HAL
open science

Optimisation en présence d'incertitudes de la répartition de la charge dans les problèmes elliptiques : Application à la mécanique des structures.

Radia Bouabdallah

▶ **To cite this version:**

Radia Bouabdallah. Optimisation en présence d'incertitudes de la répartition de la charge dans les problèmes elliptiques : Application à la mécanique des structures.. Probabilités [math.PR]. Université Montpellier, 2021. Français. NNT : 2021MONTS028 . tel-03391079

HAL Id: tel-03391079

https://theses.hal.science/tel-03391079

Submitted on 21 Oct 2021

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

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

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Mathématiques et modélisation

École doctorale I2S - Information, Structures, Systèmes

Unité de recherche UMR 5149 - IMAG - Institut Montpelliérain Alexander Grothendieck

Optimisation en présence d'incertitudes de la répartition de la charge dans les problèmes elliptiques: Application à la mécanique des structures

Présentée par Radia Bouabdallah

Le 6 Juillet 2021

Sous la direction de Bijan MOHAMMADI

Devant le jury composé de

Françoise KRASUCKI	Professeure	Université de Montpellier	Présidente
Dalila AZZAM-LAOUIR	Professeure	Université de Jijel	Rapporteure
Terence BAYEN	Professeur	Université d'Avignon	Rapporteur
Jérôme BOLTE	Professeur	Toulouse School of Economics	Examinateur
Alexandre CABOT	Professeur	Université de Bourgogne	Examinateur
Bijan MOHAMMADI	Professeur	Université de Montpellier	Directeur

UNIVERSITÉ DE MONTPELLIER

In memoria du Professeur Marc-Olivier Czarnecki

Remerciements

Je voudrais remercier grandement mon directeur de thèse, Bijan MOHAMMADI, pour toute son aide. Je suis ravie d'avoir travaillé en sa compagnie car outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse.

Mes pensées vont à Madame la Professeure Saadia TAS qui devait faire partie de mon jury de thèse et qui, malheureusement, nous a quitté beaucoup trop tôt.

J'adresse tous mes remerciements à Madame la Professeure Dalila AZZAM et à Monsieur le Professeur Terence BAYEN de l'honneur qu'ils me font en évaluant mon travail.

J'exprime ma gratitude à Madame la Professeure Françoise KRASUCKI, à Monsieur le Professeur Jérôme BOLTE et à Monsieur le Professeur Alexandre CABOT pour avoir accepté d'examiner cette thèse.

Je voudrais aussi remercier le directeur du laboratoire Institut Montpelliérain Alexander Grothendieck de m'avoir accueillie chaleureusement et dans de très bonnes conditions de travail.

Par ailleurs, je tiens également à adresser mes sincères remerciements aux permanents, aux anciens et futurs docteurs ainsi que les stagiaires pour leur soutien mutuel, leurs savoirs scientifiques et leurs échanges amicaux et les bons moments partagés pendant ces trois années de dur labeur.

Je souhaite remercier Robert RAPADAMNABA pour son soutien constant dans mon travail.

Je remercie le MINISTÈRE DE L 'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE qui a financé cette thèse.

Je remercie aussi sincérement tous mes amis qui m'ont apportée leur soutien moral pendant ces années d'études.

Un très grand MERCI à toute ma famille qui m'a gratifiée de son amour et fourni les motivations qui ont permis l'aboutissement de mes entreprises. Je leur adresse toute ma gratitude du fond du cœur.

Enfin, je remercie Mohammed qui a su me soutenir, me supporter, m'encourager pendant toute la durée de ma thèse.

Résumé. Cette thèse porte sur l'application d'une méthode directe d'optimisation de répartition des charges mécaniques sur une structure. Le but est d'amener la structure à une déformation prescrite. On montre comment tirer avantage de la présence de linéarité entre les variables d'optimisation et la solution de l'équation d'état pour décliner explicitement la solution optimale au sens des moindres carrées, et ceci sans aucune méthode itérative de type gradient. On montre, par ailleurs, comment cette méthode permet une analyse de sensibilité de la solution par rapport aux petites perturbations d'une part des charges, et d'autre part, de la déformation cible. Ces analyses de sensibilités utilisent des stratégies à faible complexité de propagation directe et rétrograde des incertitudes, et on montre que là aussi les opérations peuvent être déclinées de façon explicite.

Abstract. This thesis concerns the application of a direct method of optimization of the distribution of mechanical loads on a structure. The goal is to bring the structure to a prescribed deformation. We show how to take advantage of the presence of linearity between the optimization variables and the solution of the equation of state to decline explicitly the optimal solution in the sense of the least squares, and this without any iterative method of type gradient. We show, moreover, how this method allows an analysis of the sensitivity of the solution with respect to small disturbances on the one hand of the loads, and on the other hand, of the target strain. These sensitivity analyzes use low complexity strategies of direct and retrograde propagation of uncertainties, and we show that here too the operations can be explicitly declined.

Table des matières

1 Introduction **11**
1.1 Optimisation sous contraintes d'égalité et d'inégalité 12
1.2 Optimisation de formes par la méthode des frontières immergées . . . 12
1.3 Parallélisme, rapidité de calcul et calcul temps-réel 15
1.4 Propagation des incertitudes 15
1.5 Applications possibles 16
1.5.1 Applications aux géométries complexes 16

2 Équations elliptiques et discrétisation du problème de Poisson **19**
2.1 Approximation interne du problème 19
2.2 Maillage 20
2.3 Éléments finis de Lagrange triangulaires de degré un : les éléments finis P1 21
2.3.1 Les fonctions de base P1 21
2.3.2 Les fonctions de forme P1 21
2.4 Application à un problème elliptique modèle 23
2.4.1 Formulation variationnelle de ce problème 23
2.4.2 Écriture du problème approché en éléments finis P1 23
2.4.3 Calcul de la matrice de raideur élémentaire P1 25
2.4.4 Calcul des seconds membres élémentaires 26
2.4.5 Algorithme d'assemblage 28
2.5 Application aux problèmes elliptiques 28
2.5.1 Calcul des matrices et second-membres élémentaires 28

3 Elasticité linéaire en contraintes planes **31**
3.1 Formulation variationnelle d'un problème d'élasticité linéaire. Principe des travaux virtuels 33
3.2 Approximation par éléments finis P1 34
3.3 Calculs des matrices et second-membre élémentaires P1 35
3.4 Exemple de calcul avec Cast3M 38

4 Optimisation **45**
4.1 Définition du Problème 46
4.2 Résultats de base de l'optimisation sans contraintes 46
4.3 Projection sur un sous-espace fermé 46

4.4 Équivalence entre résolution d'un système linéaire et minimisation quadratique . . . 47
4.5 Rappel sur les moindres carrés . . . 47
4.6 Répartition de charges . . . 48
4.6.1 Supports de charges élémentaires pour le calcul des solutions élémentaires par noyau . . . 49
4.6.2 Lien avec les méthodes de descente . . . 50
4.7 Exemple numérique avec un logiciel boîte-noire . . . 51
4.7.1 Étude de l'effet de la représentation du support de la charge . 54
4.8 Application de la méthode directe en élasticité linéaire tridimensionnelle 57
4.9 Optimisation sous contrainte . . . 61
4.9.1 contrainte d'égalité . . . 61
4.9.2 Contraintes inégalités . . . 64

5 Estimation d'erreur a posteriori et adaptation de maillages 69
5.1 Analyse d'erreur a priori dans les méthodes d'éléments finis . . . 69
5.1.1 Majoration d'erreur en éléments P_k avec intégration exacte . . 70
5.1.2 Les éléments P1 en dimension deux . . . 70
5.2 Contrôle local de métrique . . . 71
5.3 Adaptation de maillage pour le problème Poisson . . . 72
5.4 Impact du contôle de la qualité du maillage sur la solution du problème d'optimisation . . . 75

6 Propagation des incertitudes 77
6.1 Introduction . . . 78
6.2 State equations and single-point optimization problem . . . 79
6.2.1 3D elasticity . . . 81
6.2.2 Illustration using $N = 5$ optimization variables . . . 81
6.3 Robust parametric optimization . . . 82
6.3.1 Moment-based optimization . . . 84
6.3.2 Discrete expressions . . . 85
6.3.3 Gradients of the moments . . . 85
6.3.4 Controlling the moments . . . 87
6.3.5 Multi-point optimization for a quadratic functional . . . 88
6.3.6 Application to our 3D elastic problem . . . 88
6.4 Linear models and backward uncertainty propagation : covariance matrix of the optimization variables . . . 93
6.4.1 $Cov^1_{\mathbf{x}}$ when a linear model $L_1 u = \mathbf{x}$ is available . . . 93
6.4.2 $Cov^2_{\mathbf{x}}$ when a linear model $L_2 \mathbf{x} = u$ is available . . . 94
6.5 Concluding remarks . . . 98

7 Conclusion et perspectives 99

8 Annexe : calcul élastique avec Cast3M 101

Chapitre 1

Introduction

Le but de cette thèse est de proposer une méthode directe d'optimisation de la répartition de charges mécaniques sur une structure pour en assurer une déformation minimale ou prescrite. Ce problème apparaît par exemple lorsqu'il faut distribuer de façon optimale une cargaison.

Cette thèse présente les éléments suivants :

- Modélisation d'un problème d'élasticité en vue de l'optimisation de la répartition de la charge mécanique,
- Optimisation de la déformation d'une structure en optimisation de la répartition des charges,
- Prise en compte de contraintes d'égalité et d'inégalité,
- Discrétisation du problème sur maillage à pas variable et adaptation de maillages non-structurés par contrôle de métrique,
- Analyse de l'effet du contrôle d'erreur de discrétisation par l'adaptation de maillage sur le résultat du problème d'optimisation,
- Analyse de sensibilité et propagation rétrograde des incertitudes.

Un avantage majeur de la technique que nous présentons est son applicabilité avec des outils informatiques boîtes-noires.

En effet, il s'agit d'un problème d'optimisation distribué sous contraintes d'égalité et d'inégalité. Ce problème fait intervenir des équations aux dérivées partielles pour la description des variables physiques, par exemple le déplacement.

Nous devons donc utiliser une méthode d'optimisation sous contrainte.

Or, comme la taille du problème d'optimisation peut être grande, on sera contraint à mettre en œuvre une optimisation utilisant les gradients de la fonctionnelle et des contraintes car les méthodes sans gradient [5, 10, 11] (e.g. algorithme recuit simulé, génétique, CMAES, Méthode de Nelder-Mead, Optimisation aléatoire, etc) seront alors trop coûteux à mettre en place.

De même, calculer par les différences finies les gradients de quantités évaluées par des outils boîtes-noires n'est pas envisageable lorsque la dimension du problème est

grande. Ainsi, le coût calculatoire devient un vrai enjeu les problèmes d'optimisation de grande taille impliquant des outils industriels.

La thèse propose d'adresser cette question dans certaines situations où il sera possible de tirer avantage de la présence éventuelle de linéarité entre la solution de l'équation d'état et les variables d'optimisation. L'approche proposée permet de décliner les opérations en termes de calculs de solutions élémentaires évaluables en parallèle de façon indépendante.

Par ailleurs, l'approche proposée peut être appliquée aux problèmes linéarisés dans le cas non-linéaire ou lorsque une dépendance linéaire n'est pas explicite. On cherchera alors à trouver les variations optimales autour d'une conception déjà disponible. Bien entendu, ceci est très utile pour analyser et/ou améliorer la stabilité de la configuration de base.

Nous allons décrire les ingrédients mathématiques des outils dans le cadre bi et tri-dimensionnel pour les problèmes d'équations aux dérivées partielles elliptiques. Nous illustrerons notre propos pour le problème de Poisson et d'élasticité linéaire en utilisant des outils de simulation boîtes-noires, notamment un code industriel.

1.1 Optimisation sous contraintes d'égalité et d'inégalité

Comme nous avons indiqué, notre problème présente toujours des contraintes d'égalité données par la présence de l'équation d'état à respecter. Mais il peut exister aussi d'autres contraintes. Dans la suite, lorsque de telles contraintes existent, nous appellerons le problème sous contraintes, sachant que la contrainte 'équation d'état' est toujours présente.

En effet, une situation classique est la présence de contrainte sur les variables d'optimisation, il s'agit de chercher le minimum d'une fonctionnelle, sous un ensemble de contraintes d'égalités et d'inégalités. Par exemple, on peut être contraint à un respect de la charge totale dans certaines situations qui correspond à une contrainte d'égalité.

De même, nous pouvons avoir besoin de prendre en compte des contraintes d'inégalité. Par exemple, pour des raisons de fiabilité ou de contrainte de construction, les charges devront rester inférieures à un certain seuil.

1.2 Optimisation de formes par la méthode des frontières immergées

Un autre intérêt de l'approche direct avec pré-calcul est qu'elle peut être appliquée à l'optimisation de formes en utilisant la méthode levet set ou de frontière immergée pour la paramétrisation des formes. Bien sûr, ceci lorsque l'équation d'état est linéaire.

La méthode de level set ou des lignes de niveau est une technique bien établie pour représenter les frontières mobiles dans un domaine. On rencontre cette technique sous divers noms : frontière immergée, domaine fictif, pénalisation géométrique,... [12, 13, 14, 27, 29, 37, 38, 39, 40]. Toutes ces méthodes permettent de prendre en compte une frontière de façon implicite par l'introduction d'une fonction distance. En particulier, la frontière n'est plus constituée de points, segments et faces du maillage. Le maillage ne décrit plus explicitement la frontière. Rappelons brièvement cette technique.

La paramétrisation d'une frontière Γ par la méthode level set est basée sur la ligne de niveau zéro de la fonction ψ :

$$\Gamma = \{x \in \Omega \ : \ \psi(x) = 0\}.$$

La fonction ψ est, par exemple, la distance Euclidienne signée à Γ :

$$\psi(x) = \pm \inf_{y \in \Gamma} \|x - y\|,$$

avec la convention de positivité si $x \in \Omega$ et négativité sinon. Ainsi, on a (en dimension d d'espace) :

$$\psi|_\Gamma = 0, \quad \psi|_{\mathbb{R}^d \setminus \Omega} < 0, \quad \psi_\Omega > 0. \tag{1.1}$$

La méthode permet de prendre en compte les mouvements ou déformations de la frontière. Si la frontière se déplace à la vitesse V, sous une forme discrète après introduction d'un pseudo-temps, en $\zeta + \delta\zeta$ la frontière s'exprime toujours par :

$$\Gamma = \{x \ : \ \psi(\zeta + \delta\zeta, x + V\delta\zeta) = 0\}.$$

On peut suivre cette frontière en suivant directement l'évolution de la fonction ψ [4, 31, 42] comme solution de l'équation de transport suivant :

$$\frac{\partial \psi}{\partial \zeta} + V \nabla \psi = 0.$$

La normale à Γ si Γ est définie par (1.1) est donnée par $n = \nabla\psi / |\nabla\psi|$ en $\psi = 0$. Ceci est utile pour exprimer, par exemple, une condition aux limites de Neumann sur un corps creux.

Sur un maillage donné, ψ est représentée par sa forme discrète ψ_h qui permet d'introduire de nouvelles frontières Γ_h dans le domaine et de prendre en compte des conditions aux limites sur ces frontières. Les corps creux sont des exemples de frontières introduites dans un domaine initialement plein. Une condition aux limites F_{Γ_h} sur ces nouvelles frontières pour une équation générique discrète générique (élasticité linéaire dans notre cas) $F_h(u_h) = 0$ peut être prise en compte par la solution de l'équation augmentée suivante :

$$F_h(u_h)\chi(\psi_h) + F_{\Gamma_h}(u_h)\delta_{\psi_h} = 0.$$

En pratique, on prolonge régulièrement F_{Γ_h} dans le domaine et δ_ψ est une fonction régulière valant 1 sur Γ_h et zéro ailleurs et permettant de localiser la frontière Γ_h.

Dans un contexte d'optimisation de formes, les déformations de formes sont spécifiées à travers $\chi(\Gamma_h)$. Ceci permet d'éviter les complications liées à la gestion des déformations de formes et aux remaillages.

Illustrons cette technique sur un problème simple avec une équation d'état linéaire comme celles que nous allons rencontrer :

$$u - u_{\mathbf{xx}} = 0, \quad \text{sur} \quad]0,1[\tag{1.2}$$

avec comme conditions aux limites :

$$u(0) = 0, \quad u(1) = 1,$$

et la présence d'une intrusion en $\mathbf{x} \in]0.5-\eta, 0.5+\eta[$. Supposons que nous avons besoin d'appliquer une conditions de Dirichlet non-homogène à l'interface entre cette intrusion et le domaine : $u(\mathbf{x} = 0.5^{\pm}) = g$. Cette expression indique que nous avons supposé $\eta << 1$, très petit devant la taille du domaine initial. Ceci est un cas courant et il est intéressant de ne pas avoir à mailler ce petit domaine.

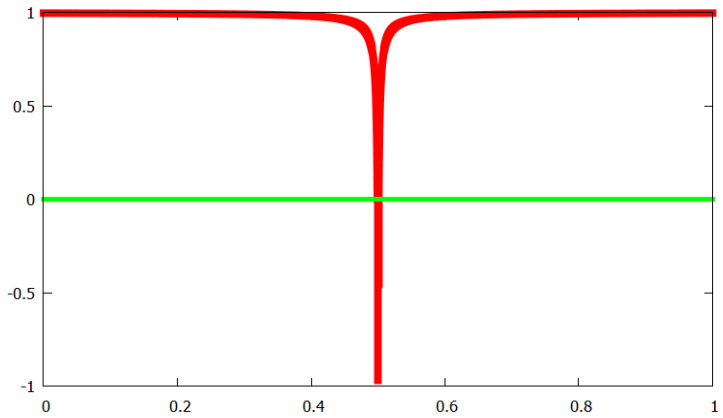

FIGURE 1.1 – Illustration de la fonction distance signée χ pour $\varepsilon = 10^{-3}$.

Prenons en compte la présence de cet objet par la méthode de level set en introduisant la paramétrisation par la fonction distance signée à ce petit domaine. On cherche u^ε solution de l'équation :

$$\chi(u^\varepsilon - u^\varepsilon_{\mathbf{xx}}) + (1-\chi)(u^\varepsilon - g) = 0, \tag{1.3}$$

où χ est définie par (illustrée par la figure 1.1 pour $\varepsilon = 0.001$) :

$$\chi = 2\left(\frac{|\mathbf{x}-0.5|}{|\mathbf{x}-0.5|+\varepsilon} - 0.5\right),$$

et si l'intrusion est réduite à un point, simplement par :

$$\chi = \frac{|\mathbf{x}-0.5|}{|\mathbf{x}-0.5|+\varepsilon}.$$

L'équation se résume à :

$$u^\varepsilon - u^\varepsilon_{\mathbf{xx}} + \frac{\varepsilon}{|\mathbf{x} - 0.5|}(u^\varepsilon - g) = 0 \tag{1.4}$$

$$(1 + \frac{\varepsilon}{|\mathbf{x} - 0.5|})u^\varepsilon - u^\varepsilon_{\mathbf{xx}} = \frac{\varepsilon}{|\mathbf{x} - 0.5|}g$$

On constate que la méthode de level set résout l'équation initiale augmentée d'un terme source et avec une nouvelle matrice de masse. Ce cadre, comme nous le verrons, convient parfaitement à notre méthode directe d'optimisation avec pré-calcul de solutions élémentaires correspondant aux cas $g = 0$ et $g = 1$.

1.3 Parallélisme, rapidité de calcul et calcul temps-réel

On s'intéresse à un aspect de l'informatique parallèle qui s'appelle parallélisme embarrassant (embarassingly parallel en anglais, également appelé parfaitement parallèle). Une tâche parallèle embarrassante peut être considérée comme un cas trivial - peu ou pas de manipulation est nécessaire pour séparer le problème en un certain nombre de tâches parallèles [2, 19]. Ceci est souvent le cas lorsqu'il y a peu ou pas de dépendance ou de besoin de communication entre ces tâches parallèles. C'est le cas dans notre méthode directe d'optimisation. Ceci est un autre avantage fondamental de la technique où les traitements élémentaires ont une nature parfaitement parallèle. En effet, ces solutions étant indépendantes, leurs pré-calculs peuvent être effectués en parallèle et de façon indépendante.

De même, la rapidité du calcul de la solution optimale, une fois les solutions élémentaires obtenues en pré-calcul, permet d'envisager le temps-réel. On pourra adapter la répartition de charge de façon instantanée pour toute évolution de la solution cible. là aussi, ceci concerne de nombreuses applications en acoustique par exemple.

1.4 Propagation des incertitudes

Nous avons montré comment l'approche directe permet d'optimiser la distribution d'un chargement pour atteindre une déformation ou déplacement cible u^*. Et nous avons vu que la solution optimale est calculée très rapidement une fois que la base de solutions élémentaires a été constituée en pré-calcul et en parallèle.

Les solutions élémentaires sont indépendantes des incertitudes éventuelles de la solution cible qui peut être prescrite, non pas de façon déterministe, mais uniquement aux travers des intervalles d'admissibilité par exemple.

La rapidité d'estimation de la distribution de charge optimale permet d'analyser la sensibilité de la solution optimale aux petites perturbations de la déformation cible et aussi d'envisager une propagation rétrograde des incertitudes.

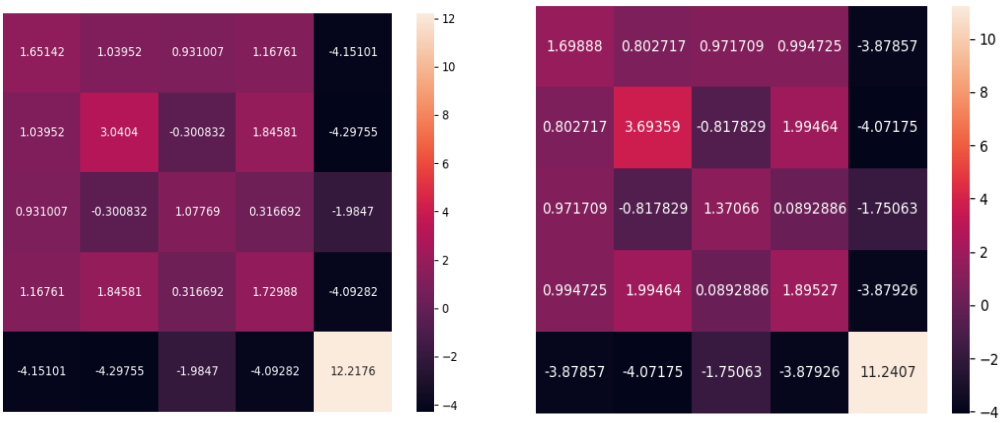

FIGURE 1.2 – Deux estimations à faible complexité de la matrice de covariance illustrant les itérations entre les paramètres d'optimisation.

Ainsi, le dernier chapitre de la thèse montre comment tirer avantage de la simplicité de la méthode directe d'optimisation lors de la propagation rétrograde des incertitudes et construire avec un coût de calcul supplémentaire négligeable la matrice de covariance des variables d'optimisation. Deux procédures d'estimation de cette matrice sont présentées en tirant avantage de l'existence de présence de linéarité entre les variables d'optimisation et la solution de l'équation d'état. La figure 1.2 montre la comparaison de ces deux estimations de $Cov_{\mathbf{x}}$.

1.5 Applications possibles

Ce problème est assez générique est concerne aussi bien les problèmes d'élasticité présentés dans cette thèse que, par exemple, la répartition de la charge électrique pour assurer une distribution de potentiel électrique cible.

Aussi, les techniques présentées dans cette thèse peuvent trouver des extensions et applications multiples. Par exemple, une question complémentaire industrielle vient de la minimisation du nombre de points d'intervention. Ainsi, une fois la base de données de solutions élémentaires constituée, on peut s'intéresser à la résolution de ce problème d'optimisation dont le coût de calcul restera contenu dans la mesure où on utilisera uniquement les solutions déjà calculées.

1.5.1 Applications aux géométries complexes

Nous allons illustrer comment la méthode précédente peut être mise en œuvre dans un cadre industriel pour un problème complexe représenté par une dalle en béton avec corps creux.

Une dalle de béton est un élément structurel commun des bâtiments modernes, consistant en une surface plane et horizontale de béton coulé, des versions améliorées des dalles solides sont les dalles à corps creux. Où les corps creux (hourdis ou bien entrevous) sont des éléments utilisés dans des constructions génie civile. Ces corps contiennent des vides à l'intérieur.

Ces dernières années, de nombreuses entreprises de construction ont commencé à développer de nouvelles formes de corps creux et différents modes de distribution de ces derniers pour avoir une dalle plus résistante, comme cobiax, beeplate, ...

L'inclusion de corps creux dans les dalles pleines permet d'augmenter les performances mécaniques et thermiques de ces dernières 1.3. Sans être exhaustif, on peut citer les points d'amélioration suivants pour une dalle comportant de telles inclusions :

— une plus grande résistance, notamment en situation sismique,
— un poids réduit,
— une économie de la quantité du béton,
— une plus grande flexibilité dans l'organisation de l'espace intérieur des dalles,
— un passage plus aisé des conduits et tuyaux,
— une mise en oeuvre accélérée,
— une réduction des coûts de construction.

Les architectes préfèrent souvent utiliser des dalles en béton pour construire un plancher au rez-de-chaussée ou aux étages supérieurs car elles peuvent supporter de lourdes charges. Mais la question qui se pose alors est la limite de ces charges par unité de surface. La question de la répartition de la charge devient donc centrale. Par exemple, une installation sécurisée d'éléments lourds, comme la climatisation, une piscine intérieure, etc, ne peut se réaliser sans prendre en compte les normes de charge maximale des planchers.

FIGURE 1.3 – Exemple d'une dalle en corps creux (image-`www.batiproduits.com`).

Chapitre 2

Équations elliptiques et discrétisation du problème de Poisson

Nous allons présenter notre méthodologie dans le cadre bidimensionnel. Cependant, sa mise en œuvre ne pose aucune difficulté en 3D.

On s'intéresse aux problèmes de la forme générale du problème P suivant :

$$(P)\begin{cases} \text{Trouver la fonction } u \text{ appartenant à l'Hilbert } V \text{ telle que :} \\ a(u,v) = l(v) \quad \forall v \in V \end{cases} \tag{2.1}$$

avec V un espace de Hilbert réel muni du produit scalaire $(\, . \, , \, . \,)$ et de la norme associée $\| \, . \, \|$, a une forme bilinéaire continue et elliptique sur V et l une forme linéaire continue sur V. Le théorème de Lax-Milgram conduit alors aux conclusions suivantes :

1) Le problème P admet une solution unique u dans V
2) Si la forme bilinéaire a est symétrique, le problème variationnel (P) est équivalent au problème de minimisation suivant :

$$\begin{cases} \text{Trouver } u \in V \text{ qui réalise le minimum de la forme quadratique} \\ J(v) = \frac{1}{2}\, a(v,v) \; - \; l(v) \end{cases} \tag{2.2}$$

2.1 Approximation interne du problème

On doit maintenant construire un sous-espace $V_h \subset V$ de dimension finie dans lequel s'écrira le problème approché P_h :

$$(P_h)\begin{cases} \text{Trouver la fonction } u_h \text{ appartenant à l'espace } V_h \text{ telle que :} \\ a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h \end{cases} \tag{2.3}$$

Ce problème admet également une solution unique car $V_h \subset V$, et donc les hypothèses du théorème de Lax-Milgram sont également vérifiées dans V_h.

Si la forme bilinéaire a est symétrique le problème variationnel P_h est équivalent au problème de minimisation suivant :

$$\begin{cases} \text{Trouver } u_h \in V_h \text{ qui réalise le minimum de la forme quadratique} \\ \qquad\qquad J(v_h) = \frac{1}{2}\, a(v_h, v_h) \; - \; l(v_h) \end{cases} \tag{2.4}$$

et on a donc évidemment

$$J(u_h) \;\geq\; J(u)$$

2.2 Maillage

Dans la méthode des éléments finis, la construction du sous-espace $V_h \subset \Omega$ nécessite la discrétisation préalable du domaine Ω en éléments géométriques simples.

En dimension un, la discrétisation préalable du domaine, un intervalle de $I\!R$, en éléments, ne posait pas de difficultés. En dimension deux et plus encore en dimension trois, la discrétisation du domaine Ω est un problème technique difficile. La qualité du maillage est cruciale pour la qualité de l'approximation. Le problème de la réalisation du maillage se pose à la fois en amont de la résolution numérique, qui s'appuie sur une discrétisation a priori, et en aval dans les techniques de maillages adaptatifs par lesquelles on s'efforce d'améliorer la qualité de la discrétisation en fonction des résultats obtenus.

On peut distinguer deux types de maillages :

— Les maillages structurés qui fournissent une discrétisation "régulière" obtenue par transformation d'une grille régulière sur un domaine rectangulaire.

— Les maillages non-structurés, principalement construits par la méthode de Delaunay-Voronoi, qui peuvent s'appliquer aux géométries les plus générales.

Un maillage en éléments finis conformes doit satisfaire les critères suivants :

1) Les éléments K_i du maillage doivent recouvrir le domaine Ω

$$\bigcup_i K_i = \bar{\Omega}$$

Ceci implique que, par exemple en dimension deux, ce domaine soit polygonal ou approché par un polygone si l'on utilise des éléments droits.

2) L'intersection de deux éléments distincts ne peut être que
- l'ensemble vide
- un sommet
- un côté
- une face (en dimension trois)

Ceci a pour but d'assurer la continuité des fonctions de V_h, on parle alors d'éléments **conformes**.

2.3 Éléments finis de Lagrange triangulaires de degré un : les éléments finis P1

Dans ce cas l'espace d'approximation V_h est un espace de fonctions continues affines par éléments triangulaires. Dans chaque triangle, la restriction des fonctions de V_h est donc un polynôme de degré un de la forme $a_0 + a_1\, x + a_2\, y$ qui est donc déterminé de façon unique par ses valeurs en trois points distincts. On choisit les trois sommets du triangle. Ceci assure la continuité globale des fonctions de V_h sur le domaine polygonal Ω. En effet sur chaque arête commune à deux triangles adjacents, les restrictions des fonctions de V_h sont des fonctions affines fixées de manière unique par la donnée de leurs valeurs aux deux sommets sur l'arête.

Globalement les fonctions de V_h seront uniquement déterminées par leurs valeurs aux sommets ou noeuds de la triangulation. **La dimension totale de V_h est donc égale au nombre de noeuds du maillage.** Dans le cas de conditions aux limites de Dirichlet sur une partie de la frontière, la dimension de V_h est évidemment réduite au nombre des noeuds associés à une valeur inconnue de la solution. Elle est donc égale au nombre de noeuds total moins le nombre de noeuds où la solution est fixée par une condition de Dirichlet.

2.3.1 Les fonctions de base P1

La construction d'une base de V_h se fait selon la technique classique de Lagrange. On prend comme fonctions de base les N fonctions w_I de V_h définies par les N conditions suivantes aux N noeuds (x_I, y_I) du maillage :

$$w_I(x_J, y_J) = \delta_{IJ}$$

On remarquera que ses fonctions ont un support réduit à l'union des triangles dont le point (x_I, y_I) est un sommet. Dans cette base une fonction de V_h s'écrit :

$$v_h(x, y) = \sum_I v_I\, w_I(x, y)$$

2.3.2 Les fonctions de forme P1

On appellera fonctions de forme les restrictions des fonctions de base dans un élément. Dans chaque triangle T de sommets A_1, A_2, A_3, il n'y a que 3 fonctions de base non-nulles. Les restrictions de ces fonctions de base $w_{I_1}, w_{I_2}, w_{I_3}$ sont les trois fonctions polynomiales de degré un prenant la valeur 1 en un des sommets et nulles aux deux autres sommets. Notons les respectivement $\lambda_1, \lambda_2, \lambda_3$.

λ_1 est le polynôme de degré un prenant la valeur 1 en A_1 et nul en A_2 et A_3.

$$\lambda_1(x, y) = a_0 + a_1 x + a_2 y$$

λ_1 est donc déterminé par le système linéaire suivant :

$$\begin{cases} \lambda_1(x_1, y_1) = a_0 + a_1 x_1 + a_2 y_1 = 1 \\ \lambda_1(x_2, y_2) = a_0 + a_1 x_2 + a_2 y_2 = 0 \\ \lambda_1(x_3, y_3) = a_0 + a_1 x_3 + a_2 y_3 = 0 \end{cases}$$

Le déterminant de ce système

$$\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2.Aire(T)$$

est différent de zéro si les points A_1, A_2, A_3 ne sont pas alignés.

En résolvant le système 3×3 ci-dessus et les systèmes analogues pour λ_2 et λ_3 on obtient les formules suivantes :

$$\lambda_1(x, y) = \frac{x_2 y_3 - x_3 y_2 + x(y_2 - y_3) + y(x_3 - x_2)}{2\, Aire\,(T)} \tag{2.5}$$

$$\lambda_2(x, y) = \frac{x_3 y_1 - x_1 y_3 + x(y_3 - y_1) + y(x_1 - x_3)}{2\, Aire\,(T)} \tag{2.6}$$

$$\lambda_3(x, y) = \frac{x_1 y_2 - x_2 y_1 + x(y_1 - y_2) + y(x_2 - x_1)}{2\, Aire\,(T)} \tag{2.7}$$

On en déduit les expressions suivantes des gradients :

$$\frac{\partial \lambda_1}{\partial x} = \frac{y_2 - y_3}{2\, Aire\,(T)} \qquad \frac{\partial \lambda_1}{\partial y} = \frac{x_3 - x_2}{2\, Aire\,(T)} \tag{2.8}$$

$$\frac{\partial \lambda_2}{\partial x} = \frac{y_3 - y_1}{2\, Aire\,(T)} \qquad \frac{\partial \lambda_2}{\partial y} = \frac{x_1 - x_3}{2\, Aire\,(T)} \tag{2.9}$$

$$\frac{\partial \lambda_3}{\partial x} = \frac{y_1 - y_2}{2\, Aire\,(T)} \qquad \frac{\partial \lambda_3}{\partial y} = \frac{x_2 - x_1}{2\, Aire\,(T)} \tag{2.10}$$

Les trois fonctions $\lambda_1, \lambda_2, \lambda_3$ s'appellent les coordonnées barycentriques du triangle T. On les désigne sous le nom “area coordinates” en anglais car elles représentent en chaque point M de coordonnées x, y le rapport des aires algébriques des triangles MA_iA_j et T . Par exemple

$$\lambda_1(M) = \frac{Aire(A_2A_3M)}{Aire(T)}$$

Une fonction quelconque de V_h prenant les valeurs v_1, v_2, v_3 aux sommets A_1, A_2, A_3 du triangle T s'écrit dans T sous la forme

$$v_h(x, y) = v_1\lambda_1(x, y) + v_2\lambda_2(x, y) + v_3\lambda_3(x, y)$$

2.4 Application à un problème elliptique modèle

Le problème de Poisson est le problème modèle idéal pour présenter la mise en oeuvre de la discrétisation par éléments finis.

Soit Ω un domaine borné polygonal de $I\!R^2$ de frontière Γ. Supposons Γ constituée de deux parties Γ_0 et Γ_1, $\Gamma = \Gamma_0 \cup \Gamma_1$.

Sur Γ_0 sont imposées des conditions de Dirichlet.
Sur Γ_1 sont imposées des conditions de Neumann.

Nous obtenons le problème mixte suivant :

$$\begin{cases} -\Delta\, u\,(x,y) = f\,(x,y) \qquad \forall\, x,y \in \Omega \\ u_{|\Gamma_0} = u_d \\ \dfrac{\partial u}{\partial n}_{|\Gamma_1} = g \end{cases} \tag{2.11}$$

2.4.1 Formulation variationnelle de ce problème

La formulation du problème s'écrit dans l'espace V des fonctions de $H^1(\Omega)$ nulles sur la partie Γ_0 de la frontière. On se ramène aux conditions de Dirichlet homogènes sur Γ_0 en introduisant une fonction auxiliaire simple prenant les valeurs imposées sur la frontière Γ_0 : $u_{0_{|\Gamma_0}} = u_d$ et en posant $u = \tilde{u} + u_0$. On obtient le problème variationnel :

$$\begin{cases} \text{Trouver } \tilde{u} \in V \text{ telle que : } \quad \forall\, v \in V \\ \displaystyle\iint_\Omega \mathbf{grad}\tilde{u}\ \mathbf{grad}v\, dxdy = \iint_\Omega f v\, dxdy - \iint_\Omega \mathbf{grad}u_0\, \mathbf{grad}v\, dxdy + \int_{\Gamma_1} g\, v\, d\gamma \end{cases}$$

Ce problème s'écrit sous la forme générale standard :

$$\begin{cases} \text{Trouver } \tilde{u} \in V \text{ telle que :} \\ a(\tilde{u}, v) = l(v) \quad \forall\, v \in V \end{cases}$$

avec $a : u, v \longrightarrow a(u,v) = \displaystyle\iint_\Omega \mathbf{grad}u\ \mathbf{grad}v\, dxdy$: forme bilinéaire symétrique sur V
$l : v \longrightarrow l(v) = \displaystyle\iint_\Omega f\, v\, dxdy - \iint_\Omega \mathbf{grad}u_0\, \mathbf{grad}v\, dxdy + \int_{\Gamma_1} g\, v\, d\gamma$: forme linéaire sur V.

2.4.2 Écriture du problème approché en éléments finis P1

Le problème approché s'écrit dans l'espace V_h des fonctions continues affines par triangles et nulles sur la partie Γ_0 de la frontière. Notons $\mathbf{I}$ l'ensemble des indices des noeuds du maillage correspondants à une valeur inconnue de la solution u. C'est

à dire ici l'ensemble des noeuds n'appartenant pas à Γ_0. Notons $\mathbf{J}$ l'ensemble des indices des sommets du maillage appartenant à Γ_0.

La solution $\tilde{u}_h$ s'écrira dans la base des w_J pour $J \in \mathbf{I}$ selon :

$$\tilde{u}_h(x,y) = \sum_{J \in \mathbf{I}} u_J \, w_J(x,y)$$

La fonction auxiliaire u_0 sera approchée par une fonction $u_{0,h}$ continue et affine par morceaux prenant les valeurs imposées sur Γ_0 et nulle sur tous les noeuds d'indices $J \in \mathbf{I}$

$$u_{0,h}(x,y) = \sum_{J \in \mathbf{J}} u_d(x_J, y_J) \; w_J(x,y)$$

Ce choix de $u_{0,h}$ présente deux avantages :

1) La solution cherchée u_h et la solution calculée $\tilde{u}_h$ prennent les mêmes valeurs aux points où la solution u est inconnue. Il n'y a donc pas de transformation a posteriori à effectuer sur les résultats.

2) Les conditions aux limites ne produisent qu'une modification limitée du système linéaire qui n'intervient que sur quelques composantes du second-membre.

En intégrant dans la formulation variationnelle tous ces éléments on obtient en définitive le problème approché dans V_h

$$\begin{cases} \text{Trouver les valeurs } u_J \text{ pour } J \in \mathbf{I} \text{ telles que :} \\ \sum_{J \in \mathbf{I}} \left(\iint_\Omega \mathbf{grad} w_J \; \mathbf{grad} w_I \, dxdy \right) u_J = \iint_\Omega f \, w_I \, dxdy \; + \int_{\Gamma_1} g \, w_I \, d\gamma \\ - \sum_{J \in \mathbf{J}} \left(\iint_\Omega \mathbf{grad} w_J \, \mathbf{grad} w_I \, dxdy \right) u_d(x_J, y_J) \quad \forall \; I \in \mathbf{I} \end{cases}$$

On obtient un système de $N_\mathbf{I}$ équations à $N_\mathbf{I}$ inconnues où $N_\mathbf{I}$ désigne le nombre de points du maillage d'indices $I \in \mathbf{I}$ donc le nombre de noeuds correspondant à des valeurs inconnues de la solution. Ce système s'écrit sous la forme matricielle

$$K \; U \; = \; F$$

où K est la matrice de raideur de coefficients

$$K_{I,J} = \iint_\Omega \mathbf{grad} w_J \; \mathbf{grad} w_I \, dxdy$$

et F le vecteur second-membre de composantes :

$$F_I = \iint_\Omega f \, w_I \, dxdy \; + \int_{\Gamma_1} g \, w_I \, d\gamma - \sum_{J \; \in \; \mathbf{J}} \left(\iint_\Omega \mathbf{grad} w_J \; \mathbf{grad} w_I \, dxdy \right) u_d(x_J, y_J)$$

où l'on reconnaît :

— un premier terme représentant les efforts surfaciques (correspondant au second-membre du problème différentiel).

— un deuxième terme, intégrale curviligne, provenant des conditions aux limites de Neumann.
— un dernier terme, expression des conditions de Dirichlet non-homogènes.

Le calcul des coefficients $K_{I,J}$ de la matrice de raideur et des composantes F_I du second-membre se fait par une procédure d'assemblage des contributions apportées par chacun des éléments T_k de la triangulation.

Par exemple pour la matrice de raideur K :

$$K_{I,J} = \iint_{\Omega} \mathbf{grad} w_J \; \mathbf{grad} w_I \, dxdy = \sum_k \iint_{T_k} \mathbf{grad} w_J \; \mathbf{grad} w_I \, dxdy \tag{2.12}$$

On observe que la matrice K est très "creuse", un grand nombre de ses coefficients sont nuls, en raison du choix de fonctions w_I de support limité.

2.4.3 Calcul de la matrice de raideur élémentaire P1

Un des outils de base essentiels à la programmation de la méthode des éléments finis est un tableau de correspondances entre les noeuds X_I du maillage global et les points d'un élément particulier : ici les sommets A_1, A_2, A_3 des éléments triangulaires.

Dans chaque élément triangulaire T_k de sommets A_1, A_2, A_3 correspondants aux noeuds X_I, X_J, X_K, les seules fonctions de base non nulles sont les fonctions w_I, w_J, w_K. Leurs restrictions dans le triangle sont respectivement les trois coordonnées barycentriques $\lambda_1, \lambda_2, \lambda_3$ calculées précédemment. La matrice élémentaire relative au triangle T_k est donc une matrice 3×3 de coefficients :

$$elemK_{i,j} = \iint_{T_k} \mathbf{grad} \lambda_j \; \mathbf{grad} \lambda_i \, dxdy \quad \forall \, i, j = 1, 2, 3$$

Comme, dans ce cas particulier simple d'éléments $P1$, les gradients sont constants par triangles, les intégrales à calculer sont des intégrales de fonctions constantes. Il suffit d'en multiplier la valeur par l'aire de l'élément.

On obtient ainsi la matrice élémentaire $P1$ de coefficients :

$$\begin{pmatrix} \frac{(y_2-y_3)^2+(x_2-x_3)^2}{4\,Aire(T)} & \frac{(y_2-y_3)(y_3-y_1)+(x_2-x_3)(x_3-x_1)}{4\,Aire(T)} & \frac{(y_2-y_3)(y_1-y_2)+(x_2-x_3)(x_1-x_2)}{4\,Aire(T)} \\ \frac{(y_2-y_3)(y_3-y_1)+(x_2-x_3)(x_3-x_1)}{4\,Aire(T)} & \frac{(y_3-y_1)^2+(x_3-x_1)^2}{4\,Aire(T)} & \frac{(y_3-y_1)(y_1-y_2)+(x_3-x_1)(x_1-x_2)}{4\,Aire(T)} \\ \frac{(y_2-y_3)(y_1-y_2)+(x_2-x_3)(x_1-x_2)}{4\,Aire(T)} & \frac{(y_3-y_1)(y_1-y_2)+(x_3-x_1)(x_1-x_2)}{4\,Aire(T)} & \frac{(y_1-y_2)^2+(x_1-x_2)^2}{4\,Aire(T)} \end{pmatrix}$$

En particulier, dans le cas du triangle rectangle isocèle de sommets

$$A_1 = (0,0), A_2 = (1,0), A_3 = (0,1)$$

souvent utilisé comme triangle de référence (voir plus loin), on a la matrice élémentaire de raideur suivante :

$$elemK = \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

2.4.4 Calcul des seconds membres élémentaires

Le second membre se compose de 3 termes :

1) Le second membre

La gestion de ces termes est très importante dans le cadre de cette thèse pour la génération des solutions élémentaires. Il s'agit d'évaluer les termes :

$$Fs_I = \iint_\Omega f\, w_I\, dxdy \qquad \forall I$$

son calcul s'effectue en sommant les contributions de chaque élément triangulaire.

$$Fs_I = \sum_k Fs_{I,T_k} = \sum_k \iint_{T_k} f\, w_I\, dxdy$$

Si f est connue analytiquement et suffisamment simple on peut calculer exactement, à la main, les intégrales. Mais en général, le second membre f est connu par ses valeurs aux noeuds. On le représente dans la base des w_I et on est ramené aux calculs suivants

$$Fs_I = \iint_\Omega \sum_J f_J\, w_J\, w_I\, dxdy \qquad \forall I$$

Ce qui ramène aux calculs de

$$\iint_\Omega w_J\, w_I\, dxdy \qquad \forall I, J$$

Dans chaque élément on doit donc calculer la matrice de masse élémentaire.

Matrices élémentaires de masse

On obtient la matrice élémentaire de masse, en utilisant les formules d'intégration exactes suivantes :

$$\iint_T \lambda_i\, dxdy = \frac{Aire(T)}{3}$$

$$\iint_T \lambda_i^2\, dxdy = \frac{Aire(T)}{6}$$

$$\iint_T \lambda_j \, \lambda_i \, dxdy = \frac{Aire(T)}{12} \qquad \text{si } i \text{ différent de } j$$

Ce qui donne le résultat :

$$elemM_{T_k} = Aire(T_k) \begin{pmatrix} \frac{1}{6} & \frac{1}{12} & \frac{1}{12} \\ ... & \frac{1}{6} & \frac{1}{12} \\ ... & ... & \frac{1}{6} \end{pmatrix}$$

Matrice de masse condensée (mass lumping)

Il peut être avantageux de calculer la matrice de masse élémentaire de manière approchée en utilisant la formule suivante (exacte sur P1) :

$$\iint_T \phi \, dxdy = \frac{Aire(T)}{3} \sum_{i=1}^{3} \phi(A_i)$$

On obtient dans ce cas une matrice de masse diagonale égale à :

$$elemM_{T_k} = \frac{Aire(T_k)}{3} \, I$$

On en déduit le second membre élémentaire surfacique :

$$\begin{pmatrix} Fs_{I,T_k} \\ Fs_{J,T_k} \\ Fs_{K,T_k} \end{pmatrix} = elemM_{T_k} \begin{pmatrix} f_I \\ f_J \\ f_K \end{pmatrix}$$

2) un terme de bord provenant des conditions de Neumann

$$Fn_I = \int_{\Gamma_1} g \, w_I \, d\gamma$$

Son calcul s'effectue sur les éléments ayant un côté sur Γ_1. Il se ramène à une intégrale simple sur un côté A d'un triangle. Si la fonction g est donnée par ses valeurs aux noeuds du maillage et si A a pour extrémités X_I et X_J, on a sur A : $g = g_I w_I + g_J w_J$ et on doit donc calculer :

$$g_J \int_A w_J w_I \, d\gamma$$

$$g_J \int_A w_J^2 \, d\gamma$$

et

$$g_I \int_A w_I^2 \, d\gamma$$

Ces calculs s'effectuent exactement par la formule de Simpson :

$$g_J \int_A w_J \, w_I \, d\gamma = \frac{Longueur(A)}{6} \, g_J$$

$$g_J \int_A w_J^2 \, d\gamma = \frac{Longueur(A)}{3} \, g_J$$

etc..

2.4.5 Algorithme d'assemblage

Supposons un maillage en N éléments T_k pour $k = 1, 2..N$. Notons A la matrice globale à assembler (matrice de raideur ou de masse globale, second membre) et a_k les matrices élémentaires correspondantes relatives à chaque élément T_k. L'algorithme d'assemblage est très simple, dès lors que l'on dispose d'un tableau *numero* associant les sommets d'un élément T_k et les noeuds du maillage global. Dans ce cas simple d'éléments triangulaires P1, chaque élément T_k comprend trois noeuds X_I, X_J, X_K correspondants aux sommets A_1, A_2, A_3 du triangle T_k. D'où l'algorithme :

POUR k = 1, N FAIRE ! boucle sur les éléments

POUR i = 1, 3 FAIRE ! boucle sur les numéros locaux
I = numero(k,i) ! numéros globaux
POUR j = 1, 3 FAIRE
J = numero(k,j)
A(I,J) = A(I,J) + a_k(i,j) ! A : matrice globale, a matrice élémentaire

FIN DES 3 BOUCLES

2.5 Application aux problèmes elliptiques

2.5.1 Calcul des matrices et second-membres élémentaires

Ces problèmes font intervenir des intégrales de la forme :

$$\iint_\Omega \mathbf{grad} u \; \mathbf{grad} v \, dxdy$$

$$\iint_\Omega u \, v \, dxdy$$

$$\iint_\Omega f \, v \, dxdy$$

et éventuellement

$$\int_\Gamma g \, v \, d\gamma$$

Leur discrétisation, utilisant les fonctions de base w_I, nécessite le calcul des intégrales suivantes :

$$\iint_\Omega \mathbf{grad} w_J \; \mathbf{grad} w_I \, dxdy$$

$$\iint_\Omega w_J \, w_I \, dxdy$$

et

$$\int_\Gamma w_J \, w_I \, d\gamma$$

Ces calculs se ramènent à des intégrales sur les éléments triangulaires et leurs arêtes de fonctions composées de puissances et de produits des coordonnées barycentriques λ_i.

Nous montrerons des exemples de calcul éléments finis dans les chapitres suivants.

Chapitre 3

Elasticité linéaire en contraintes planes

Les problèmes de calcul de structures dans le cas de l'élasticité linéaire ont été à l'origine de la méthode des éléments finis. Considérons le problème de l'équilibre d'un domaine plan Ω, fixé sur une partie Γ_0 de sa frontière Γ, soumis à des efforts surfaciques de densité $\mathbf{f}$ et à un chargement $\mathbf{g}$ sur la partie Γ_1 de Γ. Ces forces s'exercent dans le plan de Ω.

Faisons l'hypothèse de petits déplacements pour lesquels les équations de l'élasticité linéaire sont valables et adoptons le modèle de contraintes planes adapté aux plaques minces sollicitées par des forces s'exerçant dans leur plan. L'inconnue du problème est le vecteur déplacement

$$\mathbf{u} = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix}$$

Le vecteur déformation est donné, dans le cas d'un problème en contraintes planes, par ses composantes fonction des dérivées premières des déplacements selon :

$$\epsilon = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{pmatrix}$$

Ce que l'on peut écrire en utilisant la matrice d'opérateurs différentiels

$$B = \begin{pmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{pmatrix}$$

sous la forme

$$\epsilon = B\mathbf{u}$$

Les équations de l'élasticité s'expriment par une relation linéaire entre contraintes et déformation qui dans un problème en contraintes planes s'écrit :

$$\sigma = \mathbf{D}\epsilon$$

où σ est le vecteur des contraintes :

$$\sigma = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}$$

et $\mathbf{D}$ est la matrice de Hooke dont l'expression, fonction du module de Young E et du coefficient de Poisson ν, s'écrit :

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{pmatrix}$$

Enfin les équations d'équilibre entre efforts et contraintes s'écrivent :

$$\begin{cases} \dfrac{\partial}{\partial x}\sigma_{xx} + \dfrac{\partial}{\partial y}\sigma_{xy} + f_x = 0 \\ \\ \dfrac{\partial}{\partial x}\sigma_{xy} + \dfrac{\partial}{\partial y}\sigma_{yy} + f_y = 0 \end{cases}$$

où f_x et f_y sont les 2 composantes des forces de surface.

En utilisant la matrice d'opérateurs différentiels B on peut écrire les équations d'équilibre sous la forme :

$$B^T\sigma + \mathbf{f} = 0$$

soit avec

$$\sigma = \mathbf{D}\epsilon \quad \text{et} \quad \epsilon = B\mathbf{u}$$

On obtient ainsi un système d'équations aux dérivées partielles du second ordre

$$B^T\mathbf{D}B\mathbf{u} + \mathbf{f} = 0$$

dont l'inconnue est le vecteur déplacement.

Conditions aux limites

Les conditions aux limites sont habituellement de 2 types.

Conditions de type Dirichlet, dites essentielles lorsque les déplacements sont fixés, par exemple à zéro sur la frontière.

$$u = v = 0 \quad \text{sur} \quad \Gamma_0$$

Conditions de type Neuman, dites libres ou naturelles dans le cas homogène, qui s'écrivent sous la forme générale :

$$\begin{cases} \sigma_{xx}\, n_x + \sigma_{xy}\, n_y = g_x \\ \\ \sigma_{xy}\, n_x + \sigma_{yy}\, n_y = g_y \end{cases}$$

3.1 Formulation variationnelle d'un problème d'élasticité linéaire. Principe des travaux virtuels

La formulation variationnelle s'obtient comme dans le cas du laplacien par multiplication par une fonction test $\mathbf{w}$, intégration et utilisation de la formule de Green. La difficulté supplémentaire dans ce cas provient du fait que l'inconnue est un vecteur. On devra donc multiplier scalairement par une fonction test vectorielle $\mathbf{w}$. Notons w_x et w_y ses deux composantes. On obtient tout d'abord

$$\iint_\Omega \big[\big(\frac{\partial}{\partial x}\sigma_{xx} + \frac{\partial}{\partial y}\sigma_{xy}\big)\, w_x \;+ \big(\frac{\partial}{\partial x}\sigma_{xy} + \frac{\partial}{\partial y}\sigma_{yy}\big)\, w_y\big]\, dxdy \;+ \iint_\Omega \mathbf{f}\,\mathbf{w} dxdy = 0$$

On utilise la formule de Green, ce qui donne

$$\iint_\Omega \big[\, \sigma_{xx}\frac{\partial}{\partial x}w_x + \sigma_{xy}\frac{\partial}{\partial y}w_x \;+ \sigma_{xy}\frac{\partial}{\partial x}w_y + \sigma_{yy}\frac{\partial}{\partial y}w_y\,\big]\, dxdy$$

$$-\int_\Gamma \big[\sigma_{xx}w_x n_x + \sigma_{xy}w_x n_y \;+ \sigma_{xy}w_y n_x + \sigma_{yy}w_y n_y\big]\, d\gamma = \iint_\Omega \mathbf{f}\,\mathbf{w} dxdy$$

Les conditions aux limites s'intègrent dans le terme de bord. Donc avec des conditions de Dirichlet homogènes sur Γ_0 et un effort $\mathbf{g}$ s'exerçant sur Γ_1, on obtient :

$$\iint_\Omega \big[\, \sigma_{xx}\frac{\partial}{\partial x}w_x + \sigma_{xy}\,(\frac{\partial}{\partial y}w_x \;+ \frac{\partial}{\partial x}w_y) + \sigma_{yy}\frac{\partial}{\partial y}w_y\,\big]\, dxdy$$
$$= \int_{\Gamma_1} \mathbf{g}\,\mathbf{w}\, d\gamma \;+ \iint_\Omega \mathbf{f}\,\mathbf{w}\, dxdy$$

pour toute fonction vectorielle $\mathbf{w}$ de composantes dans l'espace de Hilbert V des fonctions de $H^1(\Omega)$, nulles sur Γ_0.

On obtient ainsi la formulation variationnelle de ce problème en contrainte plane

$$\begin{cases} \text{Trouver la fonction vectorielle } \mathbf{u} \text{ appartenant à } V^2 \text{ telle que :} \\ \iint_\Omega \sigma(\mathbf{u})\, \epsilon(\mathbf{w})\, dxdy = \int_{\Gamma_1} \mathbf{g}\,\mathbf{w}\, d\gamma \;+ \iint_\Omega \mathbf{f}\,\mathbf{w}\, dxdy \quad \forall\, \mathbf{w} \in V^2 \end{cases}$$

Ceci s'interprète également comme l'expression du principe des travaux virtuels et peut s'écrire, en utilisant les matrices introduites plus haut sous la forme :

$$\begin{cases} \text{Trouver la fonction vectorielle } \mathbf{u} \text{ appartenant à } V^2 \text{ telle que :} \\ \iint_\Omega \mathbf{D}B\mathbf{u}\,.\,B\mathbf{w}\, dxdy = \int_{\Gamma_1} \mathbf{g}\,\mathbf{w}\, d\gamma \;+ \iint_\Omega \mathbf{f}\,\mathbf{w}\, dxdy \quad \forall\, \mathbf{w} \in V^2 \end{cases}$$

L'existence et l'unicité de la solution de ce problème dans l'espace de Hilbert V^2 s'obtient classiquement par le théorème de Lax-Milgram. L'ellipticité de la forme bilinéaire

$$\mathbf{u}, \mathbf{w} \longrightarrow a(\mathbf{u}, \mathbf{w}) = \iint_\Omega \sigma(\mathbf{u})\, \epsilon(\mathbf{w})\, dxdy$$

dans V^2 résulte de l'inégalité de Korn qui nécessite que des conditions de type Dirichlet soient imposées sur une partie Γ_0 de mesure non-nulle. En l'absence de fixation sur une partie de la frontière, il existe une infinité de solutions à un déplacement rigide près.

La solution du problème variationnel réalise le minimum dans l'espace V^2 de l'énergie potentielle :

$$J(\mathbf{v}) = \frac{1}{2}\iint_\Omega \sigma(\mathbf{v})\, \epsilon(\mathbf{v})\, dxdy - \int_{\Gamma_1} \mathbf{g}\, \mathbf{v}\, d\gamma \; - \iint_\Omega \mathbf{f}\, \mathbf{v}\, dxdy$$

3.2 Approximation par éléments finis P1

On suppose tout d'abord le domaine Ω maillé en éléments finis triangulaires droits. On écrit le problème approché dans l'espace V_h^2 des vecteurs à composantes continues, affines par éléments triangulaires et nulles sur la partie Γ_0 de la frontière. On est ainsi assuré de l'inclusion $V_h^2 \subset V^2$, donc la méthode d'éléments finis est dite conforme. Soit $\mathbf{I}$ l'ensemble des indices des noeuds du maillage correspondant à une valeur inconnue de la solution $\mathbf{u}$, et $N_\mathbf{I}$ le cardinal de $\mathbf{I}$, nombre de noeuds appartenant à $\mathbf{I}$, l'espace V_h^2 sera de dimension $2N_\mathbf{I}$. On choisit comme base de V_h^2, selon la technique de Lagrange, l'ensemble des vecteurs :

$$\begin{pmatrix} w_i \\ 0 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 0 \\ w_i \end{pmatrix} \quad \text{pour } i = 1, ...N_\mathbf{I}$$

où les w_i sont les fonctions scalaires de base de Lagrange de l'espace des fonctions continues affines par triangles déjà rencontrées dans le cas de la discrétisation du laplacien.

Les composantes de la solution approchée s'écriront comme combinaisons linéaires des fonctions de base w_i :

$$u_h(x, y) = \sum_{j \in \mathbf{I}} u_j\, w_j(x, y)$$

$$v_h(x, y) = \sum_{j \in \mathbf{I}} v_j\, w_j(x, y)$$

En utilisant l'écriture matricielle de la formulation variationnelle, on obtient alors le système linéaire suivant :

$$\left\{\begin{array}{l}
\text{Trouver les composantes } u_j \text{ et } v_j \text{ pour } j \in \mathbf{I} \text{ telles que :} \\
\displaystyle\iint_\Omega \sum_{j\in\mathbf{I}} \mathbf{D}B \begin{pmatrix} w_j & 0 \\ 0 & w_j \end{pmatrix} \begin{pmatrix} u_j \\ v_j \end{pmatrix} . B \begin{pmatrix} w_i \\ 0 \end{pmatrix} dxdy = \\
\displaystyle\qquad \int_{\Gamma_1} \begin{pmatrix} g_x \\ g_y \end{pmatrix} \begin{pmatrix} w_i \\ 0 \end{pmatrix} d\gamma + \iint_\Omega \begin{pmatrix} f_x \\ f_y \end{pmatrix} \begin{pmatrix} w_i \\ 0 \end{pmatrix} dxdy \quad \forall\, i \in \mathbf{I} \\
\displaystyle\iint_\Omega \sum_{j\in\mathbf{I}} \mathbf{D}B \begin{pmatrix} w_j & 0 \\ 0 & w_j \end{pmatrix} \begin{pmatrix} u_j \\ v_j \end{pmatrix} . B \begin{pmatrix} 0 \\ w_i \end{pmatrix} dxdy = \\
\displaystyle\qquad \int_{\Gamma_1} \begin{pmatrix} g_x \\ g_y \end{pmatrix} \begin{pmatrix} 0 \\ w_i \end{pmatrix} d\gamma + \iint_\Omega \begin{pmatrix} f_x \\ f_y \end{pmatrix} \begin{pmatrix} 0 \\ w_i \end{pmatrix} dxdy \quad \forall\, i \in \mathbf{I}
\end{array}\right.$$

Comme il y a 2 inconnues u_i et v_i par noeud du maillage, le système ci-dessus est un système linéaire de $2N_{\mathbf{I}}$ équations à $2N_{\mathbf{I}}$ inconnues.

Numérotons les inconnues dans l'ordre $u_1, v_1, u_2, v_2, ...u_i, v_i, ...u_{N_{\mathbf{I}}}, v_{N_{\mathbf{I}}}$. Les coefficients de la matrice de raideur et du second membre se calculent alors selon :

$$\begin{pmatrix} a_{2i-1,2j-1} & a_{2i-1,2j} \\ a_{2i,2j-1} & a_{2i,2j} \end{pmatrix} = \iint_\Omega \mathbf{D}B \begin{pmatrix} w_j & 0 \\ 0 & w_j \end{pmatrix} . B \begin{pmatrix} w_i & 0 \\ 0 & w_i \end{pmatrix} dxdy$$

et

$$\begin{pmatrix} b_{2i-1} \\ b_{2i} \end{pmatrix} = \int_{\Gamma_1} \begin{pmatrix} w_i & 0 \\ 0 & w_i \end{pmatrix} \begin{pmatrix} g_x \\ g_y \end{pmatrix} d\gamma + \iint_\Omega \begin{pmatrix} w_i & 0 \\ 0 & w_i \end{pmatrix} \begin{pmatrix} f_x \\ f_y \end{pmatrix} dxdy$$

3.3 Calculs des matrices et second-membre élémentaires P1

Un élément fini P1 comporte 3 noeuds qui sont les sommets du triangle T. En chaque noeud, nous avons 2 composantes de l'inconnue, où 2 fonctions scalaires de base non nulles. Il y a donc 6 degrés de liberté par triangles. La matrice de raideur élémentaire est donc une matrice 6×6 et le second membre élémentaire un vecteur à 6 composantes. Les restrictions des fonctions de base vectorielles non nulles dans l'élément T s'expriment en fonction des 3 coordonnées barycentriques selon

$$\begin{pmatrix} \lambda_i \\ 0 \end{pmatrix} \text{ et } \begin{pmatrix} 0 \\ \lambda_i \end{pmatrix} \qquad \text{pour } i = 1, 2, 3$$

Notons $\tilde{B}$ la matrice

$$B \begin{pmatrix} \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 & 0 \\ 0 & \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 \end{pmatrix} = \begin{pmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 & 0 \\ 0 & \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 \end{pmatrix}$$

Les λ_i étant des fonctions affines, la matrice $\tilde{B}$ est une matrice à coefficients constants que l'on détermine facilement à l'aide des gradients déjà calculés des coordonnées barycentriques. On obtient :

$$\tilde{B} = \frac{1}{2\,aire(T)} \begin{pmatrix} y_2 - y_3 & 0 & y_3 - y_1 & 0 & y_1 - y_2 & 0 \\ 0 & x_3 - x_2 & 0 & x_1 - x_3 & 0 & x_2 - x_1 \\ x_3 - x_2 & y_2 - y_3 & x_1 - x_3 & y_3 - y_1 & x_2 - x_1 & y_1 - y_2 \end{pmatrix}$$

Le calcul complet de la matrice élémentaire se poursuit sans difficulté. Les intégrations portant sur des fonctions constantes, il suffit de multiplier l'intégrande par l'aire du triangle. D'où le résultat : la matrice élémentaire de raideur K_T vaut

$$K_T = aire(T)\, \tilde{B}^T\, \mathbf{D}\, \tilde{B}$$

soit

$$\frac{1}{4\,aire(T)} \begin{pmatrix} y_2 - y_3 & 0 & x_3 - x_2 \\ 0 & x_3 - x_2 & y_2 - y_3 \\ y_3 - y_1 & 0 & x_1 - x_3 \\ 0 & x_1 - x_3 & y_3 - y_1 \\ y_1 - y_2 & 0 & x_2 - x_1 \\ 0 & x_2 - x_1 & y_1 - y_2 \end{pmatrix} \mathbf{D} \begin{pmatrix} y_2 - y_3 & 0 & y_3 - y_1 & 0 & y_1 - y_2 & 0 \\ 0 & x_3 - x_2 & 0 & x_1 - x_3 & 0 & x_2 - x_1 \\ x_3 - x_2 & y_2 - y3 & x_1 - x_3 & y_3 - y_1 & x_2 - x_1 & y_1 - y_2 \end{pmatrix}$$

Le second membre élémentaire s'obtient selon le même principe. Notons S le pourtour du triangle T, on calcule ses 6 composantes par :

$$\int_{\Gamma_1 \cap S} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \\ \lambda_2 & 0 \\ 0 & \lambda_2 \\ \lambda_3 & 0 \\ 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} g_x \\ g_y \end{pmatrix} d\gamma + \iint_T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \\ \lambda_2 & 0 \\ 0 & \lambda_2 \\ \lambda_3 & 0 \\ 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} f_x \\ f_y \end{pmatrix} dxdy$$

Si les fonctions $\mathbf{g}$ et $\mathbf{f}$ sont données par leurs valeurs aux noeuds du maillage, on trouve :

$$\int_{\Gamma_1\cap S} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \\ \lambda_2 & 0 \\ 0 & \lambda_2 \\ \lambda_3 & 0 \\ 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 & 0 \\ 0 & \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} g_{x1} \\ g_{y1} \\ g_{x2} \\ g_{y2} \\ g_{x3} \\ g_{y3} \end{pmatrix} d\gamma +$$

$$\iint_T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \\ \lambda_2 & 0 \\ 0 & \lambda_2 \\ \lambda_3 & 0 \\ 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 & 0 \\ 0 & \lambda_1 & 0 & \lambda_2 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{pmatrix} dxdy$$

Soit

$$\int_{\Gamma_1\cap S} \begin{pmatrix} \lambda_1^2 & 0 & \lambda_2\lambda_1 & 0 & \lambda_3\lambda_1 & 0 \\ 0 & \lambda_1^2 & 0 & \lambda_2\lambda_1 & 0 & \lambda_3\lambda_1 \\ \lambda_1\lambda_2 & 0 & \lambda_2^2 & 0 & \lambda_3\lambda_2 & 0 \\ 0 & \lambda_1\lambda_2 & 0 & \lambda_2^2 & 0 & \lambda_3\lambda_2 \\ \lambda_1\lambda_3 & 0 & \lambda_2\lambda_3 & 0 & \lambda_3^2 & 0 \\ 0 & \lambda_1\lambda_3 & 0 & \lambda_2\lambda_3 & 0 & \lambda_3^2 \end{pmatrix} \begin{pmatrix} g_{x1} \\ g_{y1} \\ g_{x2} \\ g_{y2} \\ g_{x3} \\ g_{y3} \end{pmatrix} d\gamma +$$

$$\iint_T \begin{pmatrix} \lambda_1^2 & 0 & \lambda_2\lambda_1 & 0 & \lambda_3\lambda_1 & 0 \\ 0 & \lambda_1^2 & 0 & \lambda_2\lambda_1 & 0 & \lambda_3\lambda_1 \\ \lambda_1\lambda_2 & 0 & \lambda_2^2 & 0 & \lambda_3\lambda_2 & 0 \\ 0 & \lambda_1\lambda_2 & 0 & \lambda_2^2 & 0 & \lambda_3\lambda_2 \\ \lambda_1\lambda_3 & 0 & \lambda_2\lambda_3 & 0 & \lambda_3^2 & 0 \\ 0 & \lambda_1\lambda_3 & 0 & \lambda_2\lambda_3 & 0 & \lambda_3^2 \end{pmatrix} \begin{pmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{pmatrix} dxdy$$

Il ne reste plus pour obtenir le résultat qu'à intégrer. Si $\Gamma_1 \cap S$ est le côté A_1A_2 de T, et en calculant exactement les intégrales, le premier terme donne :

$$\frac{Longueur(A_1A_2)}{6} \begin{pmatrix} 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} g_{x1} \\ g_{y1} \\ g_{x2} \\ g_{y2} \\ g_{x3} \\ g_{y3} \end{pmatrix}$$

Si l'on fait un calcul approché par la formule des trapèzes sur l'arête A_1A_2, on trouve :

$$\frac{Longueur(A_1A_2)}{2} \begin{pmatrix} g_{x1} \\ g_{y1} \\ g_{x2} \\ g_{y2} \\ 0 \\ 0 \end{pmatrix}$$

Pour le deuxième terme (surfacique), dans le cas d'un calcul exact, on obtient :

$$\frac{Aire(T)}{12}\begin{pmatrix} 2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{pmatrix}\begin{pmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{pmatrix}$$

Si on utilise la formule d'intégration approchée exacte sur $P1$:

$$\int_T F(x,y)dxdy \approx \frac{Aire(T)}{3}[F(A_1) + F(A_2) + F(A_3)]$$

On obtient une matrice de masse condensée et le second membre surfacique devient :

$$\frac{Aire(T)}{3}\begin{pmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{pmatrix}$$

Comme dans le cadre du problème de Poisson présenté au chapitre précédent, notre approche nécessite uniquement des calculs de solutions élémentaires pour des valeurs différentes du second membre.

Ces développements aboutissent à l'obtention d'un système linéaire d'équations d'état avec les variables d'optimisation présentes au second-membre. Le plus important est une dépendance linéaire entre les variables d'optimisation et la variable d'état, solution de ce système. Dans la section suivante, nous donnons un exemple de tels systèmes à travers l'utilisation du code `Cast3M`.

3.4 Exemple de calcul avec Cast3M

Nous allons montrer un exemple de calcul de déplacement d'une structure élastique avec le logiciel industriel `Cast3M` [28] décrit en annexe A.

On considère une dalle carrée de 10 mètre de longueur et de 1 mètre d'épaisseur en béton à corps creux .

Nous considérons une configuration complexe pour mettre en évidence les capacités de la méthode d'optimisation qui sera présentée dans la suite. En particulier,

la géométrie des objets envisagés est très complexe avec la présence de corps creux ellipsoïdaux tels que discutés dans le chapitre introductif. La figure 3.4 montre une coupe de cette géométrie. Ainsi, pour être opérationnelle, une méthode d'optimisation doit être capable de manipuler ces objets complexes.

Dans la suite, les caractéristiques mécaniques du béton considéré sont de 30 GPa s'agissant du module de Young et de 0.2 pour le coefficient de Poisson. La dalle est supportée par des appuis ponctuels aux quatre coins modélisé par des conditions aux limites de Dirichlet homogènes. Les forces volumiques sont supposées négligeables.

Pour illustrer notre propos, nous allons créer une configuration synthétique en appliquant $N = 5$ charges ponctuelles réparties arbitrairement. On calcule ensuite les déplacements associés $U_{i=1,...,N}$ obtenus après application des forces f_i une à une. Le déplacement total est le cumul de ces déplacements :

$$U_{total} = \sum_{i=1}^{N} U_i = \sum_{i=1}^{N} \frac{U_i}{f_i} f_i = \sum_{i=1}^{N} f_i u_i, \quad u_i = \frac{U_i}{f_i}.$$

Par la suite, U_{total} nous servira de déplacement cible dans les problèmes d'optimisation où le vecteur des variables d'optimisation $\mathbf{x}$ sera donné par $\mathbf{x} = (f_1, ..., f_N)$.

Nous avons utilisé le logiciel `Gmsh` [15] pour la génération de maillage tétraédrique, ensuite les calculs de structure ont été effectué avec `Cast3M`.

Les figures 3.4,3.4, 3.4,3.4, 3.4 et 3.4 montrent les déplacements U_i et le déplacement total $U_{total} = \sum_{i=1}^{N} U_i$.

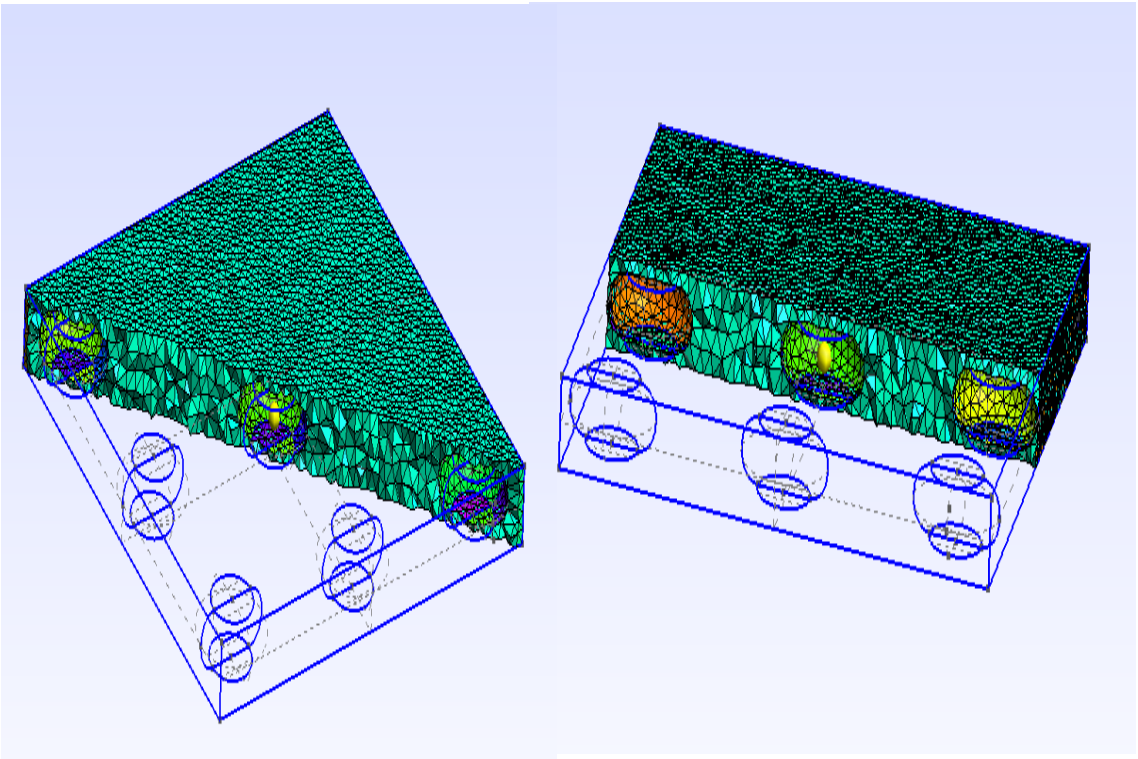

FIGURE 3.1 – Une coupe de maillage.

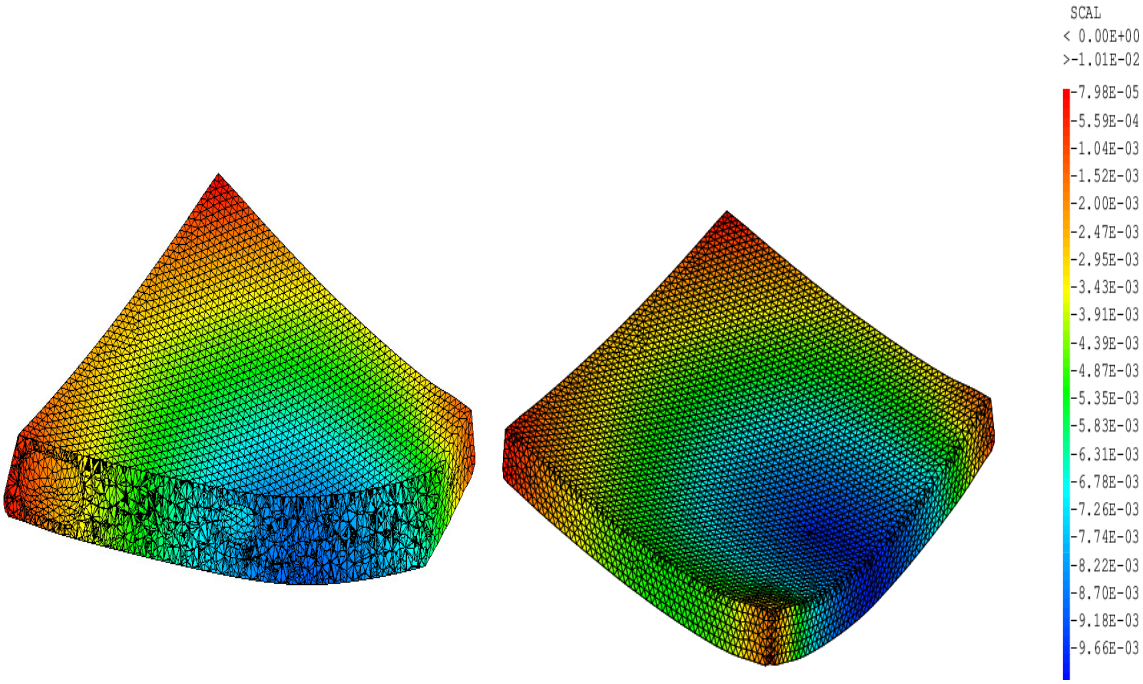

FIGURE 3.2 – Déplacement pour $f_1 = 5.510^5 N$.

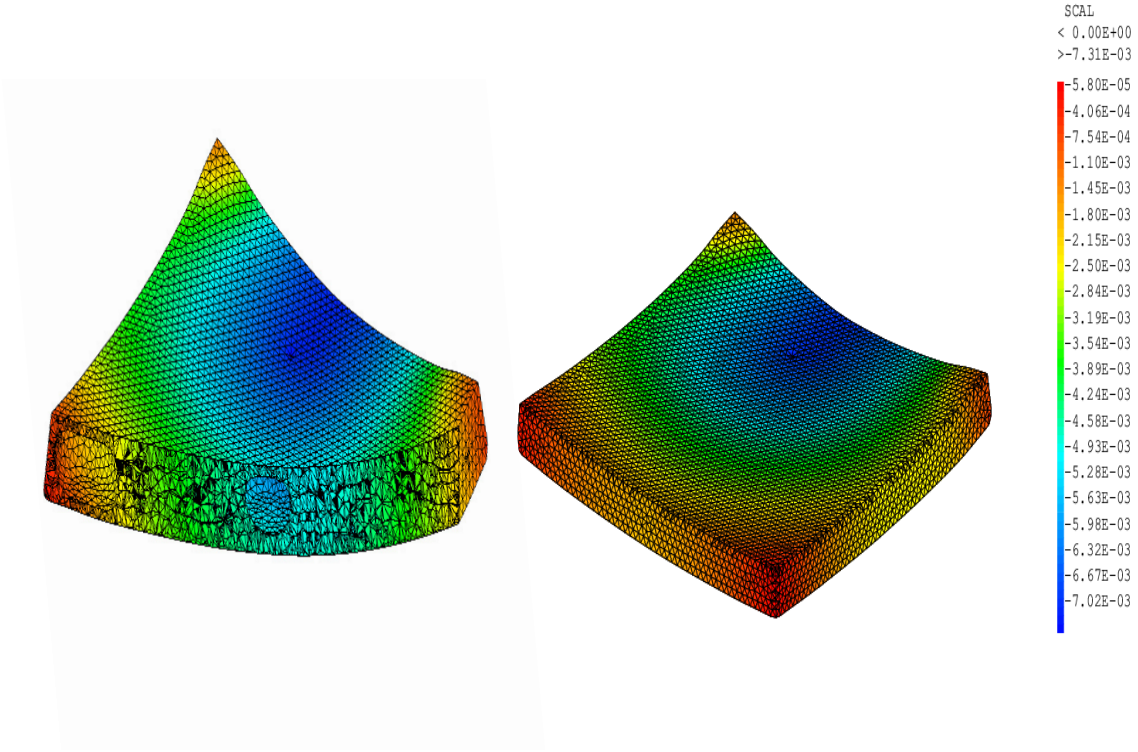

FIGURE 3.3 – Déplacement pour $f_2 = 4.10^6 N$.

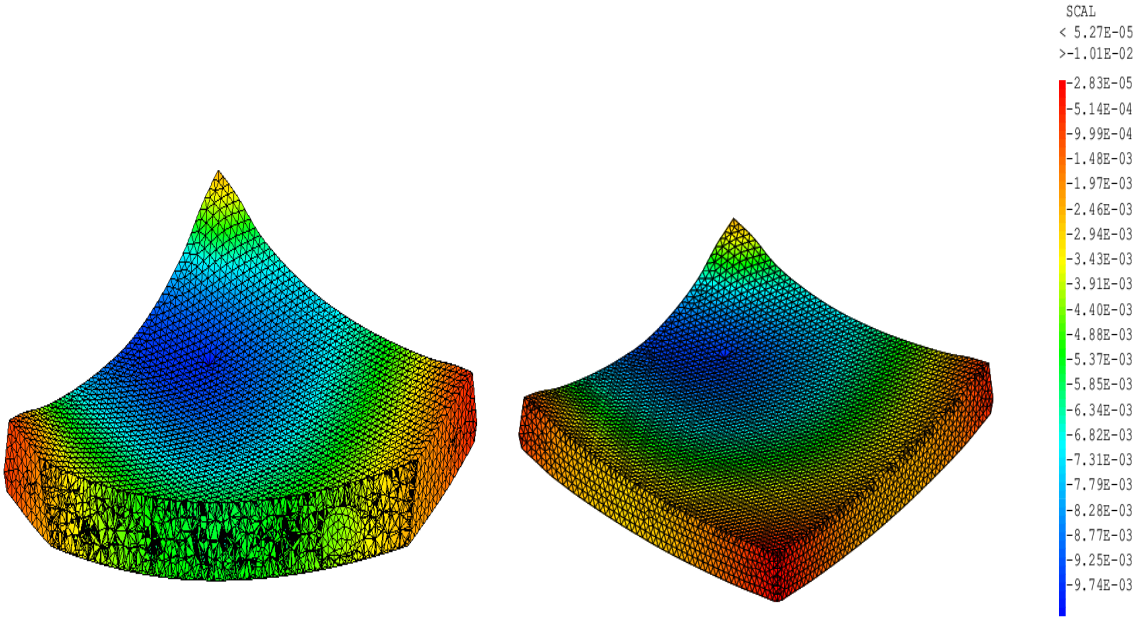

FIGURE 3.4 – Déplacement pour $f_3 = 6.10^6 N$.

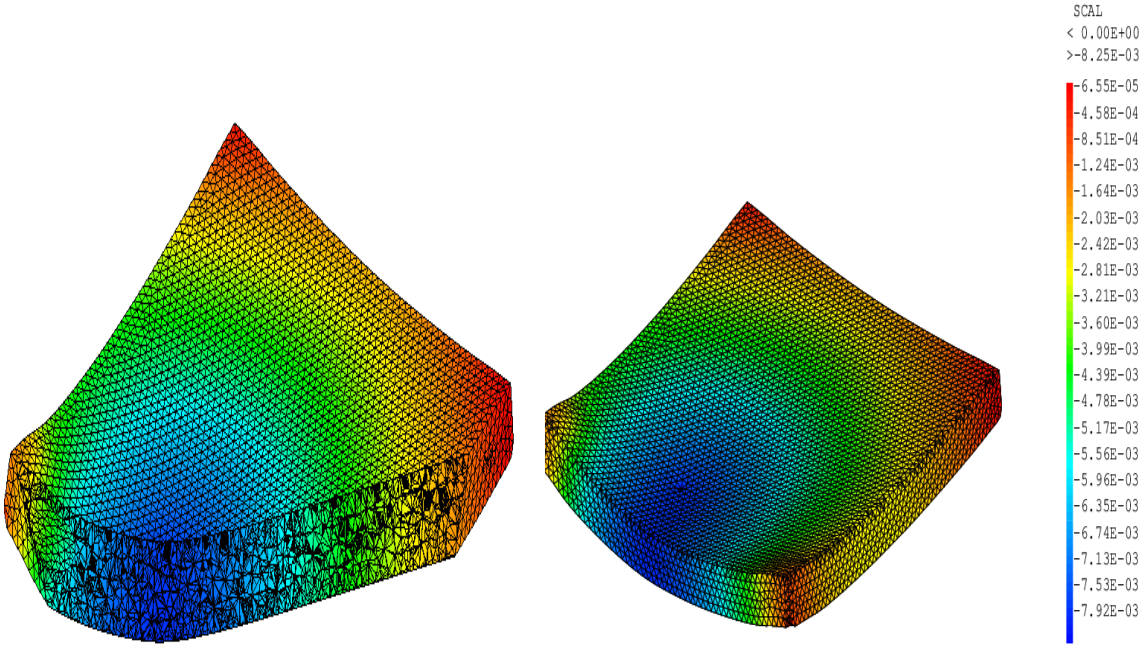

FIGURE 3.5 – Déplacement pour $f_4 = 4.510^6 N$.

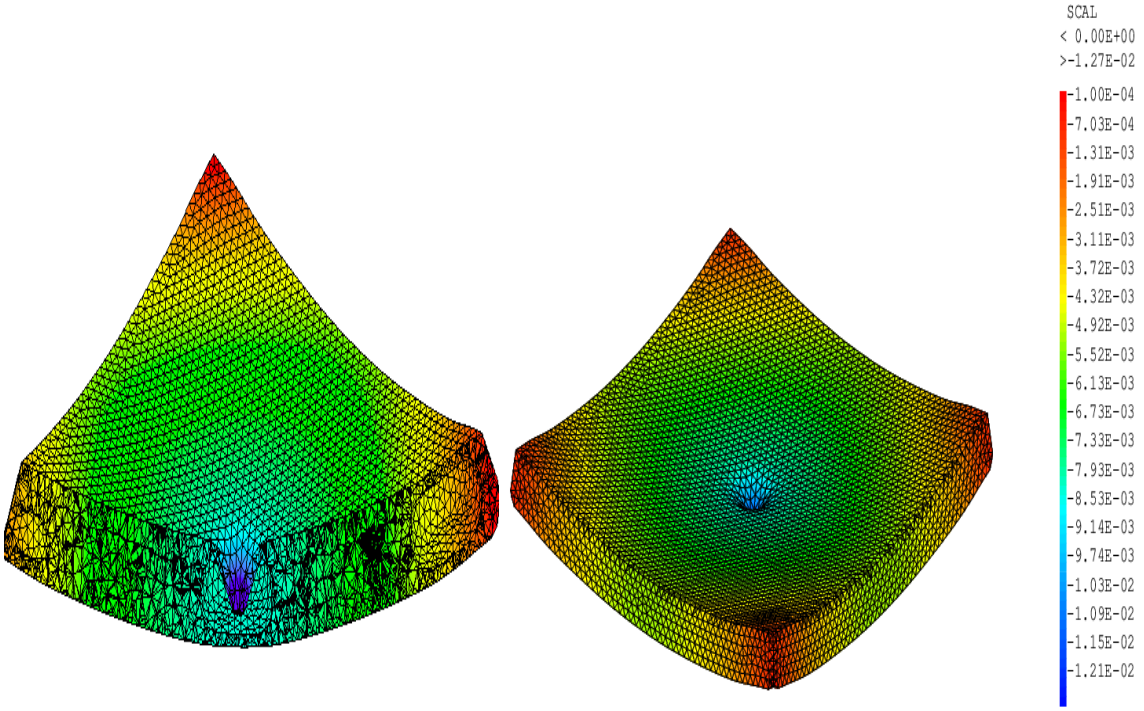

FIGURE 3.6 – Déplacement pour $f_5 = 5.10^6 N$.

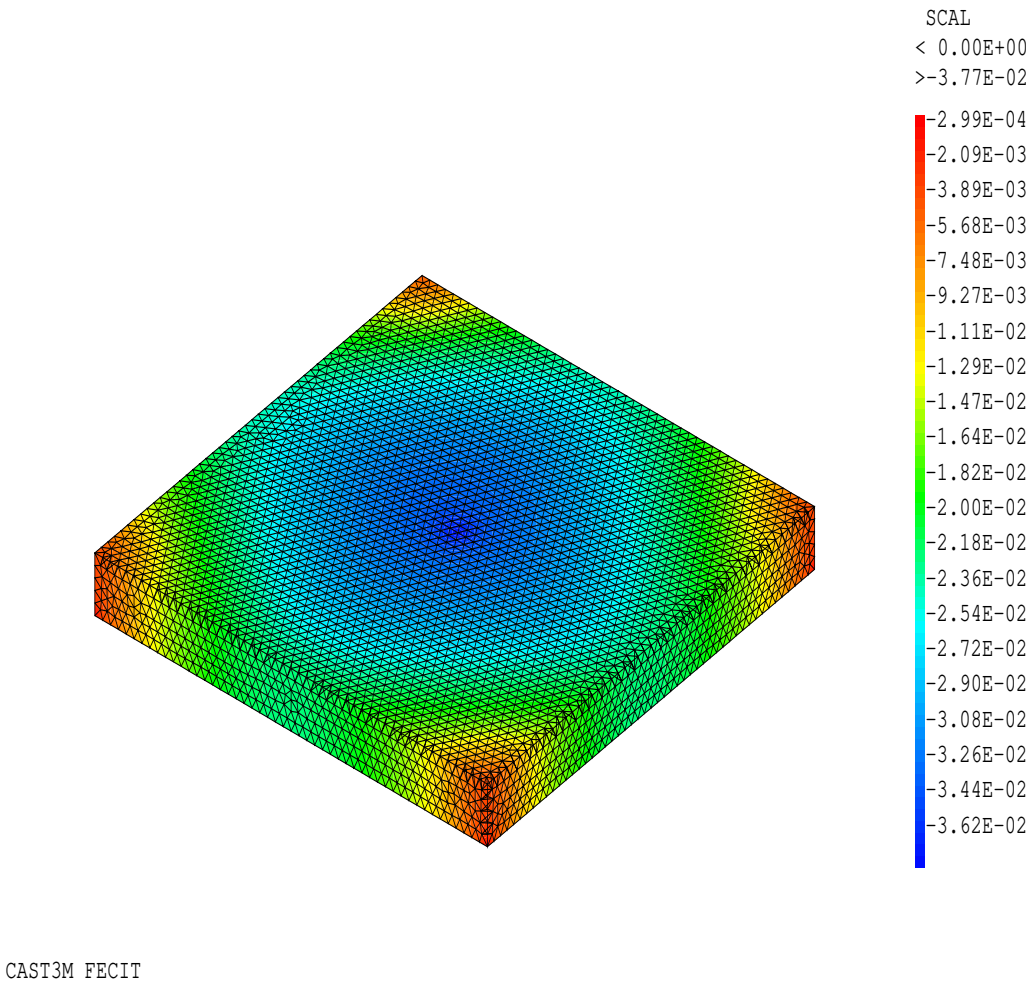

FIGURE 3.7 – Déplacement U_{total}. Ce déplacement nous servira de cible dans les problèmes inverses présentés plus loin.

Chapitre 4

Optimisation

L'optimisation tient un rôle central dans cette thèse. Elle intervient à deux niveaux. Tout d'abord pour la solution de l'équation d'état (membrane ou élasticité) et ensuite pour l'identification de la répartition du chargement optimal. Les deux problèmes sont de même nature et font intervenir l'optimisation d'une fonctionnelle quadratique. Nous allons décrire succinctement le cadre théorique partagé par cet ensemble.

Ainsi, ce chapitre aborde les points suivants :

— Définition du problème de minimisation et rappel des résultats de base d'optimisation sans contraintes et de moindres carrés.
— Contrôle de la déformation d'une membrane par optimisation de la répartition des charges mécaniques.
— Discussion sur les types de supports des charges élémentaires et possibilité d'optimisation de ces supports.
— Illustration de notre propos en utilisant le logiciel `Freefem` [18].

Ainsi nous commençons par un rappel succinct des résultats théoriques. Les théorèmes classiques d'existence, d'unicité et de convergence des méthodes de résolution supposent que la fonctionnelle coût à minimiser et le domaine admissible soient convexes. Ce contexte favorable n'est pas toujours celui des applications où l'on pourra au mieux parler d'une convexité locale.

Les méthodes déterministes de résolution nécessitent souvent le calcul du gradient de la fonction coût à minimiser. Ceci est impossible si la fonctionnelle est non-différentiable. Le calcul du gradient reste toujours un point délicat.

Le souci de simplicité de mise en oeuvre, l'utilisation croissante d'outils boîte-noire, ainsi que la nécessité d'être parfois réalisable expérimentalement, font que les différences finies restent encore la méthode la plus utilisée pour l'évaluation des gradients, et ceci malgré leur forte sensibilité au choix des incréments et malgré leur coût en stockage (taille de l'espace de contrôle).

L'approche présentée dans cette thèse permet de contourner ces difficultés et surtout de réduire la complexité des problèmes d'optimisation.

4.1 Définition du Problème

Considérons une boucle de simulation liant une paramétrisation $\mathbf{x}$ à une fonctionnelle positive J, mesurant un coût ou une contre-performance à minimiser, via la solution d'une équation d'état :

$$J(\mathbf{x}) : \mathbf{x} \to U(\mathbf{x}) \to J(\mathbf{x}, U(\mathbf{x})),$$

où U représente les variables intermédiaires où dépendantes. L'état U est solution de l'équation d'état et représente, en général, la quantité la plus complexe à évaluer. Nous avons vu des exemples d'équations d'état dans les chapitres précédents avec le problème de membrane et le système d'élasticité linéaire en trois dimensions.

Le problème de minimisation pour J sur un espace admissible pour la paramétrisation $O_{ad} \subset \mathbb{R}^N$ s'écrit :

$$\inf\{J(\mathbf{x}, U(\mathbf{x})), \quad \mathbf{x} \in O_{ad}\}. \tag{4.1}$$

Nous commencerons par un rappel des résultats théoriques classiques d'existence d'unicité et de convergence des méthodes de résolution du problème (6.5).

4.2 Résultats de base de l'optimisation sans contraintes

Nous avons déjà utilisé la solution d'un problème d'optimisation pour un problème de minimisation de l'énergie lors du calcul par la méthode des éléments finis de la solution du problème d'élasticité. Cette approche s'appuie sur le résultat général suivant :

Soit J une fonction donnée d'un espace de Hilbert H dans $\mathbb{R}$, couramment nommée fonctionnelle sur H en optimisation.

Sous les hypothèses :

1. J est strictement convexe et dérivable sur H.
2. J est coercive sur H : $(\lim_{\|\mathbf{x}\| \to \infty} J(\mathbf{x}) = +\infty)$,

alors J admet un minimum unique dans H.

Ce minimum est solution du problème

$$J'(\mathbf{x}).d = \lim_{\theta \to 0} \frac{J(\mathbf{x} + \theta d) - J(\mathbf{x})}{\theta} = 0, \quad \forall d \in H,$$

où J' est la dérivée de la fonctionnelle J.

En dimension finie, si $H = \mathbb{R}^N$, le minimum est le zéro du gradient ∇J.

4.3 Projection sur un sous-espace fermé

Un exemple fondamental de minimisation est donné par la recherche de la meilleure approximation dans un sous espace F d'un Hilbert H au sens de sa norme. Il s'agit

de trouver l'élément du sous-espace le plus proche, au sens de la norme de l'espace de Hilbert, d'un élément u quelconque du Hilbert. On doit donc minimiser la fonctionnelle strictement convexe $\|u - v\|$ sur F. Les applications sont nombreuses et variées. Exprimée dans une bas orthogonale, la meilleure approximation coïncide avec le développement de Fourier. On retrouve également toutes les techniques d'approximation au sens des moindres carrés. La méthode des éléments finis est une méthode d'approximation au sens des moindres carrés en considérant comme norme, une norme déduite de l'énergie minimisée et comme espace de projection, le sous-espace obtenu par la méthode d'interpolation choisie.

4.4 Équivalence entre résolution d'un système linéaire et minimisation quadratique

Un cas particulier important de fonctionnelle strictement convexe sur un Hilbert est le suivant :

$$J(v) = \frac{1}{2}a(v, v) - l(v) \tag{4.2}$$

avec a forme bilinéaire symétrique continue elliptique dans V et l forme linéaire continue dans V. C'est cette fonctionnelle d'énergie qui est minimisée dans les méthodes variationnelles de résolution d'équations aux dérivées partielles linéaires elliptiques, comme la méthode des éléments finis.

Dans le cas où $V = I\!R^N$, on obtient le résultat suivant que nous utiliserons pour l'identification de la charge optimale.

Si A est une matrice symétrique définie positive, il y a équivalence entre les deux problèmes suivants :

$$(1) \qquad \begin{cases} \text{Trouver } \mathbf{x} \in I\!R^N \quad \text{tel que} \\ \\ A\,\mathbf{x} = b \end{cases} \tag{4.3}$$

$$(2) \qquad \begin{cases} \text{Trouver } \in I\!R^N \quad \text{tel que} \\ \\ J(\mathbf{x}) = \dfrac{1}{2}(A\,\mathbf{x}, \mathbf{x}) - (b, \mathbf{x}) \ \text{ soit minimal} \end{cases} \tag{4.4}$$

Notre problème de recherche de charges optimales fait intervenir un deuxième problème d'optimisation de cette nature via un problème de moindres carrés.

4.5 Rappel sur les moindres carrés

Considérons le problème général d'un système linéaire sur-déterminé, c'est à dire dans lequel il y a plus d'équations que d'inconnues. C'est en particulier le cas dans le calcul de la droite des moindres carrés ou plus généralement de polynômes

d'approximation au sens des moindres carrés. On ne peut pas obtenir exactement l'égalité

$$A\mathbf{x} = b$$

si A est une matrice rectangulaire de N lignes et m colonnes avec $N >> m$. On essaie alors de minimiser l'écart entre les vecteurs $A\mathbf{x}$ et b de $\mathbb{R}^N$ en minimisant la norme euclidienne de leur différence, ou ce qui revient au même le carré de cette norme.

$$\text{Minimiser } J(\mathbf{x}) = \|A\mathbf{x} - b\|^2 = (A\mathbf{x} - b, A\mathbf{x} - b)$$

On utilise les propriétés classiques du produit scalaire $(AU, V) = (U, A^T V)$ pour obtenir :

$$J(\mathbf{x}) = (A^T A\mathbf{x}, \mathbf{x}) - 2(A^T b, \mathbf{x}) + (b, b)$$

la matrice $A^T A$ est une matrice carrée $m \times m$ symétrique définie positive (Les m vecteurs colonnes de A sont supposés indépendants). Nous avons l'équivalence de ce problème de moindres carrés avec la résolution du système linéaire

$$A^T A\mathbf{x} = A^T b$$

On retrouve ainsi le système carré de m équations à m inconnues, dit "système des équations normales".

Dans le problème de répartition de charges optimale, la matrice A est symétrique comme nous le verrons plus bas.

4.6 Répartition de charges

On considère le domaine Ω représentant une section plane d'une structure dont on cherche à contrôler la déformation en optimisation la distribution de charges mécaniques.

On suppose la structure chargée à N endroits et soutenue sur ses bords inférieur Γ_1 et supérieur Γ_2 et libres sur ses bords latéraux Γ_0.
Soit u la déformation, en mètre, dans Ω. La modélisation de ce problème s'écrit

$$\begin{cases} -\Delta u = f & \text{dans } \Omega \\ u = b_1 & \text{sur } \Gamma_1 \\ u = b_2 & \text{sur } \Gamma_2 \\ \dfrac{\partial u}{\partial n} = 0 & \text{sur } \Gamma_0 \end{cases} \tag{4.5}$$

Nous allons déterminer par une méthode de moindres carrés les valeurs à donner aux N charges f_i appliquées sur les rectangles D_i pour obtenir la déformation la plus proche possible de u_{cible} sur Ω.

Il s'agit d'un exemple typique de problème inverse linéaire. En effet, en utilisant le principe de superposition, on peut décomposer la solution inconnue en la combinaison de $N + 1$ solutions.

$$u = u_0 + \sum_i f_i u_i = u_0 + \mathbf{x}^t U, \quad \text{avec} \quad U = (u_1, ..., u_N) \in \mathbb{R}^{M \times N},$$

où M est la taille de discrétisation servant au calcul des solutions homogène u_0 et élémentaires $u_{i=1,...,N}$. A chaque charge f_i est associée un support d'application.

Nous verrons dans le chapitre suivant que si l'adaptation de maillage est utilisé, cette discrétisation sera différente pour chaque solution élémentaire. On introduira alors une discrétisation de référence sur laquelle toutes les solutions élémentaires seront projetées par interpolation.

Ainsi, suivant les notations du problème de minimisation ci-dessus, les paramètres de contrôle sont ici $\mathbf{x} = (f_1, ..., f_N)^t \in \mathbb{R}^N$.

La solution de base u_0 s'obtient par résolution du problème 4.5 dans le cas $f = 0$, donc en l'absence de charge. Les autres solutions élémentaires u_i s'obtiennent respectivement en mettant à un la charge associée sur le support de l'indice correspondant et en annulant les autres ainsi que les conditions aux limites. Par linéarité, la solution globale u obtenue en présence de l'ensemble des charges f_i est bien donnée par la combinaison ci-dessus.

Le problème se ramène au problème de minimisation de la somme des carrés des écarts à u_{cible} des déformations sur Ω.

$$\min_{\mathbf{x}} J(\mathbf{x}) \quad \text{où} \quad J(\mathbf{x})(m^4) = \frac{1}{2}\int_{s\in\Omega} (u_{cible}(s) - u(s,\mathbf{x}))^2 ds. \tag{4.6}$$

Une fois déterminées la solution homogène u_0 et les solutions élémentaires $u_{i=1,...,N}$, les coefficients $f_{i=1,...,N}$ de la solution optimale sont solutions d'un problème de moindres carrés. Le vecteur des valeurs de $\sum_i f_i u_i$ est la projection du vecteur $u_{cible} - u_0$ correspondant. On obtient donc en définitive les charges f_i en résolvant un système linéaire de N équations à N inconnues :

$$A\mathbf{x} = b, \tag{4.7}$$

avec

$$A_{i,j} = \int_{s\in\Omega} u_i(s)\, u_j(s) ds,$$

et

$$b_i = \int_{s\in\Omega} (u_{cible}(s) - u_0(s))\, u_i(s) ds.$$

4.6.1 Supports de charges élémentaires pour le calcul des solutions élémentaires par noyau

Nous avons besoin de calculer les solutions élémentaires pour des termes sources qui peuvent être appliqués sur des supports définis, comme par exemple des supports ponctuels à travers des Dirac. Or, il n'est pas aisé de garantir qu'un point de maillage se trouve au niveau du support de chaque Dirac. Il est plus commode d'introduire une approximation des Dirac par des fonctions Gaussiennes. La même démarche peut être adoptée pour d'autres types de support par l'introduction de fonction régulière à support compact, comme une fonction porte régularisée par exemple. La figure 4.1 montre quelques exemples de fonctions support que nous utilisons pour définir différents types de modes d'application des charges.

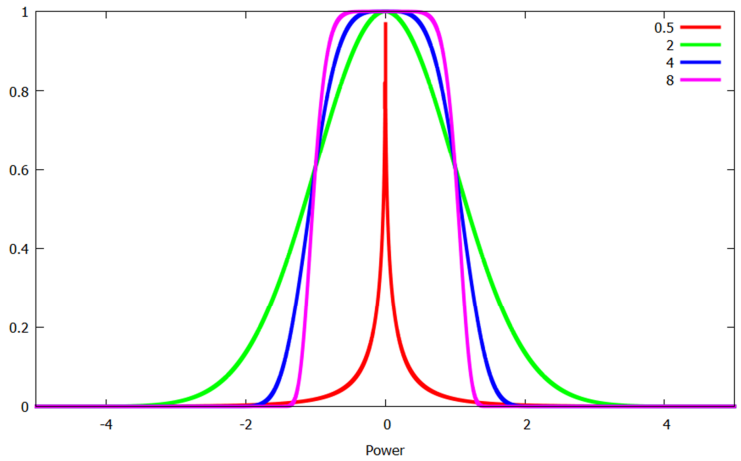

FIGURE 4.1 – Exemple de fonction noyau $\exp(-\beta\|s\|^2)$ pour le support de la charge.

Ainsi, nous introduisons l'approximation suivante pour un terme source f contenant N termes sources élémentaires de la forme :

$$f = \sum_{i=1,\dots,N} f_i \delta(s_i) \sim \sum_{i=1,\dots,N} f_i \exp(-\beta\|s - s_i\|^2). \quad \text{avec} \quad \beta >> 1,$$

Le choix du support de la Gaussienne n'est pas anodin et nous verrons son impact lors de simulations. De plus, les supports des Gaussiennes peuvent être vus comme des paramètres d'optimisation supplémentaires. On pourra associer notre optimisation des charges et une optimisation des supports des charges. Cette optimisation supplémentaire demandera cependant l'utilisation de méthode d'optimisation non-linéaire, comme des méthodes de descente. Une autre difficulté viendra du fait que les solutions élémentaires auront à être recalculées à chaque changement de support. Cette remarque permet d'appréhender la réduction du coût de calcul que la linéarité a permis d'accomplir.

4.6.2 Lien avec les méthodes de descente

La méthode présentée tire avantage de la linéarité du problème direct. En non-linéaire, on est amené à utiliser, par exemple, une méthode de descente pour construire une suite minimisante :

$$\mathbf{x}_0 = \text{donné}, \quad \mathbf{x}^{n+1} = \mathbf{x}^n - \rho \nabla_{\mathbf{x}} J(u_h(\mathbf{x}^n, s)), \quad \rho > 0.$$

Le calcul de $\nabla_{\mathbf{x}} J$ n'est pas aisé et est impossible avec les solveurs boîtes-noires. Par ailleurs, lorsque la dimension de l'espace d'optimisation est grande, on doit utiliser la méthode de l'adjoint dans les cas où l'équation d'état n'est pas auto-adjoint.

La méthode directe présentée dans cette thèse permet de s'affranchir de la résolution par méthode de gradient du problème d'optimisation.

4.7 Exemple numérique avec un logiciel boîte-noire

Nous allons illustrer notre propos en utilisant le logiciel `freefem` [18]. Nous considérons un domaine circulaire et quatre charges réparties arbitrairement. Cette répartition permet de calculer une déformation cible. Nous souhaitons retrouver la distribution de charges donnant la même déformation par la solution de notre problème directe. Ce problème est admissible et nous sommes assurés d'avoir une solution car la déformation cible a été réalisée en utilisant les charges que nous souhaitons retrouver. Ceci n'est pas toujours le cas en pratique et nous ne sommes pas toujours certains que le problème d'optimisation ait une solution. Dans certaines situations, nous pouvons avoir non unicité de la distribution optimale des charges.

Le script ci-dessous détaille les étapes de la construction.

```
// Un exemple de calcul de solutions homogènes et
// élémentaires, assemblage du problème linéaire
// et solution direct du problème d'optimisation
func bc=sin(x*y); int ibc=1;
border Omega(t=0,2*pi){x=10*cos(t);y=10*sin(t);}
mesh Th1=buildmesh(Omega1(10));
fespace Vh1(Th1,P1);
Vh1 u,v,ff;
problem Poisson(u,v)= int2d(Th1)(dx(u)*dx(v) + dy(u)*dy(v))
-int2d(Th1)(ff*v)
+ on(Omega,u=bc*ibc); //Conditions aux limites Dirichlet
////////////// Définition de la solution cible //////////////
////// -Delta u= x1 f1 + x2 f2 + x3 f3 + x4 f4  (4 sources arbitraires)
/////////////////////////////////////////////////////////////
// choix a priori de 4 paramètres d'optimisation
real x1=1, x2=2, x3=3, x4=4;
real supp=10, pow=2;  // parametres de la fonction support

Vh1 udes,
f1=exp(-supp*((x-2.5)^pow+(y)^pow)),
f2=exp(-supp*((x)^pow+(y-2.5)^pow)),
f3=exp(-supp*((x+2.5)^pow+(y)^pow)),
f4=exp(-supp*((x)^pow+(y+2.5)^pow));

ibc=1; ff=x1*f1+x2*f2+x3*f3+x4*f4;
Poisson; udes=u;  // solution cible
///
///Problème inverse = Trouver les charges réalisant u=udes
// au sens des moindres carrés.
///
```

```
/////////////// Solution homogène sans terme source avec conditions aux limites
Vh1 uL;
u=uL; ff=0; ibc=1; Poisson; uL=u;
/////////////// Solution U1
Vh1 u1; ff=f1; u=u1; ibc=0; Poisson; u1=u;
/////////////// Solution U2
Vh1 u2; ff=f2; u=u2; ibc=0; Poisson; u2=u;
/////////////// Solution U3
Vh1 u3; ff=f3; u=u3; ibc=0; Poisson; u3=u;
/////////////// Solution U4
Vh1 u4; ff=f4;  u=u4; ibc=0; Poisson; u4=u;
///////////  MATRICE A & F ///////////
/// Aij= <ui,uj>
real a11=int2d(Th1)(u1*u1);
real a12=int2d(Th1)(u1*u2);
real a13=int2d(Th1)(u1*u3);
real a14=int2d(Th1)(u1*u4);
real a22=int2d(Th1)(u2*u2);
real a23=int2d(Th1)(u2*u3);
real a24=int2d(Th1)(u2*u4);
real a33=int2d(Th1)(u3*u3);
real a34=int2d(Th1)(u3*u4);
real a44=int2d(Th1)(u4*u4);
real F1=int2d(Th1)(u1*(uL-udes));
real F2=int2d(Th1)(u2*(uL-udes));
real F3=int2d(Th1)(u3*(uL-udes));
real F4=int2d(Th1)(u4*(uL-udes));
//////////////////////////////////////
/// Résolution système linéaire
//////////////////////////////////////
real[int,int] A(4,4);
A=[[a11,a12,a13,a14],[a12,a22,a23,a24],[a13,a23,a33,a34],[a14,a24,a34,a44]];
matrix B=A;
real[int] b=[-F1,-F2,-F3,-F4];
real[int] c(b.n);
c=B*b;
set(B,solver=UMFPACK); real[int] d(4);
d=B^-1*b;
//////////////////////////////////////
/// Test a posteriori pour validation solution optimale
//////////////////////////////////////
Vh1 uopt;
u=uopt;
ff=d(0)*f1+d(1)*f2+d(2)*f3+d(3)*f4;
ibc=1;
```

```
Poisson;
uopt=u;
real J0=int2d(Th1)((uL-udes)^2);
real J=int2d(Th1)((uopt-udes)^2);
Vh1 err=udes-uopt;
cout <<"cout initial / final    " << J0<< "  ,  "<< J << endl;
}
```

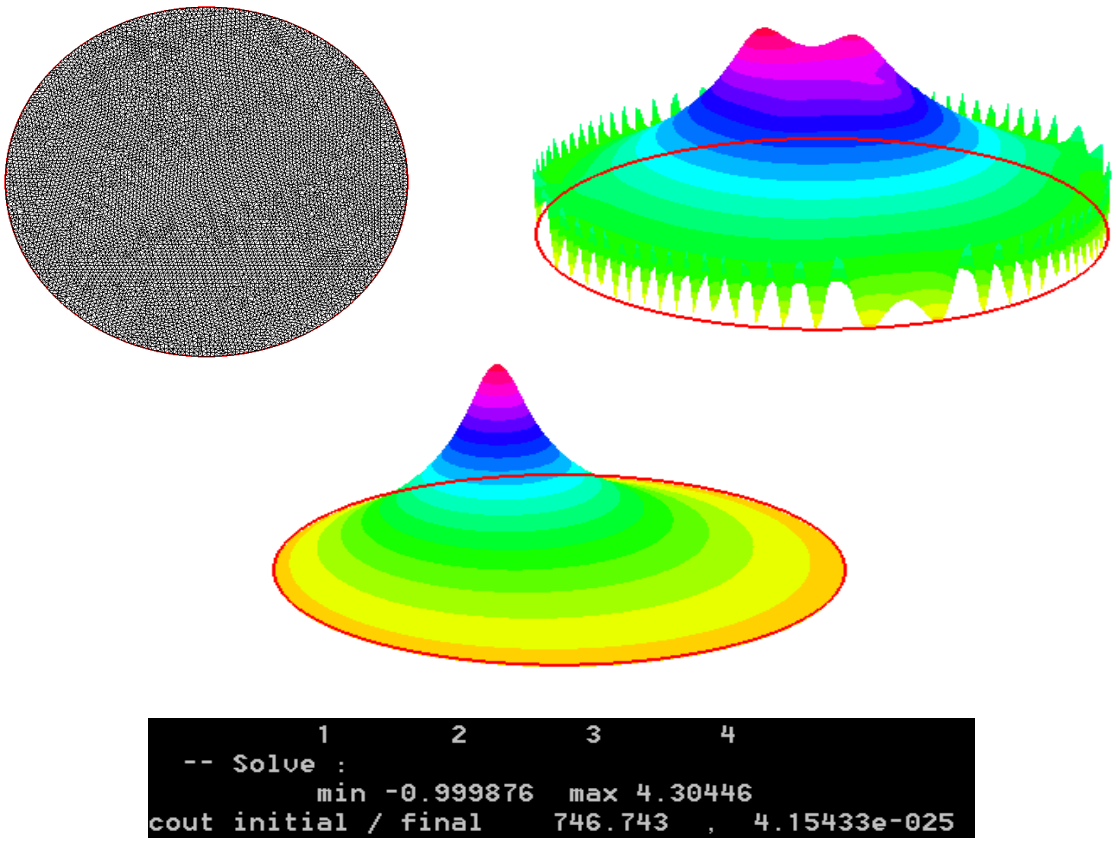

FIGURE 4.2 – Optimisation de distribution de charge par méthode directe sur un domaine circulaire sur maillage uniforme triangulaire. En haut à droite : solution cible et optimale retrouvée. Milieu : une des solutions élémentaires. Le coût initial en absence de charge est de $746m^3$. Le problème inverse est bien résolu avec un coût avec la distribution optimale voisin de 0. La distribution cible était le vecteur $\mathbf{x} = (f_1, f_2, f_3, f_4) = (1, 2, 3, 4)$.

4.7.1 Étude de l'effet de la représentation du support de la charge

Nous allons étudier l'impact de la l'étendu du support de la charge mentionnée précédemment sur la solution du problème d'optimisation. Nous considérons le paramètre β de la Gaussienne $\exp(-\beta\|s - s_i\|^2)$ comme représentation de ce support. La figure 4.3 montre la fonctionnelle $J(\mathbf{x}, \beta)$ pour un échantillonnage en β :

$$J(\mathbf{x}, \beta) = \int_{s\in\Omega} (u_{cible}(s) - u(s, \mathbf{x}, \beta))^2 ds. \tag{4.8}$$

On constate que la solution du problème d'optimisation est sensible à la bonne définition du support, mais que cette sensibilité n'est pas très forte car une valeur de β dans l'intervalle 1.5 à 2.5 donne un niveau de fonctionnelle à l'optimum raisonnable inférieur à $10^{-2}m^3$ à comparer à un niveau de fonctionnelle sans répartition de charge de plus de $700m^3$.

Au delà de β, on peut aussi se poser la question de l'impact de la définition de la puissance dans la fonction noyau, telle que donnée dans la figure 4.1 par exemple.

Pour cela nous allons considérer les paramètre α et β de notre fonction support $\exp(-\beta\|s - s_i\|^\alpha)$. La figure 4.4 montre la fonctionnelle $J(\mathbf{x}, \alpha, \beta)$ pour un échantillonnage en α et β :

$$J(\mathbf{x}, \beta) = \int_{s\in\Omega} (u_{cible}(s) - u(s, \mathbf{x}, \alpha, \beta))^2 ds. \tag{4.9}$$

On constate que, là aussi, la solution du problème d'optimisation est sensible à la bonne définition du support, mais qu'une zone de confiance peut être définie permettant d'assurer un niveau de fonctionnelle à l'optimum en dessous d'un seuil d'acceptabilité de $10^{-2}m^3$. Par exemple, tout couple (α, β) tel que $\alpha > 2$ et $\beta \in [1.5, 2.5]$ donne satisfaction.

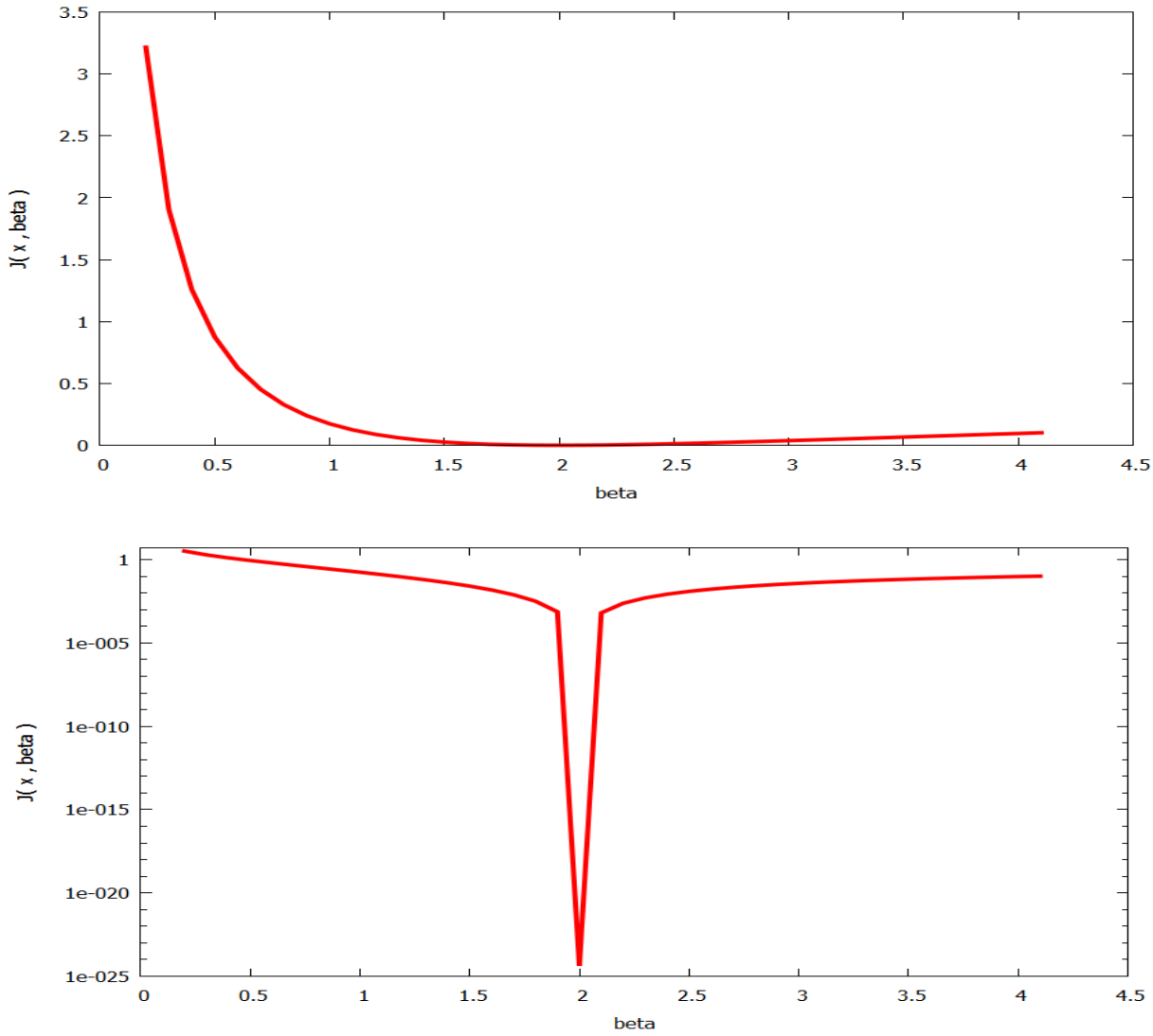

FIGURE 4.3 – Optimisation de distribution de charge par méthode directe sur un domaine circulaire sur maillage uniforme triangulaire. $J(\mathbf{x}, \beta)$ en fonction de β du support de la charge.

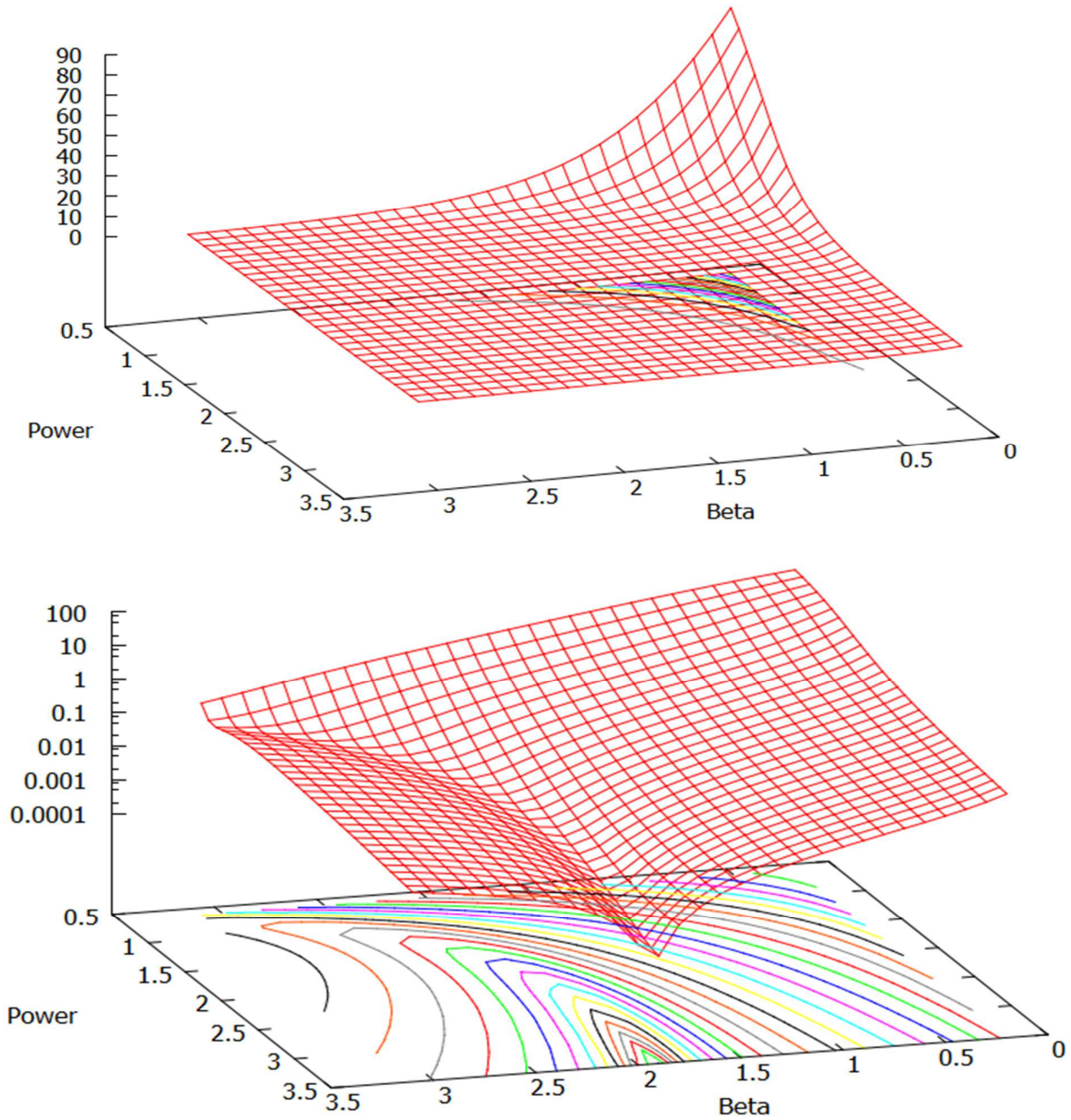

FIGURE 4.4 – Optimisation de distribution de charge par méthode directe sur un domaine circulaire sur maillage uniforme triangulaire. $J(\mathbf{x}, \alpha, \beta)$ en fonction de la puissance du noyau et l'étendu du support de la charge.

4.8 Application de la méthode directe en élasticité linéaire tridimensionnelle

Nous allons illustrer comment la méthode précédente peut être mise en œuvre dans un cadre industriel pour un problème complexe décrit dans la section 3.4. Nous souhaitons retrouver la distributions des charges donnant un déplacement vertical donné $u_{cible} = u_{total}$ illustré par la figure 3.4 et obtenu par application de cinq chargements à des endroits différents du domaine.

La fonctionnelle que nous considérons est :

$$J(\mathbf{x}) \ \ (\text{en } m^5) \ \ = \int_{s\in\Omega} (u_{cible}(s) - u(s,\mathbf{x}))^2 ds. \tag{4.10}$$

Rappelons les étapes qui doivent être suivies dans la méthode d'optimisation directe pour retrouver les chargements donnant un déplacement proche au sens des moindres carrés du déplacement cible avec N étant le nombre de chargements que nous recherchons qui est a priori inconnu.

Nous insistons sur l'utilisation boîte-noire du logiciel de calcul d'élasticité et la séparation des calculs mécaniques, la sauvegarde des solutions élémentaires et la solution du problème d'optimisation. Cette approche étant totalement générique, il est donc aisé de remplacer `Cast3M` par un autre outil.

— Calcul et sauvegarde des déplacements ($U_i,\ i = 1 \cdots N$) et du déplacement total $U_{total} = \sum_{i=1}^{N} U_i$ dans $N+1$ fichiers.
— Sous `Python` [44], on importe les fichiers des déplacements et on calcule les matrices A et b.
— On résout le système linéaire $A\mathbf{x} = b$ avec

$$A_{i,j} = \int_{s\in\Omega} u_i(s)\, u_j(s) ds,$$

et

$$b_i = \int_{s\in\Omega} (u_{cible}(s) - u_0(s))\, u_i(s) ds.$$

On constate que les chargements obtenus correspondent aux cibles attendues. Les valeurs sont données dans dans le tableau 4.1. On constate un niveau d'erreur négligeable.

En pratique, on ne connaît pas la cible de chargements. Il faut vérifier si le déplacement trouvé correspond à la cible.

La combinaison de solutions élémentaires permet ce test sans calcul supplémentaire. Si le support du chargements change, on doit recalculer les solutions élémentaires, sinon on utilise la combinaison linéaire.

Cette approche permet de tester facilement différents nombre de chargements pour voir quel chargement permet de réaliser un déplacement suffisamment proche du déplacement cible.

Pour illustrer ce propos, commençons par N=2 et allons jusqu'à N=4. Les figures 4.5, 4.6, 4.7 montrent ces déplacements correspondants.

f1 (N)	5500000.00001097
f2 (N)	4000000.00001597
f3 (N)	6000000.00000143
f4 (N)	4500000.00001526
f5 (N)	4999999.99996281

TABLE 4.1 – les chargements solution du problème inverse.

En calculant $J(\mathbf{x})$ on constate que l'ajout de nouvelles variables d'optimisation permet la convergence. En particulier, ceci nous permet de décider quand un niveau de précision suffisant prédéfini par l'utilisateur a été atteint.

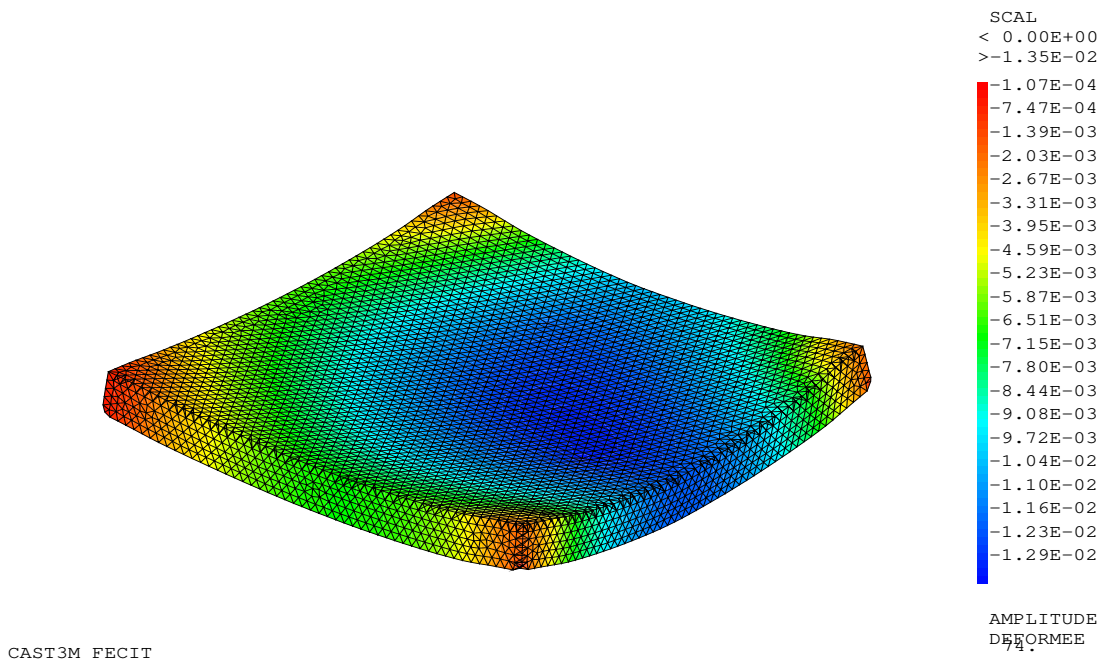

FIGURE 4.5 – Déplacement pour $N = 2$; $J = 3.463m^4$

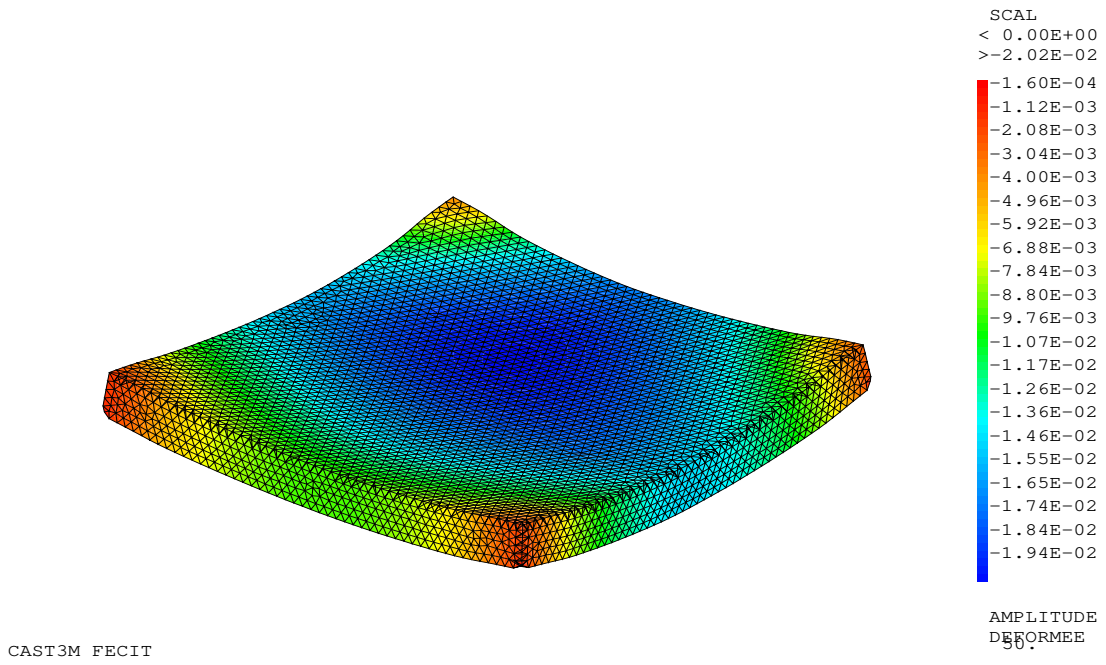

FIGURE 4.6 – Déplacement pour $N = 3$; $J = 1.457m^4$

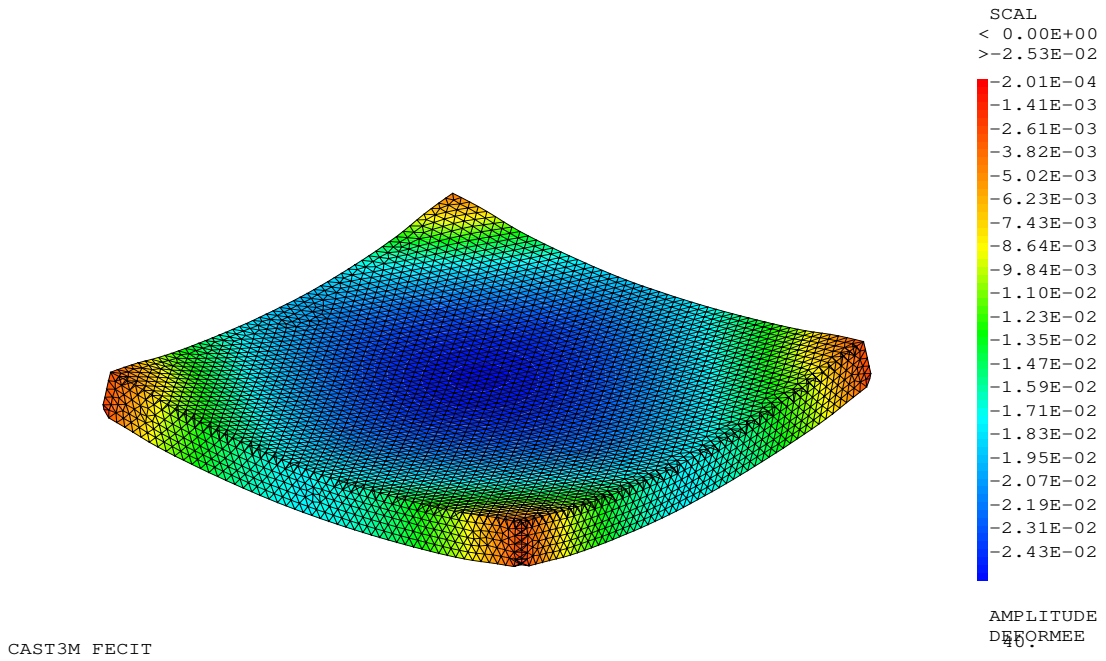

FIGURE 4.7 – Déplacement pour $N = 4$; $J = 0.451m^4$. Ainsi, ce nombre de variable d'optimisation est suffisant si un niveau cible de $1m^4$ avait été prescrit par l'utilisateur.

4.9 Optimisation sous contrainte

Nous rappelons que $\mathbf{x} = (x_1, ..., x_N) = (f_1, ..., f_N)$. Jusqu'ici nous avons considéré des problèmes d'optimisation sans contrainte. Une situation classique est la présence de contrainte sur les variables d'optimisation. Par exemple, la charge totale doit être respectée dans certaines situations.

Il s'agit d'une contrainte d'égalité :

$$\sum_{i=1}^{N} x_i = F_{total},$$

qui peut s'exprimer sous forme matricielle comme :

$$B\mathbf{x} = c,$$

avec B une matrice ligne : $B = (1, ..., 1)$ et $c = F_{total}$.

Nous pouvons aussi avoir besoin de prendre en compte des contraintes d'inégalité. Par exemple, pour des raisons de fiabilité ou de contrainte de construction, les charges auront à rester inférieures à un certain seuil :

$$x_i \leq x_i^{max}, \quad i = 1, ..., N,$$

contraintes que l'on exprimera sous formes :

$$x_i - x_i^{max} \leq 0, \quad i = 1, ..., N.$$

De même, si les charges correspondent à des poids, elles doivent rester positives. Ceci amène N contraintes d'inégalité supplémentaire de la forme $0 \leq x_i$.

On peut exprimer ces contraintes sous forme matricielle :

$$D_+\mathbf{x} - e_+ \leq 0, \quad D_-\mathbf{x} \leq 0,$$

avec $D_+ = Id_N$ est une matrice identité et $D_- = -Id_N$ et e est le vecteur $e_+ = (x_1^{max}, ..., x_N^{max})^\top$.

4.9.1 contrainte d'égalité

Considérons le problème d'optimisation sous contraintes d'égalité suivant :

$$\begin{cases} \min_{\mathbf{x} \in \mathcal{O}_{ad}} J(\mathbf{x}, U(\mathbf{x})) \\ E(\mathbf{x}, U(\mathbf{x})) = 0 \end{cases}$$

Dans l'exemple donné plus haut, $E(\mathbf{x}, U(\mathbf{x})) = B\mathbf{x} - c$ ne dépend pas de $U(\mathbf{x})$ et est simplement une contrainte linéaire. C'est ce cadre que nous allons considérer.

L'étude de l'optimisation sous contraintes égalité va nous conduire à introduire la notion de dualité et le Lagrangien. Les multiplicateurs de Lagrange sont de outils généraux de la minimisation sous-contraintes [23].

Si les hypothèses du théorème de Lagrange[1] sont vérifiées, en présence de m contraintes d'égalité, il existe m scalaires p_i^* tels que :

$$\nabla J(\mathbf{x}^*, U(\mathbf{x}^*)) + \sum_{i=1}^{m} p_i^* \nabla E_i(\mathbf{x}^*, U(\mathbf{x}^*)) = 0, \tag{4.12}$$

où $\mathbf{x}^*$ désigne l'optimum.

Dans le cadre de notre problème avec une fonctionnelle quadratique et une contrainte d'égalité linéaire, les gradients peuvent s'exprimer exactement et nous avons $m = 1$. L'expression précédente devient :

$$A\mathbf{x}^* - b + \sum_{i=1}^{m} p_i B_i^T = 0. \tag{4.13}$$

Le Lagrangien permet de donner une forme plus concise à cette expression :

$$\mathcal{L}(\mathbf{x}, p) = J(\mathbf{x}, U(\mathbf{x})) + \sum_{i=1}^{p} p_i E_i(\mathbf{x}, U(\mathbf{x})) \tag{4.14}$$

Alors,

$$\mathcal{L}(\mathbf{x}, p) = \frac{1}{2}(A\mathbf{x}, \mathbf{x}) - (b, \mathbf{x}) + (p, B\mathbf{x} - c)$$

Et on obtient $\mathbf{x}^* = A^{-1}(b - B^T p)$. On introduit la fonction duale $H(p)$ suivante :

$$H(p) = \frac{1}{2}(A\mathbf{x}^*, \mathbf{x}^*) - (b, \mathbf{x}^*) + (p, B\mathbf{x}^* - c).$$

Après calcul, on obtient :

$$H(p) = -\frac{1}{2}(A^{-1}(B^T p - b), B^T p - b) - (p, c)$$

qui est une fonction strictement concave. H admet un maximum unique qui peut être caractérisé en solvant la condition d'optimalité d'ordre 1 :

$$\nabla H(p) = -BA^{-1}B^T p + BA^{-1}b - c = 0.$$

1. Le théorème de Lagrange considère le problème de minimisation d'une fonctionnelle coût $J(\mathbf{x})$ pour $\mathbf{x} \in I\!R^n$, sous un ensemble de m contraintes

$$E_i(\mathbf{x}) = 0 \quad \forall i = 1, ..m$$

On suppose qu'il existe une solution $\mathbf{x}^*$ à ce problème et que les fonctions J et E_i soient continûment dérivables en $\mathbf{x}^*$. On suppose que la matrice jacobienne des contraintes soit de rang m au point $\mathbf{x}^*$, c'est à dire que les m vecteurs gradients des E_i au point $\mathbf{x}^*$ soient linéairement indépendants. Alors il existe m scalaires $p_1^*, p_2^*, ...p_m^*$ tels que

$$\nabla J(\mathbf{x}^*) + \sum_{i=1}^{m} p_i^* \nabla E_i(\mathbf{x}^*) = 0 \tag{4.11}$$

Les conditions de Lagrange (4.11) sont des conditions nécessaires, mais non suffisantes en général.

Ce système permet d'obtenir p^* et nous trouvons la solution optimale $\mathbf{x}^* = A^{-1}(b - B^T p^*)$. Dans notre cas, on a pu exprimer l'ensemble des gradients de façon explicite. Ceci n'est pas toujours le cas et nous pouvons être amené à procéder de façon itérative en utilisant un algorithme d'Uzawa par exemple [23].

Nous présentons des résultats d'optimisation sous contrainte en utilisant la méthode directe. Pour valider la démarche, nous allons comparer les résultats avec la situation sans contrainte pour plusieurs seuils de contrainte imposés. Plus exactement, nous considérons une charge totale contrainte égale à 100% de la charge totale du problème sans contrainte ce qui correspond à une contrainte inactive. Puis, nous considérons des valeurs de contrainte égales à 99%, 92%, 91% et 80% de la charge sans contrainte. Il s'agit de montrer comment la solution du problème évolue en fonction de cette contrainte. Ceci permet de valider la démarche en obtenant, d'abord, le résultat de l'optimisation sans contrainte lorsque la contrainte n'est pas active et, ensuite, lorsque la contrainte n'est pas trop forte.

On constate que lorsque la contrainte devient plus présente, les résultats obtenus ne sont plus admissibles d'un point de vue physique. En effet, on ne peut pas imposer de poids négatifs dans notre situation. Il est alors indispensable d'introduire des contraintes d'inégalité pour imposer la positivité des variables d'optimisation.

	Sans contrainte	100%	99%	92%	91%	80%
f1(KN)	5500	5500	5318	4048	3866	1870
f2(KN)	4000	4000	3505	40	-454	-5899
f3(KN)	6000	6000	6042	6341	6384	6853
f4(KN)	4500	4500	4162	1796	1459	-2257
f5(KN)	4999	4999	5721	10773	11495	19433

Table 4.2 – comparaison entre résultats d'optimisation sans contrainte et sous contrainte de charge totale imposée. On constate des valeurs non admissibles car négatives et ceci à partir d'un niveau de contrainte de 91%.

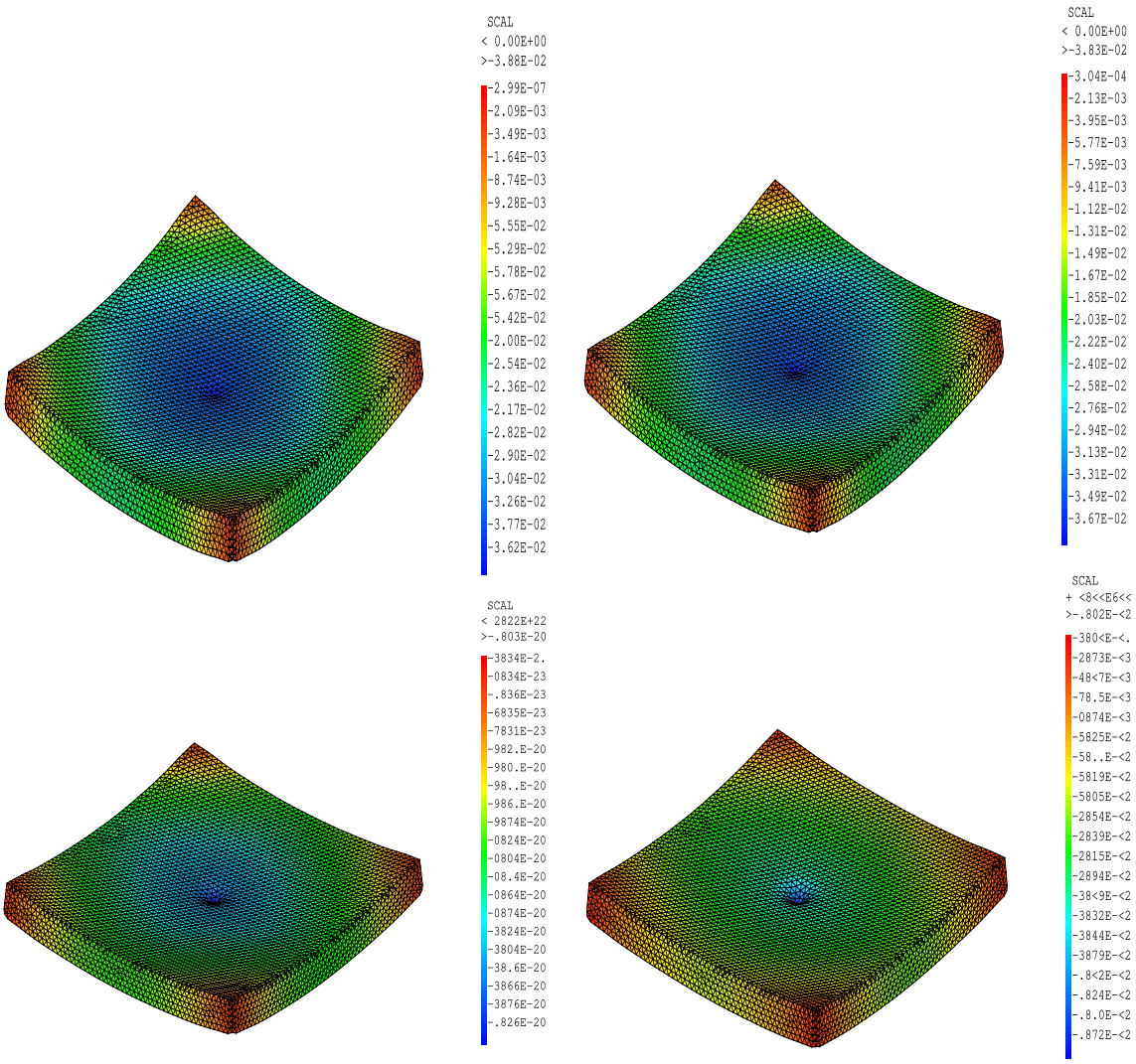

FIGURE 4.8 – Haut-Gauche : déplacement sans contraintes. Haut-Droite : avec contrainte charge totale à 99%. Bas-Gauche : avec contrainte charge totale à 92%. Bas-Droite : avec une contrainte de charge totale égale à 80%.

4.9.2 Contraintes inégalités

Pour prendre en compte les contraintes d'égalité et d'inégalités, mentionnées plus haut, pour assurer l'admissibilité de la solution, nous allons considérer le problème d'optimisation suivant :

$$\begin{cases} \min_{\mathbf{x}\in\mathcal{O}_{ad}} J(\mathbf{x}, U(\mathbf{x})), \\ E(\mathbf{x}, U(\mathbf{x})) = 0, \\ I(\mathbf{x}) \leq 0. \end{cases}$$

Avec, $I(\mathbf{x})$ qui peut s'exprimer sous forme matricielle comme indiquée plus haut.

Cependant, l'implémentation peut être très simple et éviter une formulation matricielle en considérant l'algorithme de gradient projeté lorsque les contraintes sont

de type 'boîte' :

$$0 \leq x_i \leq x_i^{max}.$$

Gradient projeté

Une première solution est l'utilisation de l'algorithme de gradient projeté qui est simple à mettre en œuvre si l'on peut facilement projeter un point sur l'ensemble admissible des solutions vérifiant les contraintes. C'est le cas pour nos contraintes de positivité. Le projeté se fait alors simplement en prenant $\bar{x}_i = \min(\max(x_i, 0), x_i^{max})$. La méthode du gradient projeté s'obtient alors facilement à partir de l'algorithme du gradient appliqué à notre problème d'optimisation sous contrainte d'égalité formulé précédemment 4.9.1 avec le Lagrangien $\mathcal{L}$ selon :

$$\begin{cases} \tilde{\mathbf{x}}^{k+1} = \mathbf{x}^k - \rho \nabla \mathcal{L}(\mathbf{x}^k, p) \\ \mathbf{x}^{k+1} = P_K(\tilde{\mathbf{x}}^{k+1}) \ \text{ selon } x_i^{k+1} = \min(\max(0, \tilde{x}_i^{k+1}), x_i^{max}) \end{cases}$$

où $\nabla \mathcal{L}(\mathbf{x}^k, p)) = A\mathbf{x}^k - b + B^T p$.

Dans cette itération p est fixe et cet algorithme fonctionne dans notre cas car $p = (BA^{-1}B^T)^{-1}(BA^{-1}b - c)$ et est donc indépendant de x^k.

En règle générale, un intérêt de cette approche est qu'elle ne nécessite pas d'inversion de système linéaire de grande taille car nous avons uniquement besoin d'effectuer des multiplications matrice-vecteur et des additions de vecteurs. Aussi, cette approche itérative permet d'envisager des contraintes plus complexes comme nous le verrons plus bas.

Nous considérons le problème visité plus haut avec $N = 5$ charges sous une contrainte d'égalité de charge totale donnée et des contraintes d'inégalité de charges unitaires maximales inférieures où égales à 100% et à 80% des charges obtenues comme solutions du problème sans contrainte.

Le tableau 4.9.2 montre les résultats obtenu par l'algorithme de gradient projeté en prenant en compte la contrainte d'égalité. On constate que la solution est inchangée pour les contraintes d'égalité supérieures à 92% et on retrouve des valeurs positives de charge désormais pour les contraintes plus fortes.

Contraintes d'inégalité générales

Si les contraintes d'inégalité sont plus complexes, de sorte qu'on ne puisse plus disposer de projecteur évident sur l'ensemble admissible, nous devons vérifier les hypothèses du théorème de **Kuhn-Tucker** [2] et utiliser, par exemple, un algorithme de type Uzawa [23].

2. Théorème de Kuhn-Tucker : Si $\mathbf{x}^*$ est un point régulier de l'ensemble des contraintes et si $\mathbf{x}^*$ est un minimum local de J alors il existe m réels p_i^* positifs ou nuls tels que

$$\begin{cases} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, p^*) = \nabla_{\mathbf{x}} J(\mathbf{x}^*, U(\mathbf{x}^*)) + \sum_{i=1}^m p_i^* \nabla E_i(\mathbf{x}^*, U(\mathbf{x}^*)) = 0 \\ p_i E_i(\mathbf{x}, U(\mathbf{x}) = 0 \quad \forall i = 1, ..m \\ E_i(\mathbf{x}, U(\mathbf{x}) \leq 0, \quad p_i \geq 0 \quad \forall i = 1, ..m \end{cases}$$

Si la fonction coût J est strictement convexe et l'ensemble des contraintes est convexe ce minimum est unique.

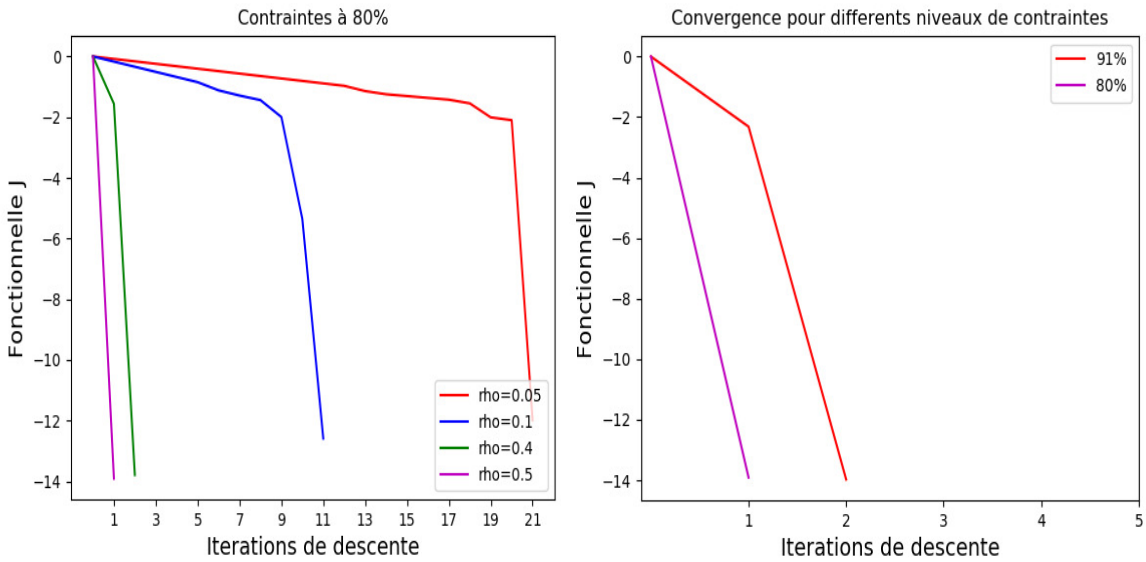

FIGURE 4.9 – Évolution de la fonctionnelle en fonction des itérations de gradient projeté. L'algorithme converge rapidement sans besoin de la résolution d'un système linéaire par méthode directe (figure de gauche). Figure de droite montre la convergence pour les deux niveaux de contraintes d'égalité avec le même pas de descente.

	Contrainte d'égalité		Contrainte d'inégalité	
	91%	80%	91%	80%
f1(KN)	3866	1870	5005	4400
f2(KN)	-454	-5899	3640	3200
f3(KN)	6384	6853	5460	4800
f4(KN)	1459	-2257	4095	3600
f5(KN)	11495	19433	4549	3999

TABLE 4.3 – Optimisation sans contraintes de positivité et après imposition de ces contraintes par la méthode de gradient projeté.

Dans notre problème avec une fonctionnelle quadratique le problème d'optimisation s'exprime par l'expression :

$$\begin{cases} \nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*, p^*) = A\mathbf{x}^* - b + D^T p^* = 0 \\ E_i(\mathbf{x}^*, U(\mathbf{x}^*)) \leq 0, \quad p_i^* \geq 0 \quad \forall i = 1, .., N \end{cases}$$

Nous utilisons l'Algorithme d'Uzawa :

L'algorithme d'Uzawa

— Initialisation : $p^0 \in I\!R_+^m$ donné , $\rho > 0$ assez petit
— Pour $k \to k+1$ faire
— Calculer x^k solution du problème primal $\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, p^k)$, $(A\mathbf{x}^k = b - B^T p^k)$
— Tester $J(\mathbf{x}^k) \leq J(\mathbf{x}^{k-1})$
— Calculer $p^{k+1} = P_K(p^k + \rho E(\mathbf{x}^k))$ où P_K est la projection sur les $p_i \geq 0$, alors ça devient
$p_i^{k+1} = \max(p_i^k + \rho E_i(\mathbf{x}^k), 0)$

Cet algorithme permet aussi de procéder à des projections simples sur les valeurs positives des multiplicateurs de Lagrange et on retrouve donc la facilité d'implémentation que nous avons avec la méthode de gradient projeté plus haut dans le cadre des contraintes de bornes supérieures et inférieures sur les variables d'optimisation.

Chapitre 5

Estimation d'erreur a posteriori et adaptation de maillages

La performance des méthodes d'éléments finis est étroitement liée à la qualité du maillage. Bien sur, on peut envisager de raffiner le maillage jusqu'à l'obtention d'une solution indépendante du maillage (ou convergée en maillage). Mais ceci implique un coût de calcul très important. On recherche alors des raffinements locaux.

Dans ce chapitre, nous rappelons quelques résultats classiques et algorithmes itératifs d'adaptation de maillages non-structurés. Nous mettons en évidence l'impact de la qualité du résultat numérique sur la solution du problème d'optimisation. Pour plus de simplicité, une partie de la présentation est faite en une dimension d'espace, mais les idées se généralisent en dimension supérieure. Nous utiliserons le logiciel `Freefem` pour illustrer notre propos.

Nous considérons le problème de l'erreur de discrétisation spatiale dans les modèles numériques et de l'adaptation du maillage afin de minimiser, ou, au moins, de contrôler cette erreur. La qualité du maillage est cruciale pour la qualité du résultat et nous souhaitons estimer l'impact de l'adaptation de maillage sur la solution de notre problème d'optimisation.

Nous commençons par des résultats de majoration d'erreur a priori dans le cas de maillages en éléments finis. Nous présentons ensuite les aspects pratiques de l'estimation d'erreur a posteriori et de l'adaptation de maillage.

Les estimations a posteriori utilisent les solutions approchées et permettent l'adaptation des maillages en pratique.

5.1 Analyse d'erreur a priori dans les méthodes d'éléments finis

Nous rappelons les résultats d'analyse de l'erreur dans le cas d'une intégration de second membre exacte.

Les résultats d'existence et d'unicité des problèmes continus et discrets de la forme $a(u, v) = l(v)$ sont bien connus et ont été rappelés précédemment.

Théorème fondamental : Soit M la constante intervenant dans l'hypothèse de

continuité de a : $a(u,v) \leq M \|u\| \|v\|$ et m la constante intervenant dans l'hypothèse d'ellipticité, on a la majoration d'erreur suivante :

$$\|u - u_h\| \leq \frac{M}{m} inf_{v_h \in V_h} \|u - v_h\| \tag{5.1}$$

Si a est symétrique :

$$a(u - u_h, v_h) = 0 \quad \forall v_h \in V_h$$

signifie que u_h est la projection de u dans V_h au sens du produit scalaire a. Dans ce cas on a

$$\|u - u_h\| \leq \sqrt{\frac{M}{m}} inf_{v_h \in V_h} \|u - v_h\| \tag{5.2}$$

Dans la suite, nous considérons des problèmes elliptiques d'ordre 2. L'espace des fonctions tests considérées est l'espace H^1 ou un sous-espace de H^1.

5.1.1 Majoration d'erreur en éléments P_k avec intégration exacte

Pour simplifier, nous supposons le domaine de calcul Ω exactement recouvert par l'ensemble des éléments du maillage. Dans le cas de triangles le domaine sera donc supposé polygonal. Nous rappelons le résultat suivant :

Dans le cas d'éléments finis de degré k (P_k et d'une solution exacte u du problème elliptique suffisamment régulière ($u \in H^{k+1}(\Omega)$), on a la majoration d'erreur suivante en norme H^1 :

$$\|u - u_h\|_{1,2} \leq Ch^k |u|_{k+1,2} \tag{5.3}$$

où h est la longueur du plus grand côté d'élément du maillage et où la constante C est indépendante de h.

Remarque : Si le domaine Ω n'est pas exactement recouvert par le maillage, on doit l'approcher par un maillage Ω_h dont la frontière suit les côtés des éléments droits ou isoparamétriques courbes choisis.

5.1.2 Les éléments P1 en dimension deux

Supposons le domaine polygonal plan Ω discrétisé par un maillage en triangles T_i. Prenons pour espace d'approximation V_h l'espace des fonctions continues affines par morceaux (affines sur les triangles T_i). Nous utilisons encore le résultat

$$\|u - u_h\|_{1,2} \leq \frac{M}{m} \|u - v_h\|_{1,2} \quad \forall v_h \in V_h \tag{5.4}$$

En choisissant à nouveau pour v_h l'interpolé $P_h u$ de u dans l'espace d'éléments finis V_h, on majore l'erreur d'approximation $\|u - u_h\|$ par une constante fois l'erreur d'interpolation $\|u - P_h u\|$.

On rappelle les résultats suivants :

Notons h la longueur du plus grand côté et θ_0 le plus petit angle au sommet de tous les triangles du maillage et notons D^2v la matrice Hessienne d'une fonction deux fois continûment différentiable v.

$$D^2v(x,y) = \begin{pmatrix} \dfrac{\partial^2 v}{\partial x^2} & \dfrac{\partial^2 v}{\partial x \partial y} \\ \dfrac{\partial^2 v}{\partial x \partial y} & \dfrac{\partial^2 v}{\partial y^2} \end{pmatrix} \tag{5.5}$$

et $|D^2v(x,y)|$ sa norme spectrale.
On a alors $\forall x, y \in \Omega$:

$$|u(x,y) - P_h u(x,y)| \leq \frac{h^2}{2}\, sup_{x,y\in\Omega}|D^2v(x,y)| \tag{5.6}$$

$$|\nabla u(x,y) - \nabla P_h u(x,y)| \leq 3\frac{h}{sin(\theta_0)}\, sup_{x,y\in\Omega}|D^2v(x,y)| \tag{5.7}$$

d'où

$$\|u - P_h u\|_{1,2} \leq C\, h\, sup_{x,y\in\Omega}|D^2v(x,y)| \tag{5.8}$$

et enfin

$$\|u - u_h\|_{1,2} \leq C\, h\, sup_{x,y\in\Omega}|D^2v(x,y)| \tag{5.9}$$

5.2 Contrôle local de métrique

Les résultats précédents suggèrent qu'on peut estimer la qualité de la solution numérique en corrélant le pas discrétisation locale et l'estimation de la dérivée seconde de la solution. En d'autres termes, on espére contrôler l'erreur sur la solution en contrôlant celle sur l'interpolation linéaire.

L'adaptation de maillages par contrôle de métrique utilise cette idée et modifie le choix du produit scalaire euclidien qui sert à définir les distances dans le maillage pour pouvoir générer les éléments dans une nouvelle métrique suivant certains critères.

On cherchera la métrique qui équirépartira l'erreur d'interpolation.

Présentons cette technique dans le cadre de la dimension 1. Cette approche se déclinera le long de chaque segment du maillage en dimension supérieure.

On a vu que l'on dispose de la majoration suivante pour l'erreur d'interpolation en élément $P1$:

$$\|u - \Pi_h u\|_0 \;\leq\; c\, h^2\, |u''|, \tag{5.10}$$

où $\Pi_h u$ est l'interpolation linéaire de u, h la taille des éléments. La métrique locale en chaque nœud x du maillage est définie par :

$$\lambda_i = \min\left(\max\left(\frac{1}{\epsilon}|u''|\,,\, \frac{1}{h^2_{\min}}\right),\, \frac{1}{h^2_{\max}}\right), \tag{5.11}$$

où ϵ, $h_{\min}$ et $h_{\max}$ sont respectivement l'erreur et les longueurs minimale et maximale désirées des mailles. La distance entre deux points sera alors définie par :

$$|x-y|_\lambda(\eta) = \sqrt{\lambda(\eta)}|x-y|, \quad \eta \in [x,y].$$

On peut ainsi choisir d'équirépartir l'erreur d'interpolation en générant des segments de même longueur h dans cette nouvelle métrique. En notant a_i la distance entre le i^{eme} et le $(i+1)^{eme}$ nœud de la discrétisation, on a :

$$a_i = \frac{h}{\sqrt{\lambda_i}}\,. \tag{5.12}$$

De la même manière, en dimension supérieure, on définit la distance locale par :

$$\langle x\,,\,y\rangle_\lambda = x^t\,\lambda\,y \iff \|x\,-\,y\|_\lambda = \sqrt{(x\,-\,y)^t\,\lambda\,(x\,-\,y)}\,. \tag{5.13}$$

Une fois le nouveau maillage obtenu, il faut **interpoler la solution** disponible depuis l'ancien maillage sur le nouveau, par exemple en utilisant une interpolation linéaire.

En résumé, l'algorithme de point fixe d'adaptation mis en place est le suivant.

A l'itération i d'adaptation, on note le maillage, la solution discrète de l'équation et la métrique par $\mathcal{H}_i$, $\mathcal{S}_i$, $\mathcal{M}_i$.

Initialisation : $i = 0$,	$\mathcal{H}_0, \mathcal{S}_0$: donné
Tant que $(i \leq ITER - MAX$ **ou**	**Test d'arrêt**) **Faire**
Calculer la métrique :	$(\mathcal{H}_i, \mathcal{S}_i) \to \mathcal{M}_i$
Générer le nouveau maillage :	$(\mathcal{H}_i, \mathcal{M}_i) \to \mathcal{H}_{i+1}$
Interpoler l'ancienne solution :	$(\mathcal{H}_i, \mathcal{S}_i, \mathcal{H}_{i+1}) \to \overline{\mathcal{S}}_{i+1}$
Calculer la nouvelle solution :	$(\mathcal{H}_{i+1}, \overline{\mathcal{S}}_{i+1}) \to \mathcal{S}_{i+1}$
$i++$	
Fin	

5.3 Adaptation de maillage pour le problème Poisson

Nous donnons le type de changements qu'il faut apporter au script donné au chapitre précédent pour introduire l'adaptation de maillage pour la solution homogène et chacune des solutions élémentaires.

On retrouve les éléments introduits dans l'expression 5.11 avec les tailles minimale et maximale des segments et le niveau d'erreur d'interpolation cible.

La vue tridimensionnelle dans la figure 5.1 montre que la solution cible ne présente pas de symetrie spatiale. Or, on constate que parfois les solutions élémentaires peuvent bénéficier de certaines symétries, comme ici. La non symétrie est liée aux paramètres d'optimisation. Nous considérons une condition aux limites de Dirichlet non homogène, mais constante et égale à 1 pour éviter un rafinement au bord pour mieux illustrer l'adaptation du maillage lié à la présence des sources.

```
int nadapt=10; //itération d'adaptation de maillage
/////////////// Solution U1
Th1=Th2;  // initialisation maillage de fond
Vh1 u1; ff=f1; u=u1; ibc=0;
for (int i=0;i<nadapt;i++){
Poisson;  //Equation de Poisson sur maillage variable
Th1=adaptmesh(Th1,u1,err=0.0025,hmax=0.5,hmin=0.01);
}
```

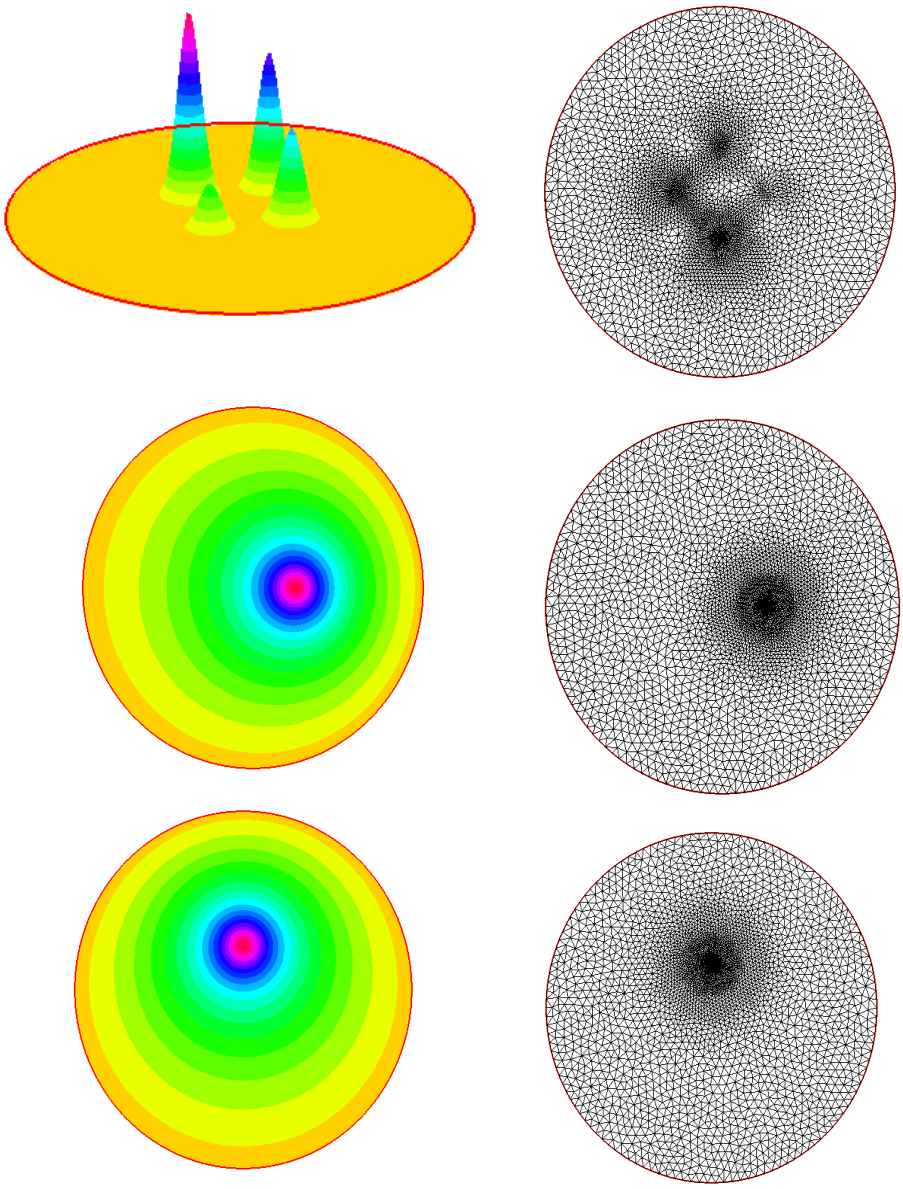

FIGURE 5.1 – Adaptation de maillage pour le problème de Poisson et solutions élémentaires : solution de référence et maillage adapté associé, ainsi que deux solutions élémentaires et maillages adaptés associés.

5.4 Impact du contôle de la qualité du maillage sur la solution du problème d'optimisation

Rappelons que notre algorithme d'optimisation directe utilise le pré-calcul d'une solution homogène et des solutions élémentaires pour des chargements unitaires. Ces solutions sont assez différentes les unes des autres et la figure 5.1 a montre que leur calcul précis exige des maillages différents. Nous nous retrouvons avec $N+1$ maillages. L'assemblage du système linéaire produisant la solution du problème d'optimisation exige le calcul de produits scalaires croisés entre ces solutions. Il faut donc exprimer ces solutions sur un maillage unique de référence pour permettre ces calculs.

Nous considérons cette fois un problème non admissible dans le sens où la déformation cible n'a pas été obtenue comme dans le chapitre précédent par une distribution de charges qui peut être atteinte par la distribution de charges recherchée.

La figure 5.2 montre la solution de ce problème d'optimisation sans et avec adaptation de maillage. Nous avons volontairement considéré un maillage très grossier pour montrer que le problème d'optimisation peut être résolu, mais peut donner un résultat qui doit être considéré avec prudence. Lorsque l'adaptation de maillage est introduit, on constate que la solution optimale est différente et que la fonctionnelle, aussi bien initialement qu'à l'optimum, est très différente avec l'adaptation de maillage. La fonctionnelle peut être vue comme un indicateur de performance. Ceci démontre que montrer une amélioration de performance n'est pas recevable et considérée avec confiance que si les erreurs possibles liées à la solution de l'équation d'état (ici problème de Poisson) sont minimisées.

La déformation cible a été générée avec 5 charges comme indiquée dans la figure 5.2 avec les valeurs de variables $(f_1, f_2, f_3, f_4, f_5) = (1, 2, 3, 4,\ 6)$. Nous essayons de trouver 4 charges qui permettent de trouver la déformation la plus proche de la déformation cible. La distribution optimale trouvée sur maillage grossier est $(f_1, f_2, f_3, f_4) = (6.6, 5030, 12.7,\ 17.6)$ alors qu'avec les maillages adaptés et une solution de l'équation d'état précise, on trouve des valeurs bien plus satisfaisantes $(f_1, f_2, f_3, f_4) = (2.6, 3.6, 4.6,\ 5.6)$.

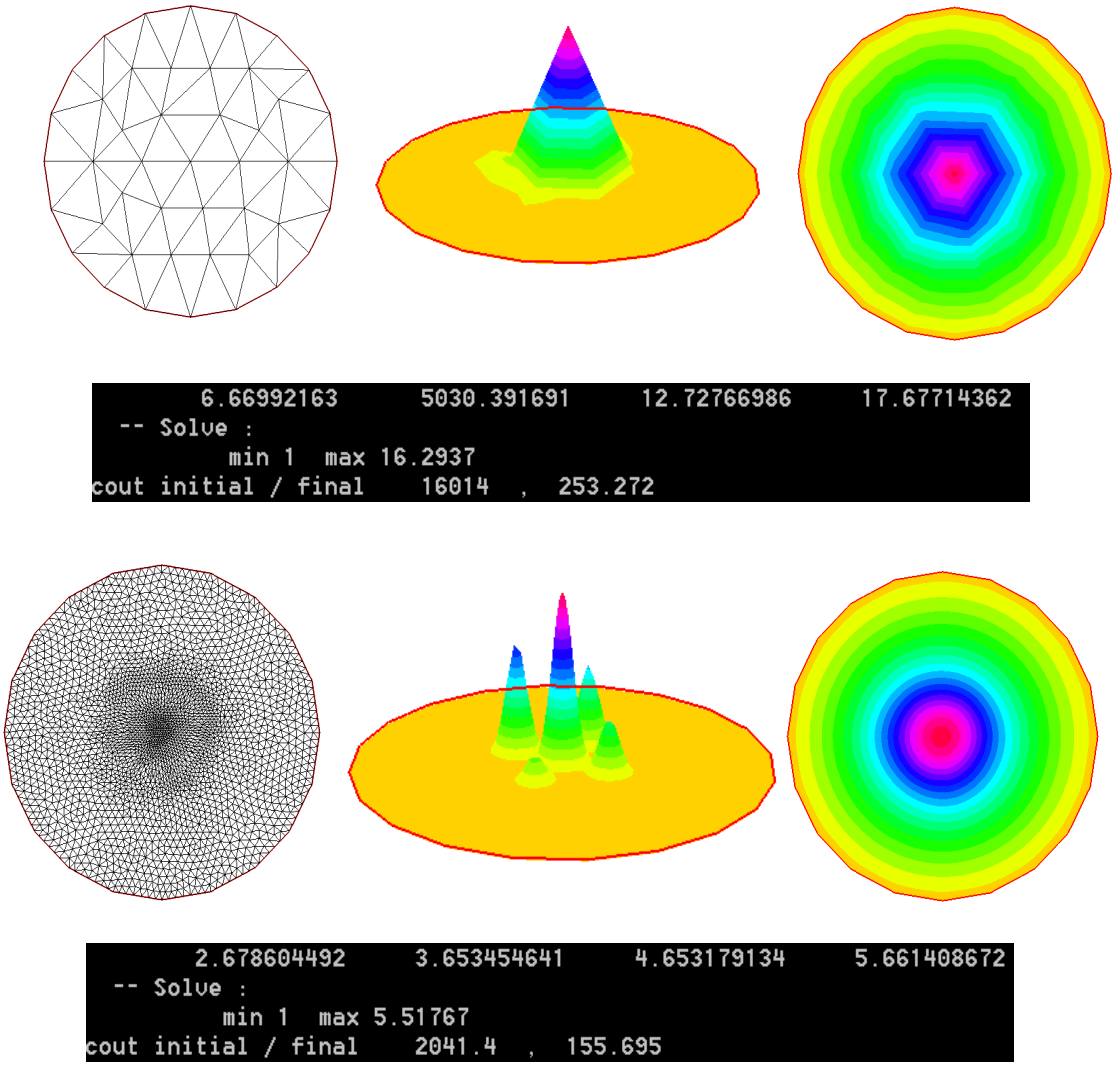

FIGURE 5.2 – Optimisation de distribution de charge par méthode directe sur un domaine circulaire sans et avec adaptation de maillage pour un problème non admissible : la déformation cible n'est pas associée à une distribution de charges atteignable par les paramètres d'optimisation envisagée. La fonctionnelle à l'optimum est clairement différente de la valeur atteinte sans adaptation de maillage, ainsi que la distribution de charges optimale.

Chapitre 6

Propagation des incertitudes

Ce chapitre présente s'intéresse à la propagation d'incertitudes dans un contexte de simulation numérique déterministe pour des problèmes d'ingénieur. Il s'agit d'exploiter les informations disponibles lors des simulations numériques et d'adopter une démarche d'analyse de données.

On verra comment la méthode directe permet une analyse de sensibilité de la solution par rapport aux petites perturbations de la déformation cible. On verra aussi comment envisager une propagation rétrograde des incertitudes grâce à la dépendance linéaire qui permet d'exprimer de façon exacte l'ensemble des opérations.

En effet, les solutions élémentaires sont indépendantes des incertitudes éventuelles de la solution cible qui peut être prescrite, non pas de façon déterministe, mais uniquement aux travers des intervalles d'admissibilité par exemple.

Ce chapitre aborde cette question grâce au contrôle des moments de la fonctionnelle et une optimisation multi-point et multi-critère. Dans une deuxième partie, nous montrons comment construire avec un coût de calcul supplémentaire négligeable la matrice de covariance des variables d'optimisation. Deux procédures d'estimation de cette matrice sont présentées en tirant avantage de l'existence de présence de linéarité entre les variables d'optimisation et la solution de l'équation d'état.

Ce chapitre, en anglais, a été soumis pour publication au jounal ARIMA (Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées) sous le titre :*Low-complexity moment-based robust design and uncertainty back-propagation.*

6.1 Introduction

The performance of a system designed for given functioning conditions often seriously degrades when these conditions are modified. Today's industrial robust design mainly relies on reduced-order modeling and intelligent sampling [1, 6, 7, 45] which either does not use high-fidelity simulations during design or uses lower accuracy than what would be affordable in a single-point optimization.

By robust design we mean a proposition which ensures similar performances over a given operation range. We will discuss the implication of this requirement on the moments of the performance functional. Our aim is to propose a methodology which permits to design a system having similar performances over a given range of its operating condition or functioning parameters. From a practical point of view, we would like this to be achieved modifying as less as possible an existing single-point optimization platform. In particular, we would like to avoid replacing the high-fidelity ingredients of the platform by low-complexity solvers. Finally, we would like the parallel time-to-solution to remain comparable to the single-point situation. We will show that our direct optimization algorithm permits to achieve such a task.

We address the solution of robust moment-based optimization problems after a multi-point reformulation. The first four moments are considered (i.e. mean, variance, skewness and kurtosis) going beyond classical engineering optimization based on the control of the mean and variance. In particular, the impact on the design of a control of the third and fourth moments is discussed. The multi-point formulation leads to discrete expressions for the moments. linking moment-based and multi-point optimizations. The linearity of the sums in the discrete moments permits an easy evaluation of their gradients with respect to the design variables. Optimal sampling issues are analyzed and a procedure is proposed to quantify the confidence level on the robustness of the design. The proposed formulation is fully parallel and the parallel time-to-solution is comparable to single-point situations.

The literature on multi-point optimization is vast and exhaustive referencing is out of our scope. This formulation has been used, in particular, to extend single-point optimization to account for additional operating conditions.

Forward and backward uncertainty propagation are obviously of great importance with a huge literature dedicated to uncertainty quantification (UQ) [3, 8, 26]. Forward propagation aims at defining, for instance, a probability density function (PDF) for a functional J knowing those of optimization variables [20, 22, 30]. This can be done, for instance, through Monte Carlo simulations or a separation between deterministic and stochastic features using Karhunen-Loeve theory (polynomial chaos theory belongs to this class) [16, 17, 45, 46]. Backward propagation aims at reducing models bias or calibrating models parameters knowing the PDF of J [9, 24, 43]. This can be seen as a minimization problem and Kalman filters [25] give, for instance, an elegant framework for this inversion assimilating the uncertainties on the observations.

We particularly discuss one aspect of UQ where the target state u^* used in an inverse problem is uncertain. This is the case, for instance, minimizing $J(u(\mathbf{x}), u^*) = \|u(\mathbf{x}) - u^*\|$ to reduce the distance between a model state $u(\mathbf{x})$ and observations or

target u^*.

Targeting uncertain data is a realistic situation as the acquired data are usually uncertain. It is therefore interesting to be able to quantify the impact of this uncertainty on the inversion results. An important information will be the sensitivity of the design to a given level of uncertainty on the data at some location. Indeed, if this sensitivity is low, this would be an indication that a more accurate acquisition there is unnecessary.

Considering the target as uncertain is also interesting because we do not always have existence of a solution for an inversion problem as u^* is not necessary solution of the state equation making an exact or deterministic inversion pointless. Also, the approach permits to go beyond inversions based on least-squares minimization involving a mean state target.

The uncertainty in measurements is also an interesting way to account for the presence of variability in the state. More generally, as a model or numerical procedure is by nature imperfect and partial, we can consider this uncertainty as a representation and estimation of these imperfections. These imperfections are even more present in inverse problems where one cannot afford the same level of resolution than for a single simulation. We therefore need to be able to quantify the impact of these weaknesses. Being able to handle uncertain targets in inverse design is therefore also useful to account for epistemic uncertainties related to possible model or solution procedures deficiencies.

The paper also addresses the issue of backward uncertainty propagation through the estimation of the covariance matrix of the optimization variables. Two procedures for the estimation of this matrix are presented taking advantage of the existence of local linear relations between the target state u^* and the optimization variables $\mathbf{x}$.

We present the different ingredients of the paper in the context of a linear problem for load distribution optimization with the objective of achieving a target deformation or displacement of a shape. The state equations we consider are those of linear elasticity and we would like to take advantage, as much as possible, of the linear relationship between the optimization variables (e.g loads) and the state variables (e.g the displacements) in the solution of the problem and the different uncertainty quantifications.

The paper starts with the description of our single-point optimization problem to illustrate the situation where the solution of the inverse problem can be explicitly expressed thanks to the mentioned linearity. Then we address robust parametric optimization and its solution through moment-based and multi-point formulations. The paper ends with two low-complexity approaches for backward uncertainty propagation and estimation of the covariance matrix of the optimization variables.

6.2 State equations and single-point optimization problem

Let us start with the single-point optimization problem of interest. We are interested by inverse problems where the functional typically measures a distance to

a target solution.

We consider a generic state equation with a linear state equation and also linear dependency between the optimization variables $\mathbf{x}$ (controlling in our case the loads distribution) and the state variable $u : \mathcal{F}(u(\mathbf{x})) = f(\mathbf{x})$, where u is therefore a function of $\mathbf{x}$.

Our single-point optimization problem reads :

$$\min_{\mathbf{x}\in\mathcal{O}_{ad}} J(u(\mathbf{x}), \alpha) \quad \text{under the constraint that} \qquad \mathcal{F}(u) = f(\mathbf{x}). \tag{6.1}$$

Where α represents the other independent variables of the problem such as geometry entities or operating conditions [21, 43]. Here, for simplicity we do not mentioned all dependencies in u, such as the spatial dependency and so forth. In what follows, α is the target solution u^* which we consider as uncertain and the functional measures the least-square misfit between the state and this target.

In single-point optimization α is fixed in opposition to multi-point or robust optimizations where α components will no longer be fixed but given either through intervals or more sophisticated probability density functions.

The source term $f(\mathbf{x})$ contains N loads expressed as $x_i\delta(s_i)$ where $\delta(s_i)$ is a spatial localization function in space variable s. One would like to optimize :

$$f(\mathbf{x}) = \sum_{i=1}^{N} x_i\delta(s_i). \tag{6.2}$$

The optimization variable is then $\mathbf{x} = (x_1, ..., x_N) \in (\mathbb{R}^+)^N$.

The inverse problem of interest aims at making the solution u as close as a target α over the calculation domain Ω, or some subset of it. Using a least-squares formulation over the domain with spatial integration, this reads :

$$J(u(\mathbf{x}), \alpha = u^*) = \frac{1}{2}\int_\Omega (u(\mathbf{x}) - u^*)^2 ds. \tag{6.3}$$

Taking advantage of the linear dependency between $\mathbf{x}$ and the solution u and using the superposition principle, the solution can be expressed through projections over $N + 1$ elementary solutions :

$$u = u_0 + \sum_{i=1}^{N} x_i u_i,$$

In this decomposition, u_0 is the homogeneous solution (the zero load situation together with non-homogeneous boundary conditions) and the other elementary solutions $u_{i=1,...,N}$ are obtained respectively with $\mathbf{x} = (0, ..., 0, 1, 0, ..., 0)$ with 1 at the ith position. The elementary solutions are obtained with homogeneous Dirichlet boundary conditions.

Once the homogeneous and elementary solutions known, the optimization variable $\mathbf{x}$ is solution of a linear system :

$$A\mathbf{x} = b(\alpha = u^*), \tag{6.4}$$

of size N derived from first order optimality condition for J, $\nabla_x J(u(\mathbf{x}), \alpha) = 0$, with :

$$A_{i,j} =< u_i \, , \, u_j > \quad \text{and} \quad b_i(\alpha = u^*) =< (u_0 - u^*) \, , \, u_i >, \quad \text{for} \quad i, j = 1, ..., N,$$

where $< v \, , \, w >= \int_\Omega v(s) \, w(s) \, ds$.

A fundamental remark now is that the cost of this inversion is negligible compared to the cost of calculating the elementary solutions. Also, it is important to notice that the matrix A does not depend on α. Therefore, any numerical transformation necessary for its inversion is made only once. Uncertainty quantification is situation where the target solution α is uncertain becomes therefore feasible at low cost as it will not require any extra solution of the state equations. In the sequel we will discuss different ways to handle this problem.

6.2.1 3D elasticity

We will illustrate the ingredients of the paper on a 3D problem where the state equation $\mathcal{F}(u) = f(\mathbf{x})$ is the elasticity system whose solution is given using the `Cast3M` [28] industrial solver on tetrahedral meshes which we consider as a black-box. Something which is made possible thanks to the formulation of the optimization problem and this is one strength of the approach.

Briefly speaking, `Cast3M` is a finite element software for structural and fluid mechanics developed by the French Alternative Energies and Atomic Energy Commission (CEA). `Cast3M` provides a mesh generation tool, partial differential equations simulation modules coupled with visualization tools.

The 3D problem we consider consists of a geometry of an elastic piece represented in a $(10m \times 10m \times 1m)$ concrete slab with hollow bodies [41] , supported by point supports at the four corners modeled by homogeneous Dirichlet boundary conditions for the displacement. The mechanical characteristics of the concrete considered are $E = 30GPa$ for Young's modulus and $\nu = 0.2$ for Poisson's ratio. This is a complex problem and a full 3D calculation is quite time consuming. The present solution based on the use of elementary 3D solution is therefore quite adapted for the optimization of load distributions for such a cases as the state equations calculations can be carried out independently from the optimization problem and in a fully parallel way. The hollow bodies involve complex mesh generation issues as shown in Figure 6.2.1. The 3D mesh has about 14400 grid points.

We assume negligible volume forces and consider a situation with a relatively small number of loads to ease the comparison between the different approaches. Considering higher number of loads does not pose any supplementary difficulty and only requires parallel calculations of additional independent elementary solutions.

6.2.2 Illustration using $N = 5$ optimization variables

We apply an arbitrary five point loads distribution (i.e. $N = 5$). Any other combination can be considered, five is suitable as enough large to illustrate the approach and enough low to show images of all elementary contributions.

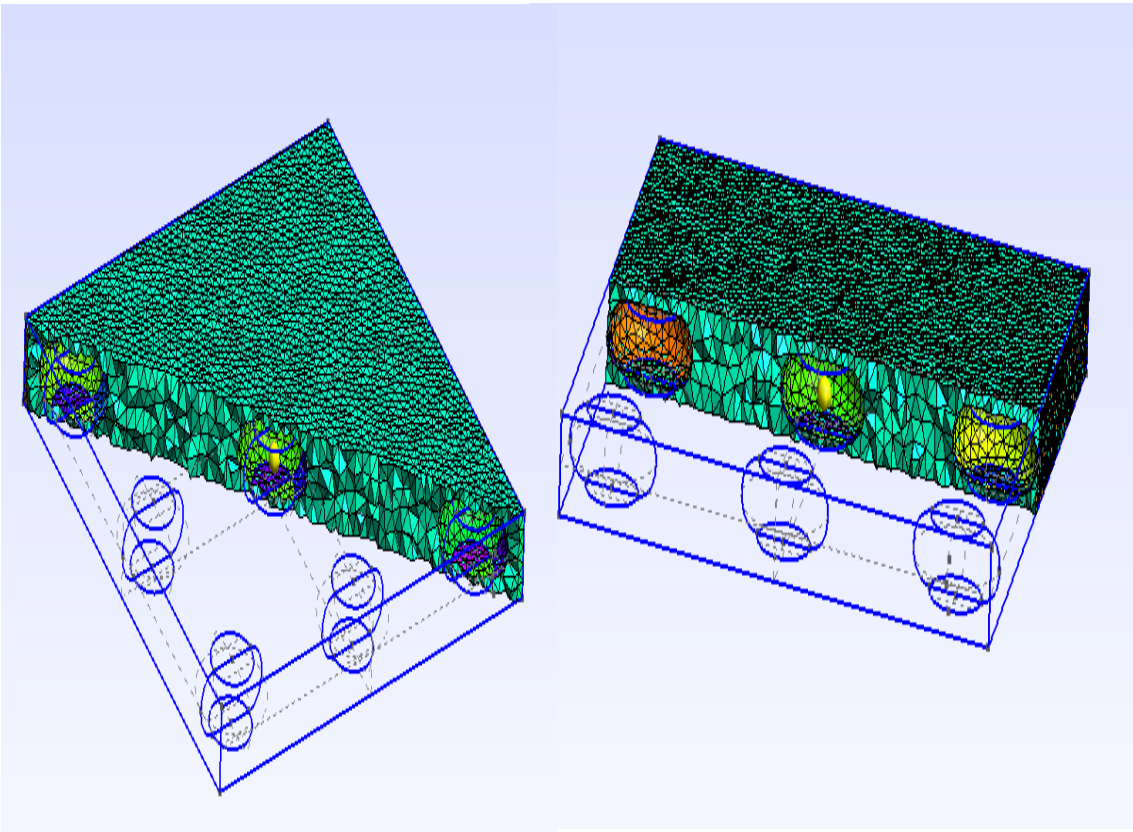

FIGURE 6.1 – Example of a complex mesh generated by `Gmsh` [15] taking into account the presence of hollow bodies.

We calculate the elementary displacements $u_{i=1,\dots,5}$ obtained after application of unit forces x_i. The total displacement is the sum of these elementary displacements $u = u_0 + \sum_{i=1}^{N=5} x_i u_i$ where u_0 is the displacement without any forces applied but in the presence of Dirichlet boundary conditions. In our problem we consider $u_0 = 0$.

Figure 6.2 shows the elementary and total structural displacements which will be considered as target deterministic displacement u^*. Uncertainty analysis will take place around this target displacement.

6.3 Robust parametric optimization

Consider a cost function involving the optimization variables $\mathbf{x}$ in an admissible ensemble $\mathbf{O}_{ad}$ and another parameter α, not considered as a design variable :

$$\min_{\mathbf{x}\in\mathbf{O}_{ad}} J(\mathbf{x},\alpha), \quad \alpha \in \mathbf{I} \subset \mathbb{R}^n, \mathbf{O}_{ad} \subset \mathbb{R}^N. \tag{6.5}$$

As mentioned, α can be an operating condition or the target solution in an inverse problem. Both the operating conditions and the target solution can be uncertain and given only through probability density functions. Handling the uncertainty on

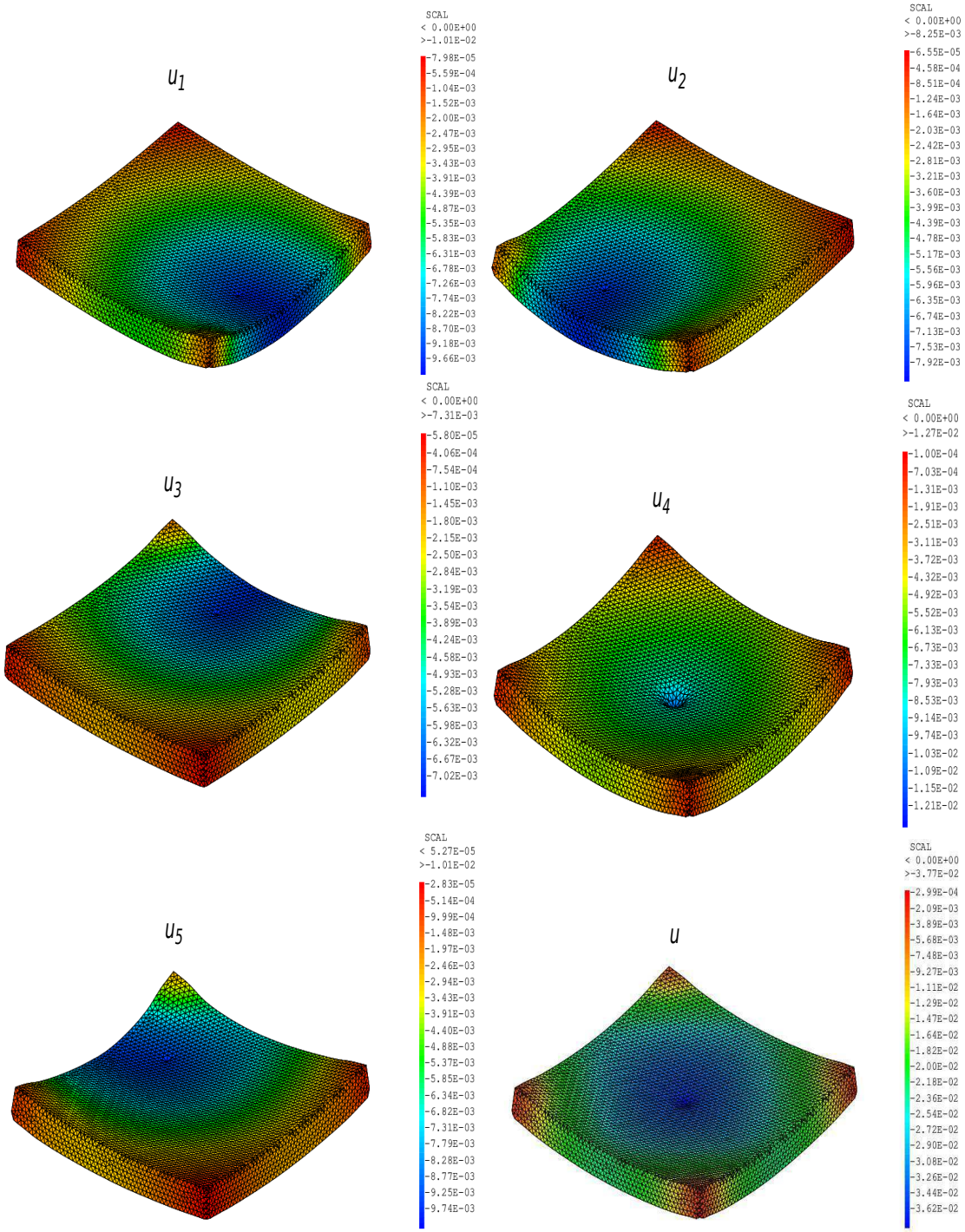

FIGURE 6.2 – The 5 first pictures represent elementary displacements $u_{i=1,\ldots,5}$ and the bottom right picture shows the total structural displacement $u = u_0 + \sum_{i=1}^{N=5} x_i u_i$, where $u_0 = 0$ in this case. The elementary displacements cannot be retrieved using symmetry considerations from each other.

the operating conditions is necessary to achieve robust design and usually requires a multi-point formulation. The number of operating condition parameters is usually small compared to the size of the optimization problem. This is a very general context and we previously visited it to address robustness issues in optimization with respect to $\mathbf{x}$ and α in general frameworks [35, 33, 34, 36].

6.3.1 Moment-based optimization

Multi-point optimization can be used to address such optimization problem [32]. The aim is to remove the dependency in α during optimization. This can be done, for instance, minimizing a functional $J(\mathbf{x})$ encapsulating this dependency under a constraint on the higher moments of J :

$$J = \mu = \mathbb{E}[J(\mathbf{x}, \alpha)], \quad \text{such that} \quad \sigma = \mathbb{E}[(J(\mathbf{x}, \alpha) - \mu)^2] \leq \sigma_0, \tag{6.6}$$

where $\mathbb{E}[.]$ is the expectation operator.

We can go beyond the first two moments and use also the third and the fourth moments. Going beyond the first two moments is important when the PDF of J deviates from a pure Gaussian distribution. Indeed, even with interval-based (with uniform PDF) or Gaussian entries there is no reason for the PDF of the solution of a simulation to remain uniform or Gaussian.

The third and fourth moments, the skewness γ and the kurtosis κ, are defined as :

$$\gamma = \mathbb{E}[(\frac{J(x, \alpha) - \mu}{\sigma})^3], \quad \kappa = \mathbb{E}[(\frac{J(x, \alpha) - \mu}{\sigma})^4]. \tag{6.7}$$

The skewness is an indication of the deviation from symmetry of the PDF and the kurtosis permits to quantify if it is tall and skinny or short and squat.

One can consider that a robust design should favor symmetry in the distribution which means a lower absolute value of skewness. This is an assumption and other non-zero values can be targeted for the absolute value of the skewness. However, the assumption of symmetry appears reasonable. Indeed, when driving a car on a straight line, one expects the car to have the same behavior for small and symmetrical disruptions in holding the steering wheel by the driver. In a Gaussian distribution we have $\gamma = 0$. Also, in a normal distribution the mean and median coincide and if a PDF is not too far from a normal distribution, the median will be near $\mu - \gamma\sigma/6$. Therefore, if $|\gamma| \to 0$ the PDF tends toward a normal distribution. This provides an inequality constraint on $|\gamma|$ as γ can be either positive or negative. For an unimodal PDF a reduction of the skewness comes when the mean and the mode of the distribution converge toward each other at given standard deviation.

Robust design means higher density near the mean which means higher kurtosis, but this is more subtle. Indeed, despite higher kurtosis means concentration of the probability mass around the mean, it could also imply thicker tails in the PDF. This means that more of the variance is the result of infrequent extreme deviations. We need therefore to define what we mean by robust design : acceptance of frequent modest deviations or acceptance of infrequent extreme ones. If operational security is a major concern the latter should be definitely avoided. We therefore consider that

a reasonable requirement would be to have a design reducing the initial kurtosis value : $\kappa \leq \kappa_0$ together with a constraint on the variance σ.

To summarize, robust moment-based optimization can be seen through the following constrained minimization problem :

$$\begin{gathered} \min_{\mathbf{x}\in\mathbf{O}_{ad}} J(\mathbf{x}) = \mu(\mathbf{x}) \quad \text{such that} \\ \sigma(\mathbf{x}) - \sigma_0 \leq 0, \quad |\gamma(\mathbf{x})| - |\gamma_0| \leq 0, \quad \kappa(\mathbf{x}) - \kappa_0 \leq 0. \end{gathered} \tag{6.8}$$

Equality constraints on the moments are particular cases of these. In cases, higher kurtosis are targeted (see discussion above), the last constraint becomes $-\kappa(\mathbf{x}) + \kappa_0 \leq 0$.

6.3.2 Discrete expressions

Monte Carlo simulations permit to recover these moments with an error decreasing as $\sigma/\sqrt{M}$ with M the number of functional evaluations and this with a convergence rate independent of N. But, for small N, classical numerical integration over-performs Monte Carlo simulations in term of complexity based on the number of functional evaluations to recover the moments at a given accuracy. As we are interested by small values of N, this latter may therefore be preferred.

Both Monte Carlo trials and numerical integration lead to the introduction of weighted sums [32] over a M-point sampling $\mathbf{I}_M$ as estimators of the previous moments (denoted with the same notation) :

$$\mu = \sum_{\alpha_k\in\mathbf{I}_M} \omega_k J(\mathbf{x}, \alpha_k), \qquad \sigma^2 = \sum_{\alpha_k\in\mathbf{I}_M} \omega_k (J(\mathbf{x}, \alpha_k) - \mu)^2, \tag{6.9}$$

$$\gamma = \sum_{\alpha_k\in\mathbf{I}_M} \omega_k (J(\mathbf{x}, \alpha_k) - \mu)^3, \qquad \kappa = \sum_{\alpha_k\in\mathbf{I}_M} \omega_k (J(\mathbf{x}, \alpha_k) - \mu)^4. \tag{6.10}$$

A major difference between Monte Carlo and numerical integration is that in the former the sampling $\mathbf{I}_M$ and the weights are chosen according to the PDF of α. In this paper we consider uniform PDF. We have therefore uniform sampling in each of the dimension of α and $\omega_k \sim 1/M$. Unbiased estimates use slightly different coefficients and also introduce corrections. For instance, one should consider $\omega_k = 1/(M-1)$ for σ, $\omega_k = M/((M-1)(M-2))$ for γ and κ. However, because we are interested by the gradients of these quantities with respect to $\mathbf{x}$ and gradient-based minimization algorithms, these changes will only have slight impacts on the amplitude of the gradient with no real incidence on the optimization history, especially if optimal descent step sizes are used. This discussion also shows that moment-based and multipoint optimization can be seen as a whole. If different PDFs are targeted than uniform, we need to redefine the sampling $\mathbf{I}_M$ and the weights ω_k accordingly. But, this is beyond the scope of this paper.

6.3.3 Gradients of the moments

The linearity in the sums permits to access to the gradients of the moments with respect to the optimization variables $\mathbf{x}$ from the gradient of the functional at the

sampling point α_k. We recall that we took $\sigma = 1$ in Equation (6.7) to ease this linearization. The different gradients read :

$$\begin{aligned}
\nabla_{\mathbf{x}}\mu &= \sum_{\alpha_k \in \mathbf{I}_M} \omega_k \nabla_{\mathbf{x}} J(\mathbf{x}, \alpha_k), \\
\nabla_{\mathbf{x}}\sigma^2 &= \sum_{\alpha_k \in \mathbf{I}_M} 2\omega_k (J(\mathbf{x}, \alpha_k) - \mu) \nabla_{\mathbf{x}} (J(\mathbf{x}, \alpha_k) - \mu), \\
\nabla_{\mathbf{x}}\gamma &= \sum_{\alpha_k \in \mathbf{I}_M} 3\omega_k (J(\mathbf{x}, \alpha_k) - \mu)^2 \nabla_{\mathbf{x}} (J(\mathbf{x}, \alpha_k) - \mu), \\
\nabla_{\mathbf{x}}\kappa &= \sum_{\alpha_k \in \mathbf{I}_M} 4\omega_k (J(\mathbf{x}, \alpha_k) - \mu)^3 \nabla_{\mathbf{x}} (J(\mathbf{x}, \alpha_k) - \mu).
\end{aligned} \tag{6.11}$$

Knowing the elementary gradients $\nabla_{\mathbf{x}} J(\mathbf{x}, \alpha_k)$ at sampling points α_k brings interesting information on the geometry of the problem. Indeed, they can be used to build a vector space and the analysis of this space provides important information on the complexity of the problem [32, 33, 34].

The natural presence of parallelism due to the M independent evaluations of the state, functional and its gradient is interesting to monitor computational complexity. In addition, as mentioned earlier for our problem, we notice that only the right-hand-sides $b_{i=1,\ldots,N}$ need to be assembled after each perturbation of $\alpha = u^*$. Therefore, the optimality condition mentioned in section 6.2 for J can be extended to μ writing :

$$\nabla_x \mu = \sum_{\alpha_k \in \mathbf{I}_M} \omega_k \nabla_{\mathbf{x}} J(\mathbf{x}, \alpha_k) = 0,$$

which gives the following linear system to solve :

$$A\mathbf{x} - \sum_{\alpha_k \in \mathbf{I}_M} \omega_k b_{\alpha_k} = A\mathbf{x} - \sum_{\alpha_k \in \mathbf{I}_M} \omega_k < (u_0 - u^*)\,,\, u >= 0,$$

with the matrix $A_{i,j} =< u_i\,,\, u_j >$ unchanged.

In the presence of constraints on moments, we can proceed in the same way to get the gradients of the moments, again using the nullity of the gradient of the first moment at the optimum :

$$\begin{aligned}
\nabla_{\mathbf{x}}\sigma^2 &= \sum_{\alpha_k \in \mathbf{I}_M} 2\omega_k (J(\mathbf{x}, \alpha_k) - \mu) \nabla_{\mathbf{x}} J(\mathbf{x}, \alpha_k) \\
&= C_1(\mathbf{x}) A\mathbf{x} - \sum_{\alpha_k \in \mathbf{I}_M} 2\omega_k (J(\mathbf{x}, \alpha_k) - \mu) b_{\alpha_k},
\end{aligned}$$

$$\begin{aligned}
\nabla_{\mathbf{x}}\gamma &= \sum_{\alpha_k \in \mathbf{I}_M} 3\omega_k (J(\mathbf{x}, \alpha_k) - \mu)^2 \nabla_{\mathbf{x}} J(\mathbf{x}, \alpha_k) \\
&= C_2(\mathbf{x}) A\mathbf{x} - \sum_{\alpha_k \in \mathbf{I}_M} 3\omega_k (J(\mathbf{x}, \alpha_k) - \mu)^2 b_{\alpha_k},
\end{aligned}$$

$$\begin{aligned}\nabla_{\mathbf{x}}\kappa &= \sum_{\alpha_k\in\mathbf{I}_M} 4\omega_k(J(\mathbf{x},\alpha_k)-\mu)^3\nabla_{\mathbf{x}}J(\mathbf{x},\alpha_k)\\ &= C_3(\mathbf{x})A\mathbf{x} - \sum_{\alpha_k\in\mathbf{I}_M} 4\omega_k(J(\mathbf{x},\alpha_k)-\mu)^3 b_{\alpha_k}.\end{aligned}$$

We see that these are four vectors involving a same matrix A and elementary contributions b_{α_k} with different weightings and scaling C_1, C_2 and C_3 which are defined by :

$$C_1(\mathbf{x}) = \sum_{\alpha_k\in\mathbf{I}_M} 2\omega_k(J(\mathbf{x},\alpha_k)-\mu),$$

$$C_2(\mathbf{x}) = \sum_{\alpha_k\in\mathbf{I}_M} 3\omega_k(J(\mathbf{x},\alpha_k)-\mu)^2,$$

$$C_3(\mathbf{x}) = \sum_{\alpha_k\in\mathbf{I}_M} 4\omega_k(J(\mathbf{x},\alpha_k)-\mu)^3.$$

6.3.4 Controlling the moments

A classical approach to improve robust design in engineering is to control the first moment under the constraint of leaving unchanged or even reduce the second one. An easy way to adapt this concept to our situation is to consider the following descent direction orthogonal to the gradient of the variance :

$$d = -\left(\nabla_{\mathbf{x}}\mu - < \nabla_{\mathbf{x}}\mu, \frac{\nabla_{\mathbf{x}}\sigma}{\parallel \nabla_{\mathbf{x}}\sigma \parallel} > \frac{\nabla_{\mathbf{x}}\sigma}{\parallel \nabla_{\mathbf{x}}\sigma \parallel}\right),$$

where $\nabla_{\mathbf{x}}\sigma/ \parallel \nabla_{\mathbf{x}}\sigma \parallel$ is the unit vector along $\nabla_{\mathbf{x}}\sigma$. Unfortunately, this formulation does not permit anymore to get the optimal solution directly inverting a linear system as it is the case in the single-point or even when minimizing solely the first moment. An iterative procedure is then necessary using some descent algorithm. We will see examples of these below.

One can generalize the previous construction to the case of more than two moments using a Gram-Schmidt orthonormalization procedure. Let us organize the four vectors in the following order :

$$(\nabla_{\mathbf{x}}\kappa, \nabla_{\mathbf{x}}\gamma, \nabla_{\mathbf{x}}\sigma^2, \nabla_x\mu). \tag{6.12}$$

To reduce the impact of the solution of the first order optimality condition for the first moment on the higher moments, we require the descent direction to be orthogonal to the subspace generated by the the gradients of these moments. This can be easily achieved using a Gram-Schmidt orthonormalization which extracts from the set of the vectors above a new set of orthonormal vectors $v_{i=1,\dots,4}$:

$$(v_1, v_2, v_3, v_4).$$

The descent direction we use in minimization is defined by :

$$d = -\|\nabla_{\mathbf{x}}\mu\| \; v_4.$$

In this way, we make sure the descent direction is orthogonal to the subspace generated by moments 2, 3 and 4. The organization of the vectors as given in (6.12) is important to make sure the direction monitoring optimization is the one given by the first moment. Once this is achieved, this descent direction can be used in an iterative procedure which is again necessary as the constrained optimality condition cannot simply be cast into a single linear system.

The examples below show that this choice of search direction leads to a reduction of the second and the fourth moments as well. This is positively surprising from a robust optimization view point as one would have actually expected the higher moments to be rather conserved.

6.3.5 Multi-point optimization for a quadratic functional

Let us illustrate our ingredients on a simple example. The functional involves a least-squares minimization :

$$J(\mathbf{x}, \alpha) = \frac{1}{2}\sum_{i=1}^{N}(x_i - \alpha)^2, \quad -0.5 \leq \alpha \leq 0.5, \;\; -5 \leq x_{i=1,\ldots,N} \leq 5, \;\; N = 40. \quad (6.13)$$

Let us solve problem (6.8) with a gradient method. Here, α is a scalar and for a given α the optimality condition for $J(\mathbf{x}, \alpha)$ gives obviously $\mathbf{x}^* = (\alpha, ..., \alpha) \in \mathbb{R}^N$. This would be the solution of a single-point optimization. Here α has a uniform PDF and we use a uniform sampling $\mathbf{I}_M$ with $M = 40$. Minimizing the mean μ without any constraint on the other moments still produces a non robust optimum as the performance has large variability over the range of the operating condition α. On the other hand, asking for the first moments to decrease under constraints on the other three produces a much flatter functional over the range of α. Figure 6.3 shows the histograms of $J(\mathbf{x}^*, \alpha_{k=1,\ldots,M})$ for these minimization. In particular, one sees how controlling more moments affects the left peak and the distribution of the frequencies.

6.3.6 Application to our 3D elastic problem

We now illustrate the multi-point discussion with moments on our 3D elasticity problem presented in section 6.2.1 where target displacement u^* is assumed uncertain with its Probability Density Function (PDF) known. The target is a large vector with a size given by the number of the 3D mesh points. We consider centered Gaussian perturbations of u^* with a standard deviation of 10%.

In our elasticity optimization problem, $\mathbf{O}_{ad} = (\mathbb{R}^+)^N$. This means that each of the optimization variable needs to be positive. This brings additional constraints to those expressed earlier on the moments. This also means that one cannot solve anymore the single-point optimization by a sole inversion of a linear system and

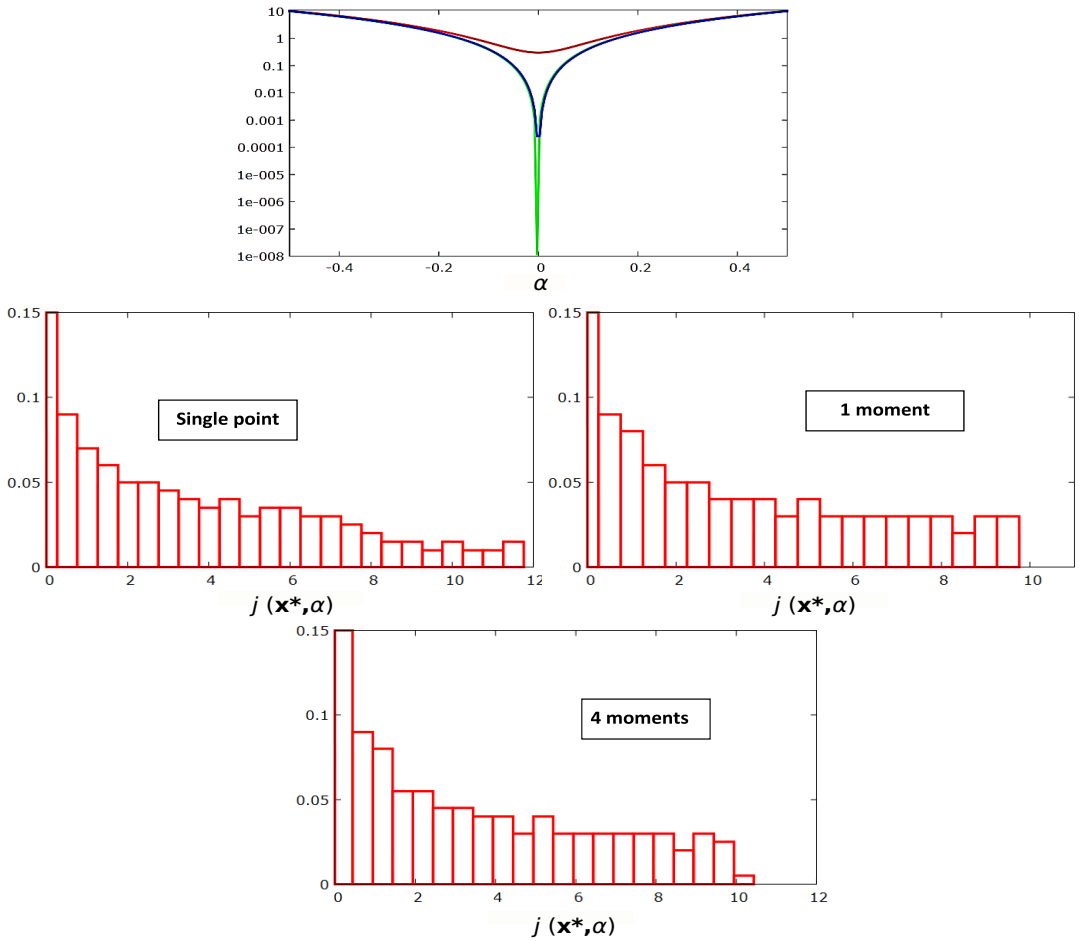

FIGURE 6.3 – Upper : functional $J(\mathbf{x}^*, \alpha)$ given by Equation (6.13) vs. α for single-point optimization, mean-based and with constraints on moments two, three and four. Lower : histograms of $J(\mathbf{x}^*, \alpha_{k=1,\ldots,M})$ for a single-point minimization and when controlling one and then all first four moments.

that, for instance, a projected gradient algorithm is necessary. This can be easily included during the iterations when minimizing the first moment under constraints on the other three.

Figure 6.4 shows the distribution of the ratio of the functional $J(\mathbf{x}^*, \alpha_k)$ to the Young modulus for $M = 1000$. One sees that the uncertainty on u^* brings large variations in J. This variability shows the importance of a robust optimization approach for this problem. The impact of considering the variability during minimization through moment based functional can be easily demonstrated looking at the moment-based solutions of the minimization versus the solution of the single-point optimization for the target u^* which corresponds to the deterministic situation. Table 6.1 features the solution of the optimization variables for this comparison. We see that the results are clearly different. Also the histograms in Figure 6.4 show that controlling all four moments greatly reduces the variability of the functional and therefore improves the robustness of the design. This can be confirmed considering the 4 moments showing the variability of the functional induced by the variability of the target displacement, considered as uncertain, for the single-point optimal solution and for the different optimal solutions obtained when controlling respectively the first, second and eventually all four moments. The first column in Table 6.2 show the first moment for $J(\mathbf{x}, \alpha)/E$ and the other columns show the second, third and fourth moments for $(J(\mathbf{x}, \alpha) - \mu)/E$. By extension, we denote these $\mu(J(\mathbf{x}, \alpha)/E)$, $\sigma((J(\mathbf{x}, \alpha) - \mu)/E)$, $\gamma((J(\mathbf{x}, \alpha) - \mu)/E)$ and $\kappa((J(\mathbf{x}, \alpha) - \mu)/E)$. The first column indicates that the third first designs give about the same average behavior while controlling all four moments clearly provide a different, and actually better, solution. Robustness improvements can be seen from the evolution of the second and fourth moments where again controlling all four moments is recommended. Also, the final density functions are slightly less symmetric when controlling more moments. However, one regain symmetry when controlling 4 moments instead of only two. As we expressed in section 6.3.1, symmetric distributions should be preferred in robust designs.

	Single-point	1 moment	2 moments	4 moments
x_1 (KN)	5500	5496	5266	7035
x_2 (kN)	4000	4000	3917	9262
x_3 (KN)	6000	5994	5518	6834
x_4 (KN)	4500	4503	4330	0
x_5 (KN)	4999	5005	5594	1772

TABLE 6.1 – Optimization variables for minimization using the deterministic target u^* (the single-point case) and when controlling 1, 2 and all 4 moments of J. The solutions are clearly different showing the impact of moment-based optimization. The robustness achieved with more moments accounted for in the design is well illustrated through the histograms in Figure 6.4.

	$\mu(J/E)$	$\sigma((J-\mu)/E)$	$\gamma((J-\mu)/E)$	$\kappa((J-\mu)/E)$
single point	74.980	0.938	-0.005	2.422
1 moment	74.940	0.865	-0.018	2.180
2 moments	74.144	0.840	0.094	2.146
4 moments	64.138	0.765	0.032	1.685

TABLE 6.2 – Distributions of the different moments for the optimal solutions obtained after a single-point optimization and when controlling first, second and eventually all four moments. This final design is superior both in performance and robustness.

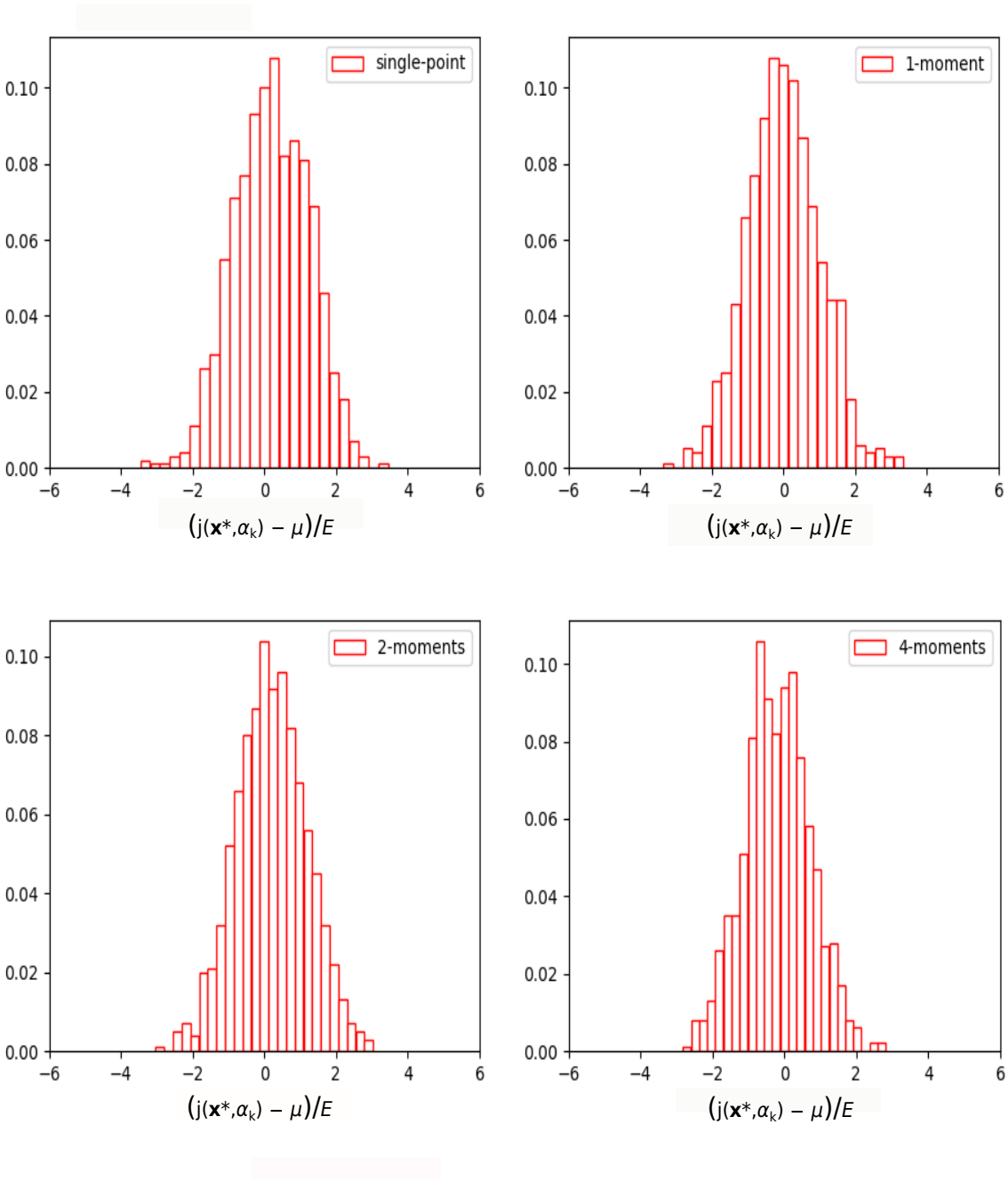

FIGURE 6.4 – Relative frequency histogram of $(J(\mathbf{x}^*, \alpha_{k=1,\ldots,M}) - \mu)/E$ with $M = 1000$. Controlling all four moments reduces the variability of the functional and therefore improves the robustness of the design.

6.4 Linear models and backward uncertainty propagation : covariance matrix of the optimization variables

We showed how to quantify the impact of target state uncertainties in inverse design through multi-point analysis and showed how it can be monitored through the moments of the functional. But the multi-point analysis requires a sampling of a large dimensional space. Something which is a burden when this dimension is large. To avoid a sampling we would like to see if an alternative low-complexity approach based on the existence of a local linear relationship between the target state u^* and the optimization variables $\mathbf{x}$ can be applied to estimate the covariance matrix $Cov^2_{\mathbf{x}}$ of the optimization variables. We compare this construction to a reference $Cov^1_{\mathbf{x}}$ construction where a sampling is necessary.

As previously, we assume the PDF of the target solution u^* known and therefore its covariance matrix Cov_{u^*}. Also, we assume u^* is admissible and that there exists a set of optimization variables realizing the target.

As mentioned in (6.1), in our problem of interest the state equation is linear and to simplify the notation, and without any loss of generality, we consider the source term $f(\mathbf{x}) = \mathbf{x}$ considering that $\mathbf{x}$ and u have a same dimension. In our implementation, however, we use spatial localization terms given in (6.2) in order to reduce the size of $\mathbf{x}$.

The construction can be applied in several useful situations. Let us describe two of them.

6.4.1 $Cov^1_{\mathbf{x}}$ when a linear model $L_1 u = \mathbf{x}$ is available

This is typically our case, for instance, with the elasticity equation and the optimization variables acting on the right-hand-side of the equation. In this case, we can express $Cov^1_{\mathbf{x}}$ with $u \sim u^*$, through :

$$\begin{aligned} Cov^1_{\mathbf{x}} &= \mathbb{E}\left((\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^\top\right) \\ &= L_1 \;\; \mathbb{E}\left((u^*-\overline{u^*})(u^*-\overline{u^*})^\top\right) \;\; L_1^\top \\ &= L_1 \;\; Cov_{u^*} \;\; L_1^\top. \end{aligned}$$

But, in our case, the solver is a black-box and we do not have L_1 in hand. In these cases, a linear model can be built through a machine learning approach using a database of simulation scenarios couples of $(\mathbf{x}-u)$ we denote by $(X-U)$ generated by this code. This procedure can be applied in nonlinear situations as well. In these cases, as the uncertainty analysis is aimed to take place around a given/optimal solution, it is enough for the surrogate model to have a local domain of validity.

L_1 is solution of the first-order optimality condition for a functional $J(L_1)$:

$$\nabla_{L_1} J(L_1) = 0 \Leftrightarrow UU^\top L_1^\top = UX^\top,$$

where U and X gather the scenarios of simulations available in the database for u and $\mathbf{x}$ and we look for the best linear relation $X^\top = U^\top L_1^\top$ linking the scenarios in U to those in X and, actually, vice versa. $L_1^\top$, and therefore L_1, can be found as solution of a least-squares problem :

$$J(L_1) = (X^\top - U^\top L_1^\top, X^\top - U^\top L_1^\top) = (X, X) - 2(XU^\top, L_1) + (L_1, L_1)_{UU^\top}.$$

The only requirement is for U to have full rank in order for $UU^\top$ to be positive definite such that the scalar product $(L_1, L_1)_{UU^\top} \geq 0$. Our formulation presented in section 6.2, and based on the introduction of a database of elementary solutions, can be seen as a particular case of machine learning taking advantage of the linearity to express $\mathbf{x} = A^{-1}b(u^*)$. We recall that the components of A and b for $i, j = 1, ..., N$ are given by :

$$A_{i,j} =< u_i \, , \, u_j > \quad \text{and} \quad b_i(u^*) =< (u_0 - u^*) \, , \, u_i > .$$

Again, the covariance matrix of the optimization variables reads :

$$\begin{aligned} Cov^1_{\mathbf{x}} &= \mathbb{E}\left((\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^\top\right) \\ &= A^{-1} \;\; \mathbb{E}\left((b(u^*) - b(\overline{u^*}))(b(u^*) - b(\overline{u^*}))^\top\right) \;\; A^{-\top} \\ &= A^{-1} \;\; Cov_{b(u^*)} \;\; A^{-\top}, \end{aligned} \tag{6.14}$$

where it is easy to estimate $Cov_{b(u^*)}$ knowing Cov_{u^*}. The elementary u_i and the homogeneous solutions are deterministic. The covariance matrix of $b(u^*)$ measures the covariance of the projection of the deviation between the uncertain target function u^* from the homogeneous solution u_0 on the elementary solutions u_i :

$$\begin{aligned} (Cov_{b(u^*)})_{ij} &= \mathbb{E}\left((b_i(u^*) - \overline{b_i(u^*)}) \; (b_j(u^*) - \overline{b_j(u^*)})^\top\right) \\ &= \mathbb{E}\left((< (\overline{u^*} - u^*) \, , \, u_i >) \; (< (\overline{u^*} - u^*) \, , \, u_j >)^\top\right). \end{aligned}$$

One interest of this formulation is that the size of this matrix is now five, the number of the optimization variables, instead of 14400, the number of the mesh grid points for Cov_{u^*}. Also, if the perturbation on u^* are centered, $\overline{u^*}$ is the deterministic target solution.

6.4.2 $Cov^2_{\mathbf{x}}$ when a linear model $L_2\mathbf{x} = u$ is available

This is not directly the situation in this paper as the optimization variables intervene in the left-hand-side. However, this analysis is powerful when the state equation is linear, or if the Jacobian of the state with respect to the optimization can be estimated which is our situation where indeed the Jacobian is explicitly available knowing the elementary solutions as described in section (6.2).

After a deterministic optimization when the state is close to the target ($u \sim u^*$), we assume the covariance matrix of u close to the covariance matrix of u^*. Using

the linear relationship $L_2\mathbf{x} = u$ we can write :

$$\begin{aligned} Cov_{u^*} &= \mathbb{E}\left((u^* - \overline{u^*})(u^* - \overline{u^*})^\top\right) \\ &= \mathbb{E}\left(L_2 \ \ (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^\top \ \ L_2^\top\right) \\ &= L_2 \ \ \mathbb{E}\left((\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^\top\right) \ \ L_2^\top \\ &= L_2 \ \ Cov_{\mathbf{x}}^2 \ \ L_2^\top. \end{aligned}$$

Unlike the previous situation, to get $Cov_{\mathbf{x}}^2$ we need to invert this expression. In cases where exact inversion is impossible a least-squares formulation can be used looking for $Cov_{\mathbf{x}}^2$ minimizing :

$$\frac{1}{2} < L_2 \ \ Cov_{\mathbf{x}}^2 \ \ L_2^\top, L_2 \ \ Cov_{\mathbf{x}}^2 \ \ L_2^\top > - < Cov_{u^*}, L_2 \ \ Cov_{\mathbf{x}}^2 \ \ L_2^\top > .$$

First-order optimality condition with respect to $Cov_{\mathbf{x}}$ gives :

$$L_2^\top L_2 \ \ Cov_{\mathbf{x}}^2 \ \ L_2^\top L_2 - L_2^\top \ \ Cov_{u^*} \ \ L_2 = 0,$$

which leads to

$$Cov_{\mathbf{x}}^2 = (L_2^\top L_2)^{-1} \ \ L_2^\top \ \ Cov_u \ \ L_2 \ \ (L_2^\top L_2)^{-1},$$

and eventually, to

$$Cov_{\mathbf{x}}^2 = L_2^{-1} \ \ Cov_{u^*} \ \ L_2^{-\top} = \left(L_2^\top \ \ {Cov_{u^*}}^{-1} \ \ L_2\right)^{-1}. \tag{6.15}$$

Therefore, knowing Cov_{u^*} one can access $Cov_{\mathbf{x}}^2$ if a linear relationship, even locally valid, is available between $\mathbf{x}$ and u. In practice, Cov_{u^*} is diagonal as the target uncertainties are often assumed independent. The inversion of Cov_{u^*} is therefore straightforward. The second expression only requires the inversion of a $N \times N$ matrix in our case. This expression in (6.15) is also interesting because it involves the inversion of a square matrix giving a least-squares sense to the inversion of rectangular matrix. An interesting application of this construction is using the state sensitivity with respect to optimization variables or the Jacobian $\mathcal{J} = (\partial u / \partial \mathbf{x})$ in a first-order linear relationship $\mathcal{J} \ \delta\mathbf{x} = \delta u$. This permits to estimate the covariance matrix of the optimization variables perturbations even in the absence of a direct linear model linking $\mathbf{x}$ and u. Actually, in situations where the state equation is linear (our case) we have $L_2 = \mathcal{J}$. In situations where there are only a few optimization variables, the Jacobian can simply be estimated by a finite differences approach approximating the ith line of the Jacobian by :

$$\mathcal{J}_{i=1,\ldots,N} =\sim \frac{(u(\mathbf{x}e_i + \varepsilon) - u(\mathbf{x}e_i - \varepsilon))}{2\varepsilon},$$

with $e_{i=1,\ldots,N}$ the canonical basis of $\mathbb{R}^N$ in our case and $0 < \varepsilon << 1$. But, we have seen in section 6.2 that u has a direct expression with respect to $\mathbf{x}$ components x_i through :

$$u = u_0 + \sum_{i=1}^{N} x_i u_i.$$

Therefore, the finite difference formula reduces to :

$$\mathcal{J}_{i=1,\dots,N} \sim \frac{(u(\mathbf{x}e_i + \varepsilon) - u(\mathbf{x}e_i - \varepsilon))}{2\varepsilon} = u_i.$$

This is another advantage of our direct formulation for the inversion as it makes evaluating $\mathcal{J}$ straightforward, not requiring any extra calculation. In addition, the Jacobian is exact in this case and is not affected by the finite differences approximation and the choice of ε. Actually, the Jacobian would be exact even using first order finite differences. This is because we recover the exact gradient expression in this case. Therefore, this second covariance estimation is well adapted to our situation and would be interesting if it gives nearly the same answer than the first construction where a sampling of the state variable variation domain is necessary.

Figure 6.5 shows the comparison of this two estimations of $Cov_{\mathbf{x}}$. We see that the matrices are very close and especially the signs of all components agree.

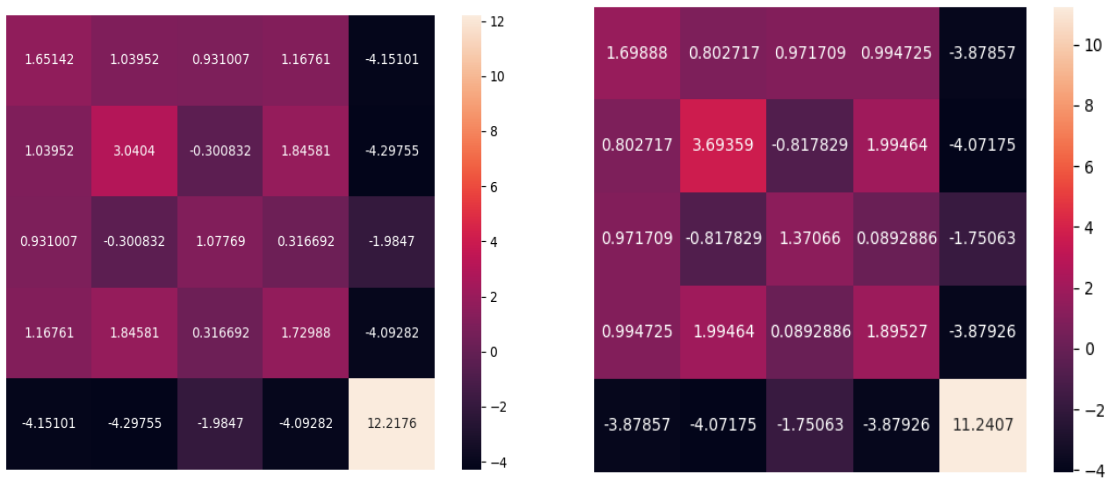

FIGURE 6.5 – Covariance Matrix $Cov^1_{\mathbf{x}}$ and $Cov^2_{\mathbf{x}}$ (in $Newton^2$) Visualization, left : $Cov^1_{\mathbf{x}}$from (6.14), right : $Cov^2_{\mathbf{x}}$ from (6.15).

Further comparisons are given in Figure 6.3 through different distances between the two covariance matrices versus the distance between the covariance matrices and a diagonal matrix with the diagonal given by mean values of the covariance matrices components (i.e. $\frac{1}{2N^2}\sum_{i,j=1}^{N}(Cov^1_{\mathbf{x}} + Cov^2_{\mathbf{x}})_{i,j}$). This is to provide an estimation of how close the two covariance matrices are versus a third fixed one.

Distance $\mathbf{d}$	$\mathbf{d}(Cov^1_{\mathbf{x}}, Cov^2_{\mathbf{x}})$	$\mathbf{d}(Cov^1_{\mathbf{x}}, CI_N)$	$\mathbf{d}(Cov^2_{\mathbf{x}}, CI_N)$
Kullback distance	0.334	165.589	166.682
Euclidean distance	1.674	17.020	16.123
Log Euclidean distance	0.737	7.193	7.469
Riemannian distance	0.811	7.193	7.469
Wasserstein distance	0.389	4.098	4.097
Log-det distance	0.286	2.00	2.086

TABLE 6.3 – Different estimations of distances between the two covariance matrices. where $C = \frac{1}{2N^2}\sum_{i,j=1}^{N}(Cov^1_{\mathbf{x}} + Cov^2_{\mathbf{x}})_{i,j}$ is the mean values of the covariance matrices components and I_N is the Nth identity matrix. One notices that, regadless of the considered distance, $Cov^1_{\mathbf{x}}$ and $Cov^2_{\mathbf{x}}$ are about at the same distance of CI_N. Also, $Cov^1_{\mathbf{x}}$ are $Cov^2_{\mathbf{x}}$ are at a distance about ten times less except for the Kullback distance where two matrices are seen much closer. This is very interesting as this distance is particularly suitable for the comparison between probability density functions and covariance matrices.

6.5 Concluding remarks

Robust parametric optimization has been addressed through the control of the moments of the functional. Moments have been estimated using a multi-point formulation. This permits to address situations where operating parameters are not anymore single valued but defined through their PDF. Hence, the first four moments of the functional have been considered to go beyond the classical mean-variance based optimization. Subtleties of what should be the target for the skewness and the kurtosis of the probability density function of the performance of the system for a robust design have been discussed.

It has been shown that controlling second, third and fourth moments drastically improves the design while the time to the solution remains comparable to the single-point situation as all the extra calculations can be carried out in a fully parallel and independent manner. The cost is also reduced thanks to our construction of the optimal solution taking advantage of the presence of the linearity between the optimization variables and the solution of the state equation. As a consequence, the approach requires quite small additional coding and computational effort.

Low-complexity solutions for backward propagation of aleatory uncertain target data has also been presented. Derivation of the covariance matrix of the optimization parameters has been discussed through two formulations. These provide uncertainty quantification analysis for the inversion solution with confidence margins on the design parameters in very large design spaces. It has been shown that taking again advantage of existing linear relationships between the target state u^* and the optimization variables $\mathbf{x}$, samplings of the design and the target solution spaces can be avoided drastically reducing the computational cost of the approach.

Chapitre 7

Conclusion et perspectives

Le but de cette thèse était de proposer une méthode directe d'optimisation de la répartition de charges mécaniques sur une structure pour en assurer une déformation minimale ou prescrite.

Nous avons vu comment mettre en évidence un problème linéaire qui produit la solution optimale au sens des moindres carrées et ceci sans aucune méthode itérative de type gradient. Et, nous avons montré comment cette approche pouvait simplement être mise en œuvre avec des logiciels boîtes-noires en utilisant le parallélisme grâce à l'indépendance totale des pré-calculs des solutions élémentaires.

L'analyse de l'effet du contrôle d'erreur de discrétisation de l'équation d'état par l'adaptation de maillage sur le résultat du problème d'optimisation a démontré la nécessité de disposer de solutions précises pour l'équation d'état. En effet, nous avons vu que la solution du problème d'optimisation pouvait changer grandement, ainsi que les valeurs des paramètres d'optimisation mettant en évidence la nécessité de contrôler l'erreur de discrétisation avant de pouvoir accorder une certaine confiance à la solution du problème d'optimisation.

Nous avons montré comment une analyse de sensibilité pouvait être introduite en utilisant avantageusement la dépendance linéaire de la fonctionnelle par rapport aux variables d'optimisation. Cette analyse a permis de propager en directe et en rétrograde les incertitudes sur la solution cibles et ainsi apporter des réponses aux questions de robustesse. En effet, dans les problèmes pratiques, la solution cible peut être, non pas déterministe, mais uniquement connue à travers des intervalles d'admissibilité, par exemple.

Cette approche peut trouver des extensions et applications multiples, notamment en optimisation discrète des points d'intervention ou bien pour l'estimation temps-réel des distributions de charges.

Chapitre 8

Annexe : calcul élastique avec Cast3M

On cherche ici à étudier numériquement une dalle en Béton avec des corps creux.

Nous avons utilisé le logiciel `Gmsh` [15] pour la création du géométrie et la génération de maillage tétraédrique et `Cast3M` [28] pour les calculs :

1. **Création de la géométrie :** pour réaliser une dalle avec corps creux on crée d'abord la dalle après on crée le premier corps creux et on le place tout au longue de la dalle (après les modifications nécessaires) et à la fin on enlève les corps creux de la dalle.

   Gmsh peut être utiliser de deux manières :

   -Soit d'une manière interactive en utilisant l'interface graphique.

   -Soit dans des fichiers texte en utilisant le langage de script propre à Gmsh, on est besoin d'ajouter au début de fichier : SetFactory("OpenCASCADE") ; pour qu'on peut utiliser les fonctions prédéfini d'Opencascade telles que : Box, Sphere,...

   La dalle : pour créer la dalle de dimension $l \times L \times h$ on a utilisé la commende Box (on peut aussi la créer à partir des points, des lignes et des surfaces) ; dalle_EXT = newv ; Box(dalle_EXT) = {0, 0, 0, l, L, h} ; la figure 8.1 montre le résultat obtenu.

   Le corps-creux : pour créer le corps creux on a suivi les étapes :

   - création des points (p0,p1,p3,p4,p5,p6) avec :
     p0 = newp ; Point(p0) = {0., 0., 0.} ;
     p1 = newp ; Point(p1) = {0., 0., 1.} ;
     p2 = newp ; Point(p2) = {0., 0.6, 1.} ;
     p3 = newp ; Point(p3) = {0., 0.6, 0.} ;
     p4 = newp ; Point(p4) = {0., 1. , 0.5} ;
     p5 = newp ; Point(p5) = {0., 0.95 , 0.75} ;
     p6 = newp ; Point(p6) = {0., 0.95 , 0.25} ;

   - par la commende 'Line' on construit les lignes (l1,l2,l3) à partir des points précédents :

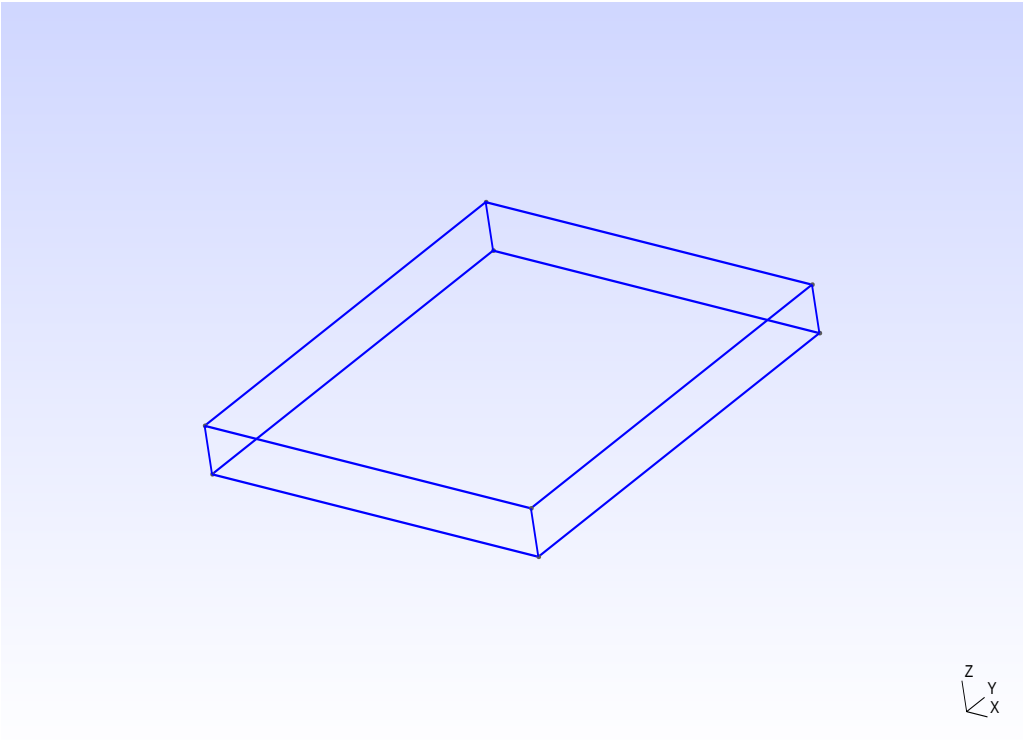

FIGURE 8.1 – dalle pleine

l1 = newl ; Line(l1) = {p1, p2} ;
l2 = newl ; Line(l2) = {p0, p3} ;
l3 = newl ; Line(l3) = {p1, p0} ;

- et on construit la courbe spline qui passe par (p3,p6, p4,p5,p2) avec :
  l4 = newl ; Spline(l4) = {p3, p6, p4,p5, p2} ;
- on définit la surface 'surf1' comme étant la surface délimitée par le conteur 'cll1' (l'ensemble de l1, l2, l3 et l4) ;
  cll1 = newll ; Curve Loop(cll1) = l4, -l2, -l3, l1 ;
  surf1 = news ; Plane Surface(surf1) = {cll1} ; la figure 8.2 montre le résultat obtenu.
- on fait une rotation à 'surf1' selon l1 d'un angle π et $-\pi$ et on fait l'union pour obtenir le corps creux 'coque' ;
  coque = newv ; BooleanUnion(coque) = { Volume{1} ; Delete ;}{ Volume{2} ; Delete ; } ; la figure 8.3 montre le résultat obtenu.

Remarque : newp, newl, newl,..., permet de donner des noms aux entités géométriques.

dalle-creuse :

-on applique sur le coque une homothétie de facteur 'dilation_factor' et par translation on obtient les autres corps(voir la figure) : Translate {4, 4, 0 } { Duplicata { Volume{sphere} ;}}

-on enlève les corps-creux de la dalle avec :

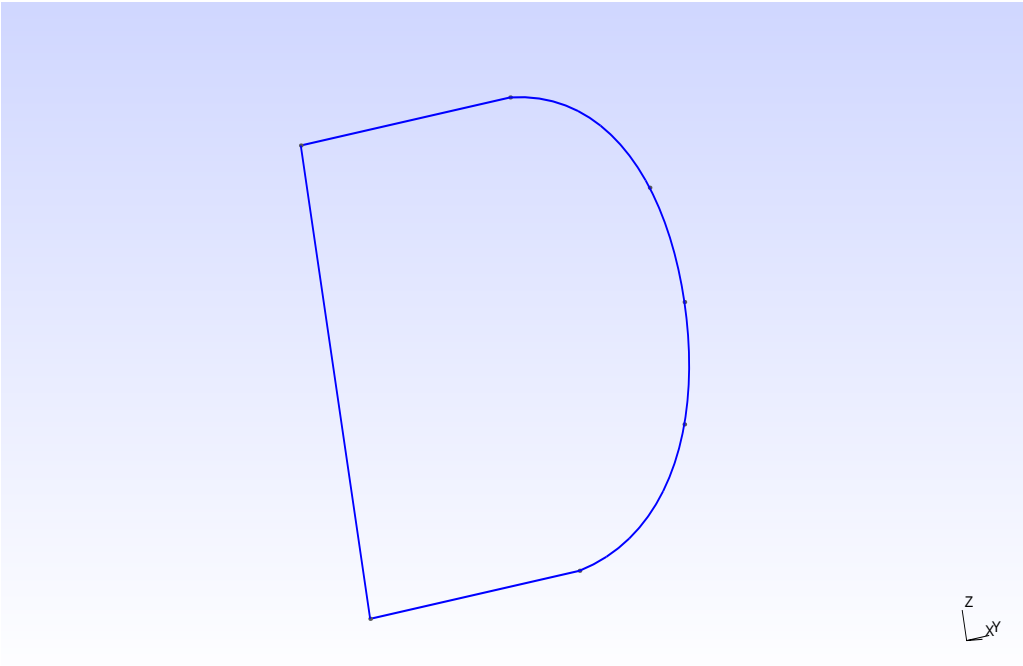

FIGURE 8.2 – Surface surf1

dalle_creuse = newv ; BooleanDifference(dalle_creuse) = { Volume{dalle} ; Delete ;}{ Volume{sphere :sphere+8} ; Delete ; } ; la figure 8.4 montre le résultat obtenu.

2. **Maillage :** Avec Gmsh il y a la possibilité de choisir les algorithmes de maillage, spécifier la taille de maille,. . . ;

   -Depuis le fichier .geo ; par exemple pour définir la taille de maillage à 0.1 on rajoute au début du .geo :
   Mesh.CharacteristicLengthMin = 0.1 ;
   Mesh.CharacteristicLengthMax = 0.1 ;
   et on crée le fichier maillage depuis le terminal telle que : gmsh -(1,2,3) fichier.geo -o fichier.(msh, unv,. . .)
   où : -1, (-2, -3) : lance le maillage en 1D, 2D
   -o fichier.(msh, unv,..) : nom et format de fichier de sortie.

   -Ou bien depuis l'interface graphique ; exemple pour spécifier la taille de maillage : Tools → Options → Mesh → General → Min/Max element size

3. **calcul :** mise en œuvre de la méthode des éléments finis, résolution du problème discrétisé. On a le code Cast3M :

   *option de calcul
   OPTI DIME 3 ;
   H = 1. ;
   L = 10. ;
   l = 10. ;
   IF = -5.E6 ;

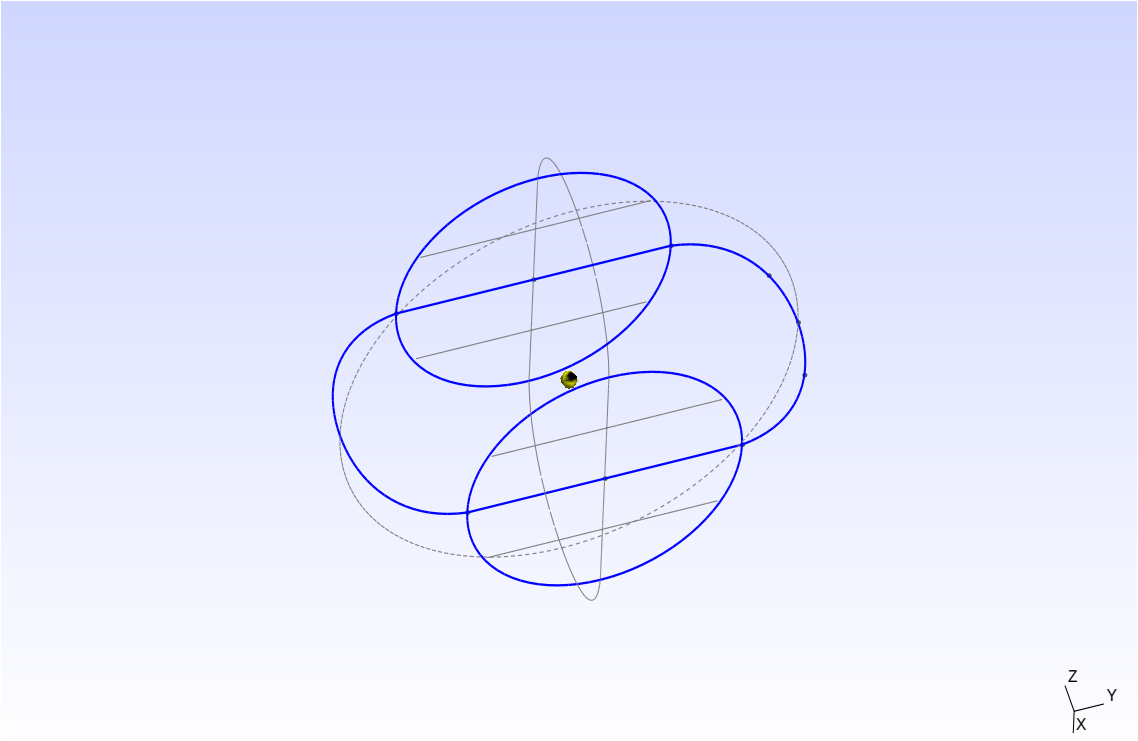

FIGURE 8.3 – corps-creux

```
*Importation de maillage
TAB1 = LIRE 'UNV' 'dalle-creuse.unv';
MAIL1 = TAB1.'MAILLAGE';
MAIL = MAIL1 ELEM 'TET4';
* Modele et Materiau ELASTIQUE
MOD = 'MODE' MAIL MECANIQUE ELASTIQUE;
MAT = 'MATE' MOD 'YOUN' 3.E+10 'NU' 0.2 'RHO' 2300. 'ALPH' 1.2E-
5;
*Coditions aux limites
POIN1 = 0. 0. 0.; POIN2 = 0. L 0.;POIN3 = l L 0.; POIN4 = l 0. 0.;
P1 = MAIL POIN 'PROC' POIN1;
P2 = MAIL POIN 'PROC' POIN2;
P3 = MAIL POIN 'PROC' POIN3;
P4 = MAIL POIN 'PROC' POIN4;
BL1 = BLOQ DEPL ROTA P1;
BL2 = BLOQ DEPL ROTA P2;
BL3 = BLOQ DEPL ROTA P3;
BL4 = BLOQ DEPL ROTA P4;
*Chargement
POIN5 = 5. 5. H;
```

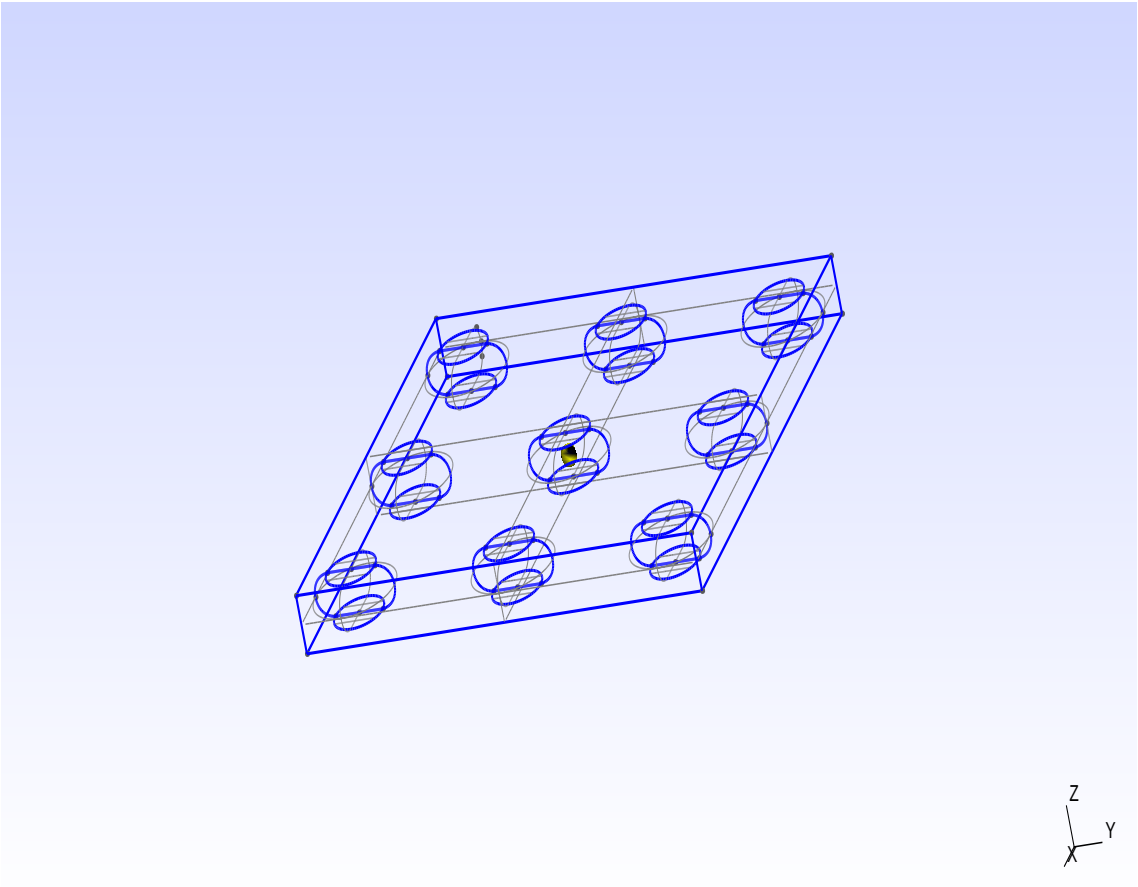

FIGURE 8.4 – Géométrie de la dalle creuse.

```
P5 = MAIL POIN 'PROC' POIN5 ;
FR = FORCE FZ IF P5 ;
*Matrice de rigidité
RI1 = RIGI MOD MAT ;
RI = RI1 ET BL1 ET BL2 ET BL3 ET BL4 ;
*Resolution
U = RESO RI FR ;
UZ = EXCO U UZ ;
*Post-traitement DEF = DEFO MAIL U 150. ;
* Trace déplacement sur configuration déformée
TRAC UZ DEF ;
FIN ;
```

La figure 8.7 montre les déplacements obtenu depuis ce code Cast3M.

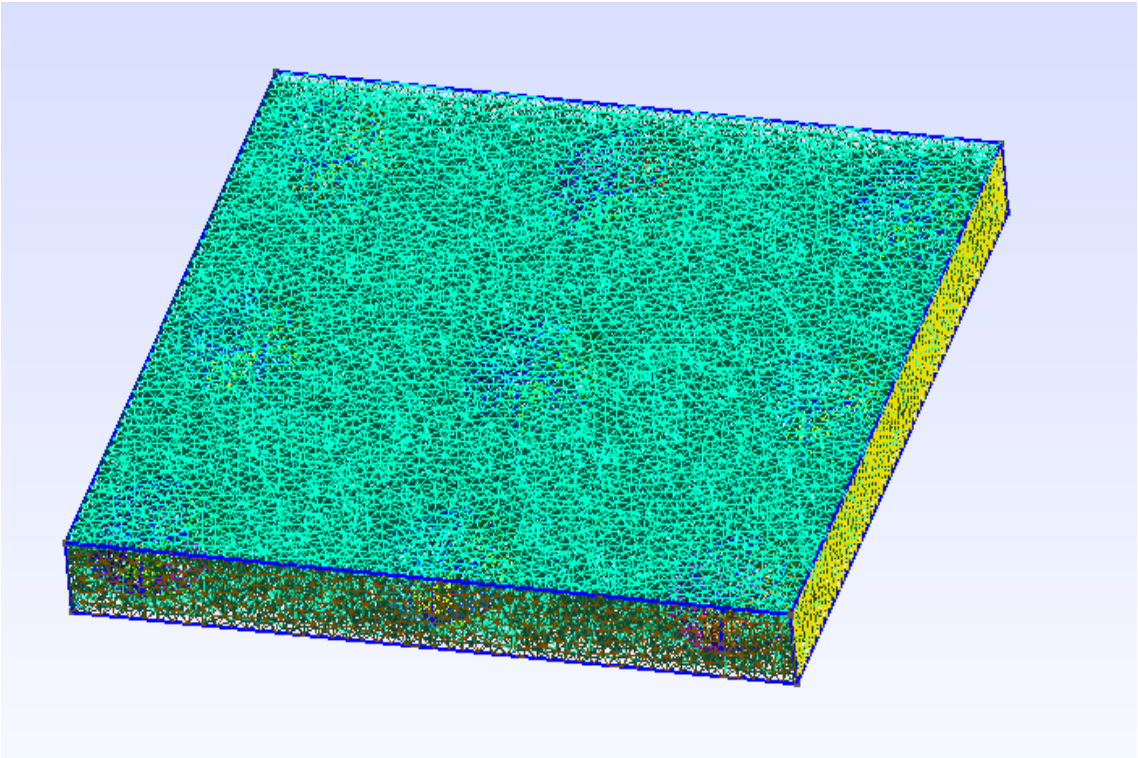

FIGURE 8.5 – Maillage dalle.

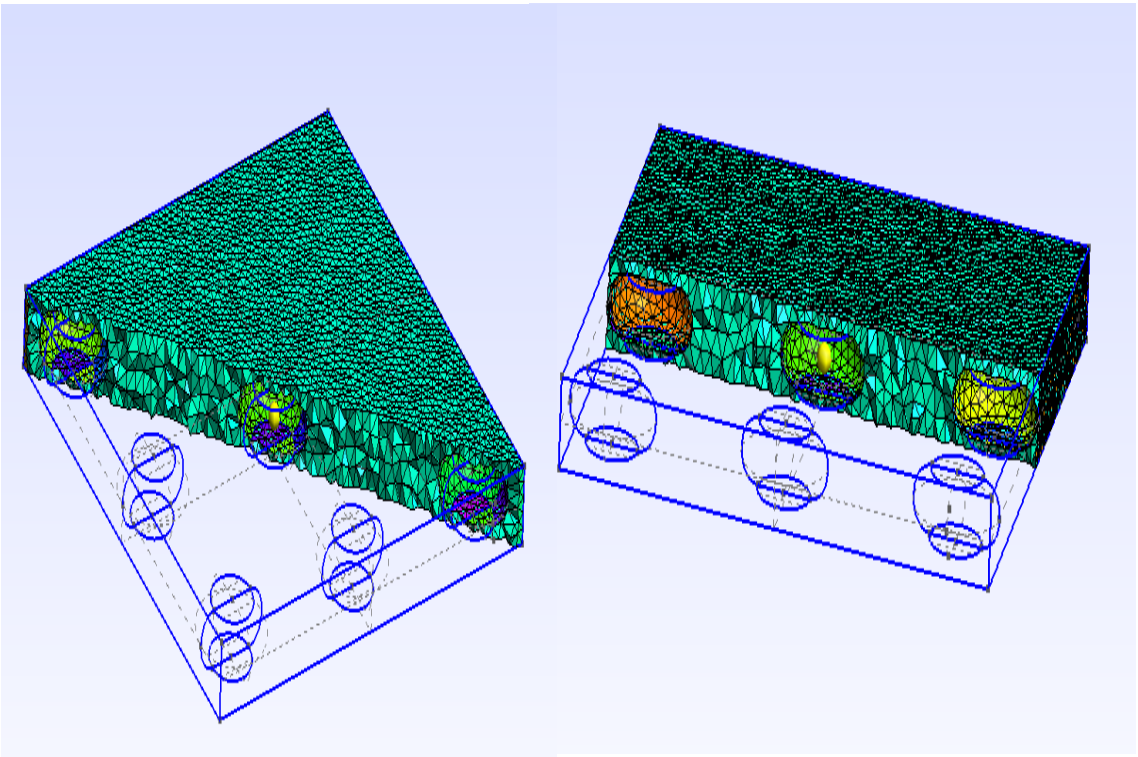

FIGURE 8.6 – Une coupe de maillage.

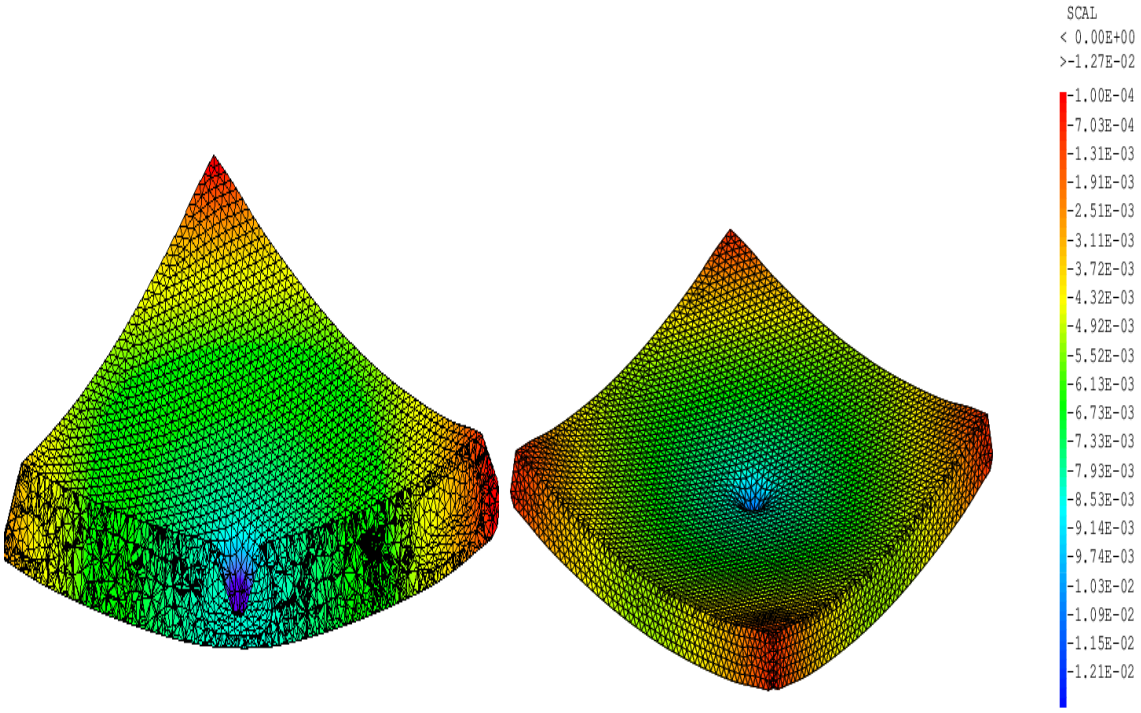

FIGURE 8.7 – Déplacement UZ,

Bibliographie

[1] K. Scheinberg A. Conn and L. Vicente. *Introduction to Derivative-Free Optimization.* SIAM, NY, 2002.

[2] Addison–Wesley. Designing and building parallel programs.

[3] AIAA. Guide for the verification and validation of computational fluid dynamics simulations. *AIAA*, G-077, 1998.

[4] F. Allaire, G. Jouve and AM. Toader. A level-set method for shape optimization. *C. R. Acad. Sci. Paris*, 2002.

[5] Luis N. Vicente Andrew R. Conn, Katya Scheinberg. *Introduction to Derivative-Free Optimization.* MPS-SIAM, 2009.

[6] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13 :147–269, 2004.

[7] S. Schmidt C. Schillings and V. Schulz. Efficient shape optimization for certain and uncertain aerodynamic design. *Computers & Fluids*, 46(1) :78–87, 2011.

[8] J. Caers. *Modeling Uncertainty in the Earth Sciences.* Wiley-Blackwell, 2011.

[9] G. Casella and R. Berger. *Statistical Inference, 2nd Ed.* Duxbury Press, London, 2001.

[10] Warren Hare Charles Audet. *Derivative-Free and Blackbox Optimization.* Springer International Publishing, 2017.

[11] Vicente L. N. Conn A. R., Scheinberg K. *Introduction to Derivative-Free Optimization.* 2009.

[12] A. Dervieux and F. Thomasset. A finite element method for the simulation of rayleigh-taylor instability. *Lecture Notes in Mathematics*, 771 :145–159, 1979.

[13] A. Dervieux and F. Thomasset. Multifuid incompressible fows by a finite element method. *Lecture Notes in Physics*, 11 :158–163, 1991.

[14] GB. McFadden DM. Anderson and A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annu. Rev. Fluid Mech.*, 30 :139–165, 1998.

[15] C. Geuzaine and J.-F. Remacle. Gmsh : a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 2009.

[16] R. Ghanem and A. Doostan. On the construction and analysis of stochastic models : characterization and propagation of the errors associated with limited data,. *J. of Comput. Phys.*, 217 :63–81, 2006.

[17] R. Ghanem and P. Spanos. *Stochastic Finite Elements : A Spectral Approach.* Springer Verlag, New York, 1991.

[18] F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4) :251–265, 2012.

[19] Shavit Nir Herlihy, Maurice. The art of multiprocessor programming,.

[20] P.G. Hoel. *Introduction to Mathematical Statistics.* John Wiley, London, 1971.

[21] M. Peyret J. Chery, B. Mohammadi and C. Joulain. Plate rigidity inversion in southern california using interseismic gps velocity field. *Geophys. J. Int.*, 187/2 :783–796, 2011.

[22] R. O. Onez J. T. Spooner, M. Maggiore and K.M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems : Neural and Fuzzy Approximator Techniques.* John Wiley, New York, 2002.

[23] Mohammadi B. Saiac JH. *Pratique de la simulation numérique.* Dunod, 2003.

[24] M. Ghil K. Ide, P. Courtier and A. Lorenc. Unified notation for data assimilation : operational, sequential and variational. *Journal of the Meteorological Society of Japan*, 75/1B :181–189, 1997.

[25] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82 :35–45, 160.

[26] K.C. Kapur and L.R. Lamberson. *Reliability in Engineering Design.* John Wiley & Sons, New York, 1997.

[27] J. Kim D. Kim and H. Choi. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys.*, 171 :132–150, 2001.

[28] E Le Fichoux. Présentation et utilisation de cast3m. *ENSTA-LME*, 2011.

[29] Z. Leveque, RJ. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Num. Anal.*, 1995.

[30] H.R. Lindman. *Analysis of Variance in Complex Experimental Designs.* Freeman, New York, 1974.

[31] B. Mohammadi. Global optimization, level set dynamics, incomplete sensitivity and regularity control. *IJCFD*, 2007.

[32] B. Mohammadi. Reduced sampling and incomplete sensitivity for low-complexity robust parametric optimization. *Int. J. Num. Meth. Fluids*, 73/4 :307–323, 2013.

[33] B. Mohammadi. Principal angles between subspaces and reduced order modeling accuracy in optimization. *Structural and Multidisciplinary Optimization*, 50/2 :237–252, 2014.

[34] B. Mohammadi. Uncertainty quantification by geometric characterization of sensitivity spaces. *Compt. Meth. Appl. Mech. Eng.*, 280 :197–221, 2014.

[35] B. Mohammadi. Value at risk for confidence level quantifications in robust engineering optimization. *Optimal Control : Applications and Methods*, 35/2 :179–190, 2014.

[36] B. Mohammadi. Backward uncertainty propagation in shape optimization. *Int. J. for Numerical Methods in Fluids*, 103(4)-DOI : 10.1002/fld.4077 :307–323, 2016.

[37] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : Algorithms based on hamilton-jacobi formulations. *J. Comp. Phys.*, 1988.

[38] CH. Bruneau P. Angot and P. Frabrie. A penalization method to take into account obstacles in viscous flows. *Numerische Mathematik*, 81 :497–520, 1999.

[39] CS. Peskin. The fluid dynamics of heart valves : experimental, theoretical and computational methods. *Annu. Rev. Fluid Mech.*, 14 :235–259, 1981.

[40] TW. Pan R. Glowinski and J. Periaux. A fictitious domain method for external incompressible viscous flows modeled by navier-stokes equations. *Comput. Meth. Appl. Mech. Eng.*, 112 :133–148, 1994.

[41] M. Schnellenbach-Held and K. Pfeffer. Punching behavior of biaxial hollow slabs. *Elsevier Science*, 2002.

[42] J.A. Sethian. *Fast Marching Methods and Level Set Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences.* Cambridge University Press, Cambridge, 2004.

[43] A. Tarantola. *Inverse problem theory and methods for model parameter estimation.* SIAM, N.Y., 1987.

[44] Guido Van Rossum and Fred L Drake Jr. *Python tutorial.* Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[45] X. Wan and G.E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary probability measures. *SIAM J. Sci. Comput.*, 28/3 :901–928, 2006.

[46] D. Xiu. *Numerical Methods for Stochastic Computations : A Spectral Method Approach.* Princeton University Press, 2010.