



HAL
open science

Modélisation de surfaces à l'aide de fonctions splines : Mahammed Tazeroualti

► **To cite this version:**

Mahammed Tazeroualti. Modélisation de surfaces à l'aide de fonctions splines :. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1993. Français. NNT : . tel-00343495

HAL Id: tel-00343495

<https://theses.hal.science/tel-00343495>

Submitted on 1 Dec 2008

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.

THESE

présentée par

Mohammed TAZEROUALTI

pour obtenir le titre de

DOCTEUR

DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE I

(arrêté ministériel du 5 juillet 1984)

Spécialité : **Mathématiques Appliquées**

**MODELISATION DE SURFACES A L'AIDE DE FONCTIONS SPLINES.
CONCEPTION D'UN VERRE PROGRESSIF.**

Date de soutenance : 26 Février 1993

Composition du jury : Rapporteurs MM. A. LE MÉHAUTÉ (Président)

C. RABUT

Examineurs MM. J.L. CHAVIN

J.F. JOUVET

B. LACOLLE

P.J. LAURENT

Thèse préparée au sein du laboratoire LMC-IMAG
à l'Université Joseph Fourier, Grenoble I.

REMERCIEMENTS

En premier lieu, je tiens à témoigner ma gratitude et ma reconnaissance à monsieur Pierre-Jean LAURENT pour son attention à mon égard. Son aide m'a été d'un grand soutien, tant sur le plan personnel que professionnel.

Messieurs Alain LE MÉHAUTÉ et Christophe RABUT, m'ont honoré en acceptant la lourde tâche d'être les rapporteurs de cette thèse. Je les en remercie.

Messieurs Jean-Louis CHAVIN et Jean François JOUVET m'ont fait un grand plaisir en acceptant de faire partie du jury.

Je tiens à exprimer toute mon amitié à monsieur Bernard LACOLLE pour la sympathie qu'il a eue à mon égard durant mon séjour au sein de l'équipe Géo-Algo.

La période que j'ai passée au sein du laboratoire L.M.C. a été très agréable grâce à la sympathie de tous ses membres. Je garderai un excellent souvenir des "bouffes inter-thésards".

Je tiens à remercier tous mes copains qui ont beaucoup contribué à ce que la période que j'ai passée à Grenoble soit des plus agréables. Je suis très sensible à leur volontarisme pour organiser mon pot de thèse. Puissent-ils croire en toutes mes amitiés.

In the last but not the least, une grande pensée à ma famille, et tout particulièrement à mes parents, qui ont fait d'énormes sacrifices pour que j'arrive jusque là. Ainsi que pour ma femme, pour sa patience durant les longues journées et soirées où je l'ai laissée seule.

C'est à eux que je dois cette thèse c'est à eux que je la dédie

A ma mère

A mon défunt père

A ma femme

INTRODUCTION GENERALE

Ce mémoire se compose de deux parties indépendantes, chaque partie traite d'un sujet.

Dans les chapitres 1 et 2 on rappelle quelques notions que l'on aura à utiliser pour la suite de ce travail.

Dans la première partie, on présente un nouvel algorithme pour la minimisation de fonctionnelles symétriques semi-définies positives. Cet algorithme est une modification de l'algorithme de Gauss-Seidel. La modification consiste à effectuer, après chaque itération de l'algorithme de Gauss-Seidel classique, une projection du vecteur résultant, sur l'orthogonal du noyau de la fonctionnelle à minimiser.

Dans le chapitre 3, on décrit ce nouvel algorithme. Le théorème de convergence est démontré à l'aide de trois lemmes.

Dans les chapitre 4 et 5, on donne deux applications de cet algorithme. Ces applications sont deux nouvelles méthodes pour le lissage de données surfaciques bruitées . Ces méthodes sont basées sur l'opération d'inf-convolution spline. Elles consistent à remplacer la résolution d'un problème de lissage à deux dimensions, qui est généralement très coûteuse, par la résolution d'une suite de problèmes de lissage à une dimension. Pour cela on écrit la fonction spline lissante f comme somme de M fonctions splines f_j : ($f = f_1 + \dots + f_M$). Les fonction f_j sont calculées par résolution d'un problème de lissage à une dimension.

Dans le chapitre 4 chaque fonction f_j est une fonction de révolution autour d'une focale ω_j , f est alors appelée spline multifocale.

Dans le chapitre 5, chaque fonction f_j est une fonction qui ne varie que selon une direction θ_j et est constante dans la direction orthogonale. f est alors appelée spline multidirectionnelle.

Nous donnons à chaque fois des exemples numériques d'application de ces deux modèles. Nous avons aussi testé une méthode pour calculer des focales ou des directions optimales.

Dans la deuxième partie, nous décrivons une méthode pour concevoir un "verre progressif".

Dans le chapitre 6, nous donnons un bref rappel sur les courbures et les directions principales de courbure d'une surface, ainsi que sur quelques éléments d'optométrie. Nous abordons, après, le problème de conception d'un verre progressif. C'est à dire une surface à laquelle on impose certaines conditions sur ces courbures et sur ces directions principales de courbure en certaines zones (haute, basse, intermédiaire, nasale et temporale). Ces zones correspondent chacune à des plages du verre où l'oeil est amené à se promener.

On impose alors aux courbures principales, dans les zones haute et basse, d'être très proches et égales à une valeur c_1 dans la zone haute et une valeur c_2 dans la zone basse. Les valeurs c_1 et c_2 , correspondent respectivement à la correction à apporter pour la vision de loin et à la correction pour la vision de près. On impose aussi à ces courbures d'être très proches et de passer de la valeur c_1 à la valeur c_2 d'une façon régulière, surtout dans la zone intermédiaire.

On impose à l'une des directions principales de courbure, dans les zones nasale et temporale, d'être proche de l'une des directions des axes de coordonnées.

Dans le chapitre 7 et 8, on définit les opérateurs qui nous permettrons d'imposer ces conditions. Chaque opérateur est relatif à une condition et fait intervenir un paramètre (fonction poids) qui module l'importance que l'on donne à cette condition dans chaque zone. On obtient ainsi un opérateur global Q .

Le problème de conception du verre est alors ramené au problème de minimisation de l'opérateur Q . Cet opérateur n'étant pas quadratique, on utilise un procédé itératif pour effectuer cette minimisation. La convergence de ce procédé n'as pas été démontrée théoriquement, elle a été testée d'une façon numérique.

Dans le chapitre 9, on visualise l'influence de chaque opérateur à travers sa fonction poids sur le verre final. On donne, à la fin du chapitre, une exemple d'utilisation du logiciel qu'on a mis en place.

Table des matières

Rappels généraux

1	Fonctions splines de lissage	3
1.1	Problème général de lissage spline	3
1.2	Calcul des fonctions splines polynomiales	4
1.2.1	Lissage par B-splines régularisées	4
1.2.2	Lissage par B-splines produit tensoriel	6
1.3	Choix du paramètre de lissage	8
1.4	Exemple : Lissage par B-splines régularisées de degré 2	10
2	Inf-convolution spline	17
2.1	Opération d'inf-convolution	17
2.2	Problème de lissage par inf-convolution spline	17
2.3	Algorithme de Gauss-Seidel	20
2.3.1	Résolution d'un système linéaire	20
2.3.2	Minimisation d'une fonctionnelle quadratique	21

Première Partie

3	Un algorithme de type Gauss-Seidel pour \dots	27
3.1	Introduction	27
3.2	Algorithme pour le calcul de \bar{y}	27
3.3	Convergence de l'algorithme	29
3.4	Applications	33
4	Lissage par splines Multifocales	35
4.1	Splines monofocales	35
4.1.1	Choix du critère de lissage J_1 :	36
4.1.2	Problème à une dimension :	36
4.2	Splines Multifocales	39
4.3	Algorithme de Gauss-Seidel modifié	41

Table des matières

4.3.1	Hypothèses de l'algorithme	41
4.3.2	Itéré de norme minimale	42
4.3.3	Algorithme	44
4.4	Expérimentations numériques	45
4.4.1	Nombre d'itérations	47
4.4.2	Choix du paramètre de lissage ρ	48
4.4.3	Choix des focales	48
4.4.4	Visualisation de quelques exemples	49
4.5	Remarques	59
4.6	Conclusion	62
5	Splines multidirectionnelles	63
5.1	Splines monodirectionnelles	63
5.2	Splines multidirectionnelles	65
5.3	Algorithme de Gauss-Seidel modifié	66
5.3.1	Hypothèses de l'algorithme	66
5.3.2	Itéré de norme minimale	67
5.3.3	Algorithme	70
5.4	Expérimentations numériques	71
5.4.1	Nombre d'itérations	73
5.4.2	Choix des directions	74
5.4.3	Choix du paramètre de lissage	75
5.4.4	Visualisation de quelques exemples	76
5.5	Remarques	85
5.6	Conclusion	85

Deuxième Partie

6	Introduction	89
6.1	Courbures normales	89
6.2	Courbures et directions principales	90
6.3	Quelques cas intéressants	91
6.4	Éléments d'optométrie	94
6.5	Modélisation	97
7	Quelques opérateurs	99

Table des matières

7.1	Opérateur d'approximation des courbures	99
7.2	Opérateur de sphéricité	100
7.3	Opérateur sur les directions principales	100
7.4	Opérateur de lissage	100
7.5	Opérateur de bord	100
7.6	Opérateur de tension	101
7.7	Opérateur total	102
8	Algorithme de calcul	103
8.1	Calcul de la fonction verre	103
8.2	Itérations sur les dérivées premières	105
8.3	Remarques	106
9	Résultats et expériences numériques	107
9.1	Introduction	107
9.2	Outils de visualisation	107
9.3	Itérations sur les dérivées premières	111
9.4	Influence des fonctions poids	113
9.5	Utilisation du logiciel de conception d'un verre progressif	119
9.6	Exemple	121
9.7	Remarque	130
9.8	Conclusion	130
	Bibliographie	132

Rappels généraux

Chapitre 1

Fonctions splines de lissage

Dans ce chapitre nous présentons les fonctions splines de lissage dans un cadre général. Nous présentons aussi un algorithme de calcul dans le cas particulier des fonctions splines polynomiales en dimension un et en dimension deux. Dans le dernier paragraphe nous traiterons de l'influence et du choix du paramètre de lissage.

1.1 Problème général de lissage spline

Soient \mathcal{X} , \mathcal{Y} et \mathcal{Z} trois espaces de Hilbert réels, L une application linéaire continue de \mathcal{X} sur \mathcal{Y} et K une application linéaire continue de \mathcal{X} sur \mathcal{Z} . On suppose donné un élément z de \mathcal{Z} .

On appelle problème de lissage spline, le problème qui consiste à chercher un élément σ de \mathcal{X} tel que :

$$\|L(\sigma)\|_{\mathcal{Y}}^2 + \rho \|K(\sigma) - z\|_{\mathcal{Z}}^2 = \min_{x \in \mathcal{X}} \|L(x)\|_{\mathcal{Y}}^2 + \rho \|K(x) - z\|_{\mathcal{Z}}^2 \quad (P)$$

où $\|\cdot\|_{\mathcal{Y}}$ et $\|\cdot\|_{\mathcal{Z}}$ sont respectivement les normes associées à \mathcal{Y} et \mathcal{Z} et ρ un réel strictement positif. σ est alors appelée fonction spline de lissage relative à L , K , z et ρ .

Le problème (P) admet une solution pour tout $z \in \mathcal{Z}$ si et seulement si $\mathcal{N}(L) + \mathcal{N}(K)$ est fermé dans \mathcal{X} et cette solution est unique si et seulement si, en plus, $\mathcal{N}(L) \cap \mathcal{N}(K) = \{0\}$.

Si la condition d'unicité n'est pas vérifiée, c'est à dire si $\mathcal{N}(L) \cap \mathcal{N}(K) = N_0$ avec $N_0 \neq \{0\}$, et si σ est solution de (P) alors tout élément de $\sigma + N_0$ est aussi solution.

Remarquons que la condition d'existence est vérifiée si $\mathcal{N}(L)$ est de dimension finie, ce qui est le cas si L est une combinaison convenable d'opérateurs différentiels. Elle l'est aussi si $\mathcal{N}(K)$ est de codimension finie, ce qui est toujours le cas si \mathcal{Z} est de dimension finie.

Pour plus de détails voir [22].

1.2 Calcul des fonctions splines polynomiales

Dans ce paragraphe nous présenterons une méthode de recherche d'une fonction spline de lissage polynomiale solution d'un problème à une dimension. Le cas bidimensionnel s'en déduit d'une façon naturelle.

1.2.1 Lissage par B-splines régularisées

Soient $t_1 \cdots t_n$, n points non nécessairement ordonnés d'un intervalle $[a, b]$ de \mathbb{R} et n réels $z_1 \cdots z_n$ supposés être les n valeurs, entachées d'erreurs, prises par une fonction inconnue f_e en ces points ou les valeurs $k_1(f_e), \dots, k_n(f_e)$ entachées d'erreurs de n fonctionnelles linéaires continues :

$$z_i = k_i(f_e) + \varepsilon_i \quad i = 1 \cdots n$$

Notons $t = (t_i)_{i=1 \cdots n}$ et $z = (z_i)_{i=1 \cdots n}$. $(\varepsilon_i)_i$ est supposé dans la suite une erreur aléatoire de loi gaussienne centrée. Notre objectif est de trouver une fonction spline polynomiale approximant f_e connaissant les z_i .

On se donne une subdivision régulière (u_0, \dots, u_{nx}) de $[a, b]$ de pas $h_x = (b - a)/nx$, et on note S_d l'espace des fonctions splines polynomiales de degré $\leq d$ définies sur $[a, b]$ de classe \mathcal{C}^{d-1} et dont la restriction à chaque intervalle $[u_j, u_{j+1}]$ est un polynôme de degré $\leq d$. S_d est un espace vectoriel de dimension $(nx + d)$. Une base de S_d est donnée par les splines de base, ou B-splines, de degré d , $(B_j)_{j=-d}^{nx-1}$. Chaque B_j est une fonction de classe \mathcal{C}^{d-1} , non négative, de support $[u_j, u_{j+d+1}]$ et polynomiale sur chacun des intervalles $[u_{j+k}, u_{j+k+1}]$ pour $k = 0, \dots, d$. Pour le calcul de ces B-splines voir [26].

On pose $\mathcal{X} = S_d$, L une application linéaire continue définie de \mathcal{X} dans $L^2([a, b])$, $\mathcal{Y} = L(\mathcal{X})$ muni de la norme $\| \cdot \|_{\mathcal{Y}}$:

$$\| y \|_{\mathcal{Y}}^2 = \int_a^b (y(t))^2 dt \quad y \in \mathcal{Y}$$

$\mathcal{Z} = \mathbb{R}^n$ muni de la norme $\| \cdot \|_{\mathcal{Z}}$:

$$\| z \|_{\mathcal{Z}}^2 = \sum_{i=1}^n (z_i)^2 \quad z \in \mathcal{Z}$$

et K une application linéaire continue définie de \mathcal{X} dans \mathcal{Z} . On notera, pour tout $f \in \mathcal{X}$, $K(f) = (k_1(f), \dots, k_n(f))$, où les k_i sont des fonctionnelles linéaires continues définies de \mathcal{X} dans \mathbb{R} .

Nous posons alors le problème de lissage spline suivant :

$$\underset{s \in S_d}{\text{minimiser}} \int_a^b (L(s)(t))^2 dt + \rho \sum_{i=1}^n (k_i(s) - z_i)^2 \quad (P_1)$$

Ce problème admet une solution car \mathcal{Z} est de dimension finie. La condition d'unicité de celle-ci s'écrit :

$$\left. \begin{array}{l} f \in \mathcal{N}(L) \\ (k_1(f), \dots, k_n(f)) = 0 \end{array} \right\} \Rightarrow f = 0$$

Supposons cette condition vérifiée.

(P_1) est appelé problème de lissage par B-splines régularisées.

Pour résoudre le problème (P_1) on écrit les éléments s de S_d sous la forme :

$$s = \sum_{j=-d}^{nx} \alpha_j B_j \quad (\alpha_j)_j \in \mathbb{R}^{(nx+d)}$$

(P_1) se transforme alors en un problème de minimisation dans $\mathbb{R}^{(nx+d)}$ d'une fonctionnelle quadratique. Il s'écrit :

$$\underset{(\alpha_j)_j \in \mathbb{R}^{(nx+d)}}{\text{minimiser}} \int_a^b \left(\sum_{j=-d}^{nx} \alpha_j L(B_j)(t) \right)^2 dt + \rho \sum_{i=1}^n \left(\sum_{j=-d}^{nx} \alpha_j k_i(B_j) - z_i \right)^2 \quad (P_2)$$

La condition d'optimalité, nullité des dérivées partielles premières en $(\alpha_j)_j$, ramène (P_2) à un problème de résolution d'un système linéaire de la forme :

$$M\alpha = D$$

où

$$\begin{aligned} M &= (M_{jl})_{j=-d \dots nx-1}^{l=-d \dots nx-1} \\ D &= (D_j)_{j=-d \dots nx-1} \\ \alpha &= (\alpha_j)_{j=-d \dots nx-1} \end{aligned}$$

et

$$\begin{aligned} M_{jl} &= \int_a^b L(B_j)(t) L(B_l)(t) dt + \rho \sum_{i=1}^n k_i(B_j) k_i(B_l) \\ D_j &= \rho \sum_{i=1}^n z_i k_i(B_j) \end{aligned}$$

On voit que la matrice M est symétrique. De plus, si L et K sont une combinaison convenables d'opérateurs différentiels, la matrice M comporte d sous-diagonales de chaque côté de la diagonale principale. Enfin pour la résolution du système (S) on utilisera la méthode de Choleski, par exemple.

1.2.2 Lissage par B-splines produit tensoriel

Soit Ω un domaine rectangulaire de \mathbb{R}^2 , $\Omega = [a, b] \times [c, d]$. On se donne deux subdivisions régulières, $U = (u_0, \dots, u_{nx})$ de $[a, b]$ et $V = (v_0, \dots, v_{ny})$ de $[c, d]$. On peut alors définir de la même façon que précédemment deux espaces de fonctions splines polynomiales S_d^x sur $[a, b]$ relativement à U et S_d^y sur $[c, d]$ relativement à V . Notons les bases de B-splines correspondantes, $(B_{j_1}^x)_{j_1}$ pour S_d^x et $(B_{j_2}^y)_{j_2}$ pour S_d^y . On construit, par produit tensoriel des deux espaces, l'espace S_d^{xy} :

$$S_d^{xy} = S_d^x \otimes S_d^y$$

La dimension de S_d^{xy} est alors le produit des dimensions des deux espaces. Une base $(C_l)_l$ de S_d^{xy} est donnée par le produit tensoriel des deux bases $(B_{j_1}^x)_{j_1}$ et $(B_{j_2}^y)_{j_2}$:

$$C_{(j_2+d)(nx+d)+j_1+d+1}(x, y) = B_{j_1}^x(x) B_{j_2}^y(y)$$

pour $j_1 = -d \dots nx - 1$ et $j_2 = -d \dots ny - 1$

On note que S_d^{xy} est l'espace des fonctions s de classe $\mathcal{C}^{(d-1)}$ sur Ω et dont la restriction à chaque rectangle $[u_{j_1}, u_{j_1+1}] \times [v_{j_2}, v_{j_2+1}]$ est sous la forme :

$$(x, y) \in [u_{j_1}, u_{j_1+1}] \times [v_{j_2}, v_{j_2+1}] \quad s(x, y) = \sum_{i,j=0}^d a_{ij} x^i y^j$$

De la même façon que précédemment, on se pose un problème de lissage avec $\mathcal{X} = S_d^{xy}$.

$$\underset{s \in S_d^{xy}}{\text{minimiser}} \int_a^b (L(s))^2(t) dt + \rho \sum_{i=1}^n (k_i(s) - z_i)^2$$

Ce problème est identique à (P_1) et se résout exactement de la même façon.

Remarques :

- 1 - On a supposé S_d^x et S_d^y de même degré d uniquement pour alléger les formules et les notations, mais la méthode reste évidemment valable si l'on prend des degrés différents.

- 2 - Le choix de l'application L , du degré des B-splines et de nx est généralement fixé par la nature de la fonction à approximer et l'ordre de continuité qu'on désire.
- 3 - Si on note \mathcal{L} la fonctionnelle semi-hilbertienne définie sur un espace de Hilbert \mathcal{X} par:

$$\mathcal{L}(x) = \int_a^b (L(x))^2(t) dt \quad \text{pour } x \in \mathcal{X}$$

on peut écrire le problème (P_1) sous la forme:

$$\underset{x \in \mathcal{X}}{\text{minimiser}} \mathcal{L}(x) + \rho \sum_{i=1}^n (k_i(x) - z_i)^2$$

\mathcal{L} est appelée critère de lissage et ρ paramètre de lissage.

- 4 - Le problème (P_1) est équivalent au problème suivant:

$$\underset{y \in \mathbb{R}^n}{\text{minimiser}} \langle y, Ay \rangle - 2 \langle y, b \rangle \quad (1.1)$$

En effet, notons:

$$\Phi(y) = \min_{x \in \mathcal{X}, K(x)=y} \mathcal{L}(x)$$

Φ est une forme quadratique réelle définie sur $K(\mathcal{X})$, on peut donc l'écrire sous la forme:

$$\Phi(y) = \langle y, \Omega y \rangle \quad \text{pour } y \in K(\mathcal{X})$$

Si \mathcal{K} est surjective, (P) est alors équivalent à:

$$\underset{y \in \mathbb{R}^n}{\text{minimiser}} \langle y, \Omega y \rangle + \rho \|y - z\|^2$$

Développons le terme $\|y - z\|^2$:

$$\|y - z\|^2 = \|y\|^2 - 2 \langle y, z \rangle + \|z\|^2$$

et éliminons le terme constant $\|z\|^2$. On obtient une écriture de (P) sous la forme (1.1) avec:

$$A = \frac{1}{\rho} \Omega + I \quad \text{et} \quad b = z$$

1.3 Choix du paramètre de lissage

Dans le paragraphe précédent nous avons traité du problème de lissage en considérant le paramètre de lissage ρ donné. Ceci n'est pas le cas en pratique. Le choix de ρ a une grande influence sur le résultat final, il permet de trouver un compromis entre l'aspect lisse de la fonction résultante, imposé par la fonctionnelle \mathcal{L} , et l'approximation des $(z_i)_i$. Plus précisément, si ρ est trop grand on aura plus tendance à minimiser le terme $\sum_{i=1}^n (k_i(x) - z_i)^2$, et donc interpoler les données. Par contre si ρ est trop faible on minimisera \mathcal{L} , et la solution sera une fonction appartenant au noyau de \mathcal{L} .

Plusieurs méthodes sont proposées dans la littérature. Nous avons choisi une méthode, dite de validation croisée généralisée, pour estimer un ρ optimal. Cette méthode est basée sur l'idée de mesurer la validité d'un ρ par sa capacité à prédire une donnée manquante.

Revenons au problème de lissage (P_1) . Pour simplifier, on suppose que les applications $(k_i)_i$ sont définies par :

$$k_i(f) = f(t_i) \quad f \in \mathcal{X} \text{ et } i = 1 \dots n$$

Le problème (P_1) devient :

$$\underset{s \in S_d}{\text{minimiser}} \int_a^b (L(s))^2(t) dt + \rho \sum_{i=1}^n (s(t_i) - z_i)^2 \quad (P_1)$$

On suppose qu'il admet une solution unique. Notons σ_ρ cette solution, et notons Y_ρ le vecteur dont les composantes sont les valeurs prises par σ_ρ aux points t_1, \dots, t_n . On a ainsi défini une application linéaire S_ρ sur \mathbb{R}^n , qui à un vecteur Z associe le vecteur Y_ρ .

$$S_\rho(Z) = Y_\rho \quad Y_\rho = (\sigma_\rho(t_i))_{i=1 \dots n}$$

On confondra S_ρ et la matrice associée.

La méthode de validation croisée généralisée consiste à chercher le ρ qui minimise la fonction de validation croisée généralisée V_0 suivante:

$$V_0(\rho) = \frac{\|Y_\rho - Z\|^2}{(1 - \frac{1}{n} \text{Tr}(S_\rho))^2}$$

$\text{Tr}(S_\rho)$ est la trace de la matrice S_ρ .

Les références sur cette méthode sont nombreuses, par exemple [9], [12], [19], [25], [30], [31], [34], [38], et les références qui y sont citées.

Pour un ρ donné l'évaluation de la fonction V_0 est très coûteuse, car elle nécessite le calcul de $Tr(S_\rho)$. Plusieurs articles publiés sur ce sujet proposent des méthodes de calcul de $Tr(S_\rho)$ dans des cas particuliers. Une méthode très générale et efficace, quand on possède un grand nombre de données, a été introduite par D.Girard. Elle est du type Monte-Carlo. L'idée est d'estimer :

$$\frac{1}{n} Tr(I - S_\rho)$$

par :

$$\frac{W^t(I - S_\rho)W}{W^tW}$$

W étant un vecteur aléatoire de \mathbb{R}^n , dont les composantes sont indépendantes et de loi normale centrée et réduite. Les qualités de cet estimateur sont étudiées dans l'article [8].

A l'aide de l'estimation précédente on ramène le problème de recherche d'une bonne estimation du ρ optimale au problème de minimisation sur \mathbb{R} de la fonction V_1 suivante :

$$V_1(\rho) = \frac{\|Y_\rho - Z\|^2}{\left((W^t(I - S_\rho)W)(W^tW)^{-1}\right)^2}$$

Pour une valeur de ρ donnée le calcul de $V_1(\rho)$ n'est pas très coûteux. Il ne nécessite plus que le calcul de deux splines, une pour le numérateur et une autre pour le dénominateur. Plus loin on donnera des exemples de calculs de la fonction de validation croisée généralisée et du choix du ρ optimal dans un cas de lissage par B-splines régularisées de degré deux.

Remarques :

- 1 - Quand on ne possède pas un grand nombre de données ($n < 500$ par exemple), on procède de la façon suivante :
 - On génère M vecteurs aléatoires W_1, \dots, W_M .
 - Pour $j = 1 \dots M$ on calcule $Tr_j = (W_j^t(I - S_\rho)W_j)(W_j^tW_j)^{-1}$ - $\frac{1}{M} \sum_{j=1}^M Tr_j$ est alors un bon estimateur de $\frac{1}{n} Tr(I - S_\rho)$
- 2 - En général la fonction V_1 est décroissante, jusqu'à atteindre un minimum ρ_{GCV} , puis croissante. Il arrive néanmoins qu'elle reste stationnaire au delà de ce minimum. C'est le cas quand on utilise un faible nombre de B-splines (nx petit).

1.4 Exemple : Lissage par B-splines régularisées de degré 2

On se donne n points $(t_i)_{i=1, \dots, n}$ de l'intervalle $[0, 1]$. A chaque point t_i on associe un réel z_i supposé correspondre à la valeur bruitée prise par une fonction f_e au point t_i :

$$z_i = f_e(t_i) + \varepsilon_i \quad \text{pour } i = 1, \dots, n$$

$\varepsilon = (\varepsilon_i)_{i=1, \dots, n}$ un vecteur aléatoire de loi $\mathcal{N}(0, \nu I)$:

$$\begin{aligned} \mathcal{E}[\varepsilon_i] &= 0 \\ \mathcal{E}[\varepsilon_i \varepsilon_j] &= \nu^2 \delta_{i,j} \end{aligned} \quad \text{pour } i, j = 1, \dots, n$$

(\mathcal{E} dénote l'espérance mathématique et $\delta_{i,j}$ le symbole de Kronecker).

On notera $Z = (z_i)_{i=1, \dots, n}$. On supposera pour la suite que $n \geq 2$.

Pour les exemples qui suivront on prendra f_e la fonction $t \rightarrow \sin(2\pi t)$, ν égal à 0.2 et $n = 600$.

On visualisera dans certaines figures la courbe représentative de la fonction f_e , les données bruitées seront représentées par des points.

Pour approcher f_e , on résout un problème de lissage par B-splines régularisées de degré 2. On se donne alors, une subdivision régulière (u_0, \dots, u_{ns}) de $[0, 1]$, on construit l'espace S_2 relatif à cette subdivision et on considère le problème (P_1) :

$$\underset{s \in S_2}{\text{minimiser}} \left(\int_0^1 (\mathcal{L}(s)(t))^2 dt + \rho \sum_{i=1}^n (s(t_i) - z_i)^2 \right) \quad (P_1)$$

où $\mathcal{L}(s) = s''$.

Le noyau de \mathcal{L} est égal à \mathcal{P}_1 , l'espace des polynômes de degré inférieur ou égal à 1. Pour tout vecteur Z de \mathbb{R}^n et ρ positif, le problème (P_1) admet une solution unique. En effet :

$$\mathcal{P}_1 \cap \{s \in S_2 \text{ telle que } s(t_i) = 0 \text{ pour } i = 1, \dots, n\} = \{0\}$$

Soit σ_ρ la solution de (P_1) . Notons Y_ρ le vecteur dont les composantes sont les valeurs prises par σ_ρ aux points t_i , $i = 1, \dots, n$, et S_ρ l'application linéaire définie sur \mathbb{R}^n telle que :

$$Y_\rho = S_\rho(Z)$$

Comme on l'a précisé précédemment, le choix de ρ et ns influence beaucoup la fonction σ_ρ .

En effet :

Si ρ est trop petit. (figure (1.1)), le problème (P_1) se ramène à chercher la fonction appartenant au noyau de L qui réalise le minimum de $\|Y_\rho - Z\|^2$. La solution sera alors proche de la droite des moindres carrés.

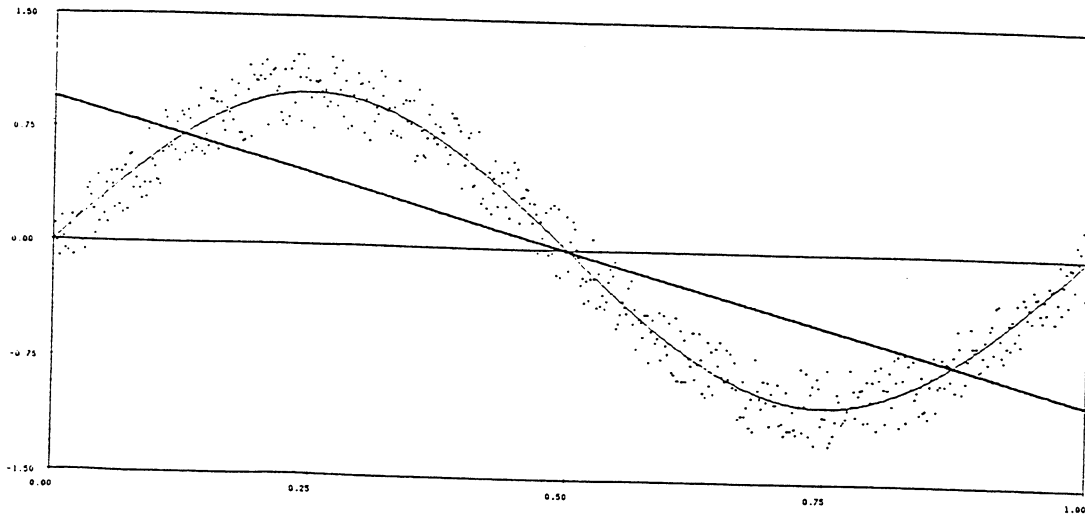


figure 1.1 Lissage avec $\rho = 10^{-8}$

Si ρ est trop grand. le problème (P_1) se ramène à chercher la fonction appartenant à S_2 qui s'écarte le moins des données z_i .

Les deux figures suivantes illustrent l'influence du choix de ns . pour $\rho = 10^8$.

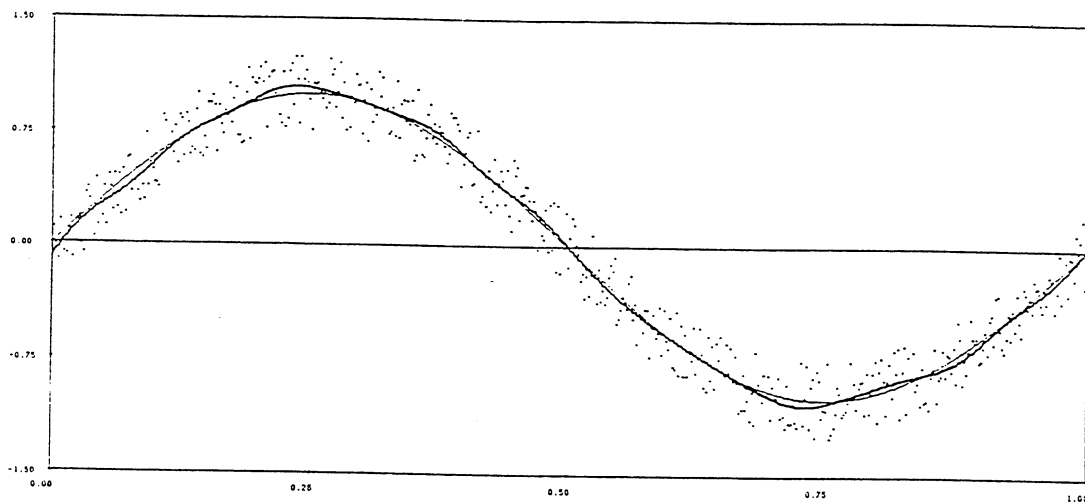


figure 1.2 Lissage avec $\rho = 10^8$ $ns = 10$

Dans le cas de la figure (1.2), où $ns = 10$, on remarque que même avec une très grande valeur de ρ la spline solution de (P_1) reste "assez lisse". Ceci est dû au faible nombre de B-splines qui ne permet pas à la solution d'interpoler les z_i .

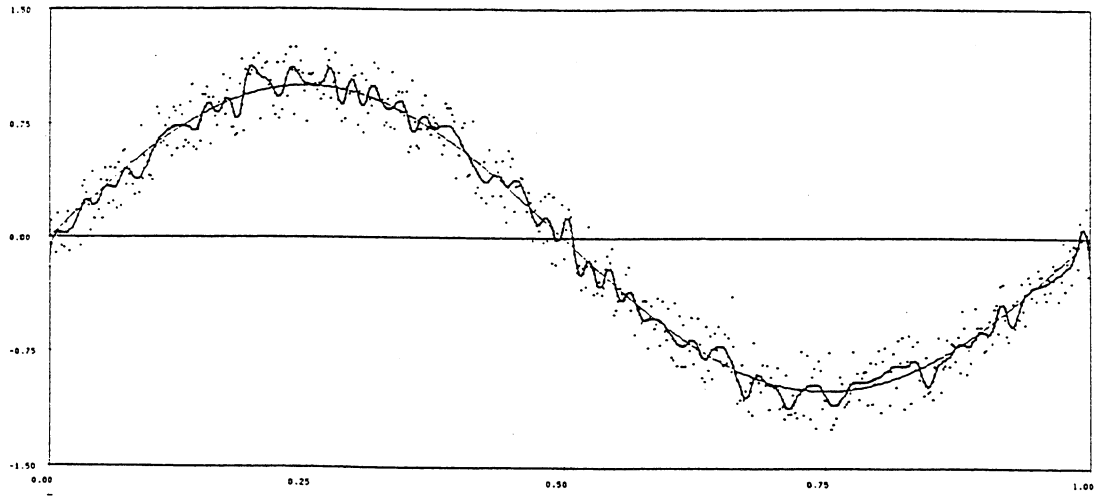


figure 1.3 Lissage avec $\rho = 10^8$ $ns = 100$

Dans le cas de la figure (1.3), où $ns = 100$, on remarque que la solution présente beaucoup plus d'oscillations que dans le cas précédent et tend à interpolier les données z_i $i = 1, \dots, n$.

On conclut de ces deux figures l'effet régulateur de ns . On en déduit aussi l'importance d'un bon choix de ρ et de ns pour l'obtention d'une bonne approximation de f_e . On rappelle que le choix de ns influence aussi le coût de résolution de (P_1) : plus ns est grand plus cette résolution est coûteuse.

Pour choisir ns on se base généralement sur des connaissances a priori sur les données. Il est inutile de choisir ns très grand si on sait que la fonction à approcher n'est pas très oscillante.

Pour le choix de ρ , l'idéal serait de prendre le ρ minimisant l'erreur exacte, $Er(\rho)$, commise en approchant le vecteur $(f_e(t_i))_i$ par Y_ρ :

$$Er(\rho) = \frac{1}{n} \sum_{i=1}^n ((Y_\rho)_i - f_e(t_i))^2$$

Cette quantité n'est généralement pas connue. On la remplace par la fonction de validation croisée généralisée $V_1(\rho)$ qu'on notera aussi $GCV(\rho)$.

$$GCV(\rho) = \frac{\|Y_\rho - Z\|^2}{\left((W^t (I - S_\rho) W) (W^t W)^{-1} \right)^2}$$

où W est un vecteur aléatoire de loi $\mathcal{N}(0, I)$:

$$\begin{aligned} \mathcal{E}[W_i] &= 0 \\ \mathcal{E}[W_i W_j] &= \delta_{i,j} \end{aligned} \quad \text{pour } i, j = 1, \dots, n$$

Dans la figure (1.4), on visualise la trace de $(I - S_\rho)$ en fonction de ρ (en continu), et quatre estimations de Monte-Carlo de cette trace en utilisant quatre vecteurs aléatoires différents (en pointillés).

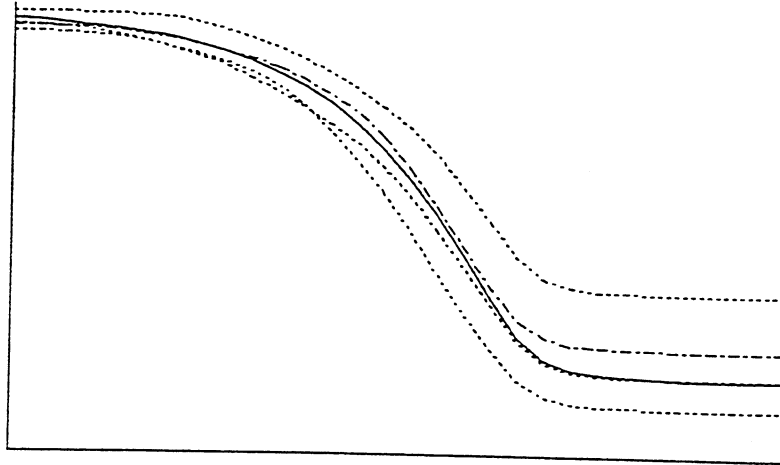


figure 1.4 Trace exacte et 4 estimations par Monte-Carlo $10^{-4} \leq \rho \leq 10^4$ en abscisses

On remarque que ces estimations dépendent du vecteur aléatoire utilisé. Elles restent néanmoins "assez voisines" de la trace exacte.

Dans les deux figures (1.5) et (1.6), on visualise la fonction Er et la fonction GCV pour $ns = 10$ et $ns = 100$. Les courbes représentatives des deux fonctions sont ramenées à la même échelle.

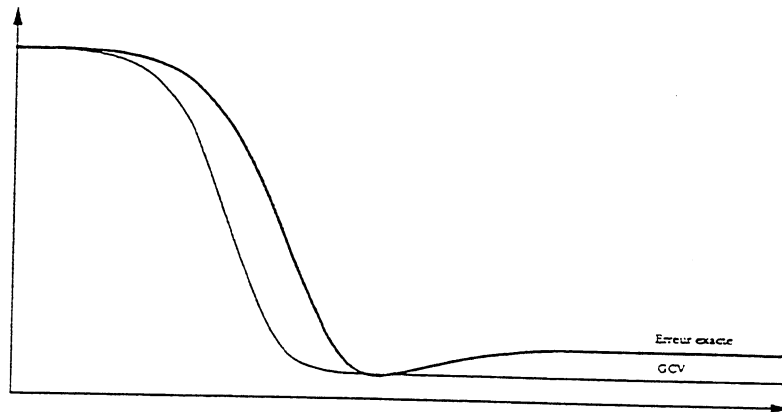


figure 1.5 Erreur exacte et GCV - $ns = 10$ $10^{-4} \leq \rho \leq 10^4$ en abscisses

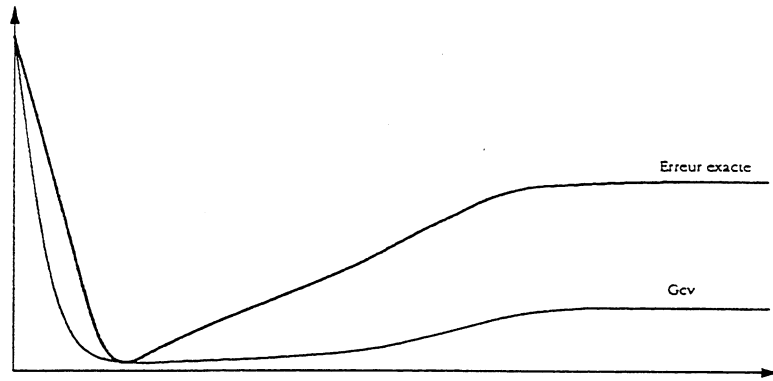


figure 1.6 Erreur exacte et GCV - $ns = 100$ $10^{-4} \leq \rho \leq 10^4$ en abscisses

On remarque que, dans les deux cas, la fonction GCV atteint son minimum, ρ_{GCV} au voisinage du minimum, ρ_{Er} de Er .

En regardant les deux graphiques, on remarque que la fonction Er présente les mêmes variations: elle décroît jusqu'à atteindre sa valeur minimale, puis augmente. Il en est de même pour la fonction GCV .

En comparant l'allure de la fonction Er dans les deux graphiques précédents, on voit bien que pour $ns = 100$, cette fonction présente une plus forte augmentation que dans le cas où $ns = 10$.

On le voit encore plus nettement dans le cas de la figure (1.7), où $ns = 2$.

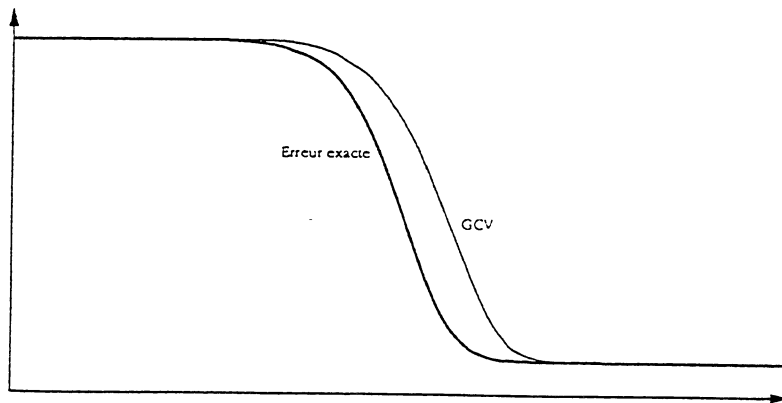


figure 1.7 Erreur exacte et GCV - $ns = 2$

Dans cet exemple, l'erreur exacte est décroissante jusqu'à atteindre sa valeur minimale puis reste pratiquement constante. Ceci veut dire que pour des valeurs de ρ supérieur à ρ_{Er} , la solution σ_ρ ne change pratiquement plus.

Dans la figures (1.8). on visualise les deux fonctions splines solution de (P_1) , l'une relativement à ρ_{GCV} et l'autre relativement à ρ_{Er} .

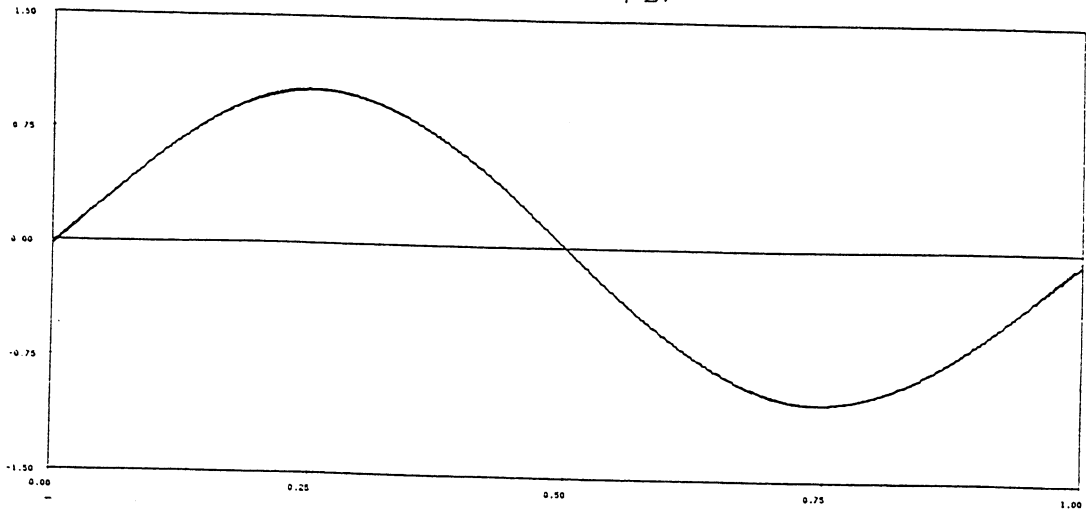


figure 1.8 Splines de lissages pour $ns = 10$

On remarque dans l'exemple que les deux splines de lissage sont "presque confondues" (indiscernables), et approchent bien la fonction f_e .

Pour étudier la validité de la méthode consistant à choisir pour ρ optimal celui qui minimise GCV , on définit l'efficacité, Eff , de cette méthode par :

$$Eff = \frac{Er(\rho_{Er})}{Er(\rho_{GCV})}$$

Plus Eff se rapproche de 1, plus la méthode est bonne.

Pour les deux exemples qu'on a traité on a obtenu les résultats suivants :

ns	Eff
10	0.974
100	0.978

Chapitre 2

Inf-convolution spline

Dans ce chapitre nous rappelons la définition et quelques propriétés de l'opération d'inf-convolution . Nous étudierons un problème de lissage par inf-convolution spline , existence, unicité, et caractérisation de la solution. Nous décrirons, en fin de chapitre, un algorithme de calcul de cette solution.

Les résultats de ce chapitre sont énoncés dans [22], [23], [24] et [27].

2.1 Opération d'inf-convolution

Soit D un domaine de \mathbb{R}^d et $E = \mathbb{R}^D$ et soit $\mathcal{L}_1 \cdots \mathcal{L}_M$ M fonctionnelles semi-hilbertiennes définies sur E , $X_1 \cdots X_M$ leurs domaines et $N_1 \cdots N_M$ leurs espaces nuls (noyaux).

Aux M fonctionnelles $(\mathcal{L}_j)_j$ on associe la fonctionnelle \mathcal{L} par :

$$\forall x \in E \quad \mathcal{L}(x) = \text{Inf} \left\{ \left(\sum_{j=1}^M \mathcal{L}_j(x_j) \right) / x_j \in X_j \text{ et } \sum_{j=1}^M x_j = x \right\} \quad (2.2)$$

\mathcal{L} est appelée l'inf-convolution des $(\mathcal{L}_j)_j$, et on note :

$$\mathcal{L} = \bigvee_{j=1}^M \mathcal{L}_j$$

\mathcal{L} est une fonctionnelle semi-hilbertienne . Son domaine X est la somme des domaines X_j et son noyau N est la somme des N_j ,

$$X = X_1 + \cdots + X_M$$

$$N = N_1 + \cdots + N_M$$

2.2 Problème de lissage par inf-convolution spline

On se donne n points $(t_i)_{i=1 \dots n}$ de D et un vecteur $Z = (z_i)_{i=1 \dots n}$ de \mathbb{R}^n supposé correspondre aux valeurs bruitées, prises par une fonction inconnue.

On se donne M fonctionnelles semi-hilbertiennes $(\mathcal{L}_j)_j$ sur E et on note $Spl_j(Z)$ la spline de lissage correspondant à \mathcal{L}_j pour les données Z et $Y_j = S_j(Z)$ le vecteur résultant pour $j = 1 \cdots M$. C'est à dire le vecteur définit par :

$$(Y_j)_i = S_j(Z)_i = (Spl_j(Z))(t_i) \quad \text{pour } i = 1 \cdots n$$

On construit la fonctionnelle semi-hilbertienne \mathcal{L} par inf-convolution des $(\mathcal{L}_j)_j$ et on considère le problème de lissage :

$$\underset{x \in E}{\text{Minimiser}} \left(\mathcal{L}(x) + \rho \sum_{i=1}^n (x(t_i) - z_i)^2 \right) \quad (I)$$

Ce problème admet une solution unique si et seulement si $N \cap I = \{0\}$, c'est à dire :

$$\left(\sum_{j=1}^M N_j \right) \cap I = \{0\}$$

ou encore :

$$\left(x_j \in N_j, \quad j = 1 \dots M \quad \text{et} \quad \sum_{j=1}^M x_j \in I \right) \Rightarrow \left(\sum_{j=1}^M x_j = 0 \right) \quad (2.3)$$

I étant l'ensemble des fonctions de E qui s'annulent aux points t_i pour $i = 1 \dots n$.
On supposera cette condition vérifiée, (ce qui implique que $N_j \cap I = \{0\}$ pour $j = 1 \dots M$).

La solution σ du problème (I) est appelée spline de lissage par inf-convolution .

D'après l'expression de l'inf-convolution (2.2), le problème (I) est équivalent au problème de minimisation (II) suivant :

$$\underset{x_j \in X_j, \quad j=1 \dots M}{\text{Minimiser}} \left(\sum_{j=1}^M \mathcal{L}_j(x_j) + \rho \sum_{i=1}^n \left(\sum_{j=1}^M x_j(t_i) - z_i \right)^2 \right) \quad (II)$$

dans le sens où si $[\sigma_1, \dots, \sigma_M]$ est solution de (II) alors $\sigma = \sigma_1 + \dots + \sigma_M$ est solution de (I).
Inversement si σ est solution de (I) alors tout M -uplet $[\sigma_1, \dots, \sigma_M]$ tel que $\sigma = \sigma_1 + \dots + \sigma_M$ et réalisant la borne inférieure dans la relation (2.2) est solution de (II).

Le problème (II) admet une solution unique si et seulement si :

$$\left(x_j \in N_j, \quad j = 1 \dots M \quad \text{et} \quad \sum_{j=1}^M x_j \in I \right) \Rightarrow (x_j = 0 \quad \text{pour} \quad j = 1 \dots M)$$

Supposons que cette condition soit vérifiée.

Soit $(\sigma_1, \dots, \sigma_M)$ la solution de (II). Il est clair que chaque σ_j est solution du problème de lissage suivant :

$$\underset{x_j \in X_j}{\text{Minimiser}} \left(\mathcal{L}_j(x_j) + \sum_{i=1}^n \left[x_j(t_i) - \left(z_i - \sum_{k \neq j} \sigma_k(t_i) \right) \right]^2 \right)$$

Autrement dit, si on note $Y_k = (\sigma_k(t_i))_i$, $k = 1 \dots M$ et $Y = (\sigma(t_i))_i = \sum_{k=1}^M Y_k$, alors on a :

$$\sigma_j = Spl_j \left(Z - \sum_{k \neq j} Y_k \right)$$

c'est à dire :

$$Y_j = S_j \left(Z - \sum_{k \neq j} Y_k \right) \quad j = 1, \dots, M$$

et

$$Y = \sum_{k=1}^M Y_k$$

On construit ainsi un système linéaire à $(n \times M)$ inconnues. Pour le résoudre on utilise une méthode itérative de Gauss-Seidel:

Itération $\nu = 0$ Tous les Y_k sont mis à 0 : $Y_k^{[0]} = 0$ pour $k = 1 \dots M$

.

.

.

Itération $\nu + 1$ Supposons $Y_1^{[\nu+1]} \dots Y_{k-1}^{[\nu+1]}$ déjà calculés. $Y_k^{[\nu+1]}$ est alors calculé par :

$$Y_k^{[\nu+1]} = S_j \left(Z - \sum_{j=1}^{k-1} Y_j^{[\nu+1]} - \sum_{j=k+1}^M Y_j^{[\nu]} \right)$$

Le vecteur résultant de l'itération $(\nu + 1)$ est alors :

$$Y^{[\nu+1]} = \sum_{k=1}^M Y_k^{[\nu+1]}$$

On arrête le processus dès qu'on a atteint une stabilisation satisfaisante, par exemple dès que :

$$\|Y^{[\nu+1]} - Y^{[\nu]}\| \leq \varepsilon$$

ε un réel fixé par l'utilisateur

2.3 Algorithme de Gauss-Seidel

Pour plus de détails voir [3] et [21].

On note E un espace euclidien de dimension finie nM , muni de la norme $\|\cdot\|$.

2.3.1 Résolution d'un système linéaire

Soit (S) le système linéaire :

$$Ax = b \quad (S)$$

où b est un vecteur de E et A une matrice carrée définie par blocs de la forme :

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1M} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ A_{M1} & \cdots & A_{MM} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_M \end{pmatrix} \quad \text{et} \quad x = \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_M \end{pmatrix}$$

Chaque (A_{ij}) est une matrice carrée $(n \times n)$.

L'algorithme de Gauss-Seidel pour la résolution du système (S) fait partie d'une classe d'algorithmes qui consistent à ramener cette résolution à celle d'un problème de point fixe de la forme :

$$x = Bx + c \quad (S')$$

où c est un vecteur de E et B une matrice $(nM \times nM)$, tels que $(I - B)$ soit inversible et que la solution unique de (S') soit solution de (S) .

La forme de (S') suggère l'utilisation d'une méthode itérative pour sa résolution : On se donne x^0 un vecteur initial arbitraire, et on définit la suite de vecteur $(x^\nu)_\nu$ par la relation :

$$x^{\nu+1} = Bx^\nu + c \quad (2.4)$$

On dira alors que l'algorithme est convergent si et seulement si : $\lim_{\nu \rightarrow +\infty} x^\nu = x$, x étant la solution de (S) .

Dans le cas de l'algorithme de Gauss-Seidel on pose :

$$B = (D - E)^{-1}F \quad \text{et} \quad c = (D - E)^{-1}b$$

où D, E et F sont définis par :

$$D = \begin{pmatrix} A_{11} & 0 & \cdots & 0 \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & 0 \\ 0 & \cdots & 0 & A_{MM} \end{pmatrix}, \quad E = - \begin{pmatrix} 0 & 0 & \cdots & 0 \\ A_{21} & 0 & \cdots & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & 0 \\ A_{M1} & \cdots & A_{M-1M} & 0 \end{pmatrix}$$

$$\text{et } F = - \begin{pmatrix} 0 & A_{21} & \cdots & A_{M1} \\ 0 & 0 & \cdots & \cdot \\ \cdot & 0 & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & A_{MM-1} \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

En effet on a :

$$\begin{aligned} Ax = b &\Leftrightarrow (D - E - F)x = b \\ &\Leftrightarrow (D - E)x = Fx + b \\ &\Leftrightarrow x = (D - E)^{-1}Fx + (D - E)^{-1}b \end{aligned}$$

On suppose ici que $(D - E)$ est inversible, ce qui revient à supposer que les sous matrices diagonales $(A_{jj})_j$ de A sont inversibles.

L'itération (2.4) s'écrit alors : $x^{\nu+1} = (D - E)^{-1}Fx^{\nu} + (D - E)^{-1}b$.

C'est à dire :

$$x_j^{\nu+1} = A_{jj}^{-1} \left(b_j - \sum_{k=1}^{j-1} A_{kj} x_k^{\nu+1} - \sum_{k=j+1}^M A_{kj} x_k^{\nu} \right)$$

Théorème : *Condition suffisante de convergence*

Si la matrice A est symétrique définie positive alors l'algorithme de Gauss-Seidel est convergent. Pour la démonstration voir [3], [21].

2.3.2 Minimisation d'une fonctionnelle quadratique

Soit Φ une fonctionnelle convexe quadratique définie sur E de la forme :

$$\Phi(x) = \langle x, Ax \rangle - 2 \langle x, b \rangle$$

où b est un vecteur de E et A une matrice symétrique définie positive.

On suppose que E s'écrit sous la forme :

$$E = V_1 \oplus \cdots \oplus V_M$$

Chaque V_j étant un sous-espace de E de dimension n .

Comme dans le paragraphe précédent on écrira A sous forme de matrice blocs et b et x sous forme vecteurs blocs de taille n :

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ A_{M1} & \cdots & A_{MM} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_M \end{pmatrix} \quad \text{et} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_M \end{pmatrix}$$

Φ admet un minimum unique x , qui est solution du système :

$$Ax = b$$

Pour calculer x , on va utiliser l'algorithme de Gauss-Seidel :

x^0 un vecteur initial arbitraire.

$$\text{Pour } j = 1 \cdots M, \quad x_j^{\nu+1} = A_{jj}^{-1} \left(b_j - \sum_{k=1}^{j-1} A_{kj} x_k^{\nu+1} - \sum_{k=j+1}^M A_{kj} x_k^{\nu} \right)$$

D'autre part si on note :

$$y_0^{\nu+1} = (x_1^{\nu}, \dots, x_M^{\nu})$$

et pour $j = 1 \cdots M$:

$$y_j^{\nu+1} = (x_1^{\nu+1}, \dots, x_j^{\nu+1}, x_{j+1}^{\nu}, \dots, x_M^{\nu})$$

on a :

$$y_j^{\nu+1} = y_{j-1}^{\nu+1} + \bar{v}_j$$

où :

$$\bar{v}_j = (0, \dots, x_j^{\nu+1} - x_j^{\nu}, \dots, 0)$$

En outre, on montre aisément que \bar{v}_j est l'unique élément de V_j qui réalise :

$$\Phi(y_{j-1}^{\nu+1} + \bar{v}_j) = \min_{v_j \in V_j} \Phi(y_{j-1}^{\nu+1} + v_j)$$

$y_j^{\nu+1}$ est donc l'unique élément de E qui réalise le minimum de Φ dans la direction V_j en partant de $y_{j-1}^{\nu+1}$.

L'algorithme de Gauss-Seidel pour la minimisation de Φ s'écrit alors :

x^0 un vecteur initial arbitraire.

Supposons que $x^\nu = (x_1^\nu, \dots, x_M^\nu)$ est le vecteur à l'itération ν .

On calcule $x^{\nu+1}$ de la façon suivante :

On pose :

$$y_0^{\nu+1} = (x_1^\nu, \dots, x_M^\nu)$$

et pour $j = 1 \dots M$ on calcule :

$$y_j^{\nu+1} = y_{j-1}^{\nu+1} + \bar{v}_j$$

où \bar{v}_j est tel que :

$$\bar{\Phi}(y_{j-1}^{\nu+1} + \bar{v}_j) = \min_{v_j \in V_j} \Phi(y_{j-1}^{\nu+1} + v_j)$$

Le vecteur de l'itération $\nu + 1$ est alors :

$$x^{\nu+1} = y_M^{\nu+1}$$

Première Partie

**Un algorithme du type Gauss-Seidel
pour des fonctionnelles symétriques
semi-définies positives**

Deux applications

Chapitre 3

Un algorithme de type Gauss-Seidel pour des fonctionnelles symétriques semi-définies positives

3.1 Introduction

Soit Φ une fonction convexe quadratique définie sur un espace Euclidien E à valeurs réelles, de la forme :

$$\Phi(y) = \langle y, Ay \rangle - 2 \langle b, y \rangle,$$

où $b \in E$ et A est une matrice symétrique semi-définie positive. On notera :

$$\mathcal{N} = \{y/Ay = 0\} = \{y/\langle y, Ay \rangle = 0\} \quad \text{et} \quad F = \mathcal{N}^\perp.$$

On cherche à minimiser la fonction Φ sur E . Pour qu'un tel minimum existe, il faut supposer que : $b \in F$. Il existe alors, un et un seul $\bar{y} \in F$ tel que :

$$\Phi(\bar{y}) = \min_{y \in E} \Phi(y).$$

Il vérifie $A\bar{y} = b$, et l'ensemble des éléments de E qui réalisent le minimum de Φ sur E est $\bar{y} + \mathcal{N}$.

Dans le cas présent on ne peut pas utiliser directement l'algorithme de Gauss-Seidel pour le calcul de \bar{y} , car Φ n'est pas définie positive. On propose alors de modifier cet algorithme par l'introduction d'une projection sur F après chaque itération. Dans la suite on décrit ce nouvel algorithme et on démontre sa convergence pour le problème de minimisation d'une fonctionnelle quadratique semi-définie positive.

3.2 Algorithme pour le calcul de \bar{y}

On se donne M sous-espaces vectoriels de E , $V_1 \cdots V_M$, qui forment une décomposition en somme directe de E :

$$E = V_1 \oplus V_2 \oplus \cdots \oplus V_M$$

et qui soient tels que : $V_j \cap \mathcal{N} = \{0\}$ pour $j = 1 \cdots M$.

Le calcul de l'itéré $y^{\nu+1}$ à partir de l'itéré y^ν se décompose en deux étapes :
La première étape consiste à calculer $\bar{y}^{\nu+1}$ par une itération classique de l'algorithme de

Gauss-Seidel, qui consiste à minimiser successivement Φ dans la direction des V_j .

La deuxième étape consiste à projeter $\bar{y}^{\nu+1}$ sur F . On appellera cela l'étape de recherche d'un itéré de norme minimale. Le vecteur résultant $y^{\nu+1}$ est alors $y^{\nu+1} = P\bar{y}^{\nu+1}$, où P est la projection orthogonale sur F . Autrement dit :

Si $y^\nu \in F$ est la valeur obtenue après l'itération ν . On pose $y_0^\nu = y^\nu$, et on calcule successivement y_1^ν, \dots, y_M^ν de la façon suivante:

Pour $j = 1 \dots M$, soit \bar{v}_j tel que :

$$\Phi(y_{j-1}^\nu + \bar{v}_j) = \min_{v_j \in V_j} \Phi(y_{j-1}^\nu + v_j).$$

Comme $V_j \cap \mathcal{N} = \{0\}$, un tel élément est unique. On pose alors :

$$y_j^\nu = y_{j-1}^\nu + \bar{v}_j$$

et $y^{\nu+1}$ est obtenu par :

$$y^{\nu+1} = P(y_M^\nu),$$

où P désigne l'opérateur de projection orthogonale sur F .

Dans le paragraphe suivant nous démontrons que la suite $(y^\nu)_\nu$ converge et a pour limite \bar{y} .

Pour faciliter les démonstrations nous adopterons les notations suivantes :

Si V est un sous-espace vectoriel de E tel que $V \cap \mathcal{N} = \{0\}$, on note \mathcal{M}_V l'opérateur qui à $y \in E$ associe l'unique point qui en partant de y réalise le minimum de Φ dans la direction V , c'est à dire :

$$\mathcal{M}_V(y) = y + \bar{v},$$

où $\bar{v} \in V$ vérifie :

$$\Phi(y + \bar{v}) = \min_{v \in V} \Phi(y + v).$$

Avec ces notations une itération de l'algorithme s'écrit :

$$y^{\nu+1} = P\mathcal{M}_{V_M} \dots \mathcal{M}_{V_2} \mathcal{M}_{V_1} (y^\nu).$$

On notera aussi W l'image de V par P , i.e. $W = P(V)$.

3.3 Convergence de l'algorithme

La démonstration de la convergence de l'algorithme repose sur trois lemmes.

Lemme 1 : Pour $y \in F$, on a :

$$P\mathcal{M}_V(y) = \mathcal{M}_W(y).$$

Preuve :

Pour un y quelconque dans F , on a :

$$\mathcal{M}_V(y) = y + \bar{v} \text{ avec } \bar{v} \text{ tel que } \forall v \in V, \quad \Phi(y + \bar{v}) \leq \Phi(y + v)$$

donc : $P\mathcal{M}_V(y) = y + \bar{w}$, où $\bar{w} = P(\bar{v}) \in W$ ($\bar{v} - \bar{w} \in \mathcal{N}$).

De plus, pour tout $w \in W$, il existe $v \in V$ tel que $P(v) = w$, ($v - w \in \mathcal{N}$) et Φ est constante sur l'ensemble $y + \mathcal{N}$. On a alors :

$$\Phi(y + \bar{w}) = \Phi(y + \bar{v}) \leq \Phi(y + v) = \Phi(y + w),$$

ce qui montre que :

$$y + \bar{w} = \mathcal{M}_W(y),$$

d'où le résultat du lemme : $P\mathcal{M}_V(y) = \mathcal{M}_W(y)$ □

Lemme 2 : Pour tout $y \in E$ on a :

$$P\mathcal{M}_V P(y) = P\mathcal{M}_V(y).$$

Preuve :

Montrons d'abord que $P\mathcal{M}_V(y) = \mathcal{M}_W P(y)$.

Par définition de \mathcal{M}_V on a $\mathcal{M}_V(y) = y + \bar{v}$, où \bar{v} est tel que :

$$\Phi(y + \bar{v}) \leq \Phi(y + v) \quad \forall v \in V$$

et donc :

$$P\mathcal{M}_V(y) = \tilde{y} + \bar{w}, \text{ où } \bar{w} = P(\bar{v}) \in W \text{ et } \tilde{y} = P(y) \in F.$$

Notons bien que $y - \tilde{y} \in \mathcal{N}$ et $\bar{v} - \bar{w} \in \mathcal{N}$. D'autre part pour tout $w \in W$, il existe $v \in V$ tel que $P(v) = w$, donc pour tout $w \in W$ on a :

$$\Phi(\tilde{y} + \bar{w}) = \Phi(y + \bar{v}) \leq \Phi(y + v) = \Phi(\tilde{y} + w),$$

c'est à dire : $\tilde{y} + \bar{w} = \mathcal{M}_W(\tilde{y}) = \mathcal{M}_W P(y)$ et donc $P\mathcal{M}_V(y) = \mathcal{M}_W P(y)$.

Enfin, d'après le lemme 1, comme $P(y) \in F$, on a :

$$\mathcal{M}_W P(y) = P\mathcal{M}_V P(y),$$

d'où le résultat. □

Lemme 3 : La fonction Φ est fortement convexe sur F , c'est à dire qu'il existe $\delta > 0$ tel que pour tout $u \in F$ et tout $v \in F$ on ait :

$$\Phi\left(\frac{u+v}{2}\right) \leq \frac{\Phi(u) + \Phi(v)}{2} - \delta\|u-v\|^2$$

Preuve

Ceci provient du fait qu'il existe $m > 0$ tel que :

$$\forall u \in F \quad m\|u\|^2 \leq \langle Au, u \rangle$$

et que l'on a l'inégalité du parallélogramme :

$$\left\|\frac{u+v}{2}\right\|^2 \leq \frac{\|u\|^2 + \|v\|^2}{2} - \frac{1}{4}\|u-v\|^2.$$

On prend $\delta = \frac{m}{4}$ □

Théorème : La suite y^ν converge et a pour limite \bar{y} .

Preuve :

Rappelons qu'une itération de l'algorithme s'écrit :

$$y^{\nu+1} = P\mathcal{M}_{V_M} \cdots \mathcal{M}_{V_2} \mathcal{M}_{V_1} y^\nu \tag{3.5}$$

On note $W_j = P(V_j)$. D'après le lemme 2 puis le lemme 1 on a :

$$P\mathcal{M}_{V_M} = P\mathcal{M}_{V_M} P = \mathcal{M}_{W_M} P,$$

donc :

$$y^{\nu+1} = \mathcal{M}_{W_M} P\mathcal{M}_{V_{M-1}} \cdots \mathcal{M}_{V_2} \mathcal{M}_{V_1} y^\nu$$

et en continuant par le même procédé, en opérant de proche en proche dans (3.5), on obtient :

$$y^{\nu+1} = \mathcal{M}_{W_M} \mathcal{M}_{W_{M-1}} \cdots \mathcal{M}_{W_1} P (y^\nu).$$

Or, d'après la construction de l'algorithme, $y^\nu \in F$, donc $P(y^\nu) = y^\nu$. L'itération de l'algorithme peut, alors, s'écrire :

$$y^{\nu+1} = \mathcal{M}_{W_M} \mathcal{M}_{W_{M-1}} \cdots \mathcal{M}_{W_1} (y^\nu).$$

Pour la suite de la démonstration on notera :

$$\begin{aligned} \bar{y}_0^\nu &= y^\nu \\ \bar{y}_1^\nu &= \mathcal{M}_{W_1} \bar{y}_0^\nu \\ &\vdots \\ &\vdots \\ \bar{y}_j^\nu &= \mathcal{M}_{W_j} \bar{y}_{j-1}^\nu \\ &\vdots \\ &\vdots \\ \bar{y}_M^\nu &= \mathcal{M}_{W_M} \bar{y}_{M-1}^\nu \\ \bar{y}^{\nu+1} &= \bar{y}_M^\nu \end{aligned}$$

Comme \bar{y}_j^ν réalise le minimum de Φ dans la direction W_j à partir de \bar{y}_{j-1}^ν , en posant $\bar{y}_j^\nu = \bar{y}_{j-1}^\nu + \bar{w}_j$, on aura :

$$\begin{aligned} \Phi(\bar{y}_j^\nu) &\leq \Phi\left(\bar{y}_{j-1}^\nu + \frac{\bar{w}_j}{2}\right) \\ &\leq \Phi\left(\frac{\bar{y}_{j-1}^\nu + \bar{y}_j^\nu}{2}\right) \end{aligned}$$

et en appliquant le lemme 3 :

$$\begin{aligned} \Phi(\bar{y}_j^\nu) &\leq \frac{\Phi(\bar{y}_{j-1}^\nu) + \Phi(\bar{y}_j^\nu)}{2} - \frac{\delta}{2} \|\bar{y}_{j-1}^\nu - \bar{y}_j^\nu\|^2 \\ &\leq \Phi(\bar{y}_{j-1}^\nu) - \delta \|\bar{y}_{j-1}^\nu - \bar{y}_j^\nu\|^2 \end{aligned}$$

Finalement :

$$\|\bar{y}_{j-1}^\nu - \bar{y}_j^\nu\|^2 \leq \frac{1}{\delta} (\Phi(\bar{y}_{j-1}^\nu) - \Phi(\bar{y}_j^\nu)).$$

Or par définition de l'algorithme on a :

$$\Phi(\bar{y}^\nu) = \Phi(\bar{y}_0^\nu) \geq \Phi(\bar{y}_1^\nu) \geq \cdots \geq \Phi(\bar{y}_M^\nu) = \Phi(\bar{y}^{\nu+1})$$

et comme cette suite, décroissante, est minorée par $\Phi(\bar{y})$, elle converge vers un $\beta \geq \Phi(\bar{y})$.

On en déduit donc que :

$$\lim_{\nu \rightarrow +\infty} \|\bar{y}_{j-1}^\nu - \bar{y}_j^\nu\| = 0, \quad \forall j = 1 \cdots M. \quad (3.6)$$

D'autre part toute la suite $(y^\nu)_\nu$ se trouve dans le compact : $\{y \in F / \Phi(y) \leq \Phi(y^0)\}$. Il existe donc une sous-suite $(y^\nu)_{\nu \in N_1}$, ($N_1 \subset \mathbb{N}$) qui converge. Notons \tilde{y} sa limite. On remarque, d'après (3.6), que :

$$\lim_{\nu \in N_1} \bar{y}_j^\nu = \tilde{y} \quad \text{pour } j = 1 \dots M.$$

Montrons que \tilde{y} est solution du problème.

Pour cela, supposons que \tilde{y} ne soit pas solution et montrons que cela nous conduit à une contradiction.

En effet il existerait un indice j tel que \tilde{y} ne soit pas minimum dans la direction W_j , donc :

$$\exists \tilde{w}_j \in W_j \quad / \quad \Phi(\tilde{y}) > \Phi(\tilde{y} + \tilde{w}_j) \quad (3.7)$$

Or, comme \bar{y}_j^ν est obtenu à partir de \bar{y}_{j-1}^ν par minimisation de Φ dans la direction W_j , on a :

$$\Phi(\bar{y}_j^\nu) \leq \Phi(\bar{y}_{j-1}^\nu + \tilde{w}_j),$$

et en passant à la limite pour $\nu \in N_1$:

$$\Phi(\tilde{y}) \leq \Phi(\tilde{y} + \tilde{w}_j),$$

d'où la contradiction avec (3.7).

Donc \tilde{y} est bien solution du problème, i.e. $\tilde{y} = \bar{y}$.

Il reste maintenant à démontrer que toute sous suite extraite de $(y^\nu)_\nu$ converge vers \bar{y} .

Si ce n'était pas le cas on pourrait former une sous suite $(y^\nu)_{\nu \in \tilde{N}}$ telle que :

$$\|y^\nu - \bar{y}\| \geq \delta > 0, \quad \text{pour tout } \nu \in \tilde{N} \subset \mathbb{N}.$$

Mais en appliquant le raisonnement ci-dessus à cette sous-suite, on pourrait en extraire une sous suite $(y^\nu)_{\nu \in N_1}$ avec $N_1 \subset \tilde{N}$ dont la limite ne pourrait être que \bar{y} . \square

3.4 Applications

On se situe dans les conditions habituelles d'un problème de lissage à deux dimensions. On suppose donné :

- un nuage de points T dans le plan \mathbb{R}^2 : $T = (t_i^1, t_i^2)_{i=1 \dots n}$
- un vecteur $Z = (Z_i)_{i=1 \dots n}$ de données bruitées correspondant aux valeurs prises par une fonction inconnue f_e aux points du nuage T

$$Z_i = f_e(t_i^1, t_i^2) + \varepsilon_i$$

avec ε_i un bruit aléatoire de loi gaussienne centrée et de variance inconnue. On se pose alors le problème de trouver une fonction spline σ lissant les données Z .

Les méthodes classiques permettant de résoudre ce problème consistent à minimiser une quantité de la forme :

$$L(x) + \rho \sum_{i=1}^n (x(t_i^1, t_i^2) - Z_i)^2$$

où L est un critère de lissage à deux dimensions . Par exemple si on prend :

$$L(x) = \iint_{\mathbb{R}^2} \left[\left| \frac{\partial^2 x}{\partial t_1^2} \right|^2 + 2 \left| \frac{\partial^2 x}{\partial t_1 \partial t_2} \right|^2 + \left| \frac{\partial^2 x}{\partial t_2^2} \right|^2 \right] \partial t_1 \partial t_2$$

on obtient les splines de lissage du type plaque mince . La difficulté principale de ces méthodes est leur coût, car elles nécessitent la résolution d'un système linéaire de taille $(n + 3) \times (n + 3)$.

Pour remédier à ce problème nous proposons, dans les deux chapitres suivants, de représenter la solution comme somme de M fonctions σ_j . Chaque σ_j est une spline de lissage relativement à un critère de lissage L_j et aux données :

$$\left(Z_i - \sum_{k=1, k \neq j}^M \sigma_k(t_i) \right)_{i=1 \dots n}$$

c'est à dire :

$$Y_j = S_j \left(Z - \sum_{k=1, k \neq j}^M Y_k \right)$$

où Y_j et S_j , $j = 1, \dots, M$, sont définis au chapitre 2.

Chaque σ_j est obtenue par résolution d'un problème de lissage à une dimension.

La somme σ des σ_j , peut être considérée comme une spline de lissage relatif à un critère de lissage L obtenu par inf-convolution des M critères L_j .

Dans le chapitre 5, chaque σ_j est une fonction de deux variables, de révolution autour d'un point ω_j qu'on appellera focale :

$$\sigma_j(x, y) = f_j \left(\sqrt{(x - \omega_j^1)^2 + (y - \omega_j^2)^2} \right)$$

On dira alors que σ est une spline de lissage multifocale.

Dans le chapitre 6, chaque σ_j est une fonction de deux variables, constante perpendiculairement à une direction faisant un angle θ_j avec l'axe des x :

$$\sigma_j(x, y) = f_j(x \cos(\theta_j) + y \sin(\theta_j))$$

On dira alors que σ est une spline de lissage multidirectionnelle.

Dans les deux cas la recherche de la solution se ramène à la résolution d'un problème de minimisation dans \mathbb{R}^{nM} , d'une fonctionnelle quadratique semi-définie positive. Pour la résolution de ce problème on utilisera l'algorithme de Gauss-Seidel modifié qu'on a introduit précédemment.

Les deux modèles de lissage que nous proposons sont basés sur l'idée de remplacer la résolution d'un problème de lissage bidimensionnel par la résolution d'une suite de problèmes de lissage unidimensionnels faciles à résoudre.

Chapitre 4

Lissage par splines Multifocales

Dans ce chapitre, nous proposons de représenter la solution σ comme somme de M fonctions σ_j , où chaque σ_j est de révolution autour d'une focale ω_j .

Pour la suite on suppose que $n \geq 1$ et que parmi les points du nuage T il y en a au moins deux qui ne soient pas à même distance de l'un des ω_j , ceci pour $j = 1 \dots M$.

4.1 Splines monofocales

Dans le plan \mathbb{R}^2 , considérons un point $\omega_1 = (\omega_1^1, \omega_1^2)$, qu'on appellera focale, et supposons X_1 un sous-espace semi-hilbertien de $E_1 = \mathbb{R}^{\mathbb{R}^+}$, avec J_1 la fonctionnelle semi-hilbertienne associée. A chaque fonction φ de X_1 on associe une fonction f de $E_2 = (\mathbb{R})^{\mathbb{R}^2}$ par :

$$f(t^1, t^2) = \varphi \left(\sqrt{(t^1 - \omega_1^1)^2 + (t^2 - \omega_1^2)^2} \right) \quad (4.8)$$

On définit sur E_2 la fonctionnelle semi-hilbertienne L_1 par :

$$L_1(f) = \begin{cases} J_1(\varphi) & \text{si } f(t^1, t^2) = \varphi \left(\sqrt{(t^1 - \omega_1^1)^2 + (t^2 - \omega_1^2)^2} \right) \text{ et } \varphi \in X_1 \\ +\infty & \text{sinon} \end{cases}$$

On définit ainsi une spline bidimensionnelle monofocale comme solution du problème :

$$\underset{f \in E_2}{\text{Minimiser}} \left(L_1(f) + \rho \sum_{i=1}^n (f(t_i^1, t_i^2) - Z_i)^2 \right) \quad (P_2)$$

La solution de ce problème peut être déduite de la solution du problème de lissage unidimensionnel :

$$\underset{\varphi \in X_1}{\text{Minimiser}} \left(J_1(\varphi) + \rho \sum_{i=1}^n (\varphi(r_i) - Z_i)^2 \right) \quad (P_1)$$

avec : $r_i = \sqrt{(t^1 - \omega_1^1)^2 + (t^2 - \omega_1^2)^2}$ pour $i = 1, \dots, n$.

4.1.1 Choix du critère de lissage J_1 :

La première idée pour le choix du critère de lissage est de prendre le critère classique :

$$J_1(\varphi) = \int_{r>0} (\varphi''(r))^2 dr$$

La dérivée première en 0 de la solution obtenue n'est pas forcément nulle, ce qui entraîne que la fonction f ne sera pas dérivable en ω_1 . On peut imposer cette contrainte en utilisant une méthode de pénalité, et en considérant le critère :

$$J_1(\varphi) = \int_{r>0} (\varphi''(r))^2 dr + \nu (\varphi'(0))^2$$

avec ν un réel assez grand.

Notre idée est d'utiliser un critère qui conduit naturellement à cette contrainte. On part, par exemple, du critère bidimensionnel plaque mince :

$$L_1(f) = \iint_{\mathbb{R}^2} \left[\left| \frac{\partial^2 f}{\partial t_1^2} \right|^2 + 2 \left| \frac{\partial^2 f}{\partial t_1 \partial t_2} \right|^2 + \left| \frac{\partial^2 f}{\partial t_2^2} \right|^2 \right] \partial t_1 \partial t_2$$

En remplaçant f par son écriture (4.8) en fonction de φ , on obtient :

$$J_1(\varphi) = \int_{r>0} 2\pi r \left(\varphi(r)''^2 + \left(\frac{\varphi'(r)}{r} \right)^2 \right) dr$$

On remarque bien que cette expression n'a de sens que si $\varphi'(r) \rightarrow 0$ quand $r \rightarrow 0$. On utilisera ce critère pour toute la suite.

4.1.2 Problème à une dimension :

Le problème (P_1) admet une solution unique si et seulement si $N_1 \cap I_1 = \{0\}$, où N_1 est le noyau de J_1 et I_1 l'ensemble des fonctions de E_1 qui s'annulent sur les $(r_i)_{i=1, \dots, n}$.

Cette condition est vérifiée, en effet :

$$N_1 = \{\text{fonctions constantes de } E_1\}$$

et donc :

$$N_1 \cap I_1 = \{0\}$$

Donc le problème admet une solution unique.

Du fait que la suite des $(r_i)_{i=1,\dots,n}$ n'est pas ordonnée et que sa mise en ordre est coûteuse, $[n \log n]$, on propose d'écrire la spline solution de (P_1) dans une base de B-splines de degré 2 sur une grille régulière (u_0, \dots, u_{N_s}) avec $u_0 = 0$ et $u_{N_s} = r_{Max} = \underset{i=1,\dots,n}{Max}(r_i)$.

La solution φ s'écrira alors :

$$\varphi(r) = \sum_{j=-2}^{N_s-1} \alpha_j B_j(r)$$

La contrainte $\varphi'(0) = 0$ entraîne que $\alpha_{-2} = \alpha_{-1}$. Donc si φ est solution de (P_1) elle sera de la forme :

$$\varphi(r) = \alpha_{-1} [B_{-2}(r) + B_{-1}(r)] + \sum_{j=0}^{N_s-1} \alpha_j B_j(r)$$

On construit ainsi une nouvelle base $(C_j)_{j=-1,\dots,N_s-1}$ par :

$$\begin{cases} C_{-1} = B_{-2} + B_{-1} \\ C_j = B_j \end{cases} \quad \text{pour } j = 0, \dots, N_s - 1$$

et on a :

$$\varphi(r) = \sum_{j=-1}^{N_s-1} \alpha_j C_j(r) \tag{4.9}$$

On notera ici qu'on prend X_1 l'espace des fonctions φ qui s'écrivent sous la forme (4.9). L'opérateur J_1 est alors définit par:

$$J_1(\varphi) = \begin{cases} \int_{r>0} 2\pi r \left(\varphi(r)''^2 + \left(\frac{\varphi'(r)}{r} \right)^2 \right) dr & \text{si } \varphi \in X_1 \\ 0 & \text{sinon} \end{cases}$$

En remplaçant φ dans (P_1) par son écriture (4.9), on se ramène à un problème de minimisation d'une fonctionnelle quadratique dans \mathbb{R}^{N_s+1} et en écrivant les conditions d'optimalité, nullité des dérivées partielles premières au minimum $(\alpha_{-1}, \dots, \alpha_{N_s-1})$, on obtient un système linéaire $(N_s + 1) \times (N_s + 1)$ de la forme :

$$M\alpha = D \tag{S}$$

$M = (M_{j,k})_{j,k}$ la matrice du système est symétrique à 5 diagonales.

La résolution du système linéaire (S) ne pose aucun problème : on utilise la méthode de Choleski.

On calcule ainsi la fonction φ solution de (P_1) . Enfin la solution σ_1 de (P_2) s'en déduit par (4.8).

Pour la suite, notons Spl_1 l'application linéaire qui à chaque vecteur Z de \mathbb{R}^n associe la solution σ_1 du problème de lissage bidimensionnel (P_2) pour les données Z :

$$Spl_1 : \mathbb{R}^n \rightarrow E_2$$

et notons S_1 la matrice de lissage correspondante.

$$S_1 : Z \in \mathbb{R}^n \rightarrow Y \in \mathbb{R}^n$$

tel que $Y_i = \sigma_1(t_i^1, t_i^2)$ pour $i = 1, \dots, n$ et $\sigma_1 = Spl_1(Z)$.

4.2 Splines Multifocales

Supposons qu'on ait M focales distinctes ω_j , $j = 1, \dots, M$, dans le plan \mathbb{R}^2 , et M fonctionnelles semi-hilbertiennes J_j sur E_1 de domaine X_j .

De la même façon que précédemment, on associe à chaque J_j la fonctionnelle semi-hilbertienne L_j définie sur E_2 . Notons par $Spl_j(Z)$ la spline de lissage correspondante à L_j pour les données Z , et par $Y_j = S_j(Z)$ le vecteur résultant.

En utilisant l'opération d'inf-convolution, on construit, à partir des L_j une fonctionnelle L :

$$L = \bigvee_{j=1}^M L_j$$

Soit $X = X_1 \oplus \dots \oplus X_M$ le domaine de L et $N = N_1 \oplus \dots \oplus N_M$ son noyau. Pour $f \in X$, notons X_f l'ensemble :

$$X_f = \left\{ (f_1, \dots, f_M) \text{ tel que } f_j \in X_j \text{ pour } j = 1, \dots, M \text{ et } \left(\sum_{j=1}^M f_j \right) = f \right\}$$

on aura, alors:

$$L(f) = \underset{(f_j)_{j=1 \dots M} \in X_f}{\text{minimum}} \left[\sum_{j=1}^M L_j(f_j) \right] \quad (4.10)$$

D'après la théorie générale de l'inf-convolution, L est une fonctionnelle semi-hilbertienne; son noyau N est la somme algébrique des noyaux N_j des L_j dans E_2 :

$$N = \sum_{j=1}^M N_j$$

On considère alors le problème de lissage bidimensionnel suivant :

$$\underset{f \in E_2}{\text{Minimiser}} \left(L(f) + \rho \sum_{i=1}^n (f(t_i^1, t_i^2) - Z_i)^2 \right) \quad (I)$$

Ce problème admet une solution unique si et seulement si :

$$\left(f_j \in N_j, \quad j = 1 \dots M \text{ et } \sum_{j=1}^M f_j \in I_2 \right) \Rightarrow \left(\sum_{j=1}^M f_j = 0 \right) \quad (4.11)$$

I_2 étant l'ensemble des fonctions de E_2 qui s'annulent aux points $t_i = (t_i^1, t_i^2)$ pour $i = 1, \dots, n$.

Cette condition est vérifiée, en effet : Pour $j = 1, \dots, M$, N_j est l'ensemble des fonctions f_j constantes, donc $f = f_1 + \dots + f_M$ est aussi une constante. Si en plus f s'annule aux points $(t_i)_i$, alors elle est identiquement nulle.

Le problème (I) admet donc, une solution unique σ qui sera appelée spline de lissage multifocale.

Comme on l'a vu précédemment, le problème (I) est équivalent, dans un certain sens, au problème (II) suivant :

$$\underset{f_j \in E_2, j = 1, \dots, M}{\text{Minimiser}} \left(\sum_{j=1}^M L_j(f_j) + \sum_{i=1}^n \left(\sum_{j=1}^M f_j(t_i^1, t_i^2) - Z_i \right) \right) \quad (II)$$

Ce problème admet une solution unique si et seulement si :

$$\left\{ \text{pour } j = 1, \dots, M, f_j \in N_j \text{ et } \sum_{j=1}^M f_j \in I_2 \right\} \Rightarrow \{f_j = 0 \text{ pour } j = 1, \dots, M\}$$

Cette proposition n'est pas vérifiée dans le cas présent. Il suffit de prendre $f_j = c_j$ pour $j = 1, \dots, M$ où les c_j sont des constantes à somme nulle : $c_1 + \dots + c_M = 0$.

4.3 Algorithme de Gauss-Seidel modifié

4.3.1 Hypothèses de l'algorithme

Dans cette section on montre que le problème (II) vérifie bien les hypothèses de l'algorithme de Gauss-Seidel modifié, c'est à dire qu'on peut le ramener à un problème de minimisation d'une fonctionnelle convexe quadratique semi-définie positive.

Pour chaque fonctionnelle L_j on peut écrire :

$$\begin{aligned} \text{Inf}_{\substack{f_j \in E \\ f_j(t_i) = Y_j^i}} L_j(f_j) &= \langle Y_j, \Omega_j Y_j \rangle_n \end{aligned}$$

où Ω_j une matrice symétrique semi-définie positive. Le problème (II) peut donc se transformer en un problème de minimisation sur $(\mathbb{R}^n)^M$ suivant :

$$\underset{Y_j \in \mathbb{R}^n, j=1, \dots, M}{\text{Minimiser}} \sum_{j=1}^M \langle Y_j, \Omega_j Y_j \rangle_n + \rho \left\| \sum_{j=1}^M Y_j - Z \right\|_n^2 \quad (P)$$

Si on note $E = (\mathbb{R}^n)^M$ et si \vec{Y} est le vecteur de E suivant :

$$\vec{Y} = (Y_1, \dots, Y_M)$$

on ramène (P) à un problème de minimisation sur E de la forme (P') :

$$\underset{\vec{Y} \in E}{\text{minimiser}} \langle \vec{Y}, A\vec{Y} \rangle_{nM} - 2 \langle \vec{Y}, b \rangle_{nM} \quad (P')$$

En effet, on a :

$$\rho \left\| \sum_{j=1}^M Y_j - Z \right\|_n^2 = \rho \left(\left\| \sum_{j=1}^M Y_j \right\|_n^2 - 2 \left\langle \sum_{j=1}^M Y_j, Z \right\rangle_n + \|Z\|_n^2 \right)$$

Le terme $\|Z\|_n^2$ étant constant, le problème (P) est équivalent à :

$$\underset{\vec{Y} \in E}{\text{minimiser}} \sum_{j=1}^M \left\langle Y_j, \left(\frac{1}{\rho} \Omega_j Y_j + \sum_{k=1}^M Y_k \right) \right\rangle_n - 2 \sum_{j=1}^M \langle Y_j, Z \rangle_n$$

et ce problème s'écrit bien sous la forme (P') avec :

$$A = \begin{pmatrix} \frac{1}{\rho}\Omega_1 + I & I & \cdots & I \\ I & \frac{1}{\rho}\Omega_2 + I & \cdots & I \\ \vdots & \ddots & & \vdots \\ I & \cdots & \frac{1}{\rho}\Omega_{M-1} + I & I \\ I & \cdots & I & \frac{1}{\rho}\Omega_M + I \end{pmatrix} \quad \text{et } b = \begin{pmatrix} Z \\ Z \\ \vdots \\ Z \\ Z \end{pmatrix}$$

La matrice A est symétrique. Elle est semi-définie positive. En effet, d'après leur définitions même, les matrices Ω_j sont semi-définies positives et l'on a pour tout Y :

$$\langle \vec{Y}, A\vec{Y} \rangle = \sum_{j=1}^M \langle \vec{Y}_j, \Omega_j \vec{Y}_j \rangle + \left\| \sum_{j=1}^M Y_j \right\|^2 \geq 0$$

Par contre, la matrice A n'est pas définie positive. Car, si l'on prend $Y_j = (\lambda_j, \dots, \lambda_j)$, $j = 1, \dots, M$ où les λ_j vérifient $\sum_{j=1}^M \lambda_j = 0$, on aura :

$$\Omega_j Y_j = 0, \quad j = 1, \dots, M \quad \text{et} \quad \sum_{j=1}^M Y_j = 0$$

donc :

$$\langle \vec{Y}, A\vec{Y} \rangle_{nM} = 0 \quad \text{avec } \vec{Y} \neq 0$$

On montre plus précisément que le noyau de A est :

$$\mathcal{N} = \left\{ (\delta_1, \dots, \delta_M) \in (\mathbb{R}^n)^M / \delta_j = \lambda_j \mathbb{1}, j = 1, \dots, M, \text{ avec } \sum_{j=1}^M \lambda_j = 0 \right\}$$

Il étant le vecteur de \mathbb{R}^n dont toutes les composantes sont égales à 1.

Le problème (P) vérifie donc les hypothèses de l'algorithme de Gauss-Seidel modifié. D'autre part, E s'écrit sous forme de somme directe :

$$E = V_1 \oplus \cdots \oplus V_M$$

où chaque V_j est l'espace :

$$V_j = \left\{ \vec{Y}_j = [Y_1, \dots, Y_M] \text{ tel que } Y_k = 0 \text{ pour } k \neq j \right\}$$

et on a bien : $V_j \cap \mathcal{N} = \{0\}$.

On peut appliquer l'algorithme de Gauss-Seidel modifié :

On se donne un vecteur initial $\vec{Y} = (0, \dots, 0)$, et on construit \vec{Y}^ν à partir de $\vec{Y}^{\nu-1}$ par une itération de l'algorithme classique de Gauss-Seidel, suivie de l'étape de recherche de l'itéré de norme minimale.

4.3.2 Itéré de norme minimale

A l'étape précédente on a calculé, par une itération de l'algorithme de Gauss-Seidel, le vecteur $\tilde{Y}^\nu = (\tilde{Y}_1^\nu, \dots, \tilde{Y}_M^\nu)$.

On calcule la projection sur F de \tilde{Y}^ν qu'on notera :

$$\vec{Y}^\nu = P_F \left(\tilde{Y}_1^\nu, \dots, \tilde{Y}_M^\nu \right) = (Y_1^\nu, \dots, Y_M^\nu)$$

P_F est la projection orthogonale sur $F = \mathcal{N}^\perp$. Il s'agit donc, de chercher l'élément $\vec{\delta}$ de \mathcal{N} tel que :

$$\left\| \tilde{Y}^\nu - \vec{\delta} \right\|_{nM} \text{ soit minimale.}$$

Comme $\sum_{j=1}^M \delta_j = 0$ pour $\delta_j = \lambda_j \mathbb{1}$ est équivalent à $\sum_{j=1}^M \lambda_j = 0$, cela revient à :

Trouver $\lambda_j, j = 1, \dots, M$, vérifiant $\sum_{j=1}^M \lambda_j = 0$ tel que :

$$\sum_{j=1}^M \left\| \tilde{Y}_j^\nu - \lambda_j \mathbb{1} \right\|^2 \text{ soit minimum}$$

Ce problème se résout aisément, en introduisant des multiplicateurs de Lagrange. La solution s'écrit alors :

$$\lambda_j = \Delta_j - \frac{1}{M} \sum_{k=1}^M \Delta_k$$

où $\Delta_j = \frac{1}{n} \sum_{i=1}^n \tilde{Y}_i^j$.

L'itéré de norme minimale est alors :

$$\vec{Y}^\nu = \left(\tilde{Y}_1^\nu - \lambda_1 \mathbb{1}, \dots, \tilde{Y}_M^\nu - \lambda_M \mathbb{1} \right)$$

c'est à dire :

$$Y_j^\nu = \tilde{Y}_j^\nu - \lambda_j \mathbb{1}, \quad j = 1, \dots, M$$

4.3.3 Algorithme

L'algorithme pour le calcul du vecteur Y résultant d'un lissage multifocale se résume de la façon suivante :

Itération 0 :

Tous les Y_j sont mis à 0 : $Y_j^0 = 0$.

⋮

Itération ν :

$Y_1^{\nu-1} \dots Y_M^{\nu-1}$ déjà calculés.

1- On calcule $\tilde{Y}_1^\nu \dots \tilde{Y}_M^\nu$ par :

$$\text{pour } j = 1, \dots, M, \quad \tilde{Y}_j^\nu = S_j \left(Z - \sum_{k=1}^{j-1} \tilde{Y}_k^\nu - \sum_{k=j+1}^M Y_k^{\nu-1} \right)$$

2- On calcule les vecteurs de norme minimale :

$$Y_1^\nu \dots Y_M^\nu$$

par :

$$Y_j^\nu = \tilde{Y}_j^\nu - \lambda_j \mathbb{1}$$

On arrête le processus dès qu'on a atteint une stabilisation satisfaisante, par exemple dès que:

$$\|Y^\nu - Y^{\nu-1}\| \leq \varepsilon$$

ou encore:

$$\frac{\|Y^\nu - Y^{\nu-1}\|}{\|Y^\nu\|} \leq \varepsilon$$

ε étant une précision fixée par l'utilisateur.

Le vecteur solution du problème de lissage multifocale est alors :

$$Y = \sum_{j=1}^M Y_j^\nu$$

4.4 Expérimentations numériques

Dans ce paragraphe on présente quelques résultats des expérimentations numériques effectuées avec le modèle de lissage multifocale, (SMF).

Pour réaliser ces expériences on se donne un nuage de points $(t_i)_i$ du carré $[0, 1] \times [0, 1]$ et n réels z_i , $n = 900$, qui sont les valeurs bruitées prises par une fonction f_e :

$$\text{pour } i = 1 \dots 900 \quad z_i = f_e(t_i) + \varepsilon_i$$

où $(\varepsilon_i)_i$ un vecteur aléatoire dont les composantes sont indépendantes et de loi normale centrée décart type 0.2.

Notations :

Pour la suite on notera :

- Y_e : Le vecteur de données exactes $Y_e = (f_e(t_i))_{i=1 \dots n}$
- Z : Le vecteur de données bruitées $Z = (z_i)_{i=1 \dots n}$
- Y^k : Le vecteur résultant à l'itération k de l'algorithme de Gauss-Seidel modifié (GSM).
- Y_ρ : Le vecteur résultant de GSM après convergence.
- M : Le nombre de focales utilisées.
- $\omega_1, \dots, \omega_M$: Les focales utilisées .
- N_{iter} : Le nombre d'itérations nécessaire pour la convergence de l'algorithme de (GSM).
- $Ec(k)$: L'écart relatif entre les deux itérés Y^k et Y^{k-1} , $Ec(k) = \frac{\|Y^k - Y^{k-1}\|}{\|Y^k\|}$
- $Er(k)$: L'écart entre l'itéré Y^k et Y_e , $Er(k) = \frac{1}{n} \|Y^k - Y_e\|$
- $E_1(\omega)$: L'écart entre l'itéré Y^1 et Z , $E_1(\omega) = \frac{1}{n} \|Y^1 - Z\|$. Le lissage est obtenu avec un nombre j de focales, dont $j - 1$ sont fixées et la $j^{\text{ème}}$ égale à ω . On effectue à chaque fois une seule itération de GSM.
- $E(\rho)$: L'écart entre Y_ρ et Y_e (erreur exacte), $E(\rho) = \frac{1}{n} \|Y_\rho - Y_e\|$
- $GCV(\rho)$: La fonction de validation croisée généralisée. On utilise une estimation de Monte-Carlo pour calculer la trace.

ρ_g : Le paramètre de lissage qui réalise le minimum de GCV .

ρ_o : Le paramètre de lissage qui réalise le minimum de Er .

N_s : La longueur de la subdivision B-spline. Le nombre de B-splines dans chaque direction est $N_s + 2$

4.4.1 Nombre d'itérations

D'après l'étude théorique de GSM, l'écart relatif entre deux itérés successifs E_c , considéré comme fonction du nombre d'itérations effectuées, est décroissant. Ceci est le cas quand on utilise un faible nombre de focales. ($M \leq 8$). Par contre si on utilise un grand nombre de focales, la décroissance de E_c n'est plus assurée. On a, en effet, rencontré des situations où E_c décroît pendant un certain nombre d'itérations, puis croît ou oscille. Ceci provient de l'instabilité numérique du modèle lorsque l'on utilise un grand nombre de focales pour un lissage SMF.

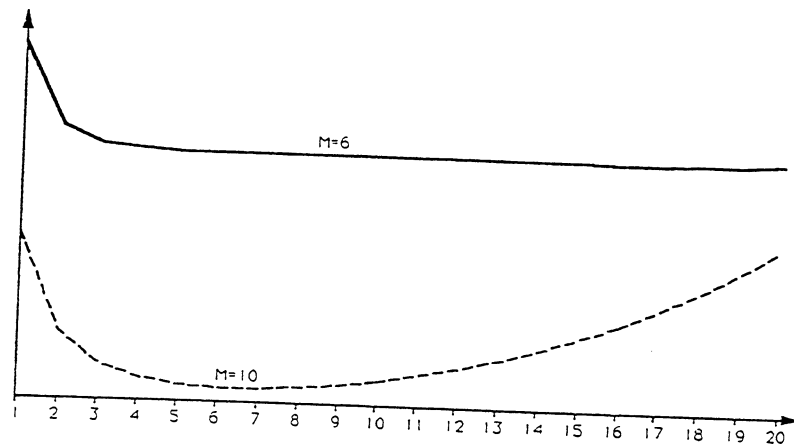


figure 4.1 E_c fonction du nombre d'itérations - $M = 6$ et $M = 10$

On remarque d'autre part, que l'écart E_r a le même comportement en fonction des itérations. On en conclut qu'il n'est pas nécessaire d'effectuer un grand nombre d'itérations de l'algorithme GSM, sinon l'algorithme a tendance à s'écarter de la solution optimale.

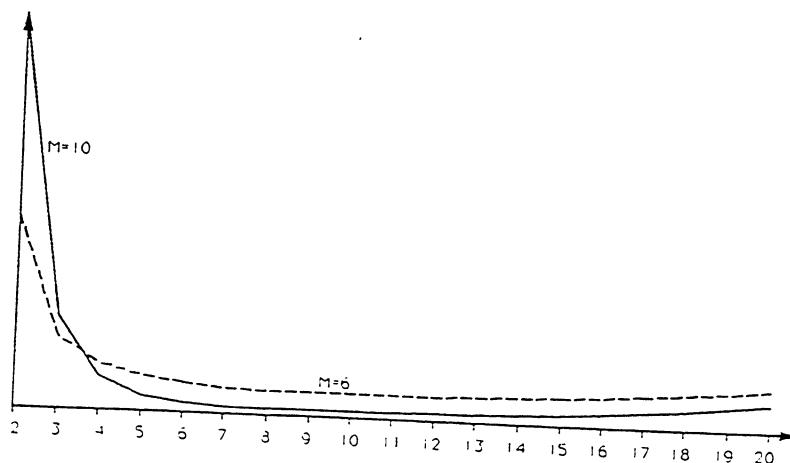


figure 4.2 E_r fonction du nombre d'itérations - $M = 6$ et $M = 10$

4.4.2 Choix du paramètre de lissage ρ

Pour le choix du paramètre de lissage ρ , on utilise la méthode de validation croisée généralisée. Un estimateur de Monte-Carl est utilisé pour estimer la trace au dénominateur. Afin de tester la validité de ce choix, on calcule à chaque fois l'efficacité :

$$Eff = \frac{E(\rho_o)}{E(\rho_g)}$$

Pour tous les exemples qu'on a étudié cette efficacité est toujours proche de 1. On peut en conclure que la minimisation de la fonction GCV donne un bon estimateur de ρ_o .

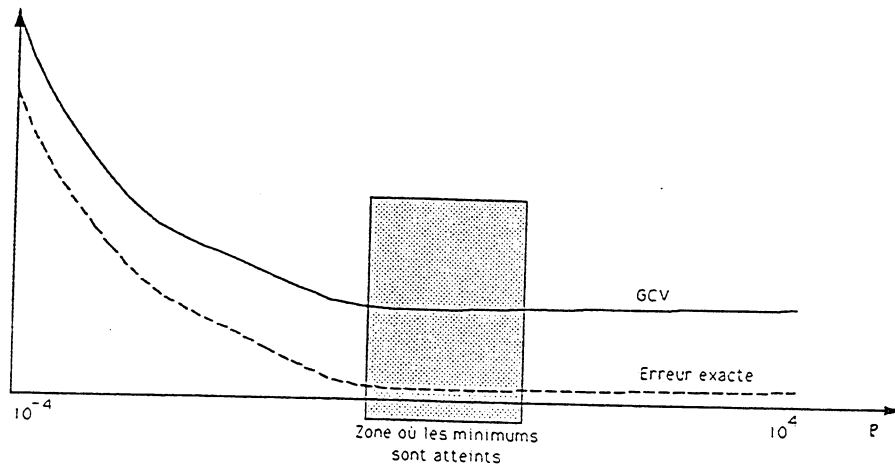


figure 4.3 Erreur exacte et GCV

4.4.3 Choix des focales

Le modèle SMF est très sensible au choix des focales. Si les données (z_i) sont générées par une fonction qui s'écrit sous forme d'une multifocale (Exemple 1), il est nécessaire d'utiliser les focales exactes pour lisser ces données. Si les focales exactes sont inconnues, on peut les rechercher en minimisant l'écart E_1 .

Par contre, si la fonction ne s'écrit pas sous forme multifocale (Exemple 3), la minimisation de E_1 donne de mauvaises focales. Un lissage effectué avec ces focales donne un très mauvais résultat.

4.4.4 Visualisation de quelques exemples

On visualise ici, trois exemples expérimentaux du modèle SMF. Dans ces exemples, les données sont générées par trois fonctions f_e^1, f_e^2 et f_e^3 de type différents. Ces fonctions sont:

Exemple 1 : La fonction est sous forme multifocale .

$$f_e^1(t^1, t^2) = \sum_{j=1}^3 \frac{1}{2} e^{-10r_j}$$

où $r_j = (t^1 - \omega_j^1)^2 + (t^2 - \omega_j^2)^2$
 et $\omega_1 = (0.5, 0.4), \omega_2 = (0.25, 0.75)$ et $\omega_3 = (0.75, 0.75)$.

Exemple 2 : La fonction est la somme d'une fonction bifocale et d'une autre fonction.

$$f_e^1 = g_1 + g_2 + g_3$$

où: $g_1(t^1, t^2) = \frac{1}{2} \cos(2\pi r_1)$,
 $g_2(t^1, t^2) = e^{-10r_2}$,
 $g_3(t^1, t^2) = -\frac{1}{2} (\sin(\pi \tau_1))^2$,

et: $\tau_1 = (t^1 \cos(\pi/2) + t^2 \sin(\pi/2))$,
 $r_1 = ((t^1 - 0.25)^2 + (t^2 - 0.5)^2)$,
 $r_2 = ((t^1 - 0.5)^2 + (t^2 - 0.25)^2)$.

Exemple 3 : La fonction f_e n'est pas multifocale.

$$f_e^3(t^1, t^2) = \frac{1}{2} (\cos(3\pi \tau_1) + \cos(2\pi \tau_2))$$

où: $\tau_1 = (t^1 \cos(\pi/3) + t^2 \sin(\pi/3))$,
 $\tau_2 = (t^1 \cos(2\pi/3) + t^2 \sin(2\pi/3))$.

Pour visualiser la fonction $E_1(\omega)$, qu'on considère comme une fonction des coordonnées des focales, (ω^1, ω^2) , on utilisera une table de motifs. Chaque motif correspond à une valeur de E_1 . Plus le motif en un point est clair plus la valeur de la fonction en ce point est faible.

Exemple 1 :

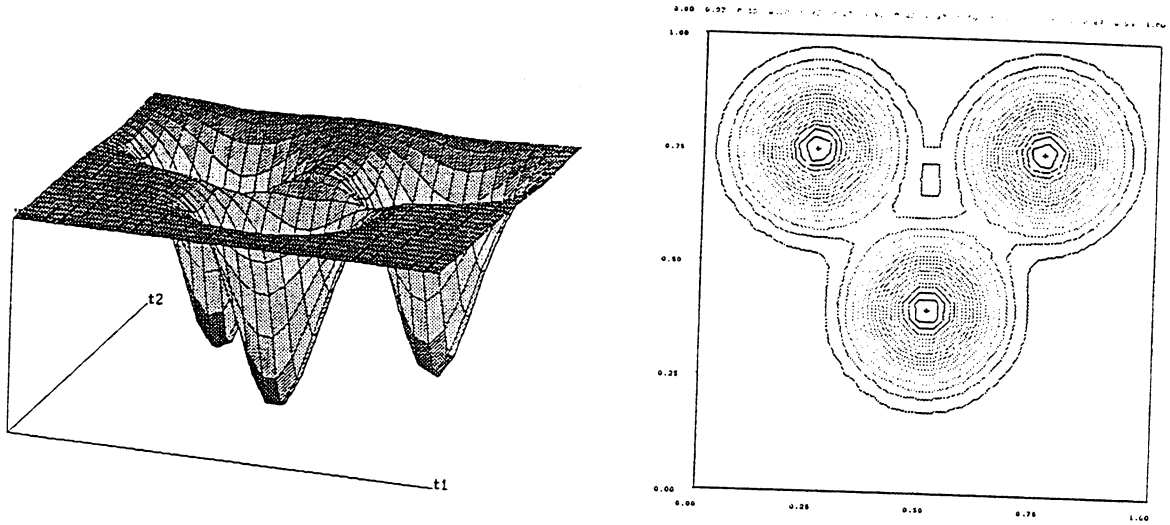


figure 4.4 Fonction exacte (à lisser)

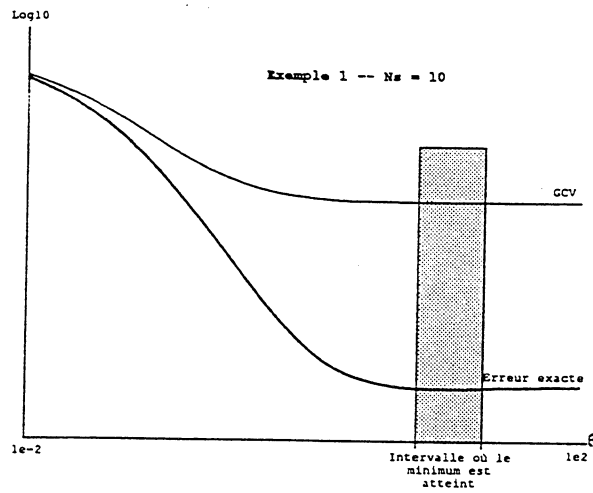


figure 4.5 Fonctions GCV et Er. $N_s = 10$ - Exemple 1

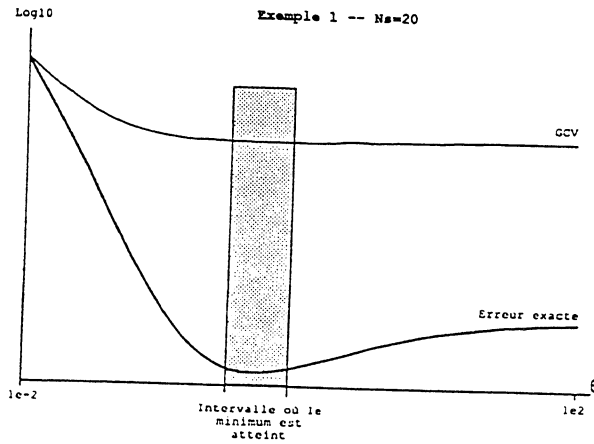


figure 4.6 Fonctions GCV et Er. $N_s = 20$ - Exemple 1

Les fonctions GCV et Er atteignent leur minimum au voisinage de la même valeur. Et l'efficacité de la méthode de validation croisée généralisée pour le choix d'un ρ optimal est:

Pour $N_s = 10$ efficacité = 0,992.

Pour $N_s = 30$ efficacité = 0.989.

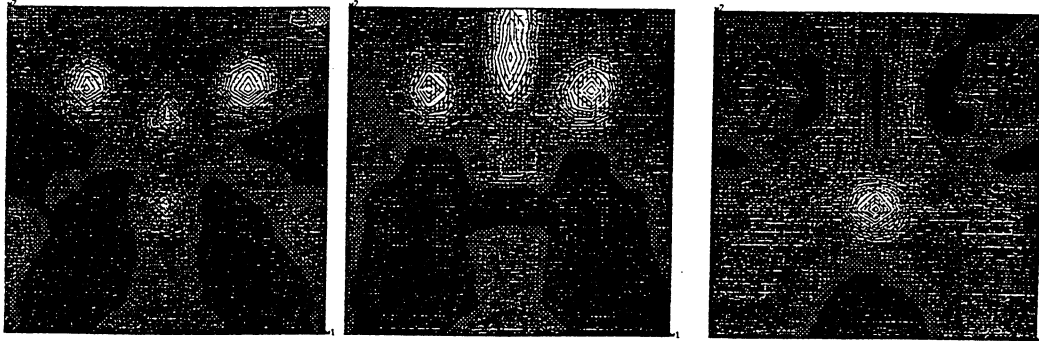


figure 4.7 Tables de niveau de la fonction E_1

$\rho = 10^3$ - Exemple 1

D'après les tables de niveau de E_1 ci-dessus, on voit bien que celle-ci admet des minima aux valeurs exactes des focales, c'est à dire en $(0.5, 0.4)$, $(0.25, 0.75)$ et $(0.75, 0.75)$.

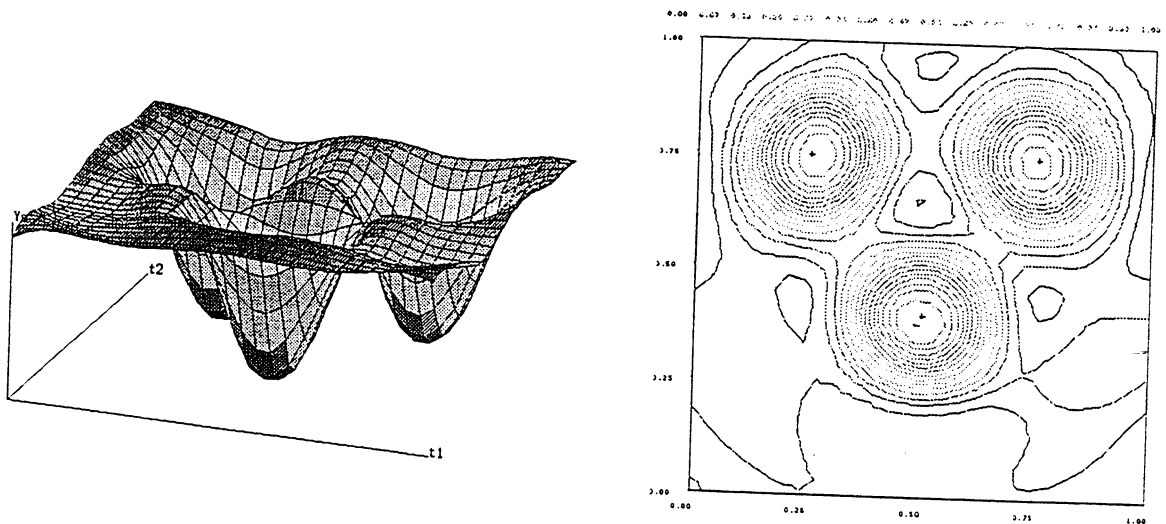


figure 4.8 Fonction lissée en utilisant les focales exactes.

$\rho = \rho_g$ et $N_s = 10$ - Exemple 1

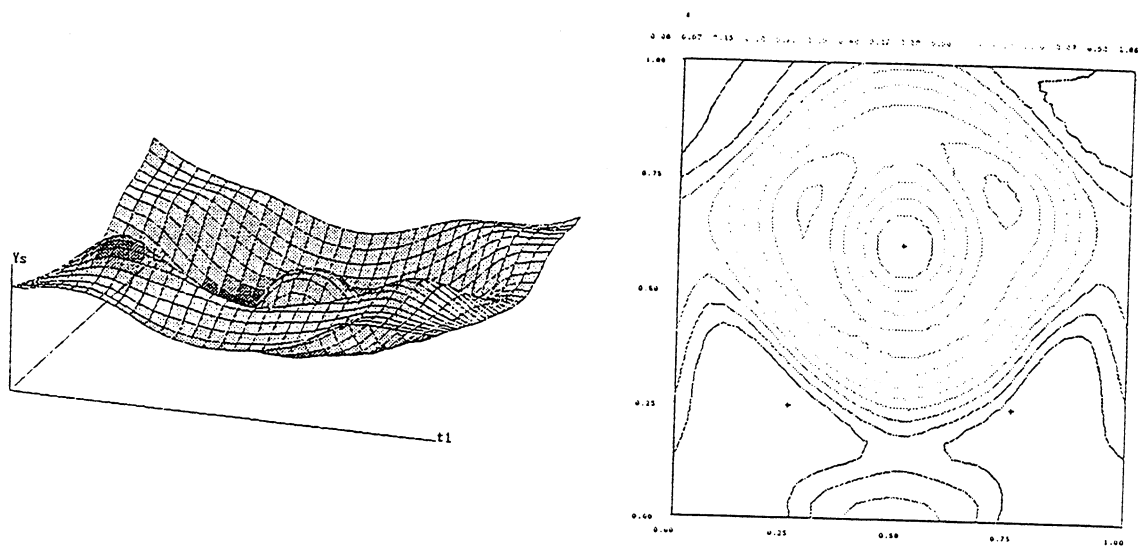


figure 4.9 Fonction lissée en utilisant des focales quelconques,
 $\rho = \rho_g$ et $Ns = 10$ - Exemple 1

On voit bien dans le graphique précédent l'importance du choix des focales dans la qualité de lissage.

Exemple 2 :

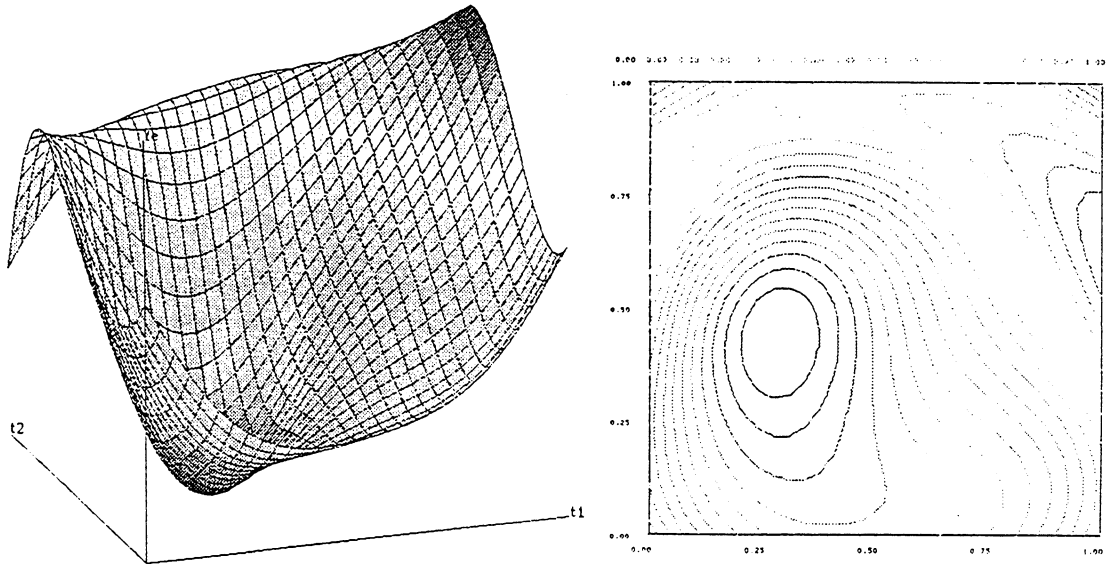
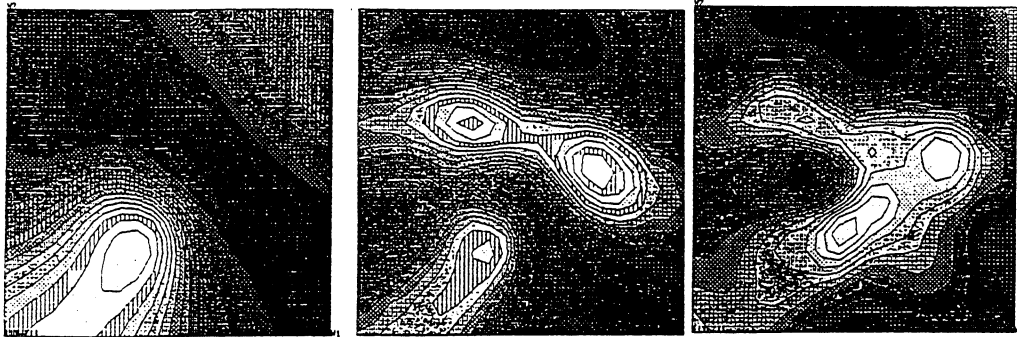


figure 4.10 Fonction exacte (à lisser)

Recherche des focales pour le lissage :

figure 4.11 Tables de niveau de la fonction E_1 .

$$\rho = 10^3 - \text{Exemple 2}$$

D'après les graphiques représentant E_1 , ci-dessus, celle-ci admet son minimum en $(0.375, 0.245)$. En réitérant la recherche de focales on trouve dans l'ordre les trois focales suivantes :

$$\omega_1 = (0.375, 0.245), \omega_2 = (0.74, 0.5) \text{ et } \omega_3 = (0.5, 0.25).$$

On effectue alors un lissage SMF avec ces trois focales. L'efficacité de la méthode de validation croisée est:

Pour $N_s = 10$ efficacité = 0.959.

Pour $N_s = 30$ efficacité = 0.998.

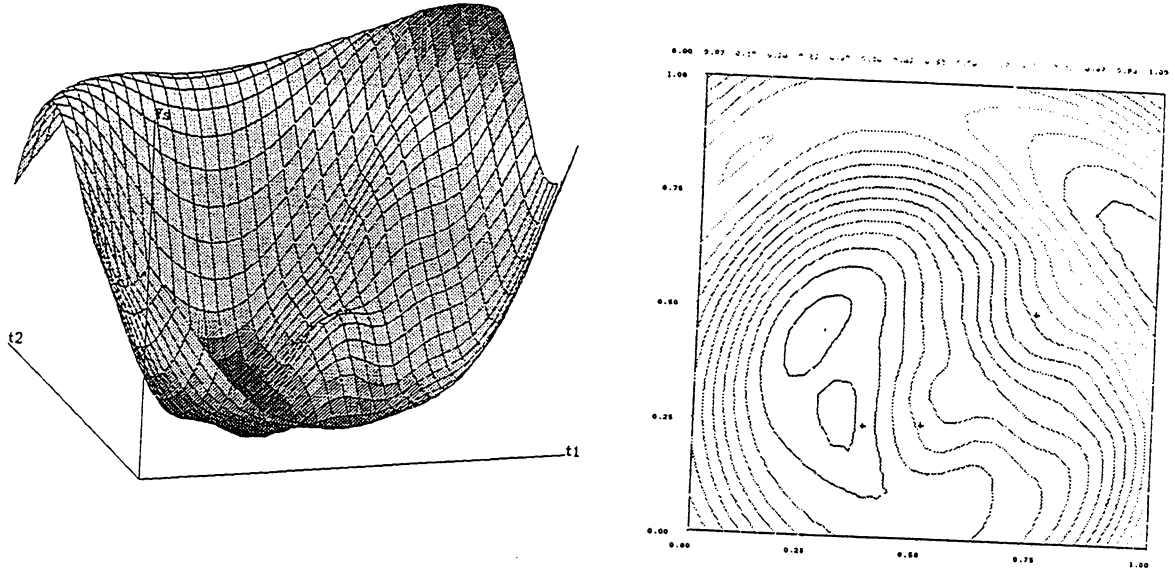


figure 4.12 Fonction lissée en utilisant les focales calculées,

$\rho = \rho_g$ et $N_s = 10$ - Exemple 2

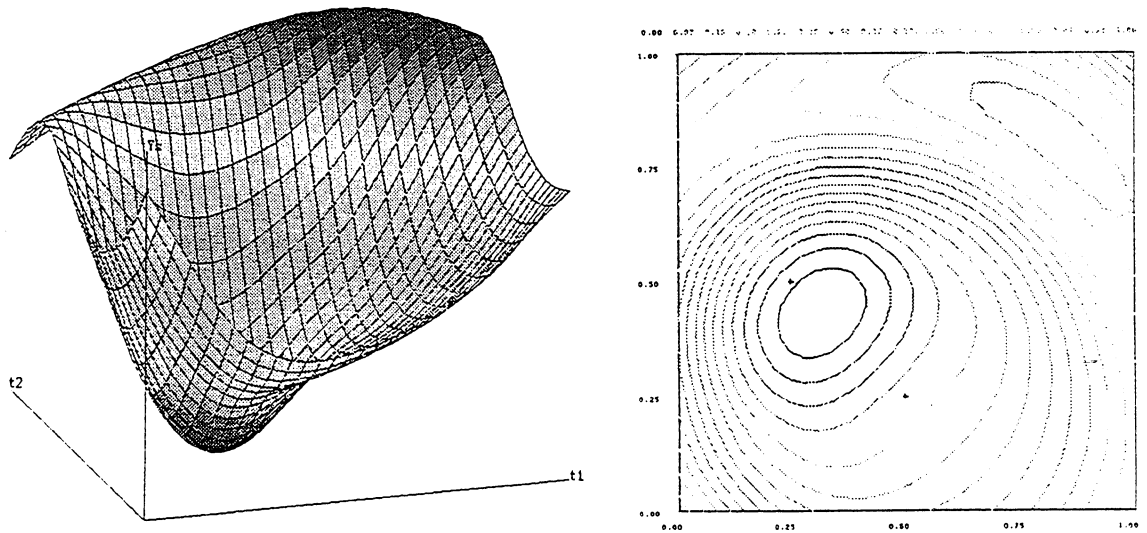


figure 4.13 Fonction lissée en utilisant des focales exactes: $(0.5, 0.25)$ et $(0.25, 0.5)$,

$\rho = \rho_g$ et $N_s = 10$ - Exemple 2

En comparant ces deux graphiques on remarque que la fonction lissée avec des focales exactes approche mieux f_e^2 . Le choix des focales par minimisation de E_1 n'est donc pas très efficace. Cette remarque se confirme avec l'exemple 3.

Exemple 3 :

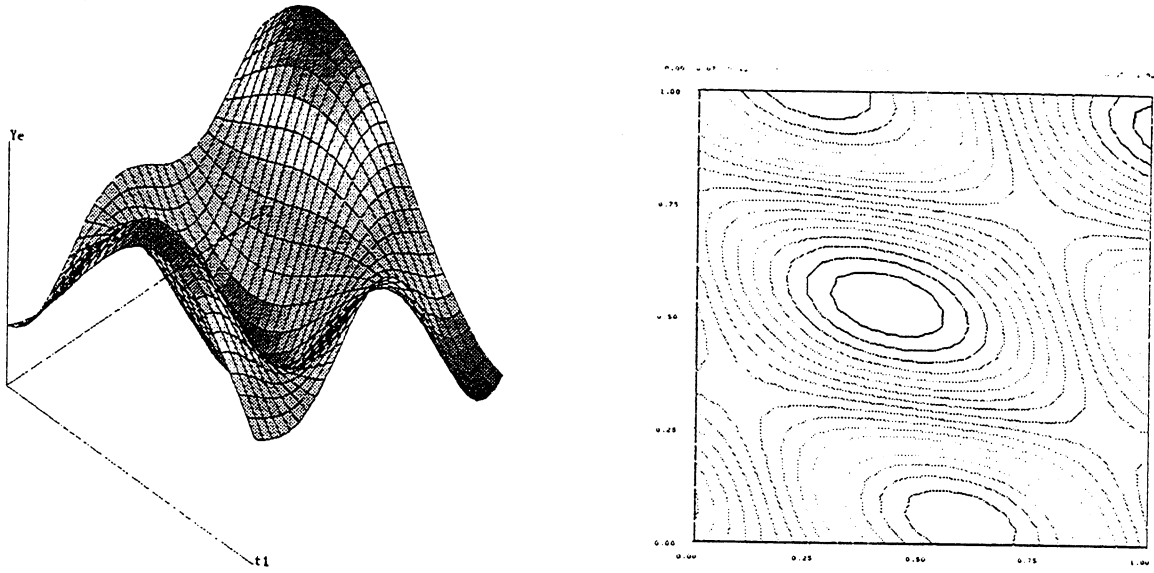
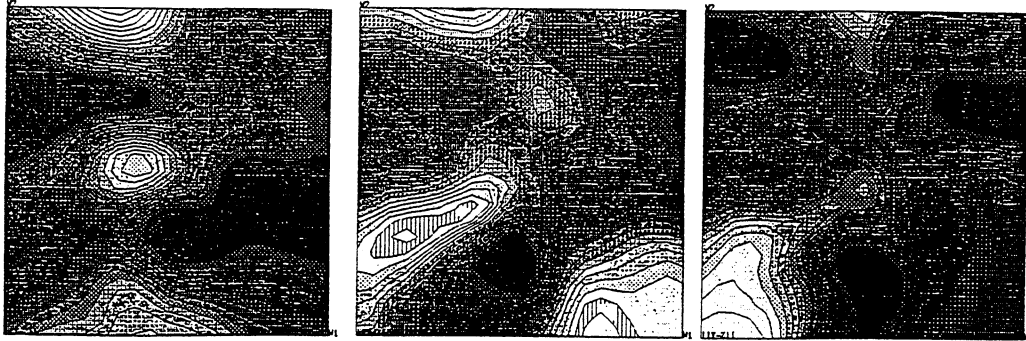


figure 4.14 Fonction exacte (à lisser)

Recherche des focales pour le lissage :

figure 4.15 Tables de niveau de la fonction E_1

$\rho = 10^3$ - Exemple 3

La fonction E_1 admet son minimum en $(0.41, 1)$. En réitérant la recherche de focales on trouve dans l'ordre les trois focales suivantes :

$$\omega_1 = (0.41, 1), \omega_2 = (0.755, 0) \text{ et } \omega_3 = (0.75, 0).$$

On remarque ici que la deuxième et la troisième focale sont presque identiques. On effectue un lissage SMF avec ces trois focales :

Efficacité:

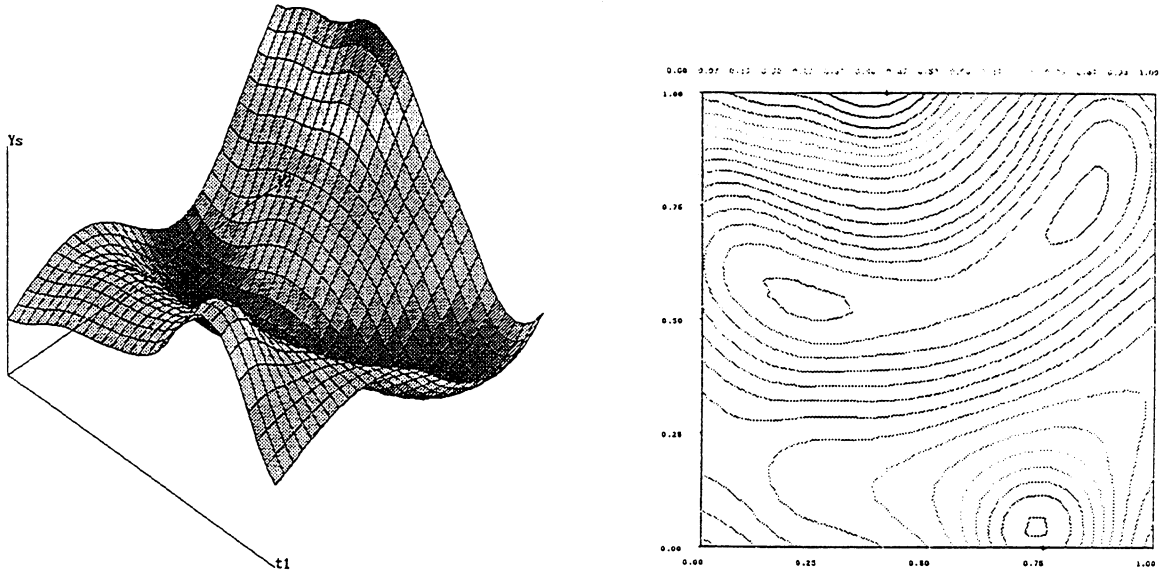
Pour $N_s = 10$ efficacité = 0.992.Pour $N_s = 30$ efficacité = 0.986.

figure 4.16 Fonction lissée en utilisant les focales calculées,

$\rho = \rho_y$ et $N_s = 10$ - Exemple 3

En examinant les résultats du lissage, on ne peut pas dire qu'on a approché la fonction f_e^3 . On obtient de meilleurs résultats en effectuant un lissage avec des focales approximatives. En regardant les courbes de niveau de la fonction exacte, on remarque que celle-ci admet un certain nombre de pics, symbolisés par des ellipses. On prend alors les centres approximatifs de ces ellipses comme focale de lissage. Le résultat obtenu est bien meilleur que le résultat obtenu en lissant les données avec les focales calculées.

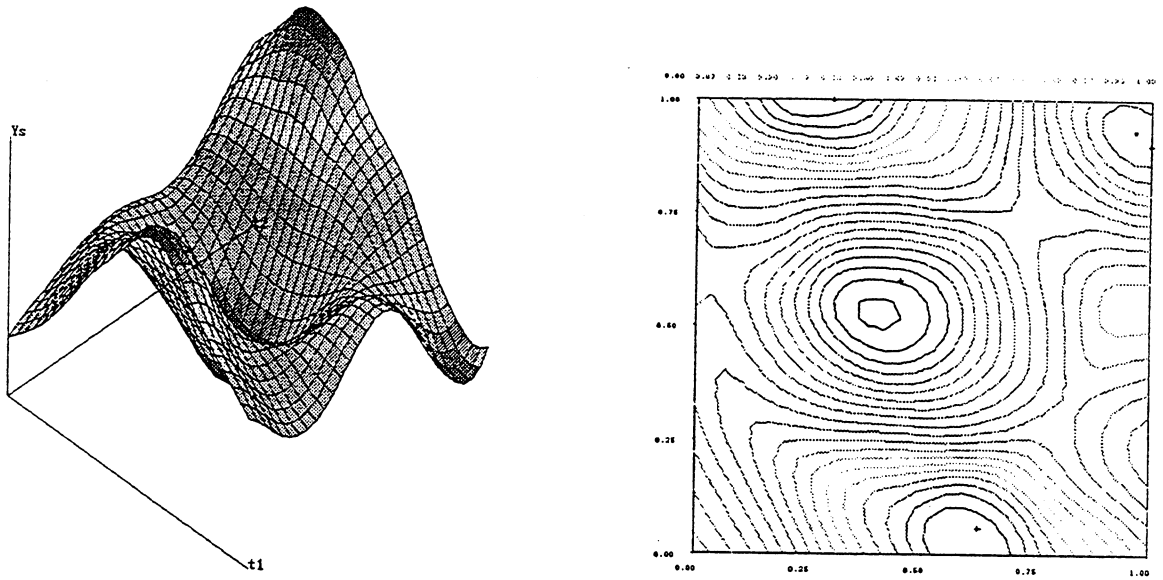


figure 4.17 Fonction lissée en utilisant des focales approximatives,

$\rho = \rho_g$ et $N_s = 10$ - Exemple 3

4.5 Remarques

- Choix du critère de lissage unidimensionnel:

Dans ce qui a précédé, on a utilisé le modèle de lissage SMFen utilisant l'opérateur de lissage unidimensionnel J_1 :

$$J_1(f) = \int_0^{r_{max}} \left(\frac{1}{r} f'^2(r) + r f''^2(r) \right) dr$$

Comme on l'a précisé précédemment, cet opérateur nous assure la dérivabilité de la fonction lissée aux focales de lissage $\omega_1, \dots, \omega_M$.

Notons σ_s la spline de lissage et $\sigma_1, \dots, \sigma_M$ les splines de lissage partiel:

$$\sigma_s = \sigma_1 + \dots + \sigma_M.$$

Chaque σ_j . Pour $j = 1, \dots, M$, est une fonction de révolution autour de la focale ω_j , elle est sous la forme :

$$\sigma_j(t) = f_j(r_j) \quad \text{où } r_j = ||t - \omega_j||$$

Chaque f_j est une fonction réelle définie sur un intervalle de \mathbb{R} de la forme $[0, r_{max}]$, à dérivée nulle en 0.

Dans l'exemple 1, les f_j , $j = 1, \dots, 3$ sont de la forme:

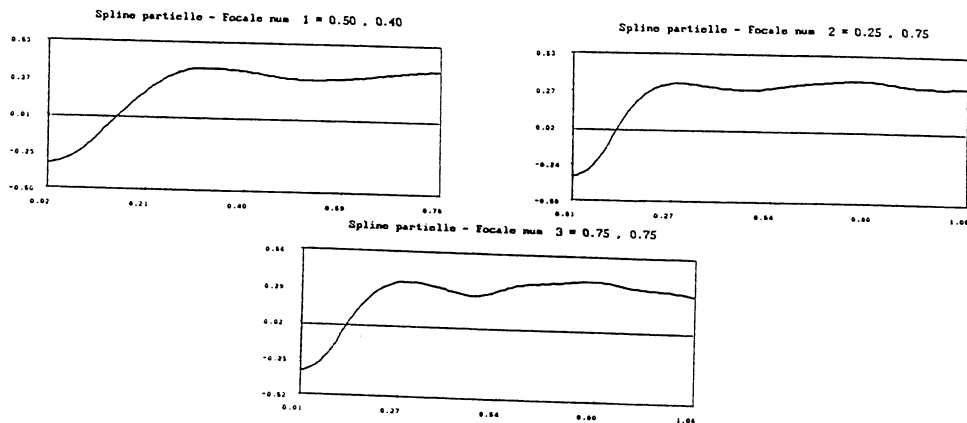


figure 4.18 Courbe de f_1 , f_2 et f_3 , opérateur de lissage unidimensionnel J_1 , focales exactes - Exemple 1

Si on utilise, au lieu de J_1 , l'opérateur de lissage unidimensionnel classique J :

$$J(f) = \int_0^{r_{max}} (f'')^2(r) dr$$

on obtient des fonctions f_j n'admettant pas nécessairement une dérivée nulle en 0. On obtient pour l'exemple 1 les f_j sous la forme :

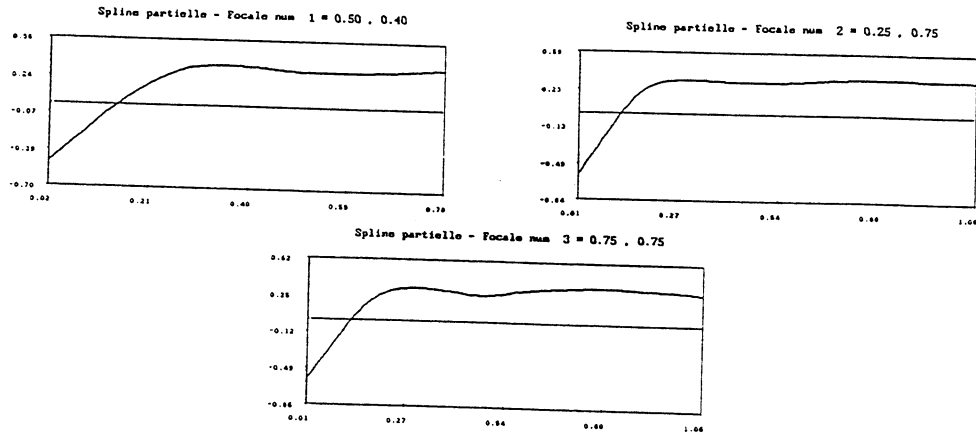


figure 4.19 Courbe de f_1 , f_2 et f_3 , opérateur de lissage unidimensionnel J , focales exactes - Exemple 1

La fonction σ_x présente alors des discontinuités des dérivées premières aux focales. Ces discontinuités sont symbolisées par des "pics pointus" :

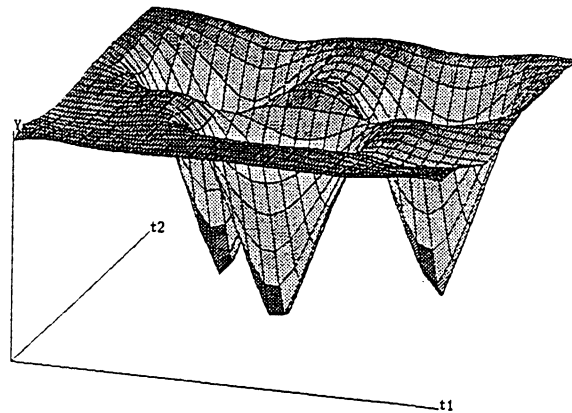


figure 4.20 Fonction lissée, opérateur de lissage unidimensionnel J_1 , focales exactes - Exemple 1

Le choix de J_1 , comme opérateur de lissage unidimensionnel, nous assure donc, la dérivabilité de la spline solution aux focales.

On a remarqué, d'autre part, que la surface de la fonction lissée σ_s présente toujours un "plat" au voisinage de chaque focale ω_j . Ceci veut dire que les dérivées partielles de celle-ci sont presque nulles dans un voisinage de ω_j . Ceci est souhaitable lorsque les ω_j sont les focales exactes (exemple 1). Par contre si les ω_j ne sont que des focales approximatives (exemple 2 et exemple 3), ceci constitue une source d'erreur.

- Recherche des focales:

On a essayé dans ce qui a précédé de rechercher les focales de lissage en minimisant E_1 . Les résultats obtenus ne sont pas convaincants lorsque la fonction à approcher ne s'écrit pas sous forme d'une multifocale. En plus la minimisation de E_1 pose certains problèmes, car cette fonction n'est pas convexe et présente beaucoup de minima locaux.

- Paramètre de lissage et nombre de B-splines:

Si la fonction à approcher s'exprime sous forme d'une multifocale:

$$\sigma^e(t) = \sum_{j=1}^M f_j^e(\|t - \omega_j\|)$$

et si parmi les fonctions f_j il y en a qui présentent beaucoup de variations et d'autres moins variantes, alors un lissage par SMF, comme on l'a décrit, peut ne pas donner des résultats très satisfaisants, car on utilise le même paramètre de lissage ρ et le même nombre N_s de B-splines pour tous les lissages monofocales. Pour améliorer la qualité du lissage, il serait plus judicieux d'utiliser un paramètre de lissage ρ_j et un nombre de B-splines N_s pour chaque lissage monofocale.

Le problème qui se pose alors, est de choisir tous ces paramètres.

- Extension en dimension $d \geq 2$:

Le modèle SMF peut s'étendre, d'une façon naturelle, au lissage de fonctions à d variables. Il suffit pour cela d'écrire la solution σ_s comme somme de M fonctions $\sigma_1, \dots, \sigma_M$. Chaque σ_j étant une fonction de "révolution" autour d'une focale ω_j :

$$\sigma_j(t) = f_j(\|t - \omega_j\|) \text{ pour } t \in \mathbb{R}^d \text{ et } j = 1 \dots M$$

où les f_j sont des fonctions à une seule variable.

4.6 Conclusion

Le modèle SMF, s'adapte bien au lissage de fonctions qui s'écrivent sous forme d'une multifocale, avec une connaissance a priori des focales exactes. Dans les cas contraires, on n'a pas de moyens efficaces pour calculer des focales approximatives qui donneraient une bonne approximation de la fonction exacte.

Ce modèle présente un grand gain en temps de calcul et en mémoire. Il présente aussi l'avantage de pouvoir approcher des fonctions réputées être difficiles à approcher par les méthodes de lissage classiques. Mais il demeure très dépendant des focales de lissage. Ce qui est un grand handicap.

Chapitre 5

Splines multidirectionnelles

Comme dans le chapitre précédent on suppose donnés un nuage T de points de \mathbb{R}^2 , $T = (t_i^1, t_i^2)_{i=1 \dots n}$ et un vecteur Z de \mathbb{R}^n , $Z = (Z_i)_{i=1 \dots n}$, dont les composantes sont les valeurs entachées d'erreurs prises par une fonction inconnue f_0 , aux points du nuage T .

$$Z_i = f_0(t_i^1, t_i^2) + \varepsilon_i \quad i = 1, \dots, n$$

$(\varepsilon_i)_i$ un vecteur aléatoire de loi normale, centrée et de variance inconnue.

Dans ce chapitre nous proposons une méthode permettant de trouver une fonction spline σ lissant les données Z . σ sera représentée comme somme de M fonctions σ_j . Chaque σ_j est une fonction constante perpendiculairement à une direction \mathcal{D}_{θ_j} faisant un angle θ_j avec l'axe des x et elle est obtenue par résolution d'un problème de lissage unidimensionnel. σ peut alors être considérée comme une spline de lissage relativement à une fonctionnelle semi-hilbertienne L obtenue par inf-convolution de M fonctionnelles semi-hilbertiennes L_j .

On suppose que $n \geq 3$ et que parmi les points du nuage T il y en a au moins trois qui ne sont pas alignés sur une droite orthogonale à l'une des direction \mathcal{D}_{θ_j} .

5.1 Splines monodirectionnelles

On se donne une direction \mathcal{D}_{θ_1} faisant un angle θ_1 avec l'axe horizontal. On considère les fonctions f de la forme :

$$f(t_i^1, t_i^2) = \varphi(t_i^1 \cos \theta_1 + t_i^2 \sin \theta_1) \quad (5.12)$$

φ étant une fonction de $H^2[a, b]$ où :

$$a = \underset{i=1 \dots n}{\text{Min}} (t_i^1 \cos \theta_1 + t_i^2 \sin \theta_1) \text{ et } b = \underset{i=1 \dots n}{\text{Max}} (t_i^1 \cos \theta_1 + t_i^2 \sin \theta_1)$$

On définit alors une fonctionnelle semi-hilbertienne L_1 par :

$$L_1(f) = \begin{cases} J_1(\varphi) & \text{si } f \text{ s'écrit sous la forme} \\ +\infty & \text{sinon} \end{cases} \quad (5.12)$$

$$\text{où } J_1(\varphi) = \int_a^b (\varphi''(\tau))^2 d\tau.$$

Le noyau N_1 de L_1 est de la forme :

$$N_1 = \left\{ f \in (\mathbb{R})^{\mathbb{R}^2} / \exists \varphi \in \mathcal{P}_1, f(t_i^1, t_i^2) = \varphi(t_i^1 \cos \theta_1, t_i^2 \sin \theta_1) \right\}$$

\mathcal{P}_1 étant l'ensemble des polynômes de degré inférieur ou égal à 1 définis sur \mathbb{R} .

Le problème de lissage selon la direction θ_1 , se ramène donc à résoudre :

$$\underset{f}{\text{Minimiser}} \left(L_1(f) + \rho \sum_{i=1}^n [f(t_i^1, t_i^2) - Z_i]^2 \right) \quad (P_2)$$

La condition d'unicité de la solution est vérifiée. En effet :

$$N_1 \cap I_2 = \{0\},$$

où I_2 est l'ensemble des fonctions f de \mathbb{R}^2 dans \mathbb{R} qui s'annulent aux points du nuage T .

La solution de ce problème peut être déduite de la solution du problème de lissage unidimensionnel suivant :

$$\underset{\varphi}{\text{Minimiser}} \left(J_1(\varphi) + \rho \sum_{i=1}^n [\varphi(\tau_i) - Z_i]^2 \right) \quad (P_1)$$

où : $\tau_i = t_i^1 \cos \theta + t_i^2 \sin \theta, i = 1 \dots n$.

Pour éviter l'opération coûteuse de mise en ordre des points $(\tau_i)_i$, on choisit plutôt une base de B-splines sur un maillage uniforme de l'intervalle $[a, b]$ et on minimisera sur l'espace de spline engendrées.

On se ramène alors à un problème de lissage par B-splines régularisées. Ce problème se résout aisément par des techniques classiques.

Pour la suite, notons par Spl_1 l'application linéaire qui à chaque vecteur Z de \mathbb{R}^n associe la solution σ_1 du problème de lissage bidimensionnel (P_2) pour les données Z .

$$Spl_1 : \mathbb{R}^n \rightarrow E_2$$

Notons S_1 la matrice de lissage correspondante.

$$S_1 : Z \in \mathbb{R}^n \rightarrow Y \in \mathbb{R}^n$$

tel que $Y_i = \sigma_1(t_i^1, t_i^2)$ pour $i = 1, \dots, n$ et $\sigma_1 = Spl_1(Z)$

5.2 Splines multidirectionnelles

On se donne un nombre fini de directions d'angles θ_j , pour $j = 1 \dots M$. On construit de la même façon que précédemment M fonctionnelles semi-hilbertiennes L_j sur $(\mathbb{R})^{\mathbb{R}^2}$. On forme par inf-convolution des L_j une fonctionnelle semi-hilbertienne L :

$$L(f) = \bigvee_{j=1}^M L_j(f)$$

Le noyau N de L est alors la somme algébrique des noyaux N_j des L_j :

$$N = \sum_{j=1}^M N_j$$

On se pose alors, le problème de lissage suivant :

$$\underset{f}{\text{minimiser}} L(f) + \rho \sum_{j=1}^M (f(t_i^1, t_i^2) - Z_i)^2 \quad (I)$$

Ce problème admet une solution unique si et seulement si :

$$\left. \begin{array}{l} f_j \in N_j \\ \sum_{j=1}^M f_j(t_i^1, t_i^2) = 0 \end{array} \right\} \begin{array}{l} j = 1 \dots M \\ i = 1 \dots n \end{array} \implies \sum_{j=1}^M f_j = 0 \quad j = 1 \dots M \quad (5.13)$$

La condition (5.13) est vérifiée; en effet :

N_j le noyau de L_j est l'ensemble des fonctions f_j de \mathbb{R}^2 dans \mathbb{R} qui sont des polynômes de degré 1 dans la direction \mathcal{D}_{θ_j} et qui sont constantes dans la direction perpendiculaire, donc $f = f_1 + \dots + f_M$ est un polynôme de degré un. Si en plus f s'annule aux points $(t_i)_i$, alors elle est identiquement nulle.

Le problème (I) admet donc, une solution unique σ qui sera appelée spline de lissage multidirectionnelle.

Comme on l'a vu précédemment, le problème (I) est équivalent, dans un certain sens (voir chapitre précédent), au problème (II) suivant :

$$\underset{f_j \in E_2, j=1 \dots M}{\text{Minimiser}} \left(\sum_{j=1}^M L_j(f_j) + \sum_{i=1}^n \left(\sum_{j=1}^M f_j(t_i^1, t_i^2) - Z_i \right)^2 \right) \quad (II)$$

Ce problème admet une solution unique si et seulement si :

$$\left\{ f_j \in N_j, j = 1 \dots M, \quad \sum_{j=1}^M f_j \in I_2 \right\} \implies \{f_j = 0 \text{ pour } j = 1 \dots M\}$$

Cette proposition n'est pas vérifiée. Il suffit de prendre $f_j = c_j$ pour $j = 1 \dots M$, où les c_j sont des constantes à somme nulle : $c_1 + \dots + c_M = 0$.

5.3 Algorithme de Gauss-Seidel modifié

5.3.1 Hypothèses de l'algorithme

De la même façon que dans le chapitre précédent, on montre que le problème (II) vérifie les conditions d'application de l'algorithme de Gauss-Seidel modifié et qu'il s'écrit sous la forme :

$$\underset{\vec{Y} \in E}{\text{minimiser}} \left\langle \vec{Y}, A\vec{Y} \right\rangle_{nM} - 2 \left\langle \vec{Y}, b \right\rangle_{nM} \quad (P')$$

où A et b sont sous la forme :

$$A = \begin{pmatrix} \frac{1}{\rho}\Omega_1 + I & I & \cdots & I \\ I & \frac{1}{\rho}\Omega_2 + I & \cdots & I \\ \vdots & \ddots & \ddots & \vdots \\ I & \cdots & \frac{1}{\rho}\Omega_{M-1} + I & I \\ I & \cdots & I & \frac{1}{\rho}\Omega_M + I \end{pmatrix} \quad \text{et } b = \begin{pmatrix} Z \\ Z \\ \vdots \\ Z \\ Z \end{pmatrix}$$

chaque Ω_j étant la matrice symétrique semi-définie positive vérifiant :

$$\begin{aligned} \underset{f_j \in E}{\text{Inf}} \quad L_j(f_j) &= \langle Y_j, \Omega_j Y_j \rangle_n \quad j = 1 \cdots M \\ f_j(t_i) &= Y_j^i \end{aligned}$$

La matrice A est symétrique semi-définie positive. Comme :

$$\left\langle \vec{Y}, A\vec{Y} \right\rangle = \sum_{j=1}^M \left\langle \vec{Y}_j, \Omega_j \vec{Y}_j \right\rangle + \left\| \sum_{j=1}^M Y_j \right\|^2,$$

son noyau \mathcal{N} est l'ensemble des vecteurs $\vec{\delta} = (\delta_1, \dots, \delta_M)$ appartenant à E , tels que :

1 - chaque δ_j , pour $j = 1 \cdots M$, correspond aux valeurs prises par une fonction appartenant au noyau de L_j :

$$\delta_j^i = a_j (t_i^1 \cos \theta_j + t_i^2 \sin \theta_j) + b_j, \quad \text{avec } a_j \in \mathbb{R}, b_j \in \mathbb{R}$$

2 - les δ_j sont de somme nulle :

$$\sum_{j=1}^M \delta_j = 0$$

En effet :

Notons f_j^δ la fonction :

$$(t^1, t^2) \rightarrow a_j (t^1 \cos \theta_j + t^2 \sin \theta_j) + b_j,$$

f_j^δ appartient au noyau de L_j et en plus $f_j^\delta(t_i) = \delta_j^i, i = 1 \dots n$.

Donc :

$$\begin{aligned} \text{Inf}_{\substack{f_j \in E \\ f_j(t_i) = \delta_j^i}} L_j(f_j) &= L_j(f_j^\delta) = 0 \end{aligned}$$

et l'on a alors :

$$\langle \delta_j, \Omega_j \delta_j \rangle_n = 0 \quad j = 1 \dots M.$$

En plus $\left\| \sum_{j=1}^M \delta_j \right\| = 0$.

On en conclut que :

$$\langle \vec{\delta}, A\vec{\delta} \rangle = 0$$

□

De la même façon que précédemment, on peut aussi écrire E sous forme de somme directe :

$$E = V_1 \oplus \dots \oplus V_M$$

On peut donc appliquer l'algorithme de Gauss-Seidel modifié, selon les deux étapes précédemment décrites.

5.3.2 Itéré de norme minimale

A l'étape précédente on a calculé, par une itération de l'algorithme de Gauss-Seidel, le vecteur $\tilde{Y}^\nu = (\tilde{Y}_1^\nu, \dots, \tilde{Y}_M^\nu)$.

On calcule la projection sur F de \tilde{Y}^ν qu'on notera :

$$\vec{Y}^\nu = P_F(\tilde{Y}_1^\nu, \dots, \tilde{Y}_M^\nu) = (Y_1^\nu, \dots, Y_M^\nu)$$

P_F est la projection orthogonale sur F . Il s'agit donc, de chercher l'élément $\vec{\delta}$ de \mathcal{N} tel que:

$$\left\| \tilde{Y}^\nu - \vec{\delta} \right\|_{nM} \text{ soit minimale}$$

Le vecteur $\vec{\delta}$ s'écrit $(\delta_1, \dots, \delta_M)$ avec :

$$\delta_j = (\alpha_j^1 t_i^1 + \alpha_j^2 t_i^2 + \alpha_j^3)_{i=1}^n \quad j = 1 \dots M,$$

où :

$$\begin{aligned} \alpha_j^1 &= a_j \cos \theta_j \\ \alpha_j^2 &= a_j \sin \theta_j \\ \alpha_j^3 &= b_j \end{aligned}$$

Si l'on note $\alpha_j = (\alpha_j^1, \alpha_j^2, \alpha_j^3)$, on a :

$$\left\{ \begin{array}{l} \langle u_j, \alpha_j \rangle = 0 \quad u_j = (\sin \theta_j, \cos \theta_j, 0) \in \mathbb{R}^3 \\ \sum_{j=1}^M \alpha_j = 0 \quad \left(\Rightarrow \sum_{j=1}^M \delta_j = 0 \right) \end{array} \right. \quad (5.14)$$

Si on note P la matrice $(n \times 3)$ telle que :

$$P^t = \begin{bmatrix} 1 & \dots & 1 \\ t_2^1 & \dots & t_2^n \\ t_1^1 & \dots & t_1^n \end{bmatrix}, \quad \text{où } t^i = (t_i^1, t_i^2) \quad i = 1 \dots n$$

le problème de recherche d'un itéré de norme minimale revient donc au problème suivant :

$$\underset{\alpha \text{ vérifiant (5.14)}}{\text{minimiser}} \quad \sum_{j=1}^M \left\| \tilde{Y}_j^\nu - P \alpha_j \right\|^2$$

δ_j^i représente les valeurs aux points (t_i) d'un polynôme de degré 1 dans la direction θ_j . On peut donc, l'écrire $\delta_j^i = a_j (t_i^1 \cos \theta_j + t_i^2 \sin \theta_j) + b_j$, et on notant R_j la matrice (3×2) définie par :

$$R_j = \begin{pmatrix} \cos \theta_j & 0 \\ \sin \theta_j & 0 \\ 0 & 1 \end{pmatrix}$$

et :

$$\lambda_j = \begin{pmatrix} a_j \\ b_j \end{pmatrix}$$

on aura $R_j \lambda_j = \alpha_j$ et :

$$\delta_j = P R_j \lambda_j$$

On veut minimiser $\sum_{j=1}^M \left\| \tilde{Y}_j - P R_j \lambda_j \right\|^2$ sous la contrainte $\sum_{j=1}^M \alpha_j = 0$, c'est à dire :

$$\sum_{j=1}^M R_j \lambda_j = 0$$

On introduit alors les multiplicateurs de Lagrange $\mu \in \mathbb{R}^3$:

$$\min_{(\lambda_j)} \max_{\mu} \sum_{j=1}^M \left\| \tilde{Y}_j^\nu - PR_j \lambda_j \right\|^2 - \left\langle \mu, \sum_{j=1}^M R_j \lambda_j \right\rangle_3$$

et avec la notation $Q_j = PR_j$:

$$\min_{(\lambda_j)} \max_{\mu} \sum_{j=1}^M \left(\left\| \tilde{Y}_j^\nu - Q_j \lambda_j \right\|^2 - \langle R_j^t \mu, \lambda_j \rangle_3 \right)$$

Les conditions d'optimalité s'écrivent :

$$\begin{cases} \frac{\partial(-)}{\partial \lambda_k} = 0 & \Rightarrow Q_k^t Q_k \lambda_k = Q_k^t \tilde{Y}_k^\nu - R_k^t \mu \\ \frac{\partial(-)}{\partial \mu} = 0 & \Rightarrow \sum_{j=1}^M R_j \lambda_j = 0 \end{cases}$$

On résout la première équation en considérant μ connu, et on reporte dans la deuxième équation. On obtient alors le système :

$$\begin{cases} (Q_k^t Q_k)^{-1} Q_k \tilde{Y}_k^\nu - (Q_k^t Q_k)^{-1} Q_k \mu & = \lambda_k \quad k = 1 \dots M \\ \sum_{j=1}^M [R_j (Q_j^t Q_j)^{-1} \tilde{Y}_j^\nu] + \sum_{j=1}^M [R_j (Q_j^t Q_j)^{-1} R_j^t \mu] & = 0 \end{cases}$$

En utilisant les notations :

$$A_j = Q_j^t Q_j, \quad B_j = A_j^{-1}, \quad H = \sum_{j=1}^M R_j B_j R_j^t \quad \text{et} \quad q = - \sum_{j=1}^M (R_j B_j Q_j^t \tilde{Y}_j^\nu)$$

on obtient le système :

$$H \mu = q$$

d'où μ et donc $\lambda_k = \begin{pmatrix} a_k \\ b_k \end{pmatrix}$, pour $k = 1 \dots M$.

L'itéré de norme minimale est donc :

$$\vec{Y}^\nu = (Y_1^\nu, \dots, Y_M^\nu)$$

où :

$$Y_k^\nu = \tilde{Y}_k^\nu - (a_k (t_i^1 \cos \theta_k + t_i^2 \sin \theta_k) + b_k)_{i=1}^n$$

5.3.3 Algorithme

L'algorithme pour le calcul du vecteur Y résultant d'un lissage multidirectionnel se résume de la façon suivante :

Itération 0 :

Tous les Y_j sont mis à 0 : $Y_j^0 = 0$.

⋮

Itération ν :

$Y_1^{\nu-1} \dots Y_M^{\nu-1}$ déjà calculés.

1- On calcule $\tilde{Y}_1^\nu \dots \tilde{Y}_M^\nu$ par :

$$\tilde{Y}_j^\nu = S_j \left(Z - \sum_{k=1}^{j-1} \tilde{Y}_k^\nu - \sum_{k=j+1}^M Y_k^{\nu-1} \right), \quad j = 1 \dots M$$

2- On calcule les vecteurs de norme minimale :

$$Y_1^\nu \dots Y_M^\nu$$

par la procédure décrite précédemment.

On arrête le processus dès qu'on a atteint une stabilisation satisfaisante, par exemple dès que:

$$\|Y^\nu - Y^{\nu-1}\| \leq \varepsilon$$

ou

$$\frac{\|Y^\nu - Y^{\nu-1}\|}{\|Y^\nu\|} \leq \varepsilon$$

ε étant une précision fixée par l'utilisateur.

Le vecteur solution du problème de lissage multidirectionnel est alors :

$$Y = \sum_{j=1}^M Y_j^\nu$$

5.4 Expérimentations numériques

Dans ce paragraphe on présente quelques expériences numériques qu'on a effectuées avec le modèle de lissage par spline multidirectionnelle (SMD).

Pour réaliser ces expériences, on se donne un nuage de points $(t_i)_{i=1\dots n}$ de $[0, 1] \times [0, 1]$ et n réels z_i , $n = 900$, qui sont les valeurs bruitées prises par une fonction f_e :

$$\text{pour } i = 1, \dots, n \quad z_i = f_e(t_i) + \varepsilon_i$$

où $\varepsilon = (\varepsilon_i)_{i=1, \dots, n}$ est un vecteur aléatoire dont les composantes sont indépendantes et de loi normale centrée et d'écart type 0.2.

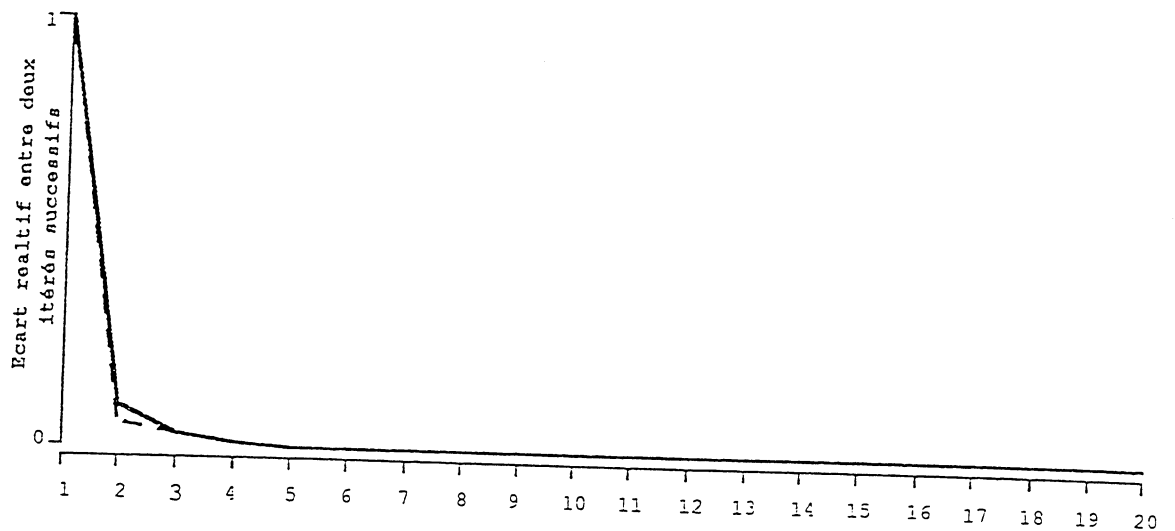
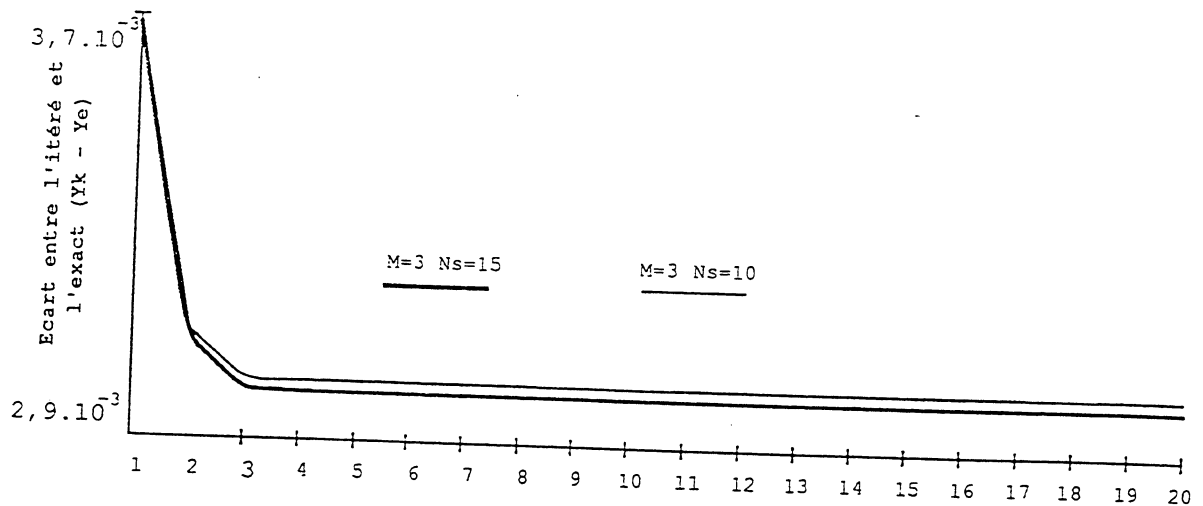
Notations :

Pour la suite on notera :

- Y_e : Le vecteur de données exactes $Y_e = (f_e(t_i))_{i=1\dots n}$
- Z : Le vecteur de données bruitées $Z = (z_i)_{i=1\dots n}$
- Y^k : Le vecteur résultant à l'itération k de l'algorithme de Gauss-Seidel modifié, (GSM).
- Y_ρ : Le vecteur résultant de l'algorithme de GSM après convergence.
- M : Le nombre de directions utilisées.
- $\theta_1, \dots, \theta_M$: Les directions utilisées (on confond la direction et l'angle qu'elle fait avec l'axe des abscisses).
- N_{iter} : Le nombre d'itérations nécessaires pour la convergence de l'algorithme de GSM.
- $Ec(k)$: L'écart relatif entre les deux itérés Y^k et Y^{k-1} , $Ec(k) = \frac{\|Y^k - Y^{k-1}\|}{\|Y^k\|}$
- $Er(k)$: L'écart entre l'itéré Y^k et Y_e , $Er(k) = \frac{1}{n} \|Y^k - Y_e\|$
- $E_1(\theta)$: L'écart entre l'itéré Y^1 et Z , $E_1(\theta) = \frac{1}{n} \|Y^1 - Z\|$
- $E(\rho)$: L'écart entre Y_ρ et Y_e (erreur exacte), $E(\rho) = \frac{1}{n} \|Y_\rho - Y_e\|$
- N_s : La longueur de la subdivision B-spline. Le nombre de B-splines dans chaque direction est $N_s + 2$

- $GCV(\rho)$: La fonction de validation croisée généralisée. On utilise une estimation de Monte-Carlo pour calculer la trace.
- ρ_g : Le paramètre de lissage qui réalise le minimum de GCV .
- ρ_o : Le paramètre de lissage qui réalise le minimum de Er .

5.4.1 Nombre d'itérations

figure 5.1 E_c en fonction du nombre d'itérationsfigure 5.2 E_r en fonction du nombre d'itérations

Dans la figure (5.1), on visualise l'évolution de E_c en fonction du nombre d'itérations effectuées. On vérifie ainsi le résultat théorique de convergence, sachant qu'on a obtenu exactement le même graphique avec tous les exemples qu'on a traité.

Dans la figure (5.2), on visualise l'évolution de E_r en fonction du nombre d'itérations effectuées. Ce graphique est obtenu pour une valeur de ρ proche de ρ_0 . On remarque que

Er se stabilise après un certain nombre d'itérations. On retrouve ce comportement dans tous les exemples qu'on a traités et pour différentes valeurs de ρ .

La conclusion qu'on tire de ce graphique et qu'il n'est pas nécessaire d'effectuer un grand nombre d'itérations. Au-delà d'un certain nombre d'itérations la fonction résultante ne change presque plus.

Le nombre d'itérations nécessaires pour stabiliser Er dépend de l'écart relatif Ec . On a remarqué qu'un écart relatif inférieur à 10^{-2} est généralement suffisant. On prendra alors $Niter$ le nombre d'itérations nécessaires pour que cette valeur soit atteinte et donc pour que Er se stabilise. On dira aussi que l'algorithme a convergé dès que $Ec \leq 10^{-2}$.

5.4.2 Choix des directions

Le choix des directions et de leur nombre influence, d'une façon évidente, le comportement du modèle. Un mauvais choix peut donner de très mauvais résultats.

Dans le cas où l'on sait a priori que la fonction à approcher s'écrit sous forme multidirectionnelle, il est évident qu'on obtient des résultats optimaux si on effectue un lissage multidirectionnel avec les directions exactes. Par contre si on en rajoute une autre, on augmente le nombre d'itérations nécessaires pour la convergence, mais le résultat final ne changera pas beaucoup, ceci parce que la spline monodimensionnelle obtenue dans cette direction sera presque nulle.

Pour choisir les directions de lissage, une fois le nombre fixé, on a testé l'usage de la fonction de validation croisée généralisée GCV , considérée comme une fonction du paramètre de lissage ρ et de la direction θ . Pour calculer la valeur de la fonction GCV en θ , on procède par un lissage monodimensionnel, ($M = 1$), et on se contente d'une seule itération de l'algorithme de Gausse-Seidel modifié. L'allure de la fonction GCV en fonction de θ est indépendante de ρ , pourvu que la valeur de celui ci soit assez grande.

D'après les tests effectués sur des exemples générés par différentes fonctions multidirectionnelles, on a remarqué que la fonction GCV admet des minima locaux au voisinage des directions exactes.

On a remarqué aussi que le numérateur de GCV , i.e. $\|Y_\rho - Z\| = E_1(\theta)$, suit les mêmes variations que GCV . Il serait donc possible d'utiliser la fonction GCV ou son numérateur pour estimer les directions de lissage dans le cas où la fonction à approcher est multidirectionnelle.

Si la fonction générant les données ne s'écrit pas sous forme multidirectionnelle, on choisit les directions de lissage par minimisation successive de E_1 de la façon suivante:

- Etape 1 - On pose $M = 1$ et on minimise $E_1(\theta)$ pour $\theta \in [0, \pi[$. On effectue pour chaque lissage une seule itération de GSM. On note θ_1 la direction où le minimum de E_1 est atteint
- Etape j - On a calculé à l'étape précédente $j - 1$ directions $\theta_1, \dots, \theta_{j-1}$. Le lissage SMF est fait avec j directions $\theta_1, \dots, \theta_{j-1}$ et θ . On effectue pour chaque lissage une seule itération de GSM. On pose $M = j$ et on minimise $E_1(\theta)$ pour $\theta \in [0, \pi[$. On note θ_j la direction où le minimum de E_1 est atteint
- Arrêt - On peut se fixer deux types de tests d'arrêt :
- 1/ On s'arrête dès qu'on a atteint un nombre M^{max} fixé de directions. Ce test sera utilisé lorsque l'on a des problèmes de mémoire.
 - 2/ On s'arrête dès que $\max_{\theta \in [0, \pi[} E_1(\theta) \leq \epsilon$. (ϵ fixé).

5.4.3 Choix du paramètre de lissage

On a testé l'usage de la fonction de validation croisée généralisée pour le choix du paramètre de lissage ρ . En comparant les courbes représentatives de GCV et de la fonction erreur exacte Er , pour différents types de fonctions, on remarque que les deux fonctions atteignent leur minimum en des valeurs de ρ très proches.

L'étude de l'efficacité de la GCV en ces exemples confirme cette remarque.

5.4.4 Visualisation de quelques exemples

Dans cette section on expérimente le modèle pour trois exemples de type différents. Dans chaque exemple les données sont engendrées par une fonction f_e différente. A chaque fois on ramène, par une homothétie convenable, le nuage des données exactes $(t_i, (Y_e)_i)_i$ dans le cube $([0, 1])^3$. Ces données sont alors bruitées par le même vecteur aléatoire d'écart type 0.2.

Ces exemples sont :

Exemple 1 : La fonction f_e est multidirectionnelle .

$$f_e^1(t^1, t^2) = \sum_{j=1}^3 (\sin(2\pi\tau_j))^2$$

où $\tau_j = t^1 \cos(\theta_j) + t^2 \sin(\theta_j)$ et $\theta_j = \frac{(j-1)}{3}\pi$

Exemple 2 : La fonction f_e est la somme d'une bidirectionnelle et d'une autre fonction.

$$f_e^2(t^1, t^2) = \sum_{j=1}^2 (\sin(2\pi\tau_j))^2 + e^{-10((t^1-0.5)^2 + (t^2-0.5)^2)}$$

où $\tau_j = t^1 \cos(\theta_j) + t^2 \sin(\theta_j)$, $\theta_1 = \frac{\pi}{3}$ et $\theta_2 = \frac{\pi}{2}$

Exemple 3 : La fonction f_e n'est pas multidirectionnelle, (f_e monofocale).

$$f_e^3(t^1, t^2) = t^1 \cos \left(4\pi \left((t^1 - 0.5)^2 + (t^2 - 0.5)^2 \right) \right)$$

Exemple 1 :

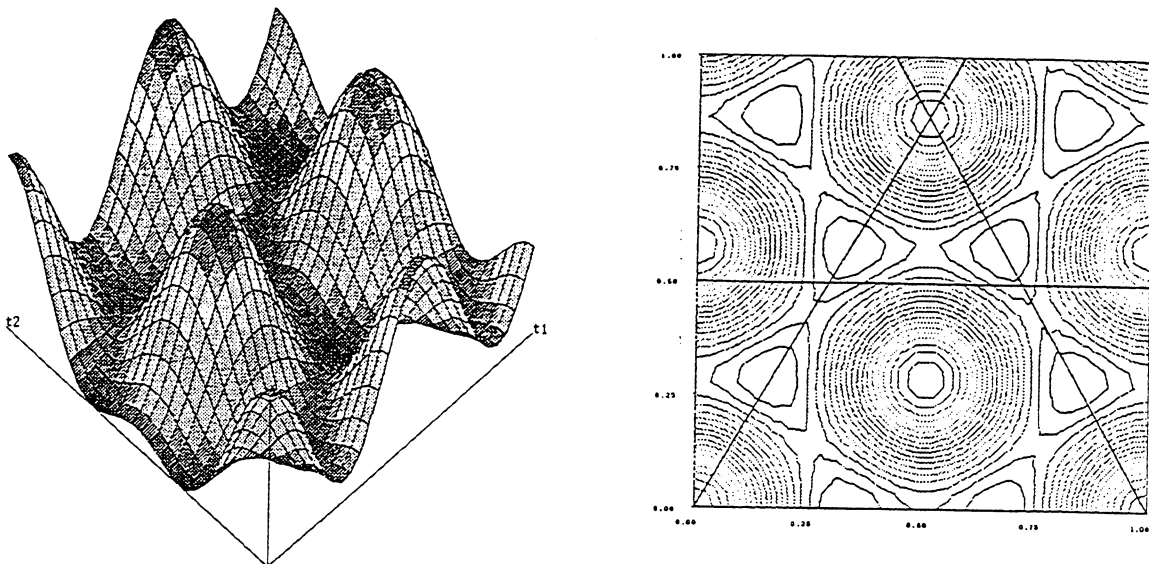


figure 5.3 Fonction exacte (à lisser)

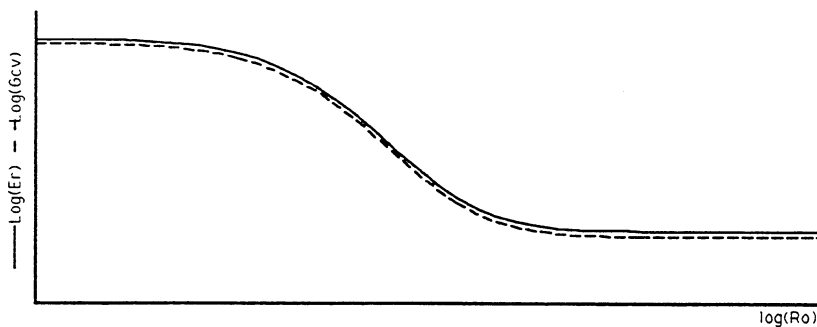


figure 5.4 Fonctions GCV et Er. $N_s = 10$ - Exemple 1

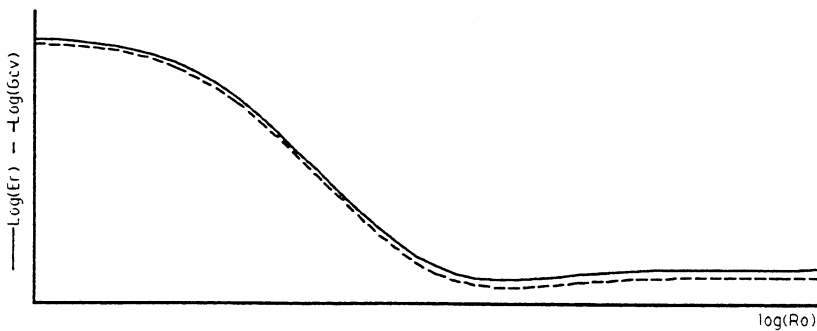


figure 5.5 Fonctions GCV et Er. $N_s = 20$ - Exemple 1

Les fonctions GCV et Er atteignent leur minimum au voisinage de la même valeur et l'efficacité de la méthode de validation croisée généralisée pour le choix d'un ρ optimal est:

Pour $N_s = 10$ efficacité = 0,9966.

Pour $N_s = 20$ efficacité = 0,9975.

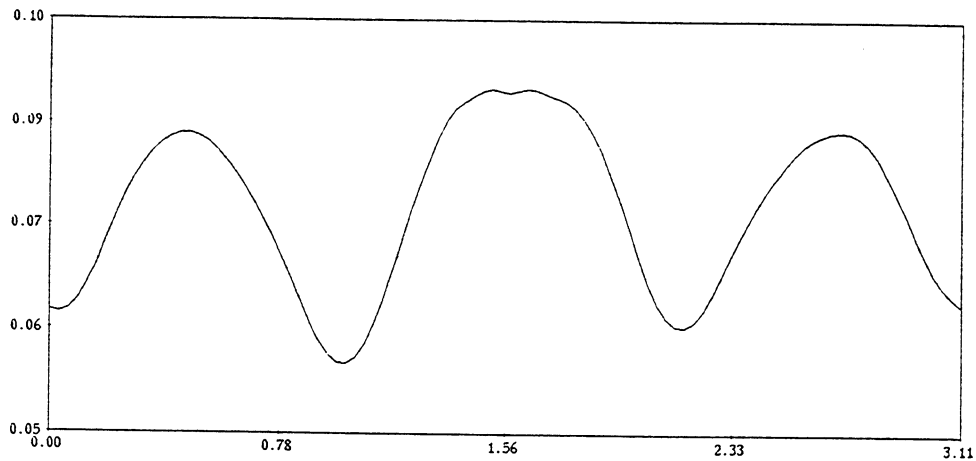


figure 5.6 Courbe de la fonction E_1 .

$\rho = 10^3$ et $\theta \in [0, \pi[$ - Exemple 1

D'après la courbe de E_1 ci-dessus, on voit bien que celle-ci admet des minima aux valeurs exactes des directions, c'est à dire en $0, \frac{\pi}{3}$ et $\frac{2\pi}{3}$.

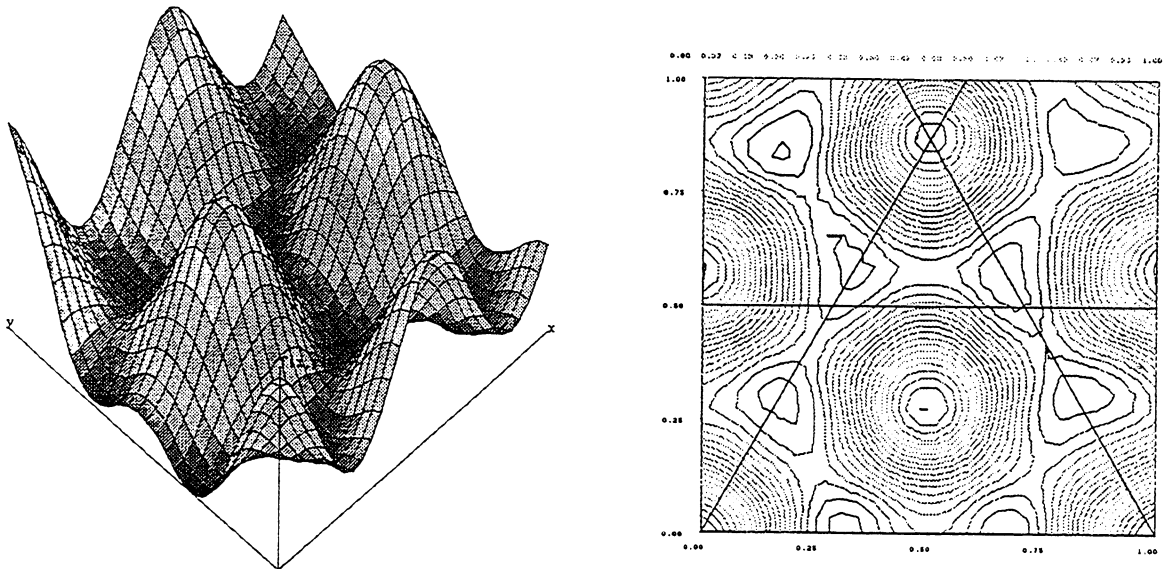


figure 5.7 Fonction lissée en utilisant les directions exactes.

$\rho = \rho_y$ et $N_s = 10$ - Exemple 1

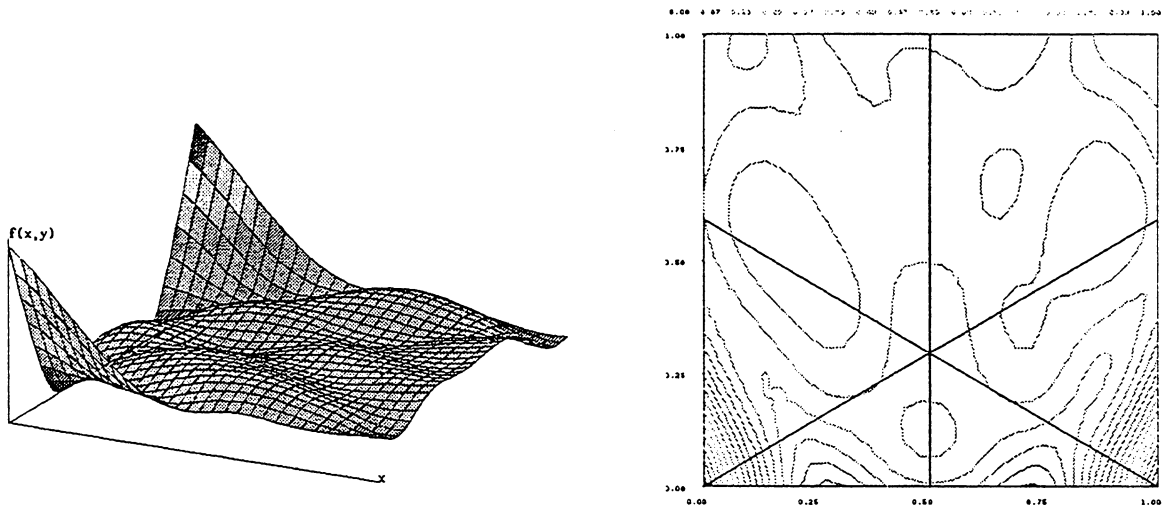


figure 5.8 Fonction lissée en utilisant de mauvaises directions ,
 $\rho = \rho_y$ et $Ns = 10$ - Exemple 1

Exemple 2 :

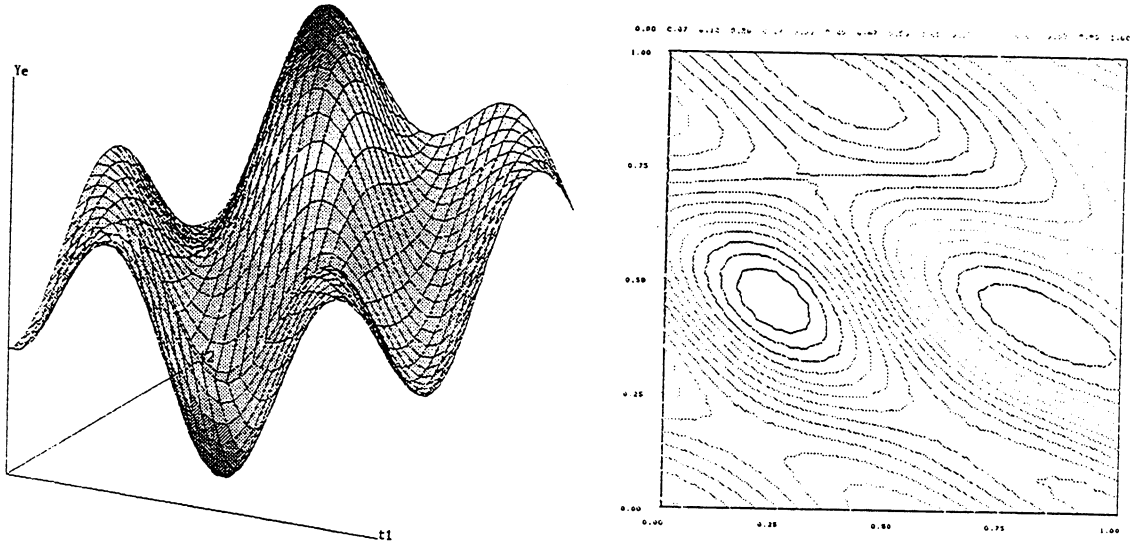


figure 5.9 Fonction exacte (à lisser)

L'efficacité de la méthode de validation croisée généralisée est :

Pour $N_s = 10$ efficacité = 0,9968.

Pour $N_s = 20$ efficacité = 0,9970.

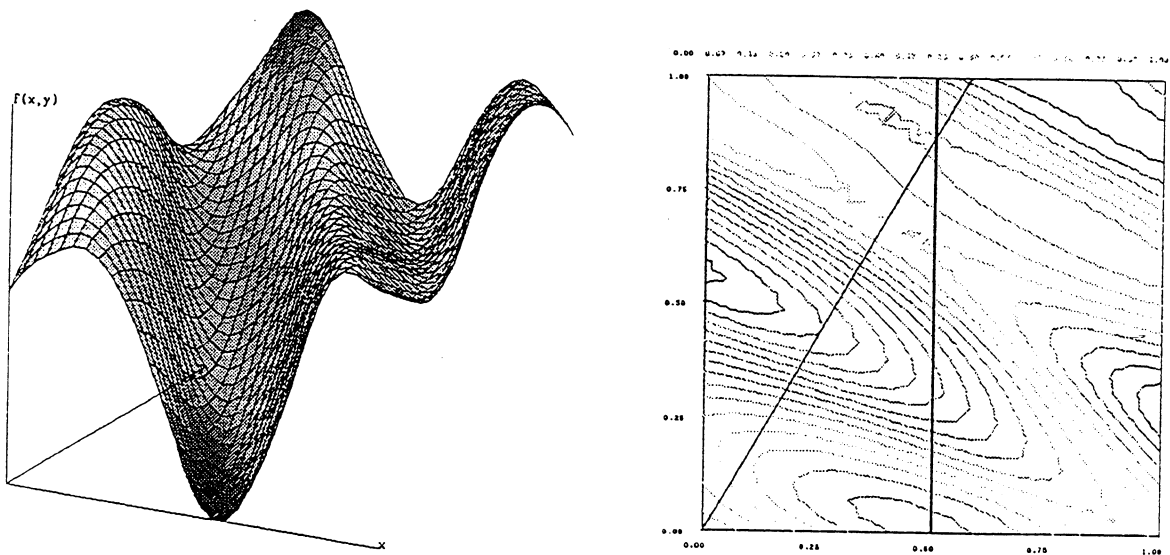


figure 5.10 Fonction lissée en utilisant les deux directions exactes,

$\rho = \rho_g, N_s = 10$ - Exemple 2

On a voulu voir ce que donnerait un lissage SMD avec des directions minimisant E_1 , en ignorant les directions exactes. On a obtenu les résultats suivants:

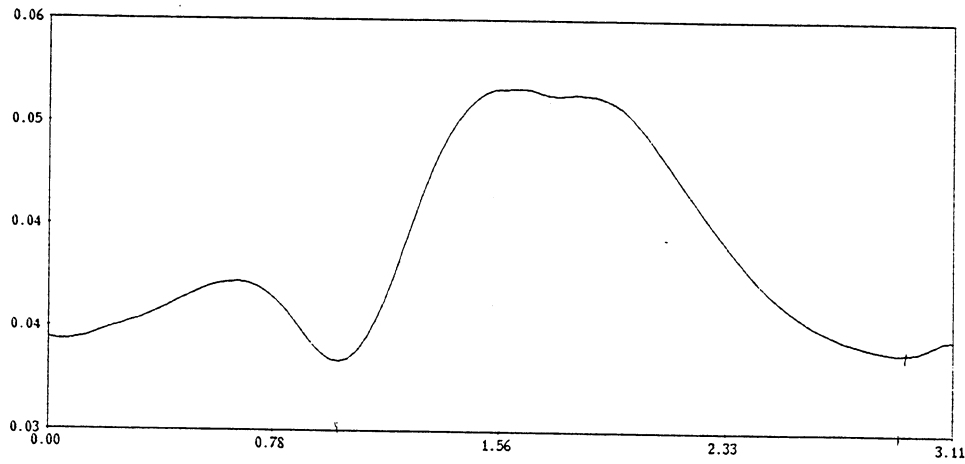


figure 5.11 Courbe de la fonction E_1 .

$$\rho = 10^3 \text{ et } \theta \in [0, \pi[- \text{Exemple 2}$$

On remarque que les deux directions où sont atteints les minima de E_1 sont différentes des directions exactes. Un lissage effectué avec ces deux directions donne une fonction lissée plus proche de celle de la fonction exacte que ce que l'on obtient avec les directions exactes.

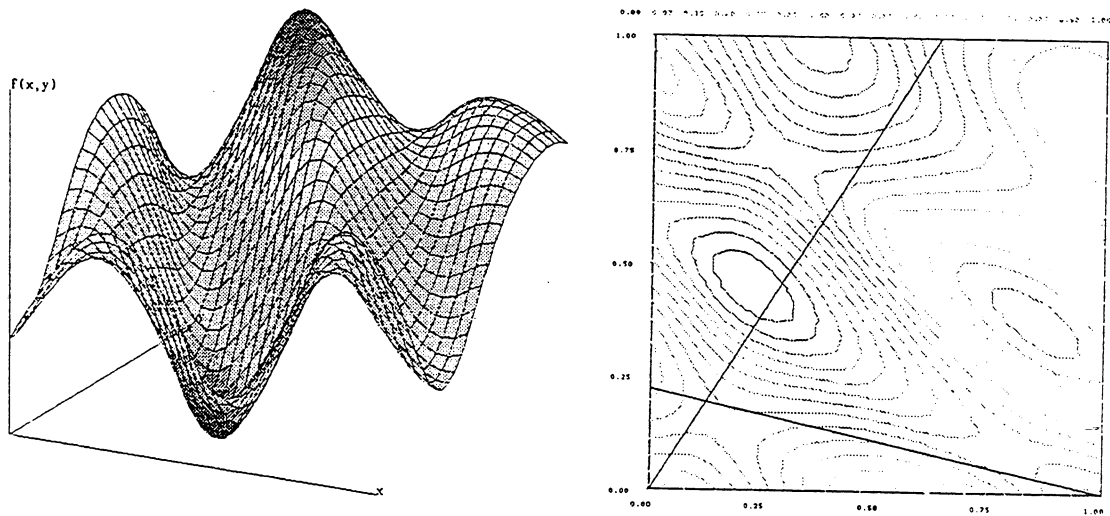


figure 5.12 Fonction lissée en utilisant les deux directions calculées par minimisation successive de E_1 . $\rho = \rho_g$. $N_s = 10$ - Exemple 2

Le résultat nous paraît naturel. En effet, quand on effectue un lissage par SMD, avec comme directions les deux directions exactes $\frac{\pi}{2}$ et $\frac{\pi}{3}$, on reconstitue uniquement la partie bidirectionnelle des données. Par contre le pic centré en $(0.5, 0.5)$ est complètement ignoré. En lissant avec les deux directions minimisant E_1 , ce pic est en partie reconstitué.

Exemple 3 :

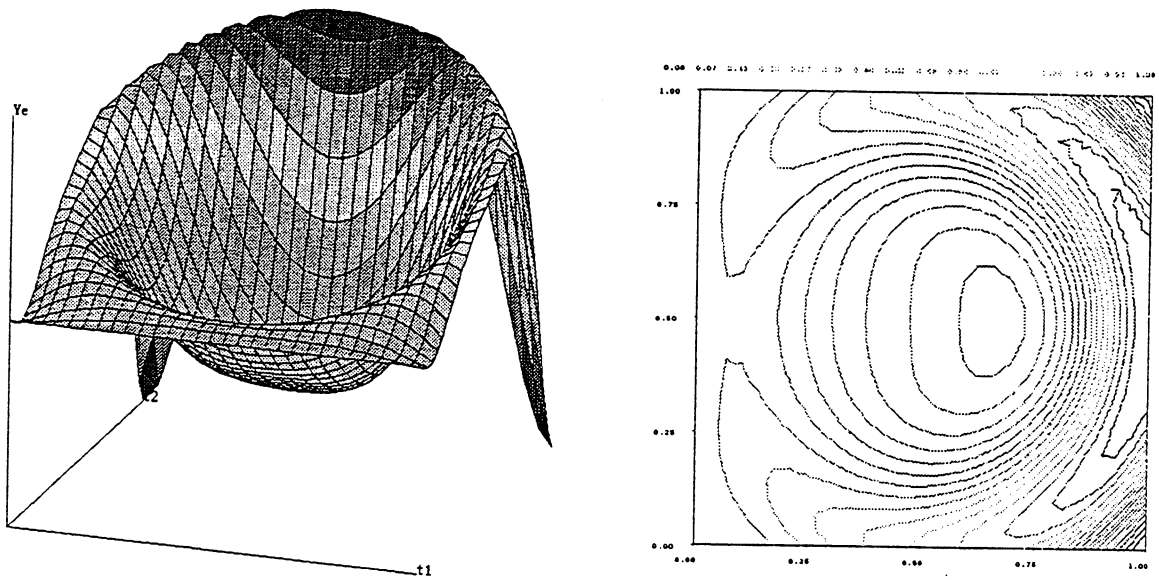
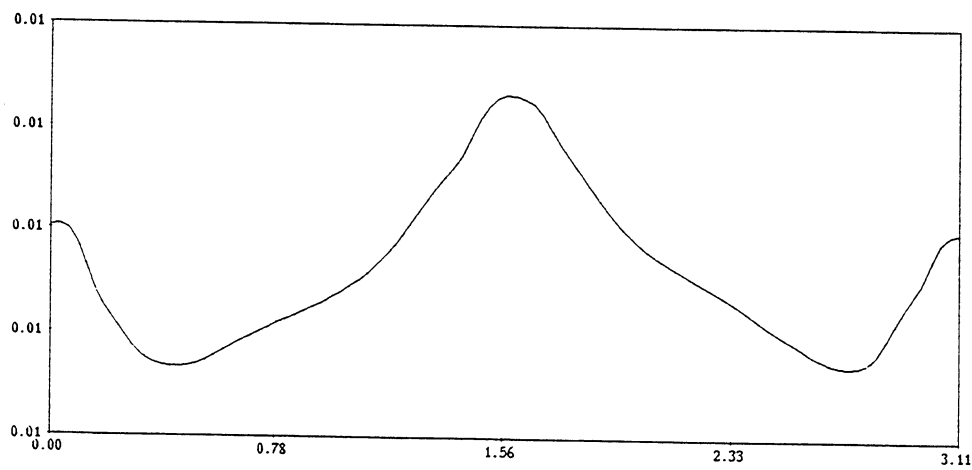


figure 5.13 Fonction exacte (à lisser)

Efficacité :

 $N_s = 10$ efficacité = 0,997 $N_s = 20$ efficacité = 0,999figure 5.14 Fonction E_1 . θ dans $[0, \pi]$.

Exemple 3

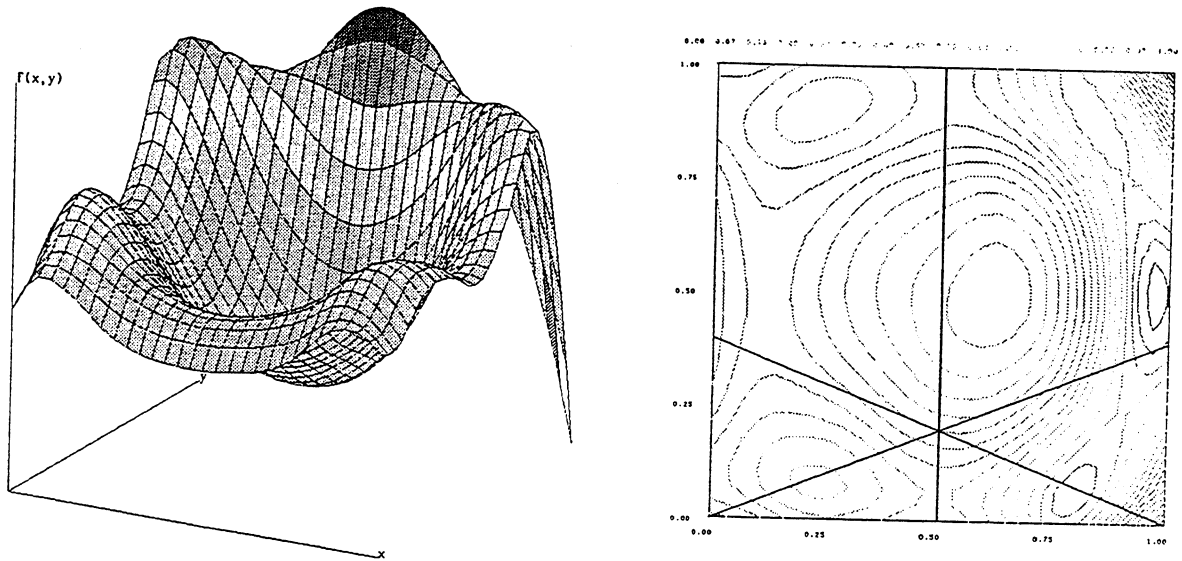


figure 5.15 Fonction lissée en utilisant les trois directions calculées par minimisation successive de E_1 suivant la procédure décrite précédemment, $\rho = \rho_g$ et $N_s = 20$ - Exemple 3

5.5 Remarques

- Combinaison des deux modèles SMD et SMF:

On peut combiner les deux modèles SMD et SMF de la façon suivante: On se donne M_D directions d'angle $\theta_1, \dots, \theta_{M_D}$ et M_F focales $\omega_1, \dots, \omega_{M_F}$, et on cherche une spline de lissage σ_s de la forme:

$$\sigma_s(t) = \sum_{j=1}^{M_D} f_j^D(\langle t, D_j \rangle) + \sum_{j=1}^{M_F} f_j^F(\|t - \omega_j\|), \quad D_j = (\cos\theta_j, \sin\theta_j)$$

f_j^D et f_j^F , pour tout j , sont des fonctions à une seule variable.

La résolution du problème de lissage se fera de la même façon que précédemment.

- Extension en dimension supérieure à 2:

On peut utiliser le modèle pour le lissage de fonctions à d variables, ($d \geq 2$). On recherchera alors, la spline solution sous la forme:

$$\sigma_s(t) = \sum_{j=1}^M f_j(\langle t, D_j \rangle) \quad t \in \mathbb{R}^d, \quad D_j \in \mathbb{R}^d \text{ et } \|D_j\| = 1$$

Chaque f_j étant une fonction à une seule variable.

5.6 Conclusion

On a construit un modèle de lissage qui permet de ramener le problème de lissage en dimension 2, ou plus généralement en dimension $d \geq 2$, à une suite de problèmes de lissage à une dimension, faciles à résoudre. Ceci nous permet un grand gain en temps de calcul et en place mémoire. Le fait que l'on puisse choisir les directions de lissage par minimisation de la fonction E_1 , rend ce modèle très efficace.

On pense que l'utilisation de ce modèle peut s'étendre facilement en dimension supérieure à 3 sans hypothèses supplémentaires.

Deuxième Partie

Conception d'un verre progressif

Chapitre 6

Introduction

6.1 Courbures normales

Soit \mathcal{S} une surface admettant une paramétrisation cartésienne $(x, y, f(x, y))$, f étant une fonction au moins 2 fois continûment différentiable et soit \mathcal{M} un point de \mathcal{S} .

On notera $T_{\mathcal{M}}$ le plan vectoriel tangent à \mathcal{S} en \mathcal{M} . Soit \vec{n} le vecteur unitaire normal à \mathcal{S} en \mathcal{M} . Une base de $T_{\mathcal{M}}$ est formée par les vecteurs $\frac{\partial \mathcal{M}}{\partial x}$ et $\frac{\partial \mathcal{M}}{\partial y}$ (que l'on suppose non nuls).

Définition : Section normale *Soit \vec{v} un vecteur tangent à \mathcal{S} en \mathcal{M} . Le plan normal à \mathcal{S} contenant \vec{v} coupe la surface \mathcal{S} suivant une courbe admettant en \mathcal{M} \vec{v} comme vecteur tangent. On appelle cette courbe la section normale de \mathcal{S} définie par le vecteur \vec{v}*

Définition : Courbure normale *On appelle courbure normale de \mathcal{S} en \mathcal{M} suivant la direction \vec{v} , la courbure de la section normale de \mathcal{S} définie par le vecteur \vec{v} . On la notera $c_{\mathcal{M}}^{\vec{v}}$*

Notations :

Dans le but de simplifier l'écriture des formules qui suivent nous adoptons les notations (dites de Gauss) suivantes :

$$E = \left\| \frac{\partial \mathcal{M}}{\partial x} \right\|^2 = 1 + f_x'^2$$

$$F = \frac{\partial \mathcal{M}}{\partial x} \cdot \frac{\partial \mathcal{M}}{\partial y} = f_x' f_y'$$

$$G = \left\| \frac{\partial \mathcal{M}}{\partial y} \right\|^2 = 1 + f_y'^2$$

$$H = \left\| \frac{\partial \mathcal{M}}{\partial x} \wedge \frac{\partial \mathcal{M}}{\partial y} \right\| = \sqrt{EG - F^2} = \sqrt{1 + f_x'^2 + f_y'^2}$$

$$L = \frac{1}{H} \det \left[\frac{\partial \mathcal{M}}{\partial x}, \frac{\partial \mathcal{M}}{\partial y}, \frac{\partial^2 \mathcal{M}}{\partial x^2} \right] = \frac{f_{xx}''}{H}$$

$$M = \frac{1}{H} \det \left[\frac{\partial \mathcal{M}}{\partial x}, \frac{\partial \mathcal{M}}{\partial y}, \frac{\partial^2 \mathcal{M}}{\partial x \partial y} \right] = \frac{f_{xy}''}{H}$$

$$N = \frac{1}{H} \det \left[\frac{\partial \mathcal{M}}{\partial x}, \frac{\partial \mathcal{M}}{\partial y}, \frac{\partial^2 \mathcal{M}}{\partial y^2} \right] = \frac{f_{yy}''}{H}$$

Notons que les quantités E, F, G, H, L, M et N sont des fonctions du point \mathcal{M}

Calcul de la courbure normale :

Soit $\vec{v} = \lambda \frac{\partial \mathcal{M}}{\partial x} + \nu \frac{\partial \mathcal{M}}{\partial y}$ un vecteur non nul ($(\lambda, \nu) \neq 0$) de $T_{\mathcal{M}}$. La courbure normale $c_{\mathcal{M}}^{\vec{v}}$ de \mathcal{S} au point \mathcal{M} suivant la direction \vec{v} est égale à :

$$c_{\mathcal{M}}^{\vec{v}} = \frac{L\lambda^2 + 2M\lambda\nu + N\nu^2}{E\lambda^2 + 2F\lambda\nu + G\nu^2}$$

Pour plus de détails voir [28]

Exemple Courbures normales dans les directions des axes :

La courbure normale de \mathcal{S} au point \mathcal{M} est :

- suivant la direction de l'axe des x (courbure horizontale) égale à :

$$c_{\mathcal{M}}^h = \frac{L}{E} = \frac{f''_{xx}}{H(1 + f_x'^2)}$$

- suivant la direction de l'axe des y (courbure verticale) égale à :

$$c_{\mathcal{M}}^v = \frac{N}{G} = \frac{f''_{yy}}{H(1 + f_y'^2)}$$

6.2 Courbures et directions principales

Les courbures normales de \mathcal{S} au point \mathcal{M} sont fonctions, du point \mathcal{M} bien sûr, mais aussi en général de la direction dans laquelle on les calcule. On a le résultat fondamental suivant :

Théorème d'Euler : *Si la courbure n'est pas la même dans toutes les directions, alors il existe une seule direction de courbure maximale, une seule direction de courbure minimale et ces deux directions sont orthogonales.*

Définition : *On appelle courbures principales de \mathcal{S} au point \mathcal{M} les courbures maximale et minimale de \mathcal{S} au point \mathcal{M} et les directions principales les directions où ces courbures sont atteintes.*

$\vec{v} \in T_{\mathcal{M}}$ (non nul) est une direction principale de \mathcal{S} au point \mathcal{M} si et seulement si

$$c_{\mathcal{M}}^{\vec{v}} = \min_{\vec{u} \in T_{\mathcal{M}}} c_{\mathcal{M}}^{\vec{u}} = c_{\mathcal{M}}^{\min} \quad \text{ou} \quad c_{\mathcal{M}}^{\vec{v}} = \max_{\vec{u} \in T_{\mathcal{M}}} c_{\mathcal{M}}^{\vec{u}} = c_{\mathcal{M}}^{\max}$$

Calcul des directions et courbures principales :

Pour la suite, sauf s'il y a risque de confusion, on notera c la courbure normale de \mathcal{S} en \mathcal{M} suivant la direction $\lambda \frac{\partial \mathcal{M}}{\partial x} + \nu \frac{\partial \mathcal{M}}{\partial y}$, et on la considérera comme une fonction de (λ, ν) .

Pour calculer les directions et courbures principales on est donc amené à minimiser ou maximiser sur \mathbb{R}^2 la fonction c . En écrivant les conditions d'optimalité : $\frac{\partial c}{\partial \lambda} = \frac{\partial c}{\partial \nu} = 0$ on obtient le système

$$\begin{cases} L\lambda + M\nu = (E\lambda + F\nu) c \\ M\lambda + N\nu = (F\lambda + G\nu) c \end{cases} \quad (P)$$

Pour que les deux équations de (P) soient compatibles il faut que :

$$\det \begin{vmatrix} L\lambda + M\nu & E\lambda + F\nu \\ M\lambda + N\nu & F\lambda + G\nu \end{vmatrix} = 0$$

d'où l'équation aux directions principales :

$$(FL - EM) \lambda^2 + (GL - EN) \lambda\nu + (GM - FN) \nu^2 = 0 \quad (EDP)$$

D'autre part, (P) est un système homogène en (λ, ν) qui doit admettre une solution non nulle, donc :

$$\det \begin{vmatrix} L - cE & M - cF \\ M - cF & N - cG \end{vmatrix} = 0$$

d'où l'équation aux courbures principales:

$$(EG - F^2) c^2 - (LG + EN - 2FM) c + (LN - M^2) = 0 \quad (ECP)$$

On vérifie aisément que le discriminant de cette équation est toujours positif ou nul.

En résolvant (EDP) on calcule les directions principales, et en résolvant (ECP) on obtient les courbures principales correspondantes.

6.3 Quelques cas intéressants

Les deux situations qui vont nous intéresser pour la suite de ce travail sont :

- 1) le cas où la surface est localement sphérique en un point,
- 2) le cas où au moins l'une des directions principales en un point est celle d'un des axes de coordonnées.

On dira par abus de langage que l'axe des x (respectivement l'axe des y) est une direction principale si l'intersection du plan horizontal Oxz (respectivement du plan vertical Oyz) avec le plan vectoriel tangent $T_{\mathcal{M}}$ est une direction principale.

Définition : Point ombilical *Un point \mathcal{M} est dit ombilical (ou un ombilic) si et seulement si toute direction de $T_{\mathcal{M}}$ est principale.*

Caractérisation d'un ombilic :

Pour qu'un point soit ombilical il faut et il suffit qu'il existe un nombre c vérifiant :

$$L = cE, \quad M = cF \quad \text{et} \quad N = cG$$

Ce nombre c est alors la courbure normale de S en \mathcal{M} dans une direction tangente quelconque.

Preuve

\mathcal{M} est ombilical si et seulement si l'ensemble des solutions de (EDP) est \mathbb{R}^2 , c'est à dire si et seulement si :

$$FL = EM, \quad GL = EN \quad \text{et} \quad GM = FN$$

ce qui est équivalent à :

$$M = \frac{L}{E}F, \quad N = \frac{L}{E}G \quad \text{et} \quad M = \frac{N}{G}F.$$

C'est à dire:

$$\frac{L}{E} = \frac{N}{G} = \frac{M}{F} = c.$$

Montrons que c est la courbure normale de f dans n'importe quelle direction de $T_{\mathcal{M}}$.

Soit $\vec{v} = \lambda \frac{\partial \mathcal{M}}{\partial x} + \nu \frac{\partial \mathcal{M}}{\partial y}$ un vecteur non nul de $T_{\mathcal{M}}$

$$\begin{aligned} c_{\vec{v}} &= \frac{L\lambda^2 + 2M\lambda\nu + N\nu^2}{E\lambda^2 + 2F\lambda\nu + G\nu^2} \\ &= \frac{cE\lambda^2 + 2cF\lambda\nu + cG\nu^2}{E\lambda^2 + 2F\lambda\nu + G\nu^2} \\ &= c \end{aligned}$$

c est donc bien la courbure normale de f dans une direction quelconque du plan tangent. \square

Dans la suite on caractérisera un point ombilical par :

$$\begin{aligned} f''_{xx}f'_xf'_y &= f''_{xy}(1 + f_x'^2) \\ f''_{xx}(1 + f_y'^2) &= f''_{yy}(1 + f_x'^2) \\ f''_{yy}f'_xf'_y &= f''_{xy}(1 + f_y'^2) \end{aligned} \tag{6.15}$$

Condition pour que l'axe des x soit une direction principale : :

Pour que la direction de l'axe des x soit une direction principale en un point il faut et il suffit qu'en ce point :

$$f''_{xx}f'_xf'_y = f''_{xy}(1 + f_x'^2) \quad (6.16)$$

Preuve

On écrit que les couples de la forme $(\lambda, 0)$, $\lambda \in \mathbb{R}$ sont solutions de (EDP), ce qui revient à dire que FL - EM = 0.

Condition pour que l'axe des y soit une direction principale : :

Pour que la direction de l'axe des y soit une direction principale en un point il faut et il suffit qu'en ce point :

$$f''_{yy}f'_xf'_y = f''_{xy}(1 + f_y'^2) \quad (6.17)$$

Preuve

On écrit que les couples de la forme $(0, \nu)$, $\nu \in \mathbb{R}$ sont solutions de (EDP), ce qui revient à dire que GM - FN = 0.

6.4 Éléments d'optométrie

On compare souvent l'œil à un appareil photographique. À l'émulsion sur laquelle doivent être formées des images nettes correspond la rétine où se trouvent engendrées des messages transmis au cerveau par le nerf optique. Le rôle du diaphragme est joué par l'iris, offrant à la lumière une ouverture variable, la pupille. Tandis qu'à l'objectif correspondent non seulement une sorte de lentille, le cristallin, mais encore tout le reste de l'intérieur de l'œil, le vitré, rempli d'un liquide analogue à de l'eau salée. La face d'entrée, la cornée, qui est bombée, joue aussi un rôle important dans la formation des images.

L'œil est à peu près à symétrie de révolution, la direction du regard étant proche de son axe. Une petite région de la rétine voisine de cet axe, la fovea, est particulièrement apte à la perception des détails.

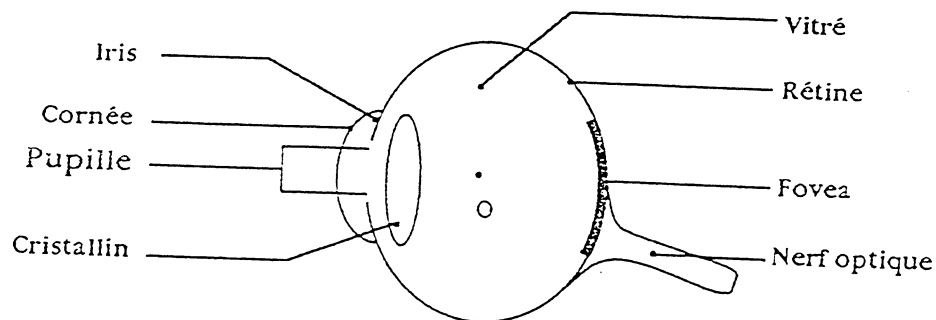


figure 6.1 l'œil humain

Si l'image A' d'un point A est située en avant ou en arrière de la fovea, le faisceau correspondant détermine sur celle-ci une tache. Cette tache est d'autant plus grande que l'image est plus éloignée de la rétine et que la pupille est plus ouverte. Une mise au point est alors nécessaire. Entre deux positions limites de A , cette mise au point est rendu possible par le mécanisme d'accommodation, qui est une augmentation de la puissance de l'œil. Elle résulte d'une modification, inconsciente, du cristallin.

Un œil est dit emmétrope lorsque, sans accommodation, il perçoit une image nette d'un point à l'infini. Sinon il est amétrope: il peut alors être myope ou hypermétrope et/ou astigmat.

Le point R , le plus éloigné dont la vision nette soit possible sans accommodation, est appelé remotum. Le point P le plus rapproché qui puisse être vu nettement en accommodation est appelé proximum.

Soit O la position de l'œil. L'amplitude d'accommodation A est par définition :

$$A = \frac{1}{OR} - \frac{1}{OP}$$

A est positive si l'on convient de compter les vecteurs positivement dans le sens de la lumière \overline{RPO} . Elle s'exprime en dioptries si les longueurs sont évaluées en mètres, et se réduit pour un emmétrope à $-\frac{1}{OP}$.

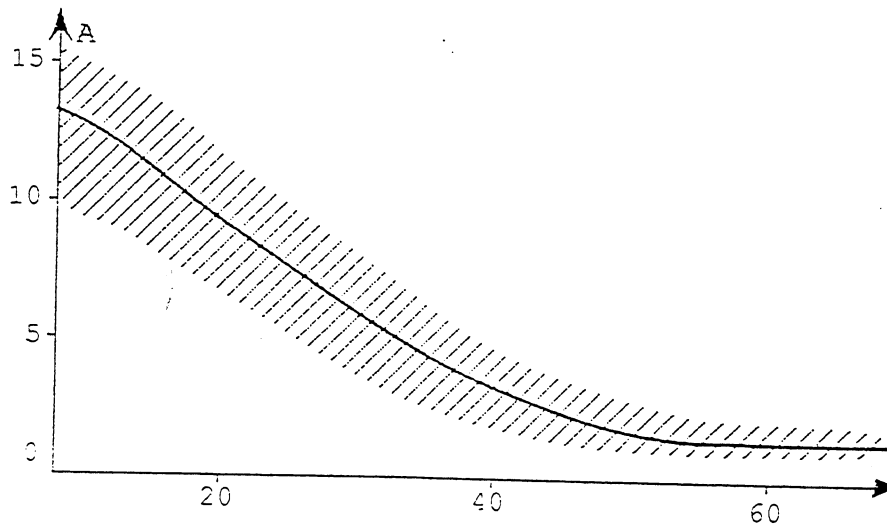


figure 6.2 Variation de l'amplitude d'accommodation avec l'âge

L'amplitude d'accommodation diminue progressivement à mesure que l'on vieillit. Même pour un œil normal, elle subit constamment de petites fluctuations. Ce phénomène, qui s'appelle la presbytie, intervient chez tous les humains, emmétropes ou amétropes, en diminuant progressivement leur faculté d'accommodation à mesure qu'ils vieillissent. L'amplitude A décroît donc en fonction de l'âge (figure (6.2)). La presbytie devient gênante à partir du moment où l'on ne peut plus voir nettement de près.

On améliore la vision à l'aide de verres, dits correcteurs, qui sont en fait des compensateurs de l'excès ΔP , positif ou négatif, de la puissance oculaire P .

Les lunettes pour voir de près diffèrent de celles pour voir de loin par une puissance plus grande en valeur algébrique. Pour éviter de changer trop souvent de lunettes, on a recours, pour la presbytie, à des verres comportant à la partie supérieure une plage pour vision lointaine et à la partie inférieure une plage pour vision rapprochée, verre bifocaux, (figure (6.3) a). Il est nécessaire que chacune des plages soit correctement corrigée pour l'astigmatisme oblique. De plus, lorsqu'on passe d'une plage à l'autre la direction du regard ne doit pas subir une trop brusque déviation.

On ne peut pas obtenir à la fois, ni pour toutes les directions du regard, ces diverses qualités.

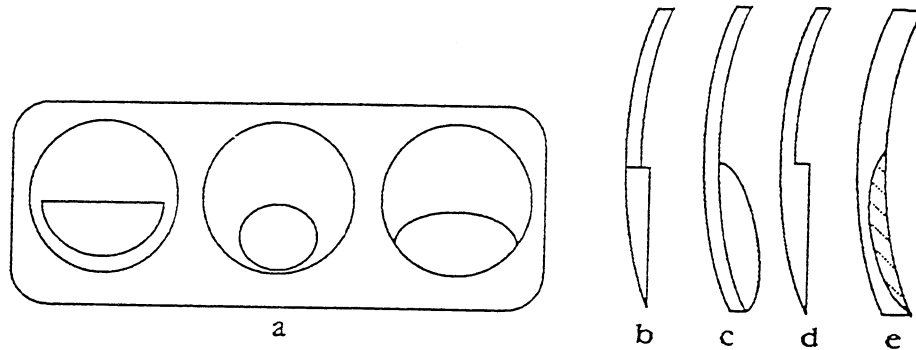


figure 6.3 Types de verres bifocaux

La solution la plus simple, dite à la Franklin (figure (6.3) b), consiste en la juxtaposition de deux demi-verres. Elle donne une ligne de séparation très visible et un assemblage fragile, elle n'est plus utilisée. Pour les emmétropes presbytes on utilise encore des verres, demi lune. La superposition par collage de deux lentilles ayant sur leurs faces adjacentes même rayon de courbure, (figure (6.3) c), est économique mais présente la plupart des défauts signalés ci-dessus. On emploie plus souvent des verres taillés dans la même pièce (figure (6.3) d) ou des pièces d'indices différents fusionnées à chaud (figure (6.3) e).

Des verres trifocaux sont utilisés quand une trop faible amplitude d'accommodation interdit, avec des bifocaux, une vision nette entre les objets très éloignés, d'une part, et les objets très rapprochés, d'autre part. Ces verres comportent, en plus des deux plages précédentes, une plage de puissance intermédiaire.

Une réalisation récente et remarquable est celle des verres progressifs ou varifocaux. Ces verres ont une des faces à courbure variant de façon continue entre la partie supérieure et la partie inférieure. Ce qui implique une variation continue entre les valeurs correspondant à la vision lointaine et à la vision rapprochée. Très satisfaisante au voisinage du plan vertical médian, la qualité des images est parfois moins bonne quand on s'écarte de ce plan.

Dans cette partie nous proposons une méthode de conception d'un verre progressif. Le verre sera représenté par une fonction σ . La fonction σ sera définie sur un domaine rectangulaire Ω subdivisé en zones suivant les conditions définies ci-dessus.

6.5 Modélisation

Dans le plan \mathbb{R}^2 , on considère le domaine $\Omega = [-40, 40] \times [-40, 40]$ qu'on subdivise en 8 zones : H, HP, I, B, BP, T, N et L, de la façon suivante.

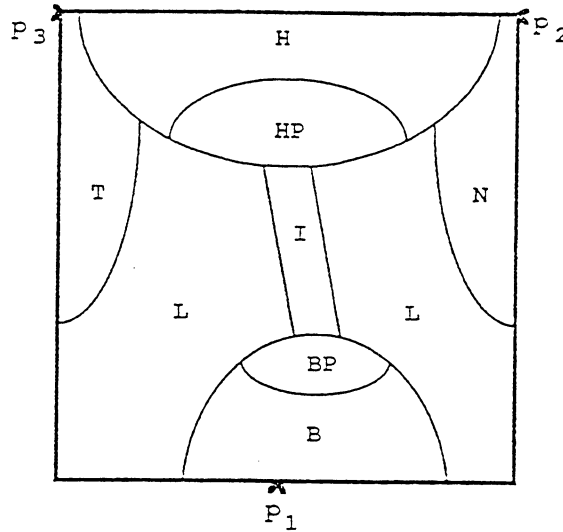


figure 6.4 Une subdivision de Ω en zones

Les zones H (haut) et HP (haut principal) correspondent à la vision de loin, les zones B (bas) et BP (bas principal) correspondent à la vision de près, la zone étroite I (intermédiaire) est celle où l'on passe de façon progressive de la vision de loin à la vision de près. Le reste du domaine est partagé en zone T (temporale), N (nasale) et enfin L (latérale).

On cherche une fonction $\sigma : \Omega \rightarrow \mathbb{R}$ à laquelle on impose, dans chaque zone, une condition sur ses courbures ou ses directions principales de la façon suivante :

- Dans $HP \cup H$: La fonction est approximativement sphérique de courbure c_1 .
- Dans $BP \cup B$: La fonction est approximativement sphérique de courbure c_2 .
- Dans I : Les courbures principales de σ doivent être assez rapprochées (sphéricité) et passer de c_1 à c_2 d'une façon assez lisse et monotone.
- Dans T et N : L'une des directions principales de σ est proche d'un axe de coordonnées.

On notera ces conditions (Cv).

On se propose de déterminer σ sous forme de produit tensoriel de B-splines de degré 4 :

$$f(x, y) = \sum_{i=-4}^{N_x-1} \sum_{j=-4}^{N_y-1} (\alpha_{ij} B_i(x) B_j(y))$$

On se donne, sur Ω , un nuage de points $(x_i, y_i)_{i=1}^{N_d}$. Ces points peuvent être uniformément répartis ou non, mais doivent couvrir toutes les zones de Ω .

Définition : fonction-poids *Une fonction-poids sur Ω est une fonction à valeur positive ou nulle, définie sur Ω , qui est constante sur chacune des 8 zones de Ω .*

Notations

Pour alléger les formules qui suivront, on notera g_i la valeur $g(x_i, y_i)$ si g est une fonction définie sur Ω .

Chapitre 7

Quelques opérateurs

Soit f une fonction définie sur Ω , au moins de classe \mathcal{C}^2

7.1 Opérateur d'approximation des courbures

D'après l'expression des courbures horizontales de la fonction f en (x, y) :

$$c_h(x, y) = \frac{L(x, y)}{E(x, y)}$$

on définit un opérateur d'approximation des courbures horizontales de la façon suivante :

$$\mathcal{A}_c^h(f) = \sum_{i=1}^{Nd} \omega_h(x_i, y_i) (c_h(x_i, y_i) - c(x_i, y_i))^2$$

ω_h étant une fonction-poids.

Dans la pratique la fonction courbure c n'est imposée que dans les zones $HP \cup H$ et $BP \cup B$. Dans les autres zones on lui donnera une valeur arbitraire (0 par exemple), mais on prendra une fonction-poids égale à 0 dans ces zones.

De la même façon, d'après l'expression des courbures verticales de f en (x, y) :

$$c_v(x, y) = \frac{N(x, y)}{G(x, y)}$$

on définit un opérateur d'approximation des courbures verticales :

$$\mathcal{A}_c^v(f) = \sum_{i=1}^{Nd} \omega_v(x_i, y_i) (c_v(x_i, y_i) - c(x_i, y_i))^2$$

On a ainsi construit un opérateur d'approximation des courbures horizontales et verticales de f par :

$$\mathcal{A}_c(f) = \mathcal{A}_c^h(f) + \mathcal{A}_c^v(f)$$

7.2 Opérateur de sphéricité

Une fonction f est localement sphérique en (x,y) si et seulement si en ce point

$$M(x, y)E(x, y) = L(x, y)F(x, y)$$

$$N(x, y)E(x, y) = L(x, y)G(x, y)$$

$$M(x, y)G(x, y) = N(x, y)F(x, y)$$

On pourra donc définir un opérateur de sphéricité locale par :

$$\mathcal{S}(f) = \sum_{i=1}^{Nd} \omega_s(x_i, y_i) \left[(M_i E_i - L_i F_i)^2 + (N_i E_i - L_i G_i)^2 + (M_i G_i - N_i F_i)^2 \right]$$

ω_s étant une fonction-poids dont le rôle est de doser l'importance relative qu'on veut donner à la sphéricité de f dans chacune des δ zones de Ω .

7.3 Opérateur sur les directions principales

Une condition nécessaire et suffisante pour que l'une des directions principales de f en (x, y) soit celle d'un axe de coordonnées (l'axe des y par exemple) est que :

$$G(x, y)N(x, y) = F(x, y)N(x, y)$$

On en déduit un opérateur aux directions principales \mathcal{D} .

$$\mathcal{D}(f) = \sum_{i=1}^{Nd} \omega_d(x_i, y_i) (G_i N_i - F_i N_i)^2$$

ω_d étant une fonction-poids dont le rôle est de déterminer l'importance qu'on veut donner au rapprochement d'une des directions principales de f de l'axe des y .

Dans la pratique cette condition n'est imposée que dans les zones T et Z . On prendra, donc, une fonction-poids qui s'annule dans les autres zones.

7.4 Opérateur de lissage

On veut que f soit lisse sur Ω , on introduit, alors, un opérateur de lissage \mathcal{J} sur Ω .

$$\mathcal{J}(f) = \rho \sum_{i=1}^{Nd} \left[\frac{\partial^8 f}{\partial x^4 \partial y^4}(x_i, y_i) \right]^2$$

ρ étant un paramètre de lissage dont le rôle est de conditionner l'aspect lisse de f et aussi de régulariser le système linéaire. On note que le choix du pas des B-splines est un paramètre qui agit dans le même sens.

7.5 Opérateur de bord

Les conditions sur la fonction cherchée portant sur les dérivées secondes (courbures et directions principales), la solution n'est déterminée qu'à un polynôme de degré 1 près. Pour la rendre unique, et faire en sorte que les valeurs de celle-ci, en au moins trois points non alignés du bord de Ω , $p_i = (c_i, d_i)$ $i = 1 \dots 3$, soient aussi proches de 0 que possible, on rajoute un opérateur de bord \mathcal{B} , de la forme suivante :

$$\mathcal{B}(f) = \mu \sum_{i=1}^3 (f(c_i, d_i))^2$$

où μ est un paramètre de pénalité très grand par rapport aux autres paramètres. En pratique on a pris $p_1 = (0, 40)$, $p_2 = (40, -40)$ et $p_3 = (-40, -40)$.

7.6 Opérateur de tension

L'une des conditions implicitement imposée sur la solution recherchée est que ses courbures ne présentent pas d'oscillations. La méthode idéale pour éviter ces oscillations est d'imposer un terme de tension sur ces courbures, par une minimisation de la dérivée de ces dernières. Dans notre cas, ceci n'est pas possible (cf Remarque 8.3). On propose, alors, d'imposer une tension aux dérivées secondes : f''_{x^2} , f''_{xy} et f''_{y^2} en minimisant la norme des dérivées troisièmes :

$$\left(f_{x^3}^{(3)}, f_{x^2y}^{(3)}, f_{xy^2}^{(3)}, f_{y^3}^{(3)} \right)$$

On en déduit un opérateur de tension \mathcal{T} défini par :

$$\mathcal{T}(f) = \sum_{i=1}^{Nd} \left(\tau_{30} \left(f_{x^3}^{(3)}(x_i, y_i) \right)^2 + \tau_{21} \left(f_{x^2y}^{(3)}(x_i, y_i) \right)^2 + \tau_{12} \left(f_{xy^2}^{(3)}(x_i, y_i) \right)^2 + \tau_{03} \left(f_{y^3}^{(3)}(x_i, y_i) \right)^2 \right)$$

où τ_{30} , τ_{21} , τ_{12} et τ_{03} sont des paramètres de tension sur les dérivées secondes de f . Plus ces paramètres sont grands, plus elles tendent à être constantes.

Plus précisément, une valeur grande de τ_{30} par exemple évitera que f''_{x^2} oscille en fonction de x , de même une valeur grande de τ_{21} diminuera les oscillations de f''_{x^2} en fonction de y ; et de même pour τ_{12} et τ_{03} pour diminuer les oscillations de f''_{y^2} par rapport à x et y respectivement.

7.7 Opérateur total

En utilisant les opérateurs qu'on vient de définir, la fonction f recherchée sera celle qui minimise la quantité $\mathcal{Q}(f)$ suivante :

$$\mathcal{Q}(f) = \mathcal{J}(f) + \mathcal{B}(f) + \mathcal{T}(f) + \mathcal{A}_c(f) + \mathcal{S}(f) + \mathcal{D}(f)$$

La fonction c (courbure théorique) qui intervient dans $\mathcal{A}_c(f)$ est prise égale à c_1 sur $H \cup HP$, c_2 sur $B \cup BP$ et 0 ailleurs. Les fonctions poids de sphéricité, de direction, et d'approximation seront choisies en fonction de l'importance qu'on accorde aux conditions correspondantes dans les différentes zones.

Chapitre 8

Algorithme de calcul

8.1 Calcul de la fonction verre

On cherche à déterminer, sous forme de produit tensoriel de B-splines de degré 4, une fonction σ vérifiant les conditions Cv.

$$\sigma(x, y) = \sum_{j=-4}^{Nx-1} \sum_{k=-4}^{Ny-1} \alpha_{jk} B_j(x) B_k(y) \quad (8.18)$$

Si on note par S_4 l'ensemble des fonctions qui s'écrivent sous la forme (8.18), il s'agit de résoudre le problème de minimisation suivant :

$$Q(\sigma) = \min_{f \in S_4} Q(f) \quad (\mathcal{P}_1)$$

En écrivant les éléments de S_4 sous leur forme (8.18), on ramène le problème (\mathcal{P}_1) à un problème de minimisation d'une fonctionnelle Q dans $\mathbb{R}^{Nx+4 \times Ny+4}$.

Ici Q n'est pas quadratique, car, pour une fonction f , $Q(f)$ fait intervenir des expressions non quadratiques des dérivées premières de f ,

$$\left(\sqrt{1 + f_x'^2 + f_y'^2} \text{ au dénominateur de } \mathcal{A}_c(f) \text{ par exemple} \right).$$

On est donc amené à utiliser un procédé itératif.

Dans un premier temps, on suppose connues ces dérivées premières (On verra plus loin comment les calculer).

Supposons, donc, que $f'_x = p$ et $f'_y = q$, soient deux fonctions connues. Dans $\mathcal{S}(f)$, l'expression $\sqrt{1 + f_x'^2 + f_y'^2}$ au dénominateur, c'est à dire en (x_i, y_i) $\sqrt{1 + p^2 + q^2}$, peut être omise, car on peut estimer qu'elle n'intervient qu'à la manière d'un poids, donc qu'elle peut être incluse dans $\omega_s(x_i, y_i)$. De même dans $\mathcal{D}(f)$. Par contre dans $\mathcal{A}(f)$ cette expression ne peut être omise. La fonctionnelle Q devient alors quadratique et elle est de la forme :

$$Q(\alpha) = J(\alpha) + B(\alpha) + T(\alpha) + A_c(\alpha) + S(\alpha) + D(\alpha)$$

où $\alpha = (\alpha_{jk})_{j=-4 \dots Nx-1}^{k=-4 \dots Ny-1}$

avec :

$$S(\alpha) = \sum_{i=1}^{Nd} \omega_s(x_i, y_i) \left[\left(\sum_{j,k} \alpha_{jk} \left(B_j''(x_i) B_k(y_i) p_i q_i - B_j'(x_i) B_k'(y_i) (1 + p_i^2) \right) \right)^2 \right. \\ \left. + \left(\sum_{j,k} \alpha_{jk} \left(B_j''(x_i) B_k(y_i) (1 + q_i^2) - B_j(x_i) B_k''(y_i) (1 + p_i^2) \right) \right)^2 \right. \\ \left. + \left(\sum_{j,k} \alpha_{jk} \left(B_j(x_i) B_k''(y_i) p_i q_i - B_j'(x_i) B_k'(y_i) (1 + q_i^2) \right) \right)^2 \right]$$

$$J(\sigma) = \rho \sum_{i=1}^{Nd} \left(\sum_{j,k} \alpha_{jk} B_j^{(4)}(x_i) B_k^{(4)}(y_i) \right)^2$$

$$B(\alpha) = \mu \sum_{i=1}^3 \left(\sum_{j,k} \alpha_{jk} B_j(c_i) B_k(d_i) \right)^2$$

$$D(\alpha) = \sum_{i=1}^{Nd} \omega_d \left(\sum_{j,k} \alpha_{jk} B_j(x_i) B_k''(y_i) p_i q_i - B_j'(x_i) B_k'(y_i) (1 + q_i^2) \right)^2$$

$$A_c(\alpha) = \sum_{i=1}^{Nd} \left[\omega_h(x_i, y_i) \left(\sum_{j,k} \alpha_{jk} \left(\frac{B_j''(x_i) B_k(y_i)}{\sqrt{(1 + p_i^2 + q_i^2)} (1 + p_i^2)} - c(x_i, y_i) \right) \right)^2 \right. \\ \left. + \omega_v(x_i, y_i) \left(\sum_{j,k} \alpha_{jk} \left(\frac{B_j(x_i) B_k''(y_i)}{\sqrt{(1 + p_i^2 + q_i^2)} (1 + q_i^2)} - c(x_i, y_i) \right) \right)^2 \right]$$

$$T(\alpha) = \sum_{i=1}^{Nd} \tau_{30} \left(\sum_{j,k} \alpha_{jk} B_j^{(3)}(x_i) B_k(y_i) \right)^2 + \tau_{21} \left(\sum_{j,k} \alpha_{jk} B_j^{(2)}(x_i) B_k'(y_i) \right)^2 \\ + \tau_{12} \left(\sum_{j,k} \alpha_{jk} B_j'(x_i) B_k^{(2)}(y_i) \right)^2 + \tau_{03} \left(\sum_{j,k} \alpha_{jk} B_j(x_i) B_k^{(3)}(y_i) \right)^2$$

Tous les opérateurs J, A, B, S, T et D sont quadratiques en $(\alpha_{jk})_{j=-4 \dots Nx-1}^{k=-4 \dots Ny-1}$. On a donc un problème de minimisation (P_1) , dans $\mathbb{R}^{Nx+4 \times Ny+4}$.

$$\underset{\alpha \in \mathbb{R}^{Nx+4 \times Ny+4}}{\text{Minimiser}} Q(\alpha) \quad (P_1)$$

En écrivant les conditions variationnelles d'optimalité: nullité des dérivées premières de Q par rapport aux α_{jk} , on obtient un système linéaire de $(Nx+4) \times (Ny+4)$ équations aux inconnues $\alpha = (\alpha_{jk})_{j=-4 \dots Nx-1}^{k=-4 \dots Ny-1}$. On résout ce système par une méthode de Cholesky par exemple.

8.2 Itérations sur les dérivées premières

Dans le paragraphe précédent on a supposé les fonctions p et q connues, ce qui n'est pas le cas en pratique. Notre idée est d'effectuer des itérations sur ces fonctions de la façon suivante :

Itération 0	$p = q = 0$	résoudre (P1) d'où σ_0
Itération 1	$p = (\sigma_0)'_x$ et $q = (\sigma_0)'_y$	résoudre (P1) d'où σ_1
	.	
	.	
	.	
Itération k+1	$p = (\sigma_k)'_x$ et $q = (\sigma_k)'_y$	résoudre (P1) d'où σ_{k+1}

On se fixe un test d'arrêt relatif aux courbures de σ_k par exemple :

$$\sum_{i=1}^{Nd} (c_h^k(x_i, y_i) - c_h^{k+1}(x_i, y_i))^2 + (c_v^k(x_i, y_i) - c_v^{k+1}(x_i, y_i))^2 \leq \varepsilon$$

où $c_h^k(x_i, y_i)$ est la courbure horizontale de σ_k en (x_i, y_i) et $c_v^k(x_i, y_i)$ est la courbure verticale de σ_k en (x_i, y_i) .

On n'a pas pu démontrer la convergence théorique de ces itérations vers une solution unique. On verra plus loin des essais numérique de cette convergence.

8.3 Remarques

Tension :

Comme précisé avant, la meilleure façon d'éviter des oscillations des courbures, est d'imposer un terme de tension à ces courbures, par un opérateur de minimisation des dérivées des courbures horizontales et verticales par exemple. On ne peut pas le faire dans notre cas, car on obtiendrait un opérateur non quadratique en (α_{ij}) qu'on ne pourra pas rendre quadratique même en utilisant l'itération sur la dérivée première précédemment décrite.

En effet on a :

$$\frac{\partial c_h(x, y)}{\partial x} = \left(\frac{f''_{xx}}{\sqrt{1 + f_x'^2 + f_y'^2} (1 + f_x'^2)} \right)'_x$$

et c'est une expression assez compliquée, mais, surtout, qui fait intervenir des produits croisés des différentes dérivées de f .

Chapitre 9

Résultats et expériences numériques

9.1 Introduction

Dans ce chapitre on présente quelques résultats numériques de la méthode. On se placera dans le cas où l'on veut concevoir un verre progressif ayant un rayon de courbure égal à 80, courbure égale à 0.0125, dans la zone HP et égal à 60, courbure égale à 0.0167, dans la zone BP .

Dans un premier temps, on visualisera l'influence de chaque opérateur, par le biais de ses paramètres, sur les courbures et les directions principales du verre résultant. Ainsi que le comportement des différentes courbures du verre en fonction du nombre d'itérations effectuées sur les dérivées premières.

Dans un deuxième temps, on donnera une description du logiciel mis au point, ainsi qu'une façon de l'utiliser pour concevoir un verre progressif donné.

9.2 Outils de visualisation

Visualisation de l'astigmatisme: "Table de couleur"

Définition : **Astigmatisme d'un verre** *On dit qu'un verre présente un astigmatisme en un point M si les courbures principales de la fonction verre, en ce point, ne sont pas égales, i.e. le point M n'est ombilical. Cet astigmatisme est alors égal à Ast :*

$$Ast = (n - 1)(C_{max} - C_{min}) = (n - 1) \left(\frac{1}{R_{min}} - \frac{1}{R_{max}} \right)$$

n étant l'indice de la matière utilisée pour le verre. $C_{max} = \frac{1}{R_{min}}$ la courbure maximale au point M et $C_{min} = \frac{1}{R_{max}}$ la courbure minimale au point M

L'astigmatisme est exprimé en dioptrie, noté δ , lorsque les rayons de courbures sont exprimés en mètre.

On parlera souvent de l'astigmatisme en un point (x, y) du plan; on sous-entend alors l'astigmatisme au point $(x, y, f(x, y))$ du verre.

Pour visualiser graphiquement l'astigmatisme en un point (x, y) du domaine Ω , on lui associe une couleur. Si l'écran de visualisation est un écran couleur, on associe à la valeur 0 de l'astigmatisme la couleur bleu (froide). Et si la valeur de l'astigmatisme est

supérieure ou égal à 1 on lui associe la couleur rouge (chaude). Les valeurs intermédiaires sont représentées par des couleurs de chaleur intermédiaire, allant du bleu au rouge en passant par le vert et le jaune, avec la convention: *Plus c'est chaud, plus c'est rouge*. Si l'écran de visualisation est monochrome (niveau de gris), on associe à chaque valeur de l'astigmatisme un niveau de gris allant du blanc, ($Ast = 0$), au noir, ($Ast \geq 1$). Dans ce rapport on utilisera ce dernier mode de visualisation.

Visualisation des directions principales:

Pour visualiser les directions principales en un point (x, y) , on utilisera trois modes:

Mode croix: On visualise, avec la couleur correspondant à l'astigmatisme en un point M , les projections des deux directions principales en ce point sur le plan (xoy) , la direction de courbure maximale étant plus épaisse que la direction de courbure minimale. La proportion de longueur entre les représentations des deux directions étant la même que la proportion entre la courbure maximale et la courbure minimale.

Mode ellipse: On visualise l'ellipse définie par les deux directions principales en un point M . Ceci illustre mieux la qualité du verre en ce point. Si cette ellipse est un cercle le point est un ombilic, sinon la forme de cette ellipse indique plus précisément l'écart entre les courbures maximale et minimale en ce point.

Mode table: On visualise des tables de couleur représentant le sinus de l'angle entre l'une des directions de courbure et l'axe des x ou l'axe des y .

Visualisation des rayons de courbure:

L'étude de certaines coupes des rayons de courbure par des plans verticaux parallèles à l'un des axes de coordonnées est très intéressante. Cette étude permet de se faire une idée très précise sur le comportement de ces rayons dans les zones importantes du verre. Les coupes de la surface représentant un des 2 rayons de courbure, par un plan parallèle au plan (yoz) , seront appelées coupes verticales. Elles nous donnent des indications sur la façon avec laquelle ce rayon passe de sa valeur dans la zone BP à sa valeur dans la zone HP . Ce passage doit se faire sans oscillations. D'autre part ces coupes représentent des fonctions de la variable y , x étant constante; on peut alors contrôler leurs oscillations par

les paramètres de tension sur l'une des dérivées troisième de f , $\frac{\partial^3 f}{\partial^2 x \partial y}$ ou $\frac{\partial^3 f}{\partial^3 y}$, f étant la fonction représentant le verre. De même les coupes de la surface représentant l'un des rayons de courbure par un plan vertical parallèle au plan (xoz) , qu'on appellera coupes horizontales, représentent la variation de ce rayon relativement à x , y étant constante. On peut ainsi, contrôler leurs oscillations par une tension sur l'une des dérivées troisième de f , $\frac{\partial^3 f}{\partial^3 x}$ ou $\frac{\partial^3 f}{\partial^2 y \partial x}$.

D'une façon générale, on a le tableau de dépendance (Tableau 9.1), qui indique le paramètre sur lequel il faut agir pour contrôler les oscillations d'un des rayons de courbure suivant une coupe horizontale ou verticale.

On lira, par exemple dans la troisième ligne du tableau, que pour atténuer les oscillations d'une coupe horizontale du rayon de courbure verticale il faut agir sur le paramètre de tension sur la dérivée $\frac{\partial^3 f}{\partial x \partial^2 y}$, c'est à dire τ_{12}

Rayon	Coupe	Dérivée	Paramètre de tension
Horizontal	Horizontale	$\frac{\partial^3 f}{\partial^3 x}$	τ_{30}
Horizontal	Verticale	$\frac{\partial^3 f}{\partial^2 x \partial y}$	τ_{21}
Vertical	Horizontale	$\frac{\partial^3 f}{\partial x \partial^2 y}$	τ_{12}
Vertical	Verticale	$\frac{\partial^3 f}{\partial^3 y}$	τ_{03}

Tableau 9.1 : Paramètres pour contrôler les oscillations des rayons de courbure

On peut aussi visualiser de la même façon les rayons de courbures minimale et maximale. Pour agir sur leurs oscillations, on suit le même principe. Mais ceci n'est pas nécessaire, car la courbure minimale, par exemple, est une combinaison linéaire des deux courbures horizontale et verticale. Il suffit donc de régler les oscillations de ces dernières pour agir sur les oscillations de C_{min} , et donc de R_{max} . Il en est de même pour la courbure maximale et donc pour R_{min} .

Autres modes de visualisation:

On utilisera souvent, pour donner une idée sur certaines quantités, les modes de visualisation classiques, surfaces et courbes de niveau. On y associera souvent l'usage des couleurs différentes, et cela en utilisant le principe: "plus c'est grand (chaud), plus c'est rouge". Si la couleur n'apporte pas d'informations supplémentaires ou n'est pas possible, on utilisera des niveaux de gris ou des motifs, suivant le principe: 'plus c'est grand, plus c'est sombre'.

9.3 Itérations sur les dérivées premières

Sachant qu'on n'a pas pu démontrer la convergence des itérations sur les dérivées premières, on a testé la méthode sur différents exemples avec des différents jeux de poids et paramètres. On n'est jamais tombé sur un cas où il n'y a pas de convergence sauf si l'on change certains paramètres au cours des itérations. On présente ici la figure (9.1) qui représente la moyenne de six courbes de convergence obtenues pour différents problèmes.

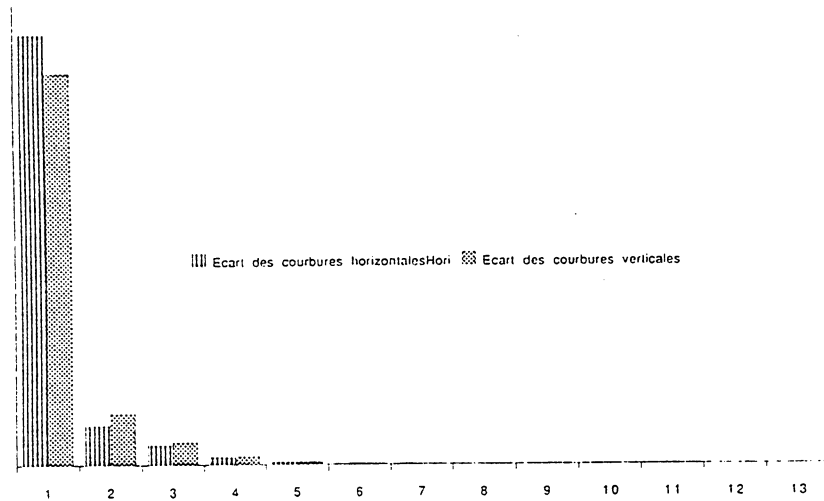


figure 9.1 *Ecart entre courbures en fonction du nombre d'itérations, même jeu de paramètres*

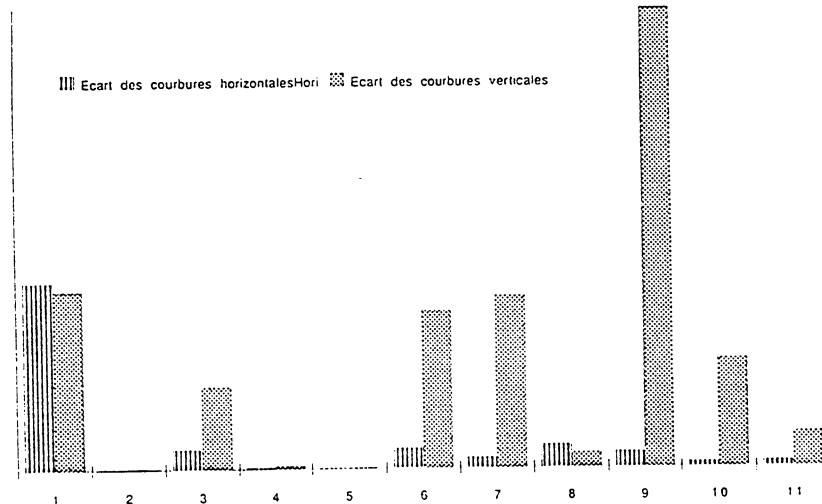


figure 9.2 *Ecart entre courbures en fonction du nombre d'itérations, jeux de paramètres différents*

Dans la figure (9.2), on présente la courbe représentant cet écart en fonction du nombre d'itérations (on a changé la valeur de certains paramètres au cours des itérations).

D'une façon générale, on a remarqué que deux à quatre itérations, après chaque changement de paramètres, suffisent pour que l'écart entre les différentes courbures de deux itérés successifs se stabilise autour d'une faible valeur (10^{-9}).

9.4 Influence des fonctions poids

Il est inutile de rappeler l'influence des fonctions poids d'approximation de courbures ω_h et ω_v sur le verre final. Elles jouent des rôles analogues au rôle que joue le paramètre de lissage dans un problème de lissage classique.

Le paramètre de bord μ est toujours pris très grand.

Fonction poids de sphéricité:

Cette fonction, qu'on a notée ω_s , nous permet de diminuer l'astigmatisme du verre dans les zones où on le désire faible, surtout dans *BP*, *I* et *HP*.

On présente plus loin deux tables représentant l'astigmatisme pour deux fonctions poids différentes.

Dans le graphique (9.3), la fonction ω_s est trop faible. On remarque de grandes valeurs de l'astigmatisme, par des couleurs moins claires, dans les zones *BP*, *I* et *HP*.

Dans le graphique (9.4), où l'on a augmenté les valeurs de la fonction poids dans les zones *BP*, *I* et *HP*, la valeur de l'astigmatisme y est très faible.

Fonction poids aux directions:

On utilise cette fonction, qu'on a notée ω_d , pour imposer à l'une des directions principales un rapprochement à l'axe des y dans les zones *T* et *N*. Si l'on prend cette fonction très faible, ou égale à 0, on ne peut pas prévoir le comportement des directions principales (figure (9.5)). Par contre si on la prend très grande, l'une des directions principales (figure (9.6)), ici la direction de courbure maximale, tend à s'identifier à l'axe des y . On ne peut pas prévoir laquelle des deux directions principales se rapprochera le plus de l'axe des y .

La situation de la figure (9.6), n'est pas l'idéal pour les opticiens. D'après leurs recommandations, il est préférable de tolérer un léger écartement entre la direction principale et l'axe car en forçant trop sur ω_d , on perd sur l'astigmatisme désiré aux zones *BP*, *I* et *HP*, figure (9.7). Il s'agit donc, de trouver un compromis entre les deux dernières situations.

Paramètres de tension:

Comme on l'a précisé précédemment, les quatre paramètres de tension sur les dérivées troisièmes de f , servent à contrôler les oscillations des rayons de courbure du verre. On présente ici quatre paires de graphiques ((9.8), (9.9), (9.10), (9.11)), chaque paire représentant des coupes dans une direction, horizontale ou verticale, d'un des rayons de

courbure, horizontal ou vertical. Les graphiques de gauche sont obtenus avec des paramètres de tension nuls. ceux de droite avec des paramètres de tension assez grands.

On remarque, d'après ces graphiques, que l'opérateur de tension joue bien son rôle. Ceci en comparant les graphiques de gauche, où il y a présence de beaucoup d'oscillations, avec ceux de droite, où ces oscillations ont été pratiquement éliminées, surtout dans les coupes verticales centrales, qui sont les plus importantes pour la conception du verre.

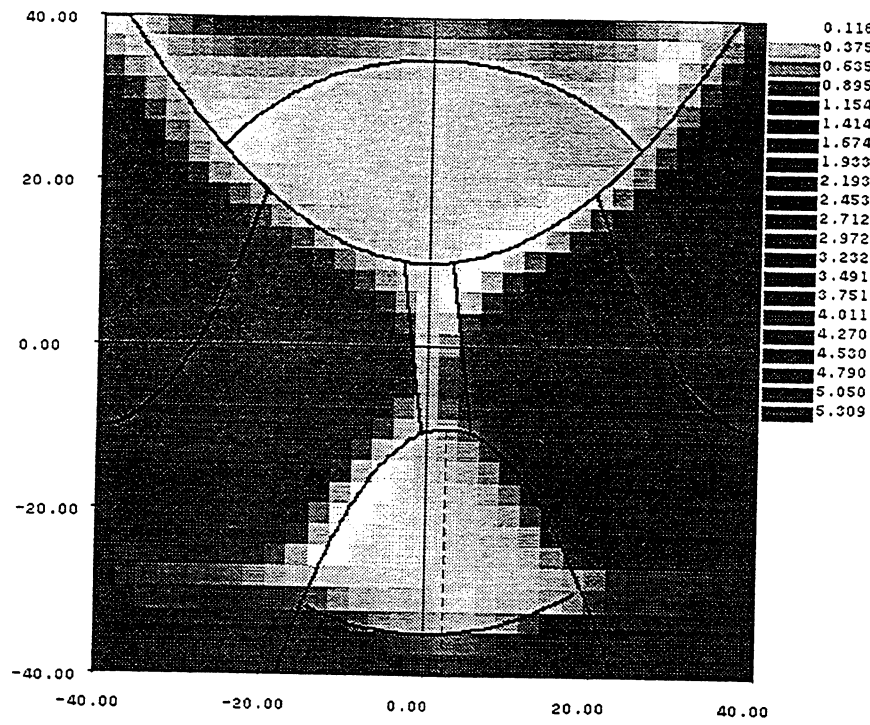


figure 9.3 Table d'astigmatisme - ω_s trop faible

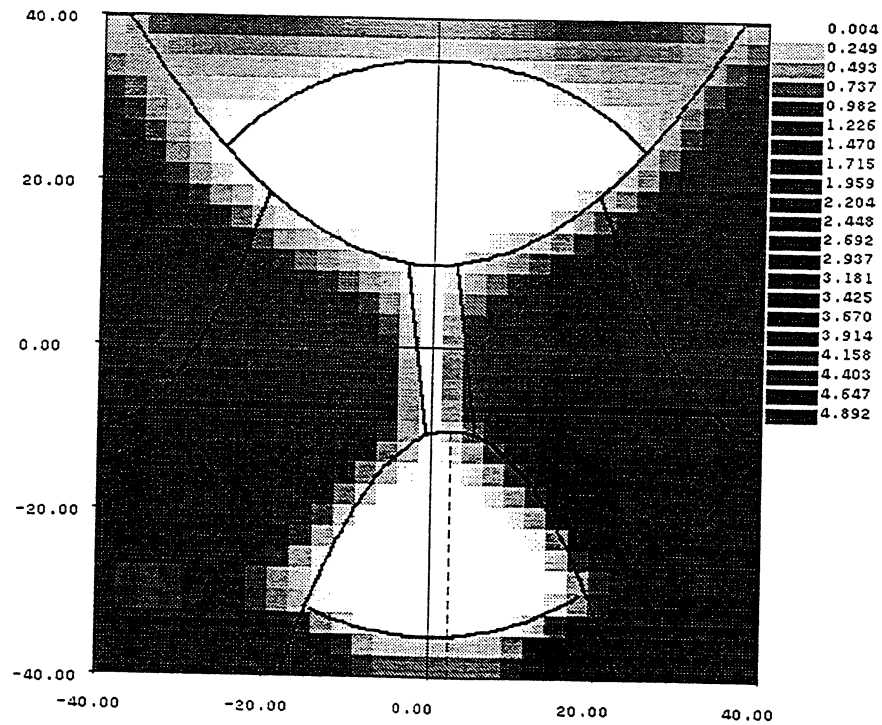


figure 9.4 Table d'astigmatisme - bon choix de ω_s pour l'astigmatisme

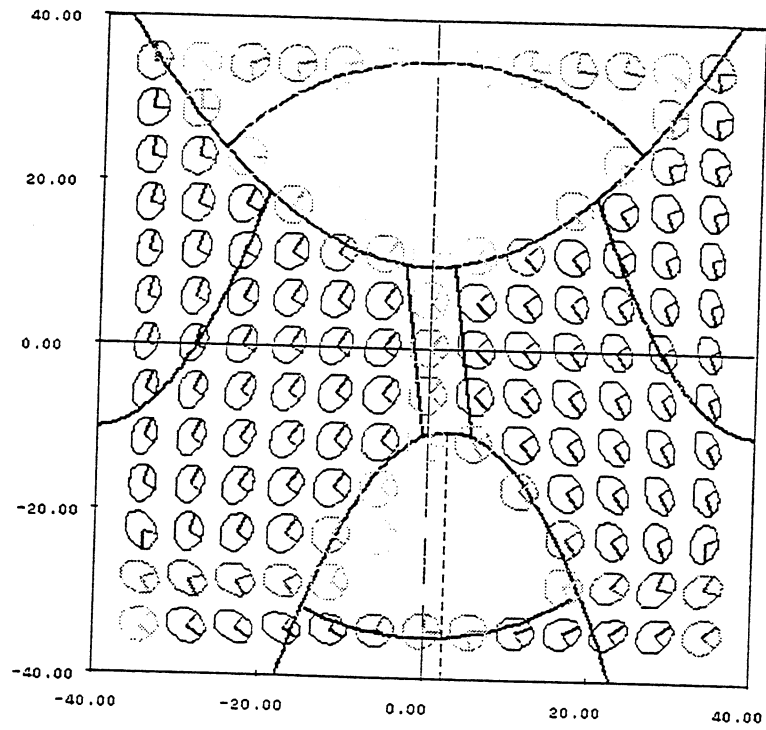


figure 9.5 Directions principales - $\omega_d = 0$

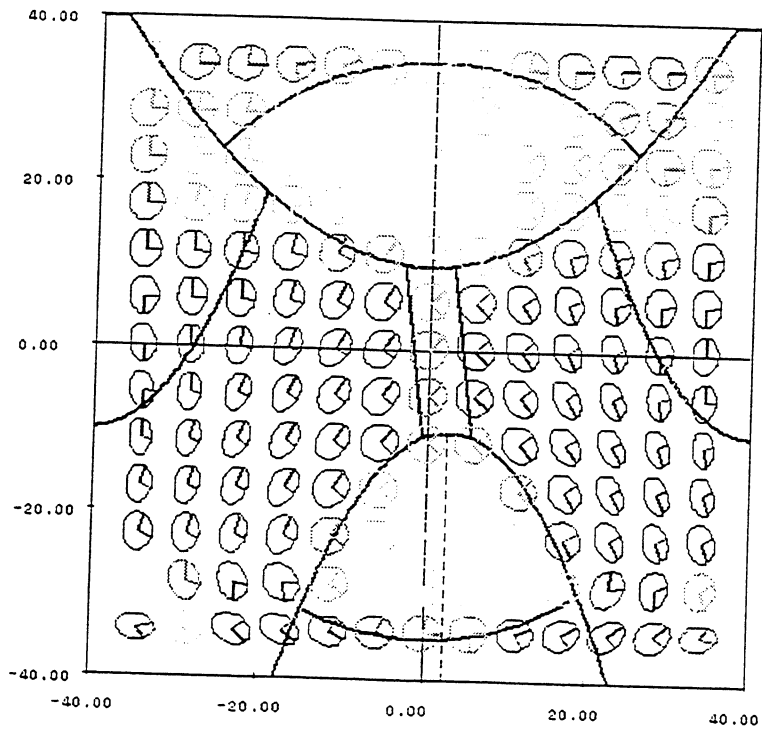


figure 9.6 Directions principales - ω_d trop grand

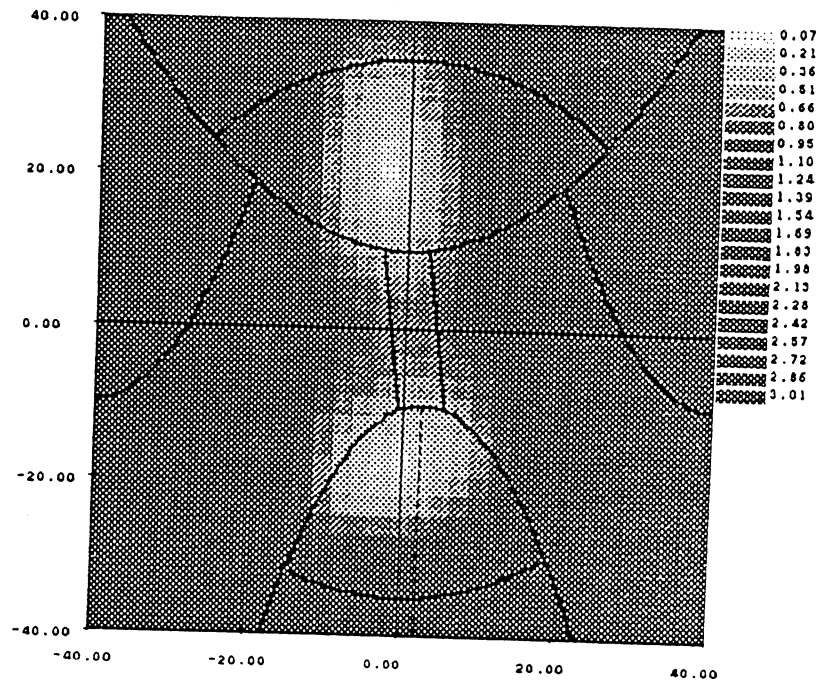


figure 9.7 Table d'astigmatisme - ω_d trop grand

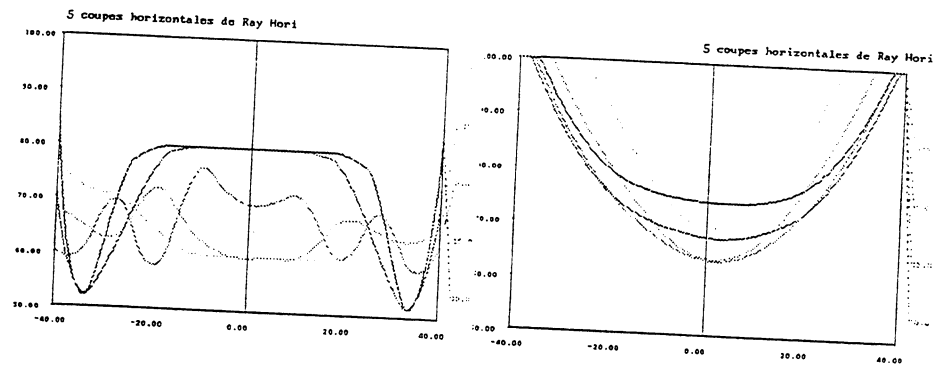


figure 9.8 Coupes horizontales du rayon horizontal
 $\tau_{30} = 0$ à gauche et $\tau_{30} = 100$ à droite

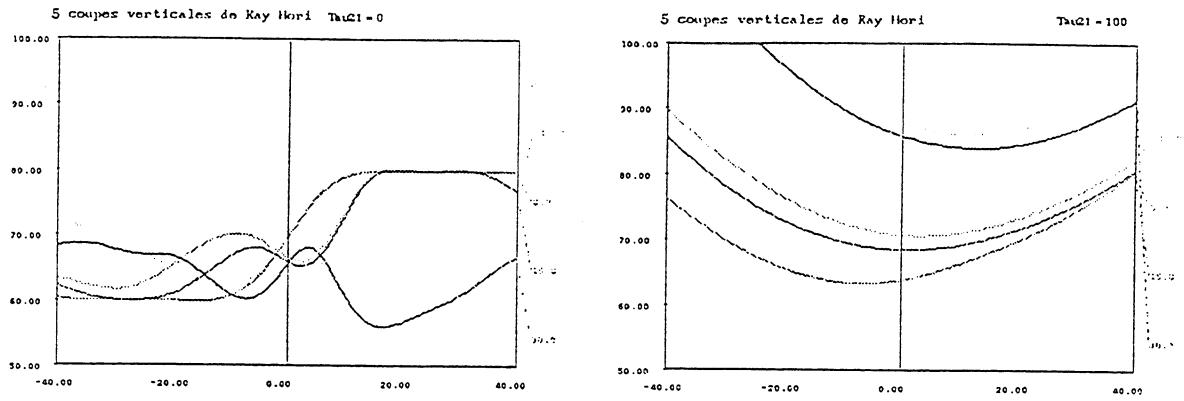


figure 9.9 Coupes verticales du rayon horizontal
 $\tau_{21} = 0$ à gauche et $\tau_{21} = 100$ à droite

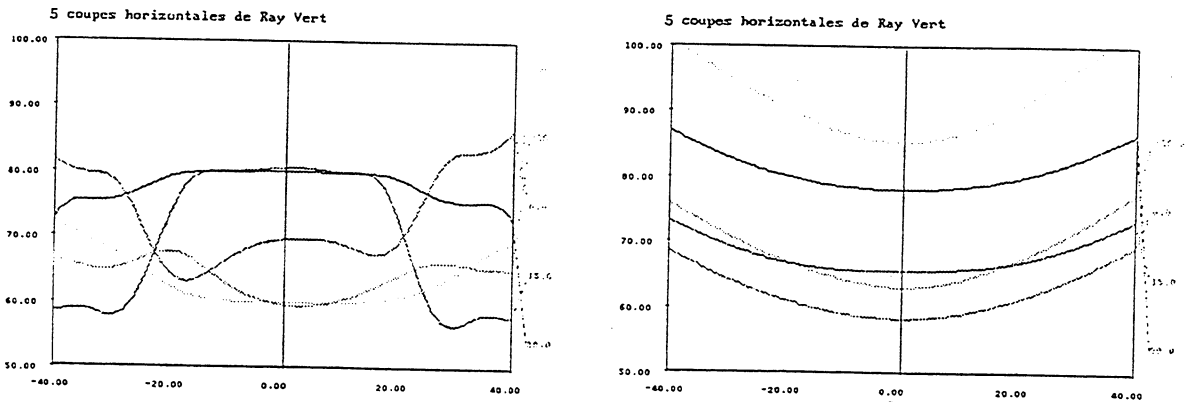


figure 9.10 Coupes horizontales du rayon vertical
 $\tau_{12} = 0$ à gauche et $\tau_{12} = 100$ à droite

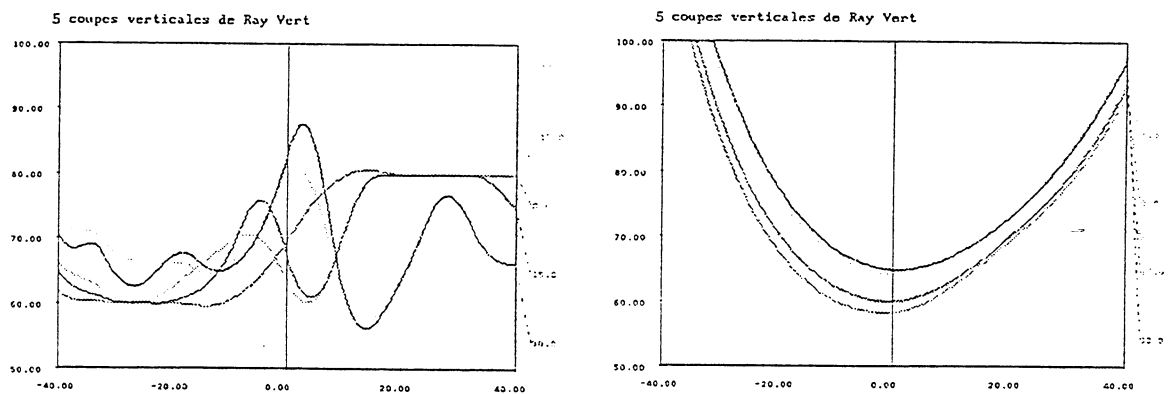


figure 9.11 Coupes verticales du rayon vertical
 $\tau_{03} = 0$ à gauche et $\tau_{03} = 100$ à droite

9.5 Utilisation du logiciel de conception d'un verre progressif

La conception du verre se déroule suivant les trois étapes suivantes:

Etape 1 - Choix des données:

Dans cette étape on choisit toutes les données initiales du problème. Sinon des données par défaut sont prises. Ces données sont:

- La subdivision de Ω en zones, sinon une subdivision symétrique est prise par défaut.
- La valeur des rayons de courbure dans les zones $HP \cup H$ et $BP \cup B$.
- Le nombre de B-splines pour chaque variable, par défaut $N_x = 10$ et $N_y = 10$.

Etape 2 - Choix des paramètres:

Dans cette étape on fait un choix de paramètres et on effectue 2 ou 3 itérations.

Ces paramètres sont:

- Les fonctions poids: ω_h , ω_v , ω_s et ω_d .
- Les autres paramètres: ρ , mu , τ_{30} , τ_{21} , τ_{12} , τ_{03} .

Etape 3 - Visualisation des résultats:

Dans cette étape on visualise les propriétés du verre conçu pour vérifier les résultats du lissage effectué. On visualise alors:

- Les coupes des rayons de courbure pour apprécier l'approximation des courbures, les oscillations des rayons et leur aspect lisse pour le passage de la zone BP à la zone HP .
- La table des directions principales pour vérifier si elles se rapprochent des axes.
- La table de l'astigmatisme pour voir sa valeur dans les zones où il doit être faible.
- On peut, pour avoir une meilleure idée, visualiser les surfaces ou les courbes de niveau qui représentent l'une des quantités qui nous intéresse.

Si le verre résultant est satisfaisant, on choisit un mode de sauvegarde ou de sortie des résultats, soit les coefficients des B-splines, les valeurs de la fonction verre et ses propriétés sur une grille régulière, sauvegardes ou impressions des graphiques.

Si le verre résultant n'est pas satisfaisant, on détermine les défauts à corriger: trop d'oscillations de l'un des rayons, grandes valeurs d'astigmatisme, ...

On revient à l'étape 2.

On peut arrêter le programme à tout moment et sauver les résultats actuels, pour reprendre au même endroit une prochaine fois.

9.6 Exemple

On veut concevoir un verre progressif vérifiant les propriétés C_v sur une subdivision donnée du domaine Ω , avec $R = 80$, le rayon de courbure dans la zone $HP \cup H$, et $r = 60$, le rayon de courbure dans la zone $BP \cup B$. On procède alors suivant les trois étapes.

Etape 1 - Choix des données:

Un écran visualisant l'ensemble des données par défaut s'affiche dès le lancement du programme.

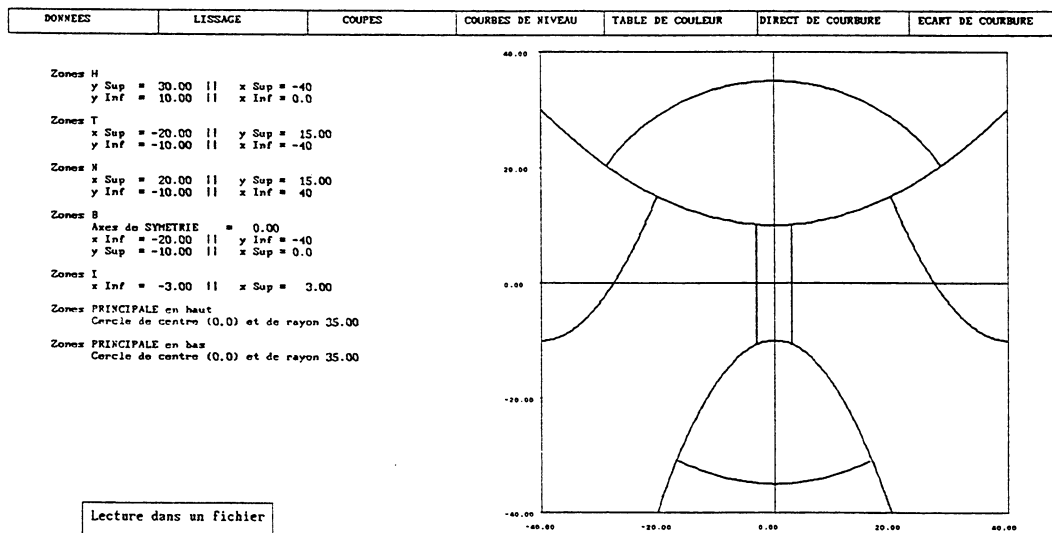


figure 9.12 Ecran des données par défaut

On change les données soit avec la souris, en cliquant sur la zone où est la donnée à changer, soit par une lecture dans un fichier.

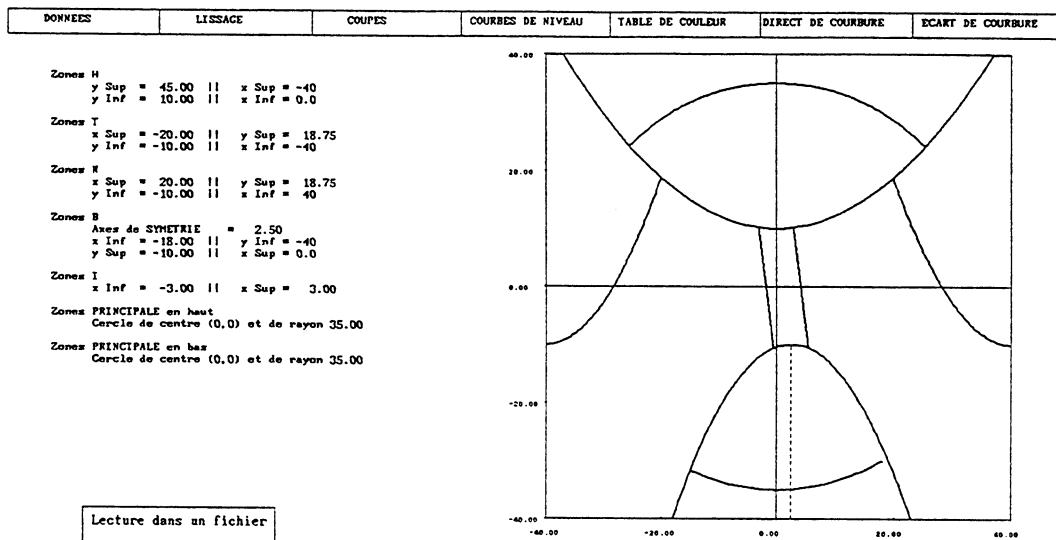


figure 9.13 Ecran des données finales

A n'importe quel moment de la conception, on peut revenir à cette étape. Tout le processus est alors réinitialisé.

Etape 2 - Choix des paramètres:

En cliquant sur la fenêtre **lissage**, les noms des différents types de fonctions poids et de paramètres s'affichent. On choisit le type de paramètres à changer, en cliquant sur la fenêtre qui lui correspond. Si les paramètres à changer sont des fonctions poids, on choisit leur valeur dans une zone de Ω en cliquant sur cette zone.

Sphericite	Directions	Approx Rv	Approx Rh
Region : B--> 1.0E+000	Region : B--> 0.0E+000	Region : B--> 1.0E+001	Region : B--> 5.0E+001
Region : BP--> 1.0E+002	Region : BP--> 0.0E+000	Region : BP--> 1.0E+003	Region : BP--> 5.0E+003
Region : I--> 1.0E+003	Region : I--> 0.0E+000	Region : I--> 0.0E+000	Region : I--> 0.0E+000
Region : H--> 1.0E+000	Region : H--> 0.0E+000	Region : H--> 1.0E+002	Region : H--> 5.0E+001
Region : HP--> 1.0E+002	Region : HP--> 0.0E+000	Region : HP--> 1.0E+003	Region : HP--> 5.0E+003
Region : T--> 0.0E+000	Region : T--> 1.0E+000	Region : T--> 0.0E+000	Region : T--> 0.0E+000
Region : N--> 0.0E+000	Region : N--> 1.0E+000	Region : N--> 0.0E+000	Region : N--> 0.0E+000
Region : L--> 1.0E+000	Region : L--> 0.0E+000	Region : L--> 0.0E+000	Region : L--> 0.0E+000

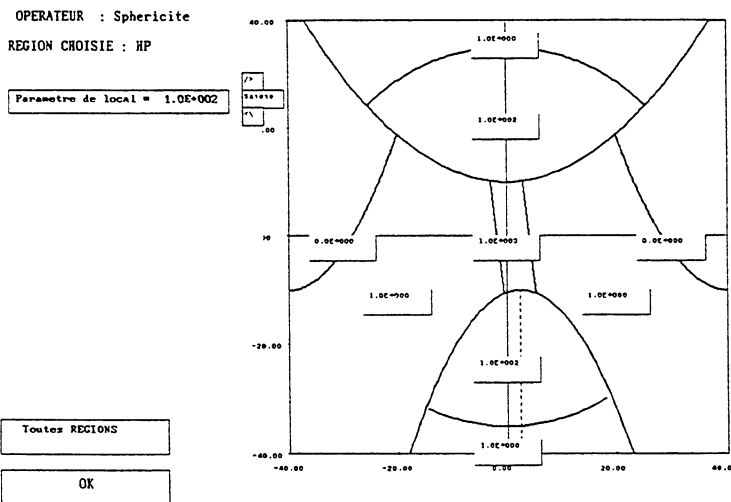


figure 9.14 Ecran des fonctions poids

On effectue alors, deux ou trois itérations.

Après chaque série d'itérations un écran, résumant les valeurs de tous les paramètres et le nombres d'itérations effectuées, s'affiche.

EXEMPLE NUMERO : 1

LES ZONES ET LES DONNEES POUR LE LISSAGE DU VERRE

Zones H
y Sup = 45.00 || x Sup = -40
y Inf = 10.00 || x Inf = 0.0

Zones T
x Sup = -20.00 || y Sup = 18.75
y Inf = -10.00 || x Inf = -40

Zones M
x Sup = 20.00 || y Sup = 18.75
y Inf = -10.00 || x Inf = 40

Zones B
Axe de SYMETRIE = 2.50
x Inf = -18.00 || y Inf = -40
y Sup = -10.00 || x Sup = 0.0

Zones I
x Inf = -3.00 || x Sup = 3.00

Zones PRINCIPALE en haut
Cercle de centro (0,0) et de rayon 35.00

Zones PRINCIPALE en bas
Cercle de centro (0,0) et de rayon 35.00

PARAMETRES DE LISSAGE DU VERRE

ZONES	Sphericite	Directions	Approx Rv	Approx Rh
B	1.0E+000	0.0E+000	1.0E-001	5.0E-001
BP	1.0E+002	0.0E+000	1.0E+003	5.0E+003
I	1.0E+003	0.0E+000	0.0E+000	0.0E+000
H	1.0E+000	0.0E+000	1.0E-002	5.0E-001
HP	1.0E+002	0.0E+000	1.0E+003	5.0E+003
T	0.0E+000	1.0E+000	0.0E+000	0.0E+000
M	0.0E+000	1.0E+000	0.0E+000	0.0E+000
L	1.0E+000	0.0E+000	0.0E+000	0.0E+000

Parametre de regularization = 1.0E-002
Poid aux coins = 1.0E+006

	D3X	D3Y	D2XDY	D2YDX
Parametres de tension	1.0E+000	1.0E+000	0.0E+000	0.0E+000

AUTRES DONNEES

Nombre de donnees totales = 1600
Nombre de Esplines = 10x10
Degres des Esplines = 4

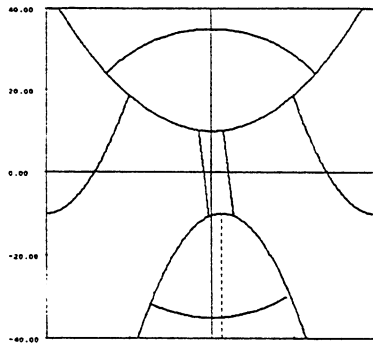


figure 9.15 Résumé de l'étape

Etape 3 - Visualisation des résultats:

Dès la fin des itérations désirées, on examine les résultats suivant les modes décrits précédemment.

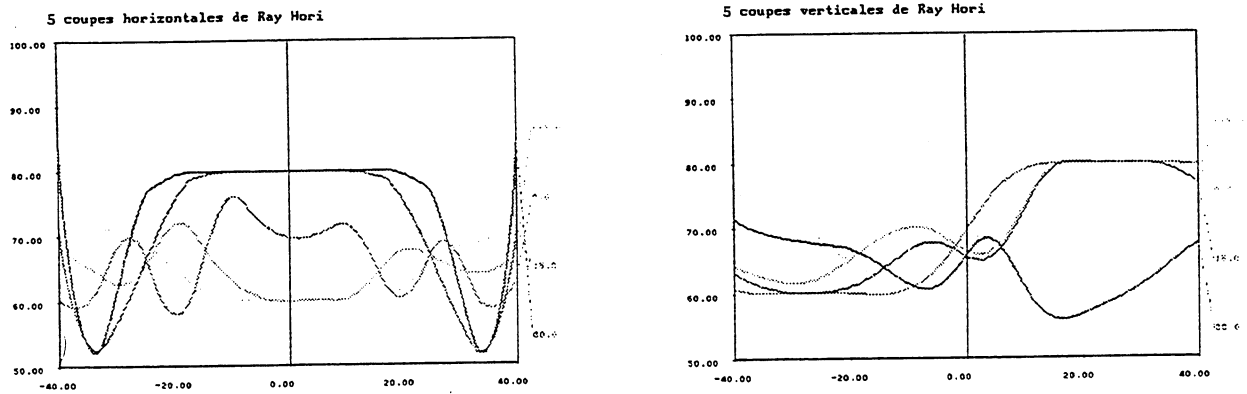


figure 9.16 Coupes horizontales et verticales du rayon de courbure horizontal

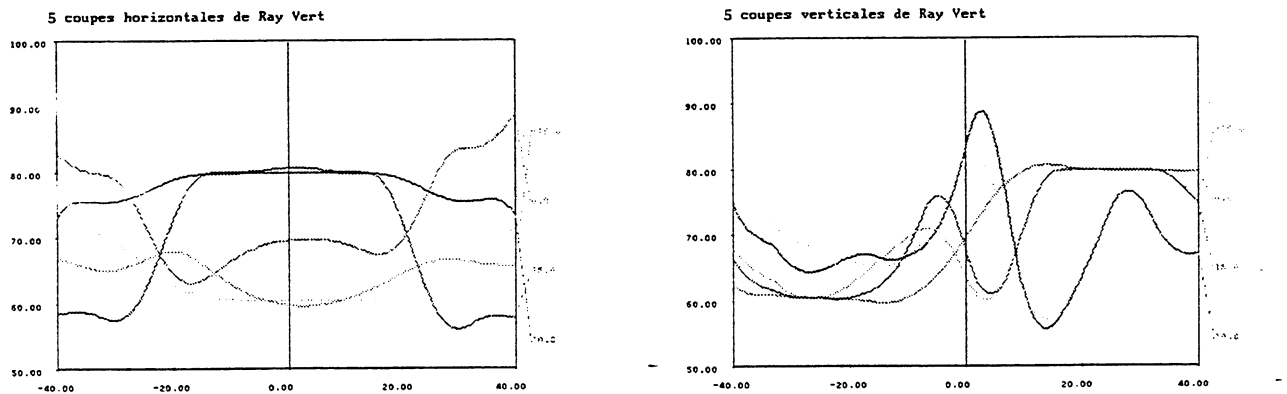


figure 9.17 Coupes horizontales et verticales du rayon de courbure vertical

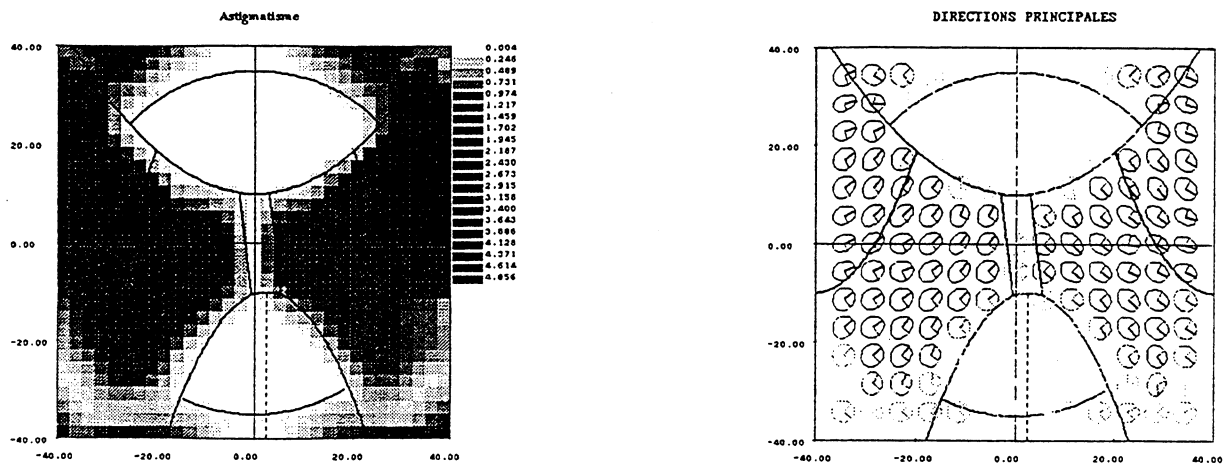


figure 9.18 Astigmatisme et directions principales

D'après les deux graphiques (9.16) et (9.17), les rayons de courbure horizontale et verticale présentent beaucoup d'oscillations. Par contre, d'après la figure (9.18), on remarque que l'astigmatisme du verre dans les zones HP et BP n'est pas trop mauvais. Il en est de même pour les directions principales. On va, donc augmenter les valeurs des paramètres de tension sur les dérivées $\frac{\partial^3 f}{\partial^2 x \partial y}$ et $\frac{\partial^3 f}{\partial x \partial^2 y}$. On ne touche pas aux autres paramètres.

On reprend l'étape 2.

Après un certain nombre d'itérations, on trouve les résultats suivants :

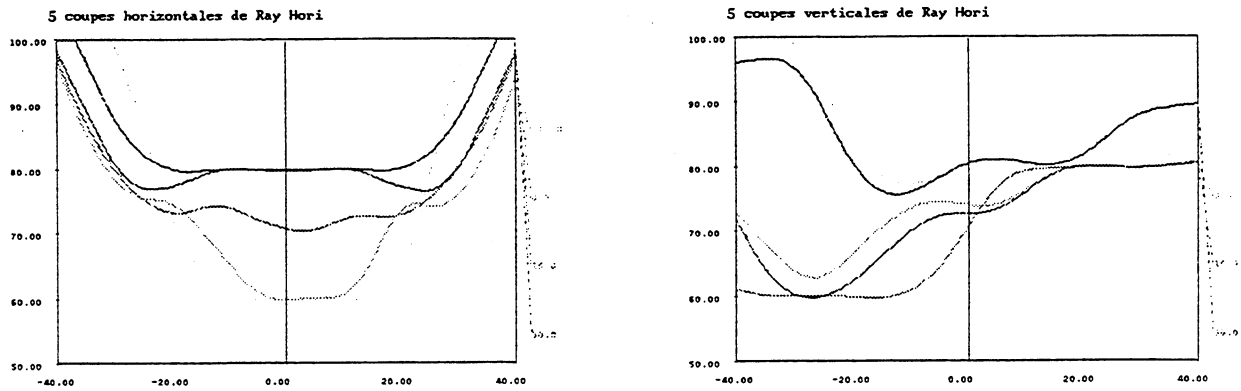


figure 9.19 Coupes horizontales et verticales du rayon de courbure horizontal

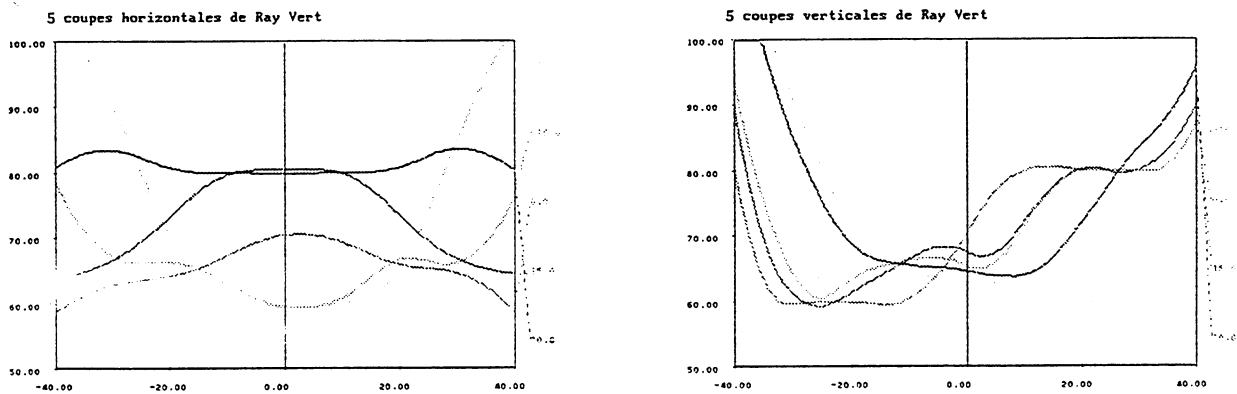


figure 9.20 Coupes horizontales et verticales du rayon de courbure vertical

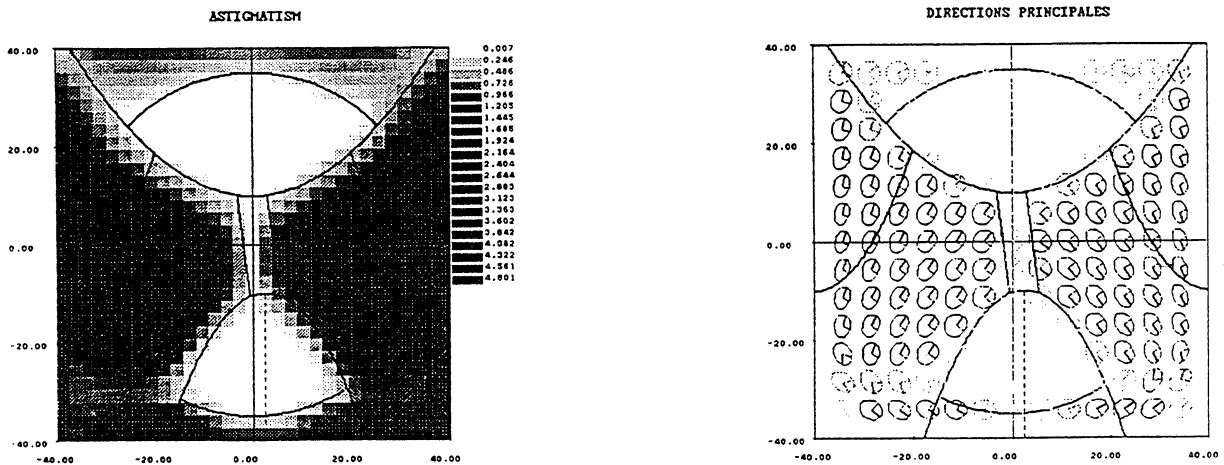


figure 9.21 Astigmatisme et directions principales

D'après l'examen des coupes des deux rayons de courbures (9.19) et (9.20), on remarque qu'on a éliminé les oscillations de ceux-ci. D'après les tables de l'astigmatisme (9.21), celui-ci est visuellement faible dans les zones HP , I et BP . De même, d'après les représentations des directions principales (9.21), celles-ci sont assez proches de celles des axes de coordonnées dans les zones T et N .

On peut se contenter de ces visualisations, mais on peut aussi effectuer un examen très précis de l'astigmatisme et des directions principales, ceci en se promenant dans la table de l'astigmatisme (9.22) avec la souris.

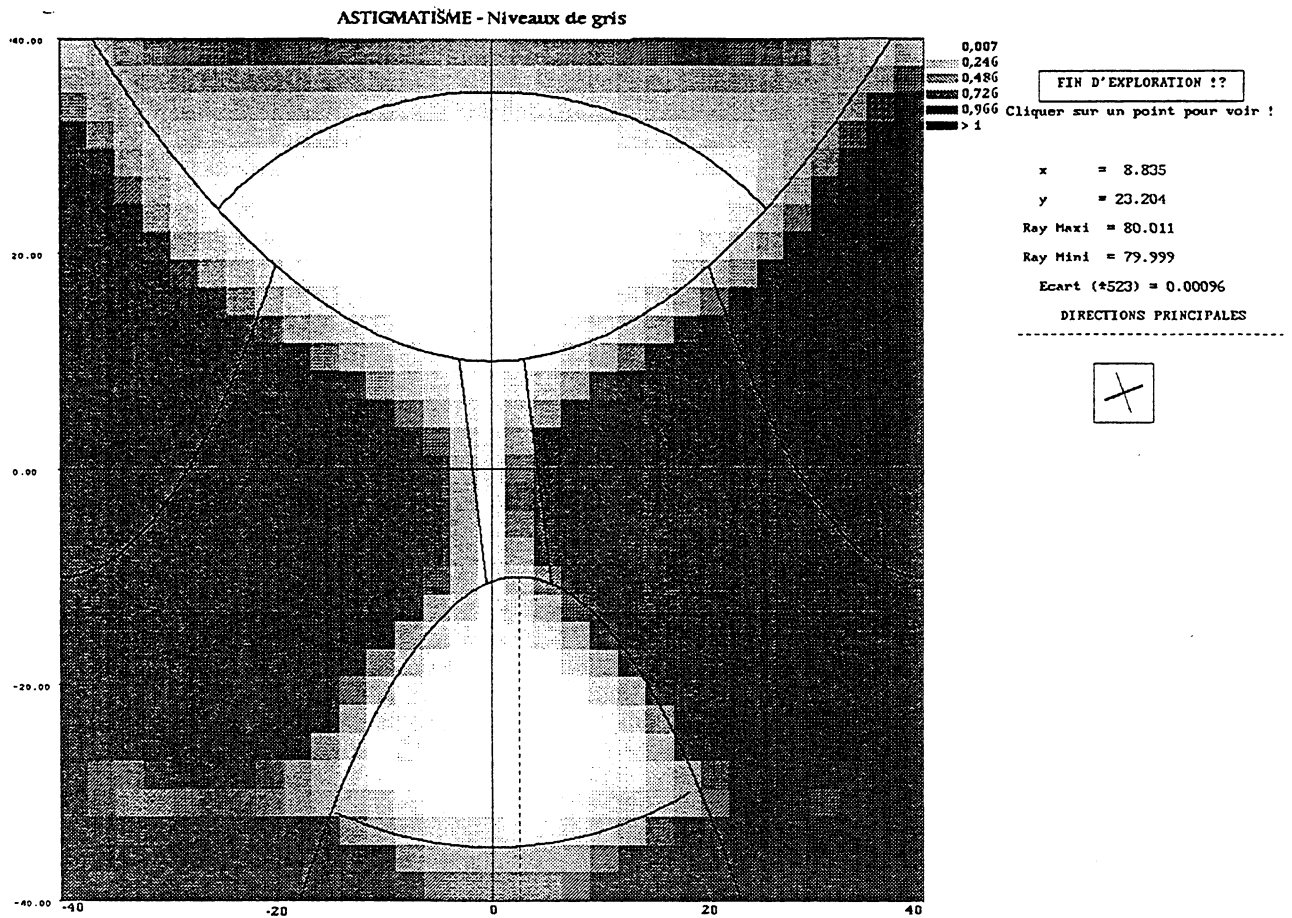


figure 9.22 Table d'astigmatisme - Examen détaillé

Chaque fois qu'on clique sur un point de la table, les coordonnées de ce point apparaissent ainsi que la valeur exacte de l'astigmatisme et des directions principales en ce point. On peut ainsi, examiner d'une façon très précise, les résultats auxquels on a abouti.

9.7 Remarque

- Degré des B-splines :

On a utilisé, pour la représentation du verre progressif, une fonction spline combinaison linéaire de B-splines de degré 4, qui est donc une fonction au moins de classe \mathcal{C}^3 . Ceci pour avoir des courbures qui soient au moins de classe \mathcal{C}^1 .

9.8 Conclusion

On a construit une méthode pour concevoir un verre progressif. D'après l'avis des opticiens qui nous ont confié ce travail, cette méthode ainsi que les résultats obtenus, sont très satisfaisants.

Le modèle qu'on propose présente l'inconvénient de nécessiter le choix d'un grand nombre de paramètres. On ne peut pas déterminer une méthode automatique pour le choix de ces paramètres, car ils dépendent des propriétés que l'on désire sur le verre.

Le grand nombre de paramètres est aussi un avantage du modèle car il lui permet une grande souplesse et lui permet aussi de concevoir différents type de verre.

La représentation du verre sous forme produit tensoriel de B-splines est facilement transformable en une représentation sous forme surface de Bézier. On peut donc le manipuler aisément avec tout les outils de CAO, et avec les algorithmes de lancer de rayons. Ceci constitue une prochaine étape dans ce travail.

Bibliographie

- [1] - **Bernstein M.S.** '*Sur les surfaces définies au moyen de leur courbure moyenne ou totale*'
Ann. Ecole Normale., 3 pp ,(XXVII), (Mai 1910).
- [2] - **Buja A., Hastie T., Tibshirani R.** '*Linear smoothers and additive models*'
Ann. Stat, 17, pp 453-554, (1989).
- [3] - **Ciarlet P.G.** '*Introduction à l'analyse numérique matricielle et à l'optimisation*'
Masson, (1988).
- [4] - **de Boor C.** '*A practical guide to splines*'
Appl. Math. Sci., 27, New York Springer, (1978).
- [5] - **Friedman J.H., Grosse E., Stuetzle W.** '*Multidimensional additive spline approximation*'
SIAM J. Sci. Stat. Comp., 4, pp 291-301, (1983).
- [6] - **Friedman J.H., Stuetzle W.** '*Projection pursuit regression*'
J. Amer. Stat. Assoc., 76, pp 817-823, (1983).
- [7] - **Girard D.** '*Practical optimal regularization of large linear systems*'
M2AN, RAIRO, Anal. Num., 20, pp 75-87, (1986).
- [8] - **Girard D.** '*Un algorithme rapide pour le calcul de la trace d'une grande matrice*'
R.R. 665-M, IMAG, Univ. de Grenoble, (1987).
- [9] - **Girard D.** '*Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille*'
R.R. 669-M, IMAG, Univ. de Grenoble, (1987).
- [10] - **Girard D.** '*Détection de discontinuités dans un signal (ou une image) par inf-convolution spline et validation croisée*'
RR 702 - I- M, IMAG, Univ. de Grenoble, (1988).
- [11] - **Girard D., Laurent P.J.** '*Spline and estimation of non linear parameters*'
In "Mathematical methods in CAGD", Academic Press, N. Y., pp 273-298, (1989).
- [12] - **Golub G.H., Heath M., Wahba G.** '*Generalized cross validation as a method for choosing a good ridge parameter*'
Technometrics, 21, pp 215-224, (1979).
- [13] - **Golub G.H., Van Loan C.F.** '*Matrix Computations*'
The Johns Hopkins Univ. Press, Baltimore, (1983).

Bibliographie

- [14] - **Graven P., Wahba G.** '*Smoothing noisy data with spline functions*'
Num. Math, 31, pp 377-403, (1979).
- [15] - **Gu C.** '*RKPACK- A general purpose minipackage for spline modeling*'
T.R. 832, Univ. Winconsin, (1988).
- [16] - **Gu C., Bates D.M, Chen Z., Wahba G.** '*The computation of GCV function through Householder tridiagonalisation with application to the fitting of interaction spline models*'
SIAM J. Matrix Anal. 10, 457-480, (1988).
- [17] - **Hastie T. Tibshirani R.** '*Generalized additive models : Some applications*'
J. Amer. Stat. Assoc, 82, pp 371-386, (1987).
- [18] - **Hubert P.J.** '*Projection pursuit*'
Ann. Statist., 13, pp 435-475, (1985).
- [19] - **Huchinson M.F., De Hoog F.R.** '*Smoothing noisy data with spline functions*'
Numer. Math., 47, 99-106, (1985).
- [20] - **Ker-Chau L.** '*From Stein's unbiased risk estimates to the method of generalized cross validation*'
Ann. Stat., 13, 4, pp 1352-1377, (1985).
- [21] - **Lasscaux P., Théodor R.** '*Analyse numérique matricielle appliquée à l'art de l'ingénieur*'
Masson, (1987).
- [22] - **Laurent P.J.** '*Approximation et optimisation*'
Herman Paris, (1972).
- [23] - **Laurent P.J.** '*Inf-convolution spline pour l'approximation de données discontinues*'
R.R. 270, IMAG, Université de Grenoble, (1981).
- [24] - **Laurent P.J.** '*Inf-convolution spline pour l'approximation de données discontinues*'
M2AN, RAIRO Anal. Num., 20, pp 89-111, (1986).
- [25] - **Laurent P.J. Utreras F.** '*Optimal smoothing of noisy broken data*'
J. of Approx. Theory and Appli., 2, pp 71-94, (1986).
- [26] - **Laurent P.J.** '*Courbes ouvertes et fermées par Bsplines régularisées*'
R.R. 652, IMAG, Université de Grenoble, (1987).
- [27] - **Laurent P.J.** '*Inf-convolution splines*'
Const. Approx., 7, pp 469-484, (1991).
- [28] - **Lelong Ferand, Arnaudès.** '*Géométrie et cinématique*'
Dunod, Paris, (1975).

Bibliographie

- [29] - **Paihua L.** *'Quelques méthodes numériques pour les fonctions splines à une ou deux variables'*
Thèse, Université de Grenoble, (1978).
- [30] - **Reinsch C.M.** *'Smoothing by spline functions'*
Numer. Math., 10, pp 173-183, (1967).
- [31] - **Reinsch C.M.** *'Smoothing by spline functions'*
Numer. Math., 16, pp 451-454, (1971).
- [32] - **Silverman B.W.** *'A fast and efficient cross-validation method for smoothing parameter choice in spline regression'*
J. Amer. Stat. Ass., 79, pp 584-589, (1984).
- [33] - **Stone C.J.** *'Additive regression and other nonparametric models'*
The Ann. Stat., 13, pp 689-705, (1985).
- [34] - **Utreras F.** *'Utilisation de la méthode de validation croisée pour le lissage des fonctions à une ou deux variables'*
Thèse, Université de Grenoble, (1979).
- [35] - **Utreras F.** *'Sur le choix du paramètre d'ajustement dans le lissage par fonctions splines'*
Numer. Math., 34, pp 15-28, (1980).
- [36] - **Utreras F.** *'Optimal smoothing of noisy data using spline functions'*
SIAM J. Sci. Stat. Comp., 2, pp 349-362, (1981).
- [37] - **Utreras F.** *'Constrained surface construction'*
Topics in Multivariate Approximation., Academic Press Inc, pp 233-254, (1987).
- [38] - **Wahba G.** *'Smoothing noisy data with spline functions'*
Num. Math, 24, pp 383-393, (1975).
- [39] - **Wahba G.** *'Partial and interactive splines for the semiparametric estimation of functions of several variables'*
In Computer Science and Statistics: Proc. Eighteenth Symp. on the interface (T.J.Boardman, ed.) 75-80. Amer.Statist., Alexandria, Va., (1986).

