



Sur l'approximation et la représentation de surfaces explicites en présence de singularités

Pascal Klein

► To cite this version:

Pascal Klein. Sur l'approximation et la représentation de surfaces explicites en présence de singularités. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1987. Français. NNT: . tel-00325040

HAL Id: tel-00325040

<https://theses.hal.science/tel-00325040>

Submitted on 26 Sep 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.

Thèse préparée au Laboratoire TIM3

THESE

présentée à

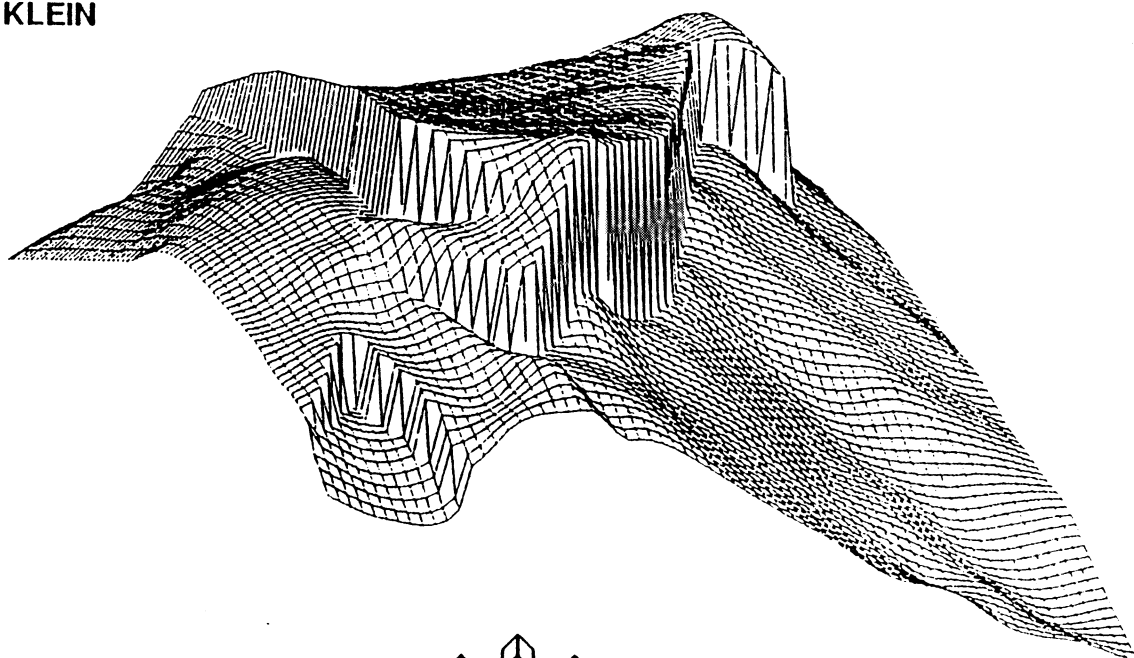
l'Université Scientifique Technologique et Médicale de Grenoble

pour obtenir le grade de

Docteur de 3^{ième} Cycle
Mathématiques Appliquées

par

Pascal KLEIN



**SUR L'APPROXIMATION ET LA REPRESENTATION
DE SURFACES EXPLICITES EN PRESENCE DE
SINGULARITES**



Thèse soutenue le 29 septembre 1987 devant la Commission d'Examen :

Président : P.J. LAURENT

**Examineurs : R. ARCANGELI
P. CHENIN
M. DUC-JACQUET
J. THOMANN**

UNIVERSITE SCIENTIFIQUE TECHNOLOGIQUE ET MEDICALE DE GRENOBLE

Président de l'Université :
M. TANCHE

Année Universitaire 1986 - 1987

MEMBRES DU CORPS ENSEIGNANT DE SCIENCES ET DE GEOGRAPHIE

PROFESSEURS DE 1ère Classe

ARNAUD Paul
ARVIEU ROBERT
AUBERT Guy
AURIAULT Jean-Louis
AYANT Yves
BARBIER Marie-Jeanne
BARBIER Jean-Claude
BARJON Robert
BARNOUD Fernand
BARRA Jean-René
BELORISKY Elie
BENZAKEN Claude
BERNARD Alain
BERTRANDIAS Françoise
BERTRANDIAS Jean-Paul
BILLET Jean
BOELHER Jean-Paul
BONNIER Jane Marie
BOUCHEZ Robert
BRAVARD Yves
CARLIER Georges
CAUQUIS Georges
CHIBON Pierre
COHEN ADDAD Jean-Pierre
COLIN DE VERDIERE Yves
CYROT Michel
DEBELMAS Jacques
DEGRANGE Charles
DELOBEL Claude
DEPORTES Charles
DESRE Pierre
DOLIQUE Jean-Michel
DOUCE Rolland
DUCROS Pierre
FONTAINE Jean-Marc
GAGNAIRE Didier
GERMAIN Jean-Pierre
GIRAUD Pierre
HICTER Pierre
IDELMAN Simon
JANIN Bernard
JOLY Jean-René
KAHANE André, détaché
KAHANE Josette
KRAKOWIAK Sacha
KUPKA Yvon
LAJZEROWICZ Jeanine
LAJZEROWICZ Joseph
LAURENT Pierre-Jean
DE LEIRIS Joel

Chimie Organique
Physique Nucléaire I.S.N.
Physique C.N.R.S
Mécanique
Physique Approfondie
Electrochimie
Physique Expérimentale CNRS
Physique Nucléaire ISN
Biochimie Macromoléculaire Végétale
Statistiques-Mathématiques Appliquées
Physique C.E.N.G- D.R.F.
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Mathématiques Pures
Géographie
Mécanique
Chimie Générale
Physique Nucléaire ISN
Géographie
Biologie Végétale
Chimie Organique
Biologie Animale
Physique
Mathématiques Pures
Physique du Solide
Géologie Générale
Zoologie
Mathématiques Appliquées
Chimie Minérale
Electrochimie
Physique des Plasmas
Physiologie Végétale
Cristallographie
Mathématiques Pures
Chimie Physique
Mécanique,
Géologie
Chimie
Physiologie Animale
Géographie
Mathématiques Pures
Physique
Physique
Mathématiques Appliquées
Mathématiques Pures
Physique
Physique
Mathématiques Appliquées
Biologie

LLIBOUTRY Louis
 LOISEAUX Jean-Marie
 MACHE Régis
 MAYNARD Roger
 MICHEL Robert
 OMONT Alain
 OZENDA Paul
 PAYAN Jean-Jacques
 PEBAY-PEYROULA Jean-Claude
 PERRIAUX Jacques
 PERRIER Guy
 PIERRARD Jean-Marie
 PIERRE Jean-Louis
 RASSAT André
 RENARD Michel
 RINAUDO Marguerite
 ROSSI André
 SAKAROVITCH Michel
 SAXOD Raimard
 SENGEL Philippe
 SERGERAERT Francis
 SOUCHIER Bernard
 SOUTIF Michel
 STUTZ Pierre
 VALENTIN Jacques
 VAN CUTSEM Bernard
 VIALON Pierre

Géophysique
 Sciences Nucléaires I.S.N.
 Physiologie Végétale
 Physique du Solide
 Minéralogie et Pétrographie (Géologie)
 Astrophysique
 Botanique (Biologie Végétale)
 Mathématiques Pures
 Physique
 Géologie
 Géophysique
 Mécanique
 Chimie Organique
 Chimie Systématique
 Thermodynamique
 Chimie CERMAV
 Biologie
 Mathématiques Appliquées
 Biologie Animale
 Biologie Animale
 Mathématiques Pures
 Biologie
 Physique
 Mécanique
 Physique Nucléaire I.S.N.
 Mathématiques Appliquées
 Géologie

PROFESSEURS de 2^{ème} Classe

ADIBA Michel
 ANTOINE Pierre
 ARMAND Gilbert
 BARET Paul
 BECKER Pierre
 BEGUIN Claude
 BLANCHI J. Pierre
 BOITET Christian
 BORNAREL Jean
 BRUANDET J. François
 BRUN Gilbert
 CASTAING Bernard
 CERFF Rudiger
 CHIARDON Michel
 CHIARAMELLA Yves
 COURT Jean
 DEMAILLY Jean-Pierre
 DENEUVILLE Alain
 DEPASSEL Roger
 DERRIEN Jacques
 DUFREYNOY Alain
 GASPARD François
 GAUTRON René
 GENIES Eugène
 GIDON Maurice
 GIGNOUX Claude
 GILLARD Roland
 GIORNI Alain
 GUIGO Maryse
 GUMUCHAIN Hervé
 GUITTON Jacques
 HACQUES Gérard

Mathématiques Pures
 Géologie
 Géographie
 Chimie
 Physique
 Chimie Organique
 STAPS
 Mathématiques Appliquées
 Physique
 Physique
 Biologie
 Physique
 Biologie
 Géographie
 Mathématiques Appliquées
 Chimie
 Mathématiques Pures
 Physique
 Mécanique des Fluides
 Physique
 Mathématiques Pures
 Physique
 Chimie
 Chimie
 Géologie
 Sciences Nucléaires
 Mathématiques Pures
 Sciences Nucléaires
 Géographie
 Géographie
 Chimie
 Mathématiques Appliquées

HERBIN Jacky
 HERAULT Jeanny
 JARDON Pierre
 JOSELEAU Jean-Paul
 KERCKHOVE Claude
 LEBRETON Alain
 LONGEQUEUE Nicole
 LUCAS Robert
 LUNA Domingo
 MANDARON Paul
 MARTINEZ Francis
 MASCLE Georges
 NEMOZ Alain
 OUDET Bruno
 PELMONT Jean
 PERRIN Claude
 PFISTER Jean-Claude
 PIBOULE Michel
 RAYNAUD Hervé
 RIEDIMANN Christine
 ROBERT Gilles
 ROBERT Jean-Bernard
 SARROT-REYNAULD Jean
 SAYETAT Françoise
 SERVE Denis
 STOECKEL Frédéric
 SOUTIF Jeanne
 SCHOLL Pierre-Claude
 SUBRA Robert
 VALLADE Marcel
 VIDAL Michel
 VIVIAN Robert
 VOTTERO Philippe

Géographie
 Physique
 Chimie
 Biochimie
 Géologie
 Mathématiques Appliquées
 Sciences Nucléaires I.S.N.
 Physique
 Mathématiques Pures
 Biologie
 Mathématiques Appliquées
 Géologie
 Thermodynamique CNRS - CRTBT
 Mathématiques Appliquées
 Biochimie
 Sciences Nucléaires I.S.N.
 Physique du Solide
 Géologie
 Mathématiques Appliquées
 Mathématiques Pures
 Mathématiques Pures
 Chimie Physique
 Géologie
 Physique
 Chimie
 Physique
 Physique
 Mathématiques Appliquées
 Chimie
 Physique
 Chimie Organique
 Géographie
 Chimie

MEMBRES DU CORPS ENSEIGNANT DE L' IUT 1

PROFESSEURS de 1^{ère} Classe

BUISSON Roger
 DODU Jacques
 NEGRE Robert

Physique IUT 1
 Mécanique Appliquée IUT 1
 Génie Civil IUT 1

PROFESSEURS de 2^{ème} classe

BOUTHINON Michel
 CHAMBON René
 CHEHIKIAN Alain
 CHENAVAS Jean
 CHOUTEAU Gérard
 CONTE René
 GOSSE Jean-Pierre
 GROS Yves
 KUHN Gérard, (Détaché)
 MAZUER Jean
 MICHOUlier Jean
 MONLLOR Christian
 NOUGARET Marcel
 PEFFEN René
 PERARD Jacques
 PERRAUD Robert
 TERRIEZ Jean-Michel
 TOUZAIN Philippe
 VINCENDON Marc

EEA. IUT 1
 Génie Mécanique IUT 1
 EEA. IUT 1
 Physique IUT 1
 Physique IUT 1
 Physique IUT 1
 EEA.IUT 1
 Physique IUT 1
 Physique IUT 1
 Physique IUT 1
 Physique IUT 1
 EEA.IUT 1
 Automatique IUT 1
 Métallurgie IUT 1
 EEA. IUT 1
 Chimie IUT 1
 Génie Mécanique IUT 1
 Chimie IUT 1
 Chimie IUT 1

MEMBRES DU CORPS ENSEIGNANT DE MEDECINE

PROFESSEURS CLASSE EXCEPTIONNELLE ET 1ère CLASSE

AMBLARD Pierre	Dermatologie	C.H.R.G.
AMBROISE-THOMAS Pierre	Parasitologie	C.H.R.G.
BEAUDOING André	Pédiatrie-Puericulture	C.H.R.G.
BEZEZ Henri	Orthopédie-Traumatologie	Hopital SUD
BONNET Jean-Louis	Ophthalmologie	C.H.R.G.
BOUCHET Yves	Anatomie	Faculté La Merci
	Chirurgie Générale et Digestive	C.H.R.G.
BUTEL Jean	Orthopédie-Traumatologie	C.H.R.G.
CHAMPETIER Jean	Anatomie-Topographique et Appliquée	C.H.R.G.
	O.R.L.	C.H.R.G.
CHARACHON Robert	Anatomie-Pathologique	C.H.R.G.
COUDERC Pierre	Pneumophthisiologie	C.H.R.G.
DELORMAS Pierre	Cardiologie	C.H.R.G.
DENIS Bernard	Pharmacologie	Faculté La Merci
GAVEND Michel	Hématologie	C.H.R.G.
HOLLARD Daniel	Chirurgie Thoracique et Cardiovasculaire	C.H.R.G.
LATREILLE René	Bactériologie-Virologie	C.H.R.G.
	Gynécologie et Qbstétrique	C.H.R.G.
LE NOC Pierre	Médecine du Travail	C.H.R.G.
MALINAS Yves	Clinique Médicale et Maladies Infectieuses	C.H.R.G.
MALLION Jean-Michel	Histologie	Faculté La Merci
MICOUD Max	Pneumologie	C.H.R.G.
	Neurologie	C.H.R.G.
MOURIQUAND Claude	Hépat-Gastro-Entérologie	C.H.R.G.
PARAMELLE Bernard	Neurochirurgie	C.H.R.G.
PERRET Jean	Clinique Chirurgicale	C.H.R.G.
RACHAIL Michel	Anesthésiologie	C.H.R.G.
DE ROUGEMONT Jacques	Physiologie	Faculté La Merci
SARRAZIN Roger	Biophysique	Faculté La Merci
STIEGLITZ Paul	Biochimie	Faculté La Merci
TANCHE Maurice		
VERAIN André		
VIGNAIS Pierre		

PROFESSEURS 2ème CLASSE

BACHELOT Yvan	Endocrinologie	C.H.R.G.
BARGE Michel	Neurochirurgie	C.H.R.G.
BENABID Alim Louis	Biophysique	Faculté La Merci
BENSA Jean-Claude	Immunologie	Hopital Sud
BERNARD Pierre	Gynécologie-Obstétrique	C.H.R.G.
BESSARD Germain	Pharmacologie	ABIDJAN
BOLLA Michel	Radiothérapie	C.H.R.G.
BOST Michel	Pédiatrie	C.H.R.G.
BOUCHARLAT Jacques	Psychiatrie Adultes	Hopital Sud
BRAMBILLA Christian	Pneumologie	C.H.R.G.
CHAMBAZ Edmond	Biochimie	C.H.R.G.
CHIROSSSEL Jean-Paul	Anatomie-Neurochirurgie	C.H.R.G.
COLOMB Maurice	Immunologie	Hopital Sud
COMET Michel	Biophysique	Faculté La Merci
CONTAMIN Charles	Chirurgie Thoracique et Cardiovasculaire	C.H.R.G.
	Néphrologie	C.H.R.G.
CORDONNIER Daniel	Radiologie	C.H.R.G.
COULOMB Max	Radiologie	C.H.R.G.
CROUZET Guy	Médecine Interne et Toxicologie	C.H.R.G.
DEBRU Jean-Luc	Biostatistiques et Informatique Médicale	Faculté La Merci
DEMONGEOT Jacques		

DUPRE Alain
 DYON Jean-François
 ETERRADOSSI Jacqueline
 FAURE Claude
 FAURE Gilbert
 FOURNET Jacques
 FRANCO Alain
 GIRARDET Pierre
 GUIDICELLI Henri
 GUIGNIER Michel

HADJIAN Arthur
 HALIMI Serge

HOSTEIN Jean
 HUGONOT Robert
 JALBERT Pierre
 JUNIEN-LAVILLAUROY Claude
 KOLODIE Lucien
 LETOUBLON Christian
 MACHECOURT Jacques

MAGNIN Robert
 MASSOT Christian
 MOUILLON Michel
 PELLAT Jacques
 PHELIP Xavier
 RACINET Claude
 RAMBAUD Pierre
 RAPHAEL Bernard
 SCHAERER René
 SEIGNEURIN Jean-Marie
 SELE Bernard
 SOTTO Jean-Jacques
 STOEBSNER Pierre
 VROUSOS Constantin

Chirurgie Générale
 Chirurgie Infantile
 Physiologie
 Anatomie et Organogénèse
 Urologie
 Hépatogastro-Entérologie
 Médecine Interne
 Anesthésiologie
 Chirurgie Générale et Vasculaire
 Thérapeutique et Réanimation
 Médicale
 Biochimie
 Endocrinologie et Maladies
 Métaboliques
 Hépatogastro-Entérologie
 Médecine Interne
 Histologie-Cytogénétique
 O.R.L.
 Hématologie Biologique
 Chirurgie Générale
 Cardiologie et Maladies
 Vasculaires
 Hygiène
 Médecine Interne
 Ophtalmologie
 Neurologie
 Rhumatologie
 Gynécologie
 Pédiatrie
 Stomatologie
 Cancérologie
 Bactériologie-Virologie
 Cytogénétique
 Hématologie
 Anatomie Pathologique
 Radiothérapie

C.H.R.G.
 C.H.R.G.
 Faculté La Merci
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.

C.H.R.G.
 Faculté La Merci

C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.

C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.
 Faculté La Merci
 Faculté La Merci
 C.H.R.G.
 C.H.R.G.
 C.H.R.G.

MEMBRES DU CORPS ENSEIGNANT PHARMACIE

AGNIUS-DELORD Claudine
 ALARY Josette
 BERIEL Hélène
 BOUCHERLE André
 CUSSAC Max
 DEMENGE Pierre
 JEANNIN Charles
 LATURAZE Jean
 LUU DUC Cuong
 MARIOTTE Anne-Marie
 MARZIN Daniel
 RENAUDET Jacqueline
 ROCHAT Jacques
 SEIGLE-MURANDI Françoise
 VERRAIN Alice

Physique
 Chimie Analytique
 Physiologie et Pharmacologie
 Chimie et Toxicologie
 Chimie Thérapeutique
 Pharmacodynamie
 Pharmacie Galénique
 Biochimie
 Chimie Générale
 Pharmacognosie
 Toxicologie
 Bactériologie
 Hygiène et Hydrologie
 Botanique et Cryptogamie
 Pharmacie Galénique

Faculté La Tronche
 Faculté La Tronche
 Faculté La Tronche
 Faculté Meylan
 Faculté La Tronche
 Faculté La Tronche
 Faculté Meylan
 Faculté La Tronche
 Faculté La Tronche
 Faculté La Tronche
 Faculté Meylan
 Faculté La Tronche
 Faculté La Tronche
 Faculté Meylan
 Faculté Meylan

A Natacha .

Cette thèse a été préparée au sein du laboratoire TIM3 de Grenoble .

Je tiens à remercier ici tous les membres du jury et tous ceux qui m'ont aidé à réaliser cette thèse:

- Monsieur LAURENT P.J. , en outre d'avoir accepté de présider le jury, pour l'estime et les nombreux encouragements qu'il m' a témoignés depuis mon entrée à l'IMAG.

- Monsieur ARCANGELI R., pour après avoir accepté de juger ce travail, a bien voulu participer au jury.

- Monsieur CHENIN P., pour son aide constante et efficace tout au long de ce travail ainsi que ses qualités scientifiques et humaines.

- Monsieur DUC-JACQUET pour ses nombreuses suggestions et ses discussions fructueuses.

- Monsieur THOMANN J., pour l'intérêt qu'il a porté à ce travail et pour avoir accepté de se déplacer pour juger cette thèse.

Enfin tous mes amis et proches sans qui rien de ce qui suit n'aurait été écrit:

A. EBERHARD (pour m'avoir prodigué de nombreux conseils en informatique), D.PELLEGRIN pour ses "coups de main", J.M. MULLER pour sa jovialité et son humour, M.H. ACQUIER et Y. PAILLET.

Mention spéciale à ma femme Catherine pour son aide et son soutien moral tout au long de ce travail.

Je remercie également ce brave Macintosh, sur laquelle j' ai frappé cette thèse et termine ici ces dernières l.i.g.n.e.s. Enfin, j'adresse mes vifs remerciements au Service de Reprographie de l'IMAG pour le tirage de cette thèse.

PLAN DE LA THESE

Chapître I : *Un tour d'horizon*

I-A Introduction.....	page I-1
I-B Méthodes de différences finies et d'estimation linéaire.....	page I-2
I-B-1 Généralités.....	page I-2
I-B-2 Approches déterministes:cas harmonique et biharmonique....	page I-3
I-C Quadriques de Shepard modifiées.....	page I-9
I-C-1 Quadriques de Shepard.....	page I-9
I-C-2 Prise en compte des discontinuités.....	page I-11
I-D Double interpolation linéaire.....	page I-14
I-D-1 Double interpolation linéaire.....	page I-14
I-D-2 Prise en compte des discontinuités.....	page I-16
I-E Méthodes de régularisation et éléments finis.....	page I-18
I-E-1 Généralités.....	page I-18
I-E-2 D^m -spline d'interpolation et d'ajustement.....	page I-21
I-E-3 Surface plaque mince sous tension et continuité contrôlée....	page I-26

Chapître II : *Application de l'inf-convolution spline pour l'approximation de surfaces singulières*

II -A Introduction.....	page II-1
II -B Inf-convolution spline.....	page II-3
II -B-1 Sous-espaces semi-hilbertiens.....	page II -3
II -B-2 Semi-noyau.....	page II -4
II -B-3 Définition,caractérisation et calcul des fonctions-spline.....	page II -5
II -B-4 Définition et caractérisation de l'inf-convolution spline d'interpolation.....	page II -7
II -B-5 Définition et caractérisation de l'inf-convolution spline d'ajustement.....	page II -11
II -C Utilisation de l'inf-convolution spline pour l'approximation de surfaces singulières.....	page II-13
II -C-1 Contexte.....	page II-13
II -C-2 Modélisation.....	page II-14
II -C-3 Choix du sous-espace V des discontinuités.....	page II-14
II -C-4 Algorithme de calcul de l'inf-convolution spline d'interpolation ou d'ajustement.....	page II-17
II -D Exemples et conclusions.....	page II-21

Chapitre III : *Un algorithme de tracé automatique de courbes isovaleurs pour des surfaces singulières*

III-A Introduction.....	page III-1
III-B Présentation de l'algorithme.....	page III-2
III-B-1 Notations, définitions et hypothèses.....	page III-2
III-B-2 Principe de l'algorithme.....	page III-4
III-B-3 Codage des cellules.....	page III-5
III-B-4 Détection et tracé d'une composante.....	page III-9
III-B-5 Tableau synthétique de l'algorithme.....	page III-18
III-C Exemples et conclusions.....	page III-21
III-C-1 Exemples.....	page III-21
III-C-2 Conclusions.....	page III-27

Chapitre IV : *SUREX: un logiciel d'approximation de surfaces explicites*

IV-A Introduction.....	page IV-1
IV-B Le logiciel SAISIE.....	page IV-2
IV-B- 1 Description et fonctionnalités du programme principal SAISIE.....	page IV-3
IV-B-2 Description et fonctionnalités du sous-programme ENTREE.....	page IV-4
IV-B-3 Description et fonctionnalités du sous-programme MODIFICATION.....	page IV-6
IV-C Le logiciel CALCUL.....	page IV-9
IV -C-1 Description du programme INFCONV.....	page IV-9
IV -C-2 Description du programmeFRANIELS.....	page IV-11
IV-D Le logiciel VISUAL.....	page IV-12
IV-D-1 Le programme COURNIV.....	page IV-12
IV-D-1 Le programme TRASURF.....	page IV-12
IV-E Le logiciel ERREUR.....	page IV-14
IV-F Quelques exemples	page IV-15

INTRODUCTION GENERALE

Cette thèse est une contribution à l'étude et la théorie de l'approximation et de la représentation de surfaces explicites singulières lorsque des points de donnée sont connus. On désire que ces surfaces soient lisses sauf pour certaines discontinuités de valeur ou de dérivée de saut fini, lesquelles sont supposées apparaître le long de lignes brisées.

Ce sujet relativement neuf dans la littérature scientifique, peu abordé dans la littérature, en raison des difficultés inhérentes à la prise en compte des discontinuités, commence à intéresser une partie de la communauté scientifique de par ses applications. De tels schémas sont fréquemment rencontrés dans plusieurs champs d'applications scientifiques. Citons quelques exemples évidents:

- CAO (Conception Assistée par Ordinateur) pour la géologie: modélisation de terrains accidentés, topographie sous-marine, strates souterrains, etc...
- Reconstructions d'images tridimensionnelles à partir de photographies aériennes.

Notre étude porte sur les discontinuités de valeur ou *failles*, et sur les discontinuités de dérivée ou *plis*. On peut modéliser l'apparition de singularités suivant trois façons que nous décrivons par type:

(1) la localisation des lignes de discontinuités est connue, avec un saut connu de celles-ci de la surface, le long de ces lignes.

(2) la localisation des lignes de discontinuités est connue, mais on ne dispose pas d'une estimation du saut de discontinuité le long de celles-ci.

(3) on ne connaît pas la localisation des lignes de discontinuités, mais toutefois les données indiquent probablement la présence de failles, à cause de forts gradients apparaissant entre celles-ci.

Le problème de type (1) peut être traité avec le même genre de techniques que le problème de type (2), bien qu'il semble qu'on puisse développer des méthodes plus appropriées pour la résolution du problème de type (1). Si dans le problème de type (2), on dispose d'une grande densité de points des deux côtés de chaque ligne, ce type est analogue au type (1). Les problèmes de type (3) requièrent un effort supplémentaire considérable: celui de localiser les failles. Ce problème de localisation est en fait intimement lié à celui de la détection de bords rencontré dans le domaine de traitements d'images. Quelques schémas sophistiqués, basés sur différentes approches mathématiques (statistiques, régularisation) ont été proposés ([Grimson 83], [Terzopoulos86]) et donnent d'assez bons résultats. En fait ce problème de localisation peut se traiter en premier lieu à part avec certaines techniques, et ensuite utiliser des méthodes appropriées pour la résolution d'un problème de type (2).

En général, les méthodes classiques d'interpolation ou d'ajustement des données ne prennent pas en compte ce type de conditions; il s'ensuit que les surfaces obtenues présentent de fortes inflexions et des oscillations indésirables qui "effacent" les

discontinuités. De même, la plupart des algorithmes de tracé de courbes isovaleurs ne tiennent pas compte des discontinuités, ce qui donne des cartes d'état-major erronées.

Dans cette thèse, nous nous intéressons au problème de type (2), dans le but d'adresser des problèmes dont la solution a le plus d'applications. Notre travail est donc une contribution à l'étude des problèmes suivants:

(P_1) Soit $\mathcal{D} = \{ (x_i, y_i, z_i) ; i = 1, \dots, n \}$ un ensemble de données dans l'espace réel, et $\mathcal{F} = \{F_i\}_{i=1, \dots, p}$ un ensemble de lignes brisées du plan. On cherche une surface définie par une fonction de deux variables S , estimée aux sommets d'une grille régulière vérifiant les conditions suivantes:

- S interpole ou ajuste les données de $\mathcal{D}$.
- S est au moins de classe C^0 sur $\mathbb{R}^2 \setminus \mathcal{F}$.
- S possède des discontinuités de valeur ou de dérivée, de saut fini, le long des éléments de $\mathcal{F}$.

(P_2) Soit une surface S connue aux sommets d'une grille régulière, tel que S admette des discontinuités de valeur le long des éléments de $\mathcal{F}$. On cherche un algorithme de tracé automatique de courbes isovaleurs de S , respectant ces conditions de discontinuité.

Dans le premier chapitre, nous faisons un tour d'horizon critique de quelques méthodes d'approximation pour le problème (P_1) proposées dans la littérature. Après avoir discuté des travaux de Bolondi, Rocca, Zanoletti ([Bolondi 76]), basées sur des méthodes déterministes (surface harmonique et biharmonique) et non déterministes (basées sur des techniques de krigeage), nous parlons de l'article de Franke et Nielson ([Franke 83]), où les auteurs développent une méthode locale d'interpolation basée sur des surfaces quadriques de Shepard modifiées; ces modifications consistent (i) à rajouter un terme local "convenable" prenant en compte les discontinuités de valeur ou de dérivée, (ii) à introduire une notion de "distance pénalisante" afin d'obtenir un schéma n'autorisant que les discontinuités de valeur. Ensuite, nous discutons d'une méthode locale d'approximation développée dans le logiciel UNIRAS ([Uniras 86]) effectuant une double interpolation linéaire, la prise en compte des failles consistant à "ignorer" les points de donnée se situant de l'autre côté de celle-ci lors de l'évaluation de S en un point de la grille. Enfin, des techniques de régularisation ([Manzanilla 86] , [Terzopoulos 86a,b]), sont discutées dans la quatrième partie: après avoir reformulé le problème (P_1) dans un cadre fonctionnel abstrait (principe variationnel), les solutions vérifiant une équation variationnelle dérivant de ce principe, sont ensuite estimées à l'aide d'une discrétisation par éléments finis.

La théorie de l'inf-convolution spline développée par Laurent ([Laurent 86a]) fonde le deuxième chapitre. Elle a déjà été mise en œuvre dans le cadre monodimensionnel par l'auteur pour le traitement de dérivées discontinues (cf ibid.), et

par Valera ([Valera 84]) pour l'apparition de pics dans le traitement de chromatogrammes de gasoils. Cette théorie étant assez générale, il nous est apparu intéressant de l'appliquer dans le cadre bidimensionnel, au problème (P_1) . Cette technique revient ici à choisir un espace fonctionnel V "traduisant" les discontinuités de valeur ou de dérivée, ce qui est réalisé en proposant une base de fonctions de V . Ce choix donne d'excellents résultats illustrés par quelques exemples, quant à l'approximation numérique.

La visualisation d'une surface explicite S s'effectue de deux façons: (i) représentation tridimensionnelle, (ii) courbes isovaleurs, ces deux visualisations étant complémentaires afin de bien appréhender le comportement local de S . Jusqu'à ce jour, peu d'algorithmes ont été développés pour la prise en compte explicite de discontinuités. C'est la raison pour laquelle dans le troisième chapitre, nous proposons un nouvel algorithme, basé sur une technique de "suivi" des composantes des courbes de niveau, dont l'illustration concluante est effectuée à la fin de ce chapitre.

Enfin dans le chapitre quatre, nous présentons succinctement le logiciel SUREX, d'approximation de surfaces explicites, écrit en langage PASCAL et développé sur micro-ordinateur compatible. Ce logiciel est composé de quelques modules: programme d'acquisition de données nécessaires ($\mathcal{D}$ et $\mathcal{F}$), quelques programmes d'approximation mettant en œuvre quelques techniques ci-dessus, et d'un programme de calculs d'erreurs diverses. Enfin nous illustrons le fonctionnement logiciel en traitant quelques exemples académiques.

CHAPITRE I :

UN TOUR D'HORIZON

I - 1

I - A Introduction

Dans ce chapitre, nous faisons un tour d'horizon sur les méthodes d'approximation de surfaces connues en un certain nombre de points et présentant des discontinuités de valeurs ou de dérivées le long de lignes polygonales. Nous pouvons classer ces méthodes suivant quatre types:

- 1 différences finies
- 2 éléments finis
- 3 approximation locale
- 4 estimation linéaire

Toutes les techniques présentées ici apportent des éléments de réponses au problème et donnent des résultats plus ou moins satisfaisants quant à la qualité du résultat de l'approximation. De plus la mise en œuvre de certaines d'entre elles sont difficiles et ont un temps d'exécution parfois exorbitant.

Dans le paragraphe B sont présentés les travaux de Bolondi, Rocca, Zanoletti (**[BOLONDI 77]**). Les deux premières approches, qualifiées de méthodes déterministes sont du type 1 et font des hypothèses de régularité sur la surface à approcher. La dernière approche est du type 3 et est basée sur le krigeage. Les auteurs y introduisent la notion très séduisante de "distance pénalisante" qui généralise dans un certain sens la notion de distance euclidienne, prise au sens de la longueur du plus court chemin joignant deux points et ne coupant pas les lignes polygonales.

Le paragraphe C est consacré aux travaux de R. Franke et G. Nielson (**[FRANKE 83]**). Les auteurs prennent pour base d'approximation les surfaces quadriques de Shepard auxquelles sont incorporées des termes caractérisant le comportement anormal de la surface au voisinage des lignes polygonales. De plus, l'utilisation de la notion de "distance pénalisante" permet d'obtenir un nouveau schéma. Toutes ces méthodes sont du type 3.

Nous décrivons dans le paragraphe D une méthode décrite dans le logiciel de cartographie UNIRAS(**[UNIRAS 86]**), qui sont de type 3, permettant de prendre les discontinuités de valeur d'une surface le long des lignes brisées.

Enfin, dans le dernier paragraphe nous parlons de techniques de régularisation qui sont de type 2. En premier lieu, nous parlons des travaux de Manzanilla(**[Manzanilla 86]**) qui apportent une réponse au problème de surfaces discontinues en formulant un problème continu en utilisant la théorie des espaces de Sobolev et des fonctions-spline au cas d'ouverts bornés à frontière non lipschitzienne. Ensuite, nous discutons des travaux de D. Terzopoulos (**[Terzopoulos 86a]**) qui sont basés sur les techniques de régularisation standard de Tikhonov pour la résolution de problèmes mal posés, résolution utilisant des techniques multigrilles.

I - 2

I - B Méthodes de différences finies et d'estimation linéaire.

Nous faisons référence ici à l'article de G. Bolondi, F. Rocca et S. Zanoletti ([Bolondi 77]). Les auteurs proposent (i) une approche déterministe qui est considérée dans les deux cas d'équations aux dérivées partielles harmonique ou biharmonique et (ii) une approche stochastique basée sur des techniques de krigeage qui évite les écueils des méthodes précédentes et permet une extension aux surfaces discontinues.

I-B-1 Généralités.

On se donne une surface explicite $z = z(x,y)$ mesurée en n points de données distincts

$$z(x_i, y_i) = z_i \quad i = 1, \dots, n$$

De plus, on sait que le long de M lignes polygonales F_i connues, de longueur finie, z présente des discontinuités de valeurs. Le but est de retrouver la valeur de la surface aux sommets d'une grille G régulière de taille Δ englobant les données:

$$z_{p,q} = z(x_0 + p h, y_0 + q h) \quad p = 1, \dots, N_x ; q = 1, \dots, N_y$$

Trois critères essentiels sont retenus pour caractériser la surface:

- a) précision
- b) régularité
- c) amplitude

La précision est complète si les courbes isovaleurs de z touchent les points de données, même lorsque ceux-ci ne sont pas aux sommets de la grille. La seule façon de garantir un tel comportement est d'introduire une grille irrégulière dont les noeuds sont ceux de G et les points de données. Cette précision augmente lorsque la taille h diminue. Donc ici, nous aurons une précision limitée dépendant de h .

La régularité de f est fortement liée à l'algorithme d'interpolation choisi. Dans le cas de fonctions harmoniques et quelquefois en utilisant la méthode stochastique, la surface tend à être "pointue" aux voisinages de points de données. Dans ces deux cas, la surface tend à prendre des valeurs moyennes entre celles des mesures.

L'amplitude des valeurs prises par la surface peut excéder celles des données; ce fait peut être un inconvénient particulièrement lorsque des structures fictives générées sont dues à de forts gradients des données. D'un autre côté, l'amplitude peut créer une dégradation du lissage de la surface de telle façon qu'un compromis doive être choisi entre la régularité et l'amplitude de la solution.

I - 3

I-B-2 Approches déterministes: cas harmonique et biharmonique

I-B-2-a Cas harmonique

Une manière simple de générer un ensemble de valeurs $z_{p,q}$ est le suivant: une correspondance est établie entre la surface et une membrane fixée aux hauteurs des points de données et coupée le long des lignes F_i . La hauteur de la surface est prise proportionnelle à la hauteur de la position d'équilibre de la surface. Ceci revient à dire que la surface $z(x,y)$ satisfait l'équation harmonique:

$$\Delta z = \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = 0$$

pour $(x,y) \neq (x_i, y_i) \quad i = 1, \dots, n$

et pour $(x,y) \notin F_j \quad j = 1, \dots, M$

De plus nous imposons que $z(x_i, y_i) = z_i \quad i = 1, \dots, n$. Une interprétation physique de ce problème est de trouver le potentiel électrique d'une surface conductrice, coupée le long des lignes F_j , et proprement connectée aux générateurs d'électricité correspondant aux données. Il faut encore définir le comportement de z sur la frontière de la grille G : une façon simple est de supposer que la dérivée normale de la surface est nulle le long des frontières.

Les algorithmes de différences finies pour la solution de ce problème sur une grille discrète sont bien connus; en général, nous avons :

$$(*) \quad z_{p,q} = (z_{p-1,q} + z_{p+1,q} + z_{p,q-1} + z_{p,q+1}) / 4$$

excepté le long de lignes F_i ou dans le voisinage des points de donnée. Un algorithme simple itératif de type Jacobi pour obtenir (*) est le suivant:

$$z_{p,q}^{k+1} = (z_{p+1,q}^k + z_{p-1,q}^k + z_{p,q+1}^k + z_{p,q-1}^k) / 4$$

jusqu'à ce que: $\max_{p,q} |z_{p,q}^{k+1} - z_{p,q}^k| \leq \epsilon$.

où ϵ est une tolérance fixée à l'avance.

L'un des avantages de cette correspondance entre la surface et la membrane est que les valeurs interpolées ne peut excéder l'amplitude des données. Il est bien connu que les fonctions harmoniques prennent leurs valeurs extrémales sur leurs frontières.

I - 4

D'autre part, il est connu que la membrane peut présenter des "pics pointus" (c'est-à-dire un gradient très grand) autour des points de donnée. Ces deux remarques précédentes conduisent à proposer une solution différente, plus "rigide": la correspondance avec une plaque élastique.

I-B-2-b Cas biharmonique

On établit une correspondance entre la surface et une plaque élastique, fixée à une propre distance du plan de données en accord avec les mesures, et invariante par rotation. Cette approche a été analysée par Briggs ([Briggs 74]). Dans ce cas, la surface $z(x,y)$ satisfait à l'équation biharmonique:

$$(**) \quad \Delta^2 z = \frac{\partial^4 z}{\partial x^4} + 2 \frac{\partial^4 z}{\partial x^2 \partial y^2} + \frac{\partial^4 z}{\partial y^4} = 0$$

Aux points de donnée et le long des lignes F_i des conditions propres sont imposées. Loin des données, de la frontière et des lignes F_i , la discrétisation de (**) sur la grille G donne la relation suivante:

$$20 z_{p,q} + 2 (z_{p+1,q+1} + z_{p-1,q+1} + z_{p+1,q-1} + z_{p-1,q-1}) - 8 (z_{p+1,q} + z_{p-1,q} + z_{p,q+1} + z_{p,q-1}) + z_{p+2,q} + z_{p-2,q} + z_{p,q+2} + z_{p,q-2} = 0$$

Ceci peut se schématiser à l'aide du dessin ci-dessous avec des notations évidentes:

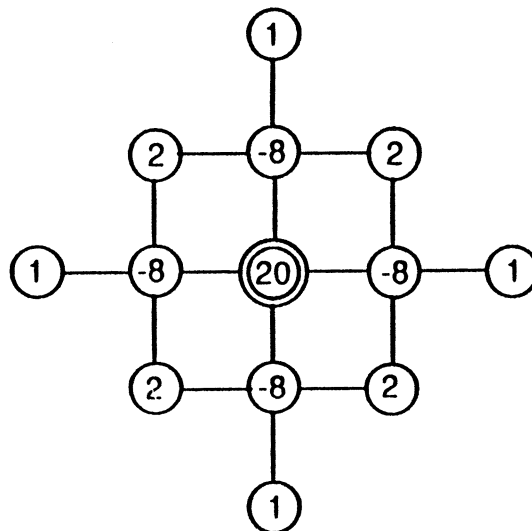


Figure I-1

I - 5

Lorsque le point de coordonnées-grille se situe sur le bord, on doit remplacer l'équation discrète précédente doit être remplacée . On schématise de la même façon que plus haut sur la figure ci-dessous les "poids" de l'équation correspondante dans le cas où un point est dans le voisinage du coin bas-gauche.

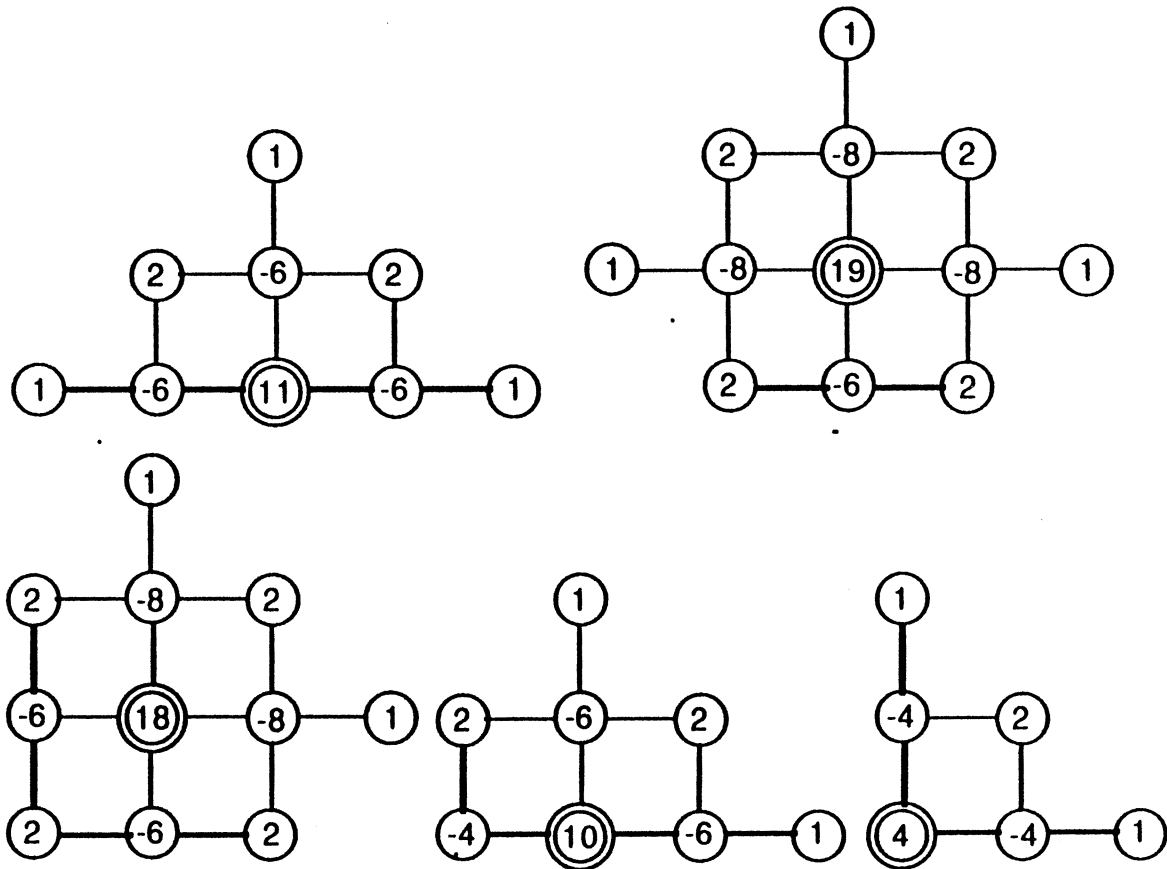


Figure I-2

Pour trouver les $z_{p,q}$, on utilise un schéma itératif de type Jacobi dont l'expression se déduit aisément de (**).

I-B-2-c Remarques et commentaires sur ces deux approches.

Si un $z_{p,q}$ coïncide avec un point de donnée, il est déjà connu; de plus lorsque un point de donnée est dans une cellule de la grille, on doit théoriquement modifier les équations discrètes correspondantes. Cependant, il est préférable pour des raisons de simplicité, de déterminer les valeurs aux sommets de la cellule contenant le point de donnée, par exemple en utilisant une méthode d'estimation linéaire (voir paragraphe I-D par exemple) et dans la suite sont fixes. Les hauteurs des autres noeuds sont estimés par un des algorithmes itératifs présentés ci-dessus.

I - 6

On doit modifier aussi les équations discrètes au voisinage des lignes. Par exemple, dans le cas harmonique, on remplace (*) par:

$$z_{p,q} = (z_{p+1,q} + z_{p-1,q}) / 2$$

si le segment reliant le noeud (p,q) aux sommets (p,q+1) ou (p,q-1) est traversé par une ligne F_i . Dans les autres cas, il est préférable d'évaluer $z_{p,q}$ par une estimation linéaire.

La convergence de cet algorithme en pratique est très lente ; on peut l'accélérer en lui préférant le procédé itératif de Gauss-Seidel (ou sur-relaxation); par exemple dans le cas harmonique on utilisera pour le procédé itératif de Gauss-Seidel:

$$z_{p,q}^{k+1} = (z_{p,q-1}^{k+1} + z_{p-1,q}^{k+1} + z_{p+1,q}^k + z_{p,q+1}^k) / 4 \quad p=1,\dots,N_x ; q=1,\dots,N_y$$

Toutefois, ceci ne se traduit pas par une réduction intéressante du nombre d'itérations.

Le problème du choix de l'ensemble de départ $\{z_{p,q}\}^0$ est difficile: on peut prendre par exemple pour chaque $z_{p,q}$ une combinaison pondérée des k premières données voisines ou utiliser une estimation linéaire.

L'un des principaux inconvénients de ces méthodes déterministes est le fait que l'on suppose que les données sont exactes.

I-B-3 Méthodes d'estimation linéaire

Supposons pour l'instant la surface continue; une approche différente pour la détermination de la surface , c'est-à-dire l'ensemble des $z_{p,q}$ est basée sur l'hypothèse que celle-ci (ou du moins son résidu après la soustraction de la tendance) est stationnaire dans le plan xy. Les auteurs font alors l'hypothèse que la surface z est un échantillonnage d'un processus stochastique bidimensionnel dont les caractéristiques statistiques sont indépendantes des coordonnées x,y. Dans ce cas on peut définir l'autocovariance du processus comme:

$$R(\alpha,\beta) = E[z(x+\alpha,y+\beta) \cdot z(x,y)]$$

où E est l'espérance. Ils supposent de plus que cette fonction est isotrope i.e $R(\alpha,\beta) = R(d)$ où $d^2 = \alpha^2 + \beta^2$ et imposent les conditions $R(0)=1$ et $R(\infty)=0$. La surface interpolante dépend fortement du choix de R: ce choix est alors un outil important pour la forme des résultats. Une fois ce choix fait, on peut interpoler les données sur une grille régulière.

Les poids d'interpolation dépendent de la position du point et sont déterminés en minimisant l'erreur carrée moyenne d'interpolation.

I - 7

Quelques calculs montrent que la surface z s'écrit :

$$z(x,y) = \sum_{i=1}^n \beta_i R(d_i) = z^t \mathcal{R}^{-1} r = z^t \alpha$$

Ici $\mathcal{R} = (r_{i,j})_{i,j=1,\dots,n} = E(z z^t)$, α est solution du système $\mathcal{R}^t \alpha = r^t$ où

$$r_{i,j} = R(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2})$$

$$r_i = R(\sqrt{(x_i - x)^2 + (y_i - y)^2})$$

Il faut remarquer que les β_i coordonnées du vecteur $z^t \mathcal{R}^{-1}$ sont indépendantes de x,y ; les $R(d_i)$ sont les composantes du vecteur r , si d_i désigne la distance euclidienne entre le $i^{\text{ème}}$ point de donnée et le point de coordonnée x,y . On voit alors que la surface est une combinaison linéaire de n surfaces de révolution identiques ($R(d_i)$) centrées sur les n points de donnée. Les coefficients de cette combinaison sont tels que $z(x_i, y_i) = z_i$.

Considérons maintenant une surface présentant des discontinuités de valeur. Une façon de prendre en compte celles-ci est de redéfinir la notion de distance pénalisante entre deux points. L'idée majeure ici est "d'étirer" le plan au voisinage des lignes de faille.

On définit la distance $\tilde{d}$ entre deux points dans le plan faillé comme la longueur du plus court chemin joignant ces deux points, ne coupant pas les lignes de discontinuités. Si la faille séparant deux points est fermée ou illimitée alors la distance est infinie et

l'autocovariance est nulle. Cette distance fait de $\mathbb{R}^2 \setminus (\bigcup_{i=1}^n F_i)$ un espace métrique.

Cette notion de distance pénalisante est très séduisante à priori car elle a une interprétation "réaliste" amusante: supposons que nous soyons au sommet d'une montagne ayant une face verticale et que nous désirions descendre par voie terrestre dans la vallée; alors on voit qu'il faut essayer de trouver un chemin de longueur la plus petite possible évitant cette face.

Toutefois, la recherche de cette distance dans un cas général est un problème difficile: il se ramène au problème du plus court chemin dans un graphe non orienté, ce qui on sait est un problème difficile et pour lesquelles on utilise des techniques d'optimisation combinatoire (le seul cas trivial est celui où on traite une seule faille rectiligne).

Il reste encore deux faits à préciser:

- le choix de la fonction autocovariance; en effet celui influence fortement la forme de la solution $z(x,y)$.
- l'inversion de la matrice $\mathcal{R}$.

I - 8

Pour le premier, les auteurs préconisent dans le cas général des gaussiennes ou des combinaisons de gaussiennes, par exemple:

$$R(d) = e^{-\left(\frac{d}{d_0}\right)^2}$$

où d_0 est un paramètre positif.

Ceci est à rapprocher des surfaces multiquadriques de révolution de Hardy ([HARDY 76]) où un paramètre positif est laissé au choix de l'analyste.

Le deuxième fait est plus ennuyeux: on ignore les hypothèses à faire sur l'autocovariance ou sur le jeu de données pour que la matrice $\mathcal{R}$ soit inversible.

Toutefois, les auteurs montrent que celle-ci est définie positive (donc inversible) si on utilise la notion de distance pénalisante et si R est une fonction du type:

$$R(d) = \int_0^\infty e^{-\alpha^2 d^2} F(\alpha) d\alpha \quad \text{où } F(\alpha) \geq 0$$

Si le nombre de points est grand, l'inversion de $\mathcal{R}$ devient impossible. En remarquant que les coefficients β_i relativement à la surface de révolution $R(d)$ centrée au $i^{\text{ème}}$ point est pratiquement indépendante des points lointains, les auteurs proposent de reconsidérer la procédure en utilisant l'interpolation en un point à partir des 8 points de donnée les plus proches. Dans ce cas l'inversion de $\mathcal{R}$ est facile. Mais il se peut que cette inverse ait été calculée plus d'une fois en passant d'un point de la grille à un autre et la surface devient discontinue car de nouveaux points de donnée entrent dans l'ensemble des 8 points. Ces discontinuités évidemment négligeables lorsque la taille h de la grille est fine, peuvent être réduites en itérant à l'aide de méthodes déterministes décrites ci-dessus.

Enfin, la méthode d'estimation linéaire possède un autre caractère attrayant, celui de lisser les données. En effet, on peut remplacer la matrice du système $\mathcal{R}$ par $\mathcal{R} + \rho \mathbf{I}$ où ρ est un réel positif réalisant un compromis entre l'approximation aux données et le lissage de la surface. Ceci d'ailleurs revient à prendre comme fonction d'autocovariance des mesures la fonction R' définie par:

$$R'(d) = \begin{cases} R(d) & \text{si } d \neq 0 \\ 1 + \rho & \text{si } d = 0 \end{cases}$$

En conclusion, nous voyons que cette méthode remédie aux problèmes non résolus par les méthodes déterministes. Toutefois sa mise en oeuvre est quasi-impossible sauf dans des cas triviaux, à cause de l'évaluation délicate de la distance pénalisante et du temps de calcul prohibitif de la surface.

I-C Quadriques de Shepard modifiées.

Dans ce paragraphe nous faisons ici référence aux travaux de Franke et Nielson ([**Franke 83**]). Ils proposent une méthode d'interpolation locale applicable à un ensemble de données quelconque. On peut classer ces méthodes en deux branches:

- (i) approximations locales pondérées,
- (ii) méthodes "d'éléments finis".

Ces dernières d'une part demandent une résolution globale d'un système linéaire et d'autre part, seule l'évaluation est locale (en général on doit calculer un polynôme dont les coefficients dépendent des valeurs aux sommets de la triangulation). A contrario, les méthodes d'approximation locale pondérée génèrent une surface dont l'expression analytique est globale, et dans certains cas (voir ci-dessous) demandent plusieurs résolutions au moindre carré de systèmes linéaires dont les coefficients des matrices associées dépendent d'un petit nombre de points.

I-C-1 Quadriques de Shepard

Exposons d'abord la méthode dans le cas où la surface est supposée continue. Soient $M_k(x_k, y_k)$ n points de donnée et z_k n réels. Les schémas d'approximation locale pondérée sont de la forme:

$$(*) \quad z(x,y) = \frac{\sum_{i=1}^n W_k(x,y) \cdot Q_k(x,y)}{\sum_{i=1}^n W_k(x,y)}$$

Ici $Q_k(x,y)$ est une approximation locale de la fonction sous-jacente définie sur le support de W_k . Q_k est appelée fonction nodale quand elle est associée à un point. W_k est une fonction poids, qu'on prend habituellement positive. Bien sûr, la plupart de ces fonctions poids W_k ont un support fini. Il peut y avoir (mais ce n'est pas nécessaire) un point associé à chacune de ces fonctions W_k . Ces fonctions poids prennent en compte les informations concernant la disposition des coordonnées des points de donnée, mais sont indépendants des z_k . Les fonctions nodales par contre, contiennent les informations relatives au comportement local de la surface. On remarque ici que l'interpolant F est une combinaison linéaire convexe des fonctions Q_k .

Shepard ([**Shepard 68**]) fut l'inventeur de cette méthode en prenant $Q_k(x,y) = z_k$ et $W_k(x,y) = 1/d_k(x,y)$ où $d_k(x,y)$ est la distance euclidienne du $k^{\text{ème}}$ point de donnée au

I - 10

point de coordonnée (x,y) . Si l'on définit la fonction poids W^* moyenne par:

$$W_k^*(x,y) = \frac{W_k(x,y)}{\sum_{i=1}^n W_i(x,y)}$$

alors W_k^* vérifie les propriétés suivantes le long de $d_i(x_i, y_i) = 0$:

$$W_i^*(x_j, y_j) = \delta_{i,j}$$

$$\sum_{i=1}^n W_i^*(x, y) = 1$$

$$\frac{\partial W_i^*}{\partial x}(x_j, y_j) = \frac{\partial W_i^*}{\partial y}(x_j, y_j) = 0 \quad i, j = 1, \dots, n$$

En particulier, z ne peut dépasser les valeurs extrémales prises par les z_k . De plus le gradient de F s'annule en chaque point de donnée, ce qui est rarement le cas de la surface sous-jacente. Donc sans modification notable, z ne possède pas de bonnes propriétés de lissage. Plusieurs extensions de cette méthode ont été construites depuis ([McLain 74], [McLain 76]), et parmi celles-ci, des tests récents ont montré que la méthode des quadriques de Shepard modifiées donnent d'excellents résultats pour une grande variété de jeux de données.

Décrivons cette technique en détail. Dans cette méthode les fonctions poids ont pour expression:

$$W_k(x,y) = \left(\frac{(R_W - d_k(x,y))_+}{R_W d_k(x,y)} \right)^2$$

Les fonctions nodales Q_k sont des quadriques contraintes à passer par le point (x_k, y_k) , les autres coefficients étant déterminés par un lissage pondéré aux moindres carrés; plus exactement si l'on pose

$$Q_k(x,y) = z_k + a_k^1(x - x_k) + a_k^2(y - y_k) + a_k^3(x - x_k)^2 + a_k^4(x - x_k)(y - y_k) + a_k^5(y - y_k)^2$$

alors on résoud pour chaque Q_k le problème de moindre carré pondéré (P_k) :

$$(P_k) \quad \text{Min}_{a_k^j : j=1, \dots, 5} \sum_{\substack{i=1 \\ i \neq k}}^n w_i(x_k, y_k) [Q_k(x_i, y_i) - z_i]^2$$

I - 11

Ici la fonction w_i est par définition : $w_i(x,y) = \left(\frac{(R_Q - d_i(x,y))_+}{R_Q d_i(x,y)} \right)^2$

Les paramètres R_W , R_Q sont des rayons dépendant de la densité du nuage de points de données. On pourrait prendre des rayons différents pour chaque point pour donner plus de flexibilité à la méthode mais dans le cas du calcul d'un interpolant général, cela se traduit souvent par une perte de temps. R_Q est le rayon d'influence des fonctions nodales, tandis que R_W est le rayon d'influence des fonctions poids. Dans le cas général, Franke et Nielson préconisent la méthode suivante pour déterminer ces rayons: soit la quantité $D_{\max} = \max_{i,j} d_i(x_j, y_j)$ alors on prend comme rayons d'influence:

$$R_Q = \frac{D_{\max}}{2} \sqrt{\frac{n_Q}{n}} \quad \text{et} \quad R_W = \frac{D_{\max}}{2} \sqrt{\frac{n_W}{n}}$$

Les valeurs de n_Q et n_W sont les nombres points de donnée anticipés qui sont à l'intérieur des cercles de rayon R_Q et R_W respectivement. Si le nuage de points est uniforme, les auteurs proposent $n_Q = 18$ et $n_Q/n_W \neq 2$. Lorsque moins de cinq points de données se situent dans le cercle de rayon R_Q de centre (x_i, y_i) il faut remplacer la quadrique Q_i par une fonction nodale affine.

En général, pour résoudre les problèmes de moindre carré, on utilise la méthode de décomposition en valeurs singulières (UDV) ou la méthode de Householder de la matrice rectangulaire du système correspondant, ce qui évite les problèmes de non-unicité.

I-C-2 Prise en compte des discontinuités

Revenons maintenant à notre problème en remarquant d'abord qu'il existe deux façons de procéder: l'expression de z peut incorporer un caractère non lisse soit dans les fonctions poids, soit dans les fonctions nodales. Si l'on prend les lignes de faille comme ayant un rapport avec le domaine de la fonction, alors le comportement anormal est incorporé dans les fonctions poids. D'autrepart, si ce comportement est vu comme étant du ressort de la surface, alors les fonctions nodales doivent prendre en compte ce comportement comme une partie des propriétés locales de la surface. Seules les fonctions nodales dont le support intersecte la ligne de discontinuité doivent reproduire ce comportement.

Franke et Nielson proposent une première méthode dénommée DPF (Distance Penalty Fault) qui consiste à remplacer (sous l'hypothèse que les lignes de faille sont grandes par rapport à R_W), la distance euclidienne dans les fonctions poids par une distance pénalisante qui est la même que celle décrite dans le paragraphe I-B. Cependant, la généralisation de cette distance pour des surfaces admettant des

I - 12

discontinuités de dérivée, n'est pas possible.

C'est pourquoi ils proposent d'incorporer dans les fonctions nodales Q_k un terme convenable avec l'approximation quadratique, le coefficient étant déterminé au même titre que ceux de la quadrique. Le but de ce terme est d'enlever le caractère anormal tel que la fonction sous-jacente restante peut être bien approximée localement par une quadrique. Dans le cas d'une seule faille rectiligne, considérons le système de coordonnées suivant: soit t la distance paramétrisée le long de cette ligne, et s la coordonnée dans la direction perpendiculaire. Ce terme doit être évidemment incorporé uniquement dans les fonctions nodales dont les points associés sont à une distance moindre que R_0 de cette ligne de discontinuité. La première méthode notée NFF1(Nodal Functions Faults) consiste à choisir le terme suivant:

$$\phi_f(s,t) = H_3(s) H_3(2t-1) \operatorname{sgn}(s)$$

$$\phi_d(s,t) = H_3(s) H_3(2t-1) |s|$$

Ici $H_3(s)$ est la cubique par morceaux : $H_3(s) = (1 - s^2(3 - 2|s|)) \mathbb{1}_{[-1,1]}(s)$, où $\mathbb{1}_{[-1,1]}$ est la fonctionnelle unité de l'intervalle $[-1,1]$. Dans ce cas, on remplace la fonction nodale Q_k par :

$$\dot{Q}_k(x,y) = Q_k(x,y) + f_k [\phi_f(s,t) - \phi_f(s_k,t_k)] + d_k [\phi_d(s,t) - \phi_d(s_k,t_k)]$$

où (s,t) , (s_k,t_k) sont les coordonnées d'un point $M(x,y)$ et d'un point de donnée M_k relativement au repère de la faille; f_k , d_k étant les coefficients à déterminer.

Pour incorporer plus de flexibilité, les auteurs proposent une deuxième méthode, notée NFF2, en ajoutant trois termes à la quadrique Q_k qui sont ici de la forme:

$$\begin{cases} \phi_f^i(s,t) = H_3(s) \{ t^{i+1}(1-t)^{5-i} \} \operatorname{sgn}(s) \\ \phi_d^i(s,t) = H_3(s) \{ t^{i+1}(1-t)^{5-i} \} |s| \end{cases} \quad i = 1, 2, 3$$

Dans ces deux expressions les parties polynômiales en la variable t sont les polynômes de Bernstein de degré 6 s'annulant ainsi que leurs dérivées aux extrémités 0 et 1. Ceci évite la propagation du caractère non lisse de ces termes dans la direction de s . On remarque que l'expression des termes ϕ_d^i ont des dérivées normales gauche et droite égales en valeur absolue, ce qui est rarement le cas de la fonction sous-jacente et donne un résultat médiocre de la surface au voisinage de la ligne de discontinuité. Dans ce cas, on remplace la fonction nodale Q_k par :

$$\dot{Q}_k(x,y) = Q_k(x,y) + \sum_{i=1}^3 f_k^i (\phi_f^i(s,t) - \phi_f^i(s_k,t_k)) + \sum_{i=1}^3 d_k^i (\phi_d^i(s,t) - \phi_d^i(s_k,t_k))$$

I - 13

Un inconvénient majeur est que ces deux dernières méthodes n'ont pas encore été généralisées au cas des lignes de discontinuité brisées, i.e. de trouver des termes représentant les discontinuités, ayant un support fini. De plus, l'utilisation d'un seul terme discontinu est inadéquate pour représenter correctement le comportement local de la surface au voisinage de la ligne de faille.

Toutefois, ces deux techniques peuvent être étendues à l'ajustement des données (c'est-à-dire lorsque celles-ci sont entachées d'erreurs) en ajustant localement le comportement local de la surface: on n'impose plus aux fonctions nodales Q_k de passer par le point M_k , c'est-à-dire on considère l'expression suivante:

$$Q_k(x, y) = a_k^0 + a_k^1(x - x_k) + a_k^2(y - y_k) + a_k^3(x - x_k)^2 + a_k^4(x - x_k)(y - y_k) + a_k^5(y - y_k)^2$$

le coefficient a_k^0 étant déterminé au même titre que les autres dans la résolution du problème (P_k).

I-D DOUBLE INTERPOLATION LINEAIRE

Nous faisons ici référence à une procédure d'interpolation (GINTPF) développée dans le logiciel UNIRAS ([UNIRAS86]) apportant une réponse pour le cas de surfaces explicites présentant des discontinuités de valeur le long de lignes brisées.

Cette procédure interpole à partir d'un ensemble de points quelconque sur une grille régulière de points spécifiée à l'aide d'une méthode de double interpolation linéaire. Il est possible ensuite de raffiner et d'ajuster la grille en lissant les valeurs de la surface aux sommets de la grille par une technique de pondération des distances.

I-D-1 Double Interpolation linéaire

Soient des triplets $(x_i, y_i, z_i)_{i=1, \dots, n}$ où les z_i sont les valeurs d'une fonction $z(x, y)$ et $M = (X_m, Y_n)$ les coordonnées d'un sommet de la grille auquel on désire une valeur d'interpolation. Un rayon d'influence R des données étant choisi, on sélectionne dans chaque quadrant d'origine M , le point M_i ($i=1, \dots, 4$) le plus proche. En général, R est pris par défaut comme la longueur de la diagonale du domaine rectangulaire englobant les données (voir plus loin).

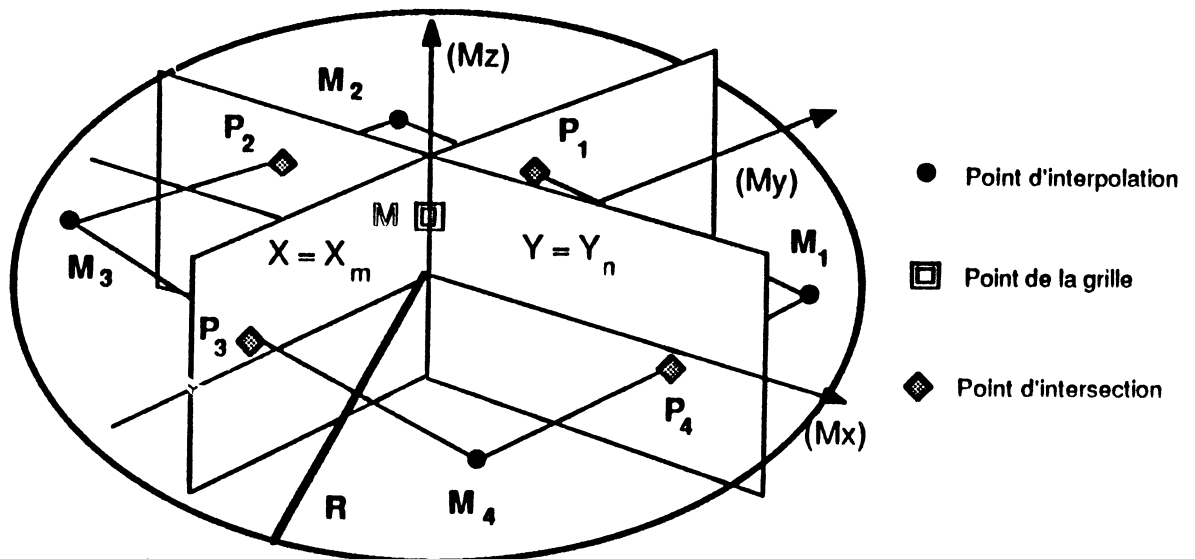


Figure I-3.

Si la distance du plus proche point de donnée au point M est plus petit que d fois la taille de la grille (par défaut, $d = 0.5$), ce point de donnée est pris comme valeur de la grille. Dans le cas contraire, on procède comme suit.

I - 15

• Si un point $M_i(x_i, y_i, z_i)$ $i=1, \dots, 4$ est trouvé dans chaque quadrant, on construit une double interpolation linéaire en deux étapes de la façon suivante:

1- on calcule les intersections $P_i(\alpha_i, \beta_i, \gamma_i)$, $i=1, \dots, 4$ des plans (M_x) , (M_y) avec les segments $[M_i, M_{1+i \bmod 4}]$. Ceci donne :

$$\begin{cases} \alpha_{2i-1} = X_m \\ \beta_{2i-1} = Y_{2i-1} + \left(\frac{Y_{2i} - Y_{2i-1}}{X_{2i} - X_{2i-1}} \right) \cdot (X_m - X_{2i-1}) \\ \gamma_{2i-1} = Z_{2i-1} + \left(\frac{Z_{2i} - Z_{2i-1}}{X_{2i} - X_{2i-1}} \right) \cdot (X_m - X_{2i-1}) \end{cases}$$

pour $i = 1, 2$

$$\begin{cases} \alpha_{2i} = X_{2i} + \left(\frac{X_{1+(2i) \bmod 4} - X_{2i}}{Y_{1+(2i) \bmod 4} - Y_{2i}} \right) \cdot (Y_n - Y_{2i}) \\ \beta_{2i} = Y_n \\ \gamma_{2i} = Z_{2i} + \left(\frac{Z_{1+(2i) \bmod 4} - Z_{2i}}{Y_{1+(2i) \bmod 4} - Y_{2i}} \right) \cdot (Y_n - Y_{2i}) \end{cases}$$

2 - on effectue une interpolation sur chaque axe et la valeur Z_{mn} au point M est la demi-somme de ces deux interpolations, i.e. :

$$Z_{mn} = \frac{1}{2} \left(\gamma_1 + \frac{\gamma_3 - \gamma_1}{\beta_3 - \beta_1} (Y_n - \beta_1) + \gamma_2 + \frac{\gamma_4 - \gamma_2}{\alpha_4 - \alpha_2} (X_m - \alpha_2) \right)$$

Remarque: Chacune de ces deux étapes est un procédé linéaire, mais la combinaison des deux ne l'est plus; en fait si l'on combine les deux expressions plus haut, on montre après quelques calculs élémentaires, que la surface d'interpolation obtenue se comporte localement comme une fraction rationnelle en les variables X,Y dont le numérateur est un polynôme de degré total 4 et le dénominateur un polynôme de degré total 2.

• Si on trouve des points de donnée seulement dans 3 quadrants, on ne peut calculer que 2 intersections avec les plans. Supposons par exemple que ce soient les points P_1, P_2 ; alors la valeur Z_{mn} assignée au point M est:

$$Z_{mn} = \gamma_1 + \frac{\gamma_2 - \gamma_1}{\alpha_2 - \beta_1} (Y_1 - \beta_1)$$

I - 16

- Si on trouve des points de donnée seulement dans 2 quadrants adjacents, on ne calcule qu'une seule intersection avec les plans. Supposons que ce soit le point P_1 , alors $Z_{mn} = \gamma_1$.

- Si on trouve des points de donnée seulement dans 2 quadrants opposés ou dans un seul quadrant, alors la valeur du point de donnée le plus proche est assignée à M.

Ensuite, une opération d'ajustement des données est effectuée en appliquant un filtre bidimensionnel à la grille. Ce filtre est conçu afin de réduire la courbure de la surface dans les régions de données éparées tout en maintenant la qualité de l'interpolation autour des points de données. Plus précisément, un entier L (appelé niveau d'ajustement) étant fixé, on effectue L fois l'itération suivante pour chaque point M (X_m, Y_n) de la grille:

$$Z_{mn} = \frac{1}{k} \sum_{i=1}^k \frac{Z_i}{d_i^3}$$

où k est le nombre de points de la grille situés dans le cercle de rayon R, Z_i et d_i sont respectivement la valeur et la distance du point M au point de la grille.

I-D-2 Prise en compte des discontinuités.

La prise en compte de la discontinuité de valeur de la surface le long d'une ligne brisée est basée sur l'hypothèse suivante: *"le comportement local de la surface d'un côté de cette ligne est en première approximation largement indépendant du comportement de l'autre côté"*. Ici "même côté" signifie les données avec une aire de vision non coupée par une ligne de faille.

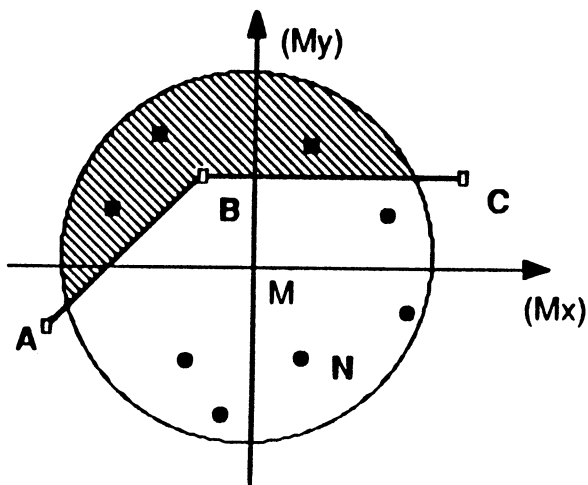


Figure I-4

I - 17

En d'autres termes, lors de l'évaluation de la surface en un point M, on ne prend en considération que les points de donnée se situant du même côté de la ligne que celui-ci.

Dans la figure ci-haut, tous les points de donnée dans la zone non hachurée sont des éventuels candidats pour l'évaluation de la surface. Pour tester si un point de donnée N est dans la zone restreinte, on cherche l'intersection du segment [MN] avec chaque segment $[A_i B_i]$ définissant la ligne brisée. Dans le cas où cette intersection est non vide, ce point est alors pris en compte. Cette intersection peut être déterminée d'une manière très simple, décrite comme suit:

un point P est l'intersection des segments [MN] et $[A_i B_i]$ si et seulement si on a:

$$\begin{cases} \overrightarrow{MP} = t \overrightarrow{MN} & \text{où } 0 \leq t \leq 1 \\ \overrightarrow{A_i P} = u \overrightarrow{A_i B_i} & \text{où } 0 \leq u \leq 1 \end{cases}$$

Il est facile de voir que ceci est équivalent à : $\overrightarrow{A_i M} = t \overrightarrow{NM} + u \overrightarrow{A_i B_i}$

Donc $[MN] \cap [A_i B_i] \neq \emptyset$ si seulement si t,u vérifient l'égalité précédente avec $0 \leq t, u \leq 1$.

Evidemment, le temps d'exécution augmente avec le nombre de segments nécessaire pour définir les lignes de discontinuité.

Cette méthode présente des intérêts lorsque la grille n'est pas trop fine et que le nombre de lignes de discontinuité est "raisonnable". Le choix du rayon d'influence R ayant un effet non négligeable sur le temps d'exécution, il est alors crucial de retenir une valeur de R n'affectant pas celui-ci; notamment en présence de failles, R doit être diminué afin de ne pas tester trop de points de données.

Toutefois cette technique présente deux inconvénients majeurs:

- On ne dispose pas d'une formule analytique globale de l'approximant, ce qui peut être fâcheux dans certaines situations où l'on veut pouvoir disposer d'une telle formule pour des calculs ultérieurs.
- On ne peut prendre en considération le cas de surfaces présentant des discontinuités de dérivées le long de lignes brisées.

I-E Méthodes de régularisation et éléments finis.

I-E-1 Généralités

Soit $z(x,y)$ une fonction de deux variables sujette à des contraintes de valeurs $(z_i)_{i=1,\dots,n}$ éventuellement entachées d'erreurs. Celles ci peuvent s'écrire en général de manière abstraite:

$$z_i = \mathcal{M}_i [z(x,y)] + \varepsilon_i \quad i=1,\dots,n$$

où les $\mathcal{M}_i$ sont des fonctionnelles de mesure et les ε_i désignent les erreurs associées à ces mesures.

De nombreuses approches systématiques, notamment les méthodes de régularisation de Tikhonov ([Tikhonov 77]) mentionnent une forme basique de régularisation pour éliminer les effets des bruits sur les données. Ces techniques de régularisation ont pour pierre de touche un principe variationnel reformulant le problème proposé et assurant l'existence, l'unicité et la stabilité de la solution. Cette reformulation est effectuée à l'aide d'une fonctionnelle qui impose des hypothèses de lissage sur l'ensemble des solutions admissibles en les restreignant aux espaces de Sobolev. Ce type de régularisation est équivalent à l'approximation optimale par fonctions-spline généralisées ([Duchon 80]). La théorie de l'approximation optimale par fonctions-spline est maintenant bien développée et une connexion a été établie entre les principes variationnels invoquant un problème de minimisation quadratique d'une (semi-) norme dans des espaces (semi-) hilbertiens ([Laurent 72]).

Formulons un principe variationnel dans le cadre général:

Soit $\mathcal{H}$ un espace vectoriel de fonctions "lisses" et $\mathcal{L}(u)$ une fonctionnelle définie sur $\mathcal{H}$ mesurant la qualité du lissage d'une fonction dans $\mathcal{H}$. Notons l'opérateur $\mathcal{M}(v) = [\mathcal{M}_1(v), \dots, \mathcal{M}_n(v)]^T$ et $z = [z_1, \dots, z_n]^T$. Un principe variationnel d'interpolation (PVI) s'écrit alors:

$$(PVI) \quad \text{trouver } u \in \mathcal{H} \text{ tel que : } \mathcal{E}(u) = \inf_{v \in \mathcal{H}; \mathcal{M}(v)=z} \mathcal{E}(v)$$

où la fonctionnelle "énergie" est : $\mathcal{E}(v) = \mathcal{L}(v)$.

Soit $\mathcal{P}$ une fonctionnelle définie sur $\mathcal{H}$ donnant une idée de "l'écart" entre la fonction et les contraintes. Un principe variationnel d'ajustement (PVA) s'énonce:

$$(PVA) \quad \text{trouver } u \in \mathcal{H} \text{ tel que : } \mathcal{E}(u) = \inf_{v \in \mathcal{H}} \mathcal{E}(v)$$

où la fonctionnelle "énergie" est : $\mathcal{E}(v) = \mathcal{L}(v) + \mathcal{P}(v)$.

Avant de donner quelques exemples de fonctionnelles de lissage $\mathcal{L}(v)$ et de

I - 19

fonctionnelles de pénalité $\mathcal{P}(v)$, on note que si la solution existe, elle satisfait la condition nécessaire du minimum donnée par l'annulation de la première variation δ de la fonctionnelle "énergie":

$$\delta \mathcal{E}(u) = 0$$

qui donne immédiatement les équations d'Euler-Lagrange associées à (PVI) ou (PVA).

Exemples classiques de fonctionnelles de "lissage" et de "pénalité".

- La classe des surfaces-spline peut être caractérisée comme éléments de l'espace de Sobolev $H^m(\mathbb{R}^2)$ qui minimise la semi-norme $|\cdot|$ définie par :

$$|v|_m^2 = \int_{\mathbb{R}^2} \sum_{i=1}^m \binom{m}{i} \left(\frac{\partial^m v}{\partial x^i \partial y^{m-i}} \right)^2 dx dy$$

L'entier positif m indique l'ordre des dérivées partielles apparaissant dans cette fonctionnelle et en outre détermine le degré de continuité $(m-1)$ des fonctions admissibles. L'espace nul N de $H^m(\mathbb{R}^2)$, c'est-à-dire l'ensemble des éléments ayant une semi-norme nulle est l'espace des polynômes définies sur $\mathbb{R}^2$ de degré total inférieur à $m-1$ de dimension $\binom{m+1}{2}$.

L'équation d'Euler-Lagrange satisfaite par la solution minimisatrice est une version m -itérée de l'équation de Laplace: $(-1)^m \Delta^m u = 0$.

Les surfaces-spline d'ordre faible ont des propriétés physiques concernant l'équilibre de corps élastiques. Deux cas classiques méritent d'être rappelés. Pour $m = 1$, la fonctionnelle s'écrit:

$$|v|_1^2 = \int_{\mathbb{R}^2} \|\nabla v\|_{\mathbb{R}^2}^2 dx dy \quad \text{où} \quad \nabla v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

qui en première approximation, est proportionnelle à l'énergie de flexion d'une membrane (et génèrent des solutions de classe C^0), tandis que pour $m = 2$, la fonctionnelle s'écrit:

$$|v|_2^2 = \int_{\mathbb{R}^2} \|\nabla^2 v\|_{\mathbb{R}^3}^2 dx dy \quad \text{où} \quad \nabla^2 v = \begin{pmatrix} v_{xx} \\ \sqrt{2} v_{xy} \\ v_{yy} \end{pmatrix}$$

qui en première approximation est proportionnelle à l'énergie de flexion d'une plaque

I - 20

mince (et génèrent des solutions de classe C^1).

Plus généralement, on considère des versions pondérées des espaces de Sobolev H^m permettant de "contrôler" les propriétés de continuité d'une spline généralisée en introduisant des fonctionnelles de lissage non quadratiques. Plus précisément, soit les fonctionnelles de "lissage" du type:

$$|v|_{p,w}^2 = \sum_{p=0}^m \int_{\mathbb{R}^2} w_p(x,y) \sum_{j_1+j_2=p} \frac{p!}{j_1! j_2!} \left(\frac{\partial^p v(x,y)}{\partial x^{j_1} \partial y^{j_2}} \right)^2 dx dy$$

Les fonctions $w(x,y) = [w_0(x,y), \dots, w_m(x,y)]^T$ sont positives, non nécessairement continues et en général non spécifiées. Si on leur impose de s'annuler en quelques endroits, cela permet d'introduire de manière sélective des discontinuités spécifiques, ce qui est une propriété intéressante qui sera illustrée dans le paragraphe I-E-3.

Remarque: Ceci en fait, est une généralisation de courbes spline sous tension i.e. les fonctions $u(x)$ minimisant la fonctionnelle de lissage:

$$\int_{\mathbb{R}^2} \alpha \left(\frac{dv}{dx} \right)^2 + \left(\frac{d^2v}{dx^2} \right)^2 dx$$

où α est un paramètre positif appelé tension. Le premier terme influence sur la "longueur" de la spline, tandis que le second influence sa "courbure". Si on augmente la tension, cela tend à faire disparaître les variations indésirables de la courbe en réduisant la "longueur" de la spline.

- Un exemple simple et très courant de fonctions de "pénalité" est le suivant: étant donné des erreurs de mesure ϵ_i indépendantes de moyenne nulle et de variance σ_i , une mesure d'incompatibilité est une norme euclidienne pondérée de l'écart entre les fonctions admissibles et les données z_i :

$$\mathcal{P}(v) = \sum_{i=1}^n \alpha_i [\mathcal{M}_i(v) - z_i]^2$$

où $\mathcal{M}_i$ est une fonctionnelle linéaire définie sur $\mathcal{H}$ et α_i un poids réel positif (par exemple $\alpha_i = 1/\lambda\sigma_i^2$ où $\lambda \geq 0$).

I - 21

Un exemple classique de fonctionnelles $\mathcal{M}_j$ peut être construit à l'aide de dérivées partielles d'ordre k :

$$\mathcal{M}_j(v) = \frac{\partial^k v}{\partial x^j \partial y^{k-j}} \bigg|_{(x_i, y_i)} \quad \text{pour } j = 0, \dots, k$$

$k=0$ est simplement l'évaluation $v(x_i, y_i)$ de la surface en un point qui peut être employé pour modéliser les contraintes : $z_i = v(x_i, y_i) + \epsilon_i = f_{(x_i, y_i)}$.

$k=1$ est l'évaluation des dérivées partielles de v en un point qui peut être utilisé pour modéliser les contraintes:

$$\begin{cases} z_i = \frac{\partial v}{\partial x}(x_i, y_i) + \epsilon_i = g_{(x_i, y_i)} \\ z_i = \frac{\partial v}{\partial y}(x_i, y_i) + \epsilon_i = h_{(x_i, y_i)} \end{cases}$$

On peut construire aisément d'autres fonctionnelles de mesure: dérivée directionnelle, moyenne locale sur des ouverts, etc...

Il est extrêmement difficile, voire impossible, d'obtenir une solution analytique de ces principes variationnels, ceci étant dû à l'irrégularité des géométries des contraintes et des discontinuités. La seule approche viable est de transformer ces problèmes en des problèmes discrets dont la solution peut être évaluée numériquement. A cette fin, la *Méthode des Eléments Finis* (M.E.F.) est une technique idéale pour retrouver le comportement local de la surface. C'est une méthode très générale, extrêmement puissante, et mathématiquement une technique d'approximation rigoureuse qui permet de sélectionner des éléments appropriés et gouverne leurs interactions suivant la nature du principe variationnel retenu. De plus, elle offre une flexibilité substantielle dans la discrétisation de domaines avec des frontières de forme irrégulières.

Nous présentons dans ce qui suit, deux méthodes d'interpolation ou d'ajustement des données basées sur ces techniques de régularisation dont la discrétisation associée utilise la M.E.F.

I-E-2 D^m-spline d'interpolation et d'ajustement.

I-E-2-1 Position du problème

Nous faisons ici référence aux travaux présentés dans la thèse de Manzanilla ([Manzanilla 86]).

Soit Ω un ouvert borné non vide de $\mathbb{R}^2$ et F un ensemble fermé, connexe, non réduit à un point, inclus dans Ω , représentant ici les discontinuités de valeur de la surface. Soit

I - 22

$\Omega' = \Omega \setminus F$. Soit $\mathcal{H} = H^m(\Omega')$ l'espace de Sobolev d'ordre m muni du semi-produit scalaire:

$$(u, v)_{m, \Omega'} = \int_{\Omega'} \sum_{i=0}^m \binom{m}{i} \frac{\partial^m u}{\partial x^i \partial y^{m-i}} \cdot \frac{\partial^m v}{\partial x^i \partial y^{m-i}} dx dy$$

et de la semi-norme associée: $|v|_{m, \Omega'} = \sqrt{(v, v)_{m, \Omega'}}$

Soit $P_{m-1}(\Omega')$ l'espace des polynômes définis sur Ω' de degré total inférieur ou égal à m . La fonctionnelle de "lissage" $\mathcal{L}(v)$ retenue par l'auteur est justement cette semi-norme.

Soit $P_{m-1}(\Omega')$ l'espace des polynômes définis sur Ω' de degré total inférieur ou égal à m . On prend ici les $\mathcal{M}_i$ comme les fonctionnelles d'interpolation ponctuelle: $\mathcal{M}_i(v) = v(a_i)$, $i=1, \dots, n$, où les a_i sont des points de Ω contenant un sous-ensemble $(m-1)$ -unisolvent, c'est-à-dire:

$$\forall p \in P_{m-1}(\Omega) ; \forall i \in [1, \dots, n] \quad (p(a_i) = 0) \Rightarrow (p = 0)$$

Remarque: On peut généraliser ceci en prenant des fonctionnelles sur $\mathcal{H}$ quelconques vérifiant :

$$\forall p \in P_{m-1}(\Omega) ; \forall i \in [1, \dots, n] \quad (\mathcal{M}_i(p) = 0) \Rightarrow (p = 0)$$

Par définition u est la D^m -spline d'interpolation dans $\mathcal{H}$ relativement à l'opérateur $\mathcal{M}$, si elle est solution du problème (PVI) correspondant. De plus on a le résultat suivant (voir [Manzanilla 86]):

Théorème 1: Le problème (PVI) admet une solution unique u caractérisée par il existe $\lambda \in \mathbb{R}^n$ tel que (λ, u) vérifie l'équation variationnelle:

$$\begin{cases} \mathcal{M}(u) = z \\ \forall v \in \mathcal{H}, \quad \langle \lambda, \mathcal{M}(v) \rangle_n + (u, v)_{m, \Omega'} = 0 \end{cases}$$

où $\langle \dots \rangle_n$ désigne le produit scalaire euclidien dans $\mathbb{R}^n$.

Prenons comme fonctionnelle "pénalité" celle définie dans l'exemple ci-dessus, α_i étant pris à $p > 0$. Par définition, u_p est la D^m -spline d'ajustement dans $\mathcal{H}$ relativement à l'opérateur $\mathcal{M}$, si elle est solution du problème (PVA) correspondant. On a de même le

I - 23

résultat:

Théorème2: Le problème (PVA) admet une solution unique u_ρ dans $\mathcal{H}$ caractérisée par l'équation variationnelle:

$$\begin{cases} u_\rho \in \mathcal{H} \\ \forall v \in \mathcal{H}, (u_\rho, v)_{m, \Omega'} = \rho < z - \mathcal{M}(u_\rho), \mathcal{M}(v) >_n \end{cases}$$

De plus, on a: $\lim_{\rho \rightarrow \infty} \|u_\rho - u\| = 0$ où $\|\cdot\|_{m, \Omega'}$ désigne la norme de $\mathcal{H}$ et u la solution du principe (PVI).

Remarque: Si l'on veut prendre en considération les discontinuités des dérivées partielles d'ordre $k+1$ ($k \in \mathbb{N}$), il est nécessaire de prendre $\mathcal{H} = H^m(\Omega') \cap C^k(\bar{\Omega})$ où m satisfait l'inégalité: $m \geq k + 2$.

I-E-2-2 Méthodes de discrétisation et éléments finis

Dans toute la suite, nous supposons que F est une courbe polygonale strictement incluse dans Ω , et afin de ne pas alourdir l'exposé, on fait l'hypothèse que Ω est polygonal.

Soit h un réel positif destiné à tendre vers 0, $\mathcal{T}^h$ une triangulation de Ω' de diamètre inférieur à h et V^h un espace de type éléments finis construite sur $\mathcal{T}^h$, de dimension finie M^h . Afin que V^h reproduise les discontinuités de la surface, il est nécessaire que la triangulation $\mathcal{T}^h$ s'appuie sur F , i.e. que F soit la réunion d'un certain nombre de côtés des triangles de $\mathcal{T}^h$.

On réécrit d'abord les versions discrétisées $(PVI)^h$ et $(PVA)^h$ des principes variationnels PVI) et (PVA):

$$(PVI)^h \text{ trouver } u^h \in V^h \text{ tel que } \mathfrak{E}(u^h) = \min_{v^h \in V^h; \mathcal{M}(v^h) = z} \mathfrak{E}(v^h)$$

où la fonctionnelle "énergie" est : $\mathfrak{E}(v^h) = (v^h, v^h)_{m, \Omega'}$.

De même, on a:

I - 24

$$(PVA)^h \text{ trouver } u^h \in V^h \text{ tel que } \mathfrak{E}(u^h) = \min_{v^h \in V^h} \mathfrak{E}(v^h)$$

où la fonctionnelle "énergie" est : $\mathfrak{E}(v^h) = (v^h, v^h)_{m, \Omega'} + \mathcal{P}(v^h)$.

Alors, sous certaines hypothèses, on montre qu'on obtient des versions discrétisées des théorèmes 1 et 2, c'est-à-dire :

Théorème 3: Le principe $(PVI)^h$ admet une solution unique u^h caractérisée par il existe $\lambda^h \in \mathbb{R}^n$ tel que

$$\begin{cases} \mathcal{M}(u^h) = z \\ \forall v^h \in V^h \quad \langle \lambda^h, v^h \rangle_n + (u^h, v^h)_{m, \Omega'} = 0 \end{cases}$$

De plus, il faut que $M^h \geq n$.

Théorème 4: Le principe $(PVA)^h$ admet une solution unique u_p^h dans $\mathcal{H}$ caractérisée par

$$\begin{cases} u_p^h \in V^h \\ \forall v^h \in V^h \quad (u_p^h, v^h)_{m, \Omega'} = \rho < z - \mathcal{M}(u_p^h), \mathcal{M}(v^h) >_n \end{cases}$$

De plus si l'hypothèse est vérifiée: $\forall v \in \mathcal{H}, \lim_{h \rightarrow 0} \inf_{v^h \in V^h} \|v - v^h\|_{m, \Omega'} = 0$,

alors on a : $\lim_{\rho \rightarrow \infty} \lim_{h \rightarrow 0} \|u - u_p^h\| = 0$

:

On peut voir que les principes $(PVI)^h$ et $(PVA)^h$ se ramènent à la résolution de systèmes linéaires: en effet, soit $(v_j^h)_{1 \leq j \leq M^h}$ une base de V^h , on peut écrire u^h (resp. u_p^h) solution

de $(PVI)^h$ (resp. de $(PVA)^h$) :

$$u^h = \sum_{j=1}^{M^h} \alpha_j^h v_j^h \quad \text{et} \quad u_p^h = \sum_{j=1}^{M^h} \alpha_{j\rho}^h v_j^h \quad \text{où } \alpha_j^h, \alpha_{j\rho}^h \in \mathbb{R} \quad j = 1, \dots, M^h$$

De plus, introduisons les matrices: $A^h = (\mathcal{M}_i(v_j^h))_{1 \leq i, j \leq M^h}$ et $R^h = ((v_i^h, v_j^h)_{m, \Omega'})$.

I - 25

- *Cas de la D^m -spline d'interpolation.*

Le principe $(PVI)^h$ est équivalent au problème: trouver $\lambda^h \in \mathbb{R}^n$, $\alpha^h \in \mathbb{R}^{M^h}$ tel que:

$$\begin{cases} A^h \alpha^h = z \\ (A^h)^T \lambda + R^h \alpha = 0 \end{cases}$$

On remarque que la taille de ce système linéaire est $n + M^h$, donc dépendant du nombre de données.

- *Cas de la D^m -spline d'ajustement.*

Le principe $(PVA)^h$ est équivalent au problème: trouver $\alpha_p^h \in \mathbb{R}^{M^h}$ tel que:

$$[(A^h)^T A^h + R^h] \alpha_p^h = 0$$

Ce système de taille M^h est donc indépendant du nombre n de données, ce qui constitue un point intéressant en pratique (il suffit de se donner M^h pas trop grand).

Pour la construction de V^h , on peut utiliser des éléments conformes de classe C^1 ou C^2 (par exemple Bell, Bogner-Fox-Schmidt) qui ont de bonnes propriétés locales et conduisent à la résolution d'un système linéaire bande défini positif, résolution est effectuée par une méthode directe (par exemple, Cholesky). Puisque l'utilisation des éléments triangulaires sont onéreuses, on prendra ces éléments de type triangle dans un ouvert polygonal pas trop grand contenant F , et de type rectangle dans un ouvert polygonal contenu dans le complémentaire de F tel que Ω soit la réunion de ces deux ouverts.

Ces deux méthodes, contrairement à celles exposées précédemment, ont une assise théorique sérieuse: on montre la convergence de ces procédés pour des données de type ponctuel sous des hypothèses sur Ω' et sur ces données pas trop contraignantes, qui sont en pratique vérifiées. Un intérêt non moins attrayant concerne la méthode d'ajustement: on peut discrétiser dans un espace V^h de dimension nettement inférieure au nombre de données. Le choix du paramètre d'ajustement p réalisant un compromis entre le lissage de u_p et le critère de pénalité, est important: on peut pour cela utiliser la méthode de validation croisée généralisée ([Girard 87] et [Utreras 79]).

I-E-3 Surface plaque mince d'ajustement sous tension et continuité contrôlée

I-E-3-1 Position du problème

Terzopoulos ([Terzopoulos 86a]) propose une approche différente en considérant une fonctionnelle de lissage $\mathcal{J}$ qui est une combinaison de fonctionnelles "spline" généralisées :

$$\mathcal{J}_{\alpha,\beta}(x,y) = \int_{\Omega} \alpha(x,y) \left\{ \beta(x,y) \left[v_{xx}^2 + 2 v_{xy}^2 + v_{yy}^2 \right] + (1 - \beta(x,y)) \left[v_x^2 + v_y^2 \right] \right\} dx dy$$

où Ω est un domaine ouvert borné de $\mathbb{R}^2$ et α, β sont des fonctions poids positives à valeurs dans $[0,1]$. Cette fonctionnelle est une combinaison convexe pondérée de la fonctionnelle de lissage $|\cdot|_2^2$ correspondant à la spline plaque mince et de la fonctionnelle de lissage $|\cdot|_1^2$ correspondant à la spline membrane. $\mathcal{J}_{\alpha,\beta}$ peut être pensée comme une fonctionnelle générant une *surface spline plaque mince sous tension* où $\alpha(x,y)$ est un paramètre de "rigidité spatiale" et $[1 - \beta(x,y)]$ un paramètre de "tension surfacique spatiale". Notons aussi que cette fonctionnelle est une illustration des fonctionnelles pondérées citées dans les exemples au début de ce paragraphe avec $w_0(x,y) = 0$, $w_1(x,y) = \alpha(x,y) [1 - \beta(x,y)]$, $w_2(x,y) = \alpha(x,y) \beta(x,y)$.

On peut contrôler les propriétés locales de continuité de la surface spline plaque mince sous tension en tout point de Ω , en spécifiant les valeurs des *fonctions de contrôle de continuité* α et β en ce point. Lorsque β tend vers 1, $\mathcal{J}_{\alpha,\beta}$ tend vers la fonctionnelle générant une spline plaque mince (de classe C^1), tandis que lorsque β tend vers 0, $\mathcal{J}_{\alpha,\beta}$ tend vers la fonctionnelle générant une spline membrane (de classe C^0) ; les valeurs intermédiaires génèrent une surface hybride de classe C^1 qui satisfait les propriétés des deux splines constituantes. De même lorsque α tend vers 0, cela désactive toute continuité et permet à la surface générée de se "fracturer" librement.

Les surfaces générées devant être capable de préserver les discontinuités de valeur et de dérivées le long de lignes connues, ne doivent pas introduire des discontinuités en d'autres localisations. Ceci peut être réalisé si (i) loin des lignes de discontinuité de valeurs et de dérivées, la surface est au moins de classe C^1 , (ii) le long des lignes de discontinuité de dérivée, elle exhibe un lissage C^0 d'une membrane, et (iii) le long des lignes de discontinuité de valeur, la fonctionnelle de lissage est désactivée telle que la surface reconstruite est libre de se "fracturer" localement. On doit alors manipuler la fonctionnelle $\mathcal{J}_{\alpha,\beta}$ comme suit:

I - 27

- le long des points de *non-discontinuité*, α et β sont non nulles.
- le long des points de *discontinuité de dérivée*, β est mis à zéro.
- le long des points de *discontinuité de valeur*, α est mis à zéro.

Les fonctionnelles de mesure $M_i(v)$ sont les fonctionnelles d'interpolation ponctuelles

$M_i(v) = v(a_i)$ où les a_i sont des points de „² vérifiant la condition de 1-unsolvence, ce qui se traduit géométriquement par le fait qu'ils ne se situent pas sur une même droite. Dans ce cas, on montre ([Terzopoulos 86a]) que l'existence et l'unicité sont assurées. La fonctionnelle de pénalité est celle définie dans les exemples du sous-paragraphe précédent.

Le choix de ces fonctionnelles de contrôle de discontinuité est un problème délicat dans un cadre général. Il est possible, via un principe variationnel augmenté décrit ci-dessous de détecter les discontinuités non connues à l'avance (problème de localisation). On augmente la fonctionnelle énergie $\mathfrak{E}(v, \alpha, \beta) = \mathfrak{L}_{\alpha, \beta}(v) + \mathcal{P}(v) + \mathfrak{D}(\alpha, \beta)$, où $\mathfrak{D}(\alpha, \beta)$, dénommée fonctionnelle "discontinuité", impose des critères de bonne structure de discontinuité. Ainsi, le principe variationnel correspondant devient:

$$(PVA) \quad \text{trouver } u, \bar{\alpha}, \bar{\beta} \text{ tel que } \mathfrak{E}(u, \bar{\alpha}, \bar{\beta}) = \inf_{v, \alpha, \beta} \mathfrak{E}(v, \alpha, \beta)$$

La fonctionnelle $\mathfrak{D}(\alpha, \beta)$ doit transformer les configurations de discontinuité de valeur et de dérivée en énergies positives. Dans sa forme la plus simple, cette fonctionnelle peut augmenter avec le nombre de discontinuités, par exemple:

$$\mathfrak{D}(\alpha, \beta) = \int_{\Omega} \mu_l \{ 1 - \alpha(x, y) \} + \mu_d \{ 1 - \beta(x, y) \} dx dy$$

où μ_l et μ_d sont des paramètres positifs de contributions des discontinuités de valeur et de dérivée respectivement. Plus généralement, on considère la fonctionnelle "discontinuité":

$$\mathfrak{D}(\alpha, \beta) = \int_{\Omega} \mu_l F(\alpha(x, y)) + \mu_d D(\beta(x, y)) dx dy$$

où F, D sont des fonctions décroissantes sur l'intervalle $[0, 1]$ tel que $F(0) = D(0) = 1$ et $F(1) = D(1) = 0$.

Bien que la fonctionnelle énergie $\mathfrak{E}$ ait un minimum unique pour α et β fixés, ce n'est

I - 28

plus le cas lorsque on permet la variation des fonctions de contrôle de discontinuité lors de la minimisation. La non-convexité de cette énergie ayant plusieurs minimas locaux rend le problème encore plus difficile à résoudre numériquement.

Une version sophistiquée proposée par divers auteurs (par exemple [Geman 85], [Terzopoulos 86a]) est implantée comme une "table de consultation" rangeant les configurations de discontinuité locales suivant des critères de "bonne continuation". On alors obtenir une bonne approximation de la fonctionnelle énergie comme une suite de fonctionnelles plus simples en un nombre fini d'étapes. Pour α et β fixés, $\mathcal{E}(v, \alpha, \beta)$ devient quadratique. Alors, en considérant les fonctions de contrôle de discontinuité comme des paramètres durant chaque étape, le problème devient un problème d'optimisation convexe pour $v(x, y)$ qui est résolu par une méthode multigrille (voir paragraphe I-E-3-2). Au début de chaque étape, des valeurs de paramètres sont calculés de manière discrète ou continue, à partir du minimum obtenu à l'étape précédente. Les solutions finales proches du minimum global $\{u(x, y), \alpha(x, y), \beta(x, y)\}$ requièrent alors plusieurs étapes de ce procédé dynamique non linéaire.

I-E-3 -2 Méthodes de discrétisation , éléments finis et méthodes multigrilles.

Soit $\mathcal{T}^h$ un triangulation de Ω que l'on précisera ci-dessous et V^h un espace de type éléments finis construite sur cette triangulation. Le choix de cet espace, contrairement à ceux retenus dans le sous-paragraphe ci-dessus est constitué d'éléments non conformes, i.e. que V^h n'est pas inclus dans l'espace $\mathcal{H}$. Ce choix est justifié par le fait que les espaces d'éléments finis conformes les plus simples (par exemple celui de Bell dont les éléments sont de classe C^1 le long des frontières des triangles) sont des polynômes de degré total élevé (5 pour les éléments de Bell) avec beaucoup de degrés de liberté (18 pour les éléments de Bell). De ce fait, c'est une tâche fastidieuse (mais pas impossible) de trouver les coefficients de la matrice du système associé au problème discret.

Par conséquent, on peut se poser des questions concernant l'existence, l'unicité de la solution approximante ainsi que la convergence. Toutefois, sous certaines conditions spécifiques, que nous ne détaillerons pas ici (nous renvoyons à l'article de Terzopoulos ([Terzopoulos 83]) et au livre de Strang et Fix ([Strang 73]), que cette condition de conformité peut être violée sans conséquences désastreuses.

Le fait d'employer des éléments non conformes ne permet pas d'évaluer la fonctionnelle énergie $\mathcal{E}(v^h)$ due aux discontinuités des dérivées de v^h apparaissant le long des frontières des éléments. Afin d'éviter cet écueil, les discontinuités entre les éléments sont ignorées en calculant les fonctionnelles locales de lissage avec chaque élément et de sommer toutes ces contributions, i.e. :

$$\mathcal{J}_{\alpha, \beta}^h(v^h) = \sum_{T \in \mathcal{T}^h} \mathcal{J}_{\alpha, \beta}(v^h)|_T$$

Le principe variationnel d'ajustement (PVA)^h devient alors:

I - 29

$$(PVA)^h \quad \text{Trouver } u^h \in V^h \text{ tel que } \mathfrak{L}^h(u^h) = \min_{v^h \in V^h} \mathfrak{L}^h(v^h)$$

où l'on prend la fonctionnelle modifiée: $\mathfrak{L}^h(v^h) = \mathfrak{L}_{\alpha\beta}^h(v^h) + \mathcal{P}(v^h)$.

Dans ces conditions, on montre l'existence et l'unicité de la solution (en fait, la démonstration est faite dans un cadre général).

Dans un but de simplification de la forme du problème, et à cause des applications choisies par l'auteur (reconstruction d'images discrètes), on suppose que Ω est un ouvert rectangulaire. Soit une triangulation uniforme $\mathcal{T}^h$ de Ω formée d'éléments carrés identiques où h est la longueur d'un élément T . De plus, toujours à cause du champ d'applications, on fait l'hypothèse restrictive que les données z_i $i = 1, \dots, n$ sont donnés sur un sous-ensemble D contenu dans celui des sommets de cette triangulation, et par conséquent, on les renomme $z_{i,j}$ pour $(i,j) \in D$.

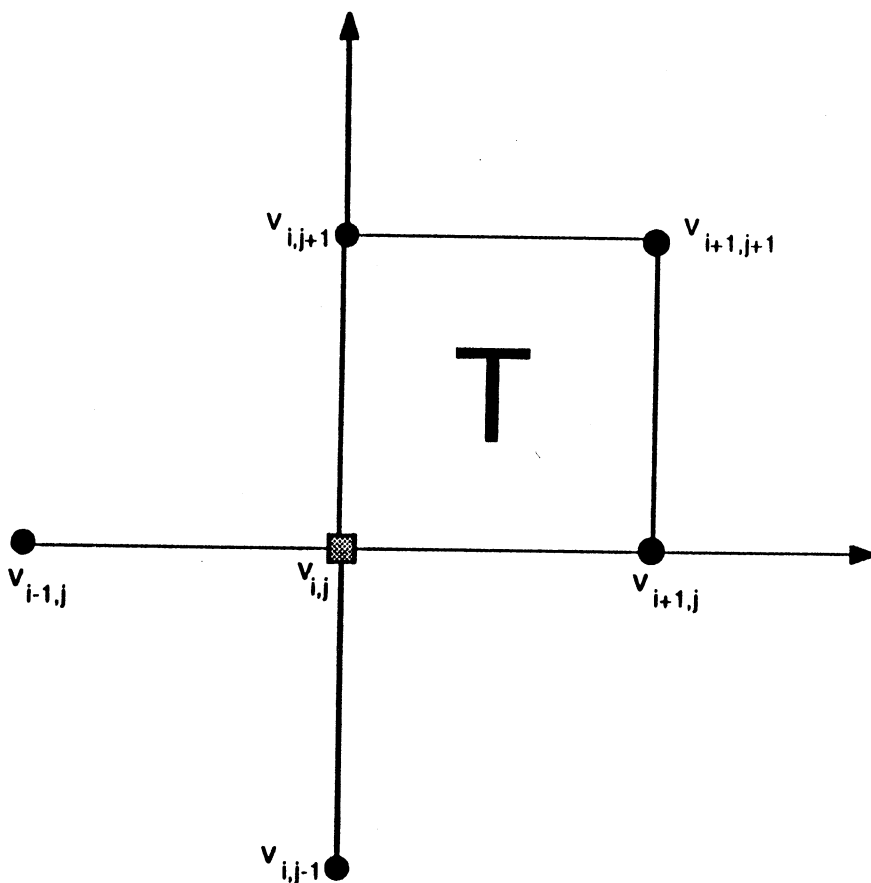


Figure I-5

I - 30

Définissons un polynôme p^T avec un élément du domaine. Le choix retenu est une quadrique:

$$p^T(x, y) = a x^2 + b xy + c y^2 + d x + e y + f$$

Les 6 paramètres $a, \dots, f$ sont déterminés en utilisant les valeurs de la fonction v^h aux noeuds de la triangulation formant un ensemble p^T -unisolvent. Un choix approprié de noeuds p^T -unisolvent sont les 6 noeuds montrés sur la figure ci-dessus, où les valeurs aux noeuds sont notés $v_{i,j}^h$. En résolvant le système d'équations associées aux valeurs valeurs, on trouve que ces six paramètres sont donnés par:

$$\begin{cases} a = \frac{1}{2h^2} (v_{i+1,j} - 2v_{i,j} + v_{i-1,j}) \\ b = \frac{1}{h^2} (v_{i+1,j+1} - v_{i,j+1} - v_{i+1,j} + v_{i,j}) \\ c = \frac{1}{2h^2} (v_{i,j+1} - 2v_{i,j} + v_{i,j-1}) \\ d = \frac{1}{2h} (v_{i+1,j} - v_{i-1,j}) \\ e = \frac{1}{2h} (v_{i,j+1} - v_{i,j-1}) \\ f = v_{i,j} \end{cases}$$

Cet élément est évidemment non conforme car le nombre de degré de liberté, en l'occurrence ici 6, est insuffisant pour garantir une continuité C^1 sur les frontières. En fait, seule la continuité est maintenue aux noeuds eux-mêmes. Mais on démontre qu'en utilisant cet élément carré non conforme, l'existence et l'unicité de la solution sont préservées, ainsi que la convergence (**[Terzopoulos 83]**).

L'étape suivante est la discrétisation du principe (PVA). L'approche classique de la M.E.F. est de trouver des fonctions de base v_i^h associées aux éléments finis engendrant V^h , puis après avoir écrit l'équation variationnelle associée au principe variationnel, de discrétiser celle-ci; ceci conduit à la résolution d'un système linéaire dont les coefficients sont estimés à partir des dérivées partielles des fonctions de base. Malheureusement, l'utilisation d'éléments non conformes n'est pas possible ici, en raison de la nature discontinue des fonctions de base rendant leurs manipulations délicates.

Une alternative évitant la manipulation et la dérivation des fonctions de base, est de résoudre directement le principe variationnel (PVI)^h en trouvant une approximation "raisonnable" de la fonctionnelle énergie $\mathcal{E}^h(\cdot)$ et de minimiser celle-ci. On fera alors l'identification de la solution approximante v^h avec ses valeurs $v_{i,j}^h$ aux noeuds de la triangulation.

I - 31

L'approximation de $\mathcal{E}^h(\cdot)$ est de ne faire intervenir que les valeurs aux noeuds de la triangulation, par conséquent d'occulter dans un premier temps, toute information issue de l'intérieur des triangles. Pour ce faire, on discrétise les dérivées partielles intervenant dans la fonctionnelle de lissage en utilisant uniquement les valeurs de la surface aux sommets de la triangulation.

Les éléments non conformes décrits ci-dessus conduisent aux approximations de type différence finie en $O(h^2)$, des dérivées partielles en un noeud (i,j) de la manière suivante ([Terzopoulos 83]):

$$v_x^h \approx \frac{1}{h} (v_{i+1,j}^h - v_{i,j}^h)$$

$$v_y^h \approx \frac{1}{h} (v_{i,j+1}^h - v_{i,j}^h)$$

$$v_{xx}^h \approx \frac{1}{h^2} (v_{i+1,j}^h - 2v_{i,j}^h + v_{i-1,j}^h)$$

$$v_{xy}^h \approx \frac{1}{h^2} (v_{i+1,j+1}^h - v_{i+1,j}^h - v_{i,j+1}^h + v_{i,j}^h)$$

$$v_{yy}^h \approx \frac{1}{h^2} (v_{i,j+1}^h - 2v_{i,j}^h + v_{i,j-1}^h)$$

Remarque: Puisque v^h est un polynôme quadratique dans chaque élément T du domaine, ses dérivées partielles secondes sont constantes et on montre aisément que les approximations de celles-ci sont en fait des formules exactes.

En faisant les approximations suivantes $\alpha(x,y) \approx \alpha_{i,j}^h$ et $\beta(x,y) \approx \beta_{i,j}^h$ au noeud (i,j) , et en introduisant les expressions discrètes des dérivées, on obtient la fonctionnelle discrète:

$$\begin{aligned} \mathcal{J}_{\alpha\beta}^h(v^h) = \sum_{(i,j) \in D} \alpha_{i,j}^h & \left\{ \beta_{i,j}^h \left[\left(v_{i+1,j}^h - 2v_{i,j}^h + v_{i-1,j}^h \right)^2 \right. \right. \\ & + 2 \left(v_{i+1,j+1}^h - v_{i+1,j}^h - v_{i,j+1}^h + v_{i,j}^h \right)^2 \\ & + \left. \left(v_{i,j+1}^h - 2v_{i,j}^h + v_{i,j-1}^h \right)^2 \right] \\ & + \left. \left[1 - \beta_{i,j}^h \right] \left[\left(v_{i+1,j}^h - v_{i,j}^h \right)^2 + \left(v_{i,j+1}^h - v_{i,j}^h \right)^2 \right] \right\} \end{aligned}$$

I - 32

De manière analogue, on obtient une version discrétisée de la fonctionnelle pénalité:

$$\mathcal{P}^h(v^h) = \sum_{(i,j) \in D} (v_{i,j}^h - z_{i,j})^2$$

La solution u^h solution du principe $(PVI)^h$ satisfait la condition de minimum:

$$\nabla \mathcal{E}(u^h) = \nabla \mathcal{L}_{\alpha\beta}(u^h) + \nabla \mathcal{P}^h(u^h)$$

où ∇ est l'opérateur gradient $\left(\frac{\partial}{\partial u_{i,j}^h} \right)$.

Cette condition conduit alors à un système linéaire d'équations simultanées satisfaites par les $v_{i,j}^h$ dénommées *équations nodales*. On peut écrire ce système sous la forme:

$$A^h u^h = f^h$$

où A^h est une matrice dépendant de α, β et de h et creuse dont chaque ligne a au plus 13 entrées non nulles. Les méthodes itératives (Gauss-Seidel, Jacobi, relaxations, méthode du gradient conjugué) pour la résolution de ce système exhibent une convergence très lente, et par conséquent inefficaces surtout lorsque le nombre de noeuds est important (supérieur à 10000).

Récemment, des *méthodes multigrilles* ([Brandt 77]) ont été développées pour la résolution de problèmes elliptiques aux limites où on a montré qu'elles convergent en un nombre d'opérations environ égal au nombre de noeuds. Cette réduction spectaculaire résulte de l'introduction d'une hiérarchie d'espaces d'éléments finis

$V^{h_1}, \dots, V^{h_N}$ sur Ω , où N est le nombre de niveaux de hiérarchie et $h_1 > \dots > h_N$ sont les

longueurs des côtés des éléments du domaine à chaque niveau. Bien qu'en théorie, il n'y ait aucune restriction sur les relations entre les h_k , le procédé se simplifie si chaque élément carré du domaine à un niveau k , est subdivisé en 4 éléments carrés identiques au niveau suivant $k+1$, i.e. de prendre $h_k = 2 h_{k+1}$.

Une hiérarchie possible de problèmes est donc donnée par la hiérarchie de N systèmes linéaires de la forme:

$$A^{h_k} u^{h_k} = f^{h_k} \quad k=1, \dots, N$$

où les solutions discrètes $u_{i,j}^{h_k}$ définissent une séquence de fonctions $u^{h_k} \in V^{h_k}, k=1, \dots, N$

La seule façon de maintenir la précision à travers la hiérarchie est de permettre des interactions dans lesquelles l'information précise dans les niveaux fins est employée dans les niveaux grossiers, ce qui signifie que la hiérarchie des systèmes ci-dessus doit être couplée et résolue simultanément.

I - 33

Un couplage approprié est donné par les *équations multiniveaux* suivant pour N niveaux:

$$A^{h_k} u^{h_k} = g^{h_k} \quad k=1, \dots, N$$

où

$$g^{h_N} = f^{h_N}$$

et

$$g^{h_k} = A^{h_k} \left(I_{k+1,k} u^{h_{k+1}} \right) + I_{k+1,k} \left(g^{h_{k+1}} - A^{h_{k+1}} u^{h_{k+1}} \right) \quad k=1, \dots, N$$

Ici $I_{k+1,k}$ désigne "l'interpolation" d'un niveau fin $k+1$ sur un niveau plus grossier k (par injection par exemple).

Notons que le second membre f^{h_k} du $k^{\text{ième}}$ problème n'apparaît que dans le niveau le plus fin N . Les seconds membres des niveaux grossiers sont modifiés dans le but de maintenir la précision au niveau le plus fin, c'est à dire qu'on impose aux solutions de coïncider: $u^{h_1} = I_{2,1} u^{h_2} = \dots = I_{2,1} \dots I_{N,N-1} u^{h_N}$. La différence entre le second membre original et le second membre corrigé, $f^{h_k} - g^{h_k}$, peut être interprétée comme une estimation de l'erreur d'approximation au niveau k relativement le plus fin.

Algorithme multiniveaux d'approximation d'une surface.

Dans les algorithmes multigrilles, la hiérarchie des niveaux communique via un flot bidirectionnel d'information, qui consiste en des procédés d'injection "fin-à-grossier", notés $I_{k+1,k}$ et des procédés d'interpolation "grossier-à-fin", notés $I_{k,k+1}$. A travers cette interaction, des hiérarchies précises d'approximation de surfaces sont générées simultanément avec nettement moins de calculs effectués qu' en résolvant directement le système au niveau le plus fin.

Plusieurs schémas ont été proposés ([Brandt 77],). Le schéma décrit ci-dessous (Algorithme A), établi par Terzopoulos ([Terzopoulos 83]), est approprié pour l'approximation de surfaces. Il progresse du niveau le plus grossier $n=1$ vers le niveau le plus fin $n=N$, et invoque à chaque étape de la boucle un algorithme B qui est un cycle multiniveaux.

Le cycle multiniveaux (algorithme B) décrit ci-dessous démarre au niveau n , faisant plusieurs cycles sur les niveaux plus grossiers $k=n-1, \dots, 1$. Il réalise des itérations de relaxation tant que le taux de convergence souhaité n'est pas atteint. A un niveau donné, si le taux de convergence décroît trop lentement, un transfert est effectué au

I - 34

Algorithme multigrilles pour l'approximation de surfaces

Algorithme A - Algorithme multiniveaux d'approximation de surfaces.

A1 - résolution de l'équation au niveau le plus grossier

- Calculer par relaxation une solution approximante u^{h_1} du système le plus grossier

$$A^{h_1} u^{h_1} = f^{h_1}$$

- $n := 1$

A2 - création d'un niveau le plus fin n .

répéter

a) $n := n + 1$; {on incrémente le niveau courant le plus fin }

b) $v_0^{h_n} := I_{n-1,n} u^{h_{n-1}}$ { première approximation au nouveau niveau }

c) Invoquer l'algorithme B (voir ci-dessous)

jusqu'à $n = N$.

Algorithme B - Cycle multiniveaux.

B1 - Initialisation du niveau courant n le plus fin.

- $g^{h_n} := f^{h_n}$ { affectation du second membre au niveau n }

- $v^{h_n} := v_0^{h_n}$ { approximation initiale de la solution }

- $\epsilon_n := 0$ { valeur temporaire }

- $k := n$

B2 - Démarrage d'une nouvelle opération au niveau k .

- $e_k^a := \infty$

B3 - Itération de relaxation.

- Faire une itération de relaxation pour le système $A^{h_k} u^{h_k} = g^{h_k}$.

- Calculer le résidu $e_k := \|g^{h_k} - A^{h_k} u^{h_k}\|$ { $\|\cdot\|$ est une norme quelconque }

B4 - Test de la convergence et de son taux.

- si $e_k \leq \epsilon_k$ alors { la convergence a été obtenue au niveau courant k }

si $k < n$ alors { on corrige alors le niveau suivant plus fin }

a) Faire la correction : $v^{h_{k+1}} := v^{h_{k+1}} + I_{k,k+1} (u^{h_k} - I_{k+1,k} v^{h_{k+1}})$

b) $k := k + 1$

c) aller à B2

sinon fin de l'algorithme B

sinon

si $k = 1$ aller à B3.

si $e_k \leq v e_k^a$ alors { le taux de convergence est satisfaisant }

a) $e_k^a := e_k$

b) aller à B3

sinon { le taux de convergence est trop lent }

aller à B5

B5 - Transfert vers un niveau plus grossier.

- $v^{h_{k-1}} := I_{k,k-1} v^{h_k}$ { première approximation au niveau grossier }

- $g^{h_{k-1}} := A^{h_{k-1}} v^{h_{k-1}} + I_{k,k-1} (g^{h_k} - A^{h_k} v^{h_k})$

- $\epsilon_{k-1} := v e_k$

- Si $k = n$ alors $\epsilon_n := \frac{1}{4} \|f^{h_{k-1}} - g^{h_{k-1}}\|$

- $k := k - 1$

- aller à B2

I - 35

niveau suivant plus grossier et lorsqu'une approximation suffisante est obtenue, une correction est effectuée au niveau suivant plus fin. Soit ϵ_k une tolérance pour la résolution des équations au niveau k . μ et ν sont des paramètres dont on donnera des valeurs appropriées ci-dessous.

- Remarques:**
- 1) La solution à l'étape 1 dans l'algorithme A est calculée aisément car il y a peu de variables nodales.
 - 2) L'opération de relaxation réalisée dans l'étape B2 est par exemple une itération de Gauss-Seidel relaxée, ou une méthode de descente.
 - 2) Les normes calculées dans les étapes B3 et B5 sont par exemple les normes discrètes L^2 ou L^∞ .
 - 3) De bonnes valeurs préconisées par Terzopoulos ([Terzopoulos 83]) pour les paramètres μ et ν sont: $\mu = 0.2$ et $\nu = 0.8$. Ces valeurs sont en fait des valeurs presque optimales, obtenues en faisant une analyse locale de Fourier afin d'avoir une bonne prédiction du taux de convergence ([Brandt 77]).
 - 4) Pour une interpolation "grossier-à-fin" $I_{k,k+1}$ dans l'étape B4, l'interpolation par l'élément polynômial p^T peut être employée. De même, de simples injections fonctionnent bien pour des transferts $I_{k+1,k}$ dans les étapes B4 et B5.

CHAPITRE II:

APPLICATION DE L'INF-CONVOLUTION SPLINE POUR L'APPROXIMATION DE SURFACES SINGULIERES

II -1

II-A Introduction

Soit $H^2(\mathbb{R}^2)$ l'espace des distributions dont toutes les dérivées partielles prises à l'ordre 2 sont des fonctions de carré sommable sur $\mathbb{R}^2$. Pour $x \in H^2(\mathbb{R}^2)$, on peut définir la quantité:

$$E_2(x) = \int_{\mathbb{R}^2} \left(\frac{\partial^2 x}{\partial s^2} \right)^2 + 2 \left(\frac{\partial^2 x}{\partial s \partial t} \right)^2 + \left(\frac{\partial^2 x}{\partial t^2} \right)^2 ds dt$$

Les fonctions-splines d'interpolation classique qui sont obtenues en minimisant le critère de lissage $E_2(x)$, avec des conditions d'interpolation ponctuelles $x(t_i) = z(t_i)$, où $t_i \in \mathbb{R}^2$ pour $i=1, \dots, n$, conduit à une approximation raisonnable de z lorsque celle-ci est dérivable, notamment lorsque $z \in H^2(\mathbb{R}^2)$. La solution de ce problème, dénommée spline "plaque mince" (voir paragraphe I-E-3) a une expression analytique simple due à Duchon ([Duchon 80]) et correspond à la position d'une plaque mince flexible passant par les points $(t_i, z(t_i))$ $i=1, \dots, n$, $E_2(x)$ étant considérée comme une approximation de l'énergie de flexion d'une plaque coïncidant avec le graphe de x lorsque ses dérivées partielles du premier ordre ne sont pas trop "grandes".

Soit maintenant le cas d'une fonction z continue, mais présentant des points de discontinuités de valeurs (ou de dérivée normale) le long d'une ligne polygonale F . Plus précisément, z appartient à l'espace des fonctions continues sur $\mathbb{R}^2$ telle que la restriction de z à l'ensemble $\mathbb{R}^2 \setminus F$, soit dans dans $H^2(\mathbb{R}^2 \setminus F)$. La fonction-spline plaque mince conduit alors à une très mauvaise approximation de f , c'est à dire qu'il se produit un effacement de la discontinuité qui est en général accompagnée d'oscillations.

Pour pallier à cette difficulté, il est alors naturel d'introduire un espace $V = L(v_1, \dots, v_m)$ de fonctions de base "traduisant" la discontinuité de la surface z le long de la ligne polygonale F et de minimiser par rapport à $x \in H^2(\mathbb{R}^2)$ et $(\alpha_1, \dots, \alpha_m)^T \in \mathbb{R}^m$ la quantité $E_2(x)$ avec les conditions d'interpolation ponctuelle:

$$x(t_i) + \sum_{j=1}^m \alpha_j v_j(t_i) = z(t_i) \quad i = 1, \dots, n$$

Si les valeurs de la fonction f ne sont pas connues de façon exacte, c'est-à-dire si les données ponctuelles sont $z_i = z(t_i) + \epsilon_i$ où les ϵ_i sont des erreurs, alors on introduit de manière analogue les fonctions-splines d'ajustement. Plus précisément, on minimise par

II -2

rapport à $x \in H^2(\mathbb{R}^2)$ et $(\alpha_1, \dots, \alpha_m) \in \mathbb{R}^m$ la quantité:

$$E_2(x) + \rho \sum_{i=1}^n \left(x(t_i) + \sum_{j=1}^m \alpha_j v_j(t_i) - z_i \right)^2$$

Pour justifier la validité de ces deux schémas, nous utilisons la théorie de l'inf-convolution spline développée par Laurent ([Laurent 86a]) dont nous rappelons le cadre et les principaux théorèmes s'y dégageant dans le paragraphe suivant.

Dans le paragraphe II-C, nous modélisons le problème d'une part grâce aux résultats du paragraphe B et d'autre part, en proposant une base de fonctions discontinues où à dérivée discontinue le long de F . Nous montrons aussi, lorsque le nombre de points de données est grand, comment la méthode de "recollement" de sous-plaques due à Paihua ([Paihua 78b]) doit être adaptée dans ce cas.

Enfin, dans le paragraphe II-D, nous présentons les résultats numériques obtenus et concluons sur l'efficacité de l'outil d'inf-convolution spline.

II -3

II-B Inf-convolution spline

Nous rappelons dans ce paragraphe la théorie de l'inf-convolution spline développée par P.J. Laurent ([Laurent 80, Laurent 81, Laurent 82]) ainsi que les résultats principaux s'y rattachant.

II-B-1 Sous-espaces semi-hilbertiens

Soient E et E' deux espaces vectoriels topologiques réels localement convexes en dualité par la forme bilinéaire $\langle x, x' \rangle$ où $x \in E$ et $x' \in E'$.

Définition: Un sous-espace vectoriel X de E muni d'un semi-produit scalaire noté (x, y) où $x, y \in X$ et de la semi-norme associée $|x| = (x, x)^{1/2}$ est dit semi-hilbertien si:

a / l'espace nul $N = \{ x \in X / |x| = 0 \}$ de la semi-norme est de dimension finie.

b/ l'espace quotient X/N muni de la norme canoniquement associée à la semi-norme est

- complet
- topologiquement inclus dans E/N (E étant muni de la topologie faible).

Exemples: 1) Soit $X = H^2(\mathbb{R}^2)$ espace des fonctions dont toutes les dérivées partielles d'ordre 2 prises au sens des distributions sont des fonctions de carré sommable sur $\mathbb{R}^2$ est un sous-espace semi-hilbertien de $E = A(\mathbb{R}^2)$ (espace des applications réelles définies sur $\mathbb{R}^2$) pour le semi-produit scalaire:

$$(x, y) = \sum_{i, j \in \{1, 2\}} \int_{\mathbb{R}^2} D_i D_j x(t) \cdot D_i D_j y(t) dt$$

- et de la semi-norme associée qui s'écrit ici:

$$|x|^2 = \sum_{i, j \in \{1, 2\}} \int_{\mathbb{R}^2} (D_i D_j x(t))^2 dt$$

où D_i désigne l'opérateur de dérivation partielle par rapport à la $i^{\text{ème}}$ variable.

X a pour espace nul $N = P_1$ l'espace des polynômes à deux variables de degré total inférieur ou égal à 1.

II -4

2) Plus généralement, soit $\tilde{H}^s(\mathbb{R}^d)$ (où $d \in \mathbb{N}^*$, $s \in \mathbb{R}^+$) l'espace des distributions tempérées x tel que la transformée de Fourier $\mathcal{F}x$ soit une fonction vérifiant:

$$\int_{\mathbb{R}^d} |\mathcal{F}x(\xi)|^2 \|\xi\|^{2s} d\xi < +\infty$$

où $\|\cdot\|$ désigne la norme euclidienne dans $\mathbb{R}^d$.

Si $s < d/2$, $\tilde{H}^s(\mathbb{R}^d)$ est un sous-espace hilbertien de $E = A(\mathbb{R}^d)$ (espace des applications réelles définies sur $\mathbb{R}^d$) muni de la norme $\|x\|_s = \|\|\xi\|^s \cdot \mathcal{F}x\|_{L^2(\mathbb{R}^d)}$.

Soit $X^{m,s}(\mathbb{R}^d)$ l'espace des distributions dont toutes les dérivées partielles d'ordre m sont dans $\tilde{H}^s(\mathbb{R}^d)$. Munissons $X^{m,s}(\mathbb{R}^d)$ de la semi-norme:

$$\|x\|_{m,s}^2 = \sum_{i_j \in \{1,\dots,d\}; j \in \{1,\dots,m\}} \|D_{i_1} \dots D_{i_m} x\|_s^2$$

et du semi-produit scalaire:

$$(x,y)_{m,s} = \sum_{\substack{i_j \in \{1,\dots,d\} \\ j \in \{1,\dots,m\}}} \int_{\mathbb{R}^d} \mathcal{F} D_{i_1} x(\xi) \cdot \mathcal{F} D_{i_1} y(\xi) \cdot \|\xi\|^{2s} d\xi$$

Si $s < d/2$ et $s > d/2 - m$ alors $X^{m,s}$ est un sous-espace semi-hilbertien de E ([Duchon 80]), d'espace nul P_{m-1} , espace des polynômes à d variables de degré total inférieur ou égal à $m-1$.

Le cas particulier $(d, m, s) = (2, 2, 0)$, permet de retrouver ici l'exemple 1.

II-B-2 Semi-noyau

Dans le but d'obtenir une formulation explicite des fonctions-spline résultant de la minimisation d'une semi-norme quadratique, J. Duchon ([Duchon 80]) a introduit la notion de semi-noyau reproduisant. De manière plus générale, soient L un sous-espace vectoriel de E' , X un sous-espace semi-hilbertien de E et H une application linéaire de L dans E .

II -5

Définition: H est un semi-noyau de L si $\forall \ell \in L \cap N^0$, on a:

$$a/ H(\ell) \in X.$$

$$b/ \langle x, \ell \rangle = (x, H(\ell)) \quad \forall x \in X$$

où N^0 est l'ensemble des éléments de E' nuls sur N (i.e. l'ensemble polaire de N dans E').

Propriétés: H possède deux propriétés essentielles:

$$\bullet \forall \ell \in L \cap N^0 \quad \langle H(\ell), \ell \rangle = (H(\ell), H(\ell)) \geq 0 \quad (\text{"semi-positivité"})$$

$$\bullet \forall \ell, m \in L \cap N^0, \forall m \in L \cap N^0$$

$$\langle H(m), \ell \rangle = (H(m), H(\ell)) = (H(\ell), H(m)) = \langle H(\ell), m \rangle \quad (\text{"semi-symétrie"})$$

Remarques: 1) Cette application H n'est pas définie de manière unique car si G est une application linéaire de L dans N, alors H + G convient aussi (ceci grâce à b/).

2) Si $L = E'$ alors l'application H' de N^0 dans X/N définie par $H'(\ell) = H(\ell) + N$ est unique. Elle peut être considérée comme une application de $(E/N)'$ dans (E/N) et alors elle est symétrique, positive et faiblement continue. Cette application est en fait le noyau de Schwartz du sous espace-hilbertien X/N de E/N .

Avant de définir l'inf-convolution spline, rappelons le contexte général des fonctions-spline.

II-B-3 Définition, caractérisation et calcul des fonctions-spline

a/ Définition

Considérons le problème de minimisation suivant :

$$(P) \quad \min_{x \in X} |x|^2 \\ \langle x, \ell_i \rangle = y_i ; 1 \leq i \leq n$$

où $\ell_i \in E'$ et $y_i \in \mathbb{R} ; i = 1, \dots, n$.

Si $\langle p, \ell_i \rangle = 0$ pour $i = 1, \dots, n$ et $p \in N$ impliquent $p = 0$, alors ce problème a une solution unique que l'on appelle fonction-spline d'interpolation.

II-6

b/ Caractérisation

Si les ℓ_i sont dans L pour lequel on a défini un semi-noyau H , alors on peut caractériser la solution du problème à l'aide du semi-noyau grâce au

Théorème1:

Un élément $\sigma \in X$ vérifiant $\langle \sigma, \ell_i \rangle = y_i, i = 1, \dots, n$ est solution du problème (P) si et seulement si il existe des réels λ_i tel que:

$$\sigma = \sum_{i=1}^n \lambda_i H(\ell_i) + p \quad \text{où } p \in N$$

où $\{\lambda_i\}_{i=1, \dots, n}$ vérifie :

$$\sum_{i=1}^n \lambda_i \langle \omega, \ell_i \rangle = 0 \quad \forall \omega \in N$$

Démonstration : voir [Laurent 86a].

c/ Calcul de la fonction spline.

Le calcul de la solution σ s'effectue comme suit :

soit $(\omega_j)_{j=1, \dots, p}$ une base de N et écrivons la solution sous la forme:

$$\sigma = \sum_{i=1}^n \lambda_i H(\ell_i) + \sum_{j=1}^p \alpha_j \omega_j$$

alors les coefficients λ_i ($i = 1, \dots, n$) et α_j ($j = 1, \dots, p$) sont obtenus en résolvant le système linéaire de taille $n + p$:

$$\begin{cases} \sum_{i=1}^n \lambda_i \langle H(\ell_i), \ell_k \rangle + \sum_{j=1}^p \alpha_j \langle \omega_j, \ell_k \rangle = y_k & k = 1, \dots, n \\ \sum_{i=1}^n \lambda_i \langle \omega_j, \ell_i \rangle = 0 & j = 1, \dots, p \end{cases}$$

Si l'on note:

- $\mathcal{K}$ la matrice $n \times n$ de terme général $\langle H(\ell_i), \ell_k \rangle$ $i, j = 1, \dots, n$
- $\mathcal{T}$ la matrice $n \times p$ de terme général $\langle \omega_j, \ell_k \rangle$ $j = 1, \dots, p; k = 1, \dots, n$
- λ, y les vecteurs de taille n , respectivement de composantes λ_i, y_i $i = 1, \dots, n$
- α le vecteur de taille p de terme α_j $j = 1, \dots, p$

II-7

on peut alors écrire le système linéaire symétrique précédent sous la forme matricielle ci-dessous :

$$\begin{cases} \mathcal{K} \lambda + \mathcal{C} \alpha = y \\ \mathcal{C}^t \lambda = 0 \end{cases}$$

II-B-4 Définition et caractérisation de l'inf-convolution spline d'interpolation

Soit X un sous-espace semi-hilbertien de E muni d'une semi-norme $|\cdot|$ et V un sous-espace vectoriel de E de dimension m n'étant pas forcément inclus dans X .
Considérons le problème d'interpolation suivant :

$$\begin{aligned} & \min_{x \in X, v \in V} |x|^2 \\ & \langle x+v, \ell_i \rangle = y_i \\ & i = 1, \dots, n \end{aligned}$$

où $\ell_i \in E'$ et $y_i \in \mathbb{R}$ $i=1, \dots, n$.

Sous certaines hypothèses qui seront précisées ci-dessous, nous verrons que ce problème admet une solution unique appelée inf-convolution spline d'interpolation.
Auparavant, justifions le terme d'inf-convolution spline .

Pour cela, considérons l'espace $X^* = X+V = \{x^* \in E / x^* = x+v \text{ où } x \in X, v \in V\}$.
Alors le problème de minimisation devient :

$$\begin{aligned} & \min_{x^* \in X^*} \left(\min_{x \in X, v \in V} |x|^2 \right) \\ & \langle x^*, \ell_i \rangle = y_i \quad x + v = x^* \\ & 1 \leq i \leq n \end{aligned}$$

Pour $x \in X$, posons $\psi(x) = \chi_V(x)$ où χ_V désigne la fonctionnelle indicatrice de V et

$$\phi(x) = \begin{cases} |x|^2 & \text{si } x \in X \\ +\infty & \text{sinon} \end{cases}$$

II- 8

Avec ces notations on peut écrire le problème sous la forme :

$$\begin{array}{ll} \min & (\min (\phi(x) + \psi(x)) \\ x^* \in E & x \in X, v \in V \\ \langle x^*, \ell_i \rangle = y_i & x + v = x^* \\ i = 1, \dots, n & \end{array}$$

Dans l'intérieur de la parenthèse, on voit apparaître une fonction de x^* qui n'est autre que l'opération d'inf-convolution des fonctions ϕ et de ψ ([Laurent 72]):

$$\begin{aligned} \phi \nabla \psi (x^*) &= \inf_{\substack{x \in X, v \in V \\ x + v = x^*}} (\phi(x) + \psi(v)) = \inf_{x \in X} (\phi(x) + \psi(x^* - x)) \end{aligned}$$

Le problème s'écrit donc:

$$\begin{array}{l} \min \quad \phi \nabla \psi (x^*) \\ x^* \in E \\ \langle x^*, \ell_i \rangle = z_i \\ i = 1, \dots, n \end{array}$$

On a alors les résultats suivants :

- Le problème précédent a au moins une solution σ^* .
- L'ensemble des solutions est alors donné par $\sigma^* + N \cap V$.
- $x^* \in X \cap (x - V)$ et solution si et seulement si $\langle x^*, y \rangle = 0$ pour tout $y \in V \cap X$.
- X^* est un sous-espace semi-hilbertien de E d'espace nul $N^* = N + V$.
- Si H est un semi-noyau de X , alors H est un semi-noyau de X^* .
- On a la caractérisation de la solution donnée par le théorème suivant

II-9

Théorème 2:

Si $(N+V) \cap \{x \in E; \langle x, \ell_i \rangle = 0 \ i=1, \dots, n\} = \{0\}$ alors il existe une solution unique σ^* dont l'expression est :

$$\sigma^* = \sum_{i=1}^n \lambda_i H(\ell_i) + p + v \quad \text{où } p \in N, v \in V$$

où $\{\lambda_i\}_{i=1, \dots, n}$, p et v vérifient les conditions suivantes :

$$\begin{cases} \sum_{i=1}^n \lambda_i \langle \omega, \ell_i \rangle = 0 & \forall \omega \in N \\ \sum_{i=1}^n \lambda_i \langle v, \ell_i \rangle = 0 & \forall v \in V \\ \langle \sigma^*, \ell_i \rangle = y_i & i=1, \dots, n \end{cases}$$

Exemple: Soit $X = H^2(\mathbb{R}^2) \subset E = A(\mathbb{R}^2)$ et choisissons pour l'instant $L = E'$. Plaçons nous dans le cas de l'interpolation ponctuelle c'est-à-dire les ℓ_i sont définies par

$$\langle x, \ell_i \rangle = \langle x, \delta_{t_i} \rangle = x(t_i) \quad \text{où } t_i = (x_i, y_i) \in \mathbb{R}^2 \quad i = 1, \dots, n$$

La connaissance de la solution σ^* en un point t , c'est à dire $\langle \sigma^*, \delta_t \rangle$ passe par celle de $\langle H(\delta_{t_i}), \delta_t \rangle$. On pose alors $K(t, t') = \langle H(\delta_{t_i}), \delta_{t'} \rangle$ appelé noyau semi-reproduisant qui est une fonction de $C(\mathbb{R}^2 \times \mathbb{R}^2)$.

Remarque: Comme le semi-noyau n'est pas unique, il en est de même du noyau semi-reproduisant de X . De plus, K n'est pas forcément symétrique et positif, contrairement à la notion classique de noyau reproduisant d'un espace hilbertien. Plus précisément, on a:

si $\ell = \sum_{i=1}^n \lambda_i \delta_{t_i} \in L \cap N^0$ et $m = \sum_{j=1}^m \mu_j \delta_{u_j} \in L \cap N^0$ où L est un sous-espace de E' alors

II- 10

- $\sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j K(t_i, u_j) = \sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j K(u_j, t_i)$ ("semi-symétrie")
- $\sum_{i=1}^n \sum_{j=1}^m \lambda_i \mu_j K(t_i, u_j) \geq 0$ ("semi-positivité")

Dans notre exemple, la solution s'écrit alors:

$$\sigma^*(t) = \sum_{i=1}^n \lambda_i K(t, t_i) + \sum_{j=1}^3 \alpha_j \omega_j(t) + \sum_{k=1}^m \beta_k v_k(t)$$

où $\omega_1(t) = 1$, $\omega_2(t) = x$, $\omega_3(t) = y$ est une base de N et $\{v_k\}_{k=1, \dots, m}$ est une base de V ;
les coefficients λ_i , α_j , β_k vérifiant :

$$\left\{ \begin{array}{ll} \sum_{i=1}^n \lambda_i K(t_h, t_i) + \sum_{j=1}^3 \alpha_j \omega_j(t_h) + \sum_{k=1}^m \beta_k v_k(t_h) = y_h & h = 1, \dots, n \\ \sum_{i=1}^n \lambda_i \omega_j(t_i) = 0 & j = 1, \dots, 3 \\ \sum_{i=1}^n \lambda_i v_k(t_i) = 0 & k = 1, \dots, m \end{array} \right.$$

Si l'on note:

- $\mathcal{K}$ la matrice $n \times n$ de terme général $K(t_h, t_i)$ $i, j = 1, \dots, n$
- $\mathcal{C}$ la matrice $n \times 3$ de terme général $\omega_j(t_h)$ $j = 1, \dots, 3; h = 1, \dots, n$
- $\mathcal{V}$ la matrice $n \times m$ de terme général $v_k(t_h)$ $k = 1, \dots, m; h = 1, \dots, n$
- λ , y les vecteurs de taille n de composantes λ_i, y_i $i = 1, \dots, n$
- α le vecteur de taille 3 de composantes α_i , $i = 1, \dots, 3$
- β le vecteur de taille m de composantes β_i , $i = 1, \dots, m$

on peut écrire le système linéaire précédent sous la forme matricielle ci-dessous:

II-11

$$\begin{cases} K\lambda + \tau\alpha + \nu\beta = y \\ \tau'\lambda = 0 \\ \nu'\beta = 0 \end{cases}$$

J. Duchon [Duchon80] a donné une expression analytique du noyau semi-reproduisant de X qui s'écrit dans cet exemple:

$$K(t, t') = C \|t - t'\|^2 \text{Log} \|t - t'\|^2$$

où $\| \cdot \|$ désigne la norme euclidienne dans $\mathbb{R}^2$ et $C = 4\pi$.

II-B-5 Définition et caractérisation de l'inf-convolution spline d'ajustement

Soit X un sous-espace semi-hilbertien de E muni d'une semi-norme $|\cdot|$ et V un sous-espace vectoriel de E de dimension m (qui n'est pas forcément inclus dans X). Si les données sont de la forme $z_i = x(t_i) + \varepsilon_i$, $i=1, \dots, n$, où les ε_i sont des erreurs sur les données, on considère le problème de régularisation suivant:

$$\min_{x \in X, v \in V} |x|^2 + \rho \sum_{i=1}^n (\langle x+v, l_i \rangle - z_i)^2$$

où $l_i \in E'$ et $z_i \in \mathbb{R}$ $i=1, \dots, n$ et ρ un coefficient positif.

Le réel ρ , appelé paramètre d'ajustement, contrôle le compromis entre l'approximation des données mesurée par la quantité $\sum_{i \in \{1, \dots, n\}} (\langle x+v, l_i \rangle - z_i)^2$ et le critère de lissage mesuré

ici par $|x|^2$.

On démontre, sous la même hypothèse du théorème 2, qu'il existe une solution unique σ^* appelée inf-convolution spline d'ajustement dont l'expression est :

$$\sigma^* = \sum_{i=1}^n \lambda_i H(l_i) + p + v \quad \text{où } p \in N, v \in V$$

II- 12

où les $\{\lambda_i\}_{i=1,\dots,n}$, p et v vérifient les conditions suivantes:

$$\left\{ \begin{array}{l} \sum_{i=1}^n \lambda_i \langle \omega, \ell_i \rangle = 0 \quad \forall \omega \in N \\ \sum_{i=1}^n \lambda_i \langle v, \ell_i \rangle = 0 \quad \forall v \in V \\ \rho(z_i - \langle \sigma^*, \ell_i \rangle) = \lambda_i \quad i = 1, \dots, n \end{array} \right.$$

Exemple: Reprenons l'exemple précédent $X = H^2(\mathbb{R}^2) \subset E = A(\mathbb{R}^2)$ et choisissons $L = E'$. Les ℓ_i sont prises comme fonctionnelles d'interpolation ponctuelle. On a alors:

$$\sigma^* = \sum_{i=1}^n \lambda_i K(t, t_i) + \sum_{j=1}^3 \alpha_j \omega_j(t) + \sum_{k=1}^m \beta_k v_k(t)$$

où $\omega_1(t) = 1$, $\omega_2(t) = x$, $\omega_3(t) = y$ est une base de N et $\{v_k\}_{k=1,\dots,m}$ est une base de V ; les coefficients λ_i , α_j , β_k vérifient :

$$\left\{ \begin{array}{l} \frac{\lambda_h}{\rho'} + \sum_{i=1}^n \lambda_i K(t_h, t_i) + \sum_{j=1}^3 \alpha_j \omega_j(t_h) + \sum_{k=1}^m \beta_k v_k(t_h) = y_h \quad h = 1, \dots, n \\ \sum_{i=1}^n \lambda_i \omega_j(t_i) = 0 \quad j = 1, \dots, 3 \\ \sum_{i=1}^n \lambda_i v_k(t_i) = 0 \quad k = 1, \dots, m \end{array} \right.$$

Ce système écrit sous forme matricielle est analogue à celui obtenu pour l'inf-convolution spline d'interpolation: il suffit d'y remplacer la matrice du noyau $\mathcal{K}$ par $\mathcal{K}' = \mathcal{K} + 1/\rho' \cdot \mathbf{J}_n$ où $\mathbf{J}_n$ désigne la matrice unité d'ordre n .

II- 13

II-C Utilisation de l'inf-convolution spline pour l'approximation de surfaces singulières.

Après avoir rappelé le contexte du problème proposé, nous utilisons les résultats du paragraphe précédent dans le but de proposer une solution qui peut être évaluée numériquement.

II-C-1 Contexte

On cherche une fonction de deux variables $z=f(x,y)$ "lisse" (ce mot sera précisé plus bas) qui est supposée :

- soit prendre des valeurs exactes z_i en des points $t_i = (x_i, y_i)$ $i=1, \dots, n$
- soit prendre des valeurs proches des z_i entachées d'erreurs ϵ_i en des points $t_i = (x_i, y_i)$ $i=1, \dots, n$.

De plus, on suppose qu'il existe p courbes polygonales dérivables par morceaux F_i de longueur finie, ouvertes telles que f ou/et sa dérivée normale admet une discontinuité de première espèce (ou de saut fini), c'est à dire plus précisément:

Soient

$$\begin{cases} x = u_i(t) \\ y = v_i(t) \end{cases} \quad 0 \leq t \leq 1 \quad i = 1, \dots, p$$

les équations paramétriques des courbes polygonales F_i ; on suppose en outre u_i, v_i injectives, continues sur $[0, 1]$ et vérifiant $(u_i(0), v_i(0)) \neq (u_i(1), v_i(1))$.

La discontinuité de la surface le long de ces courbes s'exprime par le fait que les valeurs limites de f de chaque côté des F_i lorsqu'un point (x,y) s'en rapproche indéfiniment, sont distinctes et finies; de même pour la discontinuité de la dérivée normale lorsque celle-ci existe.

II- 14

II-C-2 Modélisation.

Posons $\mathcal{F} = \bigcup_{i=1}^p F_i$

Soit $E = A(\mathbb{R}^2)$ l'espace des fonctions réelles f de deux variables définies sur $\mathbb{R}^2$.

On décompose la fonction f sous la forme: $f = \sigma + s$. L'idée essentielle est ici de reporter le critère de lissage sur la partie régulière σ de f , c'est-à-dire que $\sigma \in X = H^2(\mathbb{R}^2)$.

La partie singulière de f appartient à un sous-espace V représentant les discontinuités, de dimension finie (celle-ci dépendant de la géométrie de $\mathcal{F}$) dont on explicite une base dans le sous-paragraphe ci-dessous.

II-C-3 Choix du sous-espace V des discontinuités.

Ce choix se révèle ici très délicat, du fait que ce type de discontinuité se révèle nettement plus riche que son analogue en dimension un.

En effet, lorsque une fonction d'une variable admet une seule discontinuité de sa dérivée en un point α , Laurent ([Laurent 86a]) propose le sous-espace $V = \{\lambda (t-\alpha)_+, \lambda \in \mathbb{R}\}$ où:

$$p(t) = (t - \alpha)_+ = \begin{cases} t - \alpha & \text{si } t \geq \alpha \\ 0 & \text{sinon} \end{cases}$$

Cet espace est suffisamment riche en général pour représenter la discontinuité de la dérivée en α et donne d'excellents résultats tant pour l'interpolation que pour l'ajustement des données ([Laurent 86b]). De plus, "l'énergie" de la fonction de base est nulle, ce qui implique qu'il n'y a pas d'interférence entre le critère de régularisation que l'on minimise et le sous-espace V traduisant les discontinuités.

En dimension deux, pour une courbe polygonale F_i , il est impossible de construire des fonctions qui soient de degré total inférieur ou égal à un "de chaque côté" de F_i admettant une discontinuité de première espèce le long de cette courbe.

Toutefois, nous proposons ici une base de fonctions continûment différentiables sur $\mathbb{R}^2 \setminus \mathcal{F}$ qui en pratique, fournit de bons résultats numériques.

Considérons une courbe polygonale F_i repérée par ses sommets P_j^i , où j varie entre 1 et l_i , l_i désignant le nombre de points caractérisant F_i .

Dans la suite de ce sous-paragraphe nous omettrons l'indice i pour ne pas alourdir inutilement les notations.

II- 15

Nous supposons qu'il existe un repère orthogonal $\mathcal{R} = (P_1, \vec{S}, \vec{T})$ tel que le graphe de F par rapport à ce repère soit définie par une fonction continue. Cette hypothèse revient à dire que le vecteur $\vec{S}$ vérifie les conditions suivantes:

$$\vec{S} \cdot \overrightarrow{P_j P_{j+1}} > 0 \quad j = 1, \dots, l$$

Soient $(s_i, t_i)_{i=1, \dots, l}$ les coordonnées des points P_i , $t = a_i s + b_i$ les equations des droites $(P_i P_{i+1})_{i=1, \dots, l-1}$ relativement au repère $\mathcal{R}$.

On remarque que la suite $(s_i)_{i=1, \dots, l}$ est strictement monotone.

Posons:

$$\text{sgn}(t) = \begin{cases} 1 & \text{si } t \geq 0 \\ 0 & \text{si } t = 0 \\ -1 & \text{si } t < 0 \end{cases} \quad \mathbb{1}_I(s) = \begin{cases} 1 & \text{si } s \in I \\ 0 & \text{sinon} \end{cases} \quad \text{où } I \text{ est un intervalle de } \mathbb{R}$$

$$P_3^0(s) = s^2 \cdot (3 - 2s) \cdot \mathbb{1}_{[0,1]}(s) \quad \text{et} \quad P_3^1(s) = (1 - s^2) \cdot (3 - 2s) \cdot \mathbb{1}_{[0,1]}(s)$$

Pour $k = 1, \dots, l-2$, on pose:

$$\begin{cases} v_k^f(s, t) = \sum_{h=0}^1 P_3^h(\alpha_{k+h}(s)) \cdot \text{sgn}(f_{k+h}(s, t)) \\ v_k^d(s, t) = \sum_{h=0}^1 P_3^h(\alpha_{k+h}(s)) \cdot |f_{k+h}(s, t)| \end{cases}$$

où $\alpha_k(s)$ et $f_k(s, t)$ sont définis par :

$$\alpha_k(s) = \frac{s - s_k}{s_{k+1} - s_k} \quad \text{et} \quad f_k(s, t) = t - a_k \cdot s - b_k$$

II- 16

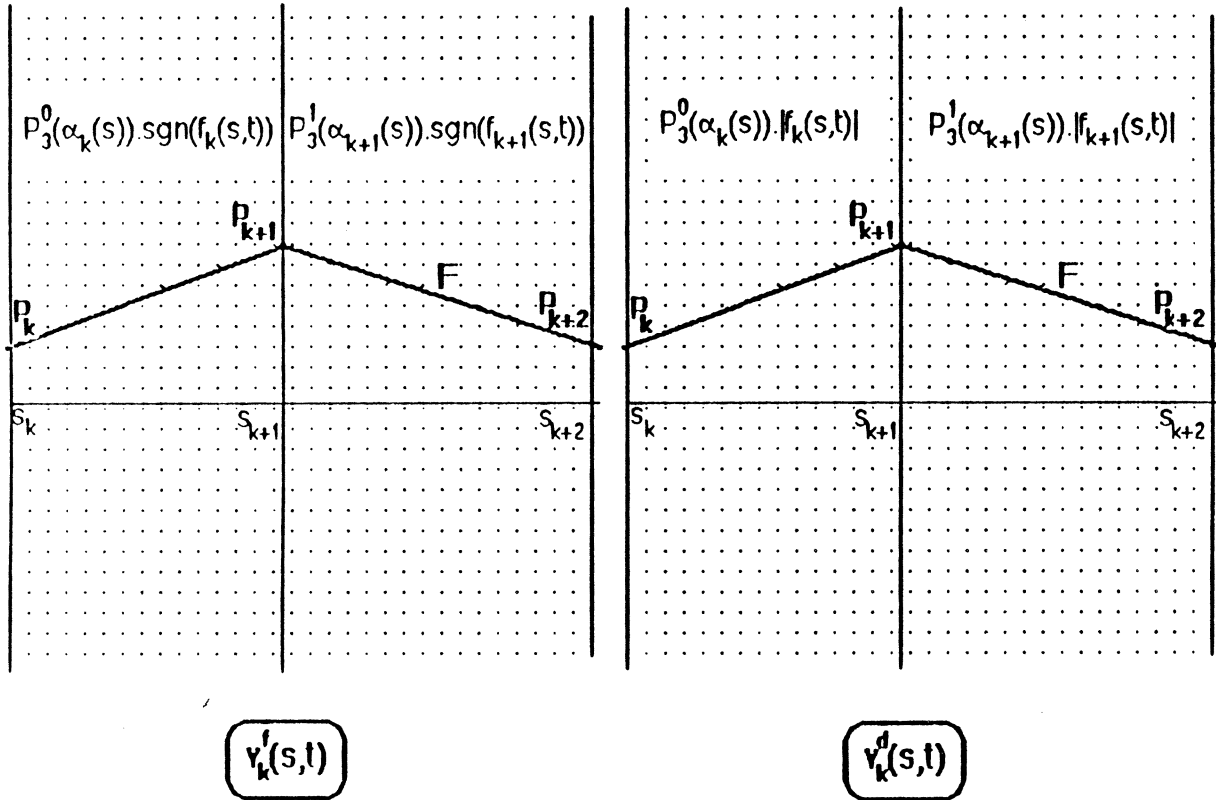


Figure II-1

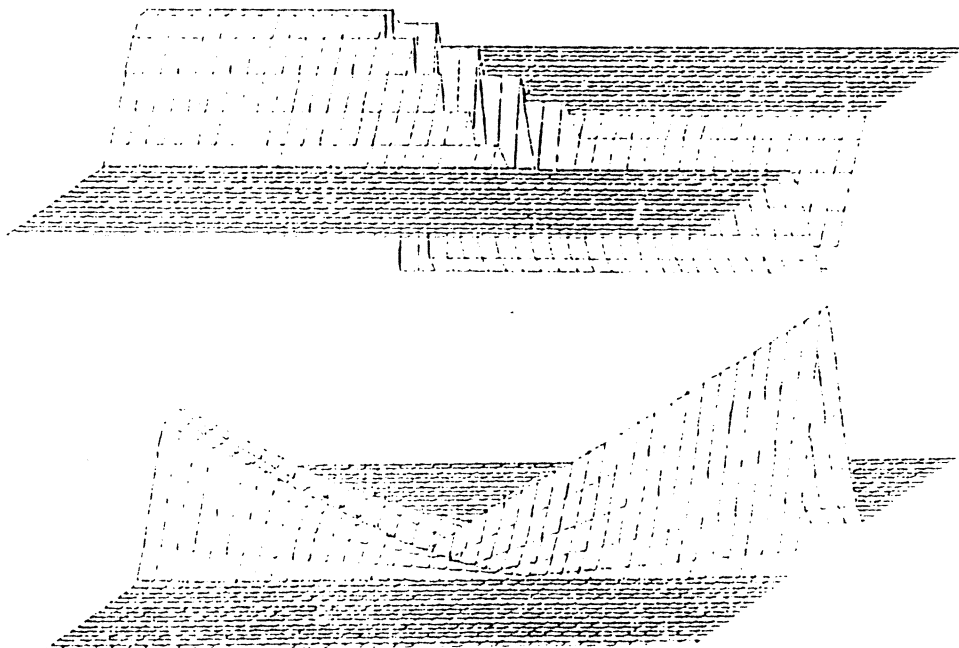


Figure II-2

II- 17

Propriétés:

- v_k^I est continue sur $\mathbb{R}^2 \setminus ([P_k P_{k+1}] \cup [P_{k+1} P_{k+2}])$
- v_k^d a une dérivée normale discontinue sur $([P_k P_{k+1}] \cup [P_{k+1} P_{k+2}])$
- $\text{supp}(v_k^I) = \text{supp}(v_k^d) = [s_k, s_{k+2}] \times \mathbb{R}$.

Nous donnons sur la figure II-1 la représentation des supports des fonctions v_k^I, v_k^d ainsi que leurs valeurs; on donne une représentation tridimensionnelle d'un exemple de ces fonctions sur la figure II-2, avec $P_1 = (6.5, 3.5)$, $P_2 = (3.5, 5)$, $P_3 = (6.5, 6.5)$ et $\vec{S} = (0, 1)$.

Donc, pour une seule courbe polygonale, la dimension du sous-espace V est:

$$m = 2 (l - 2) = 2 \cdot 1 - 4$$

Dans le cas général, si nous avons p courbes polygonales F_i , chacune étant caractérisée par des points $P_{i,j}$ ($j = 1, \dots, l_i$; $i = 1, \dots, p$) alors on considère l'espace V engendré par les fonctions:

$$v_{1,1}^I, v_{1,1}^d, \dots, v_{1,l_1-2}^I, v_{1,l_1-2}^d, \dots, v_{i,1}^I, v_{i,1}^d, \dots, v_{i,l_i-2}^I, v_{i,l_i-2}^d, \dots, v_{p,1}^I, v_{p,1}^d, \dots, v_{p,l_p-2}^I, v_{p,l_p-2}^d$$

où le premier indice indique le numéro de la courbe F_i , et le deuxième indice indique le numéro du point de la courbe F_i .

La dimension du sous-espace V vaut par conséquent:

$$m = 2 \cdot \left(\sum_{i=1}^p l_i - 2 \cdot p \right)$$

II-C-4 Algorithme de calcul de l'inf-convolution-spline d'interpolation ou d'ajustement

Nous évoquons ici deux problèmes liés au calcul de la solution:

- le choix des repères liés aux courbes F_i définissant les fonctions de base de V .
- la stratégie à adopter lorsque le nombre de points de données est grand.

II-C-4-1 Choix des repères

En pratique, le nombre de points caractérisant une courbe est petit (inférieur à 10). Dans ce cas, on calcule une droite de régression linéaire (en abscisse ou en ordonnée) basée sur ces points. Ceci nous fournit d'une part, le vecteur $\vec{S}$ pris comme vecteur directeur de cette droite et d'autre part les abscisses s_i des points $P_{i,j}$ qui sont rangés dans l'ordre croissant ou décroissant.

II - 18

II-C-4-2 Stratégie de calcul de l'inf-convolution spline selon le nombre de points de données

On a vu au paragraphe précédent que le calcul de l'inf-convolution spline d'interpolation ou d'ajustement assujettie à n points de données nécessite la résolution d'un système linéaire de taille $n+m+3$ où m désigne la dimension de l'espace V qui dépend de la géométrie des courbes polygonales F_i .

Lorsque ce nombre de points est petit (en pratique inférieur à 130) , on résoud ce système de manière directe, par exemple en utilisant la méthode de Gauss avec stratégie de pivot partiel (ou total) ou la méthode d'Householder.

Ce calcul devient impossible lorsque le nombre de points de données est grand, ceci pour deux raisons:

- place mémoire dépendant de la machine utilisée.
- instabilité du calcul due à la précision machine qui est limitée.

Pour pallier à cette difficulté, nous reprenons ici une technique de reconstruction de l'approximant due à Paihua et Utreras ([Paihua 78b]) pour le calcul d'une fonction-spline "plaque mince". Les modifications essentielles apportées concernent uniquement la prise en compte des courbes polygonales F_i .

On découpe la région d'observation en plusieurs sous-régions contenant un nombre raisonnable de points et ensuite, on reconstruit l'approximant en "recollant" les morceaux à l'aide d'une partition convexe de l'unité. Plus précisément on fait les hypothèses suivantes sur le nuage des points de données t_i :

- il existe un rectangle R le contenant.
- il est "uniforme" .

Soit $R = [a,b] \times [c,d]$ et considérons un recouvrement de R par les sous-rectangles $R_{i,j}$ définis par

$$R_{i,j} = [a_i, b_i] \times [c_j, d_j] \quad i = 1, \dots, n_x ; j = 1, \dots, n_y$$

où a_i, b_i, c_j, d_j satisfont aux conditions suivantes:

$$a = a_1 < a_2 < b_1 < a_3 < b_2 < \dots < b_{n_x} = b$$

$$c = c_1 < c_2 < d_1 < c_3 < d_2 < \dots < d_{n_y} = d$$

II - 19

De plus, considérons les ensembles suivants:

$$Q_{i,j} = \{ t_k ; k = 1, \dots, n / t_k \in R_{i,j} \} \neq \emptyset$$

$$C_{i,j} = \{ [P_{k,l}, P_{k,l+1}] k = 1, \dots, p; l = 1, \dots, l_p / [P_{k,l}, P_{k,l+1}] \cap R_{i,j} \neq \emptyset \}$$

Nous avons donc les propriétés suivantes:

$$\bullet R = \bigcup_{i=1}^{n_x} \bigcup_{j=1}^{n_y} R_{i,j}$$

$$\bullet R_{i,j} \cap R_{i,j+1} = [a_i, b_i] \times [c_{j+1}, d_j] \quad i = 1, \dots, n_x \quad ; j = 1, \dots, n_y - 1$$

$$\bullet R_{i,j} \cap R_{i+1,j} = [a_{i+1}, b_i] [c_j, d_j] \quad i = 1, \dots, n_x - 1 ; j = 1, \dots, n_y$$

Nous représentons un exemple de recouvrement sur la figure ci-dessous.

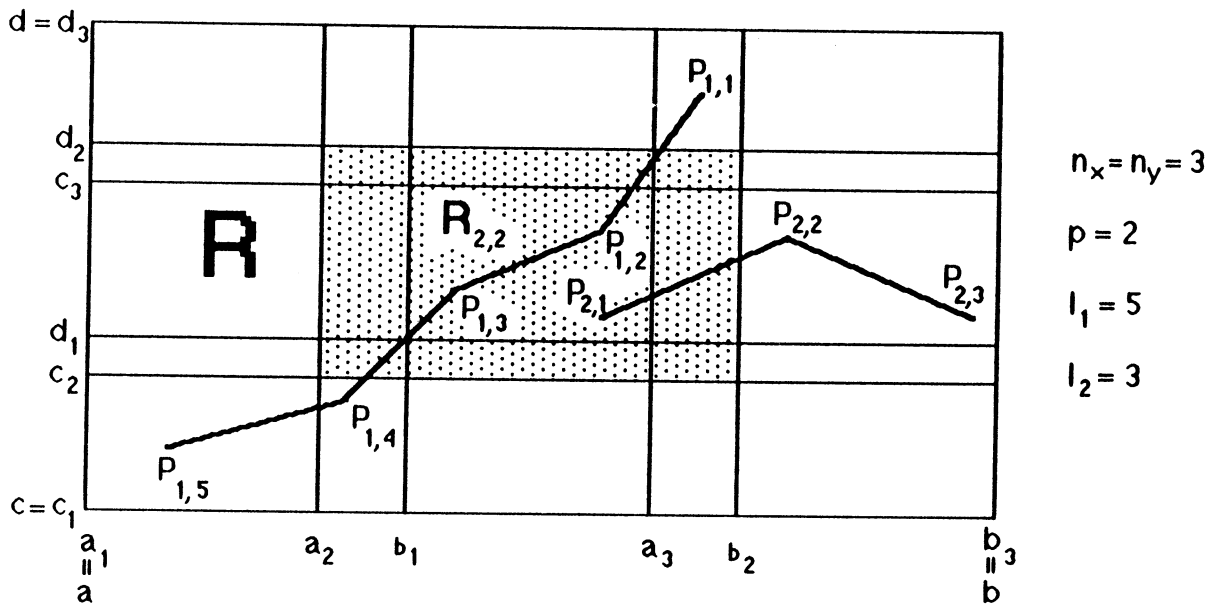


Figure II - 3

II - 20

Soit $\sigma_{i,j}$ l'inf-convolution spline telle que:

$$\left\{ \begin{array}{l} \sigma_{i,j} \text{ interpole ou ajuste les valeurs } z_k \text{ aux points } t_k \quad \forall t_k \in R_{i,j} \\ \sigma_{i,j} \text{ présente des discontinuités de valeurs ou de dérivée normale le long des} \\ \text{segments } [P_{k,l}, P_{k,l+1}] \quad \forall [P_{k,l}, P_{k,l+1}] \in C_{i,j} \end{array} \right.$$

Cette spline est appelée inf-convolution spline "locale " au rectangle $R_{i,j}$. Elle est construite avec

- les points $\{ t_k \}$ se trouvant dans le sous-rectangle $R_{i,j}$.
- les fonctions de base $v_{k,l}^f, v_{k,l}^d$ qui sont discontinues ou à dérivée discontinue le long des segments $[P_{h,m}, P_{h,m+1}]$ appartenant à $C_{i,j}$.

En ce qui concerne ce dernier point, pour un segment $[P_{h,m}, P_{h,m+1}]$ il faut prendre en compte les fonctions $v_{h,m-1}^f, v_{h,m}^f, v_{h,m-1}^d, v_{h,m}^d$ sauf si:

- $m = 1$ auquel cas nous prenons en compte les fonctions $v_{h,2}^f, v_{h,2}^d$
- $m = l_h$ auquel cas nous prenons en compte les fonctions $v_{h,l_h-1}^f, v_{h,l_h-1}^d$

Pour illustrer nos propos, si nous considérons l'exemple de la figure II-3, nous prenons ici en compte les fonctions $v_{1,1}^f, v_{1,2}^f, v_{1,3}^f, v_{2,1}^f, v_{1,1}^d, v_{1,2}^d, v_{1,3}^d, v_{2,1}^d$.

Remarque: que si un sous-rectangle $R_{i,j}$ n'est traversé par aucune courbe polygonale F_i , alors la solution est une fonction-spline plaque mince basée sur les points appartenant à $P_{i,j}$.

Les inf-convolution splines étant supposées connues (i.e. par leurs coefficients), il s'agit de "recoller" les morceaux, c'est à dire plus précisément, introduisons les fonctions suivantes:

$$\alpha(x) = (x-1)^2 \cdot (2x+1) \quad \beta(x) = 1 - \alpha(x) \quad \text{pour } x \in [0,1]$$

$$\varphi_i(x) = \frac{x - a_{i+1}}{b_i - a_{i+1}}$$

$$\phi_j(y) = \frac{y - c_{j+1}}{d_j - c_{j+1}}$$

II - 21

et

$$\alpha_i^1(x) = \alpha(\varphi_i(x)) \quad \alpha_j^2(y) = \alpha(\phi_j(y)) \quad \beta_i^1(x) = \beta(\varphi_i(x)) \quad \beta_j^2(y) = \beta(\phi_j(y))$$

Remarque: Les fonctions α et β sont dérivables sur $[0,1]$ et de plus vérifient:

$$\alpha(0) = \alpha'(0) = \beta(1) = \beta'(1) = 0$$

Les fonctions $\alpha_i^1, \alpha_j^2, \beta_i^1, \beta_j^2$ nous permettent alors d'obtenir un approximant global σ de toutes les données qui est obtenu en recollant toutes les inf-convolutions splines "locales". Plus précisément σ est défini par les relations:

$$\sigma(x,y) = \begin{cases} \sigma_{i,j}(x,y) & (x,y) \in R_{i,j} \setminus (R_{i,j-1} \cup R_{i,j+1} \cup R_{i+1,j} \cup R_{i-1,j}) \\ \alpha_i^1(x) \cdot \sigma_{i,j}(x,y) + \beta_j^1(x) \cdot \sigma_{i+1,j}(x,y) & (x,y) \in R_{i,j} \cap R_{i+1,j} \setminus (R_{i,j+1} \cup R_{i,j-1}) \\ \alpha_j^2(y) \cdot \sigma_{i,j}(x,y) + \beta_j^2(y) \cdot \sigma_{i,j+1}(x,y) & (x,y) \in R_{i,j} \cap R_{i,j+1} \setminus (R_{i+1,j} \cup R_{i-1,j}) \\ \alpha_i^1(x) \cdot [\alpha_j^2(y) \cdot \sigma_{i,j}(x,y) + \beta_j^2(y) \cdot \sigma_{i,j+1}(x,y)] + \beta_i^1(x) \cdot [\alpha_j^2(y) \cdot \sigma_{i+1,j}(x,y) + \beta_j^2(y) \cdot \sigma_{i+1,j+1}(x,y)] & (x,y) \in R_{i,j} \cap R_{i+1,j} \cap R_{i,j+1} \cap R_{i+1,j+1} \end{cases}$$

La remarque ci-dessus nous montre facilement que σ est un approximant global de classe C^1 sur le rectangle R .

II-D Résultats numériques et conclusions

Nous illustrons l'efficacité de cette technique en donnant quelques exemples académiques ainsi qu'un exemple réel; la spline est évaluée grâce à la méthode décrite dans le paragraphe précédent. Tous les résultats présentés ici ont été effectués sur un Imicro-ordinateur compatible.

Exemple 1: Figure II-4

On prend 169 données équiréparties (x_i, y_i, z_i) sur le domaine $[0,10] \times [0,10]$ où les z_i sont les valeurs prises par une fonction-test f (dessin du haut) admettant des discontinuités de valeur et de dérivée le long du segment brisé. Nous avons pris $m=1$ (c'est-à-dire une fonction de base de discontinuité de valeur et de dérivée) et $n_x = n_y = 2$. L'inf-convolution spline d'interpolation σ obtenue (en bas) reproduit fidèlement la fonction test. pour mesurer la qualité de l'interpolation nous évaluons

II - 22

l'erreur sur une grille régulière de la manière suivante: l'erreur est calculée suivant la formule:

$$ERR_{i,j} = | (\sigma_{i,j} - f_{i,j}) / (\sigma_{\max} - \sigma_{\min}) |$$

où (i,j) sont les coordonnées du point dans la grille, $\sigma_{i,j}, f_{i,j}$ sont respectivement la valeur de la spline et la valeur de la fonction-test; $\sigma_{\max}, \sigma_{\min}$ sont les valeurs extrêmes prises par la fonction-spline sur cette grille. Nous consignons les résultats dans le tableau ci-dessous.

Nombre de points	Taille de la grille	Erreur maximale	Erreur moyenne	Déviations standard	Temps CPU (sec)
169	40x 40	0.04	0.008	0.01	321

Remarque: Au voisinage de la faille, il subsiste tout de même quelques oscillations: ceci s'explique par le fait qu'il y a peu de données au voisinage de celle-ci.

Exemple 2: Figure II-5

Nous reprenons le même jeu de données pour une fonction-test admettant une dérivée normale discontinue le long du segment rectiligne $[(5,2.5),(5,7.5)]$. Nous avons pris ici $m = 1$.

La qualité du résultat est moins bonne que dans l'exemple 1, surtout au voisinage de la rupture de pente. Ceci est dû sans doute au fait que m est trop petit, et que les fonctions de base de discontinuité de dérivée apportent quelques perturbations. Nous donnons les résultats sur les erreurs sur le tableau ci-dessous.

Nombre de points	Taille de la grille	Erreur maximale	Erreur moyenne	Déviations standard	Temps CPU (sec)
169	40x 40	0.07	0.02	0.02	302

Exemple 3: Figure II-6

Avec toujours le même jeu de données, nous perturbons les données à l'aide d'un bruit gaussien de variance $s^2=0.05$ (dessin du haut). L'ajustement obtenu avec $p = 2.10^5$ permet de retrouver un résultat visuellement satisfaisant. Le choix du paramètre d'ajustement est manuel, i.e. on calcule la spline d'ajustement jusqu'à que l'on obtienne le résultat désiré. Les remarques évoquées à propos de l'exemple 1 restent ici valables.

Exemple 4: Figure II-7

Relevé d'altitude d'une partie de la Dent de Crolles située dans le massif de la Chartreuse, 196 points dans le domaine $\{0,1.6\} \times \{0,1.6\} u^2$ où $u = 2.5$ km.

II - 23

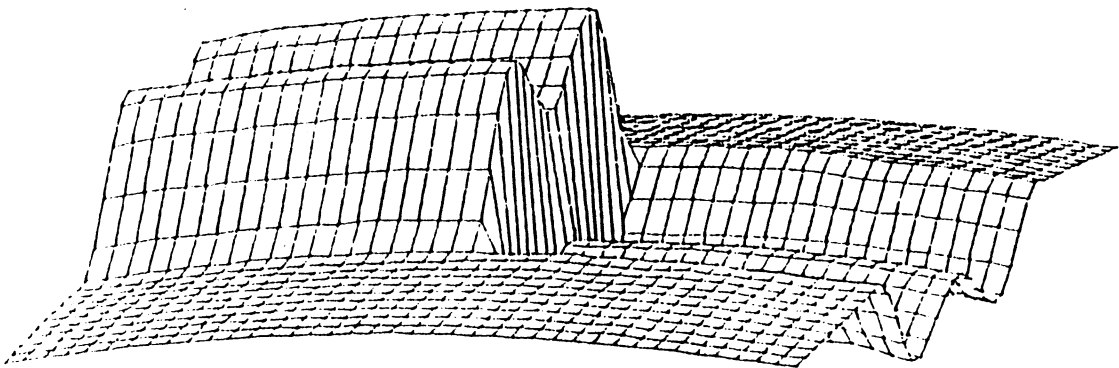
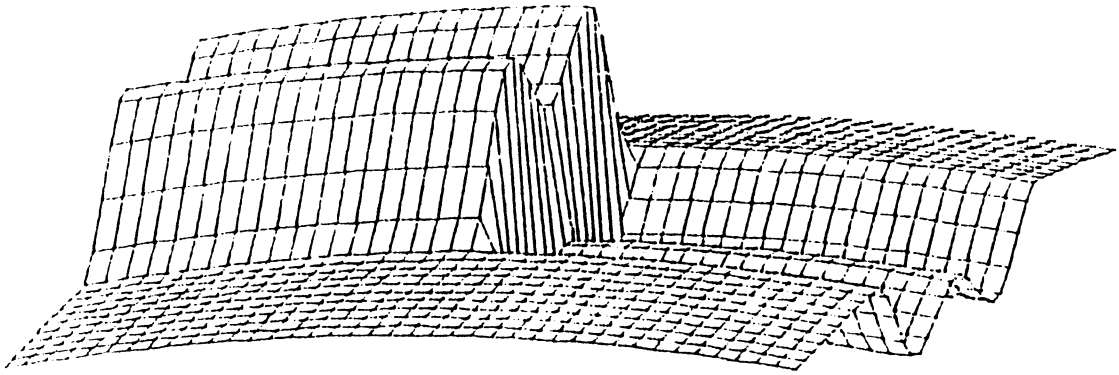


Figure II-4

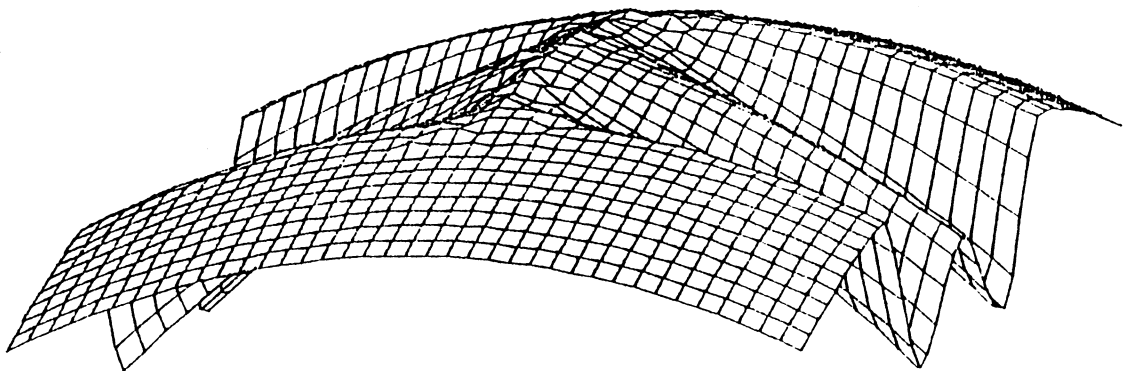
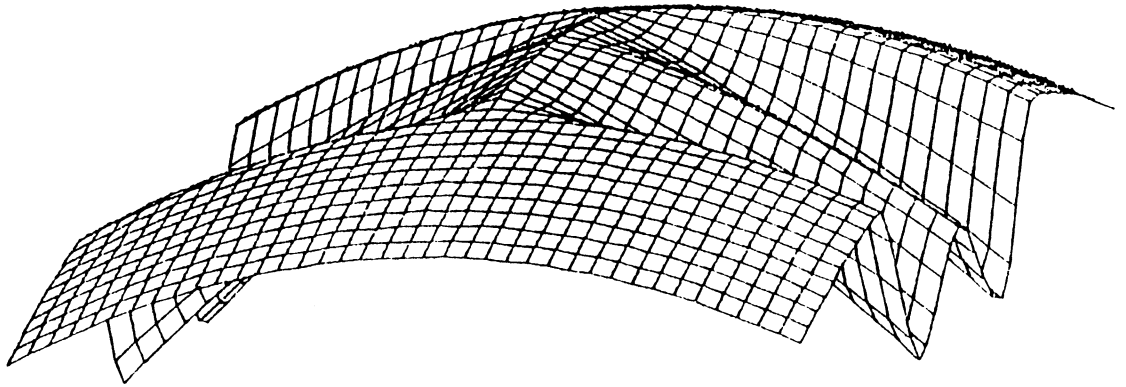


Figure II-5

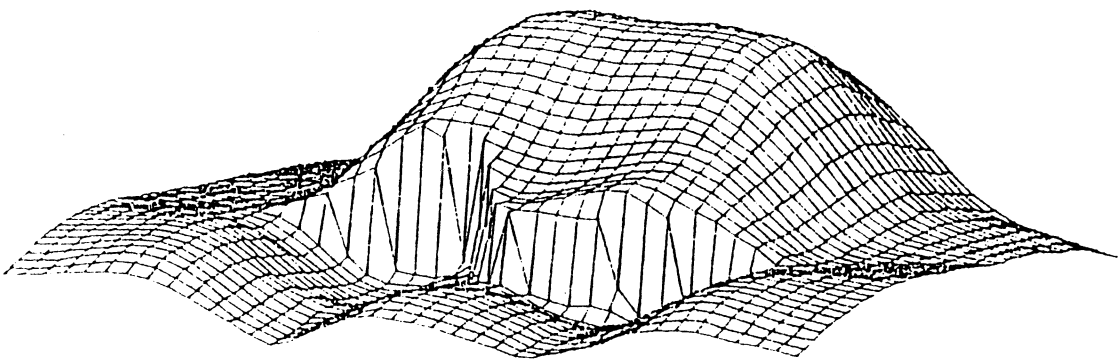
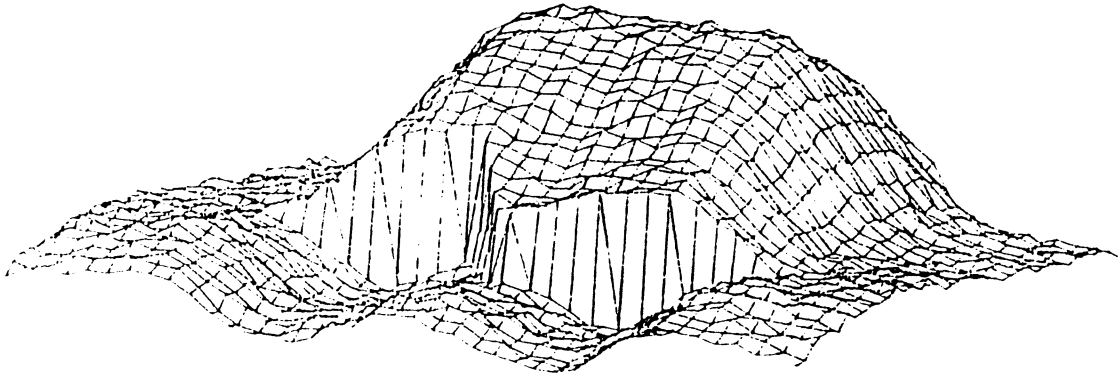
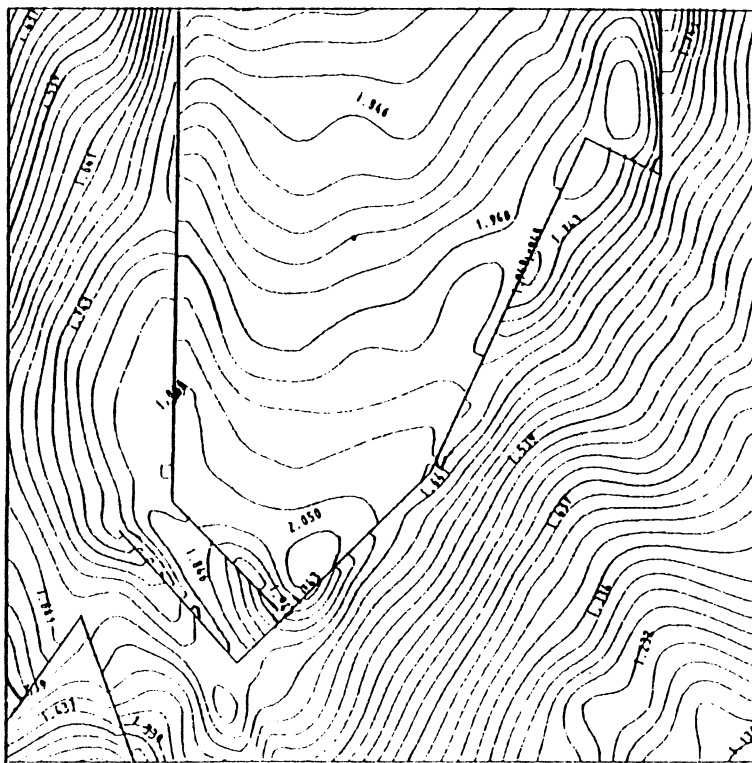
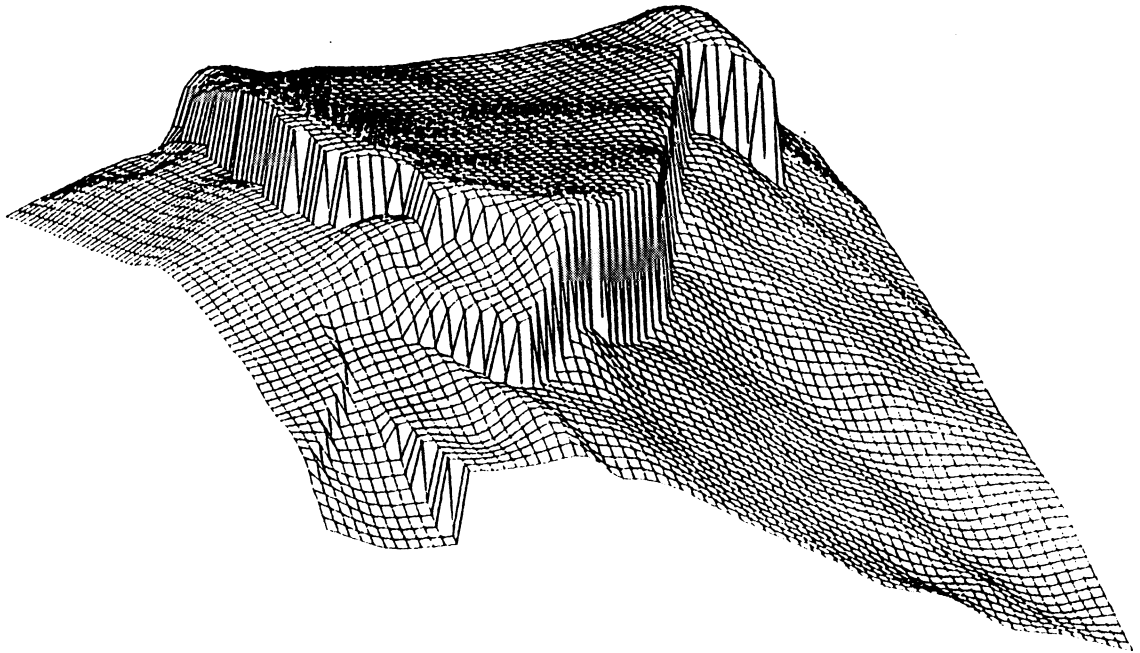


Figure II-6

II - 26



NIVEAU MIN 1.1300E+000

NIVEAU MAX 2.0500E+000

PAS 2.0444E+002

Figure II-7

II - 27

La spline d'ajustement est évaluée aux sommets d'une grille constituée de 69×69 points construite sur le même domaine. Le nombre de découpages est $n_x=n_y=3$ et $p = 10^5$.

Nombre de points	Taille de la grille	Temps CPU (sec)
196	69×69	726

Nous donnons sur la figure du bas une carte d'état major établie à partir des valeurs de la spline, grâce à l'algorithme décrit dans le chapitre III.

Conclusions

- Une limitation pour le calcul est l'encombrement mémoire dû à la matrice du système. Pour cette raison ainsi que pour le temps d'exécution, il est capital que dans le cas d'un grand nombre de donnée, le recouvrement soit bien choisi: il doit être tel que dans chaque sous-rectangle un nombre suffisant de points soit présent (en pratique une cinquantaine) ceci afin de préserver une bonne précision, et que les zones de recouvrement ne soient pas trop "minces". Lorsque le nuage de points est uniforme, on peut recourir toutefois à un procédé de découpage automatique du domaine d'observation.
- Le choix du nombre de fonctions de base est aussi important;il doit satisfaire deux critères a priori se nuisant mutuellement:
 - il doit être suffisamment grand pour approximer le mieux possible la surface aux voisinages des lignes de discontinuité
 - il ne doit pas être trop grand ceci afin de ne pas grossir la taille du système linéaire à résoudre.
- Le temps de calcul des coefficients des splines locales reste faible par rapport à celui nécessaire pour l'assemblage de l'approximant global.
- Des extremas locaux non désirés peuvent apparaître: c'est une conséquence inévitable du choix des splines plaque-mince.
- La technique de recouvrements nous donne des résultats satisfaisants, bien que l'approximant global soit utilisé uniquement pour le calcul des valeurs aux noeuds d'une grille régulière. Pour le tracé des isovaleurs, l'idéal serait de développer un algorithme prenant en compte directement la nature de cet approximant.

CHAPITRE III:
UN ALGORITHME DE TRACE AUTOMATIQUE DE
COURBES ISOVALEURS POUR DES SURFACES
SINGULIERES

III - 1

III-A Introduction

Les représentations d'une surface à l'aide de courbes de niveau sont utilisées depuis plusieurs années pour représenter des données géographiques, biologiques et autres. Par exemple, les géophysiciens ont particulièrement besoin de dresser des cartes d'état-major pour des données de gravité, d'altitude et autres paramètres géologiques et géochimiques.

L'avènement des ordinateurs a conduit à développer différentes techniques de tracé de courbes isovaleurs par programme. Plusieurs programmes qui ont été développés sont capables de tracer des cartes "raisonnables" des points de données.

Une littérature abondante traitant de ce sujet [Bolondi76, Cottafava69, Hessing72, Mallet74, Mclain76, Peltó68, Walters69], montre d'une part qu'un certain nombre de techniques plus ou moins efficaces en temps de calcul et en place mémoire ont été élaborées, et d'autre part, elle met en évidence les difficultés inhérentes au problème. Cependant, peu d'auteurs proposent des méthodes de tracé de courbes de niveau lorsque la surface présente des discontinuités le long de lignes brisées. Nous proposons ici un algorithme qui résout le problème sous certaines conditions restrictives qui sont détaillées dans le paragraphe suivant.

Le problème peut être posé comme suit:

Supposons que l'on se donne une surface explicite différentiable $z = f(x,y)$ connue uniquement aux sommets d'une grille rectangulaire (ou triangulaire isocèle), présentant des discontinuités de saut fini le long de lignes brisées D_k $k=1, \dots, p$. En pratique, nous aurons quelques centaines de points de données situés aux sommets d'un maillage rectangulaire.

La méthode se décompose en deux étapes:

1) Un codage des cellules du maillage est effectué. Il détermine la présence ou non d'une ligne D_k et dans le cas d'une présence, la position des lignes.

2) Pour un niveau donné, on détermine les côtés traversés par celui-ci, puis on calcule les points d'intersection entre le niveau et les côtés traversés et un lien convenable (on utilisera des splines cubiques paramétrées) est tracé entre ces points. La technique de détermination des côtés traversés est composée en premier lieu d'une détection d'une cellule pour laquelle le niveau intersecte celle-ci et ensuite d'un "suivi" de la courbe jusqu'à que celle-ci s'arrête.

Dans le paragraphe suivant, nous développons en détail l'algorithme.

Enfin dans le dernier paragraphe, nous présentons des exemples issus de données réelles et synthétiques et concluons en soulignant les avantages et inconvénients de la méthode.

III - 2

III-B Présentation de l'algorithme

III-B-1 Notations, définitions et hypothèses

Considérons un domaine rectangulaire $D = [a, b] \times [c, d]$ du plan.

Soit une subdivision de $[a, b]$ (respectivement de $[c, d]$) :

$$a = x_1 < x_2 < \dots < x_{n_x} \text{ tel que } x_{i+1} - x_i = p_x \quad i = 1, \dots, n_x - 1$$

$$(\text{respectivement } c = y_1 < y_2 < \dots < y_{n_y} \text{ tel que } y_{j+1} - y_j = p_y \quad j = 1, \dots, n_y - 1)$$

Définitions: 1) Le maillage rectangulaire de D suivant ces subdivisions est:

$$M = \left\{ \bigcup_{i=1}^{n_x} \{x = x_i\} \times [c, d] \right\} \cup \left\{ [a, b] \times \{y = y_j\} \right\}_{j=1}^{n_y}$$

2) Une cellule du maillage est le rectangle $C_k = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$

où $k = (j-1) n_x + i \quad i = 1, \dots, n_x - 1, j = 1, \dots, n_y - 1$.

Elle est donc repérée par son noeud bas gauche (voir figure III-0)

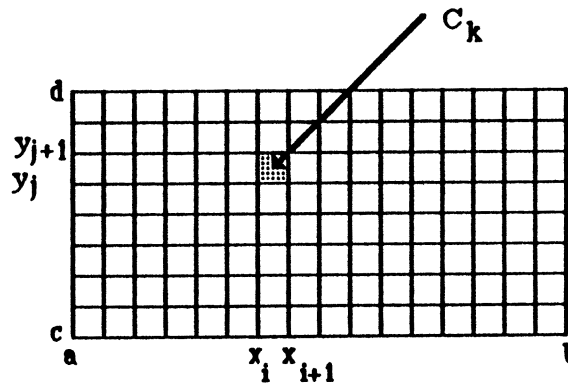


FIGURE III-0

Remarque: Lorsque l'indice k est un multiple de n_x , c'est à dire lorsque le point correspondant se situe sur le bord droit de D, la cellule n'est pas définie.

Soient des lignes brisées D_k ($k=1, \dots, p$) incluses dans le domaine D tel que les noeuds du maillage n'appartiennent pas à celles-ci.

III - 3

Posons: $D' = D \setminus \left(\bigcup_{k=1}^p D_k \right)$.

Soit $z = f(x, y)$ une fonction de 2 variables définie sur D' connue aux noeuds du maillage M ($f_{i,j} = f(x_i, y_j)$ $i = 1, \dots, n_x - 1, j = 1, \dots, n_y - 1$) et éventuellement aux centres des cellules. Notons $z_k = f_{i,j}$ où $k = (j-1) \cdot n_x + i$.

Définitions: 1) Le niveau (z_0) est par définition l'ensemble :

$$(z_0) = \{ (x, y) \in D' \text{ tel que } f(x, y) = z_0 \}$$

2) Une composante de niveau (z_0) est une composante connexe du niveau (z_0). Elle sera représentée par l'image par une application

$C_{z_0} : I = [\alpha, \beta] \rightarrow D'$ continue tel que $f(C_{z_0}(I)) = z_0$. Elle sera fermée si les extrémités coïncident c'est à dire :

$$C_{z_0}(\alpha) = C_{z_0}(\beta).$$

On repèrera les composantes par l'intersection des côtés et du niveau : ces côtés seront stockés dans un tableau de réels **intersection** $[1..n, 1..2]$, où n désigne le nombre de côtés traversés par le niveau. Le calcul de ces intersections est détaillé au paragraphe III-B-4.

On fera en outre les hypothèses suivantes :

H1 : f présente des discontinuités de saut fini le long de lignes brisées D_k .

H2 : La surface $z = f(x, y)$ est différentiable en tout point de D' . Ceci a pour conséquences:

- que les applications C_{z_0} sont différentiables sur leurs domaines de définition.
- que les extrémités des composantes ouvertes sont sur la frontière de D' .
- que les composantes fermées sont à l'intérieur de D' .

H3 : Les applications C_{z_0} sont injectives.

H4: Le maillage M est supposé suffisamment fin pour qu'un côté d'une cellule soit coupé au plus une fois par un niveau (z_0) et que la zone cartographiée ne présente pas de "plateau", c'est à dire qu'il existe au moins deux valeurs correspondant aux sommets d'une cellule qui soient distinctes.

III - 4

Enfin, si les valeurs aux centres de chaque cellule ne sont pas fournies, on les détermine en calculant la moyenne des valeurs aux sommets de la cellule. Si celle-ci est traversée par une ligne D_k , on ne prend en compte dans la moyenne que les sommets situés du même côté de D_k .

III-B-2 Principe de l' algorithme

L' algorithme s'effectue en deux phases successives:

Phase n°1 : Codage des cellules.

Cette phase qui servira ultérieurement, permet de savoir si les cellules de M sont intersectées par une ligne D_k . On explique ceci dans le paragraphe suivant.

Phase n°2 : Tracé d'un niveau .

On effectue un balayage des cellules du maillage de gauche à droite et de bas en haut. Lorsqu'une cellule C_{k_0} ne contenant pas de lignes D_k (cf. phase 1) traversée par une composante de niveau est détectée, on suit une partie de celle-ci en stockant au fur et à mesure les points d'intersection traversés jusqu'à ce que l'une des trois situations suivantes se produise:

- * la composante est fermée
- * la composante s'arrête sur le bord du domaine.
- * la composante s'arrête dans une cellule traversée par une ligne D_k .

On stockera temporairement le premier indice k_0 tel que la composante "quitte" la ligne du maillage contenant la première cellule détectée, c'est à dire:

$$k_d = k_0 - n_x + 1 + \min ((k - k_0) \geq n_x)$$
$$k \geq k_0$$
$$C_k \cap (z_0) \neq \emptyset$$

ceci afin de reprendre le balayage à partir de cet indice pour la détection éventuelle d'autres composantes.

On prendra soin de changer les codes des cellules traversées (par exemple en prenant l'opposé de ceux-ci) afin de savoir ultérieurement si une cellule traversée par le niveau a déjà été rencontrée.

III - 5

Lorsque le suivi d'une composante s'arrête, on repart de la première cellule détectée on suit l'autre branche à partir de la cellule C_{k_0} , jusqu'à que l'une des deux dernières situations citées plus haut se produise.

Les points d'intersection **intersection** $[i, j]$ ($i=1, \dots, n$; $j=1, 2$) des côtés traversés par une composante étant connus nous donne une suite de points de D ; le tracé de la composante est alors effectué grâce à une interpolation cubique paramétrée

A la fin du balayage, on réaffecte aux codes leurs valeurs initiales, ceci éventuellement pour un tracé d'un autre niveau.

III- B-3 Codages des cellules

Nous donnons ici une manière de coder les cellules du maillage traversées ou non par les lignes de discontinuité D_k ; $k=1, \dots, p$.

Les codes correspondants seront stockés dans un tableau unidimensionnel **code** .

Initialement, tous les codes sont affectés à 100, sauf ceux des cellules pour lesquelles $k=0$ modulo n_x auquel cas le code est fixé définitivement à 0.

Puisque chaque ligne D_k est formée de segments connectés , on peut donc décomposer la tâche en codant toutes les cellules traversées par chaque segment.

La cellule C_k est constituée de quatre triangles isocèles (voir figure III-1) :

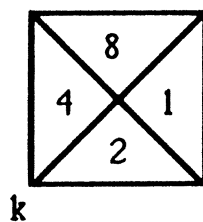


FIGURE III-1

On adoptera le codage suivant:

- Si le segment coupe le triangle supérieur alors $\text{code}[k] = \text{code}[k] \vee 8$
- Si le segment coupe le triangle gauche alors $\text{code}[k] = \text{code}[k] \vee 4$
- Si le segment coupe le triangle inférieur alors $\text{code}[k] = \text{code}[k] \vee 2$
- Si le segment coupe le triangle droit alors $\text{code}[k] = \text{code}[k] \vee 1$

III- 6

où " $\vee$ " désigne l'opération booléenne "ou" agissant bit à bit sur les mots binaires représentant les deux opérandes.

exemple: Evaluation de $6 \vee 12$

$$6 = (00000110)_2 \text{ et } 12 = (00001100)_2 \text{ donc } 6 \vee 12 = (00001110)_2 = 14$$

Ceci correspond par exemple à la situation suivante:

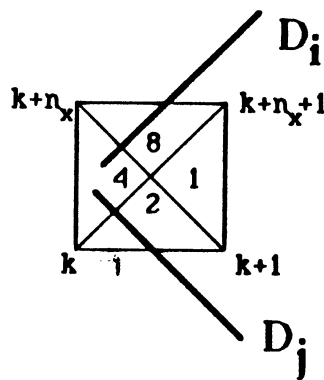


Figure III-2

Ce codage peut s'écrire sous la forme algorithmique suivante:

```

procédure code_cellule ( k: entier ; c: entier );
{ k désigne le numéro de la cellule à coder, c désigne la contribution à rajouter à
code[k] }
variables locales : b, x, i: entier;
si code[k] = 100 alors code[k] = c
    sinon
    début
        z := 0;
        b := 8;
        x := code[k];
        pour i := 1 à 4 faire
            début
                si (x ≥ b) ou (c ≥ b) alors z := z+b;
                si (x ≥ b) alors x := x-b;
                si (c ≥ b) alors c := c-b;
                b := b/2;
            fin ;
        code[k] := z;
    fin ;
fin ;

```

III- 7

On oriente le segment de telle façon que l'origine de celui-ci soit le plus bas dans le cas d' un segment oblique ou vertical, ou le plus à gauche dans le cas d'un segment horizontal.

Ceci étant fait il est facile de voir qu'il y a 4 situations qu'il est facile de distinguer (voir figure III-3):

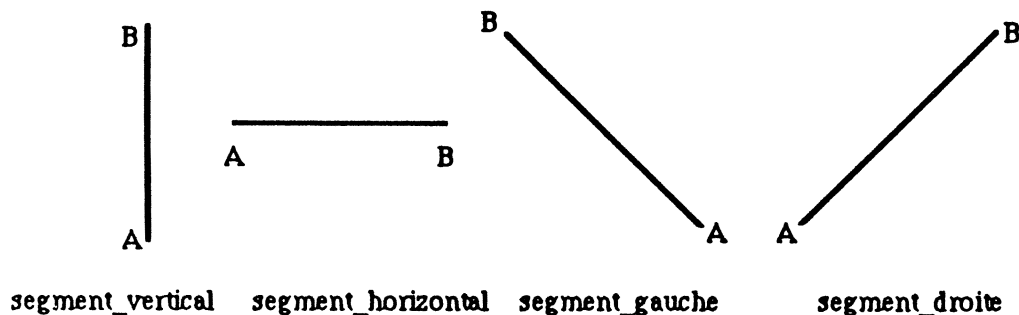


Figure III-3

On calcule ensuite les indices k_a et k_b des cellules contenant respectivement les extrémités A,B du segment.

Pour des raisons d'optimisation du temps de calcul des codages, il est apparu nécessaire de faire un traitement différent pour des segments horizontaux ou verticaux (segment_horizontal, segment_vertical) et pour des segments obliques (segment_droite, segment_gauche). Nous distinguons alors ces deux cas:

CAS 1: le segment est horizontal ou vertical

Dans le cas de segments horizontaux ou verticaux la procédure de calcul des codages des cellules traversées par ceux-ci s'effectue comme suit :

a) en premier lieu, une procédure traiter_premier code la cellule C_{k_a} selon le schéma décrit plus haut.

b) ensuite, puisque le segment n'est pas oblique, on affecte à toutes les cellules suivantes sauf la dernière C_{k_b} un code qui ne dépend de la position du segment par rapport aux centres de ces cellules.

Exemple : si le segment est vertical et si le segment est à droite du centre on affecte aux cellules le code 11 sinon le code 14 (occasionnellement le code 15 dans le cas exceptionnel ou le segment passe par le centre des cellules).

III- 8

c)enfin, une procédure `traiter_dernier` code la cellule C_{k_b} .

CAS 2: Cas où le segment est oblique

Dans le cas de segments obliques , le codage s'effectue de manière différente:

a) en premier lieu, une procédure `traiter_premier` code la cellule C_{k_a} et détermine le numéro de la cellule suivante ainsi que la direction entrante DE de celle-ci.
Par exemple, si le segment est un `segment_droite` alors on a DE='EST' ou 'SUD' .

b) on chemine ensuite pour toutes les cellules traversées sauf C_{k_b} de la manière suivante:

- *on détermine la direction sortante DS de la cellule à coder;

- *on code la cellule en fonction des valeurs de DE et DS et de la position du segment par rapport au centre de la cellule.

Exemple : si le segment est de type droite alors on a le raisonnement suivant

si le segment est au dessus du centre alors DS='NORD' sinon DS='OUEST'.

Si DE='EST' et DS='OUEST' alors

- si le segment est au dessus du centre alors on rajoute la contribution 13 au code de la cellule
- sinon on rajoute 7.

Si DE='EST' et DS='NORD' alors on rajoute la contribution 9.

Si DE='SUD' et DS='OUEST' alors on rajoute la contribution 6.

Si DE='SUD' et DS='NORD' alors

- si le segment passe à droite du centre alors on rajoute la contribution 11
- sinon on rajoute la contribution 14.

Ensuite la direction de sortie de la cellule actuelle détermine le numéro de la suivante à coder ainsi que sa direction entrante DE.

4c) enfin une procédure `traiter_dernier` code la cellule C_{k_b} en fonction de DE et de la position de l'extrémité B dans la cellule.

En définitive, une valeur comprise entre 1 et 15 de **code [k]** correspond au cas où la cellule C_k est traversée par un ou plusieurs segments des lignes D_k et une valeur égale à 100 correspond au cas où la cellule C_k n'est pas traversée.

III- 9

III- B-4 Détection et tracé d'une composante

III-B-4-a Détection d'une composante.

On effectue un balayage des cellules de gauche à droite et de bas en haut jusqu'à ce que l'on détecte une cellule C_k traversée par une composante et ne contenant pas de ligne D_k de discontinuité et telle que $k \equiv 0 \text{ modulo } n_x$.

Pour tester si une cellule C_k est traversée par une composante, on utilise la fonction suivante *énergie_niveau* :

```
fonction énergie_niveau (k: entier): entier ;  
  { $k \in [1, n_x(n_y - 1)]$ ;  $z_k$  est la valeur au sommet d'indice k de M}  
variables locales : a,b,c,d: réel ;  
                  e: entier;  
début  
  a:= $z_k$ ; b:= $z_{k+1}$ ; c:= $z_{k+n_x+1}$ ; d:= $z_{k+n_x}$ ;  
  e:=0;  
  si  $a \geq z_0$  alors e := e+8;  
  si  $b \geq z_0$  alors e := e+4;  
  si  $c \geq z_0$  alors e := e+2;  
  si  $d \geq z_0$  alors e := e+1;  
  énergie_niveau := e  
fin ;
```

Cette fonction retourne un entier compris entre 0 et 15.

Il est facile de voir que :

- * les valeurs extrêmes 0 et 15 correspondent aux cas où la cellule n'est pas traversée par le niveau z_0 .

- * pour chaque valeur de cette fonction il ne correspond qu'un seul couple (deux lorsqu'elle vaut 10 ou 5) de côtés traversés.

- * les valeurs 10 et 5 correspondent aux cas où la cellule est traversée par une ou deux composantes sur ses quatre côtés. Il est impossible de choisir à priori quels seront les côtés traversés par une des composantes. Ces deux valeurs conduisent chacune à 3 cas de dégénérescence qui sont schématisés ci-dessous (figure III-4).

III- 10

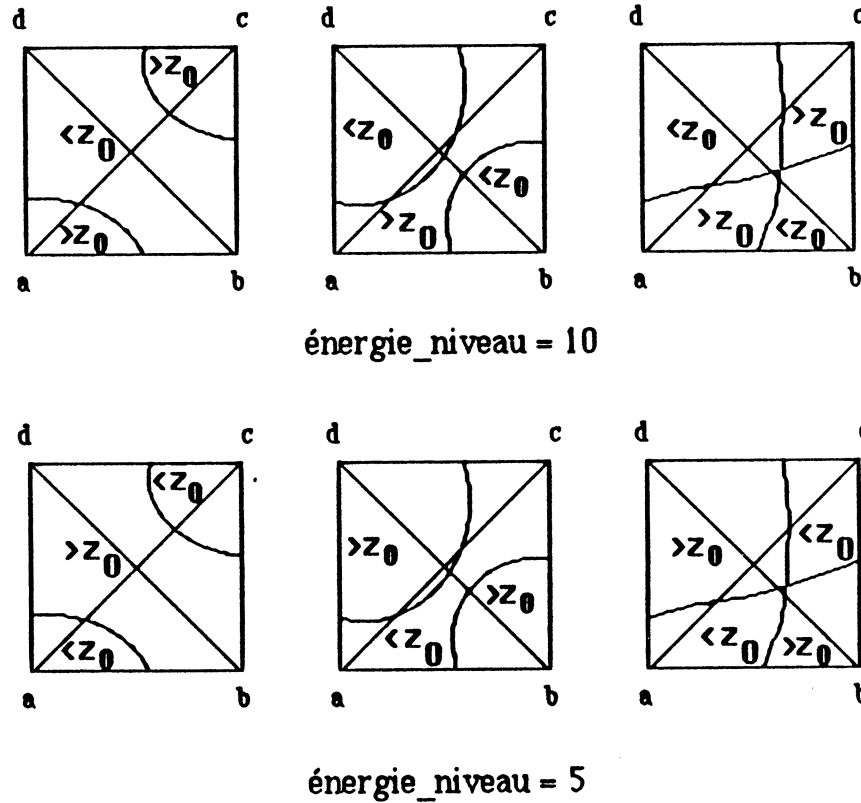


Figure III-4

Pour chaque valeur, dans les deux premiers cas on applique une heuristique basée sur la valeur de la surface au point central et qui sera expliquée dans le sous-paragraphe suivant; le dernier cas étant supposé ne pas apparaître d'après l'hypothèse H3 citée plus haut.

En résumant ceci, on détecte la première cellule suivant le balayage proposé en testant si la fonction énergie_niveau de la cellule est comprise entre 1 et 14; dans ce cas l'une des intersections du niveau avec les deux (ou les quatre) côtés est stockée dans le premier éléments du tableau intersection.

III-B-4-b Suivi d'une composante et stockage

On sait à ce stade qu'un côté (a_0, b_0) d'une cellule C_{k_0} est intersecté par une composante de niveau (z_0) et que celle-ci ne contient pas de lignes D_i .

En premier lieu, le triplet (a_0, b_0, k_0) est stocké temporairement dans un triplet (a_1, b_1, k_1) , ceci dans un double but :

III-11

- ceci permettra à tout moment de tester si la composante se referme dans une même cellule.
- dans le cas où la composante est ouverte, k_1 est l'indice à partir duquel on parcourra l'autre branche de la composante.

On chemine le long d'une partie de la composante en stockant au fur et à mesure dans le tableau **intersection** les intersections du niveau avec les côtés traversés jusqu'à ce que l'on rencontre l'une des 3 situations citées plus haut.

Dans le cas où la composante est ouverte, on inverse l'ordre des éléments du tableau **intersection** et on parcourt l'autre partie de la composante.

Le cheminement s'effectue de cellule en cellule de la manière suivante:

une procédure "suivi_cellule" ayant comme triplet "entrant" (a,b,k) où k est le numéro de la cellule et a, b sont les indices des sommets du côté intersecté, détermine un triplet sortant (a_s, b_s, k_s) où k_s est l'indice de la cellule suivante rencontrée par la composante et a_s, b_s les indices des sommets du côté rencontré.

Selon les valeurs du code de la cellule C_k , plusieurs situations sont à envisager. Nous les détaillons ci-dessous.

- $\text{code}[k]=100$

La cellule n'étant pas traversée par une ligne, il est facile de déterminer, grâce à l'énergie du niveau de la cellule, le triplet sortant (a_s, b_s, k_s) en dressant la liste de tous les cas possibles.

Si l'énergie du niveau est égale à 10 ou 5 (ce qui rappelons-le, correspond aux 2 cas de dégénérescence cités plus haut), alors :

- * d'une part, $\text{code}[k]$ prend la valeur 200
- * d'autre part, l'heuristique de choix du triplet sortant est la suivante:

Prenons l'exemple suivant (figure III-5)

III-12

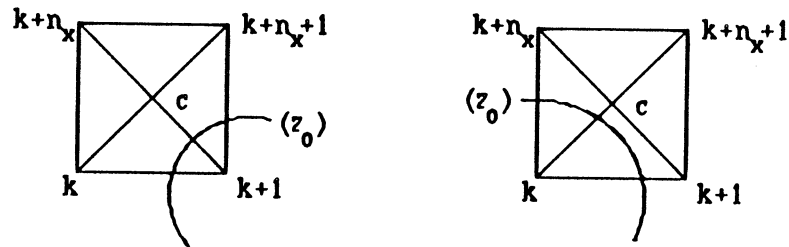


Figure III-5

Supposons que $a=k$, $b=k+1$, $z_a > z_0$ et $z_b < z_0$;

si $z_c > z_0$ alors nous choisissons $a_s = k+1$, $b_s = k+n_x+1$, $k_s = k+1$ (ceci correspond à la figure de gauche ci-dessus)

sinon nous choisissons $a_s = k$, $b_s = k+n_x$, $k_s = k-1$;

- $\text{code}[k] = 200$

Ce cas nous indique que la cellule a déjà été visitée auparavant et on applique alors l'heuristique de choix citée ci-dessus. Enfin on affecte à $\text{code}[k]$ la valeur -100 pour signifier que la cellule ne pourra plus être visitée ultérieurement.

- $\text{code}[k] \in \{7, 11, 13, 14\}$

Ces valeurs correspondent aux cas où trois triangles de la cellule sont traversés par une ou plusieurs lignes D_i .

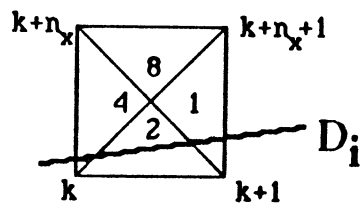


Figure III-6

On teste d'abord si le côté (a,b) intersecté par la composante est un des côtés des triangles traversés par les lignes. Si cela est le cas, alors la composante s'arrête et $(a_s, b_s, k_s) = (a, b, k)$; sinon on détermine le côté suivant intersecté en utilisant la valeur de la surface au centre de la cellule. La composante s'arrête dans cette cellule.

III-13

Exemple: (figure III-6)

Supposons que $\text{code}[k] = 7$, $a = k + n_x$, $b = k + n_x + 1$, $z_a > z_0$ et $z_b < z_0$.

Soit z_c la valeur de la surface au centre de la cellule.

Si $z_c > z_0$ alors $(a_s, b_s, k_s) = (b, c, k)$ sinon $(a_s, b_s, k_s) = (a, c, k)$.

• $\text{code}[k] \in \{3, 6, 9, 12\}$

Ces valeurs correspondent aux cas où deux triangles de C_k sont traversés par des lignes D_i .

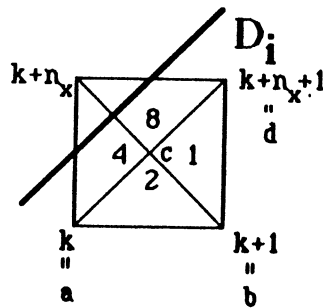


Figure III-7

Comme dans le cas précédent, on teste si le côté (a, b) est un des côtés des triangles traversés par les D_i . Si ce n'est pas le cas, alors on détermine les côtés traversés par les composantes jusqu'à que la composante s'arrête (c'est à dire jusqu'à ce que le côté rencontré soit un des côtés des triangles traversés) ou continue vers une cellule suivante (c'est-à-dire jusqu'à ce que le côté rencontré soit un côté de C_k n'étant pas un des côtés des triangles traversés).

Exemple:(figure III-7)

Supposons $\text{code}[k] = 12$, $a = k, b = k + 1$, $z_a < z_0, z_b > z_0$ et Soit z_c la valeur de la surface au centre de la cellule. Nous appliquons la démarche suivante:

si $z_c > z_0$ alors $(a_s, b_s, k_s) = (a, c, k)$ { la composante s'arrête dans la cellule C_k }

sinon début

$d = k + n_x + 1$;

si $z_d > z_0$ alors $(a_s, b_s, k_s) = (c, d, k)$ { la composante s'arrête dans la cellule C_k }

sinon $(a_s, b_s, k_s) = (b, d, k + 1)$ { la composante continue vers la cellule C_{k+1} }

fin ;

III-14

- $\text{code}[k] \in \{1, 2, 4, 8\}$

Ces valeurs correspondent aux cas où un seul triangle de C_k est traversé par des lignes D_i .

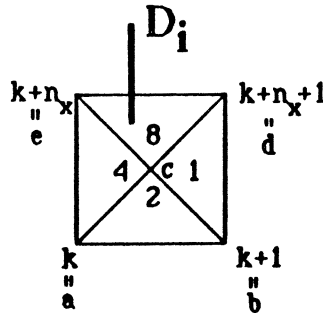


Figure III-8

Comme dans les cas précédents, on teste si le côté (a,b) est un des côtés du triangle traversé par les lignes D_i .

Si ce n'est pas le cas, alors on applique un algorithme analogue au cas précédent.

Exemple: (figure III-8)

Supposons que $\text{code}[k] = 8$, $a = k$, $b = k+1$, $z_a < z_0$, $z_b > z_0$. Soit z_c la valeur de la surface au centre de la cellule. Nous appliquons alors le raisonnement suivant:

```

si  $z_c > z_0$  alors
  début
     $e = k + n_x$ ;
    si  $z_e > z_0$  alors  $(a_s, b_s, k_s) = (c, e, k)$  {la composante s'arrête dans la cellule  $C_k$ }
    sinon  $(a_s, b_s, k_s) = (b, d, k-1)$  {la composante continue vers la cellule  $C_{k-1}$ }
  fin
sinon
  début
     $d = k + n_x$ ;
    si  $z_d > z_0$  alors  $(a_s, b_s, k_s) = (c, d, k)$  {la composante s'arrête dans la cellule  $C_k$ }
    sinon  $(a_s, b_s, k_s) = (b, d, k+1)$  {la composante continue vers la cellule  $C_{k+1}$ }
  fin ;
  
```

Après l'acquisition du triplet suivant (a_s, b_s, k_s) , on teste si celui-ci se situe sur le bord

III-15

du domaine et on calcule les coordonnées du point d'intersection de la composante et du côté(a_s, b_s). Ensuite, **code[k]** prend sa valeur opposée sauf si **code[k]**=200 (auquel cas **code[k]**:= -100) .

III-B-4-c Intersection et tracé des composantes

On détaille ici :

- le calcul du point d'intersection d'un côté (a,b) avec une composante qui sera stockée dans le tableau **intersection**. Ceci est réalisé en général grâce à un schéma d'interpolation cubique décrit ci-dessous.
- le calcul des coefficients d'une courbe paramétrée(en général cubique) à partir des valeurs du tableau **intersection** permet le tracé de la composante. De plus, on sait grâce à l'algorithme décrit dans le sous-paragraphe précédent si celle-ci est ouverte ou non.

Intersection et interpolation

Pour un côté (a,b) donné d'une cellule d'indice k, on calcule les coefficients du polynôme cubique de Lagrange interpolant z_a, z_b et deux sommets voisins aux valeurs 0, 1, 2, 3 lorsque cela est possible c'est-à-dire que lorsque ni a ni b se situent sur le bord, qu'il n'y a pas de ligne D_i à proximité immédiate de (a, b) et que ce côté est horizontal ou vertical.

Dans ce cas, on a

$$(a, b) \in \{(k, k+1), (k, k+n_x), (k+n_x, k+n_x+1), (k+1, k+n_x+1)\}$$

et le couple des deux sommets voisins est l'un des éléments de l'ensemble V suivant:

$$V = \{(a-1, b+1), (a-2, a-1), (b+1, b+2), (a-n_x, b+n_x), (a-2.n_x, a-n_x), (b+n_x, b+2.n_x)\}$$

Exemple: Pour fixer les idées, supposons que (a, b) soit un côté horizontal. Soit un booléen **cubique** mis à vrai si l'interpolation par un polynôme P_3 de degré 3 est possible. Alors les indices des sommets des points t_0, t_1, t_2, t_3 de M requis pour l'interpolation et le booléen **cubique** requis pour le calcul des coefficients s'obtiennent en appliquant l'algorithme suivant:

III-16

```
si a n'est pas sur le bord gauche de D alors
  si b n'est pas sur le bord droit de D alors
    si le côté(a-1, a) n'est pas traversé par une ligne alors
      si le côté (b, b+1) n'est pas traversé par une ligne alors
        début
           $t_0 := a-1; t_1 := a; t_2 := b; t_3 := b+1; \text{cubique} := \text{vrai}$ 
        fin
      sinon
        si a-1 n'est pas sur le bord gauche de D alors
          si le côté (a-2, a-1) n'est pas traversé par une ligne alors
            début
               $t_0 := a-2; t_1 := a-1; t_2 := a; t_3 := b; \text{cubique} := \text{vrai}$ 
            fin
          sinon cubique := faux
        sinon cubique := faux
      sinon
        si le côté (b, b+1) n'est pas traversé par une ligne alors
          si b+1 n'est pas sur le bord droit de D alors
            début
               $t_0 := a; t_1 := b; t_2 := b+1; t_3 := b+2; \text{cubique} := \text{vrai}$ 
            fin
          sinon cubique := faux
        sinon cubique := faux
      sinon { b se situe sur le bord droit de D }
        si le côté (a-2, a-1) n'est pas traversé par une ligne alors
          début
             $t_0 := a-2; t_1 := a-1; t_2 := a; t_3 := b; \text{cubique} := \text{vrai}$ 
          fin
        sinon cubique := faux
      sinon { a est sur le bord gauche }
        si le côté(b+1, b+2) n'est pas traversé par une ligne alors
          début
             $t_0 := a; t_1 := b; t_2 := b+1; t_3 := b+2; \text{cubique} := \text{vrai}$ 
          fin
        sinon cubique := faux
```

III - 17

Un moyen simple pour savoir si a est sur le bord droit de D (respectivement b sur le bord gauche) est de tester si $a \neq 1$ modulo n_x (respectivement $b \neq 0$ modulo n_x).

De même, pour savoir si le côté $(a-1,a)$ est traversé par une ligne il suffit d'effectuer le test suivant : $| \text{code}[a-1] | + | \text{code}[a-1-n_x] | > 100$.

Lorsque l'interpolation cubique est possible (c'est-à-dire lorsque **cubique** est vrai) on calcule les coefficients de P_3 en résolvant le système : $P_3(i) = z_{i_i} ; i=0, 1, 2, 3$.

On résoud ensuite l'équation $P_3(u)=z_0$ par dichotomie, ce qui fournit les coordonnées du point cherché.

Si cela n'est pas possible (par exemple, lorsque le segment est un des côtés d' un triangle d' une cellule traversée par une ligne) , le point d'intersection est évalué par interpolation linéaire.

Tracé de la composante

La composante sera représentée par une courbe cubique $s=\{x(u),y(u), u \in J\}$ paramétrée de classe C^1 , u désigne le paramètre décrivant un intervalle J de $\mathbb{R}$. Nous reprenons ici le procédé d'interpolation de Butland [Butland 80].

Considérons l'ensemble de données $E = \{ M_i=(x_i,y_i); i=1,\dots,n\}$ destinées à être interpolées par s , où on a posé $x_i = \text{intersection}[i,1]$ et $y_i = \text{intersection}[i,2]$.

Notons: k_i soit x_i soit y_i

k'_i soit $x'(u_i)$ soit $y'(u_i)$ $i=1,\dots,n$ où $u_i \in J$

$dk_i = k_{i+1} - k_i$

Butland propose la paramétrisation uniforme ($u_i=i; i=1,\dots,n$) avec la dérivée aux noeuds selon la formule ci-dessous:

$k'_i = \text{moyenne_harmonique}(dk_{i-1}, dk_i)$ $i= 2, \dots, n-1$

où :

$$\text{moyenne_harmonique}(a, b) = \begin{cases} \frac{2ab}{a+b} & \text{si } a, b \geq 0 \\ 0 & \text{sinon} \end{cases}$$

Le temps de calcul des pentes aux noeuds est en $O(n)$.

La courbe s est alors définie par $s = \{x(u), y(u) ; u \in [1,n]\}$ où

III - 18

$$\begin{cases} x(u) = p_i(u) \\ y(u) = q_i(u) \end{cases} \quad u \in [i, i+1]$$

où p_i, q_i sont des polynômes de degré 3 vérifiant :

$$\begin{cases} p_i(j) = x_j ; p'_i(j) = x'_j \\ q_i(j) = y_j ; q'_i(j) = y'_j \end{cases} \quad j \in [i, i+1] ; i = 1, \dots, n-1$$

avec x'_j, y'_j déterminés par la formule précédente.

Si la courbe est fermée, c'est à dire lorsque $M_1 = M_n$, alors on choisit :

$$k'_1 = k'_n = \text{moyenne_harmonique} (k_1 - k_{n-1}, k_2 - k_1).$$

Si ce n'est pas le cas, k'_1, k'_n sont choisis de telle façon qu'ils vérifient des conditions de bout libre ($s''(1) = s''(n) = 0$), c'est-à-dire :

$$k'_1 = (k'_2 - 3 m_1) / 2 \quad \text{et} \quad k'_n = (k'_{n-1} - 3 m_{n-1}) / 2$$

$$\text{où} \quad m_1 = k_2 - k_1 \quad \text{et} \quad m_{n-1} = k_n - k_{n-1}$$

De plus s préserve la monotonie des données [MEDINA 85].

La mise en oeuvre du calcul de s est aisée et rapide et par conséquent le tracé est facilement obtensible.

III-B-5 Tableau synthétique de l'algorithme

Auparavant, nous décrivons les rôles des variables, des procédures et fonctions utilisées dans le tableau qui sera présenté ci-après.

- k est l'indice de la cellule actuelle C_k dans laquelle on chemine.
- (a, b) est un côté de C_k traversé par (z_0) .
- k_s est l'indice de la cellule suivante C_{k_s} qu'on va traiter ultérieurement. Il est déterminé par la fonction `numéro_cellule_suivante` décrite ci-dessous.
- (a_s, b_s) est un côté de C_{k_s} traversé par (z_0) . Il est déterminé grâce à la procédure `suivi_cellule` décrite ci-dessous.

III - 19

- k_1 est l'indice de balayage du maillage M . Il sert aussi de variable tampon lorsque la composante de niveau (z_0) est ouverte.
- (a_1, b_1) est un côté qui est une variable tampon permettant de stocker un côté de la première cellule détectée traversée par (z_0) .
- `stop_composante` est un booléen indiquant si au cours du cheminement de la composante, celle-ci s'arrête sur le bord de D ou sur une ligne D_i .
- `composante_fermée` est un booléen signalant si la courbe est fermée. Ceci sert lors des calculs des coefficients de la courbe représentant la composante.
- `quitte_ligne` est un booléen indiquant si la composante "quitte" la ligne contenant la cellule actuelle C_k (voir paragraphe III-B-2).
- k_d est l'indice de la cellule à partir de laquelle on reprendra le balayage (voir paragraphe III-B-2).
- *fonction* `numéro_cellule_suivante((a,b):côté; k:entier):entier;`
 Etant donné un côté (a,b) d'une cellule C_k , cette fonction retourne l'entier k' tel que $C_k \cap C_{k'} = (a,b)$ sauf si (a,b) est sur le bord de M , auquel cas la fonction retourne l'entier 0.
- *fonction* `détection (variable (a,b):côté; k:entier):booléen;`
 Détecte si une cellule C_k est traversée par (z_0) grâce la fonction énergie niveau décrite dans le paragraphe III-B-4. Dans le cas favorable, cette fonction retourne un des côtés (a,b) intersectés avec le niveau, et l'intersection de celui-ci est stocké dans le premier élément du tableau **intersection**.
- *procédure* `suivi_cellule((a,b):côté; k:entier; variable:(a_s,b_s):côté; variable k_s:entier);`
 Etant donné un côté "entrant" (a,b) d'une cellule C_k , cette procédure
 - détermine le côté "sortant" (a_s, b_s) et le numéro k_s de la cellule suivante.
 - met à jour le booléen `stop_composante` en testant si (a_s, b_s) est sur le bord de M .
 - calcule l'intersection de (a_s, b_s) et de (z_0) . Celle ci sera stockée dans le tableau **intersection**.

III - 20

Tableau synthétique

entrées : z_0 : réel;
 n_x, n_y : entier;
 z_k $k=1, n_x \cdot n_y$.
les lignes D_i ; $i=1, \dots, p$.

début
codage des cellules; {remplit le tableau code }
répéter
 $k_1 := 1$; $k_d := 1$; $k := 1$;
si (code[k] = 100) et (détection (a, b, k)) alors
début
 $(a_1, b_1) := (a, b)$; $k_1 := k$; stop := faux; quitte_ligne := faux;
répéter { on chemine jusqu'à que la composante se referme ou s'arrête }
suivi_cellule ((a, b), k, (a_s, b_s), k_s);
si non (quitte_ligne) et ($k_s > k + n_x$) alors { si on "quitte" la ligne
début alors on stocke k_d }
quitte_ligne := vrai; $k_d := k$
fin;
si non (stop_composante) alors
début
 $(a_s, b_s) := (a, b)$; $k_s := k$
fin
sinon
code[k_s] := - code [k_s]
jusqu'à (stop_composante) ou ($k = k_1$);
si stop_composante alors
début
stop_composante := vrai; si non (quitte_ligne) alors $k := k_d + 1$;
inversion de l'ordre du tableau intersection;
 $k := \text{numéro_cellule_suivante}(a_s, b_s, k)$; $(a, b) := (a_1, b_1)$;
si $k \neq 0$ alors { si (a,b) n'est pas sur le bord de M }
répéter
suivi_cellule ((a, b), k, (a_s, b_s), k_s);
si non (stop_composante) alors
début
 $(a_s, b_s) := (a, b)$; $k_s := k$
fin
sinon
code[k_s] := -code [k_s]
jusqu'à stop_composante;
fin;
composante_fermée := ($k = k_1$);
calcul des coefficients de la courbe représentant la composante ;
tracé de la composante;
 $k_1 = k_d$;
fin;
jusqu'à $k_1 \geq n_x \cdot (n_y - 1)$
fin .

III - 21

III-C Exemples et conclusions

III-C-1 Exemples

Le programme a été écrit en Pascal Turbo sur un micro-ordinateur IBM PC et les dessins ont été effectués sur une table traçante BENSON pilotée par le micro-ordinateur.

Pour chaque exemple, nous donnons sa représentation en trois dimensions ainsi que la carte des courbes de niveaux correspondantes .

Nous donnons ici à titre d'illustration du fonctionnement de l'algorithme quatre ensembles de données :

- le premier (voir figure III-9) est un ensemble de valeurs calculées sur un maillage régulier 39×39 du domaine $[0,10] \times [0,10]$ de la fonction suivante :

$$f(x, y) = \sum_{i=1}^4 \frac{\sin(r_i(x, y))}{r_i(x, y)} \quad \text{où} \quad r_i(x, y) = \sqrt{(x-x_i)^2 + (y-y_i)^2} \quad i=1, \dots, 4$$

et $(x_1, y_1) = (5,2)$, $(x_2, y_2) = (5,8)$, $(x_3, y_3) = (2,5)$, $(x_4, y_4) = (8,5)$.

Cette surface possède 4 points de selle situés en $(3.5, 3.5)$, $(6.5, 3.5)$, $(6.5, 3.5)$, $(6.5, 6.5)$. Il faut remarquer que l'on localise ceux-ci sur la carte.

Cet exemple illustre ici le fait que l'algorithme fonctionne aussi pour des surfaces régulières.

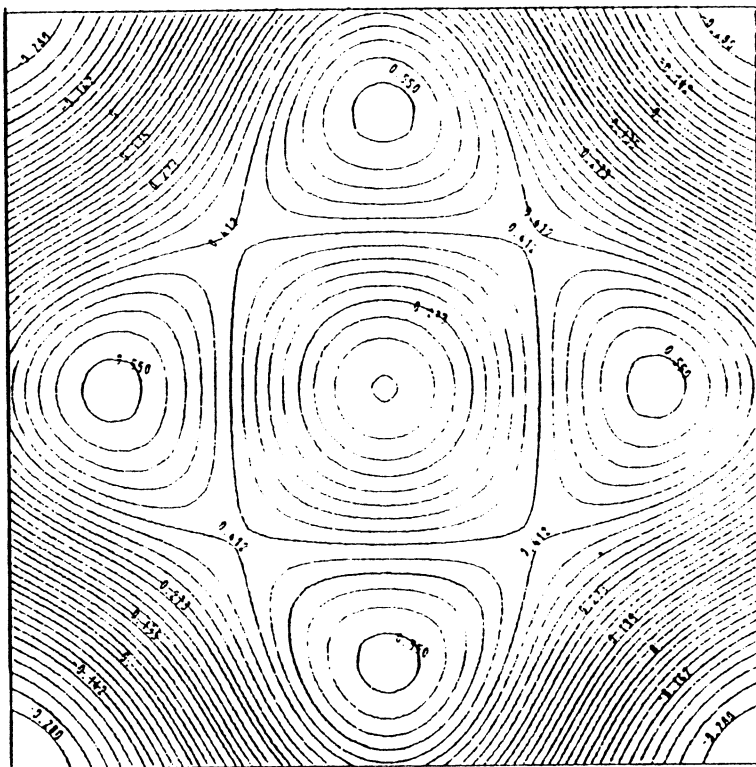
- Le deuxième exemple (voir figure III-10) est un ensemble de points d'une surface définie sur le domaine $[0,10] \times [0,10]$ évaluée aux noeuds d'un maillage régulier 38×38 possédant une discontinuité le long du segment rectiligne $\{(5,2.5), (5,7.5)\}$.

- Le troisième exemple (voir figure III-11) est un ensemble de points d'une surface définie sur le domaine $[0,10] \times [0,10]$ évaluée aux noeuds d'un maillage régulier 39×39 possédant une discontinuité le long des lignes brisées:

$$D_1 = \{(3,1), (7,1)\}$$

$$D_2 = \{(2,3), (5,7), (9,4)\}$$

III - 22



NIVEAU MIN -2 300E-001

NIVEAU MAX 5 5000E-001

PAS 2 7667E-002

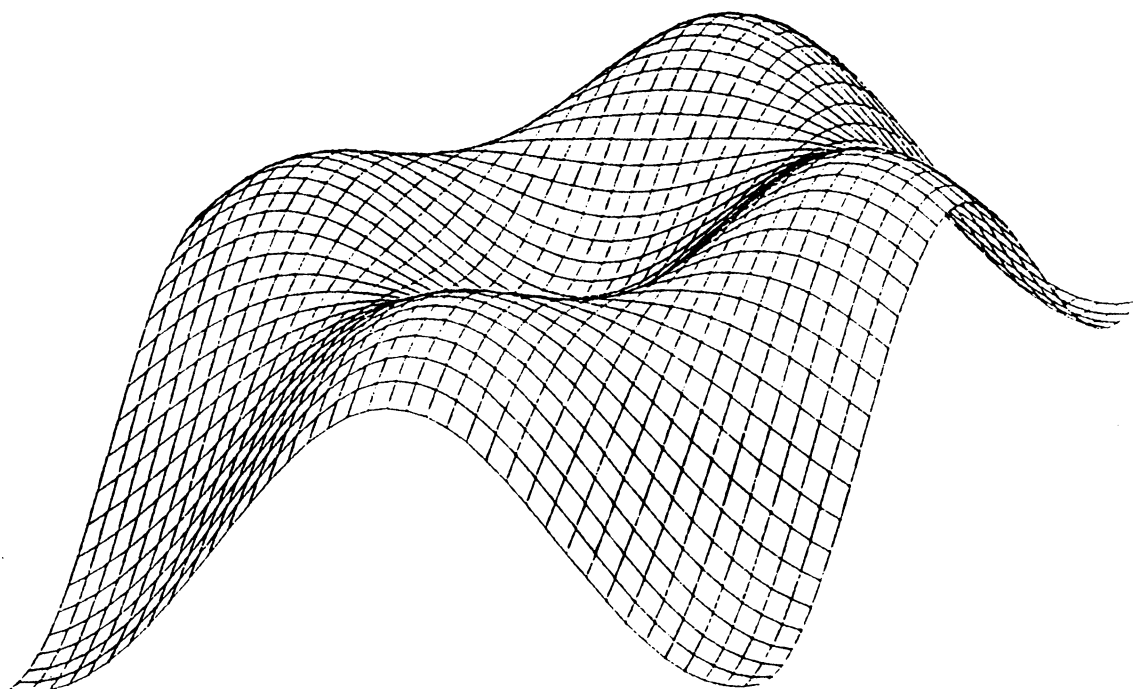


Figure III-9

III -23

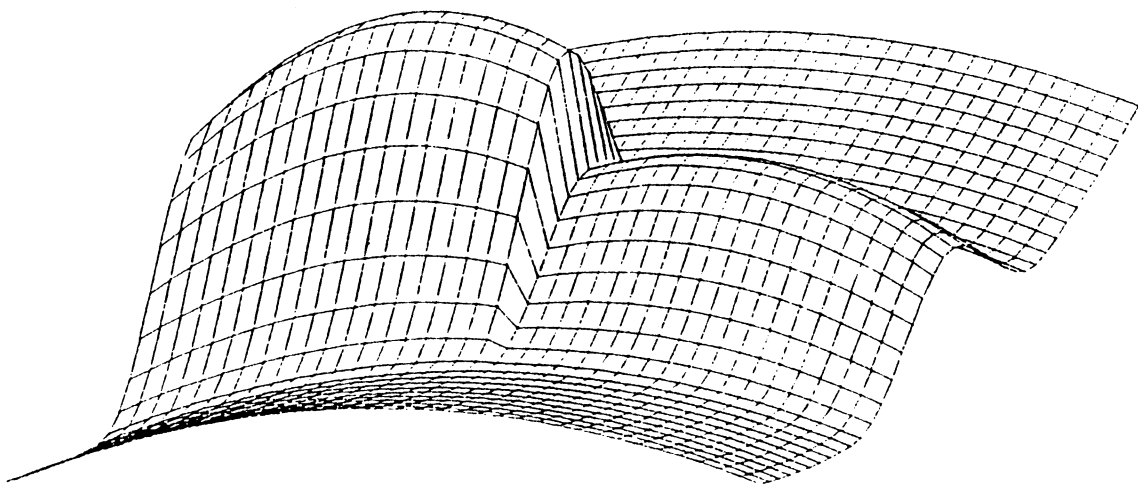
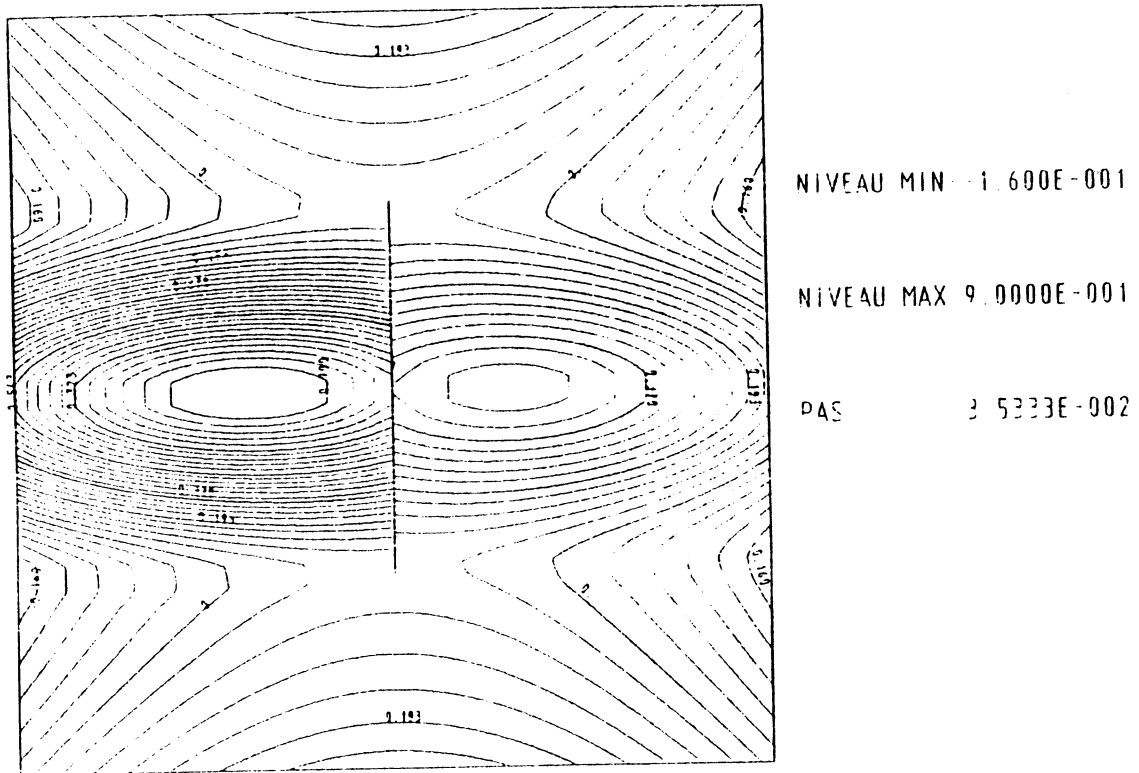


Figure III-10

PAS 0 0000E-002

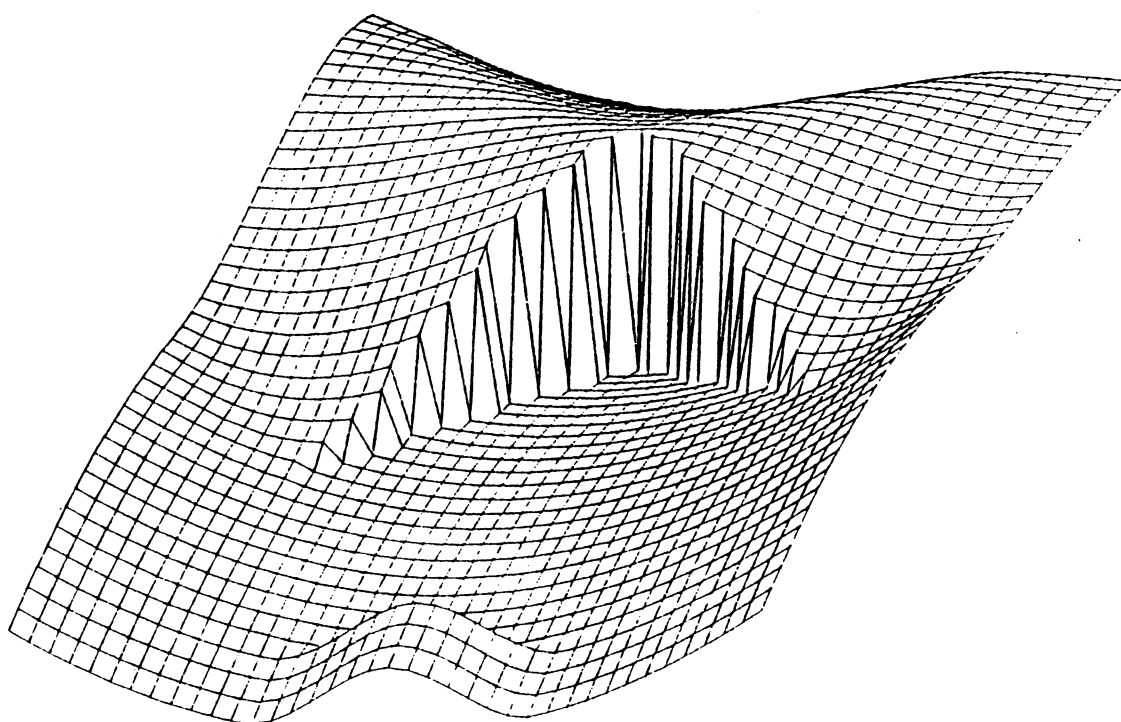
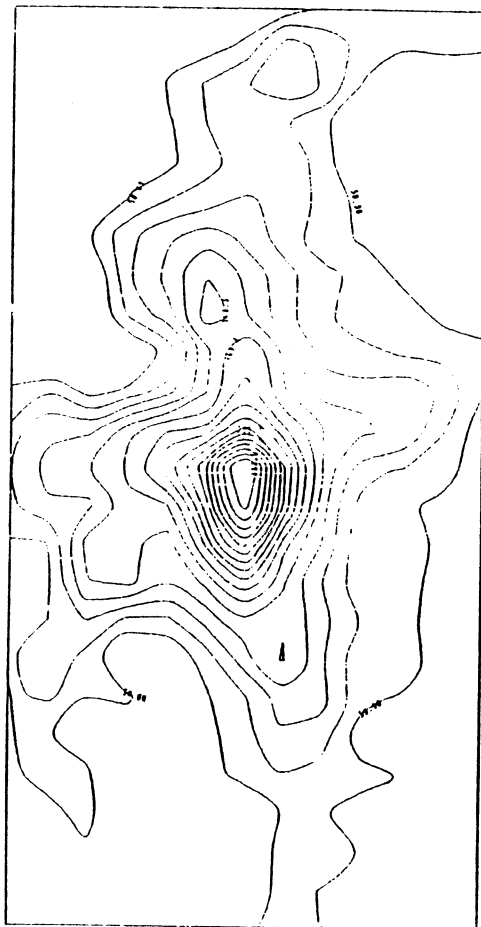


Figure III-11

III -25



NIVEAU MIN 5.0000E+001

NIVEAU MAX 1.3000E+002

PAS 1.9667E+001

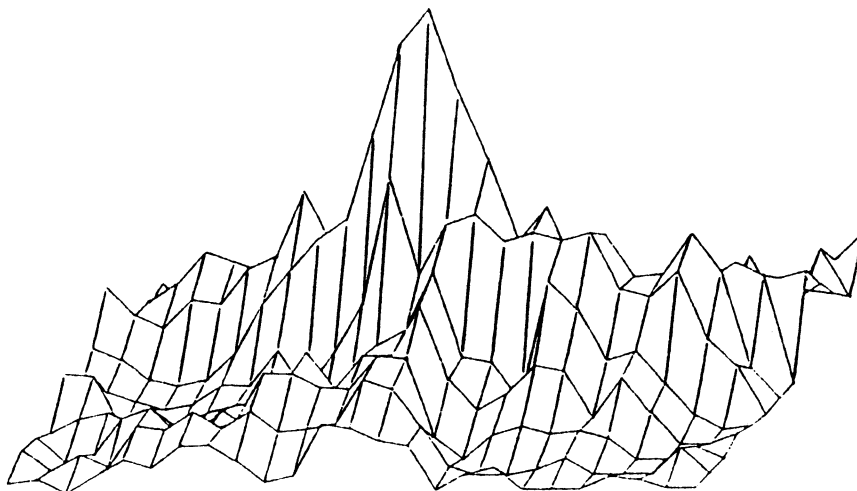


Figure III-12

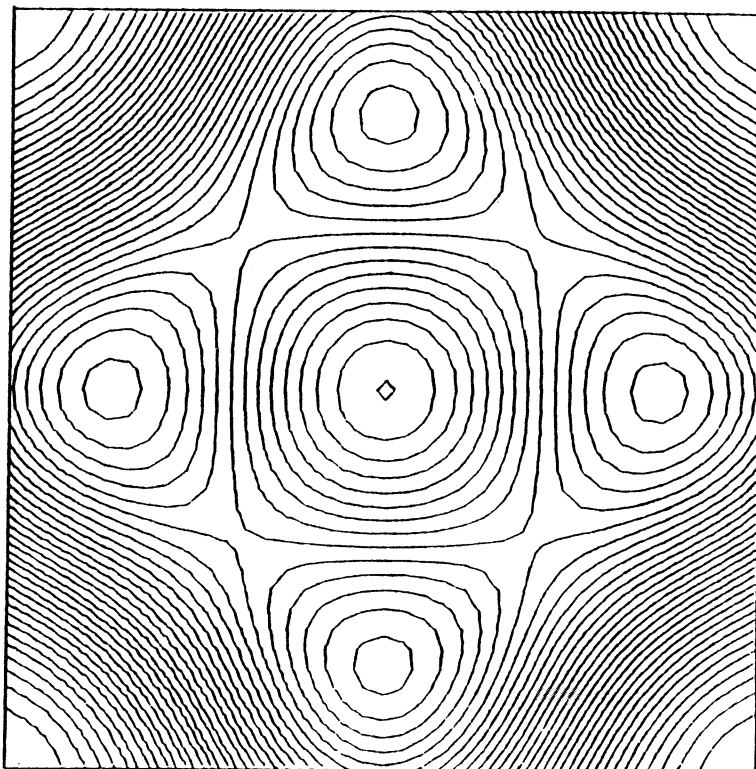
III - 26

- le quatrième exemple (voir figure III-12) est un ensemble de données réelles provenant d'observations astronomiques effectuées par le laboratoire du CERMO (Grenoble): il s'agit de l'aire intégrée de la raie de CO (transition 2-1) observée dans le nuage moléculaire géant ORION, sur la fréquence $230.538 \cdot 10^9 \text{ Hz}$. Les données sont aux sommets d'un maillage 14×25 du domaine $[20,150] \times [20,270]$.

Le point central dénommé "ORIA" (la température y est supérieure à 6000K) il correspond à une étoile qui est en train de naître.

Pour chacun des exemples présentés, les temps de tracé des dessins sont de l'ordre de 15 minutes (il y a environ 20 niveaux par carte), la majeure partie étant consacrée au tracé des courbes.

Pour diminuer le temps, on peut simplifier l'algorithme en traçant des segments de droite dans chacune des cellules traversées; en contrepartie le dessin est moins "visuellement" plaisant en général et devient "acceptable" (avec un temps d'exécution de l'ordre de 5 minutes) lorsque le maillage du domaine est suffisamment fin pour qu'on ne puisse plus distinguer à l'oeil les ruptures de pente entre chaque segment (voir figure III-13).



NIVEAU MIN: -2.800E-001

NIVEAU MAX: 5.5000E-001

PAS: 2.7667E-002

Figure III-13

III - 27

Remarques : 1) l'heuristique de choix décrite dans le paragraphe III-B-4 basée sur la valeur centrale de la surface au centre de la cellule conduit quelquefois à des résultats indésirables :

- dans le cas (relativement rare) où une ou deux composantes intersecte quatre côtés d'une cellule.
 - au voisinage d'une ligne de discontinuité D_i , si la composante s'arrête sur cette ligne au lieu de continuer vers une autre cellule; ceci s'explique par le manque d'informations sur le comportement local de la surface au voisinage de la ligne D_i .
- 2) le procédé d'interpolation cubique de Butland peut être mis en défaut: ceci est mis en évidence visuellement par une rupture de pente d'une composante.
- 3) enfin, cet algorithme est mis en défaut pour des surfaces quelconques c'est-à-dire de la forme $f(x,y,z) = 0$.

III-C-2 Conclusions

Nous avons décrit un algorithme original de tracé automatique de courbes isovaleurs de surfaces explicites régulières sauf le long de lignes brisées. Il est composé de deux étapes successives, la première préparant la seconde; il est de type énumératif, c'est-à-dire que l'on envisage tous les cas. L'apport majeur par rapport à la plupart des algorithmes existants est qu'il prend en compte le cas de singularités le long de lignes brisées. Il donne en général des résultats satisfaisants et a fait l'objet de l'écriture d'une procédure simplifiée (on ne prend pas en compte le cas des singularités de la surface) écrite en langage Pascal qui est incluse dans la bibliothèque BANG (Bibliothèque d'Analyse Numérique de Grenoble). Toutefois des améliorations peuvent y être apportées tant au niveau du temps d'exécution (amélioration et concaténation de tests) qu'au niveau du tracé du dessin (choix d'autres procédés d'interpolation moins onéreux et plus stables): ceci est à l'étude actuellement.

CHAPITRE IV :
***SUREX* :**
UN LOGICIEL D'APPROXIMATION
DE SURFACES EXPLICITES

IV - 1

IV-A Introduction

Nous présentons brièvement ici un logiciel de calcul d'approximation de **surfaces explicites** en présence éventuelle de discontinuités de valeur et de dérivée le long de lignes polygonales. Il a été écrit en langage Pascal (Turbo Pascal) sur micro-ordinateur compatible.

Ce logiciel en fait, est un ensemble de quatre logiciels indépendants s'enchaînant de façon séquentielle (s'échangeant des informations via des structures de données), dont on donne ci-dessous la séquence:

1) **SAISIE** est un logiciel convivial d'acquisition (et de modifications) des données nécessaires au calcul de la surface que l'on désire approcher, se servant en outre des possibilités graphiques du micro-ordinateur.

2) **CALCUL** est un logiciel d'évaluation de la surface par différentes méthodes: inf-convolution spline, méthodes de Franke et Nielson.

3) **VISUAL** est un logiciel de visualisation du résultat via un module de tracé de courbes isovaleurs ou/et un module de représentation tridimensionnelle de la surface obtenue autorisant différents "points de vue".

4) Enfin, le logiciel **ERREUR** permet d'estimer différents types d'erreurs discrètes lorsque les données sont issues d'une surface donnée par l'utilisateur: erreur maximale absolue et relative, erreur moyenne, écart-type. En outre, on peut visualiser la surface "erreur" via le logiciel **VISUAL**, ainsi que l'ensemble des points dont l'erreur dépasse une valeur entrée par l'utilisateur.

Nous décrivons ci-dessous , chaque logiciel séparément, en spécifiant en détail ses fonctionnalités ainsi que la structure générale des programmes constitutants.

IV - 2

IV B - Le logiciel SAISIE

Ce logiciel conversationnel permet de saisir ou de modifier sans difficultés les données (qui sont ensuite stockées dans un fichier) nécessaires au calcul d'approximation d'une surface explicite en présence de discontinuités connues le long de lignes polygonales.

L'introduction de menus (et de sous-menus) imbriqués suffisamment clairs permet facilement à l'utilisateur de saisir ou de modifier la s.d. .

Le logiciel SAISIE est composé

- du programme principal SAISIE
- de deux sous-programmes ENTREE et MODIFICATION, le premier ayant pour rôle de saisir les données, le second de modifier des données qui viennent d'être acquises ou lues dans des fichiers créés par ce programme.
- d'un module GRAPH contenant les primitives graphiques élémentaires, ainsi que des procédures de saisie graphique de points.
- d'un module utilitaire ANNEXE qui est une boîte à outils, contenant diverses procédures: lecture et enregistrement de fichiers, visualisation de menus et choix d'une de ses options, acquisition de données réelles ou entières et vérification de celles-ci, acquisition de caractères, affichage en mode clignotant des diverse questions, etc...

Ces procédures sont utilisées dans divers endroits du programme principal et des sous-programmes.

Structure de données DONNEE et fichiers associé.

La structure de données (en abrégé, s.d.) DONNEE très simple, choisie se décompose de la manière suivante:

- Les points de donnée $M_i = (x_i, y_i)_{1 \leq i \leq np}$ et les valeurs $(z_i)_{1 \leq i \leq np}$ en ces points, où np est le nombre de points. Les triplets $(x_i, y_i, z_i)_{1 \leq i \leq np}$ sont rangés dans le tableau tabpoi[1..np, 1..3].
- Un indicateur de singularité inddisc précisant l'éventuelle présence de lignes de discontinuité qui est symbolisé par un caractère égal à 'o' ou 'n'.
- Les lignes polygonales $(F_i)_{1 \leq i \leq p}$ de discontinuité de la surface définies par une suite de points $(P_i^j)_{1 \leq j \leq p_i, 1 \leq i \leq nl}$ où p_i désigne le nombre de points de la ligne F_i , et nl le nombre de lignes. Le nombre de points définissant chaque ligne est stocké dans le tableau p[1..nl] et les coordonnées de ceux-ci sont rangés dans le tableau tablig[1..nl, 1..p[1..nl], 1..2], le $j^{ème}$ point de la ligne F_i ayant pour coordonnées tablig[i, j, k], $k=1,2$.

De plus, pour chacune de ces lignes, on précise deux indicateurs: un indicateur idf_i de discontinuité de valeur et un indicateur idd_i de discontinuité de dérivée de la surface symbolisé par des caractère égaux à 'o' ou 'n'. Ils sont rangés dans les tableaux idf[1..nl]

IV - 3

et $idd[1..nl]$ respectivement.

- Enfin, des paramètres géométriques tel que les coordonnées d'un domaine polygonal D incluant les points de données, qui sont rangés dans le tableau $tabdom[1..nd,1..2]$ où nd est le nombre de points définissant D, et les coordonnées (x_{min}, y_{min}) , (x_{max}, y_{max}) d'un cadre C englobant ce domaine.

La structure de fichiers associés retenue est la suivante:

- un fichier de type 'texte' contient les informations suivantes:
 - $inddisc$.
 - nd, np, nl .
 - si $inddisc = 'o'$ et $nl \neq 0$, les tableaux $p[1..nl]$, $idf[1..nl]$, $idd[1..nl]$.
- un fichier de type 'réel' contenant les réels suivants:
 - $x_{min}, y_{min}, x_{max}, y_{max}$
 - les tableaux $tabdom[1..np,1..2]$, $tabpoi[1..np,1..3]$ et $tablig[1..nl,1..p[1..nl],1..2]$.

Remarque:a) Le choix de cette structure de fichiers associée a une motivation double:

a) l'écriture de réels sur des fichiers de type 'texte' prend environ deux fois plus de place que sur un fichier binaire, d'où un gain substantiel de mémoire morte.

b) l'introduction d'erreurs apparaissant lors de la conversion des réels binaires en chaînes de caractères, d'où l'écriture de réels sur un fichier binaire pour des raisons de sécurité.

b) les limites actuelles du programme (qui peuvent être facilement modifiées) sont:

$$np \leq 600$$

$$nd, nl \leq 10$$

$$p_i \leq 10$$

Nous présentons ci-dessous les structures et les fonctionnalités du programme principal SAISIE et des sous-programmes principaux ENTREE et MODIFICATION.

IV-B-1 Description et fonctionnalités du programme principal SAISIE.

Toutes les questions posées à différents niveaux sont affichées en mode "clignotant".

Le menu *PRINCIPAL* est affiché propose les trois options suivantes:

- a) création d'un fichier de données
- b) modification d'un fichier de données
- c) quitter le menu *principal*

Ensuite, on demande si les valeurs de la surface aux points de donnée sont calculées ou non à l'aide d'une fonction $f(x,y)$ de deux variables écrite en Pascal. La réponse donnée repérée par le booléen $link$ servira lors de l'appel d'un des deux

IV - 4

sous-programmes. Les valeurs en ces points sont entrés manuellement si le booléen link a pour valeur faux; sinon elles sont calculées par cette fonction.

- Si l'option a) est retenue, alors le sous-programme ENTREE (s.p. IV-B-2) est appelé. Si après, pour un raison quelconque, on désire modifier ce que l'on vient d'entrer, le sous-programme MODIFICATION (s.p. IV-B-3) est appelé.

- De même si l'option b) est choisie, le sous programme MODIFICATION est appelé.
- L'option c) permet de quitter le programme SAISIE.

Enfin dans ces deux cas, on enregistre la s.d. DONNEE créée ou modifiée.

IV-B-2 Description et fonctionnalités du sous-programme ENTREE.

Ce sous-programme propose le menu *entrée* :

- a) entrer les données au clavier
- b) entrer les données graphiquement

- Si l'option a) est retenue, alors la procédure ENTREE_CLAVIER est appelée (s.p. IV-B-2-a).

- Si l'option b) est choisie, la procédure ENTREE_GRAPHIC est appelée (s.p. IV-B-2-b).

IV-B-2-a Description de la procédure ENTREE_CLAVIER.

Comme le nom de la procédure l'indique, la saisie de la s.d. est effectuée au clavier. On teste l'appartenance à D de chaque point de donnée.

La saisie est alors effectuée dans cet ordre:

1: On demande les coordonnées $(x_{\min}, y_{\min})$, $(x_{\max}, y_{\max})$ définissant le cadre C englobant le domaine polygonal D grâce à la procédure CADRE (s.p. IV-B-4).

2: On demande le nombre nd de points D_i définissant le domaine D, et ensuite les coordonnées de ces points qui sont rangés dans le tableau tabdom. On vérifie pour chacun de ces points, s'il se situe dans le cadre C, ceci grâce à la procédure DANS_CADRE (s.p. IV-B-4).

3: La prise en compte de discontinuité est ensuite demandée, dont la réponse est symbolisée par un booléen inddisc.

4: Dans le cas où la réponse à l'étape précédente est positive,

4-a: le nombre nl de lignes de discontinuité F_i est entré. On teste au passage qu'il est non nul.

4-b: Pour chaque ligne, on demande les indicateurs idf_i, idd_i . Un test de

IV - 5

cohérence vérifiant que tous les deux sont différents du caractère 'n' est effectué.

4-c: Pour chaque ligne F_i , $1 \leq i \leq n_l$, les coordonnées de chaque point qui sont rangés au fur et à mesure dans le tableau `tablig`.

5: - Le nombre n_p de points de donnée est entré.
 - les coordonnées de chacun de ces points sont ensuite saisies et stockées dans le tableau `tabpoi`.

IV-B-2-b Description de la procédure `ENTREE_GRAPHIC`.

Cette procédure offre à peu près les mêmes fonctionnalités que `ENTREE_CLAVIER`, sauf que la saisie des données s'effectue de manière graphique.

Après avoir demandé les coordonnées du cadre `C`, l'écran bascule en mode alphanumérique - graphique, ceci grâce à des procédures graphiques issues du module `GRAPH`.

L'écran se partage en deux parties:

- la partie gauche visualise les entrées: points, lignes, domaines. Une distinction est faite entre les points de nature différente: les points du domaine par des croix, les points de donnée sont symbolisés par des "pixels", les points des lignes F_i par des carrés. En outre, les numéros des points et des lignes de discontinuité sont inscrits.
- la partie de droite est l'aire de dialogue où sont posées les questions, et le type de données à saisir, et éventuellement les messages d'erreur correspondants à une fausse manœuvre de l'utilisateur.

L'acquisition graphique des points et des lignes est effectuée grâce aux déplacements d'un curseur graphique (symbolisé ici par une croix latine), déplacements qui peuvent être horizontaux, verticaux ou diagonaux, effectués grâce aux touches numériques (partie droite du clavier). Le curseur ne peut se déplacer que dans les limites graphiques correspondantes au cadre `C`.

Les coordonnées-utilisateur du point en cours d'être saisi, sont affichés dans l'aire de dialogue et sont réactualisés pour chaque déplacement du curseur graphique. On valide chaque point par la touche 'return'.

Dans le cas où la surface n'est pas donnée par fonction (i.e. `link` a pour valeur faux), la valeur correspondante est demandée.

On ne demande pas a priori les nombres de points n_p, n_d , ou p_i , $1 \leq i \leq n_l$. (au contraire de `ENTREE_CLAVIER`). On finit de saisir les points d'un type donné en frappant la touche 'Q' (pour quitter). Un message en bas de l'écran rappelle d'ailleurs ceci à tout moment.

Enfin, à la fin de la saisie, une copie graphique (horizontale ou verticale) sur imprimante est proposée.

IV - 6

IV-B-3 Description et fonctionnalités du sous-programme MODIFICATION.

On effectue d'abord une lecture des fichiers associés à la s.d. .Si tout se passe bien , le sous programme offre ensuite les fonctionnalités suivantes:

- ajouter, supprimer, modifier des points ou valeurs d'interpolation.
- ajouter, supprimer, modifier des lignes F_i .
- redéfinir l'indicateur inddisc de la surface.

Plus précisément, si inddisc a pour valeur faux, le menu *modif1* suivant est affiché:

- a) quitter le menu *modif1*.
- b) redéfinir l'indicateur inddisc de la surface.
- c) ajouter des points d'interpolation.
- d) changer des points ou des valeurs d'interpolation.
- e) supprimer des points d'interpolation.

sinon le menu *modif2* suivant est affiché:

- a) quitter le menu *modif2*.
- b) redéfinir l'indicateur inddisc de la surface.
- c) ajouter des points d'interpolation ou des lignes F_i .
- d) changer des points ou des valeurs d'interpolation.
- e) supprimer des points d'interpolation ou des lignes F_i .
- f) modifier une ligne F_i .

- L'option a) de ces deux menus, permet de quitter le sous-programme MODIFICATION et de se retrouver au niveau du menu *principal*.

- L'option b) permet de "basculer" d'un de ces menus à l'autre. Ceci peut être utile dans la circonstance où on veut imposer des discontinuités d'une surface qui auparavant n'en possédait pas.

Pour toutes les autres actions, le menu *entree* (voir ci-dessus s.p. IV-B-1) est alors proposé. L'option a) appelle alors la procédure MODIF_CLAVIER, l'option b) appelle la procédure MODIF_GRAPHIC.

IV-B-3-a Description de la procédure MODIF_CLAVIER.

Cette procédure a pour en-tête: `procedure MODIF_CLAVIER (car:char).`

IV - 7

Le caractère car, déterminé par le choix d'une option du menu *modif1* ou *modif2*, caractérise l'action proposée. Enumérons-les ci-dessous:

- 'a': ajouter des points de donnée ou des lignes F_i . On propose le sous-menu:

- a) quitter ce sous-menu.
- b) ajouter des points de donnée.
- c) ajouter des lignes F_i .

L'option a) permet de se retrouver au niveau d'un des menus *modif1* ou *modif2*.

La procédure de saisie est identique à celle de ENTREE_CLAVIER. L'option b) est seulement proposée lorsque inddisc a pour valeur vrai.

- 'c': changer des points de donnée, ou uniquement des valeurs en ces points. En fait, on propose le sous-menu:

- a) quitter
- b) changer des points de donnée.
- c) changer des valeurs en des points de donnée.

L'option a) permet de se retrouver au niveau d'un des sous-menus *modif1* ou *modif2*.

Pour l'option b), il y a un léger abus de langage: "point" signifie qu'on prend en compte le point mais aussi la valeur en ce point. Enfin on enregistre ces changements.

Si l'option c) a été retenue, on demande s'il faut changer certaines ou toutes les valeurs. Le fait de pouvoir changer toutes les valeurs est d'un intérêt non négligeable: cela évite dans le cas où uniquement les valeurs sont à changer, de ressaisir toute la s.d. . Evidemment, on prend en compte le booléen link, pour l'entrée manuelle ou le calcul de ces valeurs.

- 's': supprimer des points de donnée ou des lignes F_i . On propose le sous-menu:

- a) quitter ce sous-menu.
- b) supprimer des points de donnée.
- c) supprimer des lignes F_i .

L'option a) permet de se retrouver au niveau d'un des menus *modif1* ou *modif2*.

Pour l'option b), après avoir demandé le nombre de points et les coordonnées de ceux-ci, on enregistre la suppression des points. Une action similaire est effectuée pour le choix de l'option c).

- 'm': modifier une ligne F_i . On propose le sous-menu suivant (sous réserve que $nl \neq 0$):

- a) quitter ce sous-menu.
- b) supprimer des points d'une ligne.
- c) changer des points d'une ligne.
- d) Insérer des points dans une ligne.

IV - 8

L'option a) permet de se retrouver au niveau du menu *modif2*.

Pour l'option b), après avoir demandé le nombre de points et les coordonnées de ceux-ci, on enregistre la suppression des points. Une action similaire est effectuée pour le choix des options c) et d).

IV-B-3-b Description de la procédure MODIF_GRAPHIC.

Comme pour la procédure ENTREE_GRAPHIC, cette procédure offre les mêmes fonctionnalités que la procédure MODIF_CLAVIER, la différence essentielle étant l'introduction de l'aspect graphique. Après la lecture des fichiers, la visualisation de la s.d. apparaît dans la partie gauche de l'écran. La saisie graphique de points et des lignes est identique à celle de la procédure ENTREE_GRAPHIC. Toute modification (ajout, insertion, etc...) est automatiquement réactualisée graphiquement, ce qui permet à l'utilisateur de contrôler immédiatement les actions demandées.

IV - 9

IV-C Le logiciel CALCUL

Ce logiciel offre plusieurs possibilités d'évaluation d'une surface sur une grille régulière en présence de lignes de discontinuité F_i . C'est un ensemble de programmes dont nous détaillons la liste ci-dessous:

- INFCONV est un programme de calcul d'approximation (interpolation ou ajustement) par la méthode d'inf-convolution spline.
- FRANIELS est un programme d'évaluation par des méthodes de Franke et Nielson.

Les caractéristiques communes de ces deux programmes concernent les structures de données et des résultats:

- la s.d. DONNEE lue par ces programmes est celle décrit dans le paragraphe précédent.
- la s.d. RESULTAT obtenue est indépendante du programme.

Description de la s.d. RESULTAT et du fichier associé.

Décrivons ci-dessous la s.d. RESULTAT:

- nombres de subdivisions de la grille G : p_x , p_y .
- les valeurs de la surface aux sommets de G stockés dans le tableau surf $[1..n_x, 1..n_y]$.
- les coordonnées extrémales du point bas-gauche (x_{min}, y_{min}) et du point haut-droite (x_{max}, y_{max}) de G.
- le nombre de lignes F_i : n_l .
- les coordonnées des points de chacune de ces lignes, stockées dans le tableau tablig.

Le fichier associé à cette s.d. est un fichier de réels contenant les informations décrites ci-dessus dans le même ordre. Nous décrivons ci-dessous la structure générale de ces deux programmes.

IV-C-1 Description du programme INFCONV

Ce programme reprend les idées évoquées au chapitre II. Une de ses particularités est qu'il adopte une stratégie adaptée selon le nombre de points (de donnée et des lignes F_i).

IV - 10

Il réalise dans l'ordre ci-dessous les actions suivantes:

- 1 - lecture des fichiers associés à une s.d. .
- 2 - si la taille du système linéaire est faible (≤ 130), on va au point 4.
- 3 - lecture au clavier du nombre de recouvrements n_x, n_y le long des axes; les abscisses $(a_i, b_i)_{1 \leq i \leq n_x}$ et les ordonnées $(c_j, d_j)_{1 \leq j \leq n_y}$ définissant le rectangle $R_{i,j}$ sont ensuite demandés.
- 4 - Si on désire l'ajustement des données, le paramètre d'ajustement p est alors demandé.
- 5 - On calcule pour chaque ligne F_i les vecteurs S_i, T_i tels que le graphe de F_i soit défini par une fonction dans le repère de vecteurs S_i, T_i . S_i est pris comme le vecteur directeur de la droite de régression linéaire ajustée sur les points de la ligne F_i . Si la taille du système est faible, on va au point 7.
- 6 - Pour chaque rectangle $R_{i,j}$:
 - 6a - on recherche les points de donnée et les segments des lignes "tombant" dedans.
 - 6b - la matrice du système linéaire associée à chaque "inf-convolution spline locale $\sigma_{i,j}$ " est calculée et stockée dans le tableau surf (dans le but de gagner de la place mémoire vive). La résolution est faite par la méthode de Householder, et les coefficients solutions sont stockés dans des fichiers. Cette manipulation relativement lourde (i.e. d'écrire sur mémoire morte) est justifiée par le fait que la mémoire vive accessible est faible (64K), ce qui limite sérieusement la taille des variables utilisés. On va ensuite au point 8.
- 7 - On résoud le système correspondant à l'inf-convolution spline globale et on va au point 9.
- 8 - lecture des coordonnées des points extrémaux de la grille, ainsi que ses tailles p_x et p_y . L'évaluation de l'inf-convolution spline globale est alors réalisée selon le schéma de raccordement décrit au chapitre II.
- 9 - On enregistre la s.d. RESULTAT conformément à ce qui a été écrit au début de ce paragraphe.

Remarques: a) La procédure de résolution par la méthode de Householder est prévue pour résoudre des systèmes rectangulaires de taille $m \times n$ où $m \geq n$. Ce choix se trouve justifié dans le paragraphe ci-dessous.

b) la prise en compte des discontinuités caractérisée par les indicateurs $idfi$ et idd_i est répercutée lors de l'assemblage de (ou des) la matrice du système.

IV - 11

IV-C-2 Description du programme FRANIELS.

Ce programme d'implémentation plus facile que le précédent, reprend l'algorithme du paragraphe I-C. Il ne fonctionne que pour des surfaces possédant des discontinuités le long de segments rectilignes et ne permet pour l'instant que l'interpolation.

Il réalise les actions suivantes dans cet ordre:

- 1 - lecture des fichiers associés à une s.d. .
- 2 - Calcul des rayons d'influence R_W , R_Q des fonctions poids et des fonctions nodales respectivement. On demande si l'on veut entrer n_w, n_Q (nombre de points anticipés censés se trouver dans des disques de rayons R_W , R_Q). Par défaut, on prend $n_w = 9$ et $n_Q = 18$.
- 3 - Pour $k=1, \dots, n$ (où n est le nombre de points), on résoud le problème de moindres carrés pondéré associé par la méthode de HouseHolder suivant le schéma (NFF2).
- 4 - lecture des coordonnées des points extrémaux de la grille, ainsi que ses tailles p_x et p_y . L'évaluation de l'interpolant global est alors réalisé suivant les formules décrites au paragraphe I-C.
- 5 - On enregistre la s.d. RESULTAT conformément à ce qui a été écrit au début de ce paragraphe.

Remarque: a) la résolution des problèmes pondérés aux moindres carrés se ramène à celle d'un système linéaire rectangulaire $n \times 6$. Nous préférons cette méthode à celle de la décomposition de la matrice en valeurs singulières (décomposition sous la forme UDV, où U,V sont unitaires et D diagonale) pour des raisons d'implémentation et de temps d'exécution.

b) Dans le cas où moins de 5 points de donnée interviennent pour un problème de moindre carré, on remplace la partie quadrique de la fonction nodale associée par une partie linéaire.

IV - 12

IV-D Le logiciel VISUAL

Ce logiciel permet la visualisation d'une s.d. de type RESULTAT de deux façons:

- visualisation par courbes isovaleurs.
- visualisation tridimensionnelle.

La première représentation est réalisée par le programme Courniv, le second par le programme TRASurf. Ces deux modes de visualisation sont en fait complémentaires et permettent de mieux appréhender le comportement local de la surface.

Nous décrivons la structure générale de ces deux programmes ci-dessous.

IV-D-1 Le programme Courniv.

Ce programme est l'implémentation directe de l'algorithme décrit au chapitre III. Il réalise les actions suivantes dans cet ordre:

- 1 - lecture d'une s.d. RESULTAT
- 2 - lecture au clavier des valeurs extrémales et du nombre d'isovaleurs souhaité.
- 3 - tracé des isovaleurs.
- 4 - on propose une copie graphique du dessin sur listing.
- 5 - on propose une autre visualisation. Si oui, on retourne au point 1.

Remarque: a) les valeurs sont marquées sur les courbes afin d'améliorer la lisibilité.

b) une version de ce programme pour table traçante est en cours de réalisation.

c) une version simplifiée (i.e. pour des surfaces continues) a été intégrée dans la bibliothèque B.A.N.G. .

IV-D-2 Le programme TRASurf.

Ce programme convivial donne une représentation tridimensionnelle d'une surface connue par une s.d. de type RESULTAT, ou estimée par une fonction aux sommets d'une grille régulière. Elle utilise une version dérivée de la procédure graphique TRACESURFACE de la bibliothèque B.A.N.G. . La modification apportée est la possibilité de voir la surface d'une des quatre directions cardinales.

IV - 13

Le programme réalise les actions suivantes:

- 1 -** On demande si la surface est estimée à partir d'une fonction de deux variables ou provenant d'une s.d. RESULTAT. Si non, on va au point 3.
- 2 -** lecture de la s.d. RESULTAT. On va au point 4.
- 3 -** lecture du nombre de subdivisions sur chaque axe et des coordonnées extrémales de la grille. On évalue la surface aux sommets de cette grille.
- 4 -** lecture de la direction sous laquelle on souhaite visualiser la surface.
- 5 -** représentation tridimensionnelle sur l'écran.
- 6 -** on propose une copie graphique du dessin sur listing.
- 7 -** on propose une autre visualisation. Si oui, on retourne au point 1.

IV - 14

IV-E Le logiciel ERREUR

Ce logiciel est en fait un programme de calculs et de visualisation d'erreurs commises par l'un des programmes du logiciel **CALCUL** lorsque la surface est calculée à l'aide d'une fonction f de deux variables. Les erreurs sont calculées aux sommets d'une grille G $p_x \times p_y$. Nous donnons ci-dessous la structure du logiciel.

Notons $f_{i,j}$ et $s_{i,j}$ les valeurs de la fonction f et de l'approximation s au sommet (i,j) de la grille. Les paramètres d'erreurs calculées sont alors les suivants:

- la fonction erreur absolue discrète : $err_{i,j} = |s_{i,j} - f_{i,j}|$ $1 \leq i \leq p_x$, $1 \leq j \leq p_y$
- l'erreur absolue maximale discrète : $errabsmax = \max_{(i,j) \in G} err_{i,j}$
- l'erreur "relative" maximale discrète : $errrelmax = errmaxabs / (s_{max} - s_{min})$
où $s_{max} = \max_{(i,j) \in G} s_{i,j}$ et $s_{min} = \min_{(i,j) \in G} s_{i,j}$
- l'erreur absolue moyenne discrète : $errabsmoy = \left(\sum_{(i,j) \in G} err_{i,j} \right) / p_x.p_y$
- l'erreur "relative" moyenne discrète : $errelmoy = \left(\sum_{(i,j) \in G} err_{i,j} \right) / [(s_{max} - s_{min}).p_x.p_y]$
- la déviation standard $\sigma = \sqrt{\sum_{(i,j) \in G} (s_{i,j} - f_{i,j})^2 / p_x.p_y}$

1 - lecture d'une s.d. de type RESULTAT

2 - évaluation des différents paramètres décrits ci-dessus.

3 - visualisation du nuage de points pour lesquels $err_{i,j}$ est supérieure à une valeur donnée par l'utilisateur.

4 - on stocke la fonction erreur évaluée aux sommets de G via une s.d. de type RESULTAT. Ceci permet de visualiser ensuite la surface d'erreur grâce au logiciel **VISUAL**.

IV - 15

IV-F Quelques exemples.

Nous proposons comme première application une comparaison du fonctionnement entre la méthode d'inf-convolution spline d'interpolation (ICS) et les méthodes de Franke et Nielson (voir paragraphe I-C). L'exemple est un "benchmark" proposé par ces auteurs ([FRANKE 83]) est de fait une fonction (voir figure IV-1 en haut), admettant une discontinuité de valeur et de dérivée sur le segment infini rectiligne ($y = 0.4, x \geq 0.2$) et aussi une discontinuité de dérivée le long des segments semi-infini ($x = 0.1, y \geq 0.4$) et ($x = 0.2, y \geq 0.4$). Nous testons ici les méthodes d'inf-convolution spline d'interpolation et la méthode NFF1 sur deux jeux de donnée: un ensemble de 33 points et un ensemble de 130 points (figure IV-1 en bas). Les surfaces sont estimées sur une grille régulière 40×40 points recouvrant le carré $[0,1]^2$. De plus, pour la méthode ICS, nous prenons une seule fonction de base définie sur le segment $[(0.2, 0.4), (1.2, 0.4)]$. Nous ignorons ici les discontinuités des dérivées afin de mieux saisir le comportement de la surface le long du segment de discontinuité. On visualise les surfaces du côté est.

Les surfaces d'interpolation obtenues pour le premier jeu de donnée sont représentées sur la figure IV-2, celle du haut étant obtenue par NFF1, celle du bas par ICS. On voit que du côté amont de la faille, les deux surfaces ont un mauvais comportement ; ceci est dû à deux raisons : (i) on a peu de points de donnée, (ii) une seule fonction "traduisant" les discontinuités n'est visiblement pas suffisante pour retrouver le comportement local de la surface.

Le deuxième jeu de donnée donne de meilleurs résultats sur le plan visuel (figure IV-3); il subsiste néanmoins quelques oscillations pour la surface obtenue par ICS, toujours pour la deuxième raison évoquée ci-dessus. Par contre pour la surface NFF1, le comportement est meilleur près de la faille. Toutefois dans la partie haute, des oscillations indésirables apparaissent dues certainement au fait que les rayons d'influence ont été choisis trop petits.

Nous consignons les résultats d'erreurs ci-dessous.

ICS	Nombre de points	Taille de la grille	Errmaxabs	Errmoyrel	σ
Jeu 1	33	40x 40	0.121	0.016	0.052
Jeu 2	130	40x 40	0.067	0.005	0.015

NFF1	Nombre de points	Taille de la grille	Errmaxabs	Errmoyrel	σ
Jeu 1	33	40x 40	0.138	0.015	0.045
Jeu 2	130	40x 40	0.079	0.004	0.018

IV - 16

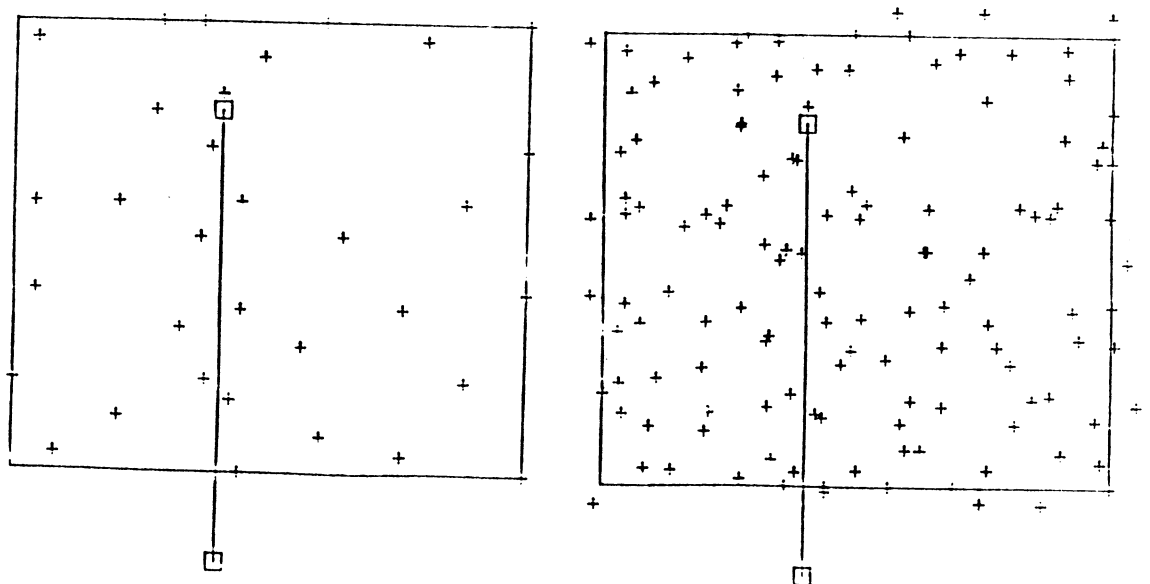
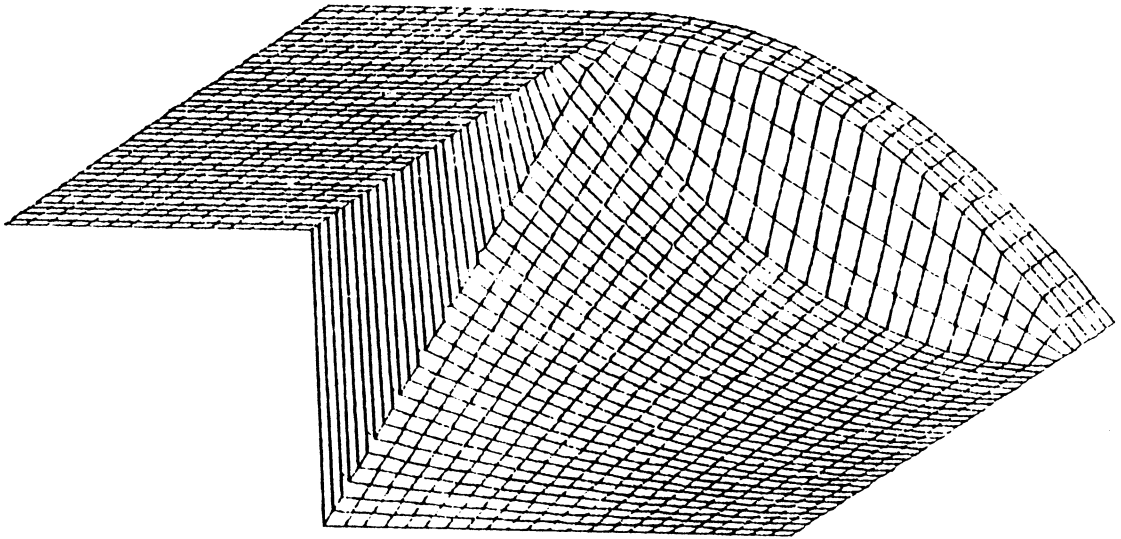


Figure IV-1

IV - 17

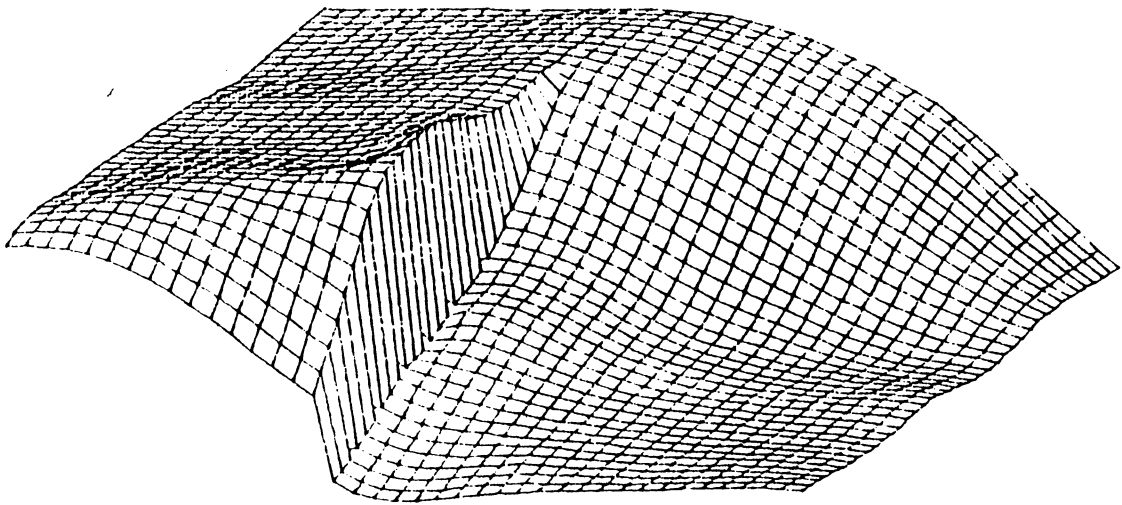
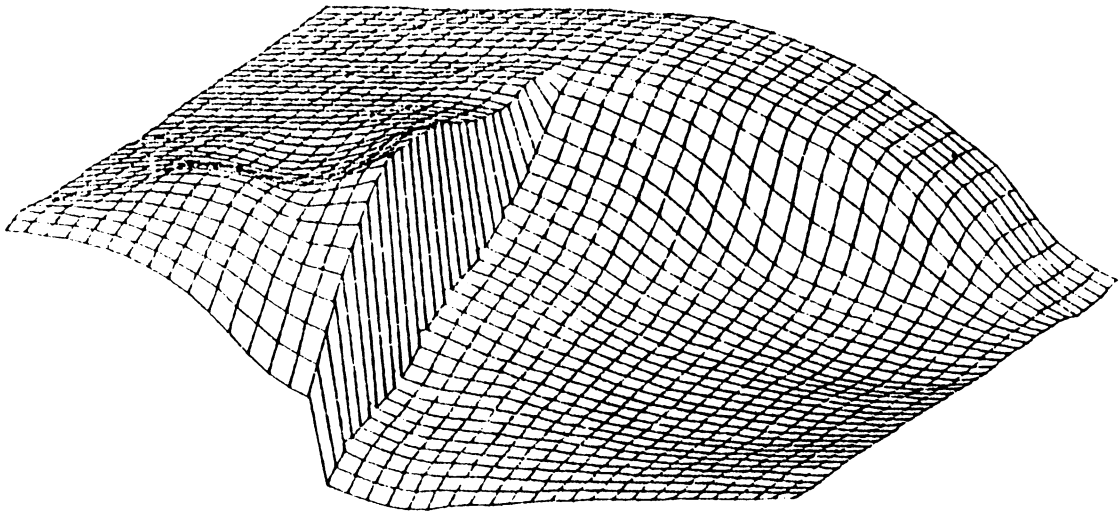


Figure IV-2

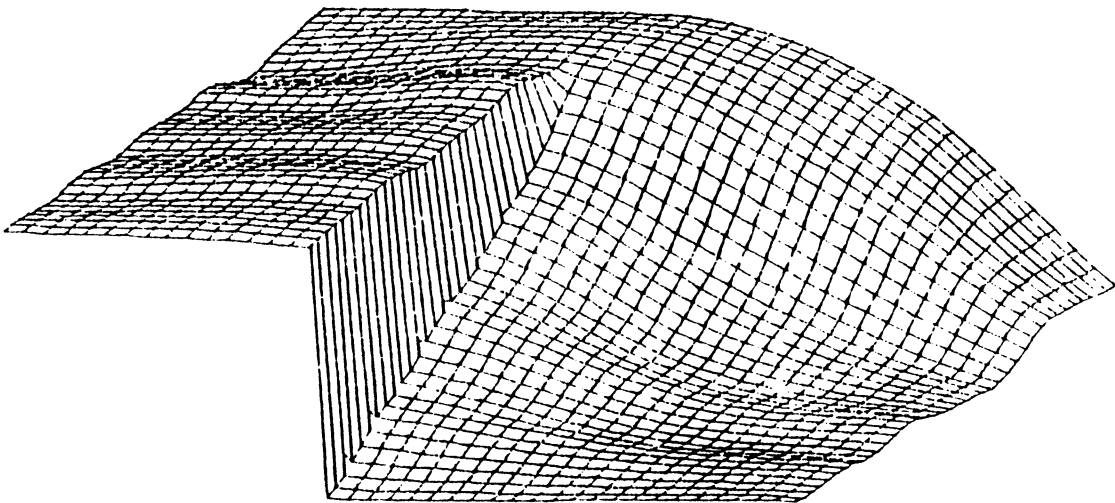
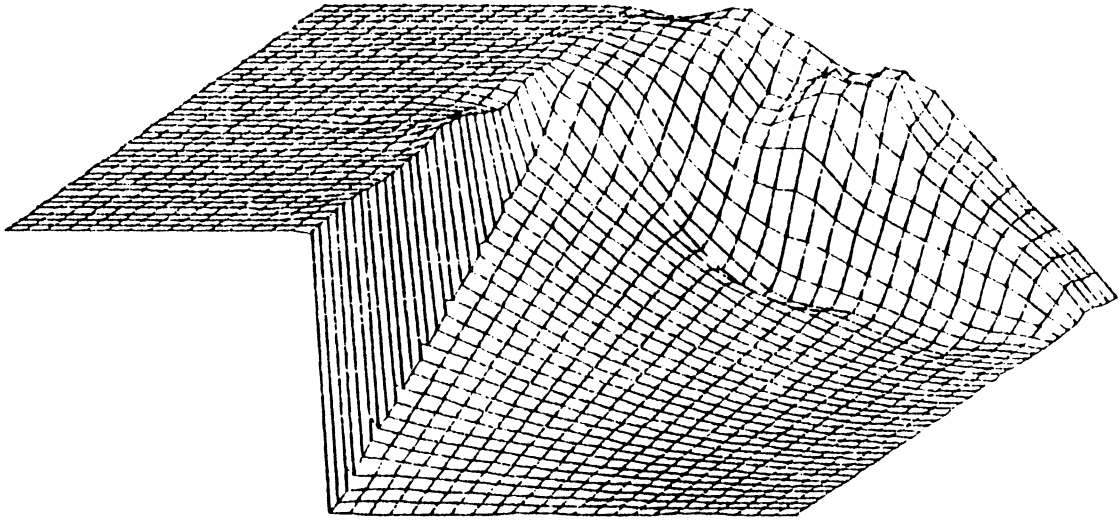


Figure IV-3

IV - 19

Ces résultats sont du même ordre de grandeur. Par contre, les temps d'exécution ne sont pas les mêmes: par exemple pour le jeu 2, le temps CPU est d'environ 5 minutes pour ICS, tandis qu'il dépasse 20 minutes pour NFF1 alors que pour le jeu 1 les temps sont du même ordre. Les erreurs de plus grande amplitude se situent non pas près de la faille, mais dans la partie en haut où il y a un fort gradient.

Le second exemple est une surface donnée par la fonction suivante:

$$z(x,y) = \begin{cases} 2.8 H_3(0.25 r(x,y)) + \sin(1.6 d_1(x,y)) / d_1(x,y) & \text{si } x \leq 5 \text{ et } y \leq 5 \\ 4 H_3(0.25 r(x,y)) + \sin(1.6 d_2(x,y)) / d_2(x,y) & \text{si } x \leq 5 \text{ et } y \geq 5 \\ H_3(0.25 r(x,y)) + \sin(1.6 d_3(x,y)) / d_3(x,y) & \text{si } x \geq 5 \text{ et } y \leq 5 \\ 4 H_3(0.25 r(x,y)) + \sin(1.6 d_4(x,y)) / d_4(x,y) & \text{si } x \geq 5 \text{ et } y \geq 5 \end{cases}$$

où $r(x,y) = \sqrt{(x-5)^2 + (y-5)^2}$, $H_3(r) = 1 - r^2(3 - 2r)$ si $0 \leq r \leq 1$, 0 sinon, et

$$d_1(x,y) = \sqrt{(x-2.5)^2 + (y-2.5)^2}$$

$$d_2(x,y) = \sqrt{(x-2.5)^2 + (y-7.5)^2}$$

$$d_3(x,y) = \sqrt{(x-7.5)^2 + (y-2.5)^2}$$

$$d_4(x,y) = \sqrt{(x-7.5)^2 + (y-7.5)^2}$$

Cette fonction (Voir figure IV-4 en haut) admet une discontinuité de valeur le long des segments $[(1,5),(9,5)]$ et $[(5,1),(5,5)]$. On se donne un jeu de 100 points équirépartis dans le rectangle $[0,10]^2$.

Nous regardons d'abord dans le cas de l'interpolation l'effet du paramètre θ intervenant dans la définition du semi-noyau reproduisant de l'espace $X^{2,s}(\mathbb{R}^2)$ (paragraphe II-2) ; θ est donné par la relation $\theta = 2 + 2s$ et le semi-noyau reproduisant est défini (à une constante multiplicative près) par :

$$K(t,t') = \begin{cases} \|t - t'\|^\theta \ln(\|t - t'\|) & \text{si } \theta \text{ est pair} \\ \|t - t'\|^\theta & \text{sinon} \end{cases}$$

Sur la figure IV-5, θ prend les valeurs 2,3,4 en allant de haut en bas. Les surfaces obtenues sont sensiblement les mêmes avec toutefois sur cet exemple, une légère amélioration lorsque θ croît. Ceci est d'ailleurs confirmé par le tableau suivant:

IV - 20

	Errmaxabs	Errmoyabs	σ
$\theta = 2$	0.09	0.026	0.035
$\theta = 3$	0.086	0.018	0.029
$\theta = 4$	0.081	0.015	0.026

La seconde série de tests consiste pour cette fonction à la "bruit" (ici avec un bruit "blanc" , d'écart -type 0.14). La figure IV-6 montre alors les différentes surfaces ICS pour différentes valeurs du paramètre d'ajustement ρ : $10^{-3}, 1, 10^3, 10^6, 10^9$. Les trois dernières valeurs donnent des surfaces visuellement ressemblantes. $\rho = 10^{-3}$ donne une surface proche de l'inf-convolution spline aux moindres carrés; tout le relief est écrasé sauf au voisinage des lignes de faille.

Une valeur raisonnable de ρ semble se trouver entre 1 et 10^3 . Pour déterminer la valeur optimale du paramètre d'ajustement, il faut ici recourir à la méthode de la validation croisée généralisée (en abrégé VCG) (voir [Girard 87]). Nous donnons la surface pour $r = 500$ sur la figure IV-7 qui, semble n'être pas "loin" de la surface obtenue par la méthode VCG.

IV - 21

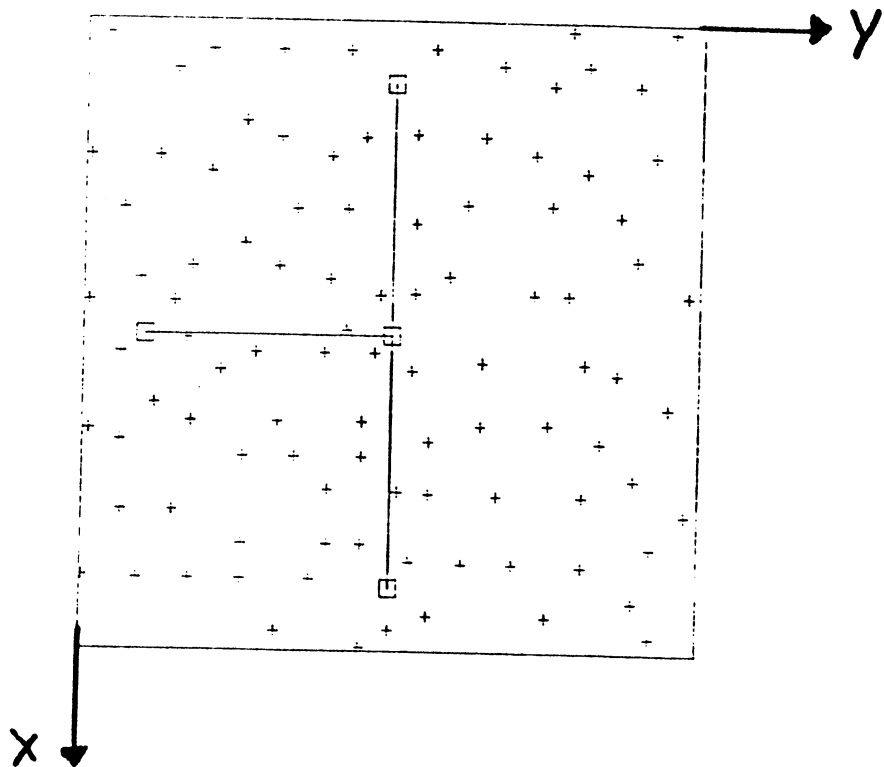
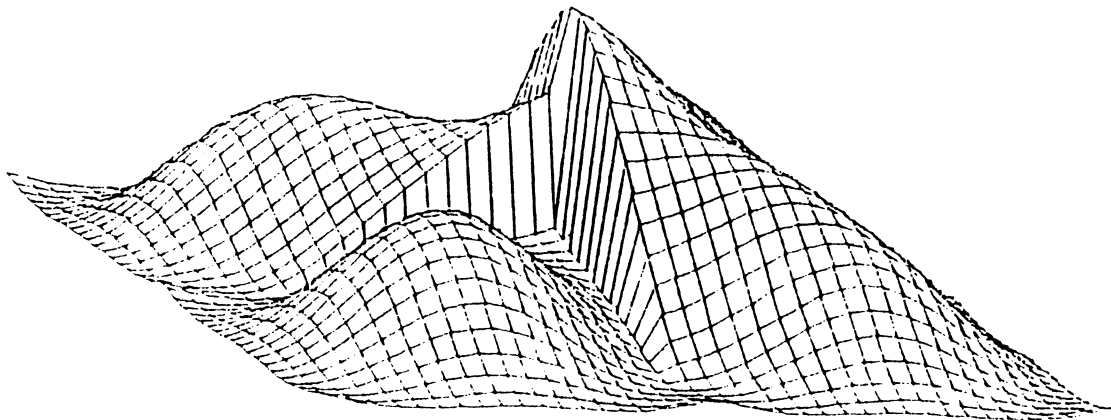
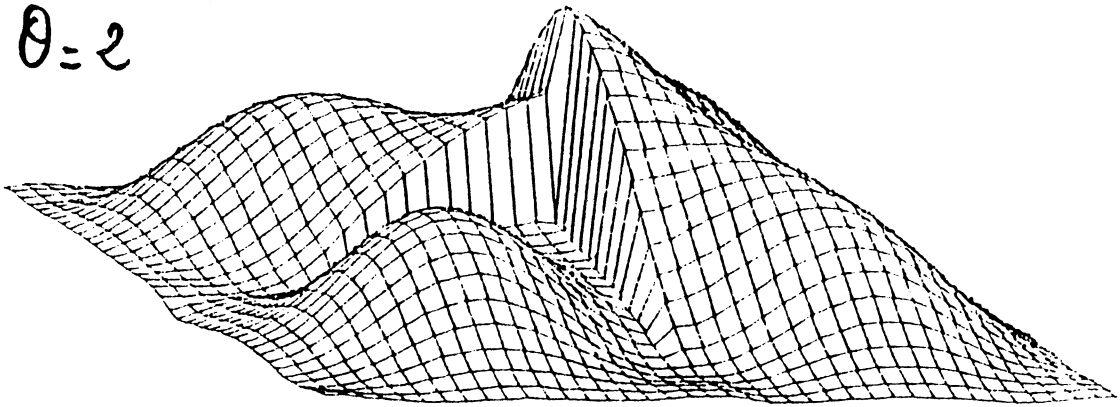


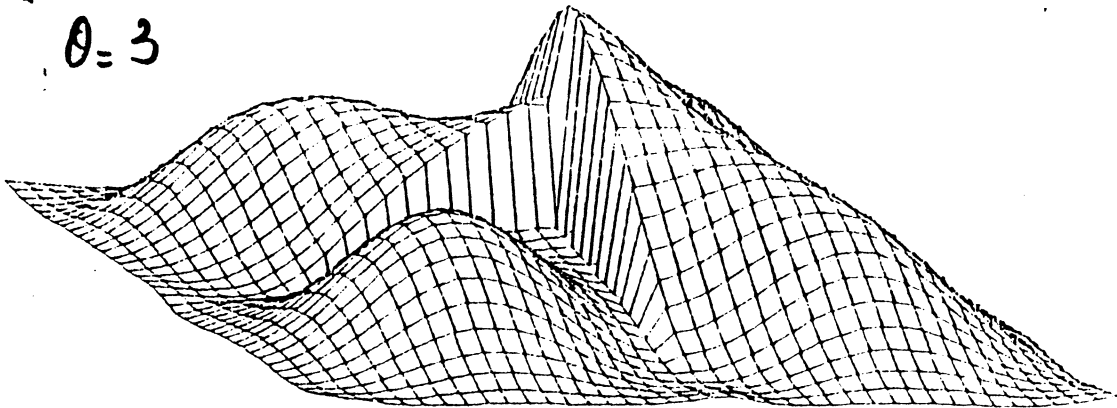
Figure IV-4

IV - 22

$\theta = 2$



$\theta = 3$



$\theta = 4$

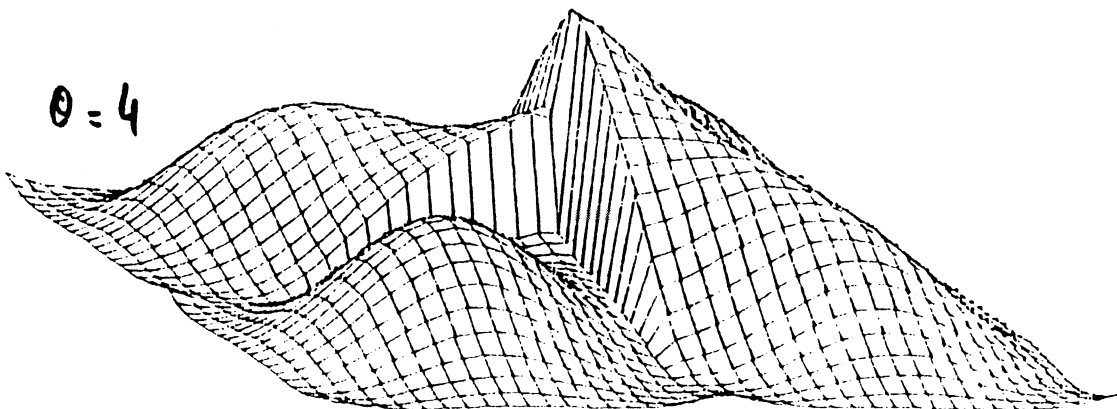
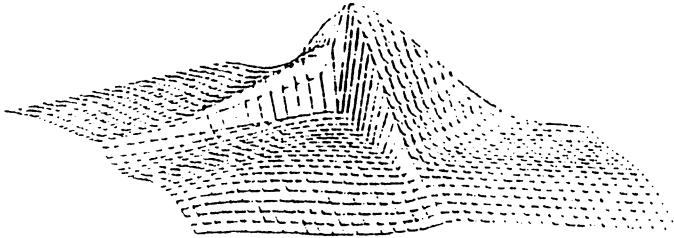


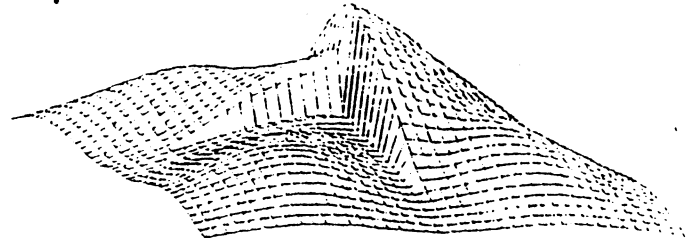
Figure IV-5

IV - 23

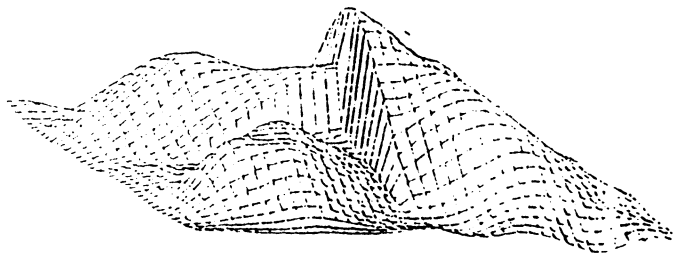
$$\rho = 10^{-3}$$



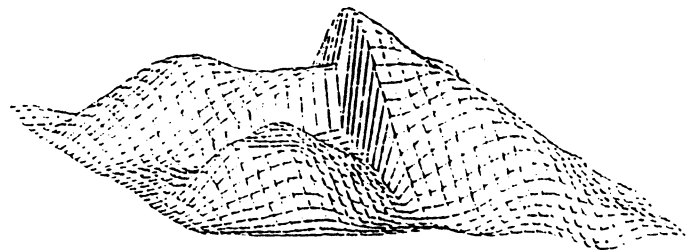
$$\rho = 1$$



$$\rho = 10^3$$



$$\rho = 10^6$$



$$\rho = 10^9$$

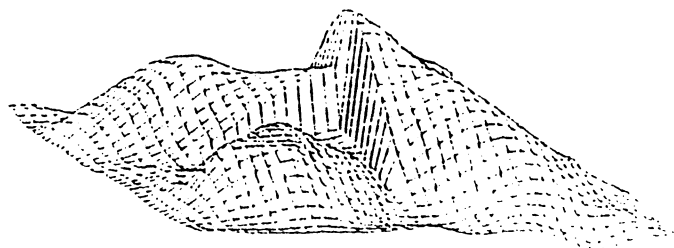


Figure IV-6

IV - 24

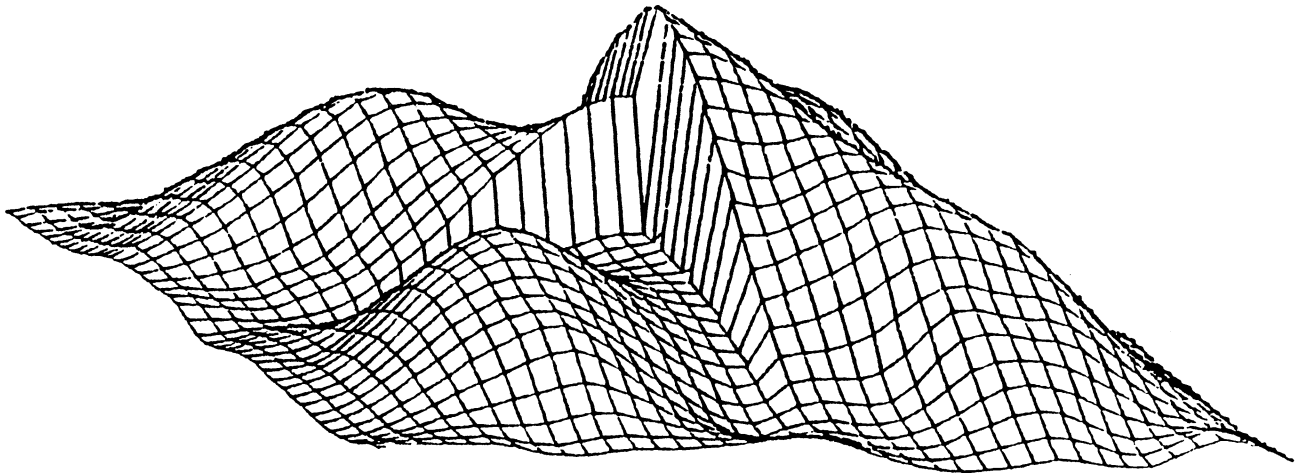


Figure IV-7

BIBLIOGRAPHIE

- [Akima 74]** Akima H.
 "A method of bivariate interpolation and smooth surface fitting based on local procedures"
 CACM vol 17 n° 1 Janvier 74
- [Akima 78]** Akima H.
 A method of bivariate interpolation and smooth surface fitting for irregularly distributed data"
 ACM TMS vol 4 n° 2 p148-159 1978
- [Arcangeli 85]** Arcangeli R., Aprato D., Manzanilla R.
 "On the construction of surfaces of class C^k "
 Preprint 1985 à paraître dans Siam JNA
- [Bolondi 76]** Bolondi G., Rocca F., Zanoletti S.
 "Automatic contouring of faulted subsurfaces"
 Geophysics vol 41 n° 6 p 1377-1393 décembre 76
- [Brandt 77]** Brandt A.
 "Multi-level adaptative solutions to boundary-value problems"
 Math. Comp. vol 31 1977a p333-390 1977
- [Briggs 74]** Briggs I.C.
 "Machine contouring using minimum curvature"
 Geophysics vol 39 n° 1 p39-48 Février 74
- [Butland 80]** Butland J.
 "A method of interpolation reasonable-shaped curves through any data"
 Proceedings Computer Graphics 80. On line publication, Middlesex UK1980 p 409-422 1980
- [Cottafava 69]** Cottafava G. , Le Moli G.
 "Automatic contour map"
 CACM vol 12 p386-391 juin 69
- [Duchon 80]** Duchon J.
 "Fonctions spline homogènes à plusieurs variables"
 Thèse d'état . Université de Grenoble . 1980
- [Franke 80]** Franke R. , Nielson G.
 "Smooth interpolation of large sets of scattered data"
 International Journal for numerical methods in engineering vol 15 p 1691-1704 1980

- [Franke 82] Franke R.
"Scattered data interpolation: tests of some methods"
Mathematics of computation vol 38 n° 157 Janvier 1982
- [Franke 83] Franke R. Nielson G.
"Surface approximation with imposed conditions"
Surfaces in CAGD . R.E. Barnhill W.E. Boehm (eds) p135-146
North Holland Publishing Company 1983
- [Geman 85] Geman S., Geman D.
"Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images."
IE³ Transactions Pattern Analysis and Machine Intelligence
PAMI6, p721-741 1985
- [Girard 87] Girard D.
"Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille."
Rapport de Recherche 669. Grenoble 1987
- [Gordon 78] Gordon W.J., Wixon J.A.
"Shepard's methods of metrics interpolation to bivariate and multivariate interpolation"
Mathematics of Computation vol 32 n° 141 Janvier 1978
- [Grimson 83] Grimson W.E.L.
"An implementation of a computational theory of visual surfaces interpolation"
Computer vision, graphics and image processing n° 22 p 39-69
- [Grimson 85] Grimson W.L., Pavlidis T.
"Discontinuity detection of visual reconstruction"
Computer vision, graphics and image processings n°30 p316-330 1985
- [Hardy 76] Hardy L.R.
"Multiquadrics equations of topography and other irregular surfaces"
Journal of Geophysical Research vol 27 n°8 p1905-1915 1976
- [Hessing 72] Hessing R.C., Lee H.K., Pierce A., Powers E.N.
"Automatic contouring by using bicubic functions"
Geophysics vol 37 n°4 p669-674 août 1972
- [Laurent 72] Laurent P.J.
"Approximation et optimisation"
Hermann 1972

- [Laurent 80] Laurent P.J.
"Spline functions using inf-convolution"
International Workshop on Approxiamtion theory, Technion, Haïfa, Israël
June 1,6 1980
- [Laurent 81] Laurent P.J.
"Inf-convolution spline pour l'approximation de données discontinues"
RR n° 270, IMAG, Université de Grenoble 1981
- [Laurent 85] Laurent P.J.
"Quadratic convex Analysis and Splines"
Proceedings of the International Conference, "Methods of Functional
Analysis in Approximation Theory", IIT Bombay, dec 16-20,1985,
Birkhäuser Verlag, ISNM76, p 17-43 1986
- [Laurent 86a] Laurent P.J.
"Inf-convolution spline pour l'approximation de données discontinues"
M²AN vol20 n°1 p 89-111 1986
- [Laurent 86b] Laurent P.J., Utreras F.I.
"Optimal smoothing of noisy broken data."
Approximation Theory and its Applications vol 2 n°2 Juin 1986
- [Mallet 74] Mallet L.
*"Présentation d'un ensemble de méthodes et de techniques de la
cartographie automatique numérique"*
Sciences de la Terre Série Informatique Géologique n°4,n°7 1974
- [Maréchal 83] Maréchal A.
"Kriging seismic data in presence of faults"
Rapport Elf-Aquitaine (Production) 1983
- [McLain 74] McLain D.H.
"Two-dimensional interpolation from scattered data"
Computer Journal vol 19 n° 2 Mars 1974 p 178-181 Mars 1974
- [McLain 76] McLain D.H.
"Drawing contours from arbitrary data points"
Computer Journal vol 17 n° 4 p318-324 1976
- [Manzanilla 86] Manzanilla R.
"Sur l'approximation de surfaces définies par une équation explicite"
Thèse de l'Université de Pau et des pays de l'Adour. 1986

- [Meinguet 84] Meinguet J.
"Surfaces Spline interpolation: basic theory and computational aspects"
S.P. Singh et al (eds) Approximation Theory and Splines Functions
p127-141 1984
- [Nielson 80] Nielson G.
"Minimum-norm interpolation in triangles"
SIAM JNA vol 17 n° 1 p 44-62
- [Paihua 78a] Paihua Montes L.
"Quelques méthodes numériques pour le calcul de fonctions spline à
une ou plusieurs variables."
Thèse. Université de Grenoble. 1978
- [Paihua78b] Paihua Montes L., Utreras Diaz F.
"Un ensemble de programmes pour l'interpolation des fonctions par
des fonctions spline de type plaque mince."
Rapport de Recherche n° 140 .Université de Grenoble 1978
- [Pelto 68] Pelto C.R., Elkins T.A., Boyd H.A.
"Automatic contouring of singularly spaced data"
Geophysics vol 33 n°3 p 424-430 Juin 68
- [Potier 83] Potier C., Vercken C.
"Approximation de fonctions de deux variables, applications en
cardiologie et en météorologie."
Thèse E.N.S.T. Avril 1983
- [Preparata 85] Preparata F.P., Shamos I.G.
"Computational Geometry: an introduction."
Springer Verlag 1985
- [Shepard 68] Shepard D.
"A two-dimensionnal interpolation from random data."
Proc. 23rd Nat. Conf. ACM p 517-523 1968
- [Strang 73] Strang G. , Fix G.J.
"An Analysis of the finite Element Methods"
Prentice-Hall, Englewoods Cliffs, NJ 1973
- [Terzopoulos 83] Terzopoulos D.
"Multilevel computational processes for visual surface reconstruction."
Computer Vision, Graphics and Image Processing n°24 p52-96 1983

[Terzopoulos 86a] Terzopoulos D.

"Image analysis using multigrids relaxation methods."

IE³ Transactions on Patterns Analysis and Machine Intelligence
vol PAMI-8 n° 2 p 129-139 Mars 86

[Terzopoulos 86b] Terzopoulos D.

"Regularization of inverse usual problems involving discontinuities."

IE³ Transactions on Patterns Analysis and Machine Intelligence
vol PAMI-8 n° 4 p 413-424 Juillet 86

[Uniras 86]

Geopak Interpolation

Geoint User's Manual 1986

[Utreras 79]

Utreras D. F.

"Utilisation de la méthode de validation croisée pour le lissage par
fonctions splines à une ou deux variables."

Thèse DI, Grenoble 1979

[Valera 84]

Valera Garcia D.

"Application de l'inf-convolution spline au traitement de chromatogram-
mes de gasoils"

Thèse D3. Université de Grenoble. 1984

[Walters 69]

Walters R.F.

"Contouring by machine: a user's guide"

The American Association of Petroleum Geologists Bulletin
vol 53 n°11 novembre 1969 p2324-2340

AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USTMG

Vu les dispositions de l'Arrêté du 16 avril 1974,

Vu les dispositions de l'Arrêté du 5 juillet 1984,

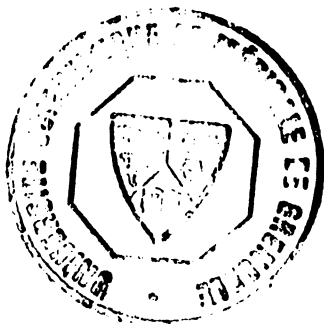
Vu les rapports de M..CHENIN....*Patrick*.....

M.....

M..KLEIN....*Pascal*..... est autorisé
à présenter une thèse en vue de l'obtention du *diplôme*...de.....
Docteur 3ème Cycle Mathématiques Appliquées

Grenoble, le.....*1987*.....

Le Président de l'Université Scientifique
Technologique et Médicale



J.J PAYAN

