



HAL
open science

Méthodes hiérarchiques pour la conception optimale de forme d'antenne à réflecteur

Benoît Chaigne, Jean-Antoine Désidéri

► **To cite this version:**

Benoît Chaigne, Jean-Antoine Désidéri. Méthodes hiérarchiques pour la conception optimale de forme d'antenne à réflecteur. [Rapport de recherche] RR-6625, INRIA. 2008, pp.45. inria-00316162v2

HAL Id: inria-00316162

<https://inria.hal.science/inria-00316162v2>

Submitted on 30 Jun 2009

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.

Méthodes hiérarchiques pour la conception optimale de forme d'antenne à réflecteur

Benoît Chaigne — Jean-Antoine Désidéri

N° 6625 — version 2

version initiale Septembre 2008 — version révisée Septembre 2008

Thème NUM



*Rapport
de recherche*

Méthodes hiérarchiques pour la conception optimale de forme d'antenne à réflecteur

Benoît Chaigne , Jean-Antoine Désidéri

Thème NUM — Systèmes numériques
Équipe-Projet Opale

Rapport de recherche n° 6625 — version 2 — version initiale Septembre 2008 — version révisée
Septembre 2008 — 48 pages

Résumé : On s'intéresse à un problème de conception optimale de forme d'antenne à réflecteur. On s'appuie sur un problème modèle purement géométrique de reconstruction de forme pour montrer que les algorithmes de descente se comportent comme des "anti-lisseurs". Ensuite, un algorithme multiniveau de correction "hautes fréquences" est développé afin d'accélérer la convergence de méthodes classiques. Les expériences numériques montrent que cette méthode est plus efficace qu'une méthode de Gradient Conjugué mais moins qu'une méthode de Quasi-Newton pour les tailles de problème considérées (<50).

Mots-clés : optimisation de forme, problèmes inverses, algorithmes hiérarchiques, multigrille, électromagnétisme, antenne à réflecteur

Hierarchical methods for the optimal shape design of a reflector antenna

Abstract: We consider an optimal shape design problem of a reflector antenna. We provide theoretical results on a simple purely geometrical shape optimization problem to show that descent algorithms can be seen as “anti-smoother”. Then, a “high frequency” multilevel correction algorithm is developed in order to speed up the convergence of classical methods. Numerical experiences show that this method is more efficient than a Conjugate Gradient method but not as good as a Quasi-Newton algorithm for the considered problem sizes (<50).

Key-words: shape optimization, inverse problems, hierarchical methods, multigrid, electromagnetics, reflector antenna

Table des matières

1	Introduction	4
2	Généralités sur le problème des moindres carrés linéaire	4
3	Problème de meilleure approximation	5
3.1	Fonctionnelle de forme	5
3.2	Fonctionnelle paramétrique	6
3.3	Application aux éléments P1	6
3.4	Meilleure approximation dans un espace polynômial	10
3.5	Étude du spectre des hessiens paramétriques	12
3.5.1	Polynômes de Bernstein	13
3.5.2	Polynômes de Legendre	15
3.5.3	Polynômes de Tchebychev	15
3.5.4	B-splines	17
3.5.5	Influence de la dimension sur le conditionnement de la matrice hessienne	19
3.6	Étude de la convergence d'une méthode de descente	20
3.6.1	Algorithme de Quasi-Newton	20
3.6.2	Description du cas-test	21
3.6.3	Précision de la solution à convergence de l'algorithme	21
3.6.4	Vitesse de convergence	23
3.6.5	Représentation de l'erreur absolue	24
3.7	Conclusion du problème de meilleure approximation	24
4	Méthodes hiérarchiques d'optimisation	28
4.1	Itération linéaire et algorithmes de descente	28
4.1.1	Généralités	28
4.1.2	Point de vue de l'optimisation	29
4.2	Hiérarchisation des espaces de recherche : opérateurs de prolongement et de restriction	29
4.2.1	Paramétrisations emboîtées, méthode Y	30
4.2.2	Préconditionnement par permutation du spectre, méthode Z	31
4.2.3	Sous-espaces propres, méthode Ω	31
4.3	Algorithme bigrille idéal	32
4.3.1	Multigrille classique	32
4.3.2	Multigrille en optimisation	32
4.3.3	Rayon spectrale du cycle idéal	33
4.4	Algorithme multigrille	33
4.5	Résultats numériques sur le problème modèle	33
4.5.1	Accélération de la convergence	35
4.5.2	Amélioration de la précision	37
5	Application à un problème de conformation d'antenne	38
5.1	Introduction à la conception optimale d'antenne à réflecteur	38
5.2	Étude du spectre des hessiens paramétriques à la solution	39
5.3	Résultats numériques	43
5.3.1	Uniformisation d'un diagramme en directivité	43

5.3.2	Problème inverse en puissance	45
6	Conclusion et perspectives	46

1 Introduction

Un problème d’optimisation de forme consiste à chercher la forme qui, dans une classe de formes admissibles, réalise l’optimum d’un critère défini *a priori*. Ce critère peut dépendre explicitement de la forme et éventuellement ou exclusivement d’une variable d’état qui en dépend implicitement, par exemple par le biais d’une équation aux dérivées partielles (où le domaine correspond alors classiquement à la forme à optimiser).

Si les formes admissibles existent dans un espace de dimension infinie, en pratique, l’espace de recherche est réduit à un espace de dimension finie (nécessairement inclus dans la classe des formes admissibles) en vue d’une optimisation numérique. Toute forme de cet espace est représentée dans une base par un nombre fini de variables appelées paramètres de conception (ou d’optimisation).

On distingue en général les approches *CAD-free*, non-paramétriques, qui considèrent la discrétisation du domaine à optimiser comme contrôle (éléments du maillage, éventuellement support de l’état), des approches paramétriques dont le contrôle est défini indépendamment du maillage (représentation de Bézier, B-splines, points de contrôle *Free-Form deformation*, etc. [10, 20]).

L’approche *CAD-free* conduit en général à l’optimisation d’un grand nombre de degrés de liberté (la taille du maillage étant potentiellement de l’ordre du million de nœuds) tandis que l’approche paramétrique n’optimise que quelques dizaines de variables, ce qui rend l’optimisation numérique moins coûteuse. Dans ces différentes configurations (approche paramétrique ou non, choix de la base de paramétrisation) on examine la convergence d’un algorithme de descente. En particulier, on souhaite déterminer si dans le contexte d’un problème d’optimisation de système gouverné par une EDP, on retrouve des propriétés de convergence adéquates à l’application d’une stratégie hiérarchique de type multigrille.

Pour cela nous nous intéressons dans un premier temps à la minimisation d’une simple fonctionnelle qui ne dépend que de la forme : nous traitons un problème classique de meilleure approximation d’une fonction de L_2 sur un intervalle fermé et borné $[a, b]$ dans un sous-espace F de dimension finie (section 3). Ce problème conduit à résoudre un système linéaire $Ax = b$ où A et b dépendent du choix des paramètres de conception x . On étudie le conditionnement et le spectre de A ainsi que la vitesse de convergence de l’algorithme de descente (il s’agit d’une extension de l’étude [8]).

Dans un second temps, après avoir mis en évidence les faiblesses de certaines paramétrisations du point de vue de la qualité de la convergence (vitesse et précision), on s’intéresse à un algorithme multiniveau afin de palier ces problèmes (section 4). Cette algorithme s’inspire des travaux [22].

Enfin, on s’attache à un problème de conception optimal d’antenne à réflecteur (section 5). Comme pour le problème purement géométrique de meilleure approximation, on étudie le spectre des hessiens paramétriques afin d’obtenir des informations sur la raideur du système et sur la convergence. On ajoute ici une difficulté supplémentaire : la non-linéarité du gradient (*i.e.* la fonctionnelle n’est pas quadratique). La convexité n’est d’ailleurs pas garantie, pas plus que l’unimodalité. Toutefois on s’affranchira de la multimodalité en choisissant un point de départ “assez proche” de la solution.

2 Généralités sur le problème des moindres carrés linéaire

On rappelle brièvement le problème des moindres carrés linéaire (en dimension finie) qui s’apparente au problème de meilleure approximation (dimension infinie).

Soit le système linéaire surdéterminé $Ax = b$, $A \in \mathbb{R}^{m \times n}$ et $b \in \mathbb{R}^m$ tel que $m > n$. Il est rare que b appartienne à l’image de A , une telle équation n’a donc généralement pas de solution.

Alternativement on cherche à minimiser le résidu $Ax - b$ au sens des moindres carrés, soit

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

pour la norme euclidienne. Par ailleurs on note que

$$\begin{aligned} \|Ax - b\|^2 &= (Ax - b)^T (Ax - b) \\ &= x^T A^T A x - 2x^T A^T b + b^T b. \end{aligned}$$

Le problème des moindres carrés revient donc à minimiser la fonction $J(x) = \frac{1}{2}x^T A^T A x - x^T A^T b$ dont le gradient est $G(x) = A^T A x - A^T b$. Le minimum est atteint pour x tel que $G(x) = 0 \Leftrightarrow A^T A x = A^T b$. Ce minimum existe et est unique puisque la matrice $A^T A$ est définie positive. Les équations du système $A^T A x = A^T b$ sont appelées équations normales.

D'un point de vue numérique il est connu que la matrice $A^T A$ est généralement très mal conditionnée. Le système est rarement résolu sous cette forme. Similairement, dans les sections qui suivent, on s'intéresse à un problème des moindres carrés dans le cadre d'espaces fonctionnels. On aboutit également à un système de type $Ax = b$ à résoudre pour lequel on étudiera le conditionnement de A .

3 Problème de meilleure approximation

Dans cette section on note H_0 l'espace des fonctions L_2 sur l'intervalle $[a, b]$ telles que $f(a) = f(b) = 0$. On munit H_0 du produit scalaire usuel dans L_2

$$(f, g) = \int_a^b f(t)g(t)dt \quad (1)$$

et de sa norme associée.

$$\|f\| = \sqrt{(f, f)} = \left(\int_a^b |f(t)|^2 dt \right)^{1/2}. \quad (2)$$

3.1 Fonctionnelle de forme

Soit \bar{f} une fonction de H_0 et F un sous-espace de H_0 . On cherche à résoudre le problème de minimisation suivant :

$$\min_{f \in F} \mathcal{J}(f) = \frac{1}{2} \|f - \bar{f}\|^2 = \frac{1}{2} \int_a^b |f(t) - \bar{f}(t)|^2 dt. \quad (3)$$

Bien évidemment, la fonctionnelle de forme \mathcal{J} est continue, différentiable et quadratique (donc convexe). Sa différentielle \mathcal{G} est telle que

$$\langle \mathcal{G}(f), \delta f \rangle = (f - \bar{f}, \delta f) = \int_a^b (f(t) - \bar{f}(t)) \delta f(t) dt, \quad \forall \delta f \in H_0 \quad (4)$$

et son hessien \mathcal{H}

$$\langle \mathcal{H}(f) \delta f, \delta g \rangle = (\delta f, \delta g) = \int_a^b \delta f(t) \delta g(t) dt, \quad \forall \delta f, \delta g \in H_0. \quad (5)$$

\mathcal{H} est bien indépendant de f , symétrique et défini positif (s.d.p.) car $\langle \mathcal{H}(f) \delta f, \delta f \rangle = \|\delta f\|^2 > 0$.

3.2 Fonctionnelle paramétrique

L'approche paramétrique consiste à représenter la courbe dans un espace de dimension finie. Elle s'exprime donc comme une combinaison linéaire finie de fonctions de bases. Soit $\{f_k\}_{k=0}^n$ une famille libre de H_0 . L'espace $F = \text{vect}\{\dots, f_k, \dots\}$ est un sous-espace de H_0 de dimension $n + 1$ (ou de degré n , pour un espace polynômial). Par construction, quel que soit f appartenant à F il existe $x \in \mathbb{R}^{n+1}$ tel que

$$f[x](t) = \sum_{k=0}^n x_k f_k(t). \quad (6)$$

Si on injecte cette représentation de f dans (3) on obtient

$$\forall f \in F, \quad \mathcal{J}(f) = \frac{1}{2} \int_a^b \left| \sum_{k=0}^n x_k f_k(t) - \bar{f}(t) \right|^2 dt.$$

On appelle fonctionnelle paramétrique l'application J définie sur \mathbb{R}^{n+1} suivante

$$J(x) = \mathcal{J}(f[x]) = \frac{1}{2} \int_a^b \left| \sum_{k=0}^n x_k f_k(t) - \bar{f}(t) \right|^2 dt. \quad (7)$$

On remarque bien que la forme de la fonctionnelle paramétrique (7), qui est la restriction à F de la fonctionnelle de forme (3), dépend de la base dans laquelle les fonctions f de F sont exprimées.

Le gradient G de J est

$$G(x) = \begin{pmatrix} \vdots \\ \langle \mathcal{G}(f[x]), f_k \rangle \\ \vdots \end{pmatrix}, \quad (8)$$

et la matrice hessienne H

$$H(x) = \begin{pmatrix} \cdots & \vdots \\ \langle \mathcal{H}(f[x]) f_k, f_j \rangle & \end{pmatrix}. \quad (9)$$

H est indépendant de x de la même manière que \mathcal{H} est indépendant de f . La fonctionnelle paramétrique s'écrit plus simplement

$$J(x) = \frac{1}{2} x^T H x - b^T x + c \quad (10)$$

où $b_k = (\bar{f}, f_k)$ et $c = \frac{1}{2} \|\bar{f}\|^2$. On note que H est bien s.d.p. puisqu'il s'agit d'un cas particulier de (5). La condition nécessaire d'optimalité $G(x) = 0 \Leftrightarrow Hx - b = 0$ implique que le problème (3) revient à résoudre le système $Hx = b$. Nous sommes donc amenés à étudier la matrice hessienne H dont les éléments h_{kj} sont donnés par

$$h_{kj} = \int_a^b f_k(t) f_j(t) dt \quad (11)$$

pour la base $\{f_k\}_{k=0}^n$ considérée.

3.3 Application aux éléments P1

Dans un premier temps on considère les éléments P1 comme base de représentation de f , supposée continue et linéaire par morceaux. Cette approche est considérée comme non-paramétrique

dans le sens où elle ne définit pas f de manière continue : elle tient compte de la discrétisation du support $[a b]$.

Par la suite on appelle paramétrisation toute approche qui consiste à représenter une courbe de manière continue à l'aide d'un nombre fini de paramètres (par exemple, une représentation de Bézier). La dimension de l'espace paramétrique est censée être très inférieure à la taille de la discrétisation.

Remarque 1. *on s'intéresse principalement aux approches paramétriques, cette première étude a pour but de fournir un fondement théorique aux sections qui vont suivre, essentiellement numériques.*

Soit \mathcal{T}_h une discrétisation uniforme de $[a b]$: $t_k = a + kh$, $h = \frac{b-a}{N}$, $k = 0, \dots, N$. Les fonctions P1 ou éléments P1 sur \mathcal{T}_h (fonctions "chapeaux", voir Figure 1) sont les fonctions définies par :

$$f_k(t) = \begin{cases} \frac{t-t_{k-1}}{h} & t \in [t_{k-1} t_k] & k > 0 \\ \frac{t_{k+1}-t}{h} & t \in [t_k t_{k+1}] & k < N \\ 0 & t \notin [t_{k-1} t_{k+1}] \end{cases} . \quad (12)$$

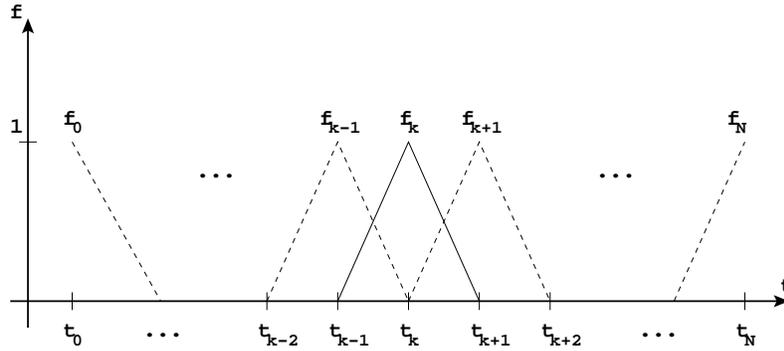


FIG. 1 – Éléments P1

Le support local des fonctions f_k implique que la matrice hessienne notée H_h a une structure bande. Ici c'est une matrice tribande dont les éléments sont :

$$h_{jk} = \begin{cases} \int_0^h \left(\frac{t}{h}\right)^2 dt & = h/3 & j = k = 1, j = k = N \\ 2 \int_0^h \left(\frac{t}{h}\right)^2 dt & = 2h/3 & 1 < j = k < N \\ \int_0^h \frac{t}{h} \left(1 - \frac{t}{h}\right) dt & = h/6 & j = k + 1, j = k - 1 \end{cases} . \quad (13)$$

On a donc $H_h = \frac{h}{6}A$ avec

$$A = \begin{pmatrix} 2 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 2 \end{pmatrix} \in \mathbb{R}^{(N+1) \times (N+1)} \quad (14)$$

ou

$$A = \begin{pmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)} \quad (15)$$

selon que l'on impose ou non des conditions au bord de type Dirichlet homogène (*i.e.* on ignore les fonctions f_0 et f_N). On se place désormais dans ce dernier cas et on étudie le spectre de $A = \Omega\Lambda\Omega^T$.

Dans un premier temps on constate que A est strictement à diagonale dominante. D'après le théorème de Gershgorin on sait que les valeurs propres de A sont telles que $\sigma(A) \subset [2 \ 6]$ et que par conséquent le conditionnement de A est borné par $\kappa_2 \leq 3$, quelle que soit la dimension du maillage N .

Soit H_Δ la matrice issue de la discrétisation par différences finies sur \mathcal{T}_h de l'opérateur laplacien $-\Delta$ appliqué aux fonctions nulles au bord :

$$H_\Delta = \frac{1}{h^2}A_\Delta = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)}. \quad (16)$$

Le spectre de $A_\Delta = \Omega_\Delta \Lambda_\Delta \Omega_\Delta^T$ est bien connu (voir [7] par exemple) :

$$\Omega_\Delta = \begin{pmatrix} S_1 & \cdots & S_{N-1} \end{pmatrix}, \quad \Lambda_\Delta = \begin{pmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_{N-1} \end{pmatrix}, \quad \mu_1 < \cdots < \mu_{N-1} \quad (17)$$

$$S_k = \begin{pmatrix} \vdots \\ \sqrt{\frac{2}{N}} \sin\left(j \frac{k\pi}{N}\right) \\ \vdots \end{pmatrix}, \quad \mu_k = 2 - 2 \cos\left(\frac{k\pi}{N}\right). \quad (18)$$

On retrouve les résultats bien connus du spectre du laplacien : les vecteurs propres sont les modes de Fourier discret sur \mathcal{T}_h (voir Figure 2). Le facteur $\sqrt{2/N}$ est un facteur de normalisation des vecteurs propres. Les valeurs propres les plus petites sont associées aux basses fréquences et les valeurs propres les plus grandes aux hautes fréquences. Le produit d'un vecteur par H_Δ (ou A_Δ) amplifie donc les hautes fréquences. Par ailleurs, quand la taille de la discrétisation augmente, puisque la plus petite valeur propre tend vers 0 et que la plus grande est bornée, le conditionnement de H_Δ tend vers l'infini : le système $H_\Delta x = b$ devient de plus en plus raide.

Il est facile de constater que

$$A = 6I - A_\Delta. \quad (19)$$

La décomposition spectrale de A est donc intimement liée à celle de A_Δ :

$$\Omega = \Omega_\Delta \quad (20)$$

$$\Lambda = 6I - \Lambda_\Delta \quad (21)$$

Les vecteurs propres sont les mêmes et l'ordre du spectre est inversé. Plus précisément, on a précédemment noté μ_k les valeurs propres de A_Δ (triées par ordre croissant) et on note λ_k les valeurs propres de A , soit

$$0 < \mu_1 < \cdots < \mu_{N-1} < 4 \quad (22)$$

et

$$\lambda_k = 6 - \mu_k \implies 6 > \lambda_1 > \cdots > \lambda_{N-1} > 2 \quad (23)$$

(voir Figure 3).

Remarque 2. *Le spectre est bien inclus dans l'intervalle prédit par le théorème de Gershgorin. Ici les inégalités sont strictes, donc $\kappa_2(A) < 3$.*

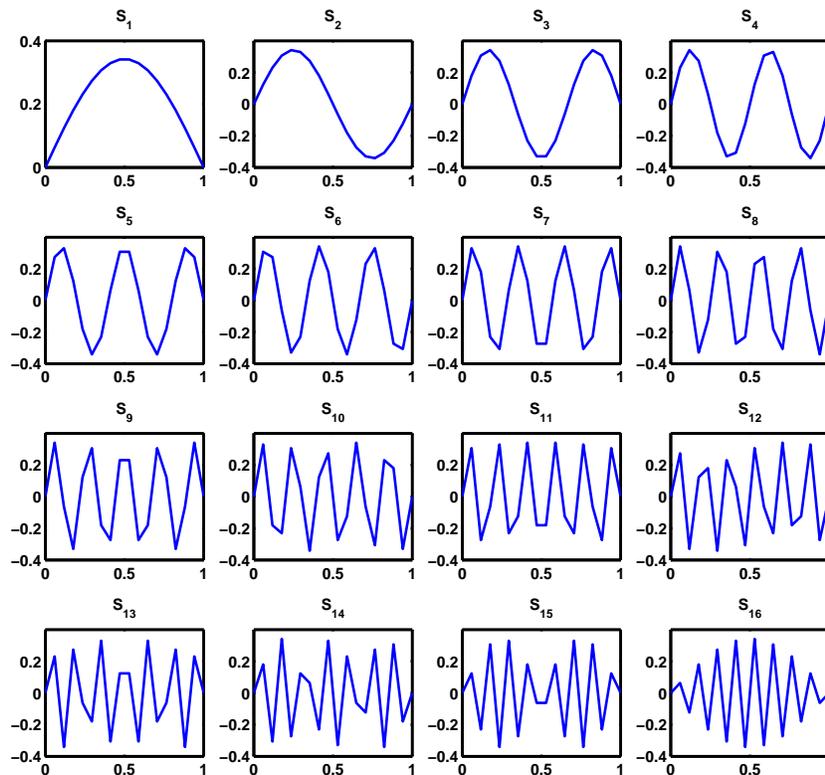


FIG. 2 – Vecteurs propres de A_{Δ} ($N = 17$) triés par ordre croissant de leurs valeurs propres (et par ordre croissant de leur fréquence).

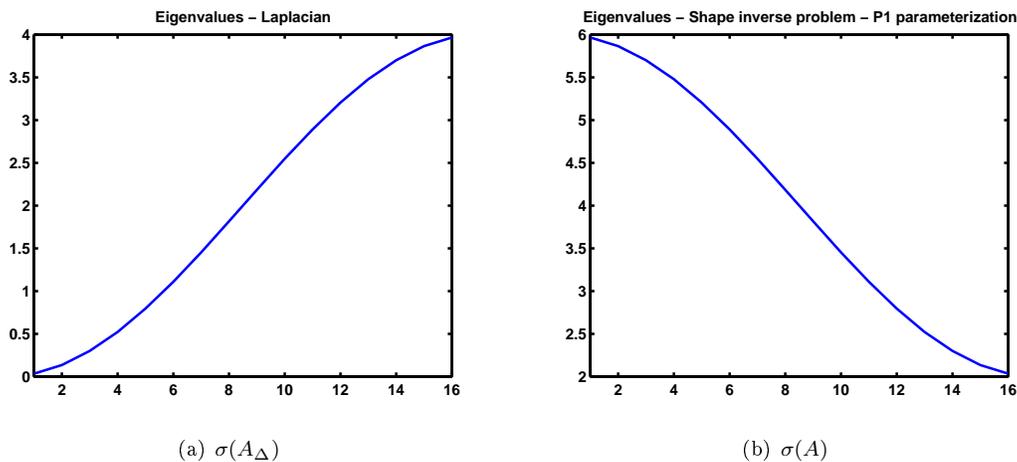


FIG. 3 – Valeurs propres de A_{Δ} et de A .

Le résultat principal de cette étude est de constater que dans le cas du problème de meilleure approximation, les valeurs propres les plus petites sont associées aux hautes fréquences et les valeurs propres les plus grandes aux basses fréquences. Le produit d'un vecteur par H (ou A) amplifie donc les basses fréquences. De plus le conditionnement est borné : ce problème est numériquement plus simple à résoudre.

L'équation de Poisson ($-\Delta f = b$) est le prototype du problème raide efficacement résolu par une méthode multigrille (MG). Le succès de cette méthode repose sur deux ingrédients : (1) les modes de hautes fréquences sont rapidement résolus (lissage); (2) les modes de basses fréquences sont transférées sur une discrétisation plus grossière et deviennent ainsi relativement les hautes fréquences (Correction de Grille Grossière). Globalement, la vitesse de convergence est théoriquement indépendante de la taille du problème, et ce malgré le mauvais conditionnement du problème originel sur la "grille" la plus fine.

Dans le cadre d'un problème d'optimisation de forme comme celui du problème de meilleure approximation, le spectre n'est pas adapté aux méthodes MG puisque les modes rapidement résolus sont les modes de basses fréquences (anti-lissage). Cependant, comme le conditionnement est très petit, on peut se demander si une telle stratégie est pertinente. Gardons toutefois en tête que le problème qui nous intéresse est l'optimisation d'un système gouverné par une EDP. Il n'est pas garanti que dans ce cas, le conditionnement soit également borné.

Par conséquent, sachant par ailleurs que nous adoptons une approche paramétrique, on souhaite : (1) vérifier que la structure spectrale du problème de meilleure approximation est conservée dans le cas paramétrique; (2) déterminer les propriétés spectrales du problème physique afin de savoir si une stratégie MG est pertinente.

3.4 Meilleure approximation dans un espace polynômial

On rappelle un résultat de meilleure approximation d'une fonction \bar{f} de H_0 dans l'espace des polynômes de degré n sur $[a, b]$ noté \mathcal{P}^n . Tout d'abord on note que \mathcal{P}^n est bien un sous-espace de H_0 puisque $[a, b]$ est borné et fermé (et dont les fonctions sont donc sommables).

Une base orthogonale de \mathcal{P}^n pour le produit scalaire (1) est donnée à partir des polynômes de Legendre P_k qui sont orthogonaux sur l'intervalle $[-1, 1]$. On a

$$\int_{-1}^1 P_k(t)P_j(t)dt = \frac{2}{2k+1}\delta_{kj} \quad (24)$$

où δ_{kj} est le symbole de Kronecker. Un simple changement de variable suffit pour obtenir les polynômes orthogonaux p_k sur $[a, b]$:

$$p_k(t) = P_k\left(2\frac{t-a}{b-a} - 1\right), \quad t \in [a, b], \quad (25)$$

et

$$\int_a^b p_k(t)p_j(t)dt = a_k\delta_{kj}. \quad (26)$$

avec $a_k = \|p_k\|^2 = \frac{b-a}{2k+1}$.

Le problème de meilleure approximation devient trivial puisque la matrice hessienne à inverser H est diagonale avec $h_{kk} = a_k$. Les coefficients du second membre $b_k = (\bar{f}, p_k)$ sont la projection orthogonale de \bar{f} sur la base p_k et l'inversion correspond à la normalisation par $a_k = \|p_k\|^2$ de chacun des coefficients de la projection.

Le spectre et le conditionnement de ce système est évident puisque qu'il est diagonal. Les vecteurs propres (en terme de paramètres de conception) sont les vecteurs de la base canonique de \mathbb{R}^{n+1} (constituant l'identité), les fonctions propres sont les fonctions p_k , et les valeurs propres sont les éléments diagonaux a_k . Le conditionnement de ce système est alors $\kappa_2(H) = \frac{a_0}{a_n} = 2n+1$.

Dans cette paramétrisation, on retrouve déjà le résultat obtenu avec les éléments P1 : les fonctions propres présentent une structure oscillante similaire aux modes de Fourier (le nombre d'oscillation du mode p_k est égale au degré k du polynôme) ; plus la fréquence est élevée, plus la valeur propre associée est petite (voir Figures 4 et 5).

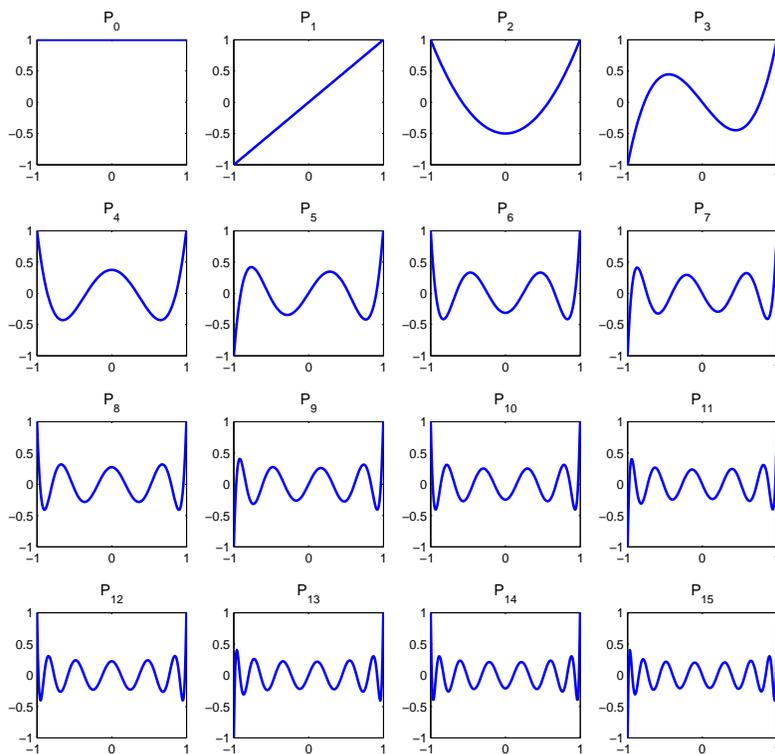


FIG. 4 – Fonctions propres dans la paramétrisation de Legendre ($n = 15$) : ce sont les polynômes de Legendre eux-mêmes.

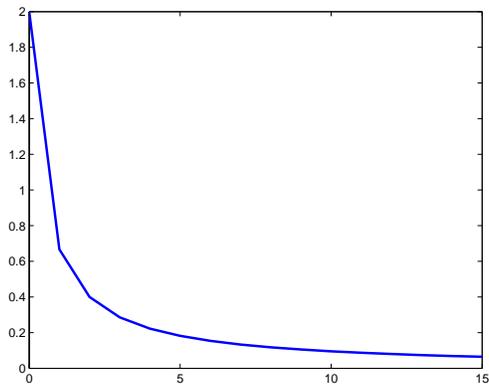


FIG. 5 – Valeurs propres dans la paramétrisation de Legendre ($n = 15$)

La paramétrisation a cependant dégradé le conditionnement qui n'est plus borné. Afin d'obtenir une matrice hessienne orthogonale (de conditionnement $\kappa_2(H) = 1$) il suffit de rendre la matrice hessienne égale à l'identité (une matrice orthogonale et diagonale est nécessairement l'identité). Pour cela il suffit donc de normaliser préalablement les p_k . La solution, $x = b$, est donc donnée par la projection orthogonale de \bar{f} sur la base orthonormée $\tilde{p}_k = \frac{p_k}{\sqrt{a_k}}$.

3.5 Étude du spectre des hessiens paramétriques

Désormais nous considérons les conditions aux limites $f(a) = f(b) = 0$. Pour une paramétrisation donnée, ces deux contraintes d'égalité s'écrivent comme des contraintes linéaires sur les paramètres x :

$$\sum_{k=0}^n x_k f_k(a) = 0, \quad \sum_{k=0}^n x_k f_k(b) = 0. \quad (27)$$

L'espace de recherche admissible a donc $(n+1) - 2 = n - 1$ dimensions. On désigne par $Z \in \mathbb{R}^{(n+1) \times (n-1)}$ une base orthogonale du noyau de l'espace des contraintes, c'est-à-dire une base orthogonale de l'espace admissible. La condition nécessaire d'optimalité s'écrit alors comme l'annulation du gradient projeté sur l'espace admissible, soit $Z^T G(x) = 0$ (voir [11]). Ce qui revient à résoudre le système $Z^T H Z x = Z^T b$ où la matrice $H_z = Z^T H Z$ est le hessien projeté. Puisque $\bar{f} \in H_0$ elle satisfait les contraintes ; le second membre $b_z = Z^T b$ décrit donc la même fonction que b mais dans la base Z ($b = Z Z^T b$). H_z est évidemment s.d.p. et admet une décomposition spectrale $H_z = \Omega \Lambda \Omega^T$. On supposera que la décomposition est ordonnée suivant les valeurs propres croissantes $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n-1}$. Les vecteurs propres $\Omega = [\omega^1 \dots \omega^{n-1}]$ correspondent chacun à une direction de l'espace admissible dans la base Z . Les vecteurs $\tilde{\omega}^k = Z \omega^k$ correspondent donc à l'expression des vecteurs propres de H_z dans l'espace de paramétrisation d'origine. Ils définissent les fonctions propres suivantes

$$\tilde{u}_k(t) = \sum_{j=0}^n \tilde{\omega}_j^k f_j(t), \quad k = 1 \dots n-1. \quad (28)$$

Par ailleurs il est clair que les vecteurs propres ω^k sont arbitrairement normalisés puisque tout vecteur colinéaire à un vecteur propre est également un vecteur propre associé à la même valeur propre. Ainsi on parle plutôt de direction propre. Dans notre cas on s'intéresse aux fonctions \tilde{u}_k . De la même manière, toute fonction propre est définie à un facteur près. On choisit donc de représenter sans perte de généralité les fonctions

$$u_k = \frac{\tilde{u}_k}{\|\tilde{u}_k\|_\infty}$$

colinéaires aux fonctions \tilde{u}_k et telles que $\|u_k\|_\infty = 1$.

On peut montrer qu'un algorithme de gradient à pas constant τ pour la minimisation de (7) est équivalent à une méthode de Jacobi pour la résolution de $H_z x = b_z$ avec coefficient de relaxation τ (voir section 4.1). Dans le cadre de ces algorithmes itératifs il est intéressant d'observer le spectre de H_z puisque la quantité $\tau(1 - \lambda_k)$ devient équivalent au coefficient d'amortissement de l'erreur relativement au mode propre u_k (τ est le pas de descente qui correspond à un coefficient de sur/sous-relaxation). Ainsi, les modes correspondants aux plus petites valeurs propres convergent plus lentement. Nous particularisons l'étude aux paramétrisations suivantes :

- polynômes de Bernstein (représentation de Bézier)
- polynômes de Legendre (polynômes orthogonaux pour le produit scalaire (1))
- polynômes de Tchebychev
- B-splines d'ordre 4 (fonctions à support local)

Remarque 3. *Les trois premières paramétrisations sont toutes des bases de l'espace des polynômes de degré n alors que la dernière est une base de fonctions splines, polynômes par morceaux.*

Dans les sections 3.5.1 à 3.5.4 on représente les modes u_k ainsi que le spectre $\sigma(H_z) = \{\lambda_k\}$ pour chacune des paramétrisations. On considère $a = 0$, $b = 1$ et $n = 9$ (espace admissible de dimension 8). À la section 3.5.5 on compare le nombre de conditionnement des matrices hessiennes issues de chacune des paramétrisations en fonction de nombre de degrés de liberté (d.d.l.).

3.5.1 Polynômes de Bernstein

Les polynômes de Bernstein sont définis sur $[0, 1]$ par

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \quad (29)$$

où $C_n^k = \frac{n!}{k!(n-k)!}$ sont les coefficients binomiaux (voir Figure 6). Sur $[a, b]$ on définit les fonctions de base

$$b_n^k(t) = B_n^k\left(\frac{t-a}{b-a}\right). \quad (30)$$

Les éléments de la matrice hessienne sont donnés par [10]

$$h_{kj} = \int_a^b b_n^k(t) b_n^j(t) dt = \frac{C_n^k C_n^j}{C_{2n}^{k+j}} \frac{b-a}{2n+1}. \quad (31)$$

Seule b_n^0 est non nulle en $t = a$ et b_n^n non nulle en $t = b$. Les contraintes (27) s'écrivent plus simplement

$$x_0 = 0, \quad x_n = 0.$$

On en déduit le hessien projeté H_z dont on peut désormais calculer le spectre numériquement. On représente sur les Figures 7 et 8 les modes propres et les valeurs propres.

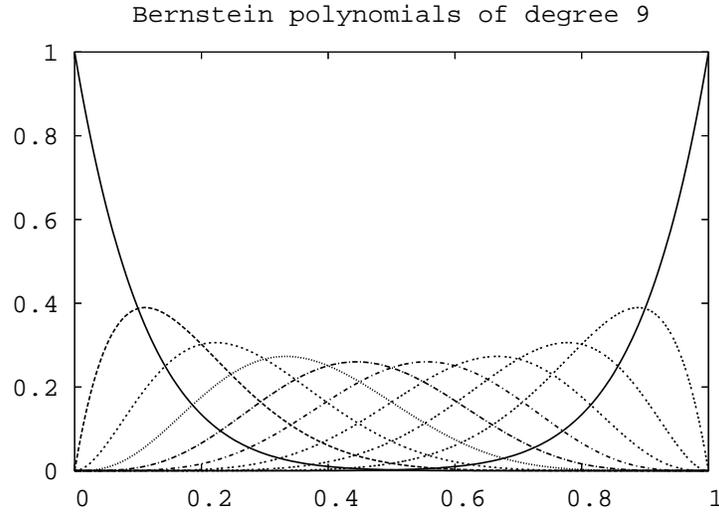


FIG. 6 – Polynômes de Bernstein de degré 9

On constate sans surprise que la structure spectrale de la matrice hessienne mise en évidence avec les éléments P1 et les polynômes de Legendre est conservée :

1. les modes sont similaires à des modes de Fourier,
2. plus la fréquence du mode est élevée, plus la valeur propre associée est petite.

La matrice est cependant très mal conditionnée : $\kappa_2 = \frac{\lambda_{n-1}}{\lambda_1} = 54141$ (soit plus de 10^4 fois celui issu de l'approche P1, non-paramétrique). L'approche paramétrique est donc beaucoup plus raide que l'approche P1, bien que le nombre de paramètres de contrôle soit plus petit.

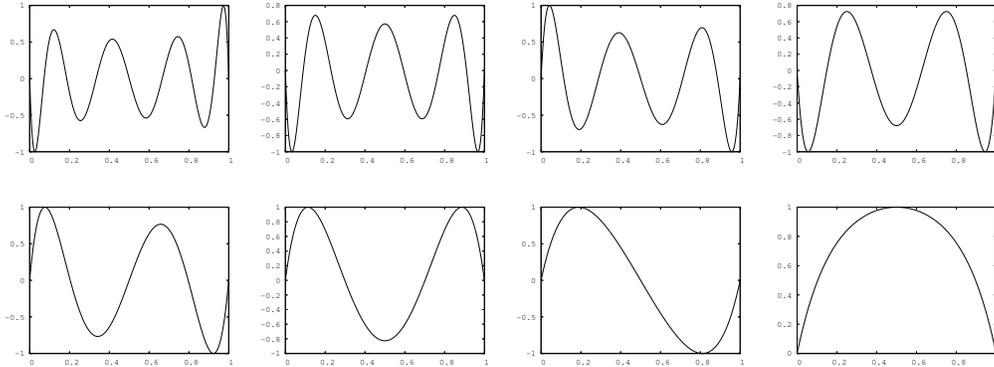


FIG. 7 – Modes propres du hessien projeté ordonnés par ordre croissant de leurs valeurs propres.

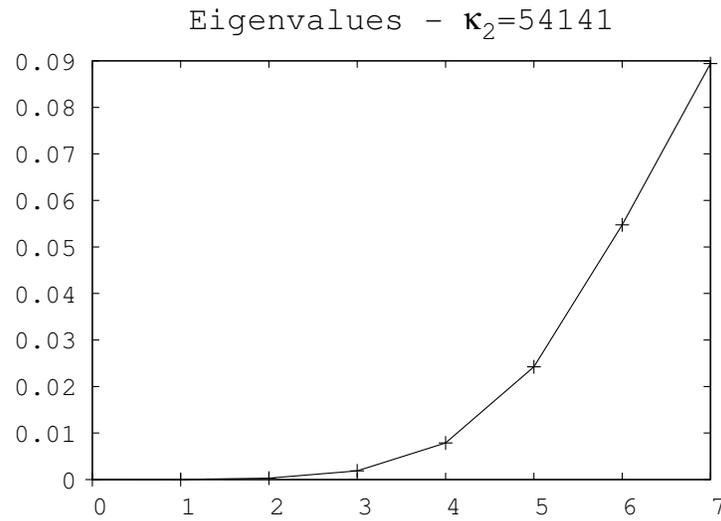


FIG. 8 – Spectre du hessien projeté.

3.5.2 Polynômes de Legendre

On a déjà défini les polynômes de Legendre à la section 3.4 (voir Figure 4) ainsi que les éléments de la matrice hessienne. Comme $p_k(a) = (-1)^k$ et $p_k(b) = 1$, les contraintes d'égalité s'écrivent

$$\sum_{k=0}^n (-1)^k x_k = 0 \quad \sum_{k=0}^n x_k = 0.$$

On calcule alors numériquement le spectre du hessien projeté (voir Figures 9 et 10).

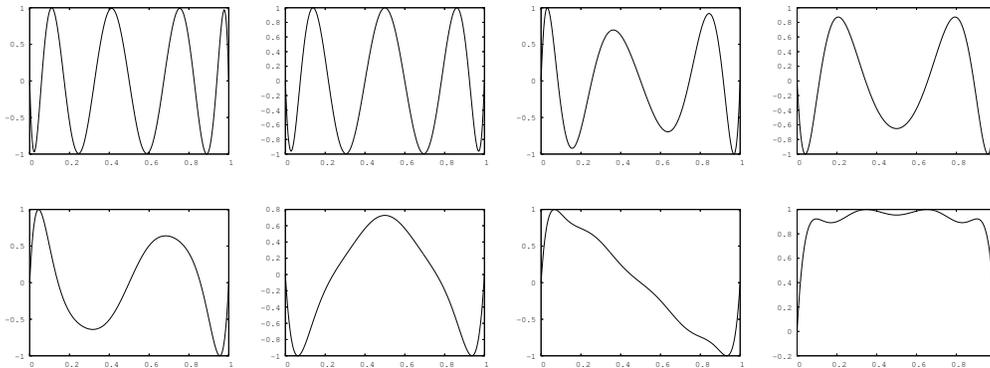


FIG. 9 – Modes propres du hessien projeté ordonnés par ordre croissant de leurs valeurs propres.

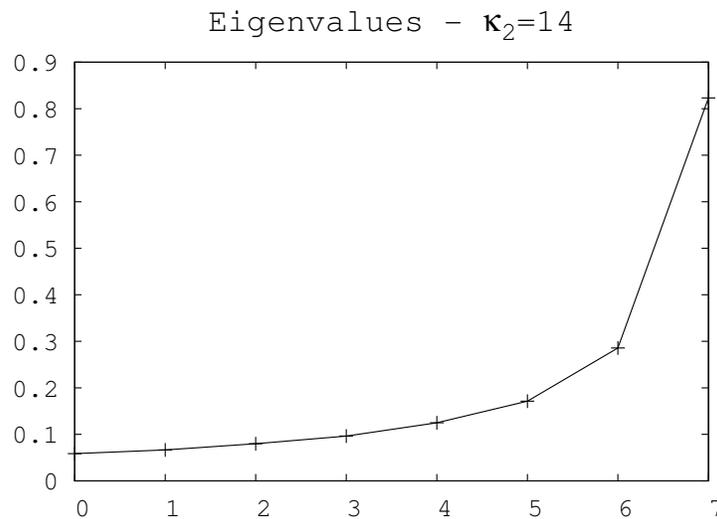


FIG. 10 – Spectre du hessien projeté.

On observe également que les modes de haute fréquence sont associés aux valeurs propres les plus petites. Le système est beaucoup mieux conditionné que celui dans la paramétrisation de Bézier-Bernstein : $\kappa_2 = 14.06$.

3.5.3 Polynômes de Tchebychev

Les polynômes de Tchebychev sont définis par

$$T_k(t) = \cos(k \arccos t), \quad t \in [-1 \ 1]. \tag{32}$$

Sur $[a, b]$ on définit les fonctions de base

$$t_k(t) = T_k \left(2 \frac{t-a}{b-a} - 1 \right), \quad t \in [a, b] \quad (33)$$

(voir Figure 11). Comme pour les polynômes de Legendre, les contraintes s'écrivent

$$\sum_{k=0}^n (-1)^k x_k = 0, \quad \sum_{k=0}^n x_k = 0.$$

Les éléments du hessien sont

$$h_{kj} = \int_a^b t_k(t) t_j(t) dt = \begin{cases} \frac{b-a}{2} \left(\frac{1}{1-(j+k)^2} + \frac{1}{1-(j-k)^2} \right) & j+k \text{ pair} \\ 0 & j+k \text{ impair} \end{cases} \quad (34)$$

On calcule alors numériquement le spectre du hessien projeté (voir Figures 12 et 13).

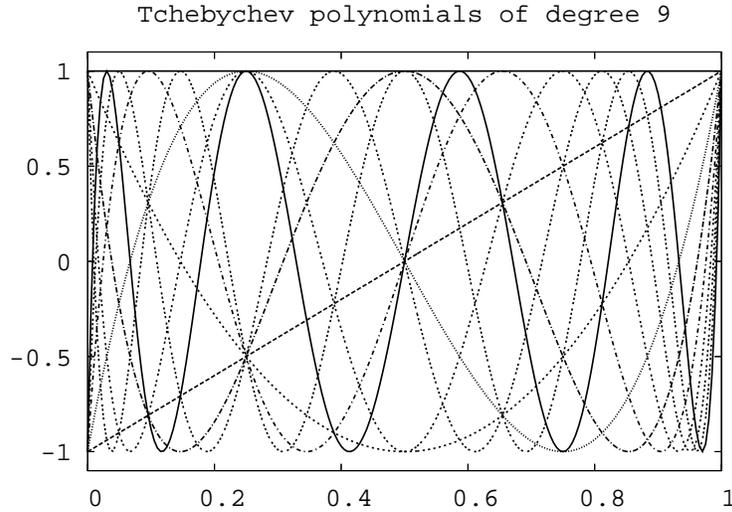


FIG. 11 – Polynômes de Tchebychev de degré 9

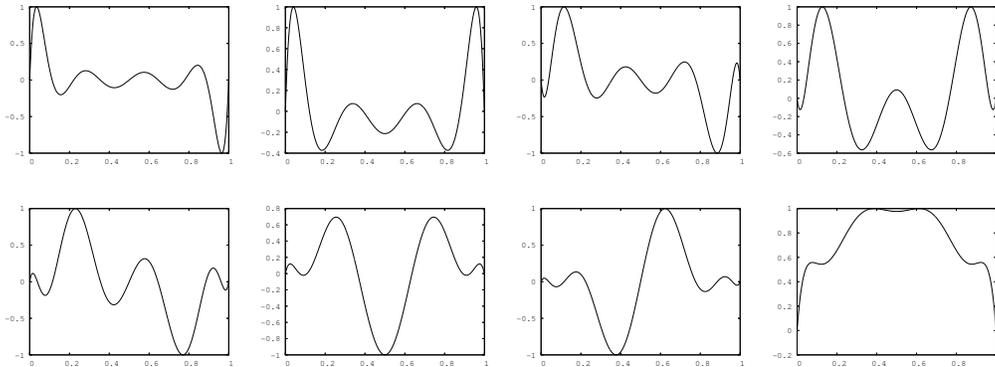


FIG. 12 – Modes propres du hessien projeté ordonnés par ordre croissant de leurs valeurs propres.

Tous les modes sont de haute fréquence hormis le mode correspondant à la plus grande valeur propre. Le nombre d'oscillations semble dépendre de la parité du mode (7 alternances de signe

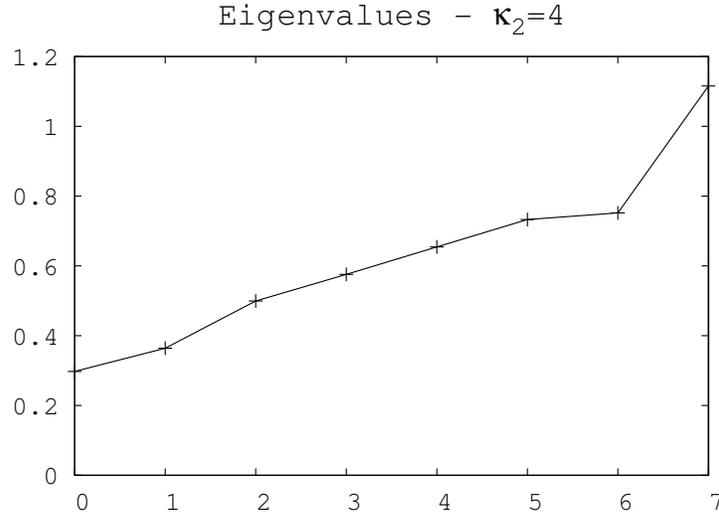


FIG. 13 – Spectre du hessien projeté.

pour les modes impairs et 6 pour les modes pairs). Cependant plus la valeur propre augmente, plus les oscillations du mode correspondant s'atténuent près des bords. Le système est très bien conditionné : $\kappa_2 = 3.75$.

3.5.4 B-splines

Les fonctions B-splines constituent une base des fonctions polynômes par morceaux (fonctions splines). L'espace de recherche diffère ainsi des autres paramétrisations (qui définissent un espace polynômial sur tout l'intervalle $[a b]$).

Les fonctions B-splines sont des fonctions à support borné. Elles sont entièrement définies par récurrence à partir des données suivantes :

- un degré d (ou ordre $d + 1$);
- une suite de $m + 1$ réels non décroissants $T = \{t_k\}_{k=1}^{m+1}$ appelés nœuds (T est le vecteur des nœuds).

On note N_k^d la k -ème fonction B-spline de degré d . Les fonctions B-splines de degré 0, constantes par morceaux, sont

$$N_k^0(t) = \chi_{[t_k t_{k+1}[} = \begin{cases} 1 & \text{si } t \in [t_k t_{k+1}[\\ 0 & \text{sinon} \end{cases} . \quad (35)$$

On a donc m fonctions N_k^0 telles que

$$\text{Supp}(N_k^0) = [t_k t_{k+1}[.$$

Puis, pour $d > 0$ et $1 \leq k \leq m - (d + 1)$, N_k^d est définie par la formule de récurrence

$$N_k^d(t) = \omega_k^d(t) N_k^{d-1} + (1 - \omega_{k+1}^d(t)) N_{k+1}^{d-1}(t) \quad (36)$$

$$\omega_k^d(t) = \begin{cases} \frac{t-t_k}{t_{k+d}-t_k} & \text{si } t_k \neq t_{k+d} \\ 0 & \text{sinon} \end{cases} . \quad (37)$$

On a donc $m - (d + 1)$ fonctions telles que

$$\text{Supp}(N_k^d) = \text{Supp}(N_k^{d-1}) \cup \text{Supp}(N_{k+1}^{d-1}) = [t_k t_{k+d+1}[.$$

Ainsi n fonctions B-spline de degré d sont nécessairement définies par un vecteur de $m = n + d + 1$ nœuds.

Remarque 4. Si le vecteur des nœuds est celui de la discrétisation \mathcal{T}_h et que le degré des splines est 1 (linéaire par morceau) alors cette paramétrisation est équivalente aux éléments P1.

La régularité de la fonction au niveau d'un nœud est déterminée par la multiplicité de ce nœud : si t_k est répété m_k fois dans le vecteur de nœuds alors N_k^d est de classe \mathcal{C}^{d-m_k} en $t = t_k$. Dans notre cas nous souhaitons une discontinuité aux bornes de l'intervalle $[a, b]$. Pour cela on construit un vecteur de nœuds tel que les valeurs extrêmes sont répétées $d + 1$ fois :

$$a = t_1 = \dots = t_{d+1} < t_{d+2} < \dots < t_{n+1} = \dots = t_{n+d+1} = b. \quad (38)$$

Nous choisissons arbitrairement de manipuler des splines cubiques ($d = 3$) de support uniforme ($t_{k+1} - t_k = \frac{b-a}{n-2d}$, $d+1 \leq k \leq n$). On représente de telles fonctions à la Figure 14 pour $n = 10$ (donc $m = n + d + 1 = 14$). Dans ce cas les contraintes sur les paramètres de contrôle s'écrivent

$$x_0 = 0, \quad x_n = 0.$$

On calcule numériquement le hessien projeté et son spectre (voir Figures 15 et 16).

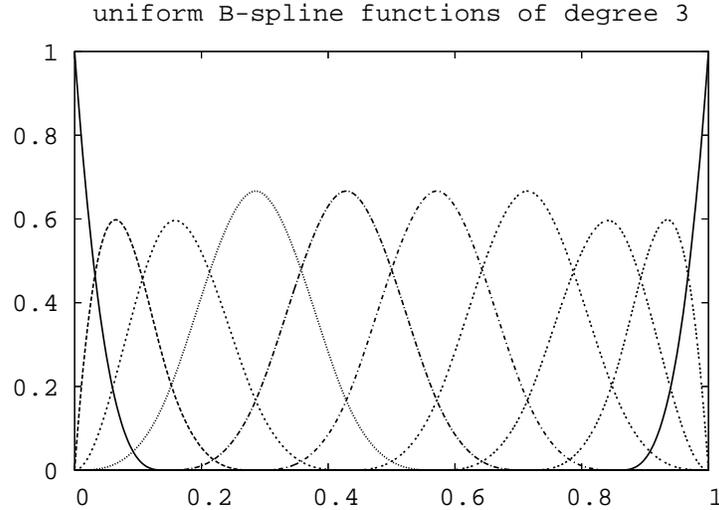


FIG. 14 – B-splines de degré 3 (d'ordre 4)

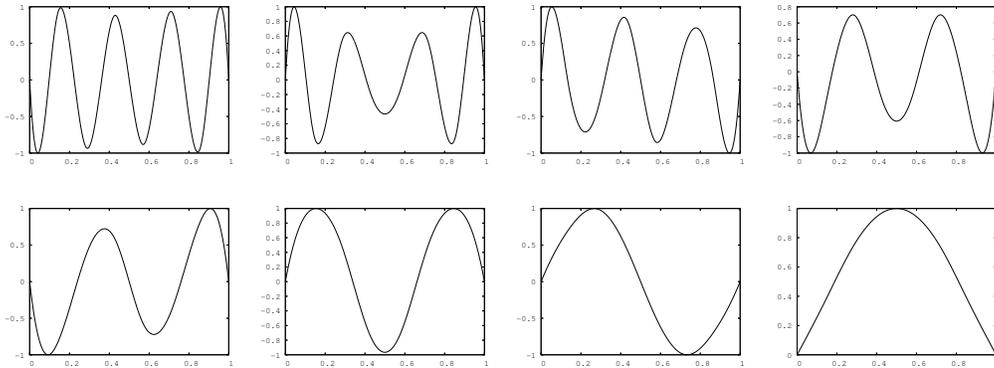


FIG. 15 – Modes propres du hessien projeté ordonnés par ordre croissant de leurs valeurs propres.

Les modes s'apparentent fortement à ceux des polynômes de Bernstein. Le hessien est une matrice $(2d + 1)$ -bande pour des splines de degré d . Le système est bien conditionné : $\kappa_2 = 21$.

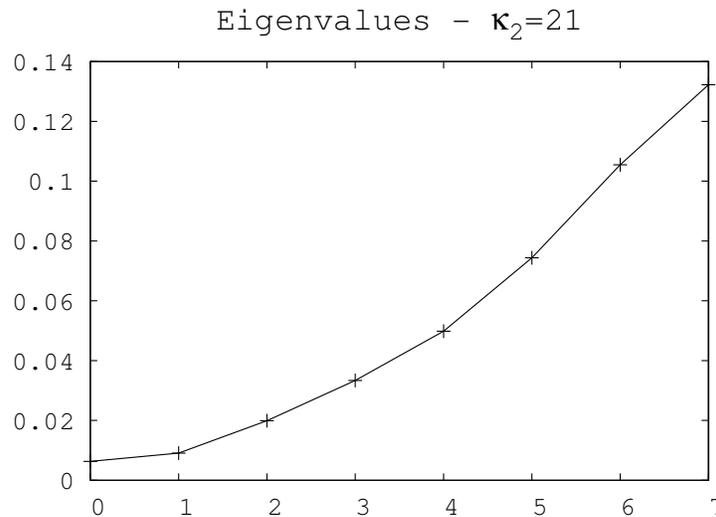


FIG. 16 – Spectre du hessien projeté.

3.5.5 Influence de la dimension sur le conditionnement de la matrice hessienne

La norme matricielle induite par la norme vectorielle $\|\cdot\|_p$ est définie par

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}. \quad (39)$$

Le nombre de conditionnement κ_p d'une matrice inversible A est

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p. \quad (40)$$

En particulier, pour $p = 2$ on a

$$\kappa_2(A) = \frac{\lambda_n}{\lambda_1}. \quad (41)$$

L'inverse du nombre de conditionnement donne une distance à la matrice non inversible la plus proche [6]. En ce sens, plus le nombre de conditionnement est élevé, plus la matrice s'apparente à une matrice non inversible. En arithmétique flottante, si cette distance est de l'ordre de la précision machine, la matrice est numériquement non inversible.

On représente à la Figure 17 une estimation numérique du nombre de conditionnement des matrices hessiennes de chacune des paramétrisations en fonction du nombre de degrés de liberté. On rapporte les observations suivantes

1. l'approche paramétrique réduit le nombre de degrés de liberté mais dégrade le conditionnement : quelque soit la paramétrisation adoptée, le problème est plus raide qu'avec l'approche non-paramétrique (le nombre de conditionnement est supérieur à 3) ;
2. en particulier la paramétrisation de Bernstein est très mal conditionnée : le nombre de conditionnement croît de manière exponentielle par rapport à la dimension du problème. Au-delà de $n = 25$ d.d.l. l'inverse du nombre de conditionnement est de l'ordre de la précision machine $\varepsilon_m \approx 2.22 \cdot 10^{-16}$: le problème n'est pas numériquement inversible ;
3. les paramétrisations de Legendre et Tchebychev sont mieux conditionnées : l'écart avec le nombre de conditionnement idéal 1 dépend linéairement de la dimension ;
4. enfin, la matrice hessienne issue de la paramétrisation B-spline d'ordre 4, (matrice bande de largeur 7) a un conditionnement constant lorsque le nombre de d.d.l. est supérieur à 7.

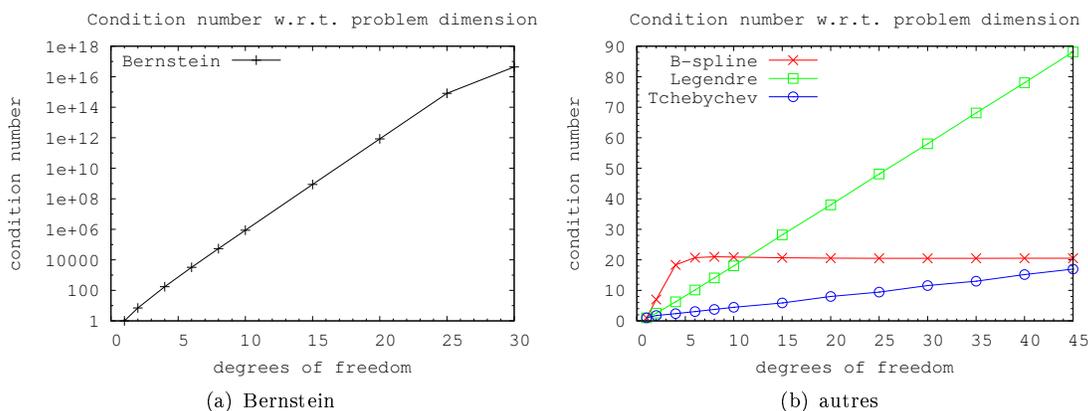


FIG. 17 – Évolution du nombre de conditionnement en fonction de la taille du problème.

Dans cette paramétrisation où le support des fonctions est local, on retrouve une propriété de l’approche non-paramétrique : le conditionnement est borné.

Enfin, dans les sections 3.5.1 à 3.5.4 on a vérifié que la structure spectrale mise en évidence avec les éléments P1 est conservée dans le cas paramétrique. Seule la paramétrisation de Tchebychev ne présente pas de modes propres similaires aux modes de Fourier. On s’attend donc à ce que ce soit les modes de hautes fréquences qui convergent lentement lorsque le problème devient raide. On illustre cette propriété par une expérience numérique à la section suivante.

3.6 Étude de la convergence d’une méthode de descente

3.6.1 Algorithme de Quasi-Newton

Soit x^0 un vecteur initial de paramètres. L’itération d’un algorithme de descente s’écrit

$$x^{i+1} = x^i + \tau^i d^i \quad (42)$$

où d^i est une direction de descente (*i.e.* telle que $G(x^i)^T d^i < 0$). Le paramètre τ^i peut être fixe ou déterminé par une recherche le long de d^i (recherche unidimensionnelle supposée simple).

Dans un algorithme de Quasi-Newton, la direction de descente est la solution du système linéaire

$$B^i d^i = -G(x^i) \quad (43)$$

où B^i est une approximation du hessien, définie positive (et nécessairement symétrique). On choisit en général $B^0 = I$, de telle sorte que la première itération est équivalente à une itération de plus grande descente. La formule de mise à jour de B^i la plus utilisée est la formule BFGS :

$$B^{i+1} = B^i + \frac{y^i y^{iT}}{y^{iT} s^i} - \frac{B^i s^i s^{iT} B^i}{s^{iT} B^i s^i} \quad (44)$$

où $s^i = x^{i+1} - x^i$ et $y^i = G(x^{i+1}) - G(x^i)$. B^{i+1} est bien symétrique et la positivité est assurée par les conditions de Wolfe.

Remarque 5 (Remarque sur le critère d’arrêt :). *Puisqu’un minimum (local) annule le gradient, un point stationnaire (de gradient nul) est donc un candidat comme minimum local. Un algorithme de gradient est donc un algorithme de recherche de point stationnaire (point fixe de l’itération (42)).*

Un critère de convergence s’exprime donc naturellement en fonction de la norme euclidienne du gradient : on s’arrête dès que cette norme est inférieure à un “petit” ε , dépendant de la précision du calcul du gradient.

Cependant, dans le cadre d'optimisation de forme paramétrique, la précision du calcul du gradient, lorsqu'elle est connue, peut varier selon la paramétrisation. Un autre critère s'exprime en fonction du résidu de la fonction coût, c'est-à-dire, on itère "tant que la fonctionnelle peut descendre"; dans ce cas on vérifie la norme du gradient a posteriori.

Si la fonction coût ne tend pas vers 0 on mesure de préférence le résidu relatif. Sinon le résidu absolu est plus adéquat. La formule (45) est un compromis entre les deux notions de résidu relatif et absolu en fonction de la valeur de la fonction coût.

$$r^i = \frac{J^i - J^{i-1}}{1 + J^{i-1}} \quad (45)$$

3.6.2 Description du cas-test

On se propose de résoudre le problème de meilleure approximation (3) à l'aide d'un algorithme de Quasi-Newton BFGS. Les espaces paramétriques considérés sont ceux de la section 3.5. La cible \bar{f} est définie pour $t \in [a, b]$, $a = -3$, $b = 7$, par l'équation

$$\bar{f}(t) = \sum_{n=0}^8 a_n \cos(ns) + b_n \sin(ns), \quad s(t) = 2\pi(t - a)/(b - a) \quad (46)$$

où les a_n sont tels que $\bar{f}(a) = \bar{f}(b) = 0$ (voir Table 1). Le vecteur initial est $x^0 = 0$ quel que soit la paramétrisation de sorte que $f[x^0] \equiv 0$, qui est évidemment admissible (voir Figure 18). On considère que l'algorithme a convergé lorsque le résidu r^i (45) est inférieure à $\varepsilon = 10^{-15}$. Les intégrales sont évaluées par la méthode de Simpson.

n	a_n	b_n
0	0.3593853805E+01	-0.1735280268E+00
1	0.1525134947E+01	-0.1678281086E+01
2	-0.1923909257E+01	0.9350947011E+00
3	0.4329616213E+01	0.1534159761E-01
4	-0.2853992139E+01	-0.6314124167E+00
5	-0.1873580031E+01	-0.2306875191E+01
6	-0.1383638992E+01	0.1325744865E+01
7	-0.2077733362E+01	-0.9480459848E+00
8	0.6642488157E+00	0.4184707832E+01

TAB. 1 – Coefficients de la fonction cible sinusoïdale

3.6.3 Précision de la solution à convergence de l'algorithme

L'expérience décrite à la section précédente est réalisée pour des paramétrisations de plus en plus fine, c'est-à-dire de degré de plus en plus élevé : $n = 5, 10, 15, 20, 25, 30, 35, 40$, et 45. La précision atteinte (valeur de la fonctionnelle à convergence de l'algorithme) est observée. Dans le cas des paramétrisations des espaces polynômiaux on s'attend à converger vers la même valeur puisque l'espace de recherche est le même et que l'optimum est unique.

On rapporte les observations suivantes :

1. Quel que soit le degré, la fonctionnelle converge vers la même valeur pour les paramétrisations bien conditionnées (Legendre, Legendre normalisé, Tchebychev). L'algorithme a convergé complètement.
2. Jusqu'à $n \leq 20$ (19 d.d.l.), dans la paramétrisation de Bernstein, la fonctionnelle converge complètement ; au-delà la précision se trouve dégradée. On rappelle que pour $n \geq 25$ le problème n'est pas numériquement inversible (voir section 3.5.5).

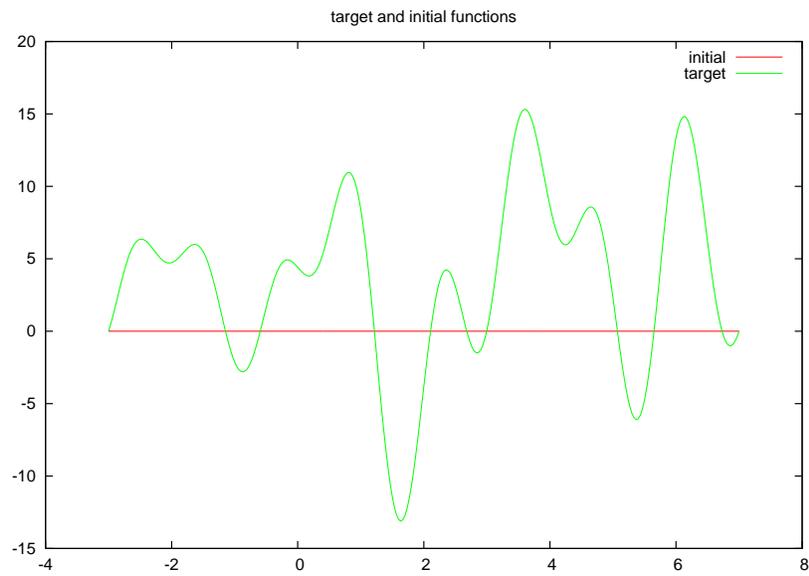


FIG. 18 – Représentation des fonctions initiale $f \equiv 0$ et cible \bar{f} .

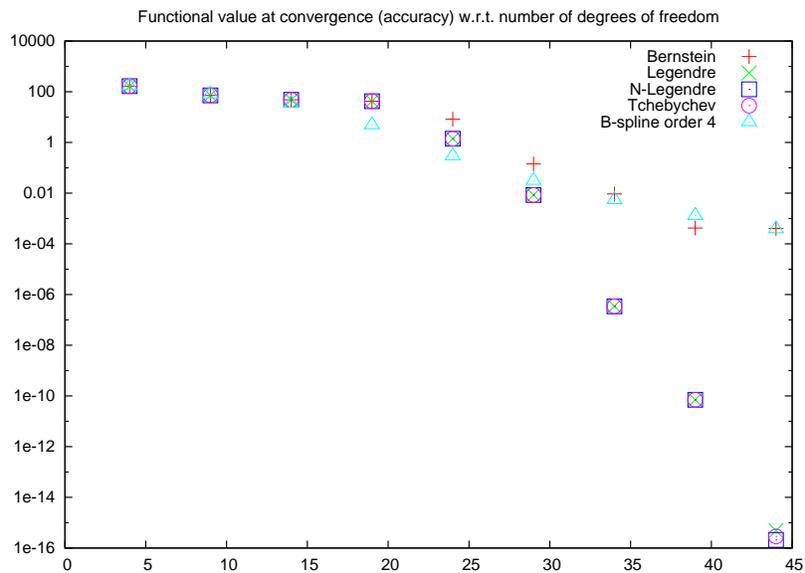


FIG. 19 – Valeur de la fonctionnelle à convergence de l'algorithme en fonction du nombre de d.d.l.

- Les espaces de recherche définis par les fonctions B-splines conduisent à une meilleure approximation de \bar{f} pour $15 \leq n \leq 25$. Pour $n \geq 30$ la solution est moins précise. Comme le conditionnement de cette paramétrisation est bon, on peut considérer que la convergence est complète.

3.6.4 Vitesse de convergence

À la section précédente on a observé la précision atteinte à convergence de l'algorithme en fonction de la dimension du problème. Dans cette section on s'intéresse à la vitesse de convergence : on mesure le nombre nécessaire d'évaluations de la fonctionnelle pour atteindre le critère de convergence ($\varepsilon = 10^{-15}$ sur le résidu). En choisissant le nombre d'évaluations plutôt que le nombre d'itérations de la méthode de descente, on prend en compte les évaluations qui ont lieu pendant la phase de recherche optimale unidimensionnelle du pas τ^i à chaque itération. Les résultats sont résumés à la Figure 20.

Par ailleurs on représente sur les Figures 21 et 22 l'historique de la convergence de l'algorithme de descente pour les problèmes de dimension $n = 20$ et $n = 45$ respectivement. On donne ainsi deux exemples où l'on représente à la fois la vitesse de convergence et la précision atteinte. La valeur optimale de la fonctionnelle tend vers 0 lorsque la dimension du problème augmente. Pour $n = 45$ la dimension est suffisamment grande pour que l'on considère que la valeur optimale soit nulle et justifie ainsi la représentation logarithmique de la fonction coût en ordonnée.

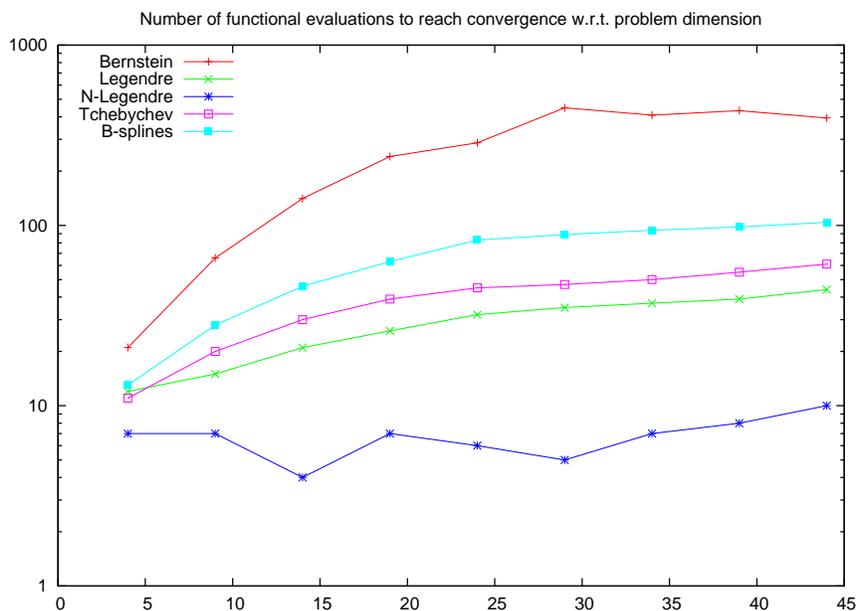


FIG. 20 – Nombre d'évaluations de la fonctionnelle pour atteindre la convergence en fonction du nombre de degrés de liberté.

Les résultats obtenus sont cohérents avec l'étude du conditionnement des hessiens conduite à la section 3.5 : plus le nombre de conditionnement κ_2 est petit, plus la convergence est rapide. La convergence dans la paramétrisation de Bernstein est la plus lente ; celle dans les paramétrisations de Legendre, Tchebychev et B-spline est plus rapide et du même ordre de grandeur ; enfin celle dans la paramétrisation de Legendre normalisée la convergence est rapide et quasiment constante par rapport à la dimension du problème (moins d'une dizaine d'itérations).

Remarque 6. Dans le cas des polynômes de Legendre normalisés, la solution s'obtient en une itération quel que soit le point initial si $\tau^0 = 1$; en effet le gradient est alors $G(x^0) = x^0 - b$ et

$x^1 = x^0 - (x^0 - b) = b$, où b est la solution (voir section 3.4). Ici τ^0 n'est pas égal à un puisque l'algorithme de descente considéré suppose que la fonctionnelle est quelconque (pas nécessairement quadratique) : τ^0 est adapté en fonction de la norme du gradient au point initial (cette méthode conduit en général à de meilleurs résultats dans le cas général) ; on converge donc en deux itérations (plus quelques unes, probablement dues aux erreurs d'arrondi).

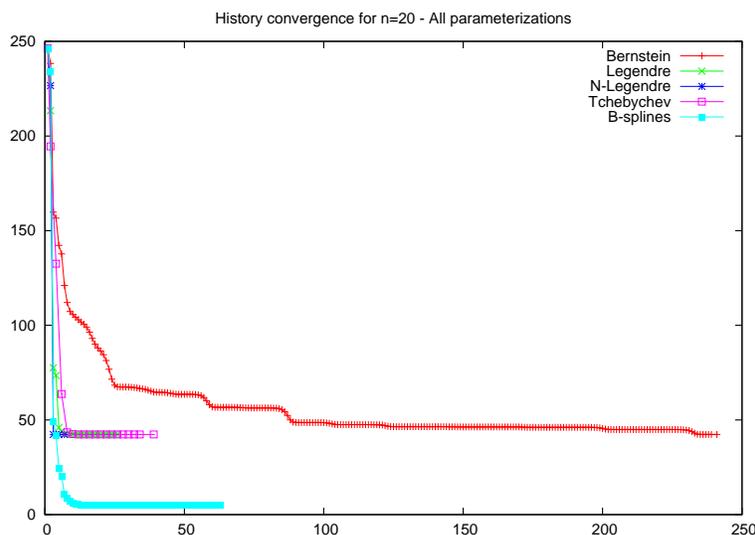


FIG. 21 – Évolution de la fonctionnelle en fonction du nombre d'évaluations pour toutes les paramétrisations, $n = 20$. La convergence dans la base de Bernstein est de 4 à 35 fois plus lente que pour les autres. Il s'agit de la base la moins bien conditionnée ($\kappa_2 \approx 10^{12}$). La précision atteinte est cependant identique à celle des paramétrisations de \mathcal{P}^{20} .

3.6.5 Représentation de l'erreur absolue

On représente sur les Figures 23, 24, et 25 l'erreur absolue $\Delta f = f[x^*] - \bar{f}$ à convergence de l'algorithme pour $n = 45$ afin de montrer que l'erreur restante à convergence de l'algorithme correspond aux modes de hautes fréquences, ceux qui convergent le moins bien ou qui n'appartiennent pas à l'espace de recherche.

On donne également l'erreur uniforme relative en pourcentage :

$$err(f) = 100 \frac{\|\Delta f\|_\infty}{\|\bar{f}\|_\infty} \quad (47)$$

3.7 Conclusion du problème de meilleure approximation

L'étude préliminaire et triviale du problème de meilleure approximation (3) a pour but d'illustrer les différents problèmes numériques qui peuvent intervenir en optimisation de forme. Dans ce cas précis, en choisissant une fonctionnelle quadratique, nous nous sommes affranchis des problèmes de multimodalité et de non convexité. De cette manière nous avons pu conduire une étude de la convergence d'un algorithme de descente classique sans se soucier de la robustesse de la méthode par rapport à la forme initiale. En particulier nous avons étudié l'influence de différentes paramétrisations sur la convergence en terme de précision et de vitesse.

Dans ce cadre on rapporte les résultats suivants :

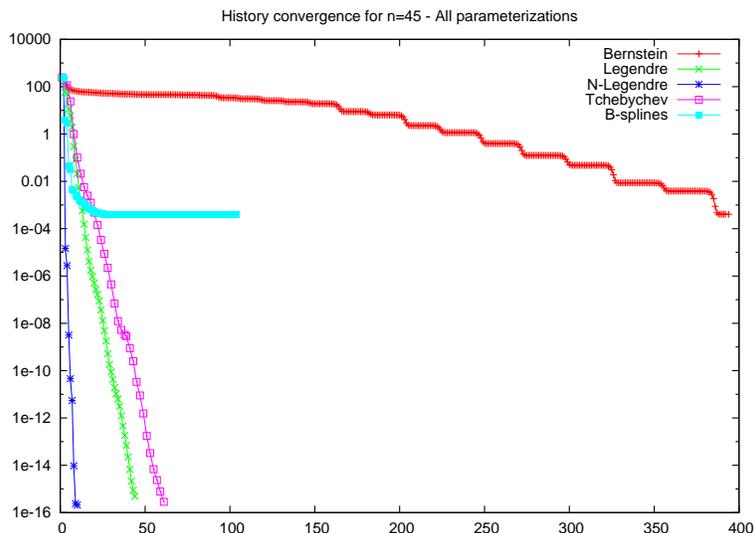


FIG. 22 – Évolution de la fonctionnelle en fonction du nombre d'évaluations pour toutes les paramétrisations, $n = 45$. La convergence dans la base de Bernstein est également bien plus lente (de 4 à 40 fois); le conditionnement est tel que $\kappa_2 > 10^{18}$. Ici la précision atteinte est très mauvaise (un rapport de 10^{12} entre les valeurs de fonctionnelle!). La précision atteinte dans la paramétrisation des B-splines est aussi médiocre (équivalente à celle de la paramétrisation de Bernstein).

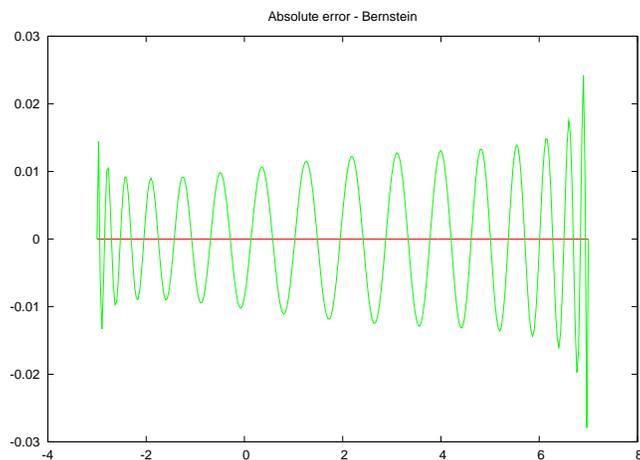


FIG. 23 – Représentation de l'erreur absolue $f[x^*] - \bar{f}$ pour la paramétrisation de Bernstein. $err(f) \approx 0.16\%$. Le nombre d'alternances de signe est de 31. Les modes de hautes fréquences n'ont pas convergé ou ne peuvent pas être résolus.

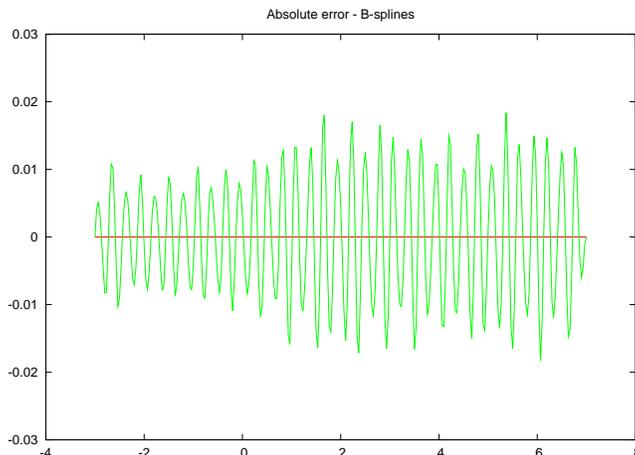
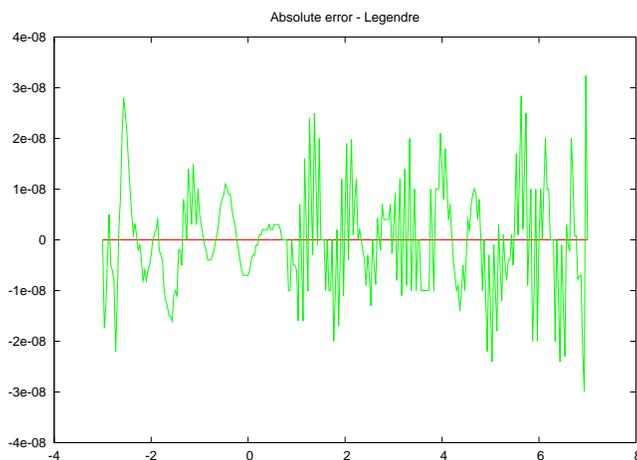


FIG. 24 – Représentation de l’erreur absolue $f[x^*] - \bar{f}$ pour la paramétrisation B-spline. $err(f) \approx 0.12\%$. Comme le système est bien conditionné (et converge rapidement, voir Figure 22), on en déduit que tous les modes ont été résolus. Pour augmenter la précision de l’approximation, il suffit de rajouter des fonctions splines ; en théorie on n’augmente pas la raideur du système puisque le conditionnement est borné.

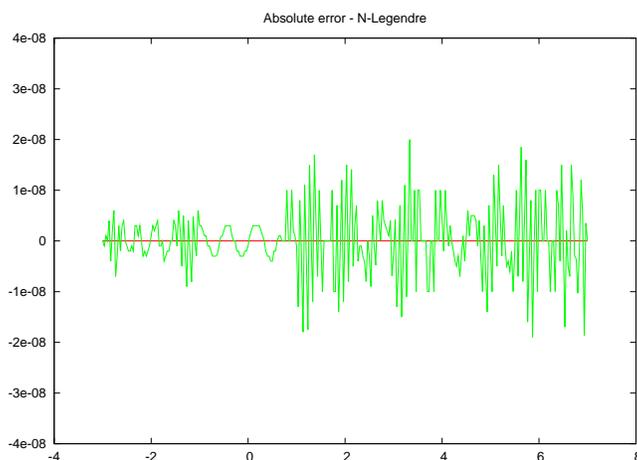
1. Une base optimale, orthonormale pour le produit scalaire (1), a été identifiée de telle sorte que la convergence soit immédiate (c’est-à-dire qu’une itération de descente correspond en réalité à la résolution complète du système linéaire équivalent au problème d’optimisation). On en déduit un résultat de référence.
2. Les hessiens paramétriques ont été étudiés. Pour toutes les paramétrisations, hormis celle dans la base de Tchebychev, les hessiens présentent une structure spectrale proche des modes de Fourier : chaque mode propre est associé à une fréquence identifiée par le nombre d’alternances de signe. Par ailleurs plus la fréquence du mode est élevée plus la valeur propre est petite. Par conséquent, ce sont les modes de hautes fréquences qui convergent le plus lentement, conformément à la vitesse de convergence d’une itération linéaire.
3. Plus le nombre de conditionnement de la matrice hessienne est grand, plus la convergence globale est lente (plus le système est raide). On en déduit que : la paramétrisation raidit le système par rapport à l’approche non-paramétrique ; en général, plus la dimension augmente, plus le conditionnement est mauvais.
4. Il existe une valeur critique sur la dimension du problème pour laquelle le nombre de conditionnement est si grand (de l’ordre de l’inverse de la précision machine) que le problème ne peut pas être résolu complètement. Plus le nombre de conditionnement croît rapidement avec la dimension de la paramétrisation, plus cette valeur critique est petite. Dans le cas de la paramétrisation de Bernstein (la moins bien conditionnée) cette dimension critique est de l’ordre de $n = 25$ (au-dessous de laquelle, bien que lente, la convergence est complète avec un algorithme de Quasi-Newton).

Il est clair que dans ces conditions, l’intérêt d’utiliser une paramétrisation telle que celle de Bernstein paraît limité. Cependant les paramètres de conception ne sont pas toujours laissés au choix de “l’optimiseur” (les déformations géométriques peuvent étre imposées par des choix technologiques). Dans ce contexte, on présente dans la section suivante un algorithme multiniveau—inspiré des méthodes multigrilles classiques—qui a pour but d’accélérer la convergence et d’améliorer la précision. On s’attachera uniquement à la base de Bernstein en tant que prototype d’une base mal conditionnée.

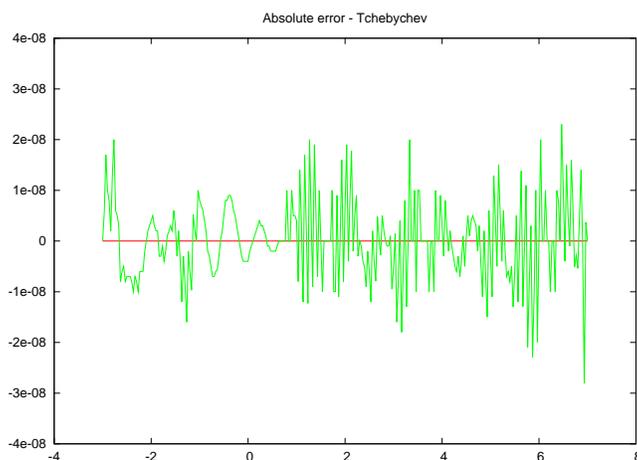
Par la suite nous considérons un problème inverse mal posé (multimodal, donc non convexe) issu d’un problème de conception optimale d’antenne à réflecteur. Cependant, afin de ne pas traiter



(a) Legendre



(b) Legendre normalisés



(c) Tchebychev

FIG. 25 – Représentation de l'erreur absolue $f[x^*] - \bar{f}$ pour les paramétrisations de Legendre et Tchebychev. $err(f) \approx 10^{-7}\%$. Tous les modes ont convergé.

la multimodalité, on cherchera (expérimentalement) un bon point de départ qui nous assurera la convergence vers un minimum supposé global.

4 Méthodes hiérarchiques d'optimisation

On trouve dans la littérature plusieurs méthodes d'optimisation multiniveau. Les méthodes dites *one-shot* cherchent à résoudre dans un même processus itératif le système d'équations non-linéaires composé des conditions nécessaires d'optimalité et de l'équation d'état [16, 2, 17, 19, 9]. Cette méthode n'a donc pas de sens quand l'état est une fonction explicite de la géométrie ou résolu par une méthode directe (équations linéaires). L'algorithme MG/Opt issue des travaux [18, 14, 15] s'inspirent de l'algorithme MG non-linéaire FAS [4, 12, 21, 5, 7] pour accélérer la convergence de l'optimisation. Cette méthode est basée sur la hiérarchisation des espaces d'approximation de la variable d'état. En revanche, dans le cas d'une optimisation paramétrique, la dimension de l'espace de recherche est fixe. On trouve une bibliographie complète sur ce sujet dans [3].

On s'intéresse ici à la hiérarchisation des paramètres de conception. En d'autres termes on cherche à résoudre la raideur liée au problème d'optimisation et non celle liée à la résolution de l'EDP. Ainsi nous ne faisons aucune hypothèse sur la manière dont on résout l'équation d'état.

4.1 Itération linéaire et algorithmes de descente

4.1.1 Généralités

Une itération linéaire s'écrit :

$$x^{i+1} = Gx^i + b \quad (48)$$

Une condition nécessaire et suffisante pour que (48) converge est que $\rho(G) < 1$. Dans ce cas on note x^* le point fixe et $e^i = x^i - x^*$ l'erreur itérative. Comme x^* vérifie $x^* = Gx^* + b$ on a

$$e^{i+1} = Ge^i \quad (49)$$

Si G est diagonalisable on note $G = T\Gamma T^{-1}$, $\Gamma = \text{diag}(0 < |g_1| \leq \dots \leq |g_n| < 1)$. Les vecteurs de T sont les vecteurs propres associés aux valeurs propres g_k . On a donc

$$\begin{aligned} T^{-1}e^{i+1} &= \Gamma T^{-1}e^i \\ \epsilon^{i+1} &= \Gamma \epsilon^i \end{aligned}$$

où $\epsilon^i = Te^i$ par définition et correspond à l'erreur dans la base modale T . Une itération correspond donc à l'atténuation de l'erreur d'un facteur $|g_k|$ dans la direction du mode correspondant :

$$\epsilon_k^{i+1} = g_k \epsilon_k^i. \quad (50)$$

Dans le cas d'une méthode de point-Jacobi avec coefficient de relaxation τ pour la résolution d'un système $Ax = b$, la matrice d'amplification G_J est

$$G_J = I - \tau A. \quad (51)$$

La matrice d'amplification de k itérations de Jacobi avec les coefficients de relaxation τ_j , $j = 1 \dots k$ s'écrit plus simplement

$$G_k = (I - \tau_k A) \dots (I - \tau_1 A). \quad (52)$$

Remarque 7. Les coefficients τ_j peuvent être réglés de manière efficace (optimale) pour atténuer l'erreur dans la direction des modes correspondant à un sous-ensemble du spectre $[a, b] \subset \sigma$. Il s'agit de la méthode de Richardson ou d'accélération de Tchebychev. En pratique cette méthode est efficace sur les modes correspondants aux grandes valeurs propres.

4.1.2 Point de vue de l'optimisation

Dans le cas d'une fonctionnelle quadratique de hessien H le gradient s'écrit

$$G(x) = Hx - b \quad (53)$$

Donc une itération de plus grande descente s'écrit

$$x^{i+1} = x^i - \tau^i(Hx^i - b) = (I - \tau^i H)x^i + \tau^i b \quad (54)$$

où le coefficient de τ^i peut être fixe ou le résultat d'une optimisation unidimensionnelle le long de $Hx^i - b$. On retrouve la méthode de Jacobi avec $I - \tau H$ comme matrice d'amplification. On peut donc écrire la matrice d'amplification pour k itérations de descente avec recherche de pas linéaire comme G_k de (52).

Dans le cas d'une méthode de Quasi-Newton on a

$$x^{i+1} = (I - \tau^i B^{i-1} H)x^i + \tau^i b \quad (55)$$

La formule BFGS (44) construit à chaque itération une approximation B^i du hessien H . L'inverse de B^i est donc un préconditionneur du gradient Hx^i . La méthode de Quasi-Newton peut donc être vue comme un préconditionnement adaptatif d'une méthode de plus grande descente.

Remarque 8. *Les expériences de la section 3.6 indiquent que le Quasi-Hessien BFGS est un bon préconditionneur puisque la convergence est complète même pour des systèmes extrêmement raides. Cette approche est possible puisque le nombre de degrés de liberté reste faible (<100). Au-delà, il n'est pas garanti que la formule reste un bon préconditionneur et l'itération devient coûteuse puisqu'elle suppose la résolution d'un système linéaire. Par ailleurs, il est clair que si la précision machine est atteinte, ce préconditionneur n'est pas suffisant pour atténuer l'erreur dans la direction des modes qui ne sont pas représentables numériquement.*

Du point de vue de l'optimisation, une petite valeur propre correspond à une direction de faible pente de la fonction coût et une grande valeur propre à une direction de grande pente (la valeur propre correspond à la courbure de la fonction coût dans la direction de son vecteur propre associé). Lorsque la fonction coût est mal conditionnée, cela veut dire qu'il existe à la fois des directions planes et des directions de grande pente. On converge facilement dans les directions de grande pente et difficilement dans les directions quasi-planes (d'où les "oscillations" observées lors de la convergence d'une méthode de plus grande descente). La fonctionnelle est peu sensible aux directions planes; il s'agit en réalité d'une sensibilité relative au choix de la paramétrisation (*i.e.* de la sensibilité par rapport aux fonctions propres de déformation définies par les vecteurs propres).

L'idée de l'algorithme hiérarchique est d'itérer sur des sous-espaces de dimension décroissante générés par les directions les plus planes, afin de supprimer petit à petit les directions sur lesquelles on a bien convergé. Un tel sous-espace est ainsi mieux conditionné.

4.2 Hiérarchisation des espaces de recherche : opérateurs de prolongement et de restriction

On note F l'espace de recherche de référence (la "grille" fine) et V un sous-espace de F (la "grille" grossière). F est isomorphe à \mathbb{R}^n et V à \mathbb{R}^m ($m < n$). Comme on l'a vu à la section 3.2, l'espace de référence est construit à partir d'une base $\{f_k\}_{k=1}^n$ donnée (paramétrisation). Tout élément v de $V \subset F$ peut donc s'écrire dans cette base. D'une manière générale on définit V par

$$V \equiv \left\{ v = \sum_{k=1}^n x_k f_k \in F \mid x = Qy, \forall y \in \mathbb{R}^m \right\} \quad (56)$$

où Q est la matrice de l'application linéaire de \mathbb{R}^m dans \mathbb{R}^n qui donne les composantes d'un élément v de V dans la base de la paramétrisation F (les colonnes de Q sont donc les composantes dans cette base d'une base de V) : c'est l'opérateur de prolongement.

Les sous-espaces V sont considérés comme des espaces de correction. En d'autres termes, pour $x^* \in F$ donné, on considère un problème de minimisation sur le sous-espace affine tel que $x = x^* + \Delta x$ où $\Delta x \in V$. Ainsi la fonctionnelle paramétrique dans un tel sous-espace devient

$$j(y) = J(x^* + Qy) \quad (57)$$

Le gradient de j s'écrit donc

$$g(y) = Q^T G(x^* + Qy) \quad (58)$$

et le hessien

$$h(y) = Q^T H(x^* + Qy) Q. \quad (59)$$

On distingue trois stratégies multiniveau selon le choix du sous-espace de correction V :

1. les sous-espaces sont des espaces paramétriques emboîtés (méthode Y) ;
2. les sous-espaces sont des espaces paramétriques emboîtés dont on préconditionne le spectre par une permutation (méthode Z) ;
3. les sous-espaces sont des espaces modaux (méthode Ω).

4.2.1 Paramétrisations emboîtées, méthode Y

Dans un premier temps on suppose que l'on dispose d'espaces de paramétrisation emboîtés $V_1 \subset V_2 \subset \dots \subset V_n$, chacun muni d'une base $\{f_k^m\}$ (par exemple les espaces polynômiaux d'ordre croissant dans la base que l'on veut, les splines d'ordre constant avec augmentation du nombre de splines par ajout de noeuds, etc.). Posons $F = V_n$. On suppose que pour tout m il existe une application linéaire notée E_m^n de \mathbb{R}^m dans \mathbb{R}^n telle que

$$\forall y \in \mathbb{R}^m, \quad x = E_m^n y \Rightarrow v = \sum_{k=1}^m y_k f_k^m = \sum_{k=1}^n x_k f_k^n \in V_m \quad (60)$$

c'est-à-dire que E_m^n est l'application qui exprime les composantes d'un élément v de V dans la base de F à partir de ses composantes dans la base de V_m . Dans le cas des espaces polynômiaux \mathcal{P}^m et \mathcal{P}^n (de dimension $m+1$ et $n+1$ resp.) dans les bases de Bernstein, on appelle *élévation du degré* cette application linéaire. Par extension on appelle *élévation du degré* cette application quelle que soit la base et l'espace considérés.

Dans ce cas on a

$$V_m \equiv \text{vect}\{f_k^m\} \equiv \left\{ v = \sum_{k=1}^n x_k f_k^n \in F \mid x = E_m^n y, \forall y \in \mathbb{R}^m \right\}, \quad Q = E_m^n. \quad (61)$$

Remarque 9. Cette application devient triviale lorsque la base des sous-espace V_m est constituée des m premiers éléments de la base de l'espace de référence F . C'est le cas de la base canonique et des bases orthogonales pour les espaces polynômiaux. On a donc

$$E_m^n = \begin{pmatrix} I_{m \times m} \\ 0_{n-m \times m} \end{pmatrix}.$$

On ne s'intéressera pas à ce cas pour les tests numériques car il a déjà été montré que cette méthode était inefficace. En effet, les sous-espaces ainsi définis contiennent les modes qui ont déjà bien ou complètement convergé lors des itérations sur les espaces de dimension supérieure. Aucun accroissement de la vitesse de convergence ni d'amélioration de la précision n'est observé [22].

4.2.2 Préconditionnement par permutation du spectre, méthode Z

Dans cette deuxième méthode on conserve l'hypothèse selon laquelle nous disposons de paramétrisations emboîtées. On exploite le fait que les modes qui ont du mal à converger sont ceux qui correspondent aux valeurs propres les plus petites et que les modes des sous-espaces correspondent aux valeurs propres les plus grandes. On construit un changement de variable basé sur la diagonalisation du hessien pour affecter les valeurs propres les plus grandes aux modes qui convergent mal.

H est une matrice réelle symétrique donc diagonalisable et telle que les vecteurs propres forment une base orthogonale de \mathbb{R}^n : $H = \Omega\Lambda\Omega^T$, $\Omega^T\Omega = \Omega\Omega^T = I_n$. On suppose (sans perte de généralité) que les valeurs propres λ_k sont triées par ordre croissant. On considère les sous-espaces suivant :

$$V_m \equiv \left\{ v = \sum_{k=1}^n x_k f_k \in F \mid x = E_m^n E_m^{nT} \Omega P \Omega^T E_m^n y, \forall y \in \mathbb{R}^m \right\}, \quad Q = E_m^n E_m^{nT} \Omega P \Omega^T E_m^n \quad (62)$$

où P est la matrice de permutation

$$P = \begin{pmatrix} & & 1 \\ & \dots & \\ 1 & & \end{pmatrix}. \quad (63)$$

Cette méthode a déjà fait l'objet d'expérimentations numériques qui ont abouti à des résultats satisfaisants sur le problème modèle [22]. Elle suppose cependant de disposer de plusieurs paramétrisations emboîtées et est assez délicate à mettre en œuvre.

4.2.3 Sous-espaces propres, méthode Ω

Dans cette dernière méthode on s'affranchit des paramétrisations emboîtées. On ne considère que l'espace F muni de sa base $\{f_k\}$. Les sous-espaces sont construits directement à partir de la diagonalisation du hessien H : le sous-espace de dimension m est l'espace généré par les m premiers vecteurs propres de H , c'est-à-dire

$$H = \Omega\Lambda\Omega^T = (\Omega_1 \quad \Omega_2) \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \begin{pmatrix} \Omega_1^T \\ \Omega_2^T \end{pmatrix} \quad (64)$$

où

$$\Omega_1 = \begin{pmatrix} \omega_1 & \dots & \omega_m \end{pmatrix}, \quad \Omega_2 = \begin{pmatrix} \omega_{m+1} & \dots & \omega_n \end{pmatrix} \quad (65)$$

et

$$V_m \equiv \text{vect}\{\Omega_1\} \equiv \left\{ v = \sum_{k=1}^n x_k f_k \in F \mid x = \Omega_1 y, \forall y \in \mathbb{R}^m \right\}, \quad Q = \Omega_1. \quad (66)$$

Le hessien paramétrique du sous-espace est

$$h = \Omega_1^T H \Omega_1 = \Lambda_1, \quad (67)$$

ce qui implique que le conditionnement de h est nécessairement meilleur puisque

$$\kappa_2(h) = \frac{\lambda_m}{\lambda_1} < \frac{\lambda_n}{\lambda_1} = \kappa_2(H). \quad (68)$$

Remarque 10. Lorsque le problème est soumis à des contraintes linéaires d'égalité, H représente le hessien projeté. Les vecteurs propres Ω appartiennent à l'espace projeté. Afin d'exprimer la correction dans l'espace de référence il est nécessaire de représenter les vecteurs propres dans l'espace de référence à l'aide de la base orthogonale du noyau des contraintes Z , soit $x = Z\Omega_1 y$ et $Q = Z\Omega_1$. Ainsi, chaque sous-espace de recherche n'est soumis à aucune autre contrainte supplémentaire puisque la base qui le construit les contient déjà toutes. Il est donc suffisant de considérer ces contraintes sur l'espace de référence uniquement.

4.3 Algorithme bigrille idéal

On rappelle les éléments de base d'un algorithme bigrille idéal pour la résolution d'un système linéaire afin de faire le parallèle entre l'algorithme classique et l'algorithme en optimisation.

Soit A une matrice s.d.p. et b un vecteur quelconque. Le système $Ax = b$ est équivalent au problème de minimisation $\min_x J(x) = \frac{1}{2}x^T Ax - x^T b$. Le gradient de J est $G(x) = Ax - b$. Le résidu est $r = Ax - b$. Quelque soit x on également l'égalité $A(x - x^*) = Ax - b = r$ où x^* est la solution et e l'erreur.

4.3.1 Multigrille classique

L'efficacité des stratégies multigrilles réside dans la complémentarité de deux méthodes. L'espace fin E est la somme de deux sous-espaces supplémentaires BF et HF. Un lisseur est une méthode efficace pour résoudre dans l'espace HF de l'erreur. La CGC est le complément, résolu sur une grille grossière, de complexité inférieure et de conditionnement plus petit. Un cycle bigrille idéal est composé de phases de lissage et de CGC. On peut mettre en évidence de manière rigoureuse la complémentarité de ces méthodes en analysant la matrice d'amplification équivalente au cycle.

L'algorithme bigrille idéal est composé de deux phases de relaxation sur la grille fine (pré- et post-) et d'une phase de correction exacte sur la grille grossière :

1. Relaxation (k itérations de descente) :

$$x^{(1)} = G_k x^{(0)} + b'$$

2. Correction Grille Grossière (résolution complète) :

$$r = Ax^{(1)} - b$$

$$RAPe = Rr$$

$$x^{(2)} = x^{(1)} - Pe = x^{(1)} - P(RAP)^{-1} Rr$$

3. Relaxation (k itérations de descente) :

$$x^{(3)} = G_k x^{(2)} + b''$$

La matrice d'amplification du cycle est

$$G = G_k \left[I - P(RAP)^{-1} RA \right] G_k \quad (69)$$

4.3.2 Multigrille en optimisation

En optimisation on a

1. Relaxation (k itérations de descente) :

$$x^{(1)} = G_k x^{(0)} + b'$$

2. Correction Grille Grossière (résolution complète) :

$$\Delta x = P \arg \min_{y \in V} j(y)$$

$$x^{(2)} = x^{(1)} + \Delta x$$

3. Relaxation (k itérations de descente) :

$$x^{(3)} = G_k x^{(2)} + b''$$

Puisque le gradient est linéaire on a

$$g(y) = P^T(Ax^{(1)} + Py) - b = P^T APy + P^T(Ax^{(1)} - b) = P^T APy + P^T r \quad (70)$$

Les conditions nécessaires d'optimalité sur la grille grossière $g(y^*) = 0$ nous donne

$$y^* = - (P^T AP)^{-1} P^T r \quad (71)$$

et donc puisque $\Delta x = Py^*$

$$x^{(2)} = x^{(1)} - P (P^T AP)^{-1} P^T r \quad (72)$$

On en déduit donc la matrice d'amplification du cycle

$$G = G_k \left[I - P (P^T AP)^{-1} P^T A \right] G_k \quad (73)$$

qui a exactement la forme de la matrice d'amplification classique 69. On note toutefois que dans ce cas, l'opérateur de restriction est nécessairement le transposé de l'opérateur de prolongement.

4.3.3 Rayon spectrale du cycle idéal

4.4 Algorithme multigrille

Nous rappelons que l'idée générale de l'algorithme multiniveau (voir Algorithme 1) est de permettre aux modes de hautes fréquences, ou plus généralement, aux modes correspondants aux valeurs propres les plus petites, de converger plus rapidement. Nous allons donc utiliser les sous-espaces de recherche définis précédemment afin d'effectuer plus d'itérations dans les directions de ces modes en réduisant petit à petit la dimension du problème.

Comme tout algorithme de descente on considère que l'on dispose d'un vecteur de paramètres initial x^0 . Les sous-espaces de recherche, y compris l'espace de référence F , sont considérés comme des espaces de correction. Plus précisément, si x est un vecteur de paramètres de conception donné (initial ou issu d'une optimisation sur un niveau précédent), on cherche à résoudre le problème de minimisation suivant

$$\min_{y \in \mathbb{R}^m} J(x + Qy) \quad (74)$$

pour la correction de x sur le sous-espace V_m . L'optimum, réalisé par y^* , nous donne la correction optimale $\Delta x^* = Qy^*$. On peut à nouveau résoudre le problème (74) pour la correction de $x + \Delta x^*$ sur un autre sous-espace de recherche.

La résolution de chaque sous-problème consiste à effectuer un petit nombre d'itérations d'une méthode de Quasi-Newton. En ce sens, la correction définie par y^* n'est pas optimale mais est censée avoir suffisamment réduit l'erreur dans la direction des modes associés aux plus grandes valeurs propres (il s'agit alors d'un "anti-lisseur" si l'on se réfère à l'analyse spectrale de la section 3.5). Cependant, par analogie avec les méthodes multigrilles classiques, on considérera une convergence complète sur le niveau de plus faible dimension, soit à l'aide d'une méthode de Newton-Raphson (NR, *i.e.* inversion du hessien exacte) soit avec une méthode de Quasi-Newton (QN, jusqu'à ce que $r^i < \varepsilon$).

On appelle cycle une séquence de sous-espaces de recherche pour la correction successive des paramètres de conception. Un tel cycle est entièrement défini par la dimension des sous-espaces : m_1, m_2, m_3, \dots . Ce cycle peut être répété autant de fois que nécessaire. Des exemples typiques de cycle sont illustrés sur les Figures 26 et 27.

4.5 Résultats numériques sur le problème modèle

On considère le cas test de la section 3.6.2. On compare les courbes de convergence de la fonctionnelle des méthodes classique (SL, rouge, +) et multiniveau (ML, vert, ×). Les deux cas suivants seront distingués :

Algorithme 1 : Algorithme d'optimisation multiniveau

Données : espace fin F , sous-espaces V_m ($F = V_n$), cycle de sous-espaces $\{m_1, m_2, \dots\}$

Entrées : vecteur initial de paramètres $x^0 \in \mathbb{R}^n$, critère de convergence ε

Sorties : vecteur final x^*

Critère de convergence sur le résidu (45) de la fonctionnelle

tant que $r^i > \varepsilon$ **faire**

Résoudre le problème de correction (74) pour chaque sous-espace du cycle

pour chaque sous-espace V_m **faire**

Vecteur de référence à corriger

$x^* \leftarrow x^i$

*Initialisation de la correction de x^**

$y^0 \leftarrow 0 \in \mathbb{R}^m$

Effectuer n_m itérations de descente ou Résoudre complètement

pour $j \leftarrow 1$ **à** n_m **faire**

$y^j \leftarrow y^{j-1} + \tau^{j-1} d^{j-1}(y^{j-1})$

$x^{i+j} \leftarrow x^* + Qy^j$

$i \leftarrow i + n_m$

$x^* \leftarrow x^i$

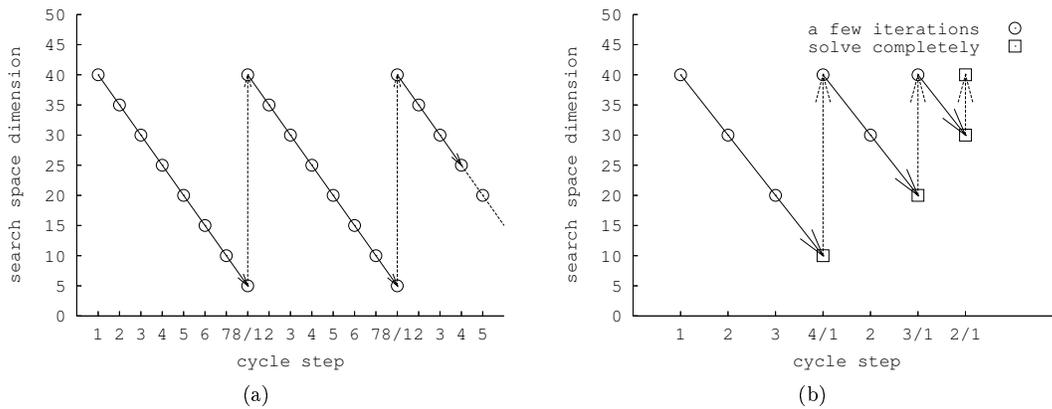


FIG. 26 – Schémas de cycles en “dent de scie”

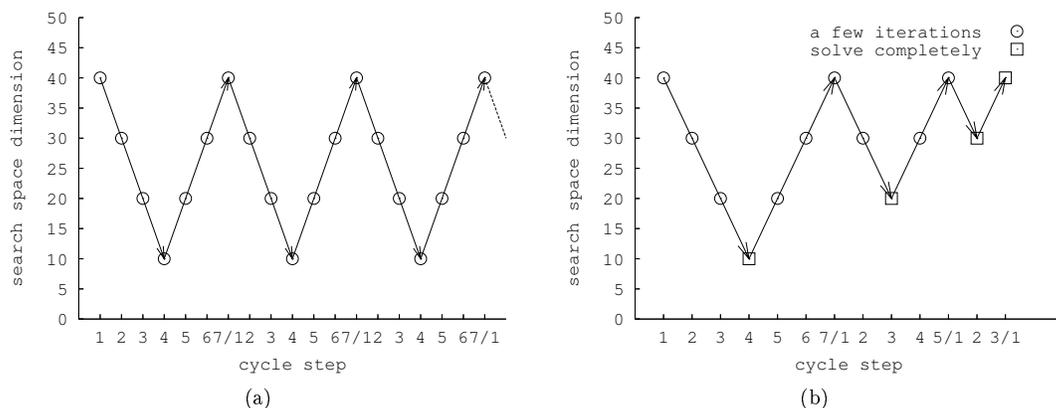


FIG. 27 – Schémas de cycles “en V”

- la dimension du problème est inférieure à la valeur critique (la convergence est complète).
- la dimension du problème est supérieure à la valeur critique (certains modes ne sont pas numériquement inversibles).

4.5.1 Accélération de la convergence

On se place tout d’abord dans le cadre d’une taille de problème inférieure à la valeur critique : $n = 20$. On considère un schéma en “dent de scie” à sept niveaux (voir Tableau 2) pour lequel on souhaite converger complètement sur le niveau de plus faible dimension (stratégie *Full-MG*, voir Figure 26b). Par convergence complète on entend utiliser les stratégies suivantes : soit une méthode directe par inversion du hessien exact (Newton-Raphson, résolution du système projeté sur le sous-espace de recherche) ; soit une méthode itérative de type QN jusqu’à satisfaction du critère de convergence (le résidu (45) est de l’ordre de la précision machine). Sur les autres niveaux on effectue simplement 2 itérations de QN. Les courbes de convergence des deux stratégies se trouvent sur les Figures 28 et 29 respectivement.

niveau	d.d.l.
1	2
2	5
3	8
4	11
5	14
6	17
7	20

TAB. 2 – dimension des sous-espaces de recherche

Résultats des expériences L’expérience de la stratégie NR (voir Figure 28) montre que si l’on est capable de résoudre complètement le problème sur le dernier niveau, alors la convergence est effectivement plus rapide (4 fois plus rapide). L’expérience de la stratégie QN (voir Figure 29) montre qu’en pratique, le coût de la convergence complète avec une méthode itérative sur le dernier niveau est tel que la vitesse de convergence globale est la même. Plus précisément, la convergence est plus rapide sur les premiers niveaux mais ce gain est perdu par la difficulté à résoudre complètement le problème sur les derniers niveaux, bien que le système soit mieux conditionné.

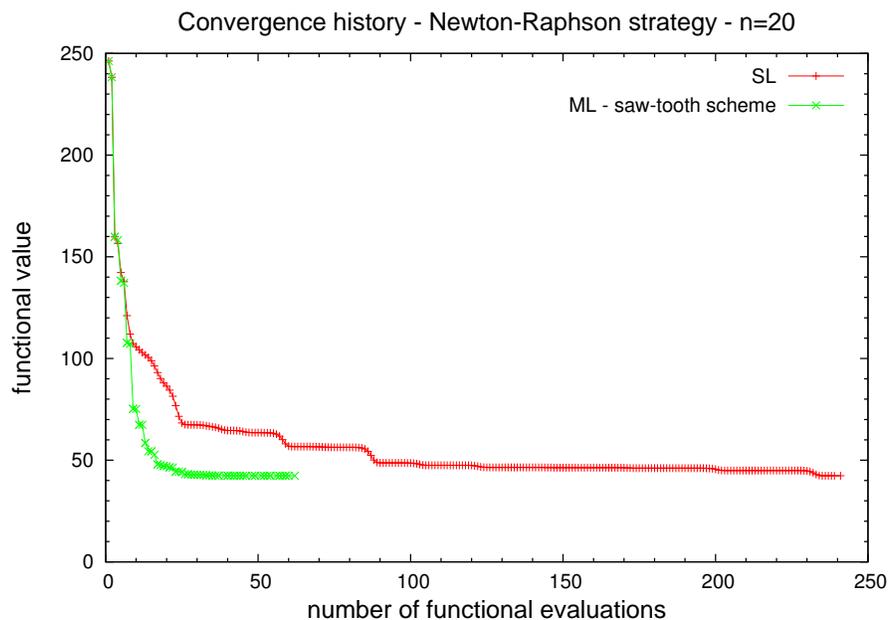


FIG. 28 – Évolution de la fonctionnelle en fonction du nombre d'évaluations ; stratégie de Newton-Raphson (inversion du hessien exact) appliquée sur le niveau de plus faible dimension.

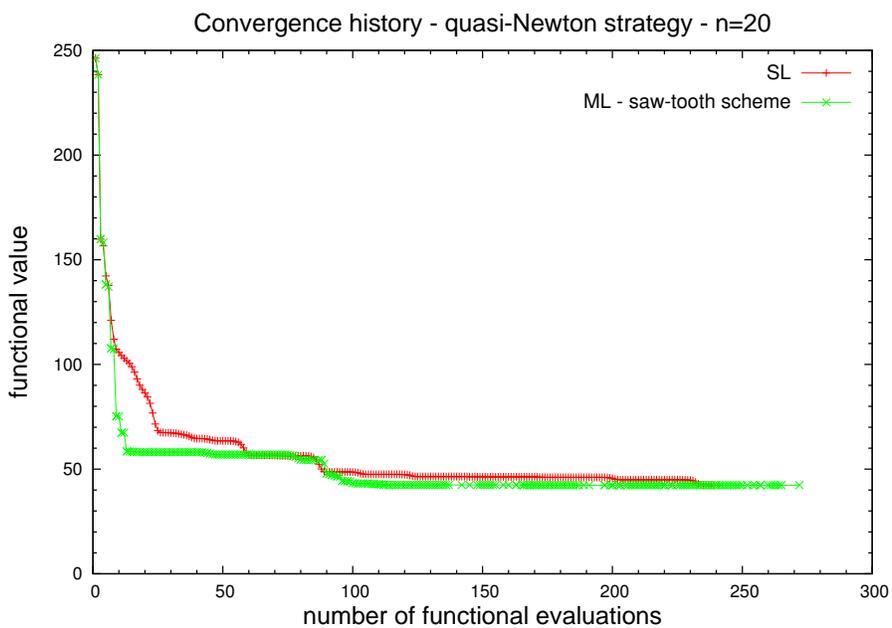


FIG. 29 – Évolution de la fonctionnelle en fonction du nombre d'évaluations ; stratégie de Quasi-Newton (inversions successives d'approximations du hessien) appliquée sur le niveau de plus faible dimension.

Remarque 11 (Remarque : erreurs d'arrondi). *Dans ce type d'exercice il faut être prudent vis-à-vis de l'existence potentiel d'erreurs d'arrondi. On effectue un certain nombre de produits matrice-vecteur pour les changements d'espace de recherche : il se peut que l'on introduise des erreurs d'arrondi à cette occasion. Les résultats obtenus peuvent être bruités par ces erreurs. La principale source d'erreurs d'arrondi est probablement due aux pas de descente sur les derniers niveaux. En effet, même mieux conditionnés, le gradient de ces espaces restent de l'ordre de la précision machine.*

4.5.2 Amélioration de la précision

Désormais on se place dans le cas où $n = 40$. On ne peut pas adopter une stratégie NR comme expérience de référence puisque le hessien n'est pas strictement défini positif à la précision machine. Le hessien de la fonction coût sur un sous-espace modal est la restriction à ces directions du hessien de l'espace de référence. Il conserve donc les mêmes plus petites valeurs propres. Par conséquent, quel que soit le sous-espace, le hessien n'est pas strictement défini positif. La résolution du système (43) (où B^i est le hessien exact) peut donc conduire à une direction qui n'est pas une direction de descente.

On cherche désormais à améliorer la précision sur les espaces de hautes fréquences. Pour cela, la fonctionnelle sur le sous-espace de dimension m est "rééquilibrée" en multipliant Q par le facteur $\frac{1}{\sqrt{\lambda_m}}$ de telle sorte que la valeur propre la plus grande du hessien soit 1. Les courbes de convergence obtenues sont représentées sur les Figures 30 et 31.

La convergence globale est à peine plus rapide avec la méthode hiérarchique. Une analyse de la convergence d'un cycle montre que les itérations dans les espaces hautes fréquences ne sont pas efficaces bien que l'erreur appartienne à ces espaces. La précision obtenue avec rééquilibrage, $J^* = 10^{-4}$, est loin de la précision optimale atteinte dans la base de Legendre, $J^* = 10^{-16}$ (voir Figure 22).

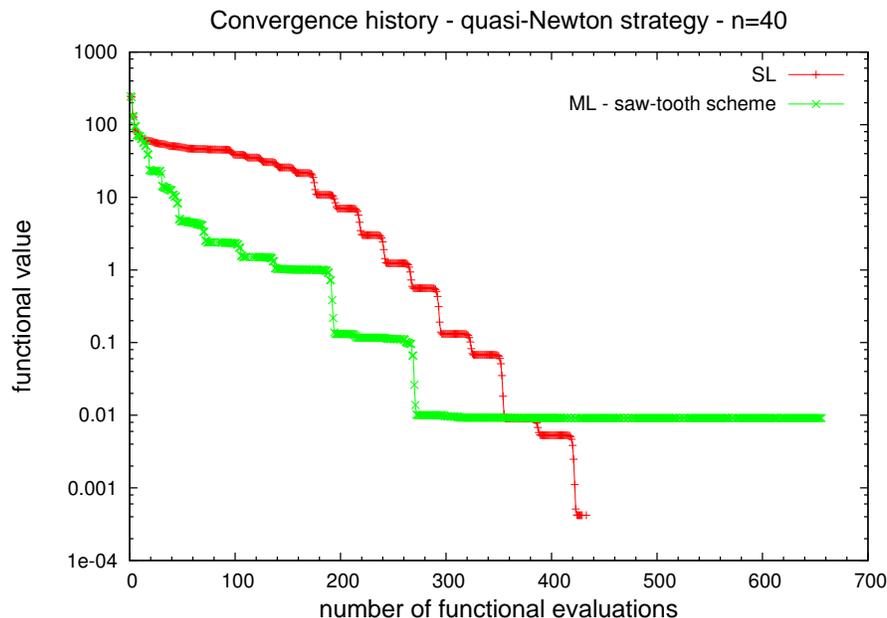


FIG. 30 – Quasi-Newton

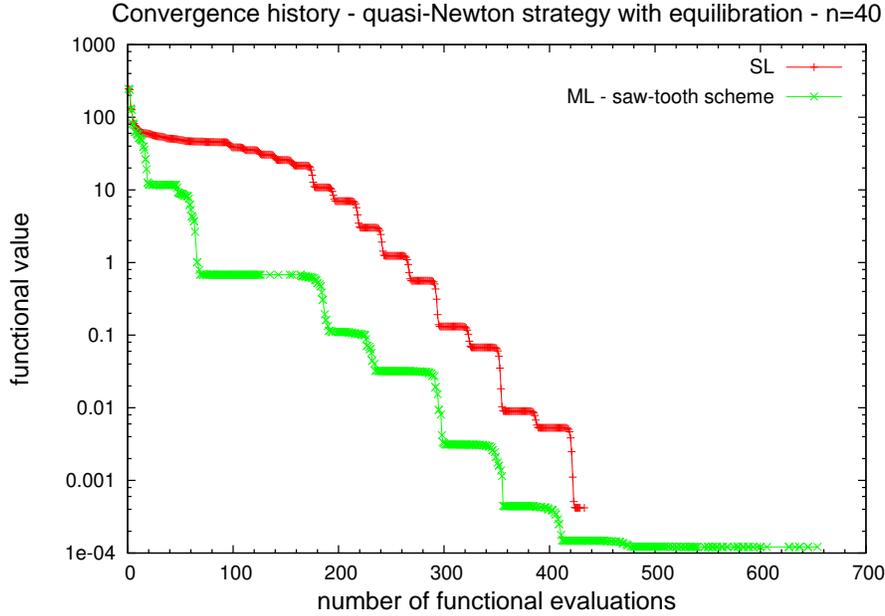


FIG. 31 – Quasi-Newton - équilibration

5 Application à un problème de conformation d'antenne

5.1 Introduction à la conception optimale d'antenne à réflecteur

Une antenne est un système physique destiné à émettre ou recevoir des ondes électromagnétiques (EM). Le rôle des antennes est de transmettre une information sous la forme d'un signal EM entre une antenne émettrice et une antenne réceptrice. Cette transmission d'information se traduit par une transmission d'énergie d'un système à un autre : en réception, le rôle d'une antenne consiste à récupérer l'énergie EM de l'espace libre ; en émission il s'agit de répartir cette énergie dans l'espace libre de telle sorte que les récepteurs puissent la récupérer.

La qualité d'un système donné se mesure à l'aide de grandeurs physiques caractéristiques. La conception d'antennes revient donc à chercher des configurations qui satisfont des critères physiques pertinents. On trouve dans [1] un ouvrage complet sur la modélisation mathématique de fonctionnelles relatives à la conception optimale d'antenne.

On se focalise ici sur l'optimisation géométrique d'antenne à réflecteur. Une antenne à réflecteur est composée d'une source primaire d'ondes électromagnétiques et d'une surface rayonnante \mathcal{S} . On considère ici comme source primaire un dipôle élémentaire placé à l'origine du repère et dont le rayonnement est connu analytiquement. On se place dans le contexte d'une propagation d'ondes en régime harmonique. On suppose également que la surface \mathcal{S} est définie par la révolution d'une courbe \mathcal{C} (méridienne).

Le critère d'optimisation porte sur le rayonnement en champ lointain u . On note S^2 la sphère "à l'infini", *i.e.* l'ensemble des directions de rayonnement. Soit $\mathcal{O}(u)$ une observation de la variable d'état (puissance, directivité, etc.) sur S^2 . Le critère s'exprime comme un problème inverse sur $\mathcal{O}(u)$ où le rayonnement idéal est notée $\bar{\mathcal{O}}$. Le critère s'écrit comme la norme L_2 de la différence entre \mathcal{O} et $\bar{\mathcal{O}}$, soit :

$$\mathcal{J}(u) = \frac{1}{2} \int_{S^2} (\mathcal{O}(u) - \bar{\mathcal{O}})^2 dS^2 \quad (75)$$

L'équation d'état détermine le champ physique u dans la configuration \mathcal{S} . On considère ici le modèle simplifié d'*Optique Physique* comme modèle de propagation : u est alors une fonction

explicite de \mathcal{S} . On en déduit la fonctionnelle de forme :

$$\mathcal{J}(\mathcal{S}) = \mathcal{J}(u(\mathcal{S})) \quad (76)$$

dont on connaît le gradient. Le hessien numérique exact est obtenu par différentiation automatique du gradient à l'aide du logiciel TAPENADE © [13].

5.2 Étude du spectre des hessiens paramétriques à la solution

Dans un premier temps on conduit une analyse numérique des hessiens projetés pour plusieurs paramétrisations. On cherche à savoir si il existe une structure spectrale similaire au cas purement géométrique. En d'autres termes on cherche à déterminer si dans le cas d'un problème d'optimisation de système physique, une itération de gradient peut être considérée comme un lisseur ou un anti-lisseur. En fonction du résultat, on en déduira la pertinence de telle ou telle méthode hiérarchique.

À la différence du problème géométrique, la fonctionnelle n'est pas quadratique. Le hessien n'est donc pas constant ni nécessairement défini positif. Puisqu'on s'intéresse à la convergence d'une méthode de gradient, on ne s'intéresse qu'à la convergence locale dans un voisinage convexe de la solution. Pour cela on définit un problème inverse de la puissance rayonnée où la cible est donnée par le rayonnement d'une forme idéale connue. Le hessien est alors calculé à la solution.

On dispose de hessiens paramétriques exacts pour les paramétrisations de Bernstein, Legendre et Tchebychev. Le spectre est calculé numériquement pour un problème de dimension $n = 8$ (10 moins 2 contraintes d'égalité). Les fonctions propres correspondent à des déformations de la méridienne. La surface rayonnante correspondante est obtenue par symétrie de révolution.

On représente les modes propres sur les Figures 32, 34, 36 et 38 et les spectres sur les Figures 33, 35, 37 et 39. On observe que les modes exhibent une structure spectrale similaire à celle du problème géométrique : on retrouve des modes oscillants à la manière des modes de Fourier, sauf pour la paramétrisation de Tchebychev (Figure 36). Notons que pour cette dernière, les modes ressemblent aux modes propres géométriques dans cette paramétrisation. D'une manière générale le comportement spectral semble être dominé par les variations géométriques plus que par les variations physiques. De même, plus la fréquence du mode augmente, plus la valeur propre associée est petite. Une itération de gradient est donc un anti-lisseur.

Remarque 12. *Si on compare le conditionnement des hessiens, celui dans la paramétrisation de Bernstein est de loin le plus mauvais. On retrouve ici une autre propriété mise à jour avec notre problème modèle. Il se peut que dans cette base, certains modes ne soient pas inversibles. Par conséquent les points de contrôle d'une représentation de Bézier semblent être inadaptés à l'optimisation de forme.*

On cherche désormais à appliquer l'algorithme multiniveau dans le contexte de conception optimale de forme d'antenne. Sachant que les modes "physiques" et les modes "géométriques" sont proches, deux stratégies sont envisagées :

1. **stratégie algébrique** : les modes qui convergent lentement sont les modes qui correspondent aux valeurs propres les plus petites ; les sous-espaces sont donnés par les vecteurs propres du hessien de la fonctionnelle en chaque début de cycle. Cette méthode est coûteuse puisque le hessien doit être calculé une fois par cycle et le coût du calcul du hessien est largement supérieur au coût d'une itération de descente.
2. **stratégie géométrique** : les modes qui convergent lentement sont des modes de hautes fréquences ; les sous-espaces sont donnés par les vecteurs propres du hessien de la fonctionnelle de meilleure approximation. Les sous-espaces ne sont donc pas optimaux dans le sens où ils ne représentent pas exactement les modes qui convergent lentement. En échange ils requièrent aucun calcul supplémentaire.

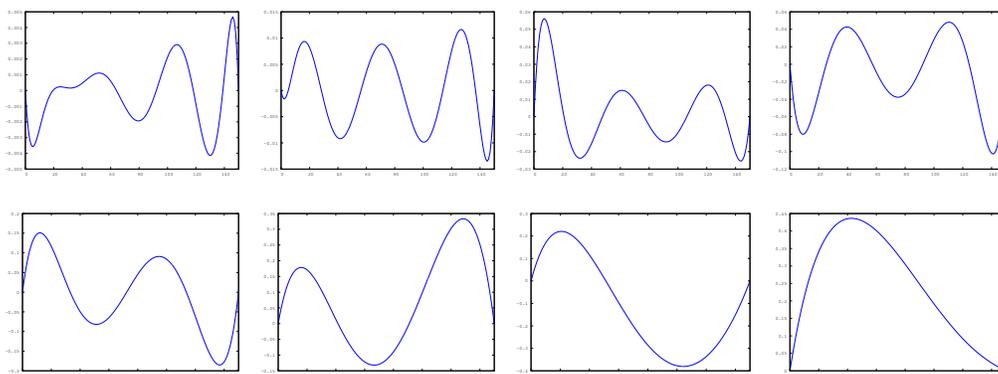


FIG. 32 – Modes propres triés par ordre croissant de leurs valeurs propres - Bernstein.

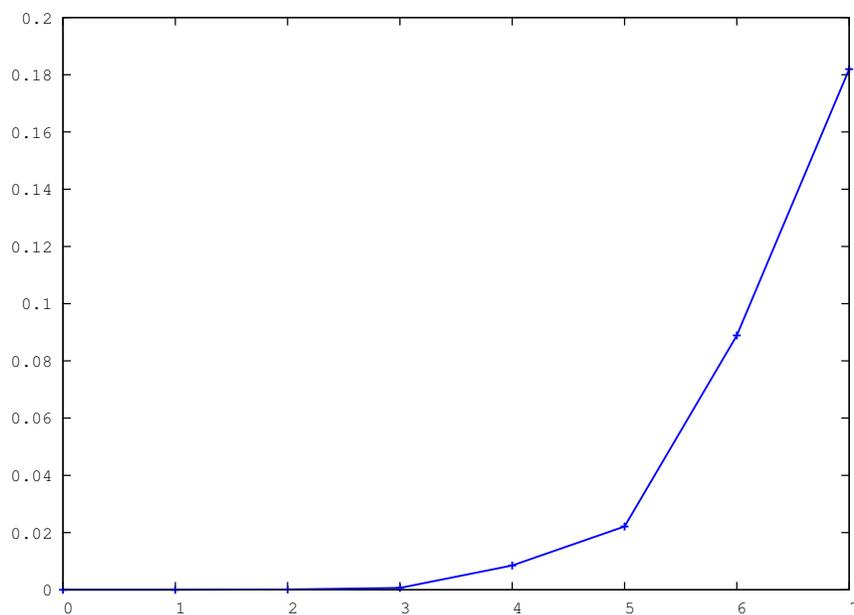


FIG. 33 – Spectre du hessien projeté - Bernstein.

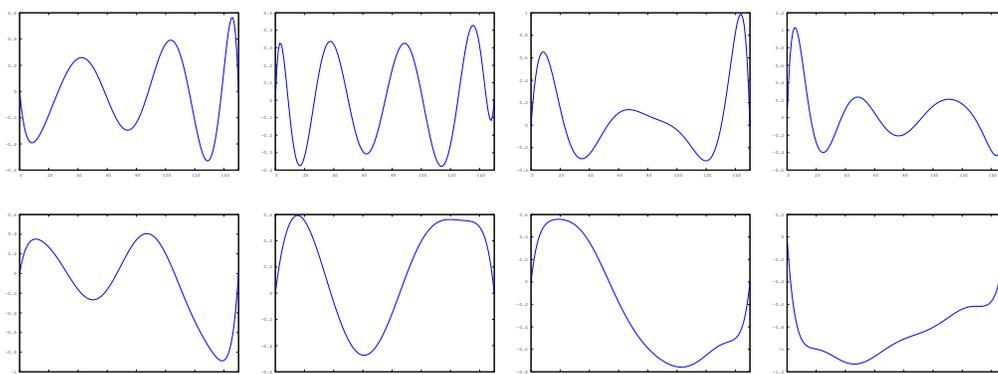


FIG. 34 – Modes propres triés par ordre croissant de leurs valeurs propres - Legendre.

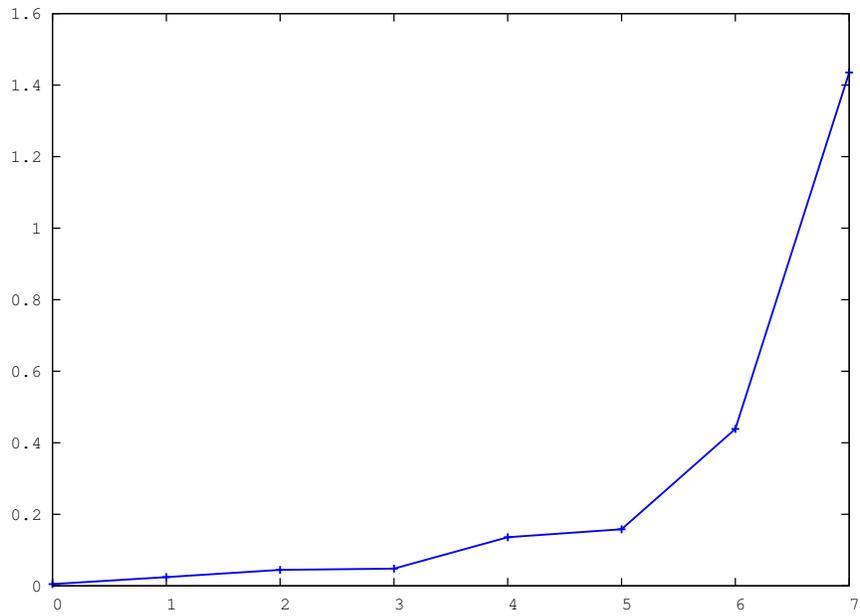


FIG. 35 – Spectre du hessien projeté - Legendre.

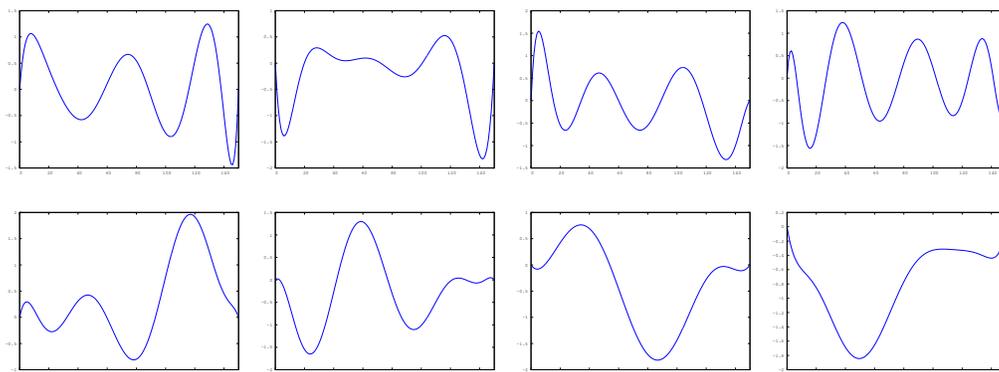


FIG. 36 – Modes propres triés par ordre croissant de leurs valeurs propres - Tchebychev.

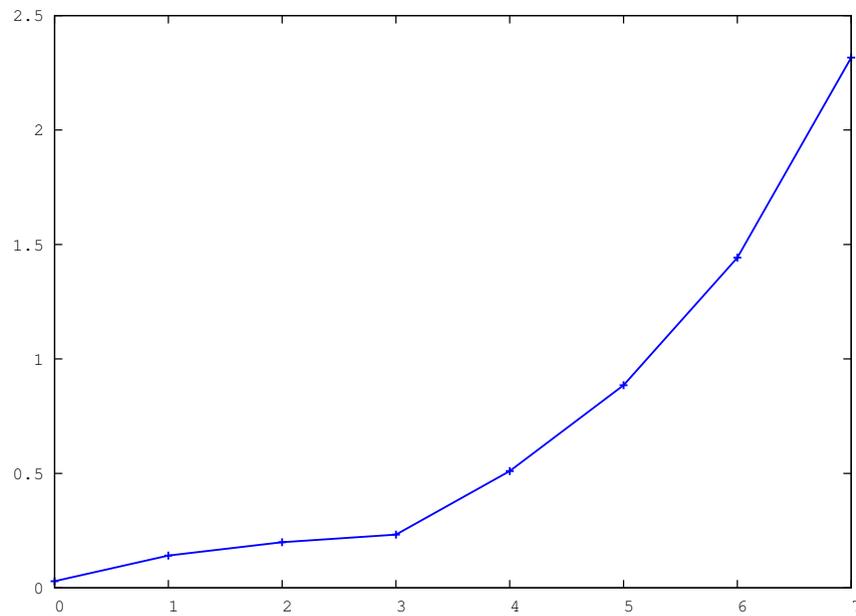


FIG. 37 – Spectre du hessien projeté - Tchebychev.

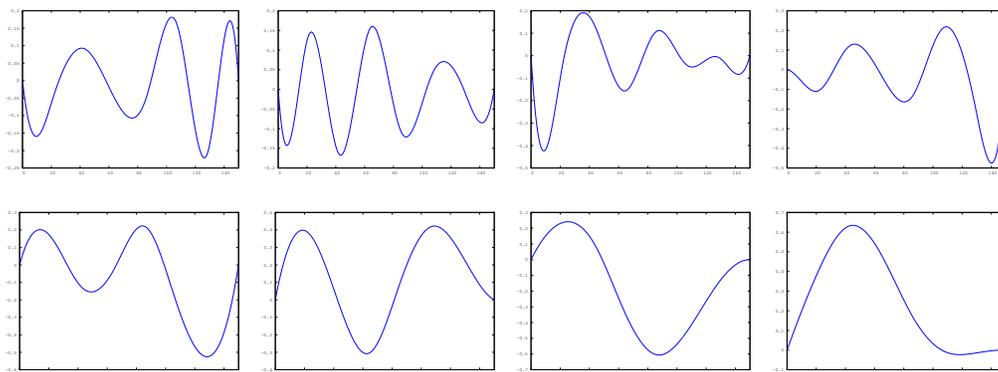


FIG. 38 – Modes propres triés par ordre croissant de leurs valeurs propres - B-splines.

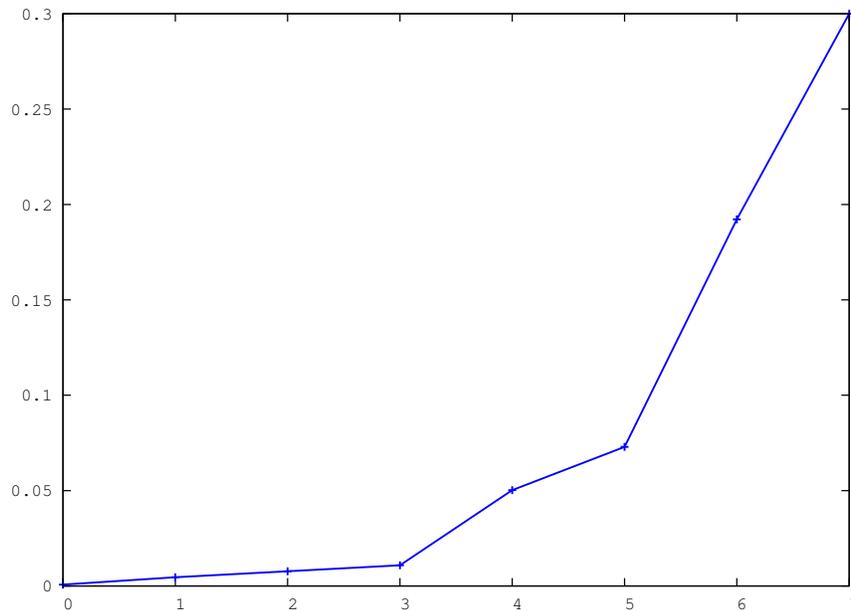


FIG. 39 – Spectre du hessien projeté - B-splines.

5.3 Résultats numériques

Deux cas-test sont considérés : un problème d'uniformisation d'un diagramme en directivité et un problème inverse en puissance. On rappelle que ces problèmes sont multimodaux (mal posés). Or une méthode de gradient n'est pas robuste par rapport aux paramètres de conception initiaux. Soit on s'assure préalablement que le vecteur de paramètre initial nous permet de converger vers la solution désirée lorsque cela est possible, soit on conduit une optimisation préliminaire par une méthode plus robuste (Algorithmes Génétiques, Optimisation par Essaim de Particules, etc.).

5.3.1 Uniformisation d'un diagramme en directivité

Dans un premier temps on met de côté le préconditionnement Quasi-Newton. On considère comme méthode itérative élémentaire un algorithme de Gradient Conjugué. On s'attend à ce que cette méthode, qui se rapproche de l'algorithme de plus grande descente, se comporte comme un anti-lisseur. Par la suite on comparera l'efficacité de l'algorithme multiniveau par rapport à une méthode simple niveau avec préconditionnement Quasi-Newton BFGS.

La Figure 40 montre bien que la méthode multiniveau Ω accélère sensiblement la vitesse de convergence, surtout en début de convergence. Sans surprise, la stratégie algébrique paraît plus efficace encore que la stratégie géométrique. Cependant, seule l'efficacité d'une itération est représentée. Il faudrait également représenter le coût du calcul du hessien pour représenter la complexité globale de l'algorithme. Il apparaît donc que la stratégie géométrique est un bon compromis pour accélérer la convergence sans connaître le hessien de la fonctionnelle.

À la Figure 41, on observe les courbes de convergence de la méthode Z . Ici également la méthode multiniveau est plus performante que la méthode simple niveau. Elle paraît cependant moins efficace au début et plus efficace après quelques cycles.

Enfin on compare ces résultats avec une méthode de Quasi-Newton dans les bases de Bernstein et Legendre à la Figure 42. Il est évident que la méthode de Quasi-Newton est de loin le meilleur préconditionneur dans cet exercice. Le problème est pourtant raide mais le nombre de degrés de liberté est assez faible pour que la formule BFGS soit efficace. De plus le coût de la résolution du système linéaire nécessaire au calcul de la direction de descente est négligeable.

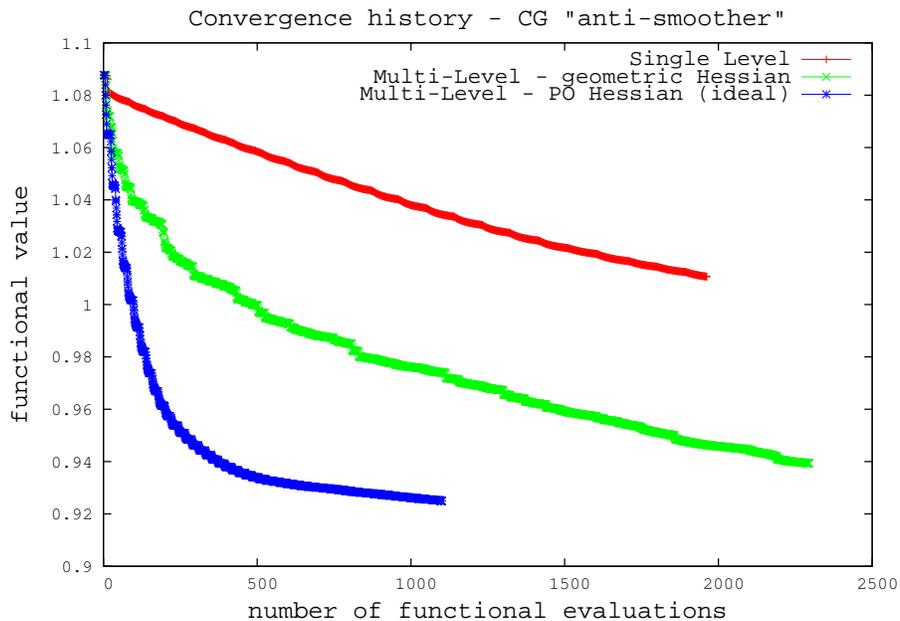


FIG. 40 – Courbe de convergence - méthode multiniveau

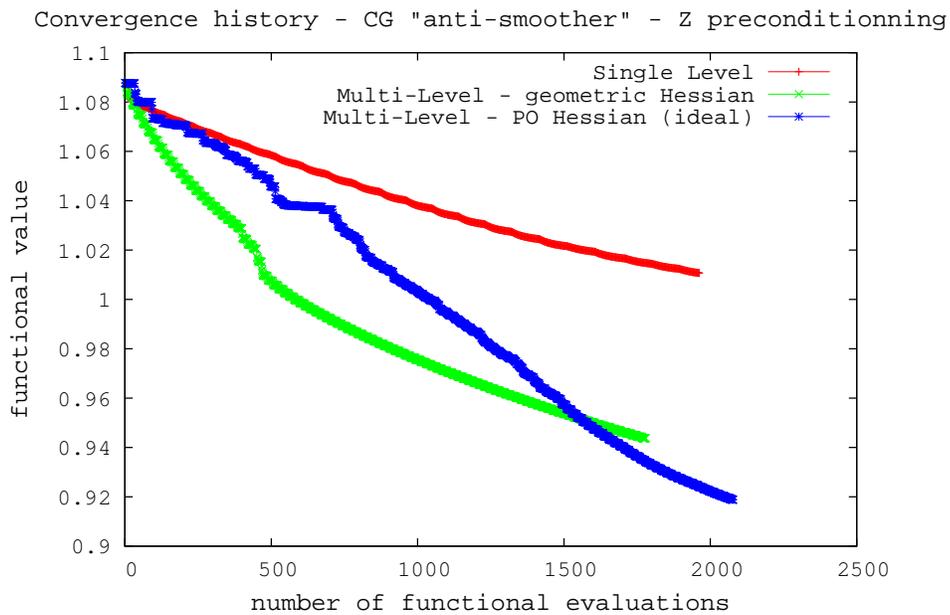


FIG. 41 – Courbe de convergence - méthode multiniveau avec préconditionnement Z

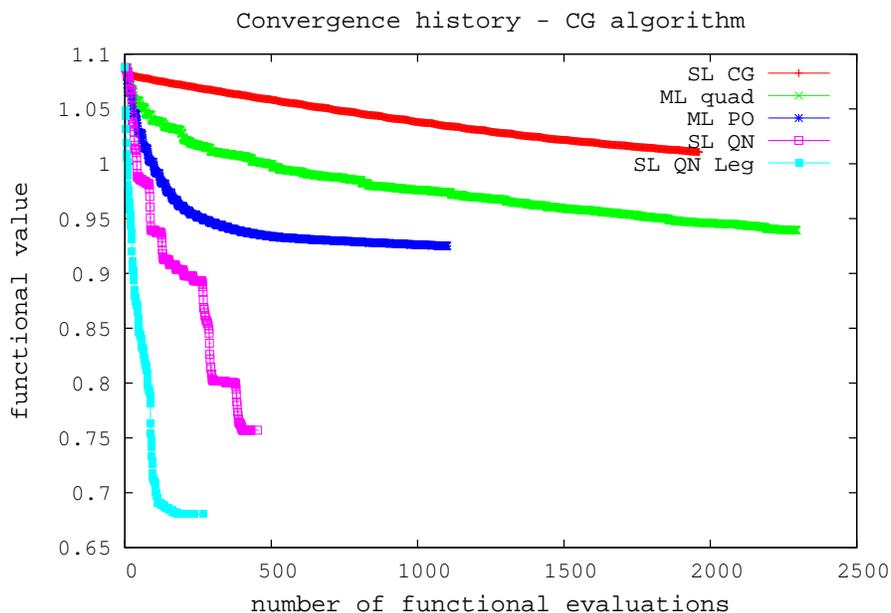


FIG. 42 – Courbe de convergence - Quasi-Newton BFGS

5.3.2 Problème inverse en puissance

Dans ce cas on cherche à appliquer l’algorithme multiniveau avec un algorithme QN-BFGS comme “anti-lisseur”. D’après les Figures 44 et 44, la méthode multiniveau, avec ou sans résolution “exacte” (Newton-Raphson) sur le niveau grossier, n’améliore rien : les courbes de convergence sont quasiment identiques.

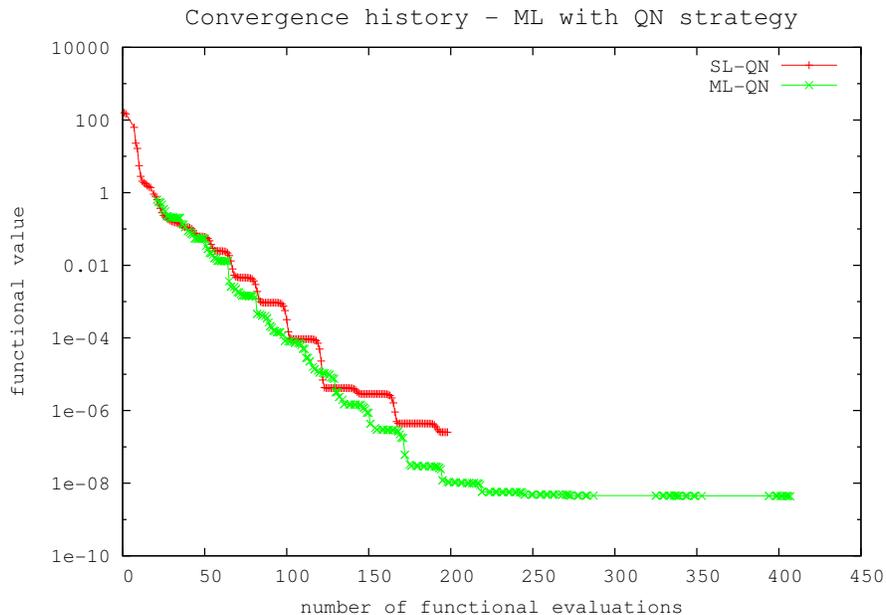


FIG. 43 – multiniveau, Quasi-Newton

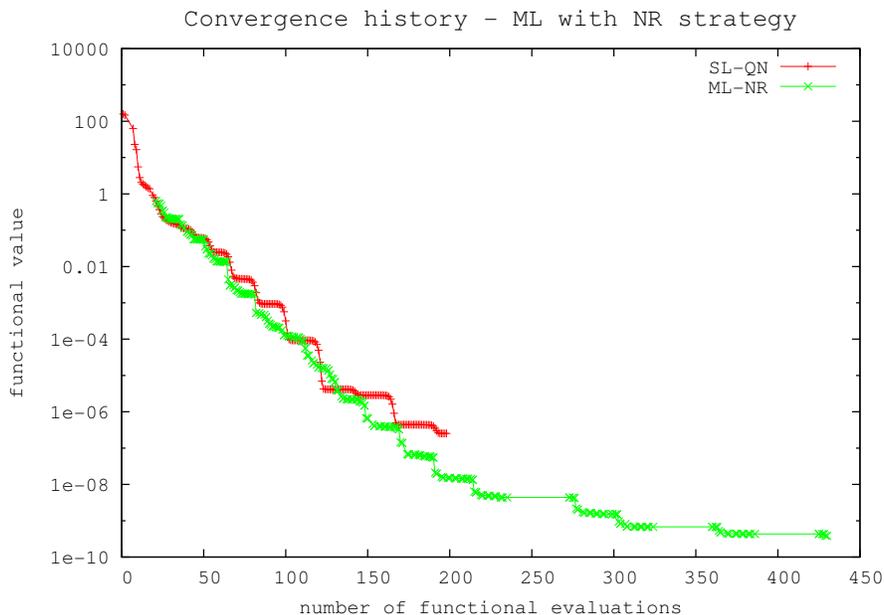


FIG. 44 – multiniveau, Newton-Raphson

6 Conclusion et perspectives

On a considéré un problème modèle d’optimisation de forme, purement géométrique et un problème de conception optimale de forme d’antenne. On a conduit une analyse spectrale des hessiens paramétriques afin de déterminer des propriétés de convergence de méthodes itératives de descente. Ces études ont montré que

1. les modes propres des hessiens paramétriques ont une structure spectrale similaire aux modes de Fourier, tant dans le problème modèle que dans le problème physique; ces modes se ressemblent ce qui tend à montrer qu’ils sont plus influencés par les variations géométriques que par les variations physiques.
2. plus la fréquence du mode est élevée, plus la valeur propre associée est petite; on en déduit qu’une itération de gradient est un “anti-lisseur”, les modes de basse fréquence convergent plus facilement que les modes de haute fréquence.
3. l’approche paramétrique réduit considérablement le nombre de variables de conception mais dégrade le conditionnement; par ailleurs, plus la dimension du problème augmente, plus le conditionnement est mauvais; dans le cas extrême de la paramétrisation de Bernstein, le mauvais conditionnement empêche toute résolution complète au-delà d’une certaine taille, pourtant raisonnable (25 d.d.l); cette observation motive l’utilisation d’une adaptation de la paramétrisation (ou changement de paramétrisation).

En se basant sur ces observations, on a développé une méthode hiérarchique inspirée des méthodes multigrilles pour accélérer la convergence d’un algorithme classique de descente. Les expériences numériques ont montré que

1. le schéma multiniveau de correction “hautes fréquences” est efficace pour accélérer la convergence de méthodes classiques telles que le Gradient Conjugué (non-linéaire).
2. l’algorithme classique de Quasi-Newton BFGS, qui peut être vu comme une méthode de préconditionnement du gradient, est beaucoup plus efficace que les méthodes multiniveau; même si la complexité en temps d’une méthode de QN est de $O(n^3)$, puisque l’approche paramétrique réduit considérablement le nombre de d.d.l., cette méthode reste globalement meilleure.

En résumé, une approche paramétrique dégrade le conditionnement du problème mais la réduction du nombre de d.d.l. permet l'utilisation d'une méthode de Quasi-Newton comme préconditionneur afin de rétablir de bonnes propriétés de convergence.

En perspective nous projetons de vérifier la pertinence des méthodes multiniveau dans les contextes suivants :

1. approche *CAD-free* pour l'optimisation d'antenne où le nombre de d.d.l. est important et où le conditionnement se dégrade avec la dimension ;
2. approche paramétrique avec évolution normale à la surface où seule une méthode de plus grande descente a du sens ;
3. approche paramétrique pour l'optimisation de plusieurs surfaces (optimisation de lentilles multicouches) demandant un grand nombre de variables ; dans ce cas la taille du problème peut limiter l'intérêt des approches Quasi-Newton.

Références

- [1] Thomas S. Angell and Andreas Kirsch. *Optimization Methods in Electromagnetic Radiation*. Springer Monographs in Mathematics. Springer-Verlag, 2004.
- [2] Eyal Arian and Shlomo Ta'asan. Smoothers for optimization problems. In N. Duane Melson, T.A. Manteuffel, S.F. McCormick, and C.C. Douglas, editors, *Seventh Copper Mountain Conference on Multigrid Methods*, volume CP3339, pages 15–30, Hampton, VA, US, 1995. NASA, NASA Conference Publication.
- [3] Alfio Borzì and Volker H. Schulz. Multigrid methods for PDE optimization. *SIAM Review*, 2008 (to appear).
- [4] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation, American Mathematical Society*, 31(138) :333–390, 1977.
- [5] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial (2nd ed.)*. Society for Industrial and Applied Mathematics, second edition, 2000.
- [6] James W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
- [7] Jean-Antoine Désidéri. *Modèles discrets et schémas itératifs. Application aux algorithmes multigrilles et multidomains*. Hermès, 1998.
- [8] Jean-Antoine Désidéri. Two-level ideal algorithm for parametric shape optimization. pages 65–85. Institute of Numerical Mathematics, Russian Academy of Sciences, 2006. Proc. of two International Conferences : Moscow, Institute of Numerical Mathematics, Russian Academy of Sciences, Sept. 16-17, 2006 and Houston, 2006.
- [9] Thomas Dreyer, Bernd Maar, and Volker H. Schulz. Multigrid optimization in applications. *Journal of Computational and Applied Mathematics*, 120(1-2) :67–84, 2000.
- [10] Gerald Farin. *Curves and Surfaces for CAGD : a practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, fifth edition edition, 2002.
- [11] Philipp E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London ; New York, 1981.
- [12] W. Hackbusch. *Multigrid Methods and Applications*. Springer-Verlag, New York, U.S., 1985.
- [13] Laurent Hascoët. Tapenade : a tool for automatic differentiation of programs. In *Proceedings of 4th European Congress on Computational Methods, ECCOMAS'2004, Jyväskylä, Finland*, 2004.
- [14] Robert Michael Lewis and Stephen G. Nash. A multigrid approach to the optimization of systems governed by differential equations. In *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, US, 2000. AIAA.
- [15] Robert Michael Lewis and Stephen G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. Sci. Comput.*, 26(6) :1811–1837, 2005.
- [16] Nathalie Marco and François Beux. Multilevel optimization : Application to one-shot shape optimum design. Research Report 2068, INRIA, Sophia-Antipolis, France, 1993.
- [17] Nathalie Marco and Alain Dervieux. Multilevel parameterization for aerodynamical optimization of 3d shapes. Research Report 2949, INRIA, Sophia-Antipolis, France, July 1996.
- [18] Stephen G. Nash. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14 :99–116, 2000.
- [19] Volker H. Schulz. Solving discretized optimization problems by partially reduced sqp methods. *Computing and Visualization in Science*, 1(2) :83–86, 1998.
- [20] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Computer Graphics*, 20(4) :151–160, 1986.
- [21] Peter Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, 1991.
- [22] Ji Chao Zhao, Jean-Antoine Désidéri, and Badr Abou el Majd. Two-level correction algorithms for model problems. Research Report 6246, INRIA, Sophia-Antipolis, France, July 2007.



Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399