul, alors pour h_n suffisamment petit on a $F(x_{n+1}) < F(x_n)$. En initialisant l'algorithme avec n'importe quel élément $x_0 \in H$, et si F est fortement convexe, x_n converge vers la solution optimale x^* du probleme :

$$\min_{x \in H} F(x)$$

En optimisation de forme, l'espace H n'est plus un espace de Hilbert mais un sous ensemble des ouverts de \mathbb{R}^n (avec habituellement $n = 2$ ou 3) et n'a pas de structure topologique pour la différentiation. Néanmoins on va pouvoir appliquer la méthode de gradient à deux conditions : définir ce qu'est la variation de forme (ce qui a été fait précédemment), et munir l'espace où s'effectue les variations de forme d'une structure Hilbertienne ou au moins Banach.

Ainsi il suffit pour appliquer la méthode de gradient, d'associer à la dérivée de forme $J'(\Omega)$ une direction de descente. On muni donc l'espace des champs de vecteurs de Ω dans \mathbb{R}^n avec une structure hilbertienne $H^1(\Omega; \mathbb{R}^n)$ par exemple (la formulation qui va suivre sera ainsi simple à résoudre avec FreeFem++). Dans ce cas, la direction de descente est l'unique élément $d \in H^1(\Omega; \mathbb{R}^n)$ tel que pour tout $\theta \in H^1(\Omega; \mathbb{R}^n)$,

$$\int_{\Omega} (\nabla d \cdot \nabla \theta + d \cdot \theta) dx = \langle J'(\Omega), \theta \rangle \quad (4.63)$$

En prenant $\theta = -hd$ on obtient dans le développement limité de J :

$$\begin{aligned} J(\Omega) &= J(\Omega_0) - h \langle J'(\Omega), d \rangle + O(h^2) \\ &= J(\Omega_0) - h \|d\|_{H^1(\Omega)^2}^2 + O(h^2) \end{aligned} \quad (4.64)$$

assurant la diminution de la fonction cout pour h assez petit.

4.4.2 Gestion de la contrainte

La contrainte peut être de différente nature : elle peut porter sur le volume ou sur le périmètre par exemple. Nous allons proposer une methode pour gérer une contrainte sur le volume. On introduit alors un multiplicateur de Lagrange l dans la fonction coût J associée à la contrainte de volume $V(\Omega) = V_0$ volume initial ou bien volume cible que l'on se fixe.

On dispose donc maintenant du Lagrangien suivant :

$$\mathcal{L}_c(\Omega) = \mathcal{L}(\Omega) + l(V(\Omega) - V_0) \quad (4.65)$$

ou $\mathcal{L}(\Omega)$ est le lagrangien définit pour le problème de base, sans contrainte. $\mathcal{L}_c(\Omega)$ désigne donc le lagrangien prenant en compte la contrainte de volume.

Remarquons que la condition d'optimalité a changé en présence d'une contrainte. Cette condition est maintenant donnée par :

$$J'(\Omega) + lV'(\Omega) = 0$$

La valeur du lagrangien l_n sera recalculée à chaque itération pour que la forme satisfasse (au mieux) la contrainte volumique de l'étape précédente. On va augmenter le lagrangien l_n à chaque étape si le volume est plus grand que le volume cible et le diminuer dans le cas contraire. Pour éviter les oscillations du volume, on relaxe l_n avec la valeur l satisfaisant la conditions d'optimalité à chaque itération et donnée en moyennant sur la frontière par :

$$l = - \frac{\int_{\partial\Omega} v ds}{\int_{\partial\Omega} ds}$$

La formule pour le calcul de l_n s'écrit donc :

$$l_{n+1} = \frac{l_n + l}{2} + \alpha_l \frac{V - V_0}{V_0} \quad (4.66)$$

avec α_l choisit convenablement en fonction du but recherché (α_l est représenté par la donnée `lagrangestep` dans le code). S'il est petit, la forme aura des libertés de mouvements, par contre plus il est grand, plus on augmente la précision sur le volume ciblé (voir les cas test partie 4.5). Le code pour cette contrainte est le suivant :

```
// Calcul du multiplicateur de lagrange de depart
lagrange=-int1d(Sh,2)(integrande)/perimeter;

:

// Mise a jour du multiplicateur de Lagrange associe au volume
lagrange=0.5*lagrange+0.5*int1d(Sh,2)(integrande)/perimeter
+lagrangestep*(volume-volume0)/volume0;
```

4.4.3 Critère d'arrêt

Le problème au niveau du critère d'arrêt est qu'on aimerait bien pouvoir arrêter l'algorithme lorsque l'on est proche du minimum local ou global. Une façon de faire serait de regarder la variation de la fonction coût au fur et à mesure des itérations et de décider selon un certain critère d'arrêter l'algorithme lorsque la fonction coût ne varie plus assez ou bien que la dérivée de forme est assez petite dans une certaine norme.

Mais ce raisonnement peut très bien empêcher l'apparition d'un autre minimum local meilleur que le précédent ou bien même empêcher d'atteindre le minimum global (si l'on autorise la fonction coût à effectuer des augmentations légères).

Ce qui est donc usuellement fait est l'arrêt de l'algorithme au bout d'un certain nombre d'itérations, avec la possibilité de reprendre l'algorithme plus tard à partir du dernier maillage si l'on en considère l'utilité.

4.4.4 Algorithme utilisé

Algorithme d'optimisation géométrique
<p>Mise en place des données initiales : déclarations $N_{max}, f, g, V_{cible}, \alpha_l, h, \dots$</p> <p>Construction forme/maillage initial</p> <p>Initialisation Calcul de la solution initiale sur le maillage Calcul du volume initial Initialisation du lagrangien</p> <p>Boucle Tant que $n < N_{max}$ Calcul de la solution Calcul de la solution adjointe Calcul du lagrangien à l'étape n : l_n Calcul de la direction de descente Optimisations (optionnel, voir section 4.7) - Optimisation du pas - Retournement des triangles - Contrôle du pas - Oscillations Déformation du maillage</p> <p>Fin Tant que</p> <p>Tracé des courbes Courbes de fonction coût, volume et condition d'optimalité Courbe de la solution sur le maillage final</p>

4.5 Cas tests effectués

Voici une liste descriptive non exhaustive des cas tests utilisés pour valider les algorithmes ainsi que les résultats numériques obtenus, les figures et commentaires. L'accent sera porté sur la compréhension du phénomène physique, l'interprétation des courbes obtenues et l'évolution du raisonnement sur le sujet.

4.5.1 Conduction et forces volumiques

Plaçons nous dans le cas suivant de la conduction de la figure 4.6.

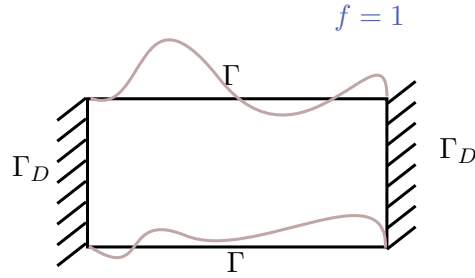


FIG. 4.6 – Schéma de la conduction avec force volumique seulement

Avec la Compliance

Fichier utilisé : CasTestCarreComplianceFVol.edp

Rappelons la compliance :

$$J(\Omega) = \int_{\Omega} f u dx = \int_{\Omega} u dx$$

On veut donc minimiser l'intégrale de u sur le domaine complet. La forme initiale étant un rectangle comme on peut le voir sur la figure 4.6, la forme aura donc intérêt à se "creuser" là où u est grand et à s'élargir là où u est petit. L'optimisation se fera aussi en respectant une contrainte volumique. Regardons sur la figure 4.7 la solution u du problème de conduction sur un rectangle pour situer les endroits où u est grand et petit.

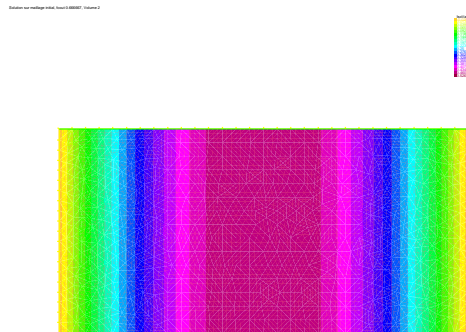


FIG. 4.7 – Solution initiale sur maillage initial, Volume=2

La figure 4.7 montre que u est grand au centre et petit vers les bords dirichlet, nul sur les bords. Ainsi la forme aura intérêt à se creuser au centre et à s'élargir vers les bords, tout en essayant de tendre vers le volume cible.

Nous remarquerons que le paramètre *lagrangestep* de l'algorithme (qui correspond au α_l de la section 4.4.2, Gestion de la contrainte) contrôle la précision sur le volume. Plus il est grand, plus la volume de la forme sera proche du volume cible (ici le volume cible est fixé au volume initial).

Regardons sur la figure 4.8 ce que donne l'optimisation avec un pas de descente fixé à 0.05, un nombre d'itération maximum de 50 et un volume cible de 2.

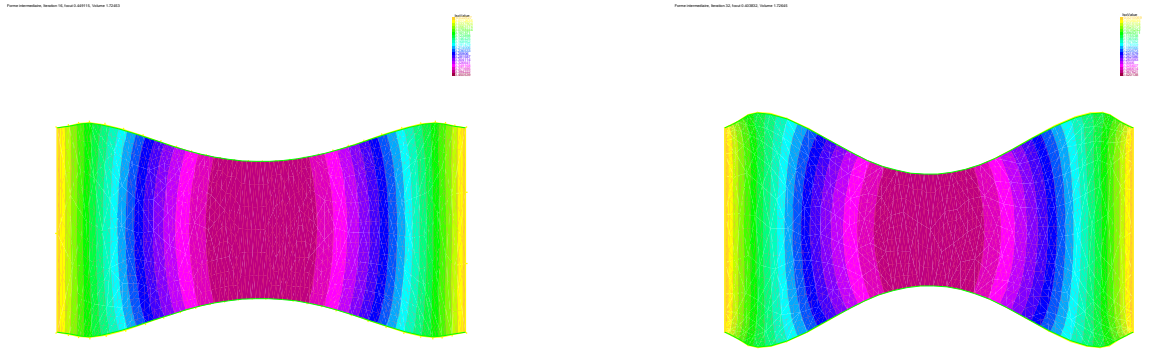


FIG. 4.8 – Solutions et formes à l'itération 16 (gauche) et 32 (droite)

Les volumes sont respectivement aux itérations 16 et 32, 1,724 et 1,726 et les valeurs des fonctions coûts sont respectivement 0,449 et 0,403. Regardons plus précisément sur la figure 4.9 les variations sur toutes les itérations du volume et de la fonction coût :

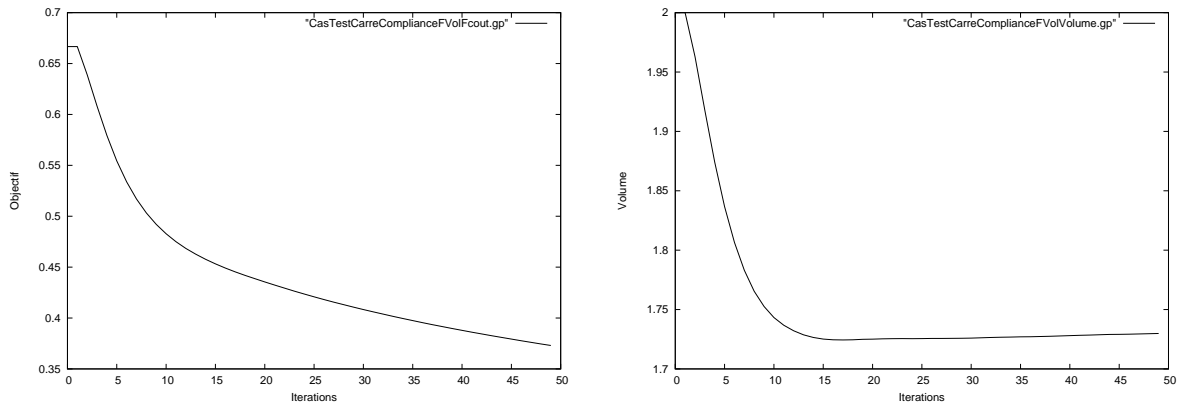


FIG. 4.9 – Valeur de la fonction coût (gauche) et volume (droite) sur 50 itérations

La figure 4.9 montre bien une diminution de la fonction coût et la prise de forme prévue par l'argument précédent. Remarquons que la pente de la fonction coût aux dernières itérations n'est pas encore proche de 0, l'algorithme peut donc encore améliorer la forme. Le volume diminue jusqu'aux itérations 15-20 à la valeur d'environ 1,72, assez éloigné du volume cible de 2 et remonte ensuite pour les itérations précédentes mais faiblement. Regardons sur la figure 4.10 l'algorithme sur 300 itérations pour mieux se rendre compte de l'évolution asymptotique :

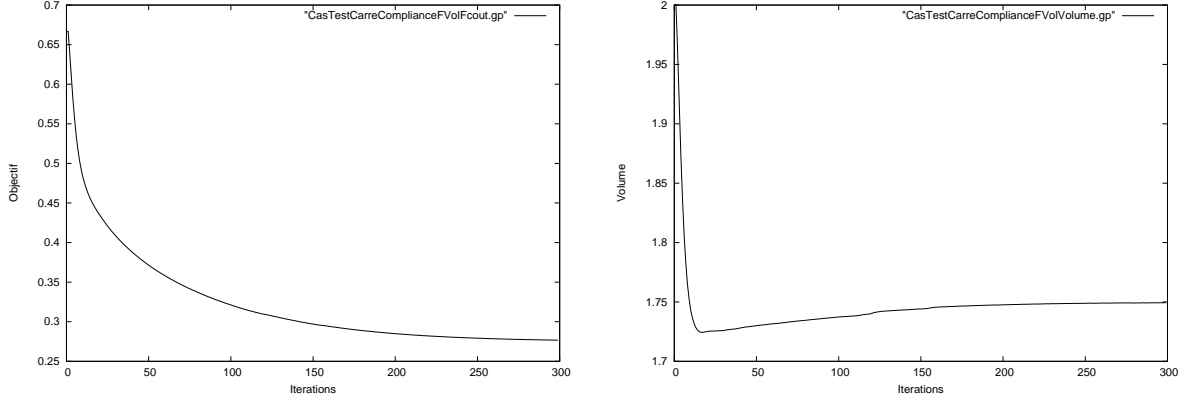


FIG. 4.10 – Valeur de la fonction coût (gauche) et volume (droite) sur 300 itérations

On voit sur la figure 4.10 que la pente de la fonction coût est presque nulle et le volume n'augmente plus trop et commence à stagner au bout des 300 itérations. On a donc bien optimisé la forme sous contrainte volumique mais le volume n'est pas proche du volume cible. On a laissé une marge sur la précision du volume en prenant un petit *lagrangestep* de 3.

Prenons un *lagrangestep* plus élevé de 50 sur 50 itérations. Cela nous donne la figure 4.11.

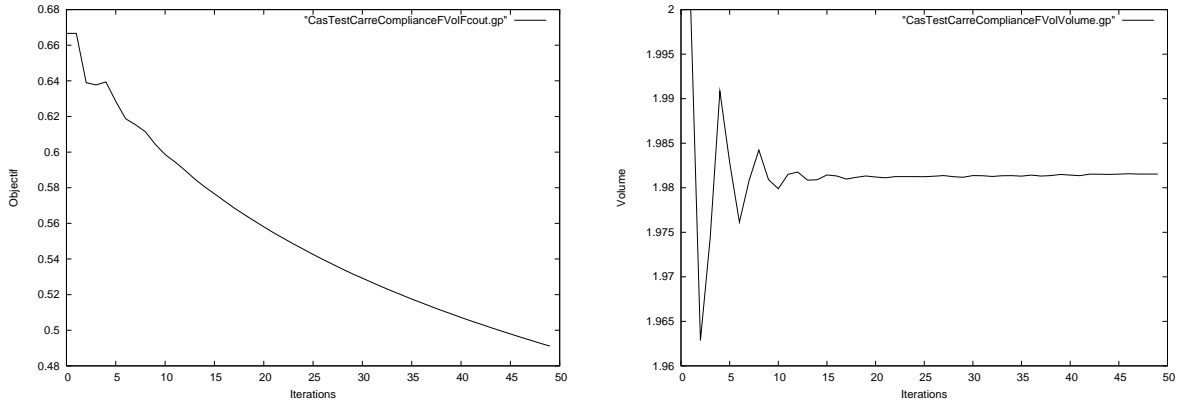


FIG. 4.11 – Valeur de la fonction coût (gauche) et volume (droite) sur 50 itérations

La figure 4.11 montre que la première "grosse" variation de volume de 2% sur les deux premières itérations est nécessaire pour modifier la forme assez pour avoir une diminution de la fonction coût conséquente. Le volume augmente ensuite à partir de la 2^e itération, faisant stagner la fonction coût. En effet, plus le volume de la forme diminue, plus la fonction coût diminue car étant une intégrale de $u > 0$ sur Ω . Les variations suivantes du volume sont fait d'oscillations autour de ce qui apparaît comme une valeur limite proche de 2. Il y a donc compétition entre la diminution du volume pour faire diminuer J et la contrainte volumique.

En effet, si l'on met α_l à 0 on obtient la solution et forme finale (au bout de 100 itérations) de la figure 4.12.

La diminution de la forme est plus élevée au centre du rectangle que sur les bords. Le volume semble diminuer sans restriction mais plus rapidement au centre. L'optimisation n'est évidemment pas terminée dans ce cas et semble qu'elle ne se terminera que lorsque le volume sera nul, que la fonction coût sera 0, et que le mince filament joignant les parties gauches et droites disparaisse : autant dire que numériquement l'algorithme ne s'arrêtera pas avant un retournement de triangle.

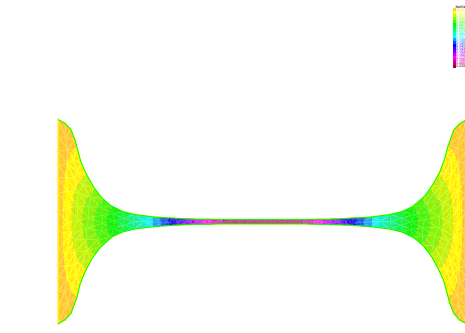


FIG. 4.12 – Solution finale, 100 itérations, Volume=0.35

Voici sur la figure 4.13 les courbes de la fonction coût et du volume correspondantes :

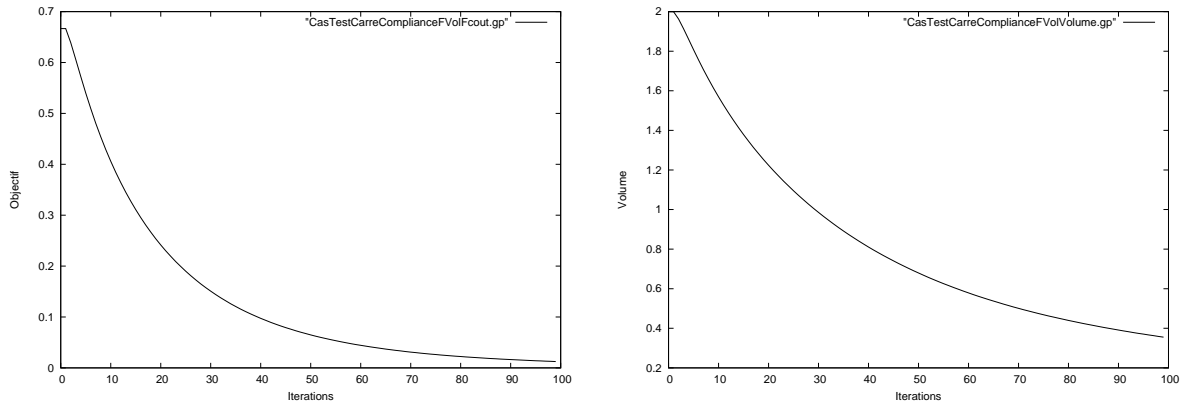


FIG. 4.13 – Valeur de la fonction coût (gauche) et volume (droite) sur 100 itérations

Les deux courbes montrent une diminution forte mais de moins en moins rapide au fil des itérations. La fonction coût et le volume semblent tendre vers 0 : résultat conforme à la théorie. u étant toujours positif par principe du maximum sur l'équation d'état et sans contrainte de volume avec une diminution de la fonction coût tend que le volume diminue, on a légitimement une convergence vers 0 de J et du volume.

Plaçons nous dans le cas de figure où l'on va faire varier α_l pendant l'optimisation et comparons le résultat avec l'optimisation sans cette variation de α_l . Regardons d'abord le cas de la figure 4.14 sur 200 itérations avec un α_l augmentant de 1 unité à chaque itération et commençant à $\alpha_l^0 = 3$.

On constate qu'à partir de la 90ième itération, le volume oscille autour du volume cible et la fonction coût aussi oscille aussi tout en diminuant en moyenne à vue d'oeil très faiblement. Sur cet exemple, à partir d'un α_l trop élevé (ici une limite de $\alpha_l^{lim} = 90$ a été mise en évidence sur cet exemple spécifiquement), il y a oscillation du volume et de J . Ses oscillations sont en fait dues au fait que l'on s'approche du volume cible, ainsi, la quantité de variation relative du volume dans le calcul du multiplicateur de lagrange l_n change de signe dès que $V - V_0$ change de signe : négatif au départ car ici l'optimisation a tendance à diminuer le volume ($V < V_0$), positif lorsque l'on dépasse le volume cible. On voit aussi une remontée de la fonction coût avant la 10ème itération du fait que le volume doit augmenter pour satisfaire la contrainte volumique : la diminution du volume jusqu'alors pour faire diminuer la fonction coût, n'est plus assez forte pour compenser la contrainte volumique.

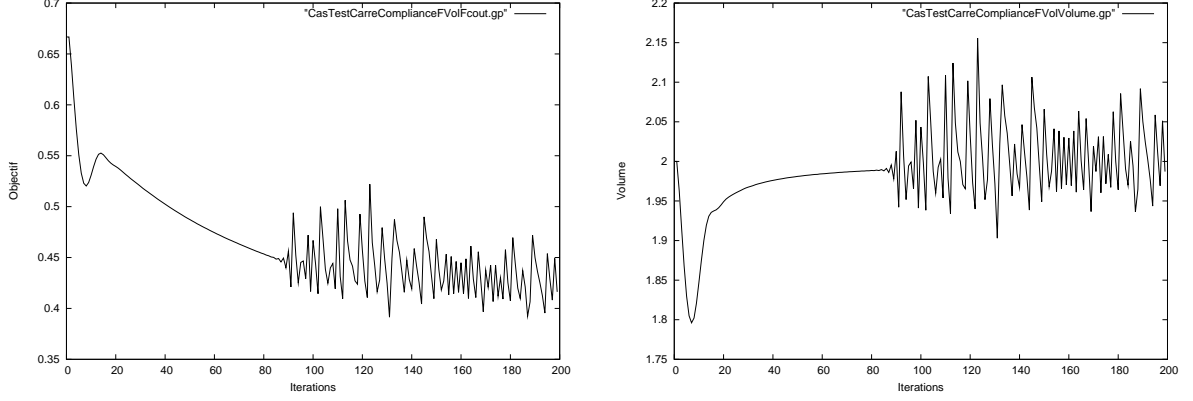


FIG. 4.14 – Valeur de la fonction coût (gauche) et volume (droite) sur 200 itérations, $\alpha_l^{n+1} = \alpha_l^n + 1$

On sait que l'on a au moins une forme optimale qui réalise le minimum de J pour chaque volume Ω . Ainsi la recherche d'une forme optimale sous une contrainte volumique passe par un paramétrage sur la précision que l'on souhaite posée sur le volume cible donc un paramétrage sur α_l et/ou un paramétrage sur le volume cible. Un α_l élevé va avoir tendance à ne pas permettre de grosse variation de volume lorsqu'on sera proche du volume cible, et imposera de grosse variation lorsqu'on en sera loin. De plus, être proche du volume cible s'accompagnera d'oscillation empêchant la bonne continuation de l'optimisation.

Regardons ce qui se passe au niveau de la condition d'optimalité. D'après (4.27), écrivons l'entité qui nous permettra de vérifier si elle est vérifiée :

$$C_{opt} = \frac{\int_{\Gamma} |2fu - |\nabla u|^2| ds}{\int_{\Gamma} ds}$$

ce qui peut se voir comme la valeur absolue moyenne de $2fu - |\nabla u|^2$ sur Γ , traduisant un max pour la condition d'optimalité. On veut que C_{opt} converge vers 0 lorsque la forme converge vers la solution optimale. Etant une condition nécessaire, il faut la vérifier si l'on suppose avoir obtenue une forme optimale satisfaisante. Cependant numériquement ce n'est pas la bonne condition d'optimalité à vérifier : en effet, dans le code, on résout le problème de l'extension (recherche de la direction de descente) en tenant compte de la contrainte volumique en résolvant :

$$\int_{\Omega} (\nabla d \nabla \theta + d \cdot \theta) dx = \langle J'(\Omega) + l, \theta \rangle$$

où l est le multiplicateur de lagrange de la contrainte volumique. Ainsi la condition d'optimalité s'écrit pour correspondre à ce que l'on résout :

$$C_{opt} = \frac{\int_{\Gamma} |2fu - |\nabla u|^2 + l| ds}{\int_{\Gamma} ds}$$

Ainsi, le tracé de C_{opt} sur un grand nombre d'itérations (200 itérations avec un pas de 0.1), en se limitant à la partie libre sans les extrémités (car on a un problème à la jonction Γ et Γ_D de définition et régularité de l'intégrande intervenant dans C_{opt} : voir la partie 4.7.4), nous donne la figure 4.15.

On remarque que C_{opt} diminue et fini par converger vers 0, i.e. lorsque la fonction coût ne varie quasiment plus et la forme optimale non plus. On est aux alentours de 0.016 à la dernière itération pour la valeur moyenne en valeur absolue : on a bien convergence vers 0 (La courbe en prenant en compte toute la partie libre, sans rien exclure, est de même type mais la valeur finale est plus élevée).

On a donc mis ici en évidence sur un exemple simple l'importance du volume en compétition avec la force de variation de la forme et l'importance de la considération du nombre d'itérations dans les résultats obtenus : on peut ne pas avoir obtenu une forme optimale satisfaisante à la fin des itérations

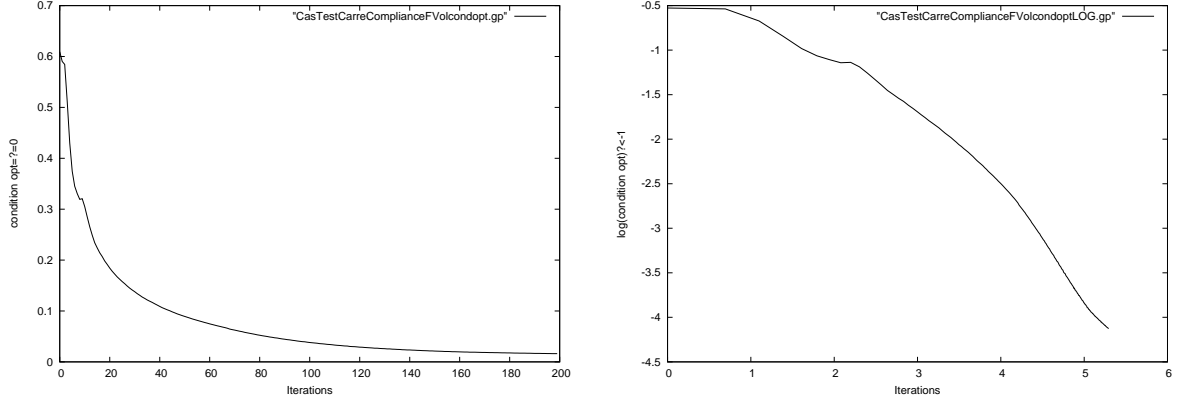


FIG. 4.15 – C_{opt} , 200 itérations

données. La condition d'optimalité est assez vérifiée à en juger par le calcul précédent. Notons que le pas n'est pas ici optimisé : voir 4.7 pour l'optimisation du code.

Avec l'énergie de déformation

Fichier utilisé : CasTestCarreGradientFVol.edp

En prenant la fonction coût suivante

$$J(\Omega) = \int_{\Omega} k(x) |\nabla u|^2 dx$$

on obtient avec $\alpha_l = 3$, $k = 1$ sur Ω , pas de 0.05 et 200 itérations les deux courbes suivantes de la figure 4.16.

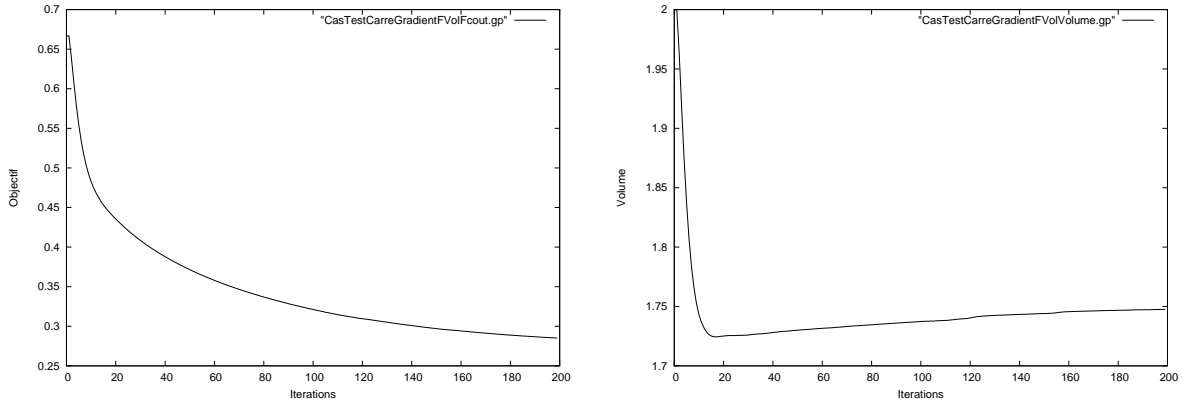


FIG. 4.16 – Valeur de la fonction coût (gauche) et volume (droite) sur 200 itérations, α_l petit

La forme à la dernière itération est la même que pour la compliance. En comparant les résultats trouvés sur la compliance (figure 4.10) avec ceux de l'énergie de déformation avec $k = 1$ (figure 4.16), on remarque une même convergence vers une même forme : les courbes des fonctions coût et volume se ressemblent beaucoup. Ce résultat est logique au vu du fait que la compliance peut être vu comme l'intégrale du carré du gradient sur Ω (voir la section (4.2)) : le problème est finalement le même, donc les résultats numériques doivent être proches, ce qui est le cas.

Regardons toujours avec $k = 1$, l'évolution de la condition d'optimalité, en prenant un pas deux fois plus grand de 0.1, sur 200 itérations. On obtient la figure 4.17. La condition d'optimalité converge bien vers 0 et son logarithme nous montre sur les dernières itérations son ordre de convergence en log-log.

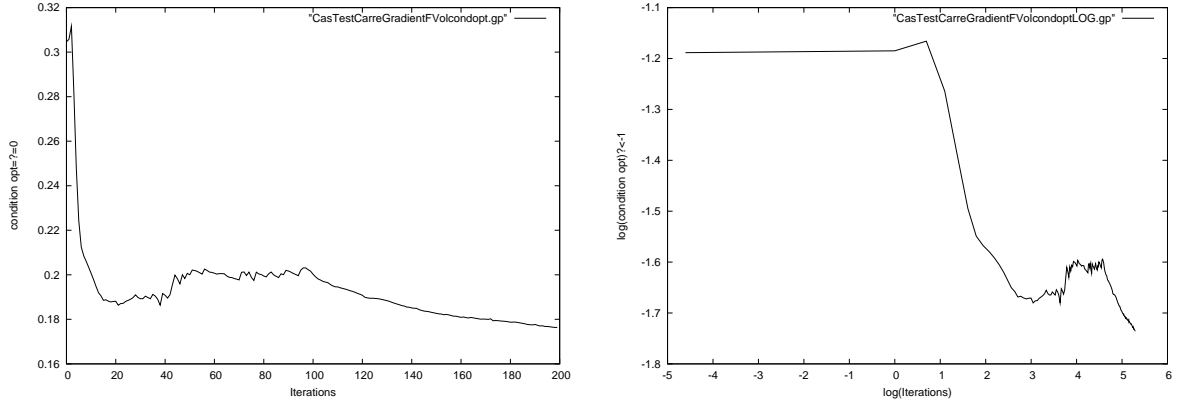


FIG. 4.17 – C_{opt} (gauche) et son logarithme (droite) sur 200 itérations, α_l petit, pas=0.1

Voyons ce qui se passe si l'on change la fonction k pour qu'elle vaille 1 à gauche du domaine (si $0 < x < 1$) et 0.2 à droite du domaine (si $1 < x < 2$). Cela s'interprète comme vouloir minimiser J qui est localisé sur la partie gauche du domaine. Cela revient donc à modifier plus fortement la forme à gauche du domaine au vue du critère de l'énergie de déformation (qui peut être vu comme la compliance si $k = 1$ sur tout le domaine). On obtient alors, toujours avec les même conditions, les figures 4.18 et 4.19.

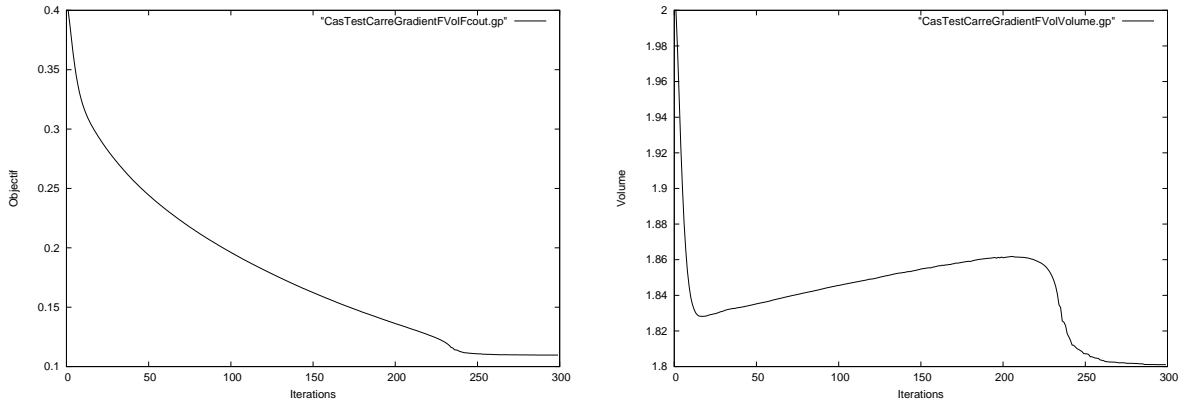


FIG. 4.18 – Valeur de la fonction coût (gauche) et volume (droite) sur 300 itérations, α_l petit

La fonction coût diminue bien. La forme obtenue est en accord avec le raisonnement disant que l'optimisation se ferait plus particulièrement sur le côté gauche du domaine. On peut remarquer aux environs de l'itération 230, un passage à un palier où la fonction coût ne variera plus beaucoup : ce palier correspond au moment où l'optimisation a créé un filament fin sur la partie gauche du domaine. La forme ne peut plus amincir ce filament et l'optimisation se stabilise. On dirait que la forme tend à vouloir créer un trou au niveau du filament. Le volume diminue puis remonte jusqu'à environ la 230ième itération pour redescendre et se stabiliser. Normalement le volume tend à remonter une première fois pour aller se stabiliser sans jamais redescendre. Mais là, à cause de la création du filament, l'optimisation n'a d'autre choix que de faire diminuer le volume pour continuer l'optimisation.

Regardons l'effet de la création d'un trou au départ en prenant $k = 1$: on change donc de topologie et l'ensemble des formes admissibles change complètement. La figure 4.20 montre la fonction coût et le volume à partir d'une forme initiale avec un trou de 0.2 de rayon centré dans le rectangle.

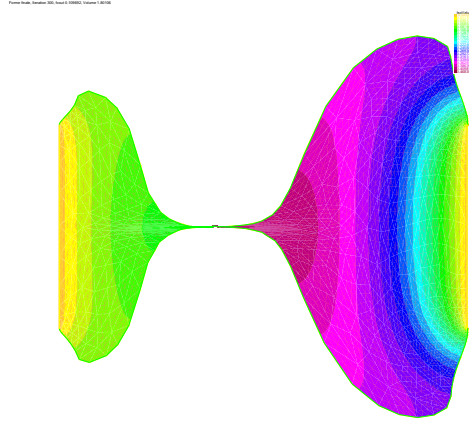


FIG. 4.19 – Solution finale $k = \mathbf{1}_{0 < x < 1} + 0.2 * \mathbf{1}_{1 < x < 2}$, 300 itérations, Volume=1.8010

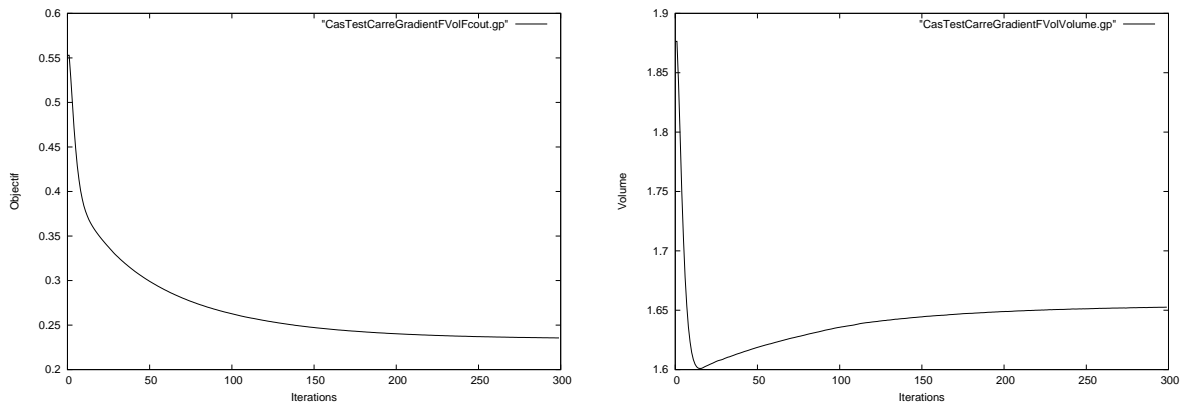


FIG. 4.20 – Valeur de la fonction coût (gauche) et volume (droite) sur 300 itérations, α_l petit, 1 trou

La fonction coût diminue sans jamais remonter et finit par stagner. Le volume diminue puis remonte pour stagner : il y a eu création de deux filaments (tendance à créer deux trous) comme le montre la figure 4.21. La valeur de la fonction coût à la dernière itération est de 0.2355 (pour un volume de 1.6525) plus petit que le 0.2767 de l'optimisation sans trou (pour un volume de 1.7493). La création d'un trou a donc fait diminuer la valeur de la fonction coût (mais à volume non égal donc la comparaison est biaisée mais vaut tout de même la peine que l'on en parle).

Avec deux trous on obtient une valeur pour la fonction coût de 0.2237 (pour un volume de 1.6238) meilleur que la figure à 1 trou mais avec un volume plus faible et plus proche de la figure sans trou : meilleur comparaison avec la figure sans trou. La figure 4.22 montre les variations de la fonction coût et du volume et la figure 4.23 montre la forme finale. Par création de trous successifs on améliore la forme optimale que l'on obtient par notre algorithme.

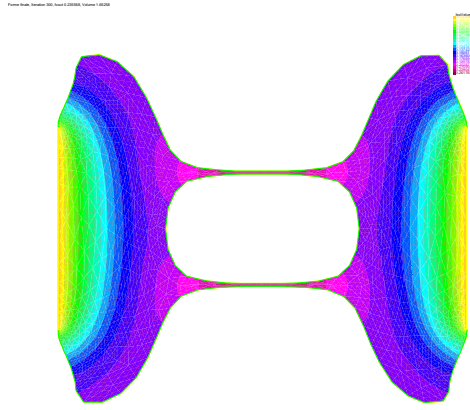


FIG. 4.21 – Solution finale 1 trou, 300 itérations, Volume=1.6525

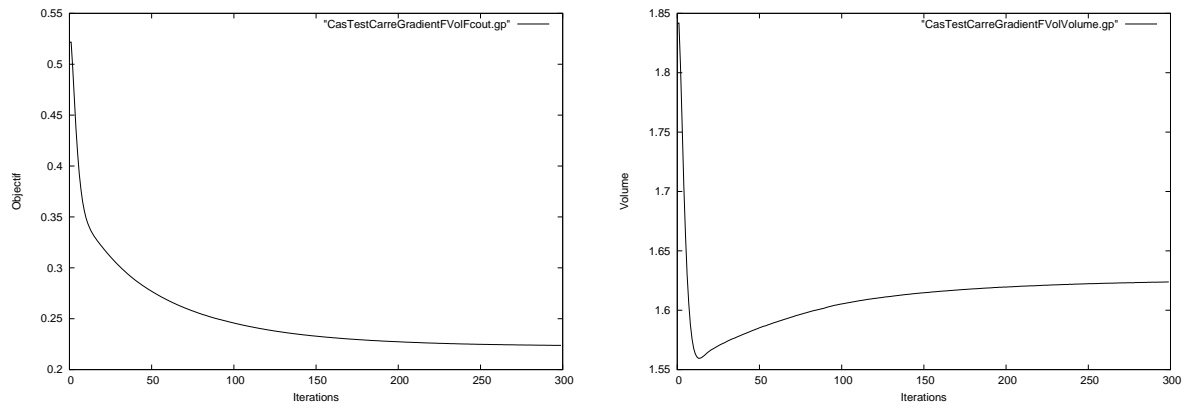


FIG. 4.22 – Fonction coût (gauche) et volume (droite) sur 300 itérations, α_l petit, 2 trous

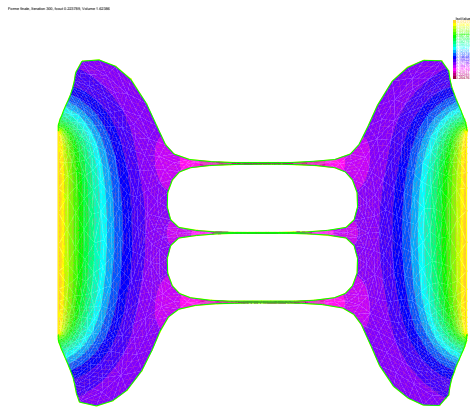


FIG. 4.23 – Solution finale 2 trous, 300 itérations, Volume=1.6238

4.5.2 Conduction et variation du Neumann : rectangle

Avec la compliance

Fichier utilisé : CasTestCarreCompliance.edp

Plaçons nous dans le cas de la conduction et supposons que le bord Γ_N peut varier tout comme Γ : voir figure 4.24.

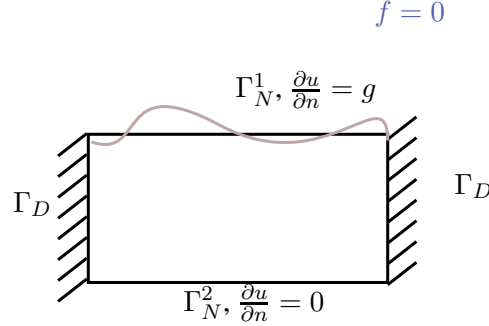


FIG. 4.24 – Schéma de la conduction avec variation du bord Neumann

Supposons que le maillage soit en "forme de maison" : un rectangle et un "toit". La figure 4.25 montre le maillage et la solution initiale u sur ce maillage :

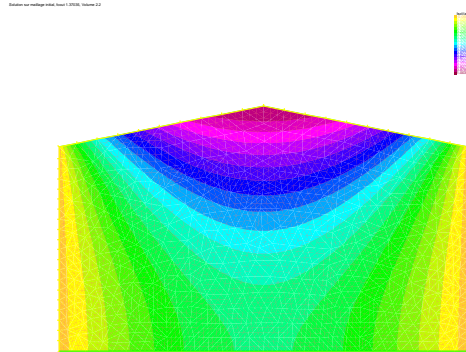


FIG. 4.25 – Solution initiale sur maillage initial, Volume=2,2

On essaye ici de minimiser la compliance s'écrivant

$$J(\Omega) = \int_{\Gamma_N} g u ds = \int_{\Gamma_N} u ds$$

avec $g = 1$ et $f = 0$ (pas de force volumique), avec la particularité que seul le bord Γ_N peut varier durant l'optimisation. Ainsi si l'on souhaite une forme optimale ayant un volume de 2 par exemple, au vu de la solution sur le maillage "maison" (u plus élevé au centre du toit) l'optimisation aura tendance à creuser au milieu de Γ_N et à se remplir vers les bords de Γ_N .

En paramétrant l'algorithme avec 150 itérations, $\alpha_l = 2$ et un volume cible valant le volume initial $V_0 = 2.2$ et le pas de descente valant 0.005, la figure 4.26 montre ce que l'optimisation donne :

On arrive donc bien à diminuer la fonction coût, le volume diminuant aussi. La fonction coût commence à stagner vers la fin des 150 itérations, le volume étant un peu supérieur à 1,95. On a donc obtenu

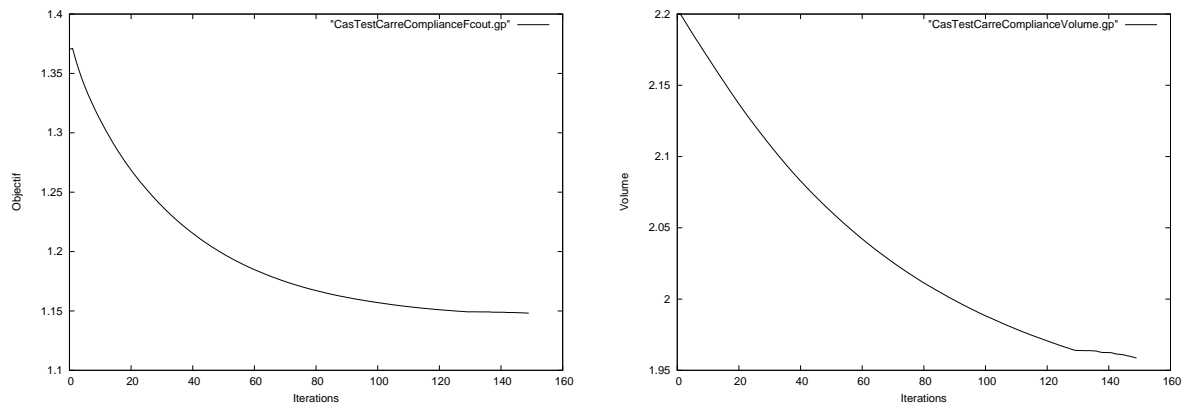


FIG. 4.26 – Valeur de la fonction coût (gauche) et volume (droite) sur 150 itérations, α_l petit
une meilleur forme que le toit dont le volume est proche de 2 et ressemblant à la figure 4.27.

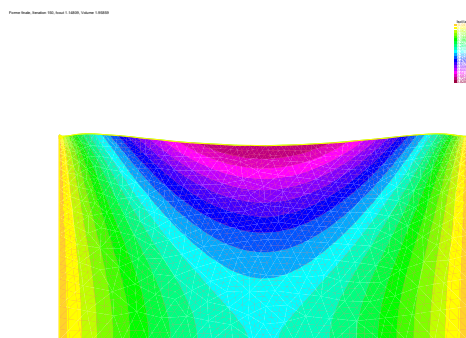


FIG. 4.27 – Solution finale, Volume=1,9585

La forme obtenue est en accord avec les prévisions intuitives précédentes. Cependant, les "creux" et "bosses" observées sont moins impressionnantes que dans l'exemple avec la force volumique seule de la section 4.5.1.

Remarquons quelque chose d'intéressant : si l'on pose le volume cible à 2 (et non plus 2,2) et que l'on prend $\alpha_l = 30$ (élevé pour converger vers ce volume), la figure 4.28 montre les courbes que l'on obtient. La forme finale est sur la figure 4.29.

On constate une plus rapide diminution de la fonction coût dans les premières itérations. Mais le rythme se compense ensuite pour stagner à des valeurs sensiblement différentes avec un volume différent.

Si l'on augmente le pas (prenons un pas de 0.05 : 10 fois plus important que le précédent), pour converger plus vite vers la forme optimale, la figure 4.30 montre ce qui se passe sur 120 itérations, volume cible de 2 et $\alpha_l = 2$.

La fonction coût diminue bien plus rapidement en dessous du seuil 1.15 mais fini par remonter à la 20^e itération environ et finira par stagner. Le volume diminue et commencera à stagner sur la fin. Cette remontée de la fonction coût est due à un pas de descente trop important à partir d'un certain moment. On doit donc contrôler le pas d'une manière ou d'une autre : voir la partie 4.7 traitant de l'optimisation du code.

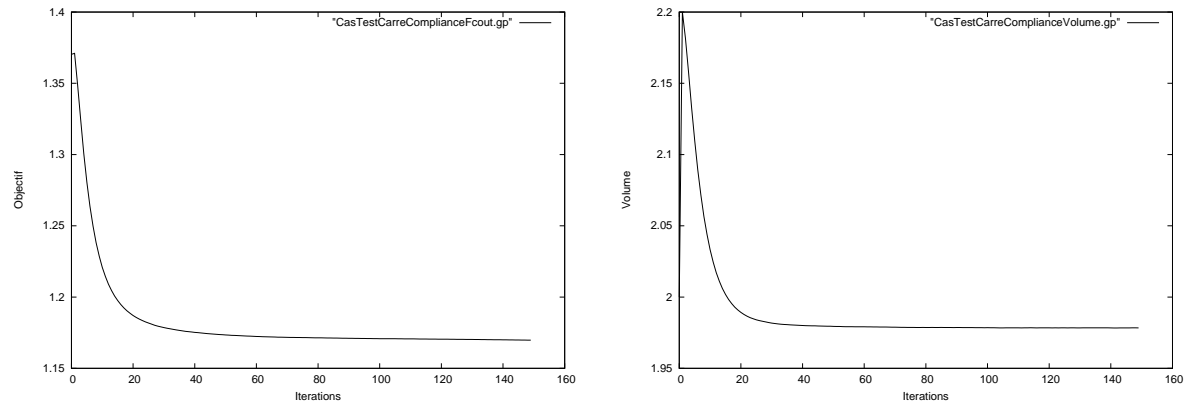


FIG. 4.28 – Valeur de la fonction coût (gauche) et volume (droite) sur 150 itérations, α_l grand

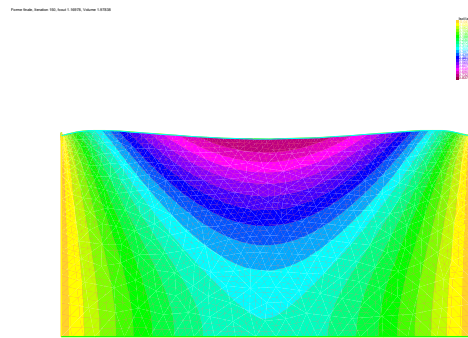


FIG. 4.29 – Solution finale, Volume=1,9783

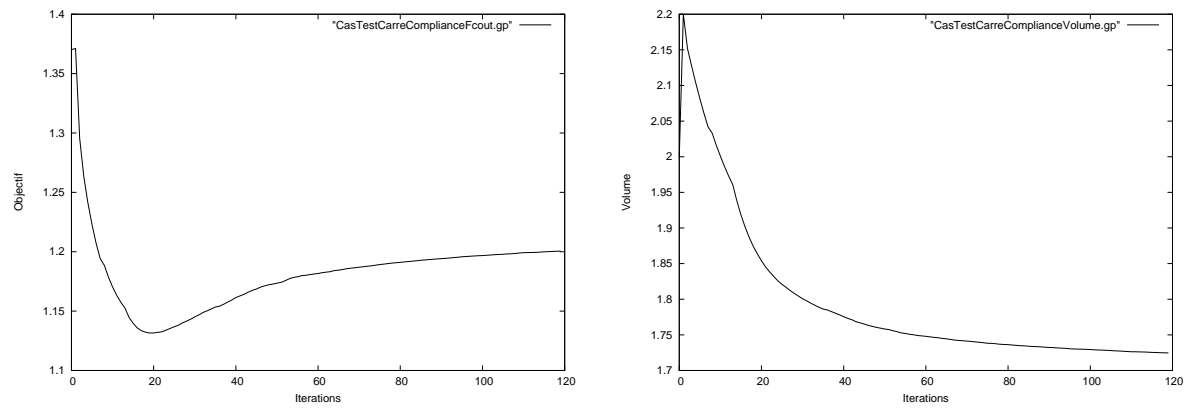


FIG. 4.30 – Valeur de la fonction coût (gauche) et volume (droite) sur 120 itérations, α_l petit

On a réussi ici à optimiser une forme en faisant varier la partie qui reçoit des forces surfaciques dites suiveuses (en utilisant la formule (4.25)) et à mettre en évidence la nécessité du contrôle du pas.

Avec l'énergie de déformation

Fichier utilisé : CasTestCarreGradient.edp

Regardons le même problème en utilisant la fonction coût d'énergie de déformation :

$$J(\Omega) = \int_{\Omega} k(x) |\nabla u|^2 dx$$

Avec les mêmes données : 0.005 pour le pas, $f = 0$, $g = 1$, $\alpha_l = 2$, volume cible 2.2, on obtient les mêmes courbes et la même forme en prenant $k = 1$. Localisons maintenant la fonction coût et prenons $k = \mathbf{1}_{x < 1} + 0.2 * \mathbf{1}_{x > 1}$. La figure 4.31 montre la fonction coût et le volume sur les 150 itérations de l'optimisation.

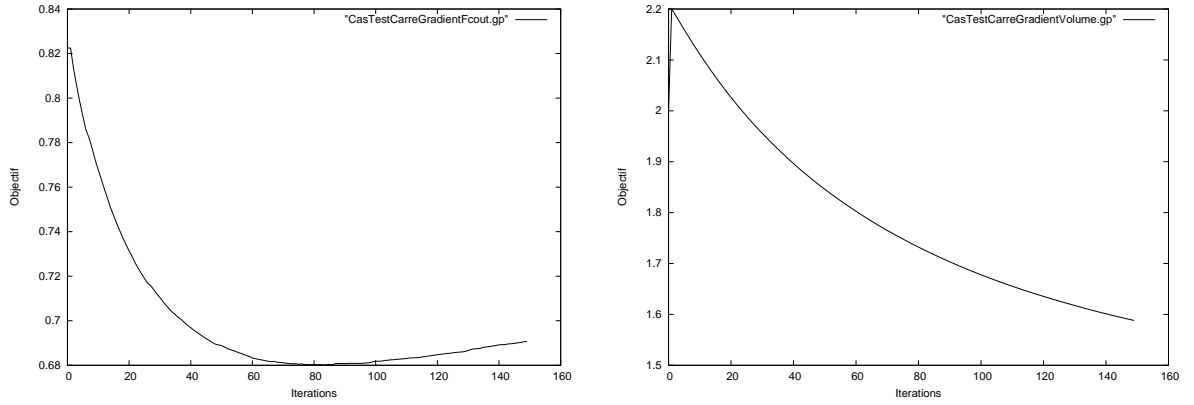


FIG. 4.31 – Valeur de la fonction coût (gauche) et volume (droite) sur 150 itérations, α_l petit

On constate qu'à partir d'une certaine itération, la fonction coût augmente, traduisant le fait que l'on ne contrôle pas le pas. Le volume quand à lui diminue tranquillement. La forme obtenue au bout des 150 itérations est celle de la figure 4.32. Au bout de 80 itérations (quand la fonction coût est au plus bas) on obtient une forme que l'on peut voir sur la même figure 4.32.

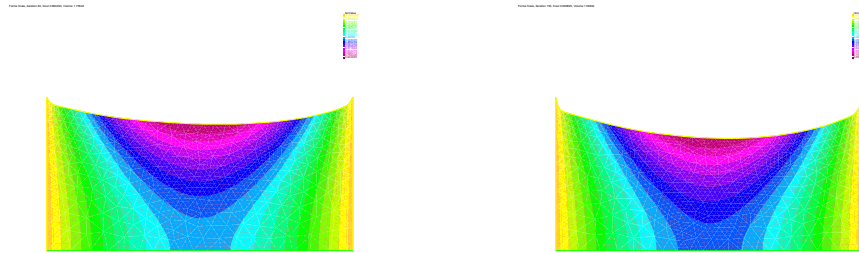


FIG. 4.32 – Solutions à l'itération 80 (gauche) et 150 (droite)

On observe un creux un peu plus important sur la partie droite de la figure que sur la partie gauche. Mais au final, il est difficile de commenter le résultat d'une telle optimisation, on peut seulement dire qu'au vu des courbes de la fonction coût et du volume, on a évolué vers une meilleure forme que la forme initiale et que passé l'itération 80 on doit maîtriser le pas. Voir optimisation du code à la partie 4.7.

4.5.3 Conduction et variation du Neumann : carré troué

Avec la compliance

Fichier utilisé : CasTestCarreTroueCompliance.edp

Regardons ce qu'il en est numériquement avec $\alpha_l = 4$, 50 itérations, un volume cible de volume initial et un pas de 0.02. La figure 4.35 montre la forme au bout des 50 itérations, la figure 4.33 montre les courbes de la fonction coût et du volume et la condition d'optimalité est visible sur la figure 4.34.

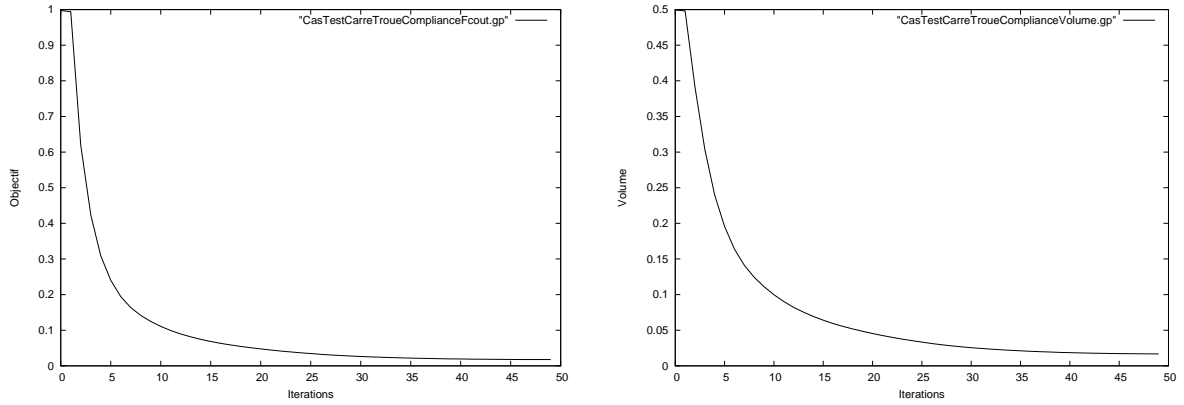


FIG. 4.33 – Fonction coût (gauche) et volume (droite) sur 50 itérations

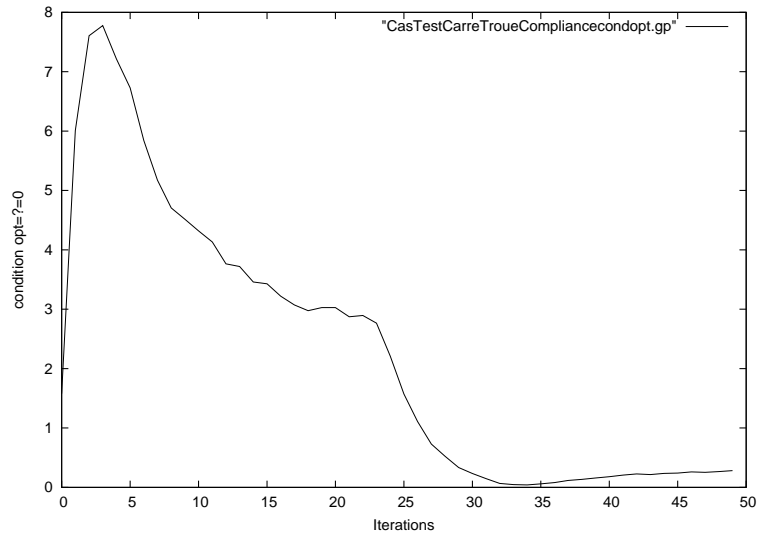


FIG. 4.34 – Tracé de la condition d'optimalité sur 50 itérations

On remarque que la fonction coût et le volume convergent vers une valeur petite. Le bord extérieur a pris la forme d'un cercle de rayon décroissant comme prévu. Les reliquats des coins ont disparus lors de l'optimisation avant l'itération 30, seul apparaît après le passage par l'itération 30, 4 reliquats des quatres coins : la forme continue d'évoluer mais la condition d'optimalité commence à augmenter. Les reliquats aux quatres coins permettent de conserver un certain volume pour la forme. La contrainte volumique portée par le lagrangien l_n à chaque itération a eu un petit effet sur l'irréversible diminution du volume bien que le terme $\alpha_l \frac{V - V_0}{V_0}$ devienne de plus en plus grand. On peut prendre comme forme "optimale" la forme aux alentours de l'itération 30, là où la condition d'optimalité est assez petite mais cette forme ne possèdera pas un volume proche de ce que l'on recherchait. La condition d'optimalité finit

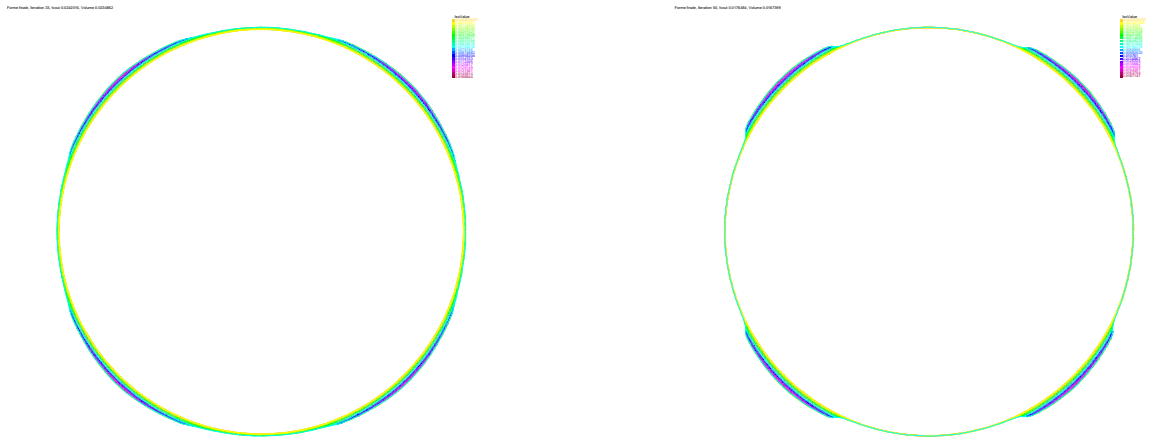


FIG. 4.35 – Solution au bout des 33 itérations et des 50 itérations

par réaugmenter probablement à cause que le volume devient trop bas ou bien du fait que l'on traite de forces suiveuses ce qui peut entraîner quelques problèmes du fait du terme supplémentaire induit dans le calcul de la dérivée de forme.

Si l'on commence avec un $\alpha_l = 20$ par exemple et un volume cible de volume initial, la forme obtenue sur 100 itérations est celle de la figure 4.38, les fonctions coûts et volume sont visibles sur 4.36 et la condition d'optimalité est visible sur 4.37.

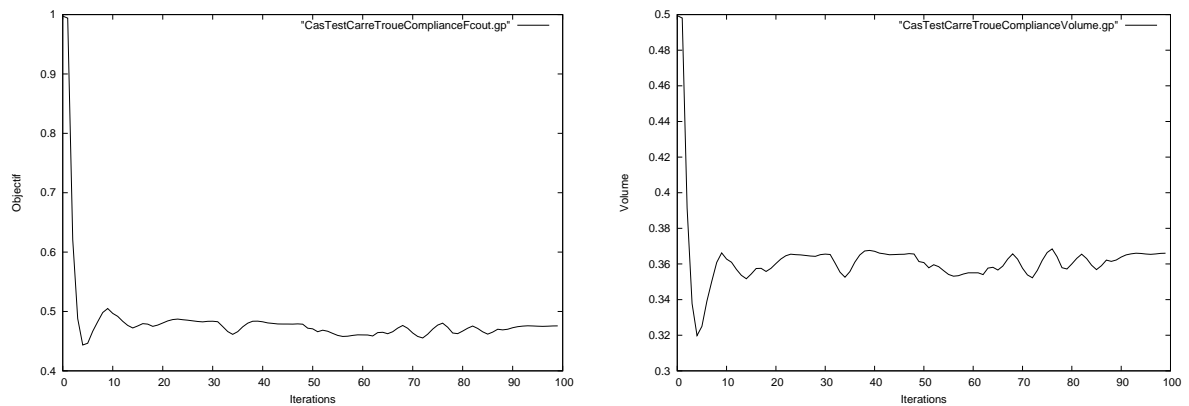


FIG. 4.36 – Fonction coût (gauche) et volume (droite) sur 100 itérations

La forme tend vers un anneau dont le volume tend vers une certaine valeur (lors de la première itération, le volume a chuté d'un coup). La fonction coût tend vers une certaine valeur aussi (avec oscillation autour de cette valeur dès les premières itérations). La contrainte volumique n'a pas été très bien prise en compte, le volume réel étant assez éloigné du volume cible. Mais on a évité que le volume tende vers un volume nul. On pourrait ainsi augmenter α_l pour s'assurer que la contrainte volumique soit satisfaite mais il faudrait se méfier alors de la mise en place possible d'oscillations (en effet, α_l étant élevé, la contrainte sur le volume est donc très marquée et la forme peut donc osciller entre des volumes variant autour du volume cible). La condition d'optimalité n'est pas vraiment satisfaisante : on peut toujours blâmer le fait que l'on traite de force suiveuse et/ou que la contrainte volumique n'est pas assez satisfaite. C'est surtout ce dernier point qui pose problème : si la forme doit et peut seulement par exemple diminuer de volume pour que la fonction coût diminue, alors il y a compétition entre la diminution du volume de la forme et la contrainte volumique.

Prenons un exemple où les deux bords sont des cercles : on commence avec un anneau. $r_1 = 2$ et

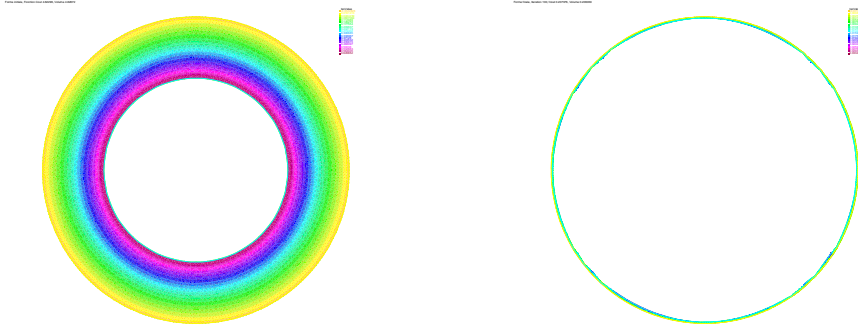


FIG. 4.40 – Forme initiale (gauche) et forme finale (droite) aux bout des 100 itérations

Les résultats numériques correspondent à la théorie (convergence vers un anneau ultra fin) mais ne sont pas satisfaisants : le problème réside dans l'application de la contrainte de volume, dont on constate dans cet exemple, que son implementation nécessite quelques ajouts ou changements. Par exemple on pourrait arrêter l'algorithme dès que la forme s'éloigne trop du volume cible une fois proche de celui ci, augmenter drastiquement α_l , ou bien utiliser à bonne escient la fonction *cut* de l'algorithme permettant de définir dans quelles zones on autorise la forme à varier ou encore augmenter α_l en conséquence. En tout cas, le code doit être amélioré.

4.5.4 Conduction et forces volumiques : carré troué

Avec la compliance

Fichier utilisé : CasTestCarreTroueComplianceFVol.edp

On suppose donc une force volumique $f = 1$. En résolvant le problème sur un carré troué, avec un pas de 0.25, un volume cible valant le volume initial et sur 100 itérations, on observe sur les figures 4.41 et 4.42 que la forme tend vers un anneau dont le volume diminue jusqu'à atteindre un certain volume.

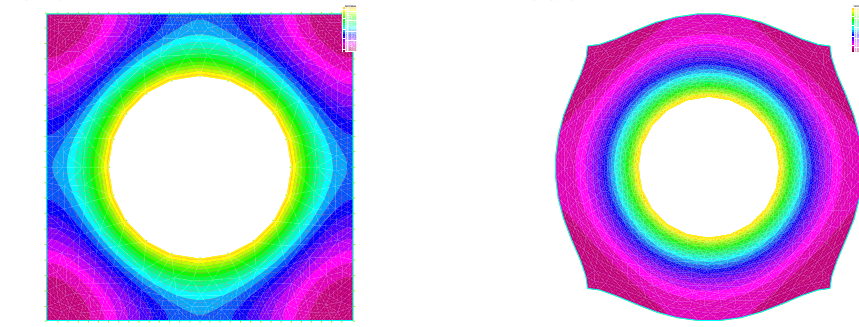


FIG. 4.41 – Forme initiale (gauche) et forme finale (droite) aux bout des 100 itérations

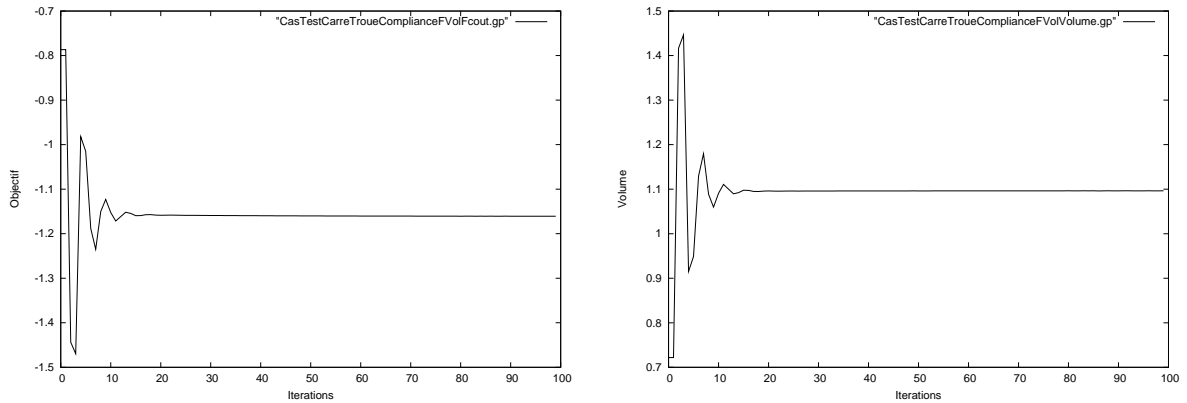


FIG. 4.42 – Fonction coût (gauche) et volume (droite) sur 100 itérations

La partie libre tend à s'arrondir (sauf les coins qui perdurent). Dans le cas des forces suiveuses de la partie 4.5.3, les coins avaient disparu par l'utilisation de l'extension de la normale (voir 4.7.6 et l'utilisation dans le code FreeFem++ du calcul de la divergence de cette normale). La forme tend donc bien vers un anneau. On initialisera les formes suivantes avec un anneau.

Regardons maintenant si α_l joue son rôle de paramètre de contrainte volumique en comparaison avec le cas de la partie 4.5.3.

Regardons d'abord le cas $r_2 < r_1$. En partant du même problème mais avec $r_1 = 1.5$ et $r_2 = 1$ (bord Dirichlet à l'extérieur) sur 40 itérations et $\alpha_l = 2$, on obtient les figures 4.43 et 4.44.

La forme converge bien vers celle prévue théoriquement, la fonction coût et le volume converge vers des valeurs non nulles. La dernière valeur de la condition d'optimalité de la figure 4.45 est 5×10^{-4} : convergence vers 0 de la condition d'optimalité. Regardons à présent le cas $r_2 = 2 > r_1 = 1.5$ sans rien changer d'autre aux données du problème. Théoriquement la forme devrait converger vers un anneau "ultra" fin, voyons ce que l'on obtient sur les figures 4.46 et 4.47.

La figure 4.48 montre la condition d'optimalité dont la dernière valeur vaut 1.4×10^{-4} . La fonction coût et le volume converge. Ici la contrainte volumique a été prise en compte et joue son rôle : bien

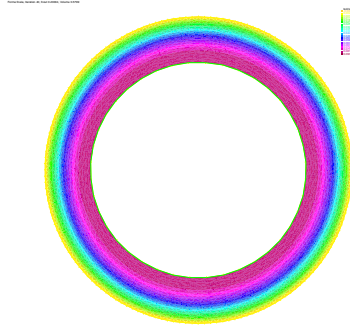


FIG. 4.43 – Forme finale (droite) aux bout des 40 itérations

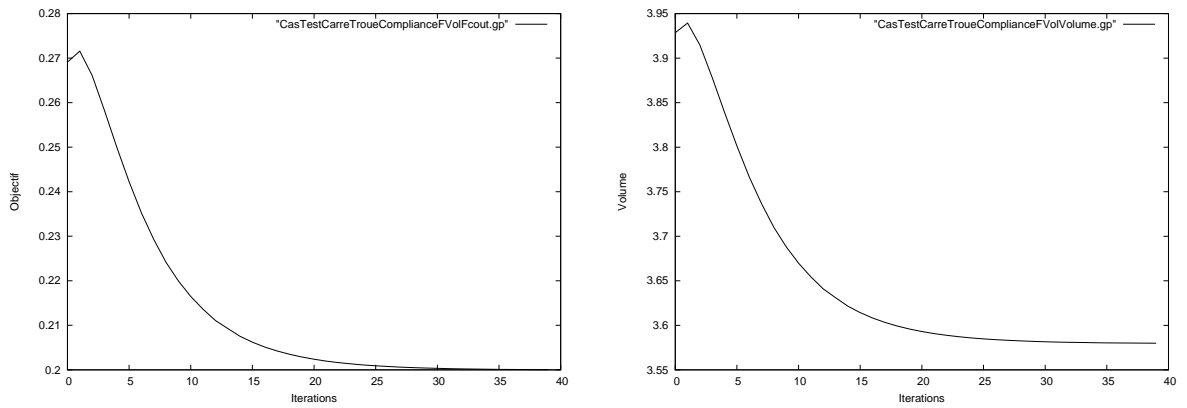


FIG. 4.44 – Fonction coût (gauche) et volume (droite) sur 40 itérations

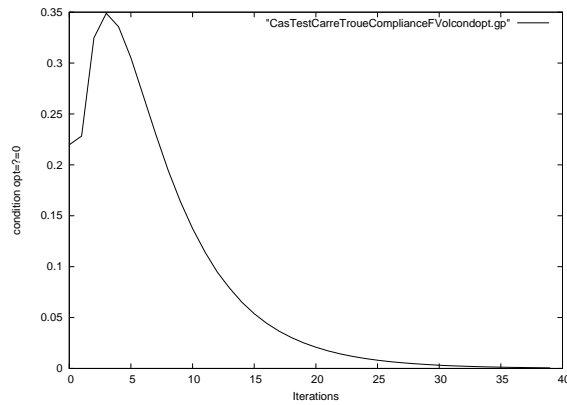


FIG. 4.45 – Tracé de la condition d'optimalité sur 40 itérations

que la fonction coût diminue en diminuant le volume de la forme, la contrainte volumique empêche de réduire le volume dans une certaine mesure (même remarque pour le cas précédent). On a donc bien convergence vers un anneau qui n'est pas ici ultra fin à cause de la fonctionnalité de la contrainte de volume dû à la valeur non négligeable de α_l prise.

Dans les deux cas, plus α_l est petit, plus la forme finale est fine.

4.5.5 Conduction L

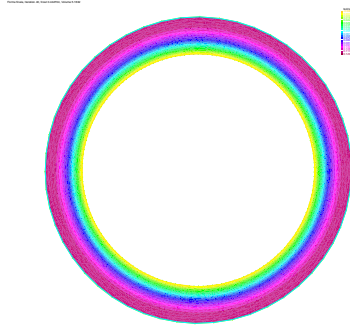


FIG. 4.46 – Forme finale (droite) aux bout des 40 itérations

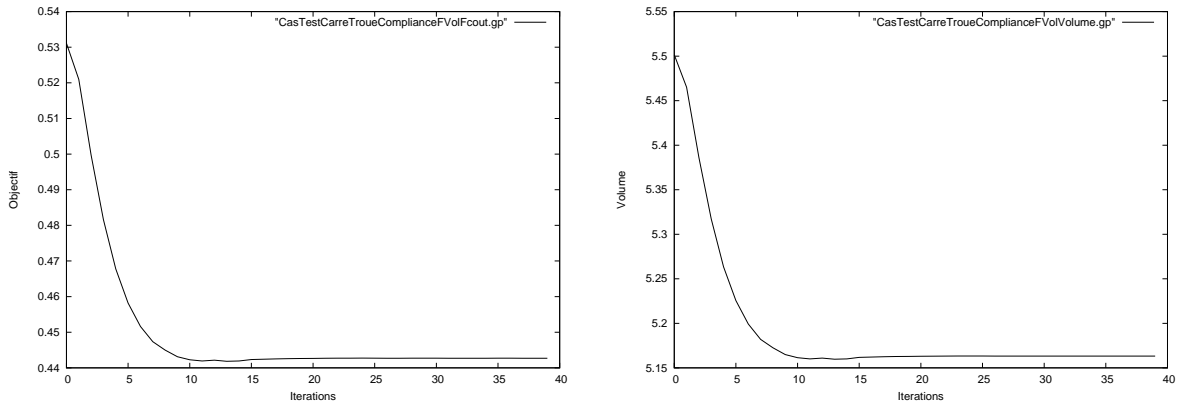


FIG. 4.47 – Fonction coût (gauche) et volume (droite) sur 40 itérations

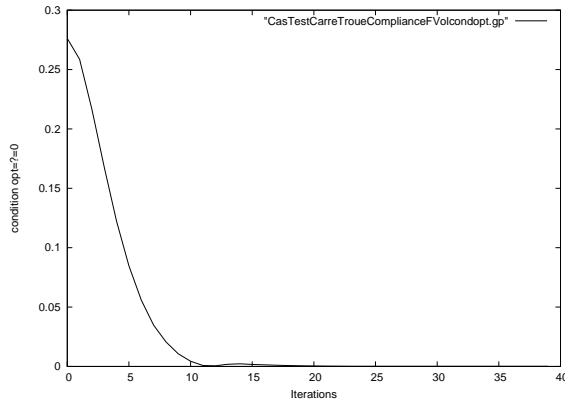


FIG. 4.48 – Tracé de la condition d'optimalité sur 40 itérations

Avec la compliance

Fichier utilisé : CasTestLCompliance.edp

Regardons la figure 4.49. La fonction *cut* est juste là pour indiquer que seul le coin est autorisé à varier. A droite on peut voir la forme initiale avec la solution sur le maillage. On pose $f = 0$ et $g = 1$ et un volume cible valant le volume initial 12. On chauffe sur le bord Neumann (bord à droit de la forme) et on maintient à température sur le bord Dirichlet (bord supérieur de la forme). Pour que $J(\Omega) = \int_{\Gamma_N} g u ds = \int_{\Gamma_N} u ds$ puisse diminuer, il faut que u diminue sur le bord Γ_N (on voit sur la forme initiale que u est le plus élevé sur le bord Γ_N). Pour ce faire, la température à l'intérieur du L

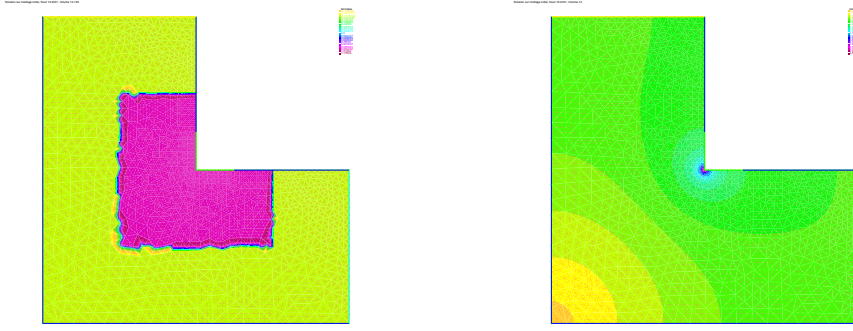


FIG. 4.49 – Fonction *cut* (gauche) et forme/solution initiale (droite)

doit diminuer en moyenne, et donc la forme doit s'agrandir pour permettre une homogénéisation de la température. Sur la forme initiale, u vaut 5,40 sur Γ_N . Regardons ce que donne l'optimisation sur la figure 4.50 :

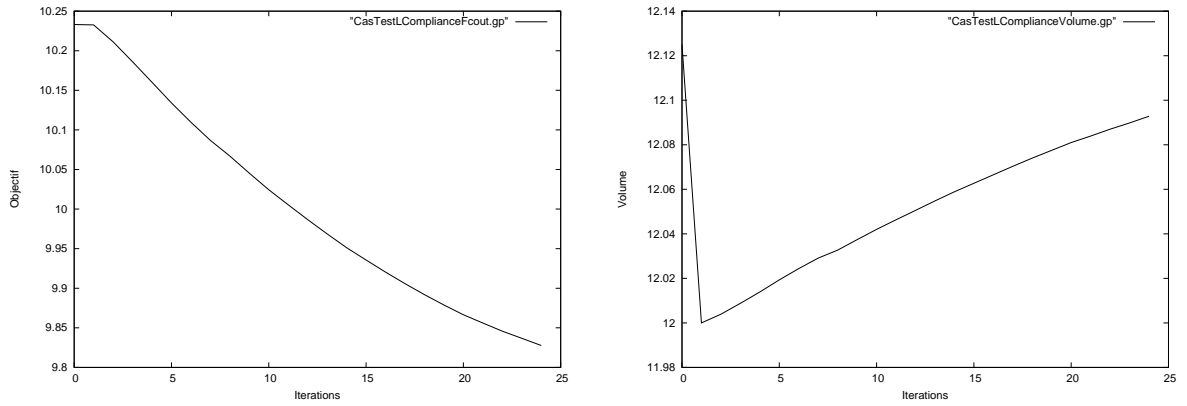


FIG. 4.50 – Valeur de la fonction coût (gauche) et volume (droite) sur 25 itérations, α_l petit

Au bout de 25 itérations, l'algorithme crash, à cause d'une collision de triangles (les collisions de triangles sont dures à maîtriser : l'utilisation de `checkmovemesh` dans `freefem` est soumise à un jeu de paramètres). Deux raisons à cela : le pas de descente était trop grand pour éviter la collision, et l'irrégularité du coin du L s'est propagé et a fini par créer cette situation de collision. On constate parallèlement que la fonction coût a diminué, que le volume a bien augmenté et que u a diminué sur le bord Γ_N et vaut maintenant 5,20. La figure 4.51 montre la forme finale juste avant le crash des triangles :

Pour éviter de se propager l'irrégularité du coin, il suffit de remplacer le coin par un petit bord arrondi. En faisant cela et sur 180 itérations on obtient la figure 4.52 :

La fonction coût diminue puis commence à stagner et le volume ne fait qu'augmenter. Voyons la forme finale sur la figure 4.53 juste avant collision sur un des deux bords.

On remarque que u vaut 5,01 sur Γ_N : conforme à l'intuition. De plus, il est clair ici que le rôle de $\alpha_l = 2$ est minime ici : le volume explose du fait que la fonction coût diminue facilement en augmentant le volume et parce que la quantité $\frac{V-V_0}{V_0}$ n'est que plus petite que V_0 est relativement grand par rapport à $V - V_0$ (qui dépend en partie si l'on autorise de grand changement sur la frontière). Par contre on n'a pas de création/transport d'irrégularité du fait que l'on commence avec un coin arrondi. Regardons sur la figure 4.54 maintenant l'effet d'un α_l assez élevé de 350 sur 60 itérations :

La fonction coût à une bonne allure et stagne sur la fin, de même que le volume. Voyons sur la figure 4.55 la forme et la solution finale :

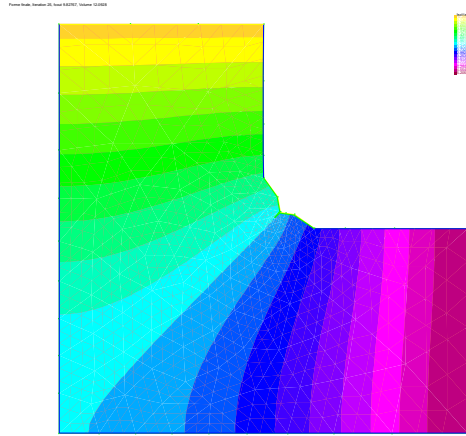


FIG. 4.51 – Solution finale avant crash, Volume=12,0928

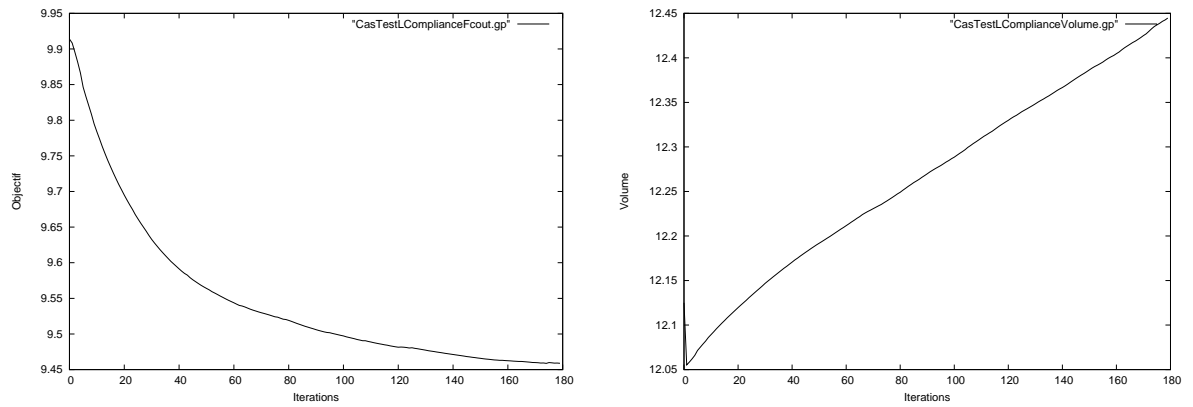


FIG. 4.52 – Fonction coût (gauche) et volume (droite) sur 180 itérations, α_l petit, coin arrondi

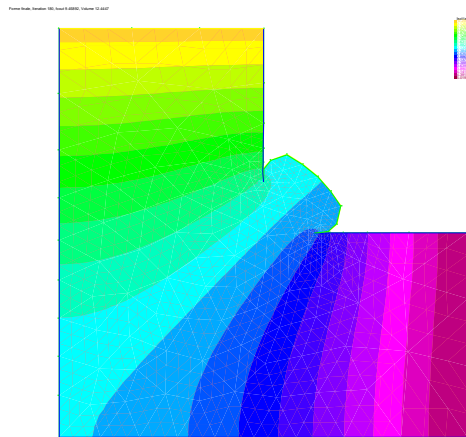


FIG. 4.53 – Solution finale avant crash, Volume=12,4447

La forme finale est de volume proche de $12+1/8$ qui correspond au volume de la forme d'un L avec un raccord diagonal pour "boucher" le coin. On a donc réussi à contrôler l'augmentation massive du volume en renforçant α_l et pour au final se retrouver avec une forme de volume final proche de $12+1/8$. La valeur $\alpha_l = 350$ a été trouvée en tâtonnant : il y a donc bien un paramétrage de α_l à effectuer pour arriver à ses fins. Notons qu'encore une fois, u a diminué sur Γ_N et vaut 5,14 maximum.

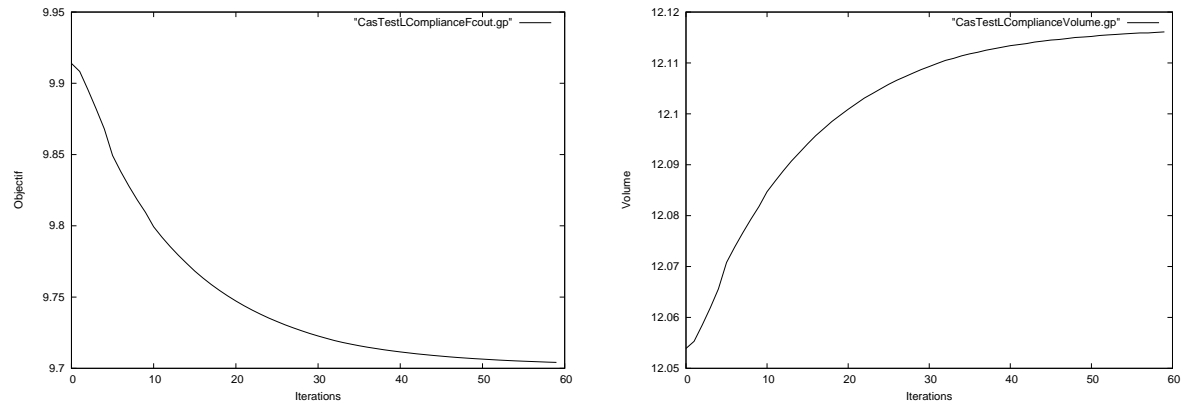


FIG. 4.54 – Fonction coût (gauche) et volume (droite) sur 180 itérations, α_l grand, coin arrondi

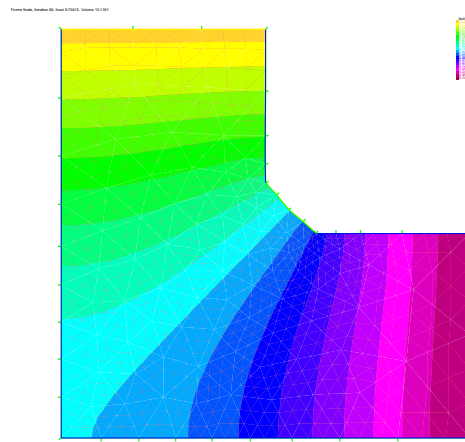


FIG. 4.55 – Solution finale , Volume=12,1161

Regarde maintenant le cas où le volume cible est plus petit que le volume initial. Etant donné que l'optimisation tend à augmenter le volume, voyons comment la forme va évoluer. On peut voir sur la figure 4.56 les formes aux itérations 100 et 400 et sur la figure 4.57

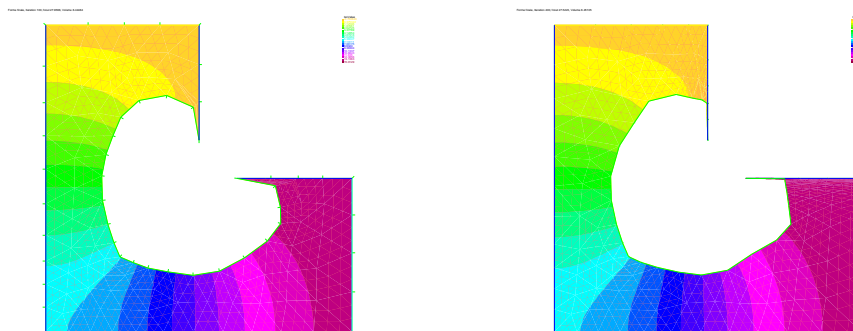


FIG. 4.56 – Forme à la 100ème itération (gauche) et à la 400ème (droite)

L'optimisation aura donc tendance évidemment à creuser la forme pour satisfaire la contrainte de volume dans un premier temps. La fonction coût augmente donc au départ. Elle diminuera par la suite. L'optimisation a tendance à vouloir faire disparaître les bords du coin.

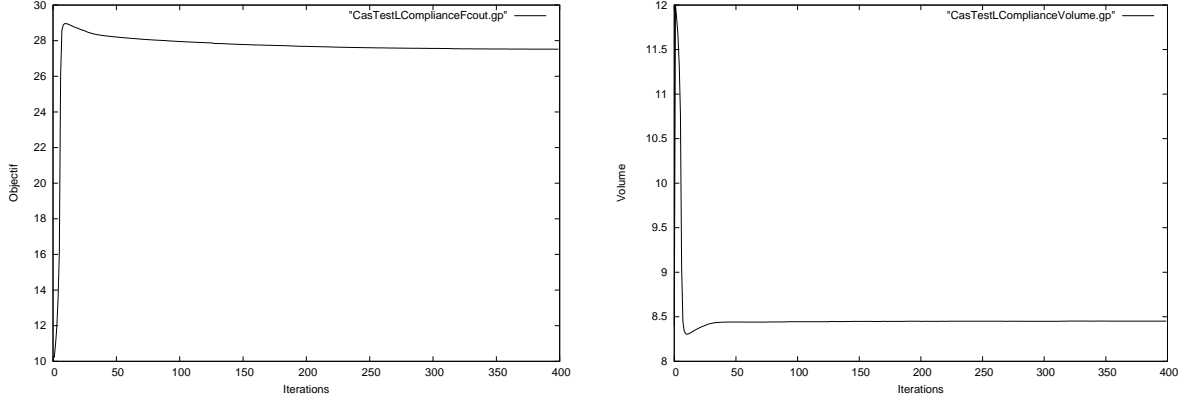


FIG. 4.57 – Fonction coût (gauche) et volume (droite) sur 400 itérations, α_l grand, coin arrondi

Avec l'énergie de déformation

Fichier utilisé : CasTestLGradient.edp

On considère la fonction coût :

$$J(\Omega) = \int_{\Omega} k(x) |\nabla u|^2 dx$$

Si $k = 1$ on obtient les mêmes résultats qu'avec la compliance. Si on prend k valant 1 sur un cercle centré en l'origine de rayon 0.25 (sachant que la partie qui est optimisée est de l'ordre de 0.5) et vallant 0.2 sur le reste du domaine. On concentre donc l'optimisation sur le cercle centré de rayon 0.25.

La figure 4.59 donne une image plus précise du k mis en jeu. En prenant 100 itérations, 0.005 comme pas, $f = 0$ et $g = 1$ et un volume cible de $12 + 1/8$ on obtient l'évolution de la fonction coût et du volume sur la figure 4.58.

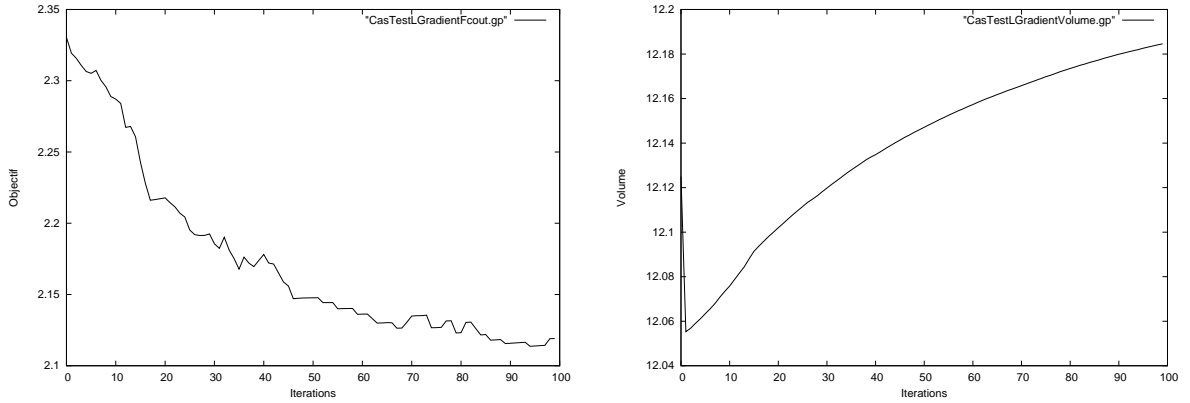


FIG. 4.58 – Fonction coût (gauche) et volume (droite) sur 100 itérations, $\alpha_l = 50$, coin arrondi

La fonction coût diminue bien mais chaotiquement, avec de faibles augmentations par accoups. Le volume ne fait qu'augmenter (la diminution de départ est due au fait que le volume à l'itération de départ est le volume cible qui est supérieur au vrai volume de la forme initiale). La fonction coût semble entamer une stagnation sur la fin : il faudrait continuer au delà de 100 itérations pour voir la suite.

On voit que l'optimisation a privilégié d'abord les bords optimisables situés dans le cercle de rayon 0.25, ce qui a eu pour effet l'augmentation du volume dans cette zone plus rapidement que dans les autres zones, d'où l'aspect bombé de la partie optimisée finale.

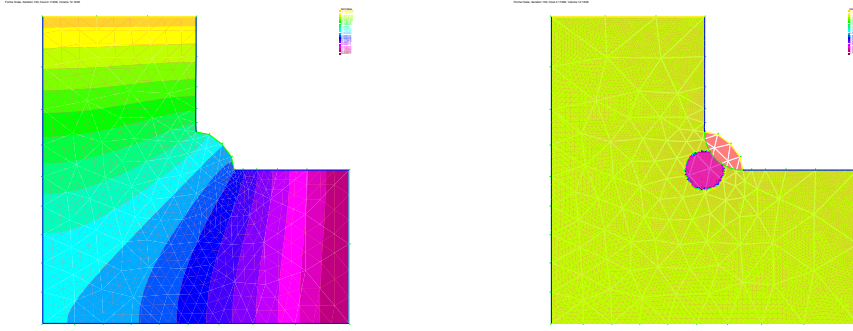


FIG. 4.59 – Forme finale, 100 itérations (gauche) et $k = \mathbf{1}_{(x^2+y^2)<0.25^2} + 0.2 * \mathbf{1}_{(x^2+y^2)>0.25^2}$ (droite)

4.5.6 Conduction M

Voici le schéma sur la figure 4.60 du M utilisé en conduction dans la suite :

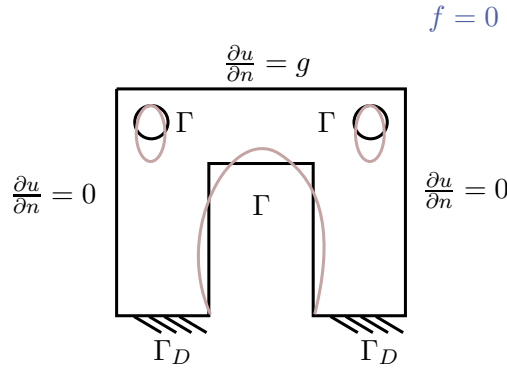


FIG. 4.60 – Schéma de la conduction sur le M (troué ou non)

Avec la compliance

Fichier utilisé : CasTestMCompliance.edp

On prend $f = 0$, $g = 1$ et un pas de 0.025. On se trouve dans le même type de problème que le L. On chauffe à un endroit (bord supérieur) et on maintient à une température à un autre (bords Dirichlet). L'optimisation aura donc tendance à vouloir augmenter le volume, tout aussi drastiquement que pour le L. Il faudra donc poser un α_l assez élevé pour optimiser tout en essayant de satisfaire la contrainte.

Regardons d'abord le M sans les trous, pour α_l petit et avec un volume cible valant le volume initial. Voici l'évolution de la fonction coût et du volume sur la figure 4.61 sur 9 itérations.

A l'itération 10, l'irrégularité des coins fait arrêter l'algorithme. La singularité au coin se propage lors de l'optimisation, la forme ayant tendance à s'agrandir. On peut voir la forme à la 9e itération sur la figure 4.62 : on remarque que le coin s'est refermé sur lui même.

Pour éliminer l'irrégularité, plaçons des formes arrondies au lieu des coins. Avec ce changement on obtient les figures 4.63, 4.64 et 4.65 sur 100 itérations.

On constate bien une augmentation du volume et une diminution de la fonction coût jusqu'à convergence vers un palier. La forme n'a pas hérité d'irrégularités. La condition d'optimalité a convergé vers

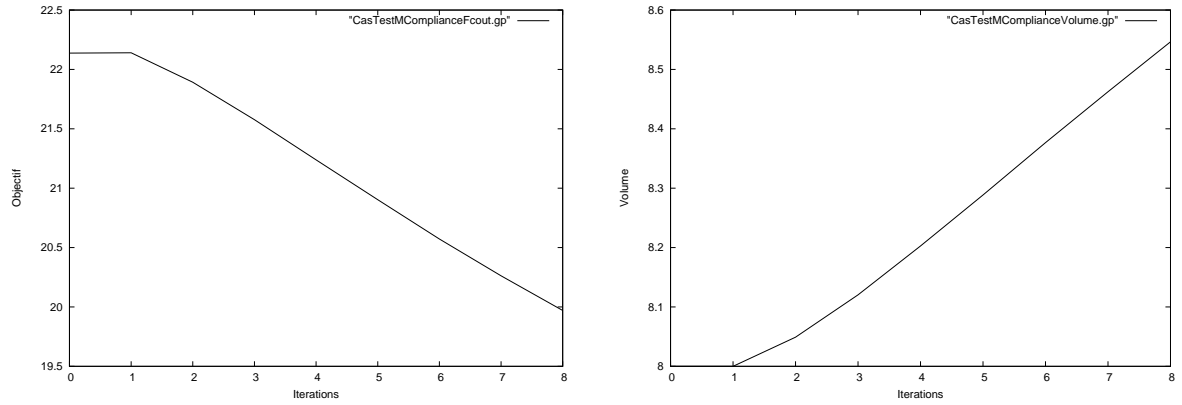


FIG. 4.61 – Fonction coût (gauche) et volume (droite) sur 9 itérations, α_l petit

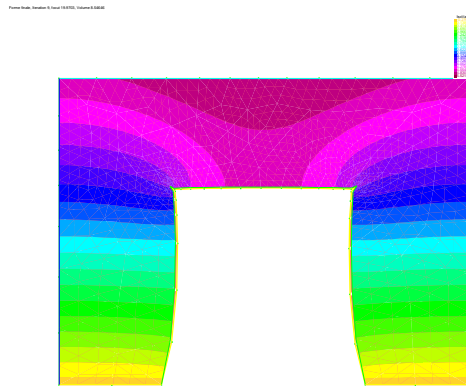


FIG. 4.62 – Forme itération 9

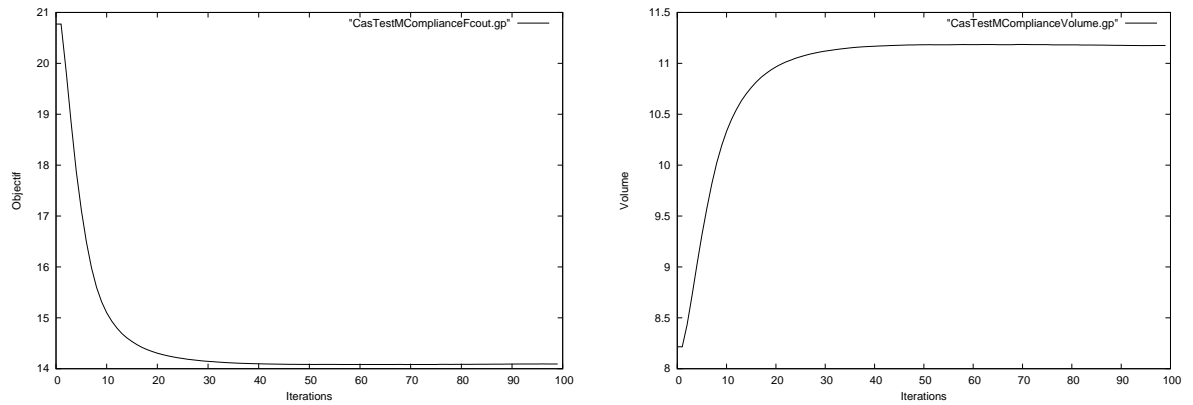


FIG. 4.63 – Fonction coût (gauche) et volume (droite) sur 100 itérations, α_l petit

0 (dernière valeur 0.017). Remarquons que du moment que la fonction coût stagne, cela signifie que la forme ne varie pas trop, ce qui donne à chaque itération une solution de la conduction sensiblement la même, donc un C_{opt} ne variant pas beaucoup.

Si on regarde maintenant le même problème (avec les mêmes données) mais en prenant $\alpha_l = 200$ pour forcer la forme à tendre dès le début vers le volume souhaité. Sur 200 itérations on obtient les figures 4.66, 4.67 et 4.68.

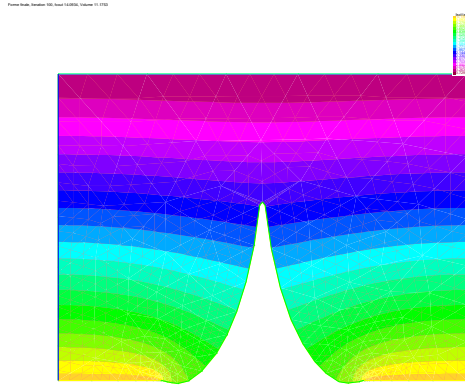


FIG. 4.64 – Forme itération 100

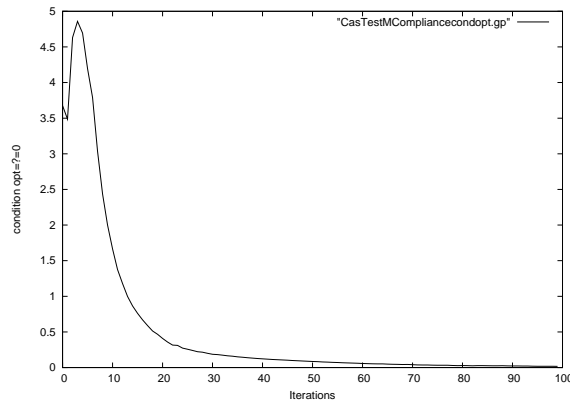


FIG. 4.65 – C_{opt} sur 100 itérations

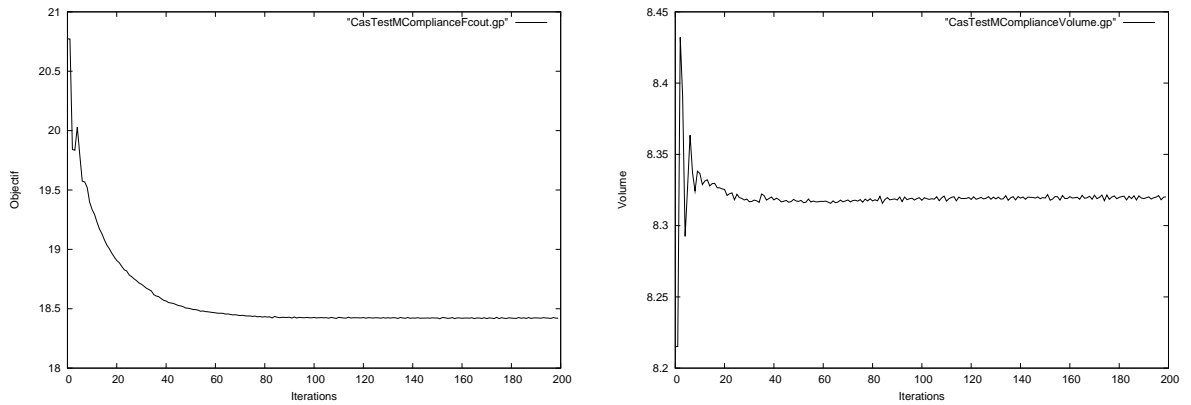


FIG. 4.66 – Fonction coût (gauche) et volume (droite) sur 200 itérations, α_l grand

On constate bien une diminution de la fonction coût, une approximation du volume vers le volume cible notable et la convergence de la condition d'optimalité vers 0 (dernière valeur : 0.024). La forme peut difficilement encore être améliorée, la pente de la fonction coût sur les dernières itérations étant assez proche de 0 (la fonction coût oscille sur la fin du fait qu'elle a atteint un optimum).

Regardons le maillage troué de deux trous. Avec α_l petit on obtient les figures 4.69), (4.70 et 4.71.

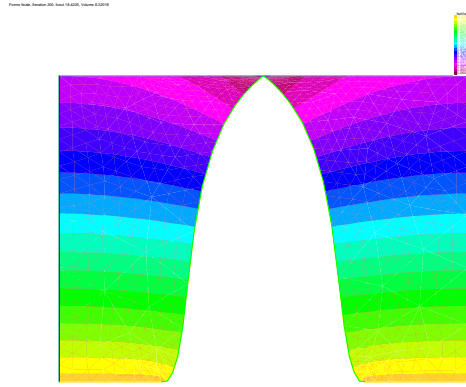


FIG. 4.67 – Forme itération 200

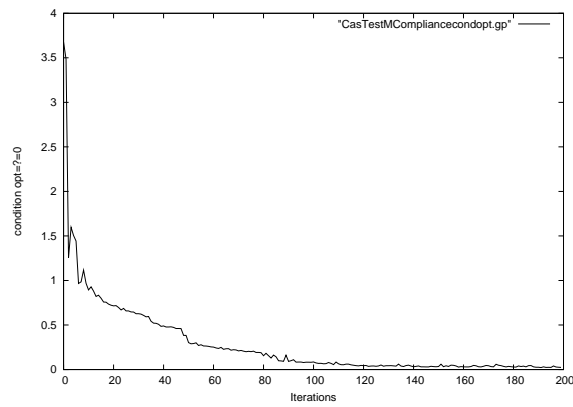


FIG. 4.68 – C_{opt} sur 200 itérations

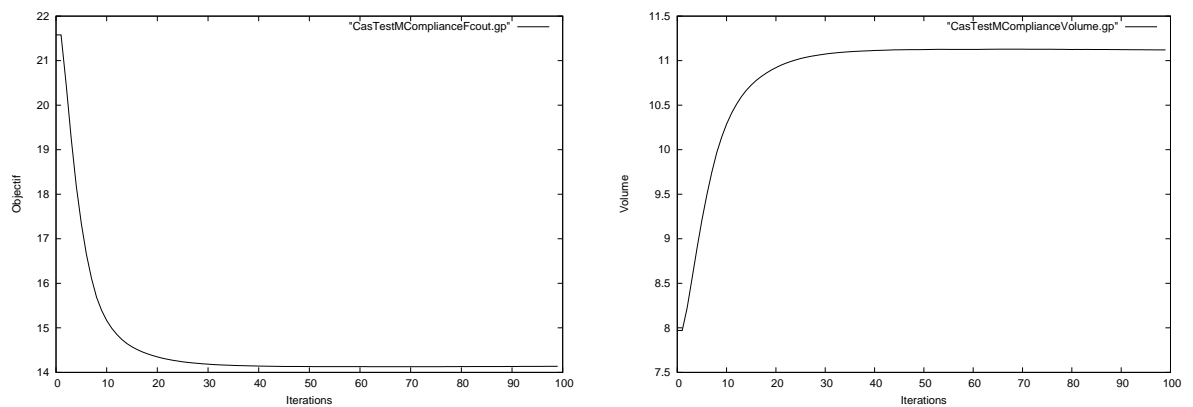


FIG. 4.69 – Fonction coût (gauche) et volume (droite) sur 100 itérations, α_l petit, maillage troué

La forme tend à s'agrandir, et à faire disparaître les trous. La fonction coût diminue bien, le volume augmente et converge vers un seuil.

La condition d'optimalité converge vers 0 (dernière valeur 0.061).

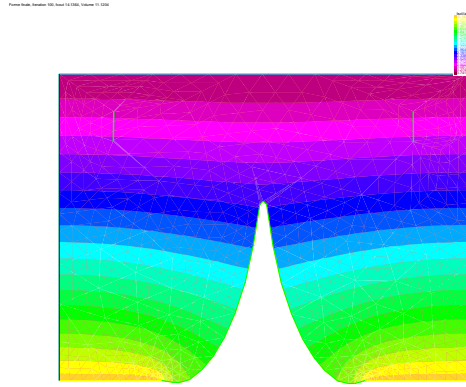


FIG. 4.70 – Forme itération 100, maillage troué

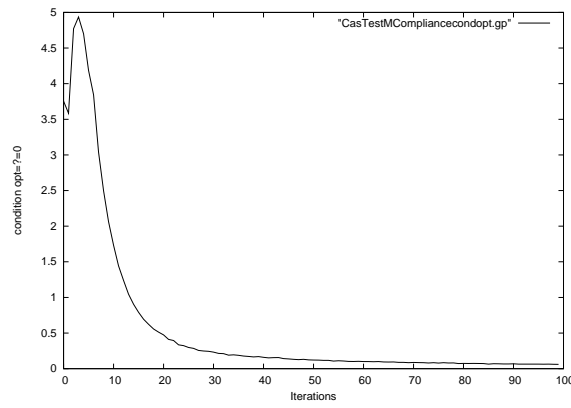


FIG. 4.71 – C_{opt} sur 100 itérations, maillage troué

En augmentant, sur le même exemple, la valeur de α_l à 200, on obtient les résultats suivants sur les figures 4.72, 4.73 et 4.74.

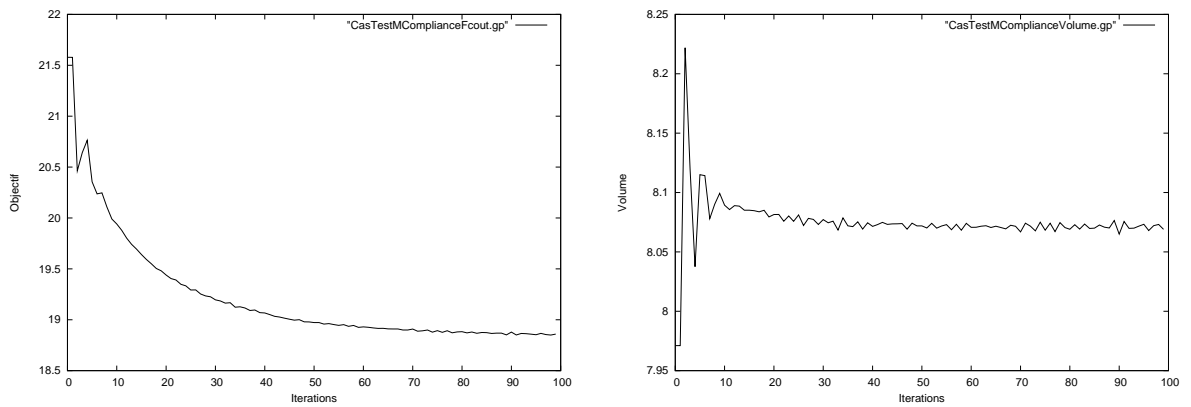


FIG. 4.72 – Fonction coût (gauche) et volume (droite) sur 100 itérations, α_l grand, maillage troué

De même ici, la fonction coût diminue et converge, le volume s'approche du volume cible.

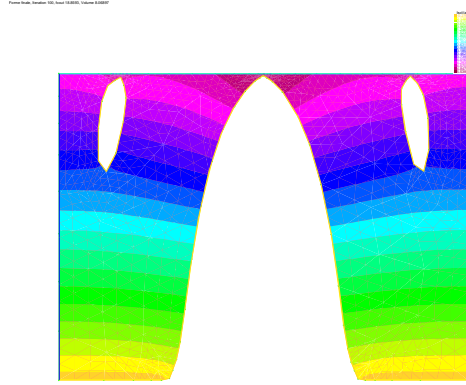


FIG. 4.73 – Forme itération 100, maillage troué

Quand à la condition d’optimalité, elle diminue aussi et ne converge pas tout à fait vers 0 mais presque (dernière valeur 0.18). L’optimisation est ici quasi terminée au vu de la courbe de la fonction coût.

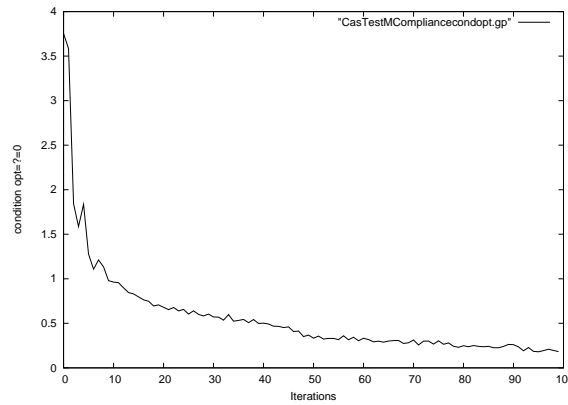


FIG. 4.74 – C_{opt} sur 100 itérations, maillage troué

La forme presque finale du M troué est bien ce à quoi l’on pouvait s’attendre : la formation d’une forme en "V" à l’envers, pour que le flux de chaleur s’harmonise mieux avec les deux branches menant aux conditions Dirichlet et l’allongement des trous dans la direction du flux de chaleur pour améliorer son passage.

On n’a ici pas encore touché au pas. On a pris un pas de 0.025 : un pas de 0.005 aurait convergé mais en beaucoup plus d’itérations. Par contre un pas plus grand n’aurait pas forcément convergé : il faut contrôler le pas pour un algorithme optimisé : voir la partie 4.7. La condition d’optimalité a bien été vérifiée pour chaque cas testé : il y a bien convergence vers 0. La contrainte volumique n’est quand à elle vérifiée qu’approximativement. Il existe d’autres procédures pour approximer le volume mais ce n’est pas la question ici.

Regardons maintenant l’optimisation d’un autre critère sur la même forme : critère de l’énergie de déformation localisée.

Avec l'énergie de déformation

Fichier utilisé : CasTestMGradient.edp

Le cas $k = 1$ donne les mêmes résultats que la compliance. Prenons $k = \mathbf{1}_{x < 0} + 0.2 * \mathbf{1}_{x > 0}$. On localise sur la partie gauche du M l'effort d'optimisation. On prend $\alpha_l = 200$, pas de 0.005, sur 300 itérations. La figure 4.75 montre l'évolution de la fonction coût et du volume et la figure 4.76 montre la forme finale et la fonction k de localisation.

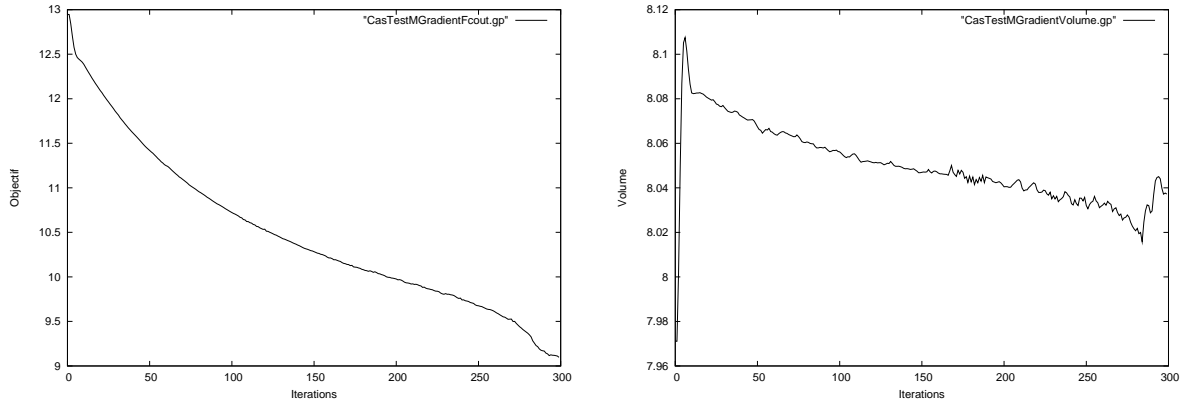


FIG. 4.75 – Fonction coût (gauche) et volume (droite) sur 300 itérations, $\alpha_l = 200$, maillage troué

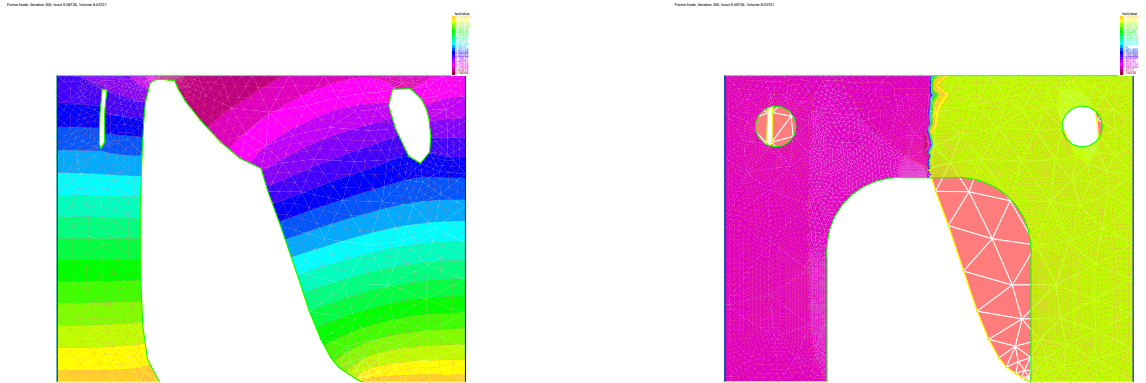


FIG. 4.76 – Forme finale (gauche) et fonction $k = \mathbf{1}_{x < 0} + 0.2 * \mathbf{1}_{x > 0}$ (droite)

La fonction coût diminue tranquillement, sa diminution n'étant apparemment pas terminée. Le volume suit une évolution comparable au cas de la compliance mais plus chaotiquement. La forme optimale privilégie donc le passage du flux de chaleur sur la gauche du domaine. A gauche le trou s'est bien allongé dans la direction du flux alors qu'à droite il est encore sous forme ovale.

Code optimisé

Utilisons sur le même exemple avec la compliance le code optimisé et comparons les résultats. Sur la figure 4.77 on trace les courbes sans et avec optimisation du code.

L'optimisation de forme sur cet exemple est en gros deux fois plus rapide avec l'optimisation du code que sans. Voir la partie 4.7 pour des précisions sur le code optimisé.

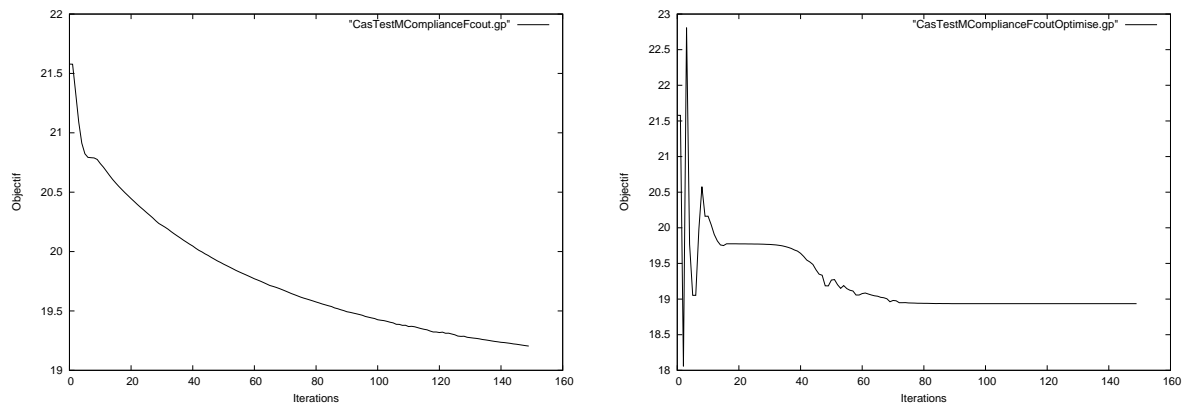


FIG. 4.77 – Comparaison sans optim du code (gauche) et avec (droite) des fonctions coûts sur le M

4.5.7 La console en élasticité linéarisée

Avec la compliance

Fichier utilisé : ElasticiteComplianceConsole-Allaire.edp

Le code de G. Allaire et O. Pantz "ElasticiteComplianceConsole-Allaire.edp" traite, comme son nom l'indique, de l'évolution de la forme d'une console (cantilever en anglais) avec comme fonction coût la compliance dans le cas de l'élasticité linéarisée. Nous allons étudier l'évolution de la fonction coût et du volume de cette console. Après lancement du code (qui est optimisé, voir partie 4.7), on obtient les formes initiale et finale suivantes sur la figure 4.78 au bout de 30 itérations, avec un α_l petit. La figure 4.79 montre l'évolution de la fonction coût et du volume.

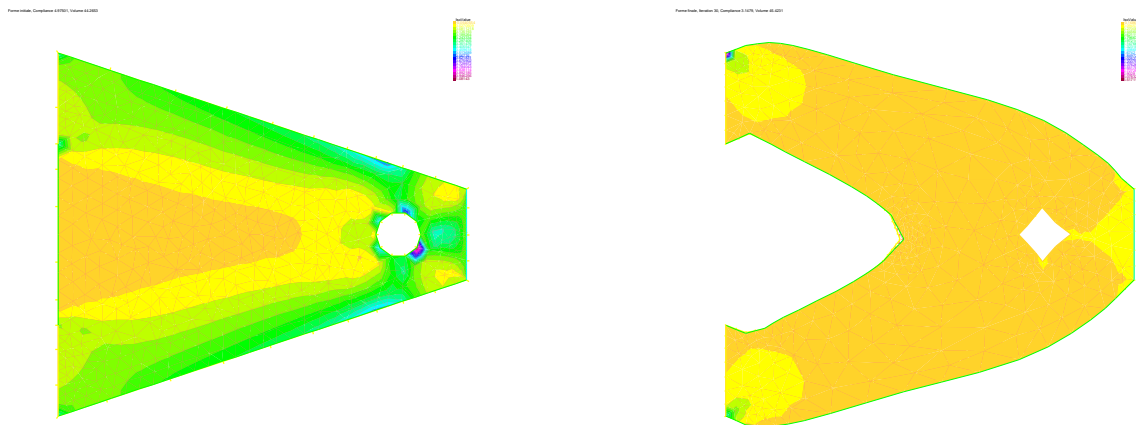


FIG. 4.78 – Forme initiale (gauche) et finale (droite)

La courbe de la fonction coût diminue avec quelques sursauts duent aux retour en arrière qu'effectue l'optimisation dans le code avec la direction de descente. Le volume diminue aussi : on remarque que si l'on prend un α_l élevé, la forme de base de la console n'évolue pas. l'optimisation de la console passe forcément par une diminution du volume. Voyons comment évolue une console avec plusieurs trous à l'initialisation sur la figure 4.80.

En jouant sur le nombre de trous on constate un effet sur la valeur minimale que l'on peut trouver pour la compliance, tournant autour de 3.50. Plus il y a de trous, meilleur est la compliance (diminution de la compliance) : ici sur la forme avec une multitude de trous la compliance finale est de 3.50. Le mince filament qui s'est crée sur la gauche de la figure tend à penser que la forme souhaite une suppression de

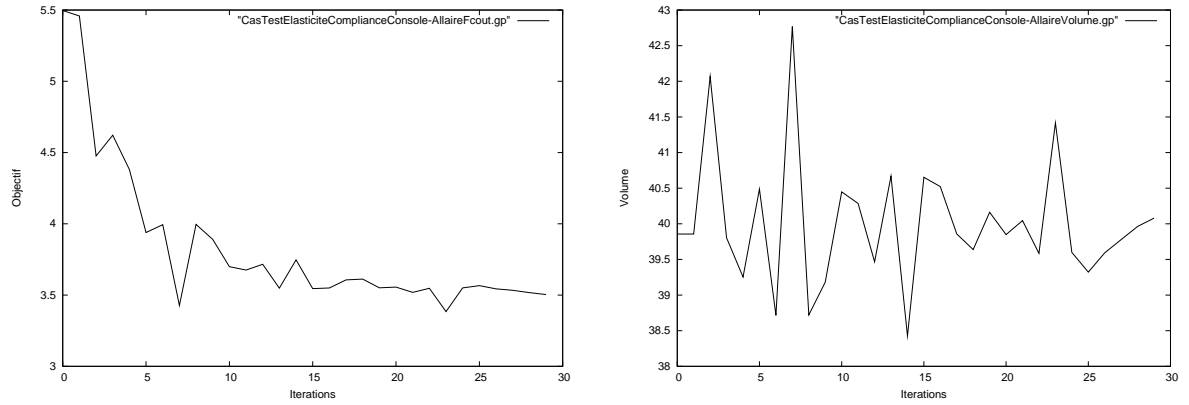


FIG. 4.79 – Fonction coût (gauche) et volume (droite) sur 30 itérations, $\alpha_l = 2$, maillage console

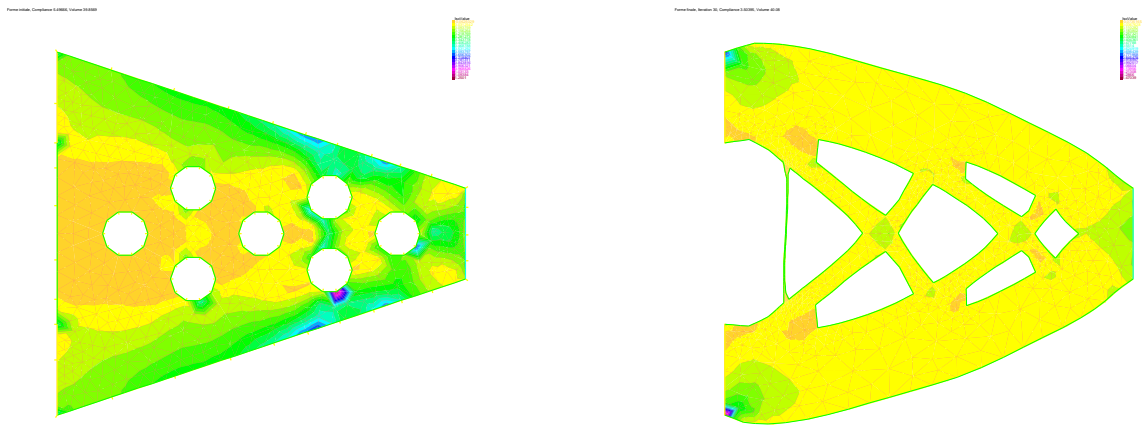


FIG. 4.80 – Forme initiale (gauche) et finale (droite)

ce filament ce qui reviendrait à la suppression d'un trou : optimisation non possible en optimisation par variation de frontière (géométrique). Mais on peut recommencer l'optimisation en supprimant le trou correspondant au départ de l'optimisation et on obtient la figure 4.81 dont la valeur de la compliance au bout de 30 itérations est de 3.44 et au bout de 60 itérations est de 3.40.

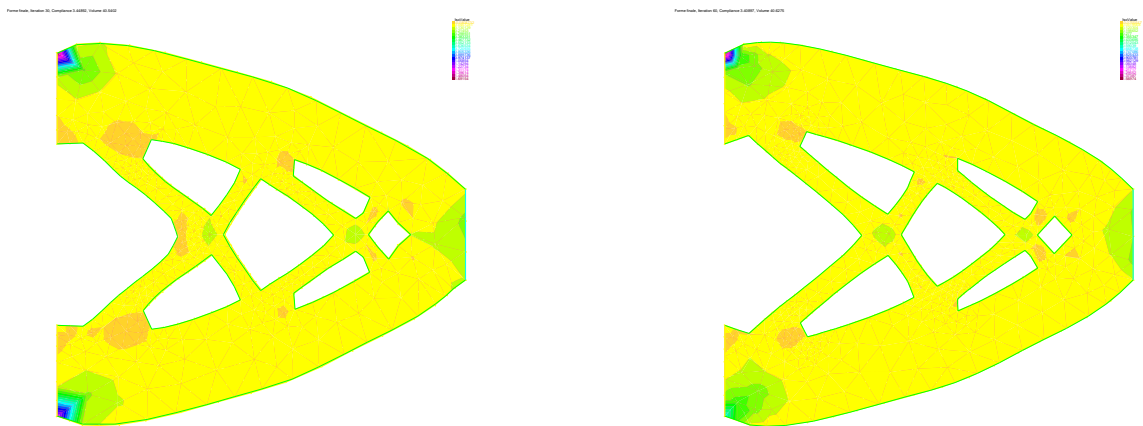


FIG. 4.81 – Forme a la 30ième itération (gauche) et finale (60ième itération) (droite)

Voici les courbes de fonction coût et de volume sur la figure (4.82) :

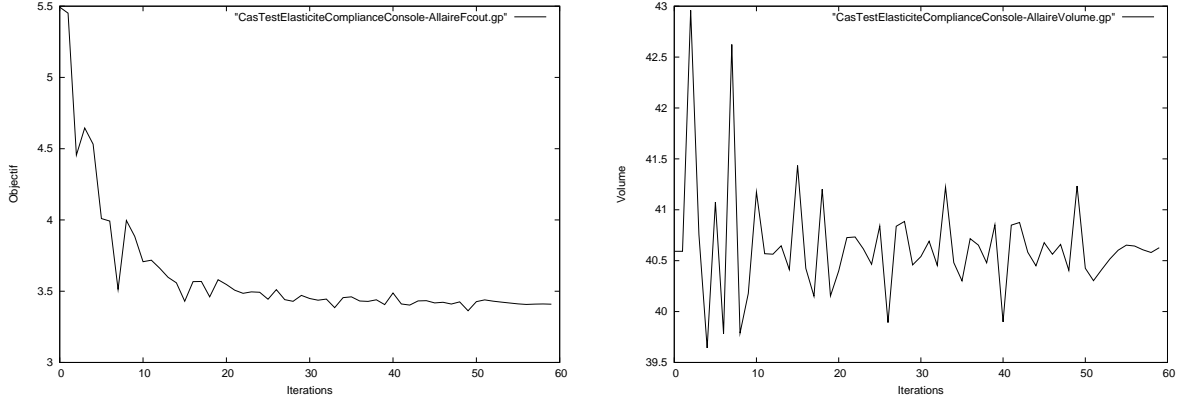


FIG. 4.82 – Fonction coût (gauche) et volume (droite) sur 60 itérations, $\alpha_l = 2$, maillage console

A noter que pour l'exemple avec les 7 trous, une optimisation au delà de 30 itérations provoque un croisement de mesh du au fait que l'optimisation tend a vouloir faire disparaître un trou comme expliqué précédemment. Et ce malgré la précaution de l'utilisation de checkmovemesh pour prévenir ce genre de chose (voir partie 4.7.2)

Avec l'énergie de déformation

Fichiers utilisés : CasTestElasticiteGradientConsoleOptimise.edp ou CasTestElasticiteGradientConsole.edp

En reprenant le code avec la compliance de la section précédente et en y ajoutant les fonctionnalités liées à l'utilisation d'une nouvelle fonction coût énergie de déformation avec poids de la forme

$$\int_{\Omega} k(x) A e(u) : e(u) dx$$

on obtient les deux codes de cette section : un optimisé et l'autre non.

Regardons si en prenant $k = 1$ on est conforme aux résultats théoriques de la partie 4.3.3 nous indiquant que ce problème peut être vu comme une minimisation de la compliance. Pour une meilleur optimisation on prendra l'algorithme optimisé comme algorithme test. On se place sur la console trouée de 6 trous.

Ainsi les formes initiales et finales obtenues au bout de 60 itérations sont celles de la figure 4.83. Les courbes de la fonction coût et du volume peuvent être visualisées sur la figure 4.84.

On remarque que les deux algorithmes avec compliance et énergie de déformation ont conduit à presque la même forme, en assurant une diminution et une convergence de la fonction coût dans les deux cas. L'algorithme avec l'énergie de déformation semble sur un premier test bien écrit dans le sens où il conduit à des résultats similaires à ceux de la compliance (différents rien qu'au vu de l'évolution des fonctions coûts mais similaire pour la forme : on n'obtient pas exactement le même résultat peut être à cause du calcul du lagrangien qui se révèle différent et de la nécessaire extension de la normale à l'intérieur du domaine, entrainant une évolution de la forme quelque peu ralentie).

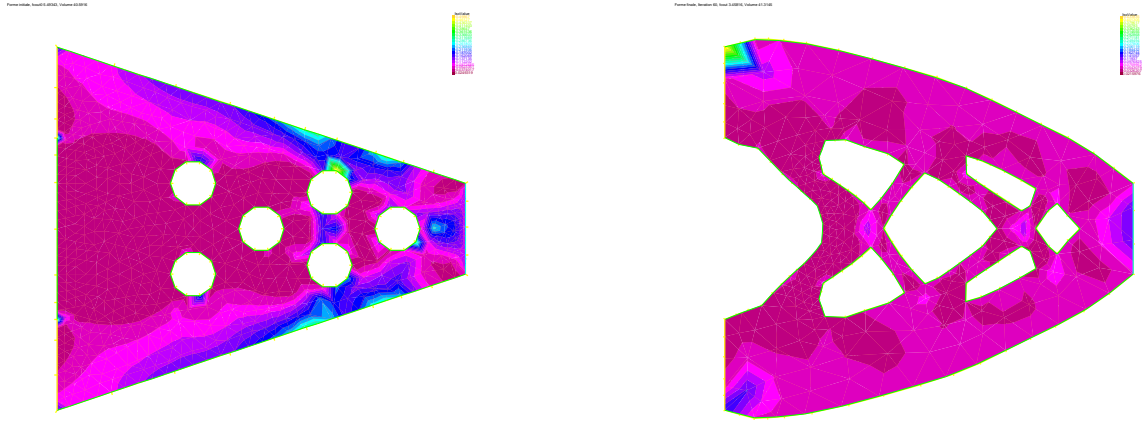


FIG. 4.83 – Forme initiale (gauche) et finale (droite)

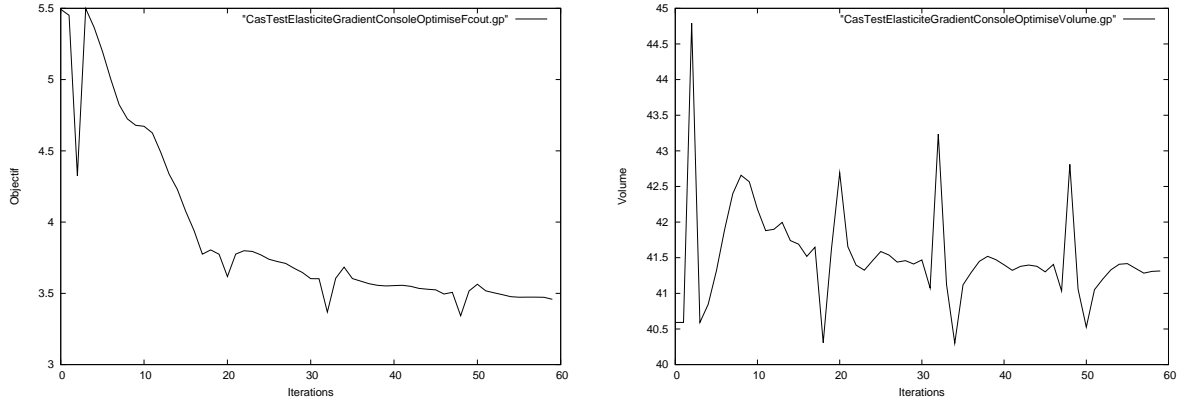


FIG. 4.84 – Fonction coût (gauche) et volume (droite) sur 30 itérations avec $\alpha_l = 2$

4.6 Contrainte géométrique

On souhaite ici imposer une autre contrainte cette fois ci sur la géométrie de la forme. On souhaite que la forme ne puisse pas évoluer dans une certaine zone. Pour ce faire il existe plusieurs méthodes. Nous choisissons tout d'abord de retranscrire la contrainte en utilisant une pénalisation avec un multiplicateur de lagrange et une certaine fonction continue qui sera positive sur la zone interdite et nulle sur le reste de Ω . On construit donc :

$$\mathcal{L}(\Omega) = J(\Omega) + \frac{1}{\delta} \int_{\Omega} F(x)^2 dx$$

On prend $F = \max(0, \min(x_1, x_2))$ avec x_1 et x_2 les coordonnées d'un point dans le plan. F représente la partie positive de la fonction $\min(x_1, x_2)$ dont voici une représentation (exemple du L) sur la figure 4.6. La fonction F est nulle sur la partie grise et non nulle sur la partie "interdite", le tout étant continu.

En dérivant ce nouveau lagrangien on obtient :

$$\left\langle \frac{\partial \mathcal{L}}{\partial \Omega}, \theta \right\rangle = \left\langle J'(\Omega), \theta \right\rangle + \frac{1}{\delta} \int_{\partial \Omega} F^2 \theta \cdot n ds = \left\langle J'(\Omega), \theta \right\rangle + \frac{1}{\delta} \int_{\Gamma} F^2 \theta \cdot n ds$$

On pourra faire varier δ à chaque itération pour renforcer la contrainte géométrique : par exemple on peut typiquement diviser δ par 2 à chaque itération. Le fait de faire varier δ va permettre de contrôler la force que l'on souhaite accorder à la contrainte géométrique, mais cela nous rendra dépendant de cette paramétrisation.

Voici un premier exemple effectué sur le L (fichier **CasTestLComplianceOptimise.edp**) habituel en diminuant δ au fur et à mesure des itérations, figure 4.85 :

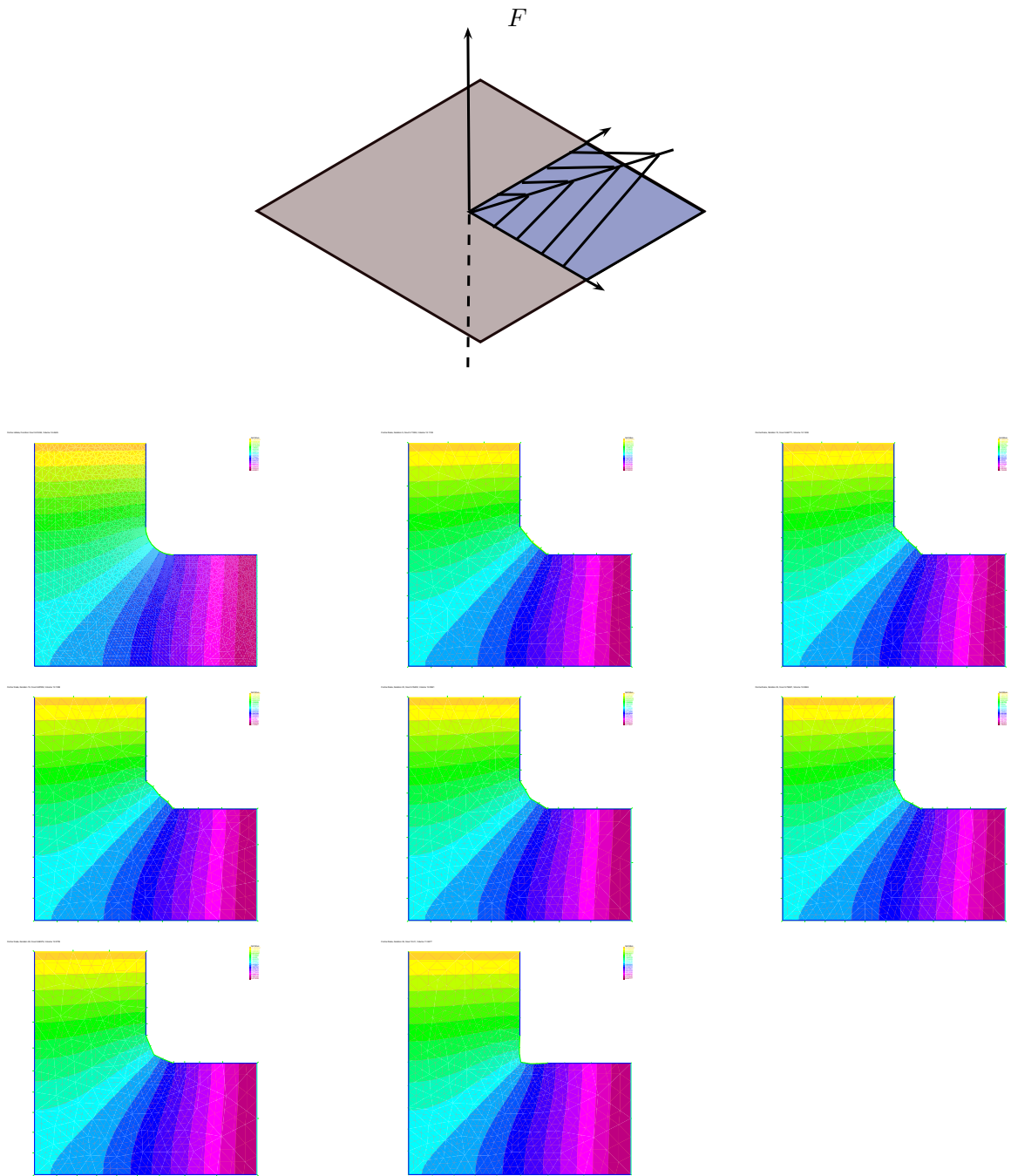


FIG. 4.85 – Différentes étapes de l'optimisation, initiale, étape 5, 10, 15, 20, 25, 30 et 35

La forme commence par augmenter de volume, et, au fur et à mesure que δ diminue, rendant encore plus pressante la contrainte géométrique, la forme va commencer à vouloir s'assurer que rien n'existe sur la "zone interdite". On se retrouve ainsi avec la forme à l'itération 35, "point d'équilibre" entre la volonté de la forme à augmenter sa surface et la contrainte géométrique.

4.7 Optimisation du code

4.7.1 Direction de descente

Choisir le pas à chaque étape est difficile : lorsqu'il est trop petit l'algorithme risque de ne pas converger ou d'avoir un taux de convergence faible, trop grand l'algorithme est instable. Un choix est fait en comparant les directions de descente précédentes et actuelles. Si les directions de descente sont dans des sens opposés, le produit scalaire $(d_n, d_{n-1})_{H^1(\Omega)^n}$ est négatif et on présente que l'algorithme devient instable. On diminue alors le pas de descente et on revient en arrière : l'itération suivante sera initialisée avec la forme précédente Ω_{n-1} . Si $(d_n, d_{n-1})_{H^1(\Omega, \mathbb{R}^n)}$ est positif et assez petit, le pas est augmenté. Voir [1] pour plus d'information et le code suivant utilisé :

```
// On calcule la norme (au carre) du gradient de forme
norme=int2d(Th)(dx(eui)^2+dx(evi)^2+dy(eui)^2+dy(evi)^2+eui^2+evi^2);
// On compare le gradient de forme avec le gradient precedent
produit=int2d(Th)(dx(eui)*dx(eui2)+dx(evi)*dx(evi2)+dy(eui)*dy(eui2)+dy(evi)*dy(evi2)
               +eui*eui2+evi*evi2);

////////////////////////////////////
// Mise a jour du pas de descente //
////////////////////////////////////
// Si le gradient de la fonction cout est dans la meme direction que le gradient
// de l'iteration precedente (produit>0), on augmente le pas de descente.
// Si le gradient de la fonction cout n'est pas dans la meme direction que le
// gradient de l'iteration precedente (produit<0), on fait marche arriere et
// on diminue le pas de descente (on evite ainsi les oscillations).

if ((produit<0)&(!arriere))
    {pasef=pasef*norme/(norme-produit)/4.;
      cout<<"***** Pas de descente trop important: Marche arriere !!! *****"<<endl;
      Th=Thprec;
      arriere=true;}
else
    {arriere=false;}
;
if ((produit>0)&(produit<(norme/2.)))
    {pasef=pasef*2*norme/(2*norme-produit);
      cout<<"On augmente le pas de descente"<<endl;}
;
if (produit>(norme/2.))
    {pasef=pasef*4./3.;
      cout<<"On augmente le pas de descente"<<endl;}
;
```

4.7.2 Retournement de triangle

A chaque itération il y a le risque de voir des triangles se retourner lors de la variation du maillage. Pour éviter ça, lorsqu'on détecte qu'un triangle sera retourné, on diminue le pas de descente avant de déformer le maillage. La fonction FreeFem++ utilisée est `checkmovemesh` : voir la partie 6. L'utilisation de cette précaution comporte un problème, celui de devoir être paramétré sur le mouvement des mailles, ainsi on peut vouloir être sûr que les maillages ne se chevauchent pas, mais pour ce faire il serait indiqué de ne quasiment pas bouger le maillage... Voici le code utilisé :

```
real A,B=checkmovemesh (Th,[x,y])/100.;
while (B > (A=checkmovemesh(Th,[x+pasef*eui,y+pasef*evi])) )
{
    cout << "Probleme de triangle retourne : on diminue le pas de descente" << endl;
    pasef= pasef/2;
}
```

4.7.3 Contrôle du pas

A partir d'une certaine itération, on peut vouloir que la fonction coût n'augmente plus du tout. Il arrive en effet que quelques itérations fassent augmenter la fonction coût, que la forme soit en compétition avec la contrainte volumique, ou bien que le pas soit trop grand tout simplement (car pour le pas de descente h doit être assez petit). Ainsi, il faudra à chaque itération, calculer la direction de descente, déformer le maillage, calculer la solution sur ce maillage et voir si la fonction coût augmente. Si c'est le cas alors on reviens en arrière et on diminue le pas de descente h (on le divise par 2 par exemple) et on recommence à partir de l'étape de la déformation du maillage.

La question se pose de choisir quand stopper les augmentations possibles de la fonction coût : il se peut que, des variations de volume étant possibles, la fonction coût ait besoin d'augmenter pour arriver à un maillage de volume différent. Stoppé trop tôt, on ne convergera pas vers une forme possiblement optimale du volume que l'on se fixe. Ainsi il est intéressant de voir si en regardant les variations de volume, cela permettrait d'avoir un critère pour le début de l'interdiction.

Un critère possible serait de commencer l'interdiction lorsque l'on est proche à ϵ près du volume cible en variation relative depuis au moins m itérations consécutives (m modifiable). Cela implique donc de stocker m fois au maximum le volume de chaque étape et de vérifier si l'on est proche ou pas du volume cible et si le nombre de fois que l'on a été proche est m .

Mais comme on le voit, ce petit ajout est sujet à paramétrisation, ϵ dépendant de α_l . $m = 2$ ou 3 semble acceptable. De plus ce système est coûteux car on calcul la solution sur le maillage pour au final se rendre contre peut-être de la nécessité de revenir en arrière... En même temps il n'y a pas beaucoup de choix au regard du contrôle du pas a priori.

4.7.4 Irrégularités aux coins

Nous allons expliquer formellement pourquoi il y a des problèmes de régularité de la solution de l'équation d'état aux coins.

Prenons le cas d'une forme possédant un coin et que les deux bords touchant ce coin sont de nature différentes : un bord libre de contrainte Γ_N et un bord Dirichlet Γ_D . On pose $u = 0$ sur Γ_D et $\frac{\partial u}{\partial n} = 0$ sur Γ_N que l'on notera simplement Γ . On a donc le schéma de la figure 4.86.

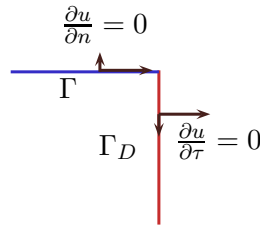


FIG. 4.86 – Irrégularité d'un coin

Prenons le problème simple de la conduction :

$$\begin{cases} -\Delta u = f & \text{dans } \Omega_0 \\ u = 0 & \text{sur } \Gamma_D \\ \frac{\partial u}{\partial n} = 0 & \text{sur } \Gamma \end{cases} \quad (4.67)$$

Occupons nous seulement de la régularité de la solution sur Ω et ses bords. On remarque sur la figure 4.86 que $\frac{\partial u}{\partial \tau} = 0$ sur Γ_D . Cela est dû au fait que $u = 0$ sur Γ_D . On ne contrôle donc pas à priori le saut qu'il y a des dérivées normales et tangentielles au coin. Moins formellement, si l'on prend f dans $L^2(\Omega)$ alors la solution u du problème est dans $H^2(\Omega)$. Or $H^2(\Omega)$ s'injecte compactement dans $W^{1,p}(\Omega)$, $\forall p < \infty$ qui lui même s'injecte compactement dans $C^{0,\alpha}(\overline{\Omega})$. Donc $u \in L^\infty(\Omega)$. Par contre ∇u est dans $H^1(\Omega)$ ce qui n'implique pas que ∇u soit borné. Il n'est que $L^p(\Omega)$, $\forall p < \infty$.

Donc on peut avoir un problème sur la régularité de u et surtout ∇u si l'on n'impose pas des conditions de régularités de la frontière. Et ce n'est pas le fait qu'il y ait un changement de type de bord qui joue grandement. Si l'on disposait de deux bords Dirichlet, les dérivées tangentielles seraient nulles sur chacun des côtés du coin, et il y aurait toujours un saut de la dérivée normale, induisant une explosion du gradient de la solution. Numériquement sur un maillage, la valeur du gradient peut varier drastiquement d'une maille à l'autre. Dans nos exemples de la partie 4.5, en général seules les mailles extrêmes (mailles des coins) ont été retirées du calcul.

Pour mettre en évidence numériquement l'explosion du gradient au coin, il suffit de calculer les conditions d'optimalités sans exclure les coins lors des précédents exemples de la section 4.5. Il s'en suit une condition d'optimalité élevée. D'où l'optimisation par suppression des coins avec généralement une fonction nommée *cut* qui définit la frontière à optimiser.

4.7.5 Eviter les oscillations

Pour éviter certaines oscillations du bord, on peut avoir recours à l'utilisation de deux maillages : un maillage fin Sh et grossier Th. Le grossier sera utilisé pour modifier la géométrie et le fin sera utilisé pour les calculs des solutions et du pas de descente. A chaque itération, lorsque le pas de descente est calculé, on utilise le maillage grossier Th pour la déformation, le contrôle avec *checkmovemesh* des mailles retournables et du contrôle de la direction de descente. Déformation faite, on raffine Th pour obtenir le nouveau Sh et pouvoir ainsi y relancer des calculs à l'étape suivante. Les lignes de code pour l'utilisation de maillages fins et grossiers étant éparpillées dans l'algorithme, il serait difficile d'en afficher seulement les parties : le lecteur ira regarder les codes commentés pour une meilleure compréhension.

4.7.6 Extension de la normale

La normale à un bord est normalement définie seulement sur ce même bord. Mais il est parfois utile d'étendre cette définition à tout le domaine pour pouvoir calculer par exemple ces variations sur le bord. En effet, avec FreeFem++ il n'est pas possible d'implémenter une méthode utilisant les dérivées partielles de la normale (utilisation de la divergence de la normale impossible : $\text{div}(n)$). Il faut pour cela, soit l'interpoler dans un espace plus adéquat, soit étendre la normale à tout le domaine. Pour ce faire il suffit de considérer les extensions harmoniques suivantes :

$$\begin{cases} -\Delta\phi_x = 0 & \text{dans } \Omega \\ \phi_x = n_x & \text{sur } \partial\Omega \end{cases} \quad (4.68)$$

$$\begin{cases} -\Delta\phi_y = 0 & \text{dans } \Omega \\ \phi_y = n_y & \text{sur } \partial\Omega \end{cases} \quad (4.69)$$

Il est alors possible ensuite de calculer les dérivées partielles de ϕ_x et ϕ_y . Ainsi le calcul de la divergence de la normale sur le bord se ramène au calcul de la divergence de $\phi = (\phi_x, \phi_y)$ sur le même bord avec FreeFem++.

Chapitre 5

Optimisation par lignes de niveaux

5.1 Cadre théorique

Le but sera d'ici d'obtenir une forme post-traitement, sous un maillage fixe en utilisant l'équation d'Hamilton-Jacobi. Les éléments théoriques qui suivent sont tirés de [1].

Soit $D \subset \mathbb{R}^n$ le domaine dans lequel toute les formes admissibles Ω sont incluses. On dispose donc de notre forme Ω et de vide autour $D \setminus \Omega$. On définit le bord de Ω par une fonction ψ dite de ligne de niveau (level-set) dans D par $\psi(x) = 0$ sur le bord de Ω , $\psi(x) < 0$ dans Ω et $\psi(x) > 0$ dans $D \setminus \overline{\Omega}$. La normale n à la forme Ω est donné par $\frac{\nabla \psi}{|\nabla \psi|}$.

On étend ensuite la solution u de l'équation d'état à tout D par l'approche "ersatz material" qui consiste à remplir le milieu $D \setminus \Omega$ d'un matériaux proche du vide et évitant la singularité de la matrice de rigidité. On définit un tenseur d'élasticité A^* qui se révèle être un mélange du tenseur A de Ω et de matériaux imitant le vide dans $D \setminus \Omega$:

$$A^*(x) = \rho(x)A$$

avec ρ valant 1 dans Ω et très petit dans $D \setminus \Omega$.

Prenons u solution du problème de l'élasticité linéarisée, sans force volumique. Au cours du processus d'optimisation, la forme va évoluer selon un temps fictif (qui correspondra au pas de descente de l'algorithme). Si la forme évolue donc dans le temps, alors l'évolution de la fonction de ligne de niveau est gouverné par une équation d'Hamilton-Jacobi. On a $\psi(t, x(t)) = 0$ pour tout $x(t) \in \partial\Omega(t)$. En différentiant par rapport à t il vient :

$$\frac{\partial \psi}{\partial t} + \dot{x}(t) \cdot \nabla \psi = \frac{\partial \psi}{\partial t} + V n \cdot \nabla \psi = 0$$

et de par la définition de la normale n on a :

$$\frac{\partial \psi}{\partial t} + V |\nabla \psi| = 0$$

L'équation de Hamilton-Jacobi est bien posée dans tout D si V est connu partout.

Numériquement nous allons utiliser un schéma up-wind : voir section suivante.

5.2 Algorithme

Nous avons déjà vu qu'une direction de descente adéquate pour la minimisation de la fonction coût est $\theta = -vn$ (voir section 4.4.1). La composante normale $-v$ est prise comme la vitesse d'advection dans l'équation d'Hamilton-Jacobi ce qui donne :

$$\frac{\partial \psi}{\partial t} - v |\nabla \psi| = 0 \quad (5.1)$$

Il est équivalent de transporter ψ de (5.1) que de bouger la frontière de Ω (qui est la ligne de niveau 0 de ψ) le long de la direction de descente du gradient $-J'(\Omega)$.

L'algorithme qui s'en suit est le suivant :

Algorithme d'optimisation par méthode level-set
<p>Mise en place des données initiales : déclarations</p> <p>Construction forme/maillage initial</p> <p>Ω_0 forme initiale</p> <p>Fonction level-set ψ_0 correspondante</p> <p>Initialisations</p> <p>Boucle Tant que $n < N_{max}$</p> <p>Calcul de la solution</p> <p>Calcul de la solution de l'équation adjointe</p> <p>Optimisations</p> <p>Résolution de H-J par schéma up-wind</p> <p>Déformation de la forme</p> <p>Fin Tant que</p> <p>Tracé des courbes</p> <p>Courbes de fonction coût, volume et condition d'optimalité</p> <p>Courbe de la solution sur le maillage final</p>

Chapitre 6

Conclusion

J'ai donc étudié théoriquement et numériquement l'optimisation de forme par une méthode dite "géométrique" (à variations de frontières) dans le cas de fonctions coûts compliance et énergie de déformation sur deux types de problèmes : la conduction et l'élasticité linéarisée. Le cas spécial de l'anneau a été traité pour décrire explicitement des solutions et un contre-exemple d'unicité de la forme optimale. Des cas tests ont ensuite été effectués pour étudier la bonne implémentation des codes.

Beaucoup de problèmes théoriques se dressent en optimisation de forme pour l'existence et l'unicité. On a vu au départ une façon d'assurer l'existence par régularité.

En ce qui concerne l'utilisation du α_l , tout dépend si l'on souhaite faire de la contrainte volumique un paramètre peu variable ou non : élevé, il freine l'optimisation, une moindre variation de volume étant moins facile à effectuer et petit, il donne plus de champ libre à l'optimisation. Pour contrer ce problème, une première idée est de se poser des problèmes qui ne requièrent pas qu'une tendance d'augmentation ou diminution de volume : par exemple le cas du M avec des trous, permet d'utiliser l'optimisation de forme à bon escient. Voir partie 4.5.6.

En ce qui concerne les forces suiveuses : le codage d'une optimisation d'un bord soumis à des forces surfaciques était soumis à la condition de trouver un moyen de coder correctement la partie supplémentaire correspondant aux forces suiveuses. C'est ce qui a été effectué en codant l'extension de la normale. Ainsi, les formes avec coins évoluent maintenant vers des formes sans coins si l'optimisation le permet (voir partie 4.5.3).

Les optimisations du code sont assez utiles mais demandent un effort de calcul supplémentaire. Il est aussi possible que l'on ait besoin de calculer une solution et de déformer le maillage pour, en fin de compte, revenir en arrière et cela fait perdre du temps. L'optimisation de la direction de descente fonctionne correctement et permet d'améliorer sensiblement la vitesse de convergence vers le minimum local. Je n'ai utilisé que très peu l'optimisation de code d'évitement des oscillations en utilisant un maillage fin et grossier car ces oscillations n'interviennent que dans certains cas. Le retournement de triangle est un vrai problème lorsque une forme fait converger deux bords l'un vers l'autre, il serait intéressant de voir si la "fusion" de bords (donc destruction de trous) est une possibilité envisageable en optimisation géométrique dans FreeFem++ tout d'abord. Enfin, le contrôle du pas est intéressant pour obtenir au final une courbe qui ne montre pas d'augmentation de la fonction coût, mais se révèle trop cher en calcul : à chaque étape, tout est calculé pour au final (peut-être) retourner à l'étape précédente.

Les atouts et inconvénients de l'optimisation géométrique sont de plusieurs ordres. Il en ressort que cette méthode ne peut pas changer la topologie de la forme par définition même : seul le bord de la forme varie. Au cours de l'optimisation, la forme aura tendance à converger vers des optima locaux, d'autant plus qu'elle ne peut changer sa topologie. Ses optima locaux sont très dépendants du paramétrage sur la contrainte volumique (à travers α_l).

L'optimisation par ligne de niveaux (méthode level-set), aussi abordé durant ce stage permet, quand à elle, de supprimer les trous, donc de changer de topologie, par contre, s'établissant sur un maillage fixe donné au départ, elle permet une précision de la forme en direct relation avec la densité de maille du

support de départ. Le gradient topologique est un outil non abordé ici qui permet de trouver l'endroit où il est préférable de créer un minuscule trou : cela permet de changer de topologie en créant des trous et plus seulement en les détruisant.

Références

- [1] G. Allaire, F. Jouve, A-M. Toader, Structural optimization using sensitivity analysis and a level-set method, *J. Comp. Phys.* 194 (2004) 363-393.
- [2] G. Allaire, Conception optimale de structures, Cours de troisième année MAP562, Ecole Polytechnique (2004).
- [3] F. Murat, J. Simon, Sur le contrôle par un domaine géométrique, Internal report No 76015, Laboratoire d'Analyse Numérique de l'université Paris 6 (1976).
- [4] F. Murat, J. Simon, Etude de problèmes d'optimal design, *Lecture Notes in Computer Science* 41, pp.54-62, Springer Verlag, Berlin (1976).
- [5] O. Pironneau, Optimal shape design for elliptic systems, Springer-Verlag, New York (1984).
- [6] J. Sokolowski, J.-P. Zolesio, Introduction to shape optimization. Shape sensitivity analysis, Springer Series in Computational Mathematics, 16, Springer, Berlin (1992).
- [7] G. Allaire, O. Pantz, Structural Optimization with FreeFem++, R.I. 586, Ecole Polytechnique, (2005)
- [8] Page personnelle de G. Allaire : www.cmap.polytechnique.fr/~allaire/freemem.html

Annexe

Petit mot sur FreeFem++

Description

FreeFem++ a été utilisé pour écrire la totalité des codes de ce stage. Il permet de résoudre des EDP avec la méthode des éléments finis. Notons qu'il existe maintenant un FreeFem3d permettant de résoudre des EDP en 3 dimensions.

Mots clés utilisés

`int`, `real`, `string`, `bool`, `func`, `fespace`, `mesh`, `border` : pour les définitions.
`macro` pour l'utilisation de macros.
`problem` : pour la définition d'un problème d'EDPs.
`cout << ... << endl` : pour l'affichage de données dans le terminal.
`plot` : pour le tracé des courbes.
`adaptmesh` : pour l'adaptation du maillage
`movemesh` : pour déformer le maillage
`checkmovemesh` : pour vérifier s'il y a des triangles retournés après déformation du maillage.
`exec` : pour utiliser des commandes en mode terminal à partir du fichier .edp.
Interfacage avec gnuplot avec la gestion des fichiers .gp : voir section suivante.

Lien Gnuplot

Voici un exemple d'interfacage avec gnuplot. On insère tout d'abord toutes les données recueillies dans un fichier selon un format spécifique simple :

```
//GNU PLOT////////////////////////////////////
{
ofstream gnu("CasTestLComplianceFcoutOptimise.gp");

//gnuplot
for (int i=0;i<niter;i++)
{
gnu << xx[i] << " " << yy[i] << endl;
}

}
```

et on utilise ensuite la méthode *exec* :

```
exec("echo 'set xlabel \"Iterations\" 0,0\
set ylabel \"Objectif\" 0,0\
plot \"CasTestLComplianceFcoutOptimise.gp\" w l\
set term postscript\
set output \"CasTestLComplianceFcoutOptimise.eps\"\
replot\
pause 2\
quit' | gnuplot");
```

On a donc au final création des fichiers **CasTestLComplianceFcoutOptimise.gp** contenant les informations du graphe et **CasTestLComplianceFcoutOptimise.eps** image du graphe.

Codes

Voici deux des codes implémentés.

Code CasTestCarreTroueComplianceOptimise.edp : code utilisant les optimisations de 4.7, une fonction coût compliance et utilisé dans la section 4.2.

```

/*****
Optimisation dun carre troue ou dun cercle trou dun carre
Conditions aux bords : type Dirichlet et Neumann
Fonction cout : la compliance.(probleme auto-adjoint)
Eq detat : -laplacien(u)=f
Condition D : u=0 sur bord Dirichlet
Condition neumann : d(u)/dn=g sur bord neumann
Condition libre : d(u)/dn=0 sur bord libre
*****/

// Donnees :
int niter=30; // Nombre d'it\{e}rations
real fcout; // Valeure de la fonction cout
real volume0; // Volume initial
real volume; // Volume de la forme d\{e}form\{e}
real lagrange; // Multiplicateur de lagrange pour la contrainte volumique
real lagrangestep=2; // Plus il est eleve, meilleur est la precision sur le volume contrainte
real perimeter; // Perimetre de la partie variant
string legende; // L\{e}gende pour les sorties graphiques
real pi=4*atan(1) ; // Nombre pi
func f=0.; // Force volumique (non utilise ici)
func g=1.; // Force surfacique
real step=0.1,pas; // Pas de descente
real multVol=1.;//10./11.; // Multiplicateur réglant le volume cible
real[int] xx(niter),yy(niter),vol(niter),condopt(niter); // Tableaux affichage gnuplot
bool arriere=false; // pour loptimisation
real produit, norme; // pour loptimisation

// Fonction d\{e}finissant la zone \{a} optimiser
// (=1 sur la fronti\{e}re optimisable, =0 sinon)
//func cut=((x-0.5)^2+(y-0.5)^2<0.49^2); // cut pour trou interieur dirichlet
func cut=((x-0.5)^2+(y-0.5)^2>0.11^2); // cut pour trou exterieur dirichlet

// D\{e}finition du domaine
// 1:Condition de Dirichlet
// 2:Condition Libre
// 3:Condition de Neumann non nulle
mesh Sh;

border b1(t=0,2){x=t;y=0;label=2;}; // bord inferieur
border b2(t=0,1){x=2;y=t;label=1;}; // bord droit
border b3(t=2,0){x=t;y=1;label=3;}; // bord superieur
border b4(t=1,0){x=0;y=t;label=1;}; // bord gauche
border rond(t=0,pi){x=1+1*cos(t);y=1+1*sin(t);label=3;};// trou
border sup1(t=1,0){x=t;y=0.2*t+1;label=3;}; // bord superieur
border sup2(t=2,1){x=t;y=-0.2*t+1.4;label=3;}; // bord superieur

```

```

// Pour carre troue
real ray1=0.1; // rayon du trou
border trou(t=0,2*pi){x=0.5+ray1*cos(t);y=0.5+ray1*sin(t);label=1;}; // trou interieur
border b1n(t=0,1){x=t;y=0;label=3;}; // bord inferieur
border b2n(t=0,1){x=1;y=t;label=3;}; // bord droit
border b3n(t=1,0){x=t;y=1;label=3;}; // bord superieur
border b4n(t=1,0){x=0;y=t;label=3;}; // bord gauche

real ray2=0.2; // Rayon du trou 2
border trou2(t=0,2*pi){x=0.5+ray2*cos(t);y=0.5+ray2*sin(t);label=3;}; // trou exterieur

border ar1n(t=pi/2.,pi){x=0.1+0.1*cos(t);y=0.9+0.1*sin(t);label=3;}; // coin arrondi 1
border ar2n(t=pi,3*pi/2.){x=0.1+0.1*cos(t);y=0.1+0.1*sin(t);label=3;}; // coin arrondi 2
border ar3n(t=3*pi/2.,2*pi){x=0.9+0.1*cos(t);y=0.1+0.1*sin(t);label=3;}; // coin arrondi 3
border ar4n(t=0,pi/2.){x=0.9+0.1*cos(t);y=0.9+0.1*sin(t);label=3;}; // coin arrondi 4
border b1m(t=0.1,0.9){x=t;y=0;label=3;}; // bord inferieur
border b2m(t=0.1,0.9){x=1;y=t;label=3;}; // bord droit
border b3m(t=0.9,0.1){x=t;y=1;label=3;}; // bord superieur
border b4m(t=0.9,0.1){x=0;y=t;label=3;}; // bord gauche

// Construction du maillage
// carre de base
//Sh= buildmesh (b1(30)+b2(20)+b3(30)+b4(20));

// carre avec bord rond superieur
//Sh= buildmesh (b1(30)+b2(20)+b4(20)+rond(30));

// carre avec partie superieur en toit
//Sh= buildmesh (b1(30)+b2(20)+b4(20)+sup1(10)+sup2(10));

// carre de base neumann!=0 avec trou interieur dirichlet
//Sh= buildmesh (b1n(30)+b2n(20)+b3n(30)+b4n(20)+trou(-40));

// carre de base neumann!=0 avec cercle exterieur dirichlet
//Sh= buildmesh (b1n(-30)+b2n(-20)+b3n(-30)+b4n(-20)+trou(20));

// cercle de base neumann!=0 avec trou interieur dirichlet
Sh= buildmesh (trou2(100)+trou(-40));

// cercle de base neumann!=0 avec trou exterieur dirichlet : changer les rayons et la fonction cut
//Sh= buildmesh (trou2(-100)+trou(200));

// carre bords arrondi de base neumann!=0 avec trou interieur dirichlet
//Sh= buildmesh (b1m(30)+b2m(20)+b3m(30)+b4m(20)
+trou(-50)+ar1n(20)+ar2n(20)+ar3n(20)+ar4n(20));

// Definition des espaces delements finis
fespace Vh1(Sh,P1);
fespace Vh2(Sh,[P2,P2]);
fespace VH1D2(Sh,P2);

VH1D2 u,v,gh=g,nxx,nny;
Vh2 [theta1,theta2],[d1,d2],[d1p,d2p];
Vh1 integrale,integrande2;

// Mise en place dune macro :
// (on definira les produit scalaire sans lintegrale qui, elle, sera utilisee dans la FV)
// produit scalaire H1 (integrale de grad(u).grad(phi)+u.phi) :

```

```

macro psH1(u,v,w,s)
dx(u)*dx(w)+dy(u)*dy(w)+dx(v)*dx(s)+dy(v)*dy(s)+u*w+v*s

// Probleme detat
problem etat(u,v) =
    int2d(Sh)(
        dx(u)*dx(v)+dy(u)*dy(v)
    )
    -int2d(Sh)(f*v)
    -int1d(Sh,3)(g*v)
    +on(1,u=0)
;

// Probleme d'extension
problem extension([d1,d2],[theta1,theta2]) =
    int2d(Sh)(
        psH1(d1,d2,theta1,theta2)
    )
    -int1d(Sh,2,3)(
        (
            (theta1*N.x+theta2*N.y)*(2*f*u-dx(u)^2-dy(u)^2+lagrange)
        )*cut
    )
    -int1d(Sh,3)(
// Utilisation de l harmonisation de la normale
2.*(theta1*N.x+theta2*N.y)*(g+u*(dx(nnx)+dy(nny)))
    )
    )
    +on(1,2,d1=0,d2=0)
;

// Probleme harmonisation NX
problem harmonisationX(nnx,v) =
    int2d(Sh)(
        dx(nnx)*dx(v)+dy(nnx)*dy(v)
    )
    -int1d(Sh)((dx(nnx)*N.x+dy(nnx)*N.y)*v)
    +on(1,2,3,nnx=N.x)
;

// Probleme harmonisation NY
problem harmonisationY(nny,v) =
    int2d(Sh)(
        dx(nny)*dx(v)+dy(nny)*dy(v)
    )
    -int1d(Sh)((dx(nny)*N.x+dy(nny)*N.y)*v)
    +on(1,2,3,nny=N.y)
;

// Calcul du volume initial : qui sera le volume cible
volume0=int1d(Sh)(x*N.x+y*N.y)/2;
volume0=multVol*volume0;
cout <<"Volume initial = "<<volume0 <<endl;

// Initialisation
int iter;

// Resolution du syst\{e\me d etat

```

```

etat;

// Calcul du perimetre
perimeter=int1d(Sh,3)(1.);
cout <<"perimetre depart = "<< perimeter <<endl;

// Init du calcul de la condition doptimalite pour affichage gnuplot
condopt[0]=int1d(Sh,3)(abs(-dx(u)*dx(u)-dy(u)*dy(u)))/perimeter;

// Calcul de la fonction cout au depart
real fcout0=int2d(Sh)(f*u)+int1d(Sh,3)(g*u);
cout <<"Valeure fonction cout depart = "<< fcout0 <<endl;

// Ini pour affichage gnuplot de la fonction cout et du volume
xx[0]=0;
yy[0]=fcout0;
vol[0]=volume0;

// Init pas pour control du pas
pas=step;

// Calcul de lintegrande du gradient de la fonction cout :
integrande=(-dx(u)*dx(u)-dy(u)*dy(u))*cut;

// Calcul du multiplicateur de lagrange de depart (contrainte sur le volume)
lagrange=-int1d(Sh,2)(integrande)/perimeter;
cout <<"lagrange depart = "<< lagrange <<endl;

// Impression de la forme initiale
legende="Forme initiale, Fonction Cout "+fcout0+", Volume "+volume0;
plot(Sh,u,fill=1,wait=0,value=1,cmm=legende,ps="CasTestCarreTroueComplianceOptimiseFormeIni.eps");

// Adaptation du maillage
Sh=adaptmesh (Sh,u);

// Boucle d'optimisation
for (iter=1;iter< niter;iter=iter+1)
{
cout <<"Iteration " <<iter <<" -----" <<endl;

// Resolution des systemes
etat;

// Calcul du perimetre
perimeter=int1d(Sh,3)(1.);

// Calcul de la fonction cout
fcout=int2d(Sh)(f*u)+int1d(Sh,3)(g*u);
cout<<"fcout = "<<fcout<<endl;
xx[iter]=iter;
yy[iter]=fcout;

// On calcule le volume
volume=int1d(Sh)(x*N.x+y*N.y)/2;
cout <<"Volume = "<<volume <<endl;
vol[iter]=volume;

// Calcul de lintegrande du gradient de la fonction cout :

```

```

integrande=(-dx(u)*dx(u)-dy(u)*dy(u))*cut;
integrande2=(2*(g+u*(dx(nnx)+dy(nny)))-dx(u)*dx(u)-dy(u)*dy(u))*cut;

// Mise \{a} jour du multiplicateur de Lagrange associ\{e} au volume
lagrange=0.5*lagrange+0.5*(int1d(Sh,2)(integrande)+int1d(Sh,3)(integrande2))/perimeter
+lagrangestep*(volume-volume0)/volume0;
cout << "Lagrange = " << lagrange << endl ;

// Extension du gradient de forme \{a} l'int\{e}rieur du domaine
harmonisationX;
harmonisationY;
extension;

Sh = adaptmesh (Sh,u);

// On calcule la norme (au carr\{e}) du gradient de forme
norme=int2d(Sh)(psH1(d1,d2,d1,d2));
//grad<<sqrt(norme)<<endl;
// On compare le gradient de forme avec le gradient pr\{e}c\{e}dent
produit=int2d(Sh)(psH1(d1,d2,d1p,d2p));

//////////////////////////
// Mise \{a} jour du pas de descente //
//////////////////////////
// Si le gradient de la fonction cout est dans la m\{e}me direction que le gradient
// de l'it\{e}ration pr\{e}c\{e}dente (produit>0), on augmente le pas de descente.
// Si le gradient de la fonction cout n'est pas dans la m\{e}me direction que le
// gradient de l'it\{e}ration pr\{e}c\{e}dente (produit<0), on fait marche arri\{e}re et
// on diminue le pas de descente (on \{e}vite ainsi les oscillations).
if ((produit<0)&(!arriere))
    {pas=pas*norme/(norme-produit)/4.;
    cout<<"***** Pas de descente trop important: Marche arriere !!! *****"<<endl;
    arriere=true;
}
else
    {arriere=false;}
;
if ((produit>0)&(produit<(norme/2.)))
    {pas=pas*2*norme/(2*norme-produit);
    cout<<"On augmente le pas de descente"<<endl;}
;
if (produit>(norme/2.))
    {pas=pas*4./3.;
    cout<<"On augmente le pas de descente"<<endl;}
;

// On va verifier avant de bouger le maillage que des triangles ne seront pas retournes
if (!arriere)
{
while (checkmovemesh (Sh,[x,y])/10. > checkmovemesh(Sh,[x-pas*d1,y-pas*d2]) )
    {
    cout << "Pas de triangle retourne"<< endl;
    pas= pas/2.;
    }
}

// Mouvement du maillage

```

```

Sh = movemesh (Sh,[x-pas*d1,y-pas*d2]);

cout<<"Pas de descente effectif = "<<pas<<endl;

// On interpole le gradient sur le maillage d\{'e}form\{'e}
[d1p,d2p]=[0,0];
d1p[]=d1[];d2p[]=d2[];

};

// Impression
legende="Forme finale, Iteration "+iter+", fcout "+fcout+", Volume "+volume;
plot(Sh,u,fill=1,value=1,cmm=legende); // Affichage du déplacement u

// Augmentation du lagrangestep si l iteration devient trop grande
/*if (iter>=50)
{
lagrangestep=lagrangestep+5;
}*/

//harmonisationX;
//harmonisationY;

// Condition doptimalite a cette iteration
condopt[iter]=int1d(Sh,2)(cut*abs(-dx(u)^2-dy(u)^2+lagrange))/perimeter
+int1d(Sh,3)(cut*abs(-dx(u)^2-dy(u)^2+2.*g+2.*u*(dx(nnx)+dy(nny))+lagrange))/perimeter;
}

// Affichage derniere condition doptimalite
cout << "Condition doptimalite finale : "<< condopt[niter-1] << endl;

/*****Trace des courbes*****/

//GNUPLOT////////////////////////////////////
{
ofstream gnu("CasTestCarreTroueComplianceOptimiseFcout.gp");

//gnuplot
for (int i=0;i<niter;i++)
{
gnu << xx[i] << " " << yy[i] << endl;
}

}

//GNUPLOT////////////////////////////////////
{
ofstream gnu("CasTestCarreTroueComplianceOptimiseVolume.gp");

//gnuplot
for (int i=0;i<niter;i++)
{
gnu << xx[i] << " " << vol[i] << endl;
}

}

//GNUPLOT////////////////////////////////////

```

```

{
ofstream gnu("CasTestCarreTroueComplianceOptimisecondopt.gp");

//gnuplot
for (int i=0;i<niter;i++)
{
gnu << xx[i] << " " << condopt[i] << endl;
}
}

exec("echo 'set xlabel \"Iterations\" 0,0\
set ylabel \"Objectif\" 0,0\
plot \"CasTestCarreTroueComplianceOptimiseFcout.gp\" w l\
set term postscript\
set output \"CasTestCarreTroueComplianceOptimiseFcout.eps\"\\
replot\
pause 2\
quit' | gnuplot");

exec("echo 'set xlabel \"Iterations\" 0,0\
set ylabel \"Volume\" 0,0\
plot \"CasTestCarreTroueComplianceOptimiseVolume.gp\" w l\
set term postscript\
set output \"CasTestCarreTroueComplianceOptimiseVolume.eps\"\\
replot\
pause 2\
quit' | gnuplot");

exec("echo 'set xlabel \"Iterations\" 0,0\
set ylabel \"condition opt=?=0\" 0,0\
plot \"CasTestCarreTroueComplianceOptimisecondopt.gp\" w l\
set term postscript\
set output \"CasTestCarreTroueComplianceOptimisecondopt.eps\"\\
replot\
pause 2\
quit' | gnuplot");

// On imprime la forme finale
legende="Forme finale, Iteration "+iter+", fcout "+fcout+", Volume "+volume;
plot(Sh,u,fill=1,value=1,cmm=legende,ps="CasTestCarreTroueComplianceOptimise.eps");

```

Code CasTestMGradient.edp : code utilisant la fonction coût d'énergie de déformation utilisé dans la section [4.5.6](#).

```

/*****
Optimisation dun M
Bords : type Dirichlet et Neumann
Fonction cout : gradient avec poid k(x)
Eq detat : -laplacien(u)=f
Condition D : u=0 sur bord Dirichlet
Condition neumann : d(u)/dn=g sur bord neumann
Condition libre : d(u)/dn=0 sur bord libre
*****/

// Donnees :
int niter=300; // Nombre d'it\{'e}rations

```



```

real fcout; // Valeure de la fonction cout
real volume0; // Volume initial
real volume; // Volume de la forme d\{'e}form\{'e}e
real lagrange;
real lagrangestep=200; // plus il est eleve, meilleur est la precision sur le volume contrainte
real perimeter;
string legende; // L\{'e}gende pour les sorties graphiques
real pi=4*atan(1) ;
func f=0.;// force volumique
func g=1.;
real step=0.005,pas;
real multVol=1.; // multiplicateur volumique de la contrainte de volume

real[int] xx(niter),yy(niter),vol(niter);

// Fonction d\{'e}finissant la zone \{'a} optimiser
// (=1 sur la fronti\{'e}re optimisable, =0 sinon)

func cut=(abs(x)<1.95)*(y<0.95)*(y>-1.85);
//func cut=1.;

//func kcut=1.;
func kcut=(x<0)+0.2*(x>0);

// D\{'e}finition du domaine
// 1:Condition de Dirichlet
// 2:Condition Libre (parties optimisables)
// 3:Condition de Neumann non nulle
// 4:Condition de Neumann homogene
mesh Sh;

//////////
// Domaine M conduction
real rayon=0.7; // rayon pour les coins "arrondi"

border a(t=-2,1) { x=-2; y=t;label=4; };
border b(t=-2,2) { x=t; y=1;label=3; };
border c(t=1,-2) { x=2; y=t;label=4; };
border b1(t=2,1) { x=t; y=-2;label=1; };
border b2(t=-1,-2) { x=t; y=-2;label=1; };
border iR1(t=-2,-rayon) { x=1; y=t;label=2; };
border iR2(t=1-rayon,-1+rayon) { x=t; y=0;label=2; };
border iR3(t=-rayon,-2) { x=-1; y=t;label=2; };
border i1(t=-2,0) { x=1; y=t;label=2; };
border i2(t=1,-1) { x=t; y=0;label=2; };
border i3(t=0,-2) { x=-1; y=t;label=2; };
border rond1(t=0,pi/2) {x=1-rayon+rayon*cos(t);y=-rayon+rayon*sin(t);label=2;};
border rond2(t=pi/2,pi) {x=-1+rayon+rayon*cos(t);y=-rayon+rayon*sin(t);label=2;};

real ray=0.2 ;
border z1(t=0,2*pi) {x=-1.5+ray*cos(t);y=0.5+ray*sin(t);label=2;};
border z2(t=0,2*pi) {x=1.5+ray*cos(t);y=0.5+ray*sin(t);label=2;};

// avec coin et trous
//Sh= buildmesh (a(-20)+b(-20)+c(-20)+b1(-5)+b2(-5)+i1(-20)+i2(-20)+i3(-20)+z1(-15)+z2(-15));

//avec coin arrondi et trous
Sh= buildmesh (a(-20)+b(-20)+c(-20)+b1(-5)+b2(-5)+iR1(-20)+iR2(-20)+iR3(-20)

```

```

+rondd1(-5*(1+rayon))+rondd2(-5*(1+rayon))+z1(-15)+z2(-15));

// avec coin et sans trous
//Sh= buildmesh (a(-20)+b(-20)+c(-20)+b1(-5)+b2(-5)+i1(-20)+i2(-20)+i3(-20));

// Fin de construction du domaine
////////////////////////////////////

// Definition des espaces de elements finis
espace Vh1(Sh,P1);
espace Vh2(Sh,[P2,P2]);
espace VH1D2(Sh,P2);

VH1D2 u,p,v,gh=g,kcut=kcut;
Vh2 [theta1,theta2],[d1,d2];
Vh1 integrale;

// Mise en place d'une macro :
// (on definira le produit scalaire sans l'integrale qui, elle, sera utilisee dans la FV)

// produit scalaire H1 (integrale de grad(u).grad(phi)+u.phi) :
macro psH1(u,v,w,s)
dx(u)*dx(w)+dy(u)*dy(w)+dx(v)*dx(s)+dy(v)*dy(s)+u*w+v*s

// Probleme d'etat
problem etat(u,v) =
    int2d(Sh)(
        dx(u)*dx(v)+dy(u)*dy(v)
    )
    -int2d(Sh)(f*v)
    -int1d(Sh,3)(g*v)
    +on(1,u=0)
;

// Probleme adjoint :
problem adjoint(p,v) =
    int2d(Sh)(
        dx(p)*dx(v)+dy(p)*dy(v)
    )
    +int2d(Sh)(kcut*(dx(u)*dx(v)+dy(u)*dy(v)))

    +on(1,p=0)
;

// Probleme d'extension
problem extension([d1,d2],[theta1,theta2]) =
    int2d(Sh)(
        psH1(d1,d2,theta1,theta2)
    )
    -int1d(Sh,2,3)(
        (
            (theta1*N.x+theta2*N.y)*(0.5*kcut*(dx(u)^2+dy(u)^2)-p*f+dx(u)*dx(p)+dy(u)*dy(p)+lagrange)
        )
    )
    +on(1,3,4,d1=0,d2=0)
;

```

```

// Calcul du volume initial : qui sera le volume cible
volume0=int1d(Sh)(x*N.x+y*N.y)/2;
volume0=multVol*volume0;
cout <<"Volume initial = "<<volume0 <<endl;

vol[0]=volume0;
// Initialisation
int iter;

// Resolution du syst\{e}me d etat
etat;
adjoint;

// Calcul du perimetre
perimeter=int1d(Sh,3)(1.);
cout <<"perimetre depart = "<< perimeter <<endl;

// Calcul de la fonction cout au depart
real fcout0=int2d(Sh)(kcut*(dx(u)^2+dy(u)^2));
cout <<"Valeure fonction cout depart = "<< fcout0 <<endl;

xx[0]=0;
yy[0]=fcout0;

// Calcul de lintegrande du gradient de la fonction cout :
integrande=(kcut*(dx(u)^2+dy(u)^2)-p*f+dx(u)*dx(p)+dy(u)*dy(p))*cut;

// Calcul du multiplicateur de lagrange de depart (contrainte sur le volume)
lagrange=-int1d(Sh,2)(integrande)/perimeter;
cout <<"lagrange depart = "<< lagrange <<endl;

// Impression de la forme initiale
legende="Forme initiale, Fonction Cout "+fcout0+", Volume "+volume0;
plot(Sh,u,fill=1,wait=0,value=1,cmm=legende);

// Adaptation du maillage
Sh=adaptmesh (Sh,u);

// Boucle d'optimisation

for (iter=1;iter< niter;iter=iter+1)
{
cout <<"Iteration " <<iter <<" -----" <<endl;

// Resolution des systemes
etat;
adjoint;

// Calcul du perimetre
perimeter=int1d(Sh,3)(1.);

// Calcul de la fonction cout
fcout=int2d(Sh)(kcut*(dx(u)^2+dy(u)^2));
cout<<"fcout = "<<fcout<<endl;

xx[iter]=iter;
yy[iter]=fcout;

```

```

//On calcule le volume
volume=int1d(Sh)(x*N.x+y*N.y)/2;
cout <<"Volume = "<<volume <<endl;
vol[iter]=volume;

// Calcul de l'integrande du gradient de la fonction cout :
integrande=(kcut*(dx(u)^2+dy(u)^2)-p*f+dx(u)*dx(p)+dy(u)*dy(p))*cut;

// Mise \'{a} jour du multiplicateur de Lagrange associ\'{e} au volume
lagrange=0.5*lagrange+0.5*int1d(Sh,2)(integrande)/perimeter
+lagrangestep*(volume-volume0)/volume0;
cout << "Lagrange = " << lagrange << endl ;

// Extension du gradient de forme \'{a} l'int\'{e}rieur du domaine
extension;

Sh = adaptmesh (Sh,u);

// On va verifier avant de bouger le maillage que des triangles ne seront pas retournes
pas=step;
while(checkmovemesh(Sh,[x,y])/10.>checkmovemesh(Sh,[x-pas*d1,y-pas*d2]))
{
pas=pas/2;
}

// On modifie le maillage avec la direction d=[d1,d2]
Sh = movemesh (Sh,[x-pas*d1,y-pas*d2]);

// Impression
legende="Forme finale, Iteration "+iter+", fcout "+fcout+", Volume "+volume;
plot(Sh,u,fill=1,value=1,cmm=legende); // Affichage du déplacement u
}

/*****Trace des courbes*****/

//GNU PLOT////////////////////////////////////
{
ofstream gnu("CasTestMGradientFcout.gp");

//gnuplot
for (int i=0;i<niter;i++)
{
gnu << xx[i] << " " << yy[i] << endl;
}

}

//GNU PLOT////////////////////////////////////
{
ofstream gnu("CasTestMGradientVolume.gp");

//gnuplot
for (int i=0;i<niter;i++)
{
gnu << xx[i] << " " << vol[i] << endl;
}

}

```

```

exec("echo 'set xlabel \"Iterations\" 0,0\
set ylabel \"Objectif\" 0,0\
plot \"CasTestMGradientFcout.gp\" w l\
set term postscript\
set output \"CasTestMGradientFcout.eps\"\\
replot\
pause 2\
quit' | gnuplot");

exec("echo 'set xlabel \"Iterations\" 0,0\
set ylabel \"Volume\" 0,0\
plot \"CasTestMGradientVolume.gp\" w l\
set term postscript\
set output \"CasTestMGradientVolume.eps\"\\
replot\
pause 2\
quit' | gnuplot");

// On imprime la forme finale
legende="Forme finale, Iteration "+iter+", fcout "+fcout+", Volume "+volume;
plot(Sh,u,fill=1,value=1,cmm=legende,ps="CasTestMGradient.eps");
plot(Sh,kcuth,fill=1,value=1,cmm=legende,ps="CasTestMGradientKcut.eps");

```

Table des figures

4.1	Schéma d'une membrane avec trois bords différents : dirichlet, Neumann et libre	11
4.2	Schéma de la conduction sur un carré troué	16
4.3	Tracé de la fonction coût avec $r_1 = 2$ sur $[2, 3]$ en fonction de r_2	17
4.4	Tracé de la fonction coût avec $r_2 \in [0, 3]$ et $r_1 = 2$	18
4.5	Tracé de la fonction coût sur $[0, 3]$ avec $r_1 = 1.5$	21
4.6	Schéma de la conduction avec force volumique seulement	28
4.7	Solution initiale sur maillage initial, Volume=2	28
4.8	Solutions et formes à l'itération 16 (gauche) et 32 (droite)	29
4.9	Valeur de la fonction coût (gauche) et volume (droite) sur 50 itérations	29
4.10	Valeur de la fonction coût (gauche) et volume (droite) sur 300 itérations	30
4.11	Valeur de la fonction coût (gauche) et volume (droite) sur 50 itérations	30
4.12	Solution finale, 100 itérations, Volume=0.35	31
4.13	Valeur de la fonction coût (gauche) et volume (droite) sur 100 itérations	31
4.14	Valeur de la fonction coût (gauche) et volume (droite) sur 200 itérations, $\alpha_l^{n+1} = \alpha_l^n + 1$	32
4.15	C_{opt} , 200 itérations	33
4.16	Valeur de la fonction coût (gauche) et volume (droite) sur 200 itérations, α_l petit	33
4.17	C_{opt} (gauche) et son logarithme (droite) sur 200 itérations, α_l petit, pas=0.1	34
4.18	Valeur de la fonction coût (gauche) et volume (droite) sur 300 itérations, α_l petit	34
4.19	Solution finale $k = \mathbf{1}_{0 < x < 1} + 0.2 * \mathbf{1}_{1 < x < 2}$, 300 itérations, Volume=1.8010	35
4.20	Valeur de la fonction coût (gauche) et volume (droite) sur 300 itérations, α_l petit, 1 trou	35
4.21	Solution finale 1 trou, 300 itérations, Volume=1.6525	36
4.22	Fonction coût (gauche) et volume (droite) sur 300 itérations, α_l petit, 2 trous	36
4.23	Solution finale 2 trous, 300 itérations, Volume=1.6238	36
4.24	Schéma de la conduction avec variation du bord Neumann	37
4.25	Solution initiale sur maillage initial, Volume=2,2	37
4.26	Valeur de la fonction coût (gauche) et volume (droite) sur 150 itérations, α_l petit	38
4.27	Solution finale, Volume=1,9585	38
4.28	Valeur de la fonction coût (gauche) et volume (droite) sur 150 itérations, α_l grand	39
4.29	Solution finale, Volume=1,9783	39
4.30	Valeur de la fonction coût (gauche) et volume (droite) sur 120 itérations, α_l petit	39
4.31	Valeur de la fonction coût (gauche) et volume (droite) sur 150 itérations, α_l petit	40
4.32	Solutions à l'itération 80 (gauche) et 150 (droite)	40
4.33	Fonction coût (gauche) et volume (droite) sur 50 itérations	41
4.34	Tracé de la condition d'optimalité sur 50 itérations	41
4.35	Solution au bout des 33 itérations et des 50 itérations	42
4.36	Fonction coût (gauche) et volume (droite) sur 100 itérations	42
4.37	Tracé de la condition d'optimalité sur 100 itérations	43
4.38	Solution au bout des 100 itérations	43
4.39	Fonction coût (gauche) et volume (droite) sur 100 itérations	43
4.40	Forme initiale (gauche) et forme finale (droite) aux bout des 100 itérations	44
4.41	Forme initiale (gauche) et forme finale (droite) aux bout des 100 itérations	45
4.42	Fonction coût (gauche) et volume (droite) sur 100 itérations	45
4.43	Forme finale (droite) aux bout des 40 itérations	46
4.44	Fonction coût (gauche) et volume (droite) sur 40 itérations	46
4.45	Tracé de la condition d'optimalité sur 40 itérations	46
4.46	Forme finale (droite) aux bout des 40 itérations	47
4.47	Fonction coût (gauche) et volume (droite) sur 40 itérations	47

4.48	Tracé de la condition d'optimalité sur 40 itérations	47
4.49	Fonction <i>cut</i> (gauche) et forme/solution initiale (droite)	48
4.50	Valeur de la fonction coût (gauche) et volume (droite) sur 25 itérations, α_l petit	48
4.51	Solution finale avant crash, Volume=12,0928	49
4.52	Fonction coût (gauche) et volume (droite) sur 180 itérations, α_l petit, coin arrondi	49
4.53	Solution finale avant crash, Volume=12,4447	49
4.54	Fonction coût (gauche) et volume (droite) sur 180 itérations, α_l grand, coin arrondi	50
4.55	Solution finale , Volume=12,1161	50
4.56	Forme à la 100ème itération (gauche) et à la 400ème (droite)	50
4.57	Fonction coût (gauche) et volume (droite) sur 400 itérations, α_l grand, coin arrondi	51
4.58	Fonction coût (gauche) et volume (droite) sur 100 itérations, $\alpha_l = 50$, coin arrondi	51
4.59	Forme finale, 100 itérations (gauche) et $k = \mathbf{1}_{(x^2+y^2)<0.25^2} + 0.2 * \mathbf{1}_{(x^2+y^2)>0.25^2}$ (droite)	52
4.60	Schéma de la conduction sur le M (troué ou non)	52
4.61	Fonction coût (gauche) et volume (droite) sur 9 itérations, α_l petit	53
4.62	Forme itération 9	53
4.63	Fonction coût (gauche) et volume (droite) sur 100 itérations, α_l petit	53
4.64	Forme itération 100	54
4.65	C_{opt} sur 100 itérations	54
4.66	Fonction coût (gauche) et volume (droite) sur 200 itérations, α_l grand	54
4.67	Forme itération 200	55
4.68	C_{opt} sur 200 itérations	55
4.69	Fonction coût (gauche) et volume (droite) sur 100 itérations, α_l petit, maillage troué	55
4.70	Forme itération 100, maillage troué	56
4.71	C_{opt} sur 100 itérations, maillage troué	56
4.72	Fonction coût (gauche) et volume (droite) sur 100 itérations, α_l grand, maillage troué	56
4.73	Forme itération 100, maillage troué	57
4.74	C_{opt} sur 100 itérations, maillage troué	57
4.75	Fonction coût (gauche) et volume (droite) sur 300 itérations, $\alpha_l = 200$, maillage troué	58
4.76	Forme finale (gauche) et fonction $k = \mathbf{1}_{x<0} + 0.2 * \mathbf{1}_{x>0}$ (droite)	58
4.77	Comparaison sans optim du code (gauche) et avec (droite) des fonctions coûts sur le M	59
4.78	Forme initiale (gauche) et finale (droite)	59
4.79	Fonction coût (gauche) et volume (droite) sur 30 itérations, $\alpha_l = 2$, maillage console	60
4.80	Forme initiale (gauche) et finale (droite)	60
4.81	Forme a la 30ième itération (gauche) et finale (60ième itération) (droite)	60
4.82	Fonction coût (gauche) et volume (droite) sur 60 itérations, $\alpha_l = 2$, maillage console	61
4.83	Forme initiale (gauche) et finale (droite)	62
4.84	Fonction coût (gauche) et volume (droite) sur 30 itérations avec $\alpha_l = 2$	62
4.85	Différentes étapes de l'optimisation, initiale, étape 5, 10, 15, 20, 25, 30 et 35	63
4.86	Irrégularité d'un coin	65