HAL
open science

Quelques méthodes numériques pour le calcul de fonctions splines à une et plusieurs variables.

Luis Paihua Montes

To cite this version:

Luis Paihua Montes. Quelques méthodes numériques pour le calcul de fonctions splines à une et plusieurs variables.. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1978. Français. NNT : . tel-00288660v1

HAL Id: tel-00288660

https://theses.hal.science/tel-00288660v1

Submitted on 18 Jun 2008 (v1), last revised 19 Jun 2008 (v3)

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

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

THESE

présentée à

Université Scientifique et Médicale de Grenoble

Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR DE 3ème CYCLE

Analyse Numérique

par

Luis PAIHUA MONTES

QUELQUES METHODES NUMERIQUES POUR LE CALCUL DE FONCTIONS SPLINES A UNE ET PLUSIEURS VARIABLES

Thèse soutenue le 11 mai 1978 devant la Commission d'Examen :

Président : P.J. LAURENT

Examinateurs : M. ATTEIA
N. GASTINEL
M. DUC-JACQUET
J. THOMANN

I. M. A. G.
Laboratoire d'Informatique et de
Mathématiques Appliquées de Grenoble
U.S.M.G. - I.N.P.G. - C.N.R.S.
BIBLIOTHEQUE
BP 53 - 38041 Grenoble Cedex

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Monsieur Gabriel CAU : Président
Monsieur Joseph KLEIN : Vice-Président

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	AMBLARD Pierre	Clinique de dermatologie
	ARNAUD Paul	Chimie
	ARVIEU Robert	I.S.N.
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale A
	BEAUDOING André	Clinique de pédiatrie et puériculture
	BELORIZKY Elie	Physique
	BARNARD Alain	Mathématiques pures
Mme	BERTRANDIAS Françoise	Mathématiques pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques pures
	BEZES Henri	Clinique chirurgicale et traumatologie
	BLAMBERT Maurice	Mathématiques pures
	BOLLIET Louis	Informatique (I.U.T. B)
	BONNET Jean-Louis	Clinique ophtalmologie
	BONNET-EYMARD Joseph	Clinique hépato-gastro-entérologie
Mme	BONNIER Marie-Jeanne	Chimie générale
MM.	BOUCHERLE André	Chimie et toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BOUTET DE MONVEL Louis	Mathématiques pures
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique rhumatologique et hydrologique
	CALAS François	Anatomie
	CARLIER Georges	Biologie végétale
	CARRAZ Gilbert	Biologie animale et pharmacodynamie

.../...

MM.	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques pures
	CHARACHON Robert	Clinique ot-rhino-laryngologique
	CHATEAU Robert	Clinique de neurologie
	CHIBON Pierre	Biologie animale
	COEUR André	Pharmacie chimique et chimie analytique
	COUDERC Pierre	Anatomie pathologique
	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumophtisiologie
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DODU Jacques	Mécanique appliquée (I.U.T. I)
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	FONTAINE Jean-Marc	Mathématiques pures
	GAGNAIRE Didier	Chimie physique
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Analyse numérique
	GAVEND Michel	Pharmacologie
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	KAHANE André	Physique générale
	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	KRAVTCHENKO Julien	Mécanique
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
Mme	LAJZEROWICZ Janine	Physique
MM.	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LE ROY Philippe	Mécanique (.I.U.T. I)

.../...

MM.	LLIBOUTRY Louis	Géophysique
	LOISEAUX Jean-Marie	Sciences nucléaires
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LOUP Jean	Géographie
Mlle	LUTZ Elisabeth	Mathématiques pures
MM.	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Clinique cardiologique
	MAYNARD Roger	Physique du solide
	MAZARE Yves	Clinique Médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEGRE Robert	Mécanique
	NOZIERES Philippe	Spectrométrie physique
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET Jean	Séméiologie médicale (neurologie)
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REVOL Michel	Urologie
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-Chirurgie
	SARRAZIN Roger	Clinique chirurgicale B
	SEIGNEURIN Raymond	Microbiologie et hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction mécanique (I.U.T. I)
	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire
Mme	VERAIN Alice	Pharmacie galénique
MM.	VERAIN André	Physique biophysique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale

PROFESSEURS ASSOCIES

MM.	CRABBE Pierre	CERMO
	SUNIER Jules	Physique

PROFESSEURS SANS CHAIRE

Mlle	AGNIUS-DELORS Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Gilbert	Géographie
	BENZAKEN Claude	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique (I.U.T. I)
	BUISSON René	Physique (I.U.T. I)
	BUTEL Jean	Orthopédie
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique (I.U.T. I)
	DELOBEL Claude	M.I.A.G.
	DEPASSEL Roger	Mécanique des fluides
	GAUTRON René	Chimie
	GIDON Paul	Géologie et minéralogie
	GLENAT René	Chimie organique
	GROULADE Joseph	Biochimie médicale
	HACQUES Gérard	Calcul numérique
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et médecine préventive
	IDELMAN Simon	Physiologie animale
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
Mme	KAHANE Josette	Physique
MM.	KRAKOWIACK Sacha	Mathématiques appliquées
	KUHN Gérard	Physique (I.U.T. I)
	LUU DUC Cuong	Chimie organique - pharmacie
	MICHOULIER Jean	Physique (I.U.T. I)
Mme	MINIER Colette	Physique (I.U.T. I)

.../...

MM.	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	PFISTER Jean-Claude	Physique du solide
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	M.I.A.G.
	REBECQ Jacques	Biologie (CUS)
	REYMOND Jean-Charles	Chirurgie générale
	RICHARD Lucien	Biologie végétale
Mme	RINAUDO Marguerite	Chimie macromoléculaire
MM.	SARROT-REYNAULD Jean	Géologie
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
MM.	STIEGLITZ Paul	Anesthésiologie
	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	ARMAND Yves	Chimie (I.U.T. I)
	BACHELOT Yvan	Endocrinologie
	BARGE Michel	Neuro-chirurgie
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamie
MM.	BOST Michel	Pédiatrie
	BOUCHARLAT Jacques	Psychiatrie adultes
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BRODEAU François	Mathématiques (I.U.T. B) (Personne étrangère habilitée à être directeur de thèse)
	BERNARD Pierre	Gynécologie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie papetière
	CHIAVERINA Jean	Biologie appliquée (EFP)
	COLIN DE VERDIERE Yves	Mathématiques pures
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	CORDONNER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie

.../...

MM.	CYROT Michel	Physique du solide
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie végétale
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	FAURE Gilbert	Urologie
	GAUTIER Robert	Chirurgie générale
	GIDON Maurice	Géologie
	GROS Yves	Physique (I.U.T. I)
	GUIGNIER Michel	Thérapeutique
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	JALBERT Pierre	Histologie
	JUNIEN-LAVILLAVROY Claude	O.R.L.
	KOLODIE Lucien	Hématologie
	LE NOC Pierre	Bactériologie-virologie
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et médecine préventive
	MALLION Jean-Michel	Médecine du travail
	MARECHAL Jean	Mécanique (I.U.T. I)
	MARTIN-BOUYER Michel	Chimie (CUS)
	MASSOT Christian	Médecine interne
	NEMOZ Alain	Thermodynamique
	NOUGARET Marcel	Automatique (I.U.T. I)
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse)
	PEFFEN René	Métallurgie (I.U.T. I)
	PERRIER Guy	Géophysique-glaciologie
	PHELIP Xavier	Rhumatologie
	RACHALL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RAMBAUD Pierre	Pédiatrie
	RAPHAEL Bernard	Stomatologie
Mme	RENAUDET Jacqueline	Bactériologie (pharmacie)
MM.	ROBERT Jean-Bernard	Chimie-physique
	ROMIER Guy	Mathématiques (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse)
	SAKAROVITCH Michel	Mathématiques appliquées

.../...

MM.	SCHAERER René	Cancérologie
Mme	SEIGLE-MURANDI Françoise	Crytogamie
MM.	STOEBNER Pierre	Anatomie pathologie
	STUTZ Pierre	Mécanique
	VROUSOS Constantin	Radiologie

MAITRES DE CONFERENCES ASSOCIES

MM.	DEVINE Roderick	Spectro Physique
	KANEKO Akira	Mathématiques pures
	JOHNSON Thomas	Mathématiques appliquées
	RAY Tuhina	Physique

MAITRE DE CONFERENCES DELEGUE

M.	ROCHAT Jacques	Hygiène et hydrologie (pharmacie)

Fait à Saint Martin d'Hères, novembre 1977

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1979-1980

Président : M. Philippe TRAYNARD

Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	BENOIT Jean	Radioéléctricité
	BESSON Jean	Chimie Minérale
	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du Solide - Cristallographie
	BOIS Philippe	Mécanique
	BONNETAIN Lucien	Génie Chimique
	BONNIER Etienne	Métallurgie
	BOUVARD Maurice	Génie Mécanique
	BRISSONNEAU Pierre	Physique des Matériaux
	BUYLE-BODIN Maurice	Electronique
	CHARTIER Germain	Electronique
	CHERADAME Hervé	Chimie Physique Macromoléculaires
Mme	CHERUY Arlette	Automatique
MM.	CHIAVERINA Jean	Biologie, Biochimie, Agronomie
	COHEN Joseph	Electronique
	COUMES André	Electronique
	DURAND Francis	Métallurgie
	DURAND Jean-Louis	Physique Nucléaire et Corpusculaire
	FELICI Noël	Electrotechnique
	FOULARD Claude	Automatique
	GUYOT Pierre	Métallurgie Physique
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du Solide - Cristallographie
	LACOUME Jean-Louis	Géographie - Traitement du Signal
	LANCIA Roland	Electronique - Automatique
	LESIEUR Marcel	Mécanique
	LESPINARD Georges	Mécanique
	LONGEQUEUE Jean-Pierre	Physique Nucléaire Corpusculaire
	MOREAU René	Mécanique
	MORET Roger	Physique Nucléaire Corpusculaire
	PARIAUD Jean-Charles	Chimie - Physique
	PAUTHENET René	Physique du Solide - Cristallographie
	PERRET René	Automatique

.../...

MM.	PERRET Robert	Electrotechnique
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	POLOUJADOFF Michel	Electrotechnique
	POUPOT Christian	Electronique - Automatique
	RAMEAU Jean-Jacques	Chimie
	ROBERT André	Chimie Appliquée et des matériaux
	ROBERT François	Analyse numérique
	SABONNADIERE Jean-Claude	Electrotechnique
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie - Physique
Mme	SCHLENKER Claire	Physique du Solide - Cristallographie
MM.	TRAYNARD Philippe	Chimie - Physique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

CHERCHEURS DU C.N.R.S. (Directeur et Maître de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahïm	Maître de Recherche
	BRONOEL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KAMARINOS Georges	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique) E.N.S.E.E.G.

MM.	ALLIBERT Michel	
	BERNARD Claude	
	CAILLET Marcel	
Mme	CHATILLON Catherine	
MM.	COULON Michel	
	HAMMOU Abdelkader	
	JOUD Jean-Charles	
	RAVAINE Denis	
	SAINFORT	C.E.N.G.

.../...

MM. SARRAZIN Pierre
SOUQUET Jean-Louis
TOUZAIN Philippe
URBAIN Georges

Laboratoire des Ultra-Réfractaires ODEILLO

E.N.S.M.E.E.

MM. BISCONDI Michel
BOOS Jean-Yves
GUILHOT Bernard
KOBILANSKI André
LALAUZE René
LANCELOT Françis
LE COZE Jean
LESBATS Pierre
SOUSTELLE Michel
THEVENOT François
THOMAS Gérard
TRAN MINH Canh
DRIVER Julian
RIEU Jean

E.N.S.E.R.G.

MM. BOREL Joseph
CHEHIKIAN Alain
VIKTOROVITCH Pierre

E.N.S.I.E.G.

MM. BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
JAUSSAUD Pierre
Mme JOURDAIN Geneviève
MM. LEJEUNE Gérard
PERARD Jacques

E.N.S.H.G.

M. DELHAYE Jean-Marc

E.N.S.I.M.A.G.

MM. COURTIN Jacques
LATOMBE Jean-Claude
LUCAS Michel
VERDILLON André

à ma fille

ma femme

mes parents

J'exprime ma reconnaissance à Monsieur DUC-JACQUET qui a dirigé cette thèse ; son attention constante, ses nombreux conseils et encouragements m'ont été précieux dans l'accomplissement de ce travail.

Je témoigne ma gratitude à Monsieur LAURENT pour ses judicieuses remarques et l'honneur qu'il me fait en présidant ce Jury.

J'adresse tous mes remerciements à Monsieur GASTINEL qui m'a souvent utilement conseillé et pour avoir eu l'amabilité de participer au Jury.

Que soient aussi remerciés Messieurs ATTEIA et THOMANN d'avoir accepté de faire partie du Jury.

Je tiens aussi à remercier :

Toute l'Equipe d'Analyse Numérique et particulièrement Monsieur DUCHON, avec qui j'ai eu de fructueuses discussions, et qui m'a souvent encouragé.

Mme BICAIS qui, malgré un planning chargé, a assuré avec compétence la frappe de ce texte, ainsi que le service de reproduction pour la qualité de son travail.

PREMIERE PARTIE

SPLINE CUBIQUE

CHAPITRE I - LA METHODE DU TRANSPORT DES RELATIONS POUR LE CALCUL D'UNE FONCTION SPLINE D'INTERPOLATION (SPLINE CUBIQUE) CHOIX DE NORMALISATION

I.1 - INTRODUCTION ---------- 3
I.2 - PRESENTATION DES METHODES ---------- 5
I.3 - ETUDE DES SUITES DEFINIES PAR L'ALGORITHME DE LA METHODE (A) ---------- 8
I.4 - ALGORITHMES AVEC LES NORMALISATIONS SUGGEREES ---------- 11
I.5 - PAS EQUIDISTANTS (PAS H) ---------- 20
I.6 - COMPARAISON DES ALGORITHMES ---------- 25

CHAPITRE II - CALCUL DE LA FONCTION SPLINE D'INTERPOLATION POUR UN GRAND NOMBRE DE NOEUDS

II.1 - INTRODUCTION ---------- 31
II.2 - DECOUPAGE DE L'ENSEMBLE DES NOEUDS EN DEUX SOUS-ENSEMBLES DISJOINTS ---------- 31
II.3 - CALCUL DU VECTEUR CORRECTION Δ ---------- 36
II.4 - STRATEGIE SUGGEREE POUR TROUVER LA SPLINE POUR UN GRAND NOMBRE DE NOEUDS ---------- 44

CHAPITRE III - FONCTION SPLINE D'AJUSTEMENT D'ORDRE 2

III.1- INTRODUCTION ---------- 49
III.2- PRESENTATION DES ALGORITHMES ---------- 50
III.3- ETUDE DE LA METHODE (B) - CAS PAS EQUIDISTANTS ---------- 53
III.4- CALCUL DE LA SPLINE D'AJUSTEMENT AU MOYEN DE DEUX SPLINES D'AJUSTEMENT LOCALES ---------- 58

REFERENCES ---------- 72

DEUXIEME PARTIE

ALGORITHMES D'OBTENTION NUMERIQUE DES FONCTIONS SPLINE A DIMENSION $d \geq 2$

CHAPITRE I - FONCTION SPLINE DU TYPE PLAQUE MINCE EN DIMENSION 2

I.1 - INTRODUCTION ---------- 78
I.2 - QUELQUES PROPRIETES DE LA MATRICE K ---------- 81
I.3 - METHODES NUMERIQUES ---------- 84
I.4 - FONCTION SPLINE D'AJUSTEMENT DU TYPE PLAQUE MINCE ----- 112
I.5 - RESULTATS NUMERIQUES ET CONCLUSIONS ---------- 113

CHAPITRE-II - ADAPTATION DES METHODES DECRITES PRECEDEMMENT AUX FONCTIONS SPLINE INVARIANTES PAR ROTATION EN DIMENSION $d \geq 2$

II.1 - INTRODUCTION ---------- 127
II.2 - CALCUL DE LA FONCTION SPLINE D'INTERPOLATION ET AJUSTEMENT AU MOYEN DES METHODES DECRITES AU CHAPITRE I ----- 130

CHAPITRE-III - CONSTRUCTION D'UNE SURFACE D'INTERPOLATION POUR UN GRAND NOMBRE DE NOEUDS D'INTERPOLATION

III.1- INTRODUCTION ---------- 136
III.2- CONSTRUCTION DE S POUR LE RECTANGLE CITÉ ---------- 137
III.3- CONSTRUCTION DE S POUR UN NOMBRE DE POINTS ASSEZ GRAND- 143
III.4- RESULTATS NUMERIQUES ET CONCLUSIONS ---------- 147

REFERENCES ---------- 166

PREMIERE PARTIE

SPLINE CUBIQUE

INTRODUCTION

Les fonctions spline à une variable, sont actuellement très utilisées, et la théorie a été largement développée, voir par exemple [1], [2], [3], [4], [13],[17], [20], [21].

La stabilité numérique, de la méthode utilisée pour l'obtention de ces fonctions pour n (nombre de noeuds) grand, est l'un des problèmes importants. Voir [5], [9], [10], [14], [15], [22],[23].

Dans cette partie nous nous intéressons à l'étude de deux méthodes :

- Méthode du transport des relations présentée en [12] (méthode A)
- Méthode présentée en [7] (méthode B).

Le premier chapitre est consacré à l'étude détaillée de l'algorithme de la méthode A, avec comme objectif principal l'obtention d'autres algorithmes qui soient moins coûteux, mais aussi stable que celui donné en [16]. De plus on étudie la similitude qui existe entre les algorithmes des méthodes A et B.

Le deuxième chapitre a pour but le calcul numérique de la fonction spline d'interpolation pour un grand nombre de noeuds. D'abord on présente la relation qui existe entre la "spline globale" (relative à l'ensemble des noeuds) et deux "splines locales" (splines construites) sur des sous-ensembles de noeuds. On suggère ensuite une méthode d'obtention numérique de la spline cherchée, à l'aide précisément de splines locales.

Dans le troisième chapitre on étudie l'obtention d'une spline d'ajustement au moyen des deux méthodes A et B.

CHAPITRE - I

LA METHODE DU TRANSPORT DES RELATIONS POUR LE CALCUL D'UNE FONCTION SPLINE D'INTERPOLATION (SPLINE CUBIQUE)

CHOIX DE NORMALISATIONS

I.1 - INTRODUCTION

Soient f la fonction à interpoler définie sur [a,b]

$x_i \in [a,b] \quad i = 1,\dots,n$ et $a < x_1 < \dots < x_n < b$

S_n la spline d'interpolation cubique basée sur les noeuds $x_1,\dots,x_n$.
Notons :

$$f_i = f(x_i)$$

$$\lambda_i = S_n'(x_i) \qquad \beta_i = S_n''(x_i) \qquad \gamma_i^+ = S_n'''(x_i^+) \qquad i = 1,\dots,n$$

$$h_i = x_{i+1}-x_i \qquad m_i = \frac{f_{i+1}-f_i}{h_i} \qquad i = 1,\dots,n-1$$

La méthode du transport des relations (voir [12]) qu'on appellera METHODE (A) construit les suites $\{a_i\}$, $\{b_i\}$, $\{c_i\}$, $\{\tilde{a}_i\}$, $\{\tilde{b}_i\}$, $\{\tilde{c}_i\}$ $i = 1,\dots,n$, qui sont les coefficients du système suivant :

$$(1.1)\begin{cases} a_i\lambda_i + b_i\beta_i = c_i \\ \tilde{a}_i\lambda_i + \tilde{b}_i\beta_i = \tilde{c}_i \qquad i = 1,\dots,n \end{cases}$$

La particularité de ces suites, qui sont obtenues de façon récurrente (voir plus loin) est qu'elles croissent assez vite, et leur calcul est difficile et imprécis.

En conséquence il est recommandé de normaliser les deux relations linéaires de (1.1) à chaque pas, c'est-à-dire de les diviser pour un nombre convenable de telle manière que les nouvelles suites $\{\frac{a_i}{t_i}\}$, $\{\frac{b_i}{t_i}\}$, $\{\frac{c_i}{t_i}\}$ (resp. $\{\frac{\tilde{a}_i}{\tilde{t}_i}\}$, $\{\frac{\tilde{b}_i}{\tilde{t}_i}\}$, $\{\frac{\tilde{c}_i}{\tilde{t}_i}\}$), restent bornées

La normalisation adoptée dans [12] (appelée usuelle) est la plus naturelle mais aussi la plus coûteuse. Elle consiste à prendre $t_i = \sqrt{a_i^2+b_i^2+c_i^2}$ (resp. $\tilde{t}_i^2 = \sqrt{\tilde{a}_i^2+\tilde{b}_i^2+\tilde{c}_i^2}$) ; cette normalisation donne des résultats numériques satisfaisants même pour n grand [16].

En étudiant le comportement des suites $\{a_i\}$ et $\{b_i\}$ (resp. $\{\tilde{a}_i\}$ et $\{\tilde{b}_i\}$) on trouvera une relation importante entre elles, ce qui va nous permettre d'étudier deux autres normalisations qui diminuent le coût de la méthode (A).

De plus, on met en évidence qu'avec la deuxième normalisation proposée l'algorithme de la méthode (A) devient l'algorithme de la méthode présenté en [7] (appelée METHODE (B)).

Pour le cas particulier de "Pas équidistants" les simpliciations dans l'algorithme de la méthode (A) sont importantes pour les deux normalisations suggérées et on étudie ces simplifications.

Enfin on donne des tableaux comparatifs des algorithmes obtenus avec les différentes normalisations.

I-2 - PRESENTATION DES METHODES

Dans la méthode (A) on considère S_n sous la forme :

$$S_n(x) = \begin{cases} f_1 + \lambda_1(x-x_1) & \text{si } x \in [a,x_1] \\ f_i + \lambda_i(x-x_i) + \frac{\beta_i}{2}(x-x_i)^2 + \frac{\gamma_i^+}{6}(x-x_i)^3 & x \in [x_i,x_{i+1}] \quad i=1,\dots,n-1 \\ f_n + \lambda_n(x-x_n) & \text{si } x \in [x_n,b] \end{cases}$$

Si on note $v(x) = [S_n(x), S_n'(x), S''(x), S'''(x)]^T$ on constate que :

$$(1.2) \qquad \begin{cases} v(x_{i+1}^-) = a(h_i)v(x_i^+) \\ v(x_i^+) = a(-h_i)v(x_{i+1}^-) \end{cases}$$

où

$$a(h) = [a_{kj}^h] \qquad a_{kj}^h = \begin{cases} \frac{h^{j-k}}{(j-k)!} & j \geq k \quad k = 1,\dots,4 \\ 0 & j < k \quad j = k,\dots,4 \end{cases}$$

La méthode (A) est basée sur les deux remarques suivantes (voir [12])

a) Si l'on connaît deux relations linéaires indépendantes en x_i^- et si ces relations sont aussi indépendantes de la relation $f_i = S_n(x_i)$ alors on peut trouver deux relations linéaires en x_i^+.

b) Si l'on connait deux relations linéaires en x_i^+ alors on peut "transporter" ces relations en x_{i+1}^- en utilisant (1.2).

La méthode consiste à appliquer alternativement les phases (a) et (b) précédentes partant de

i) x_1^- avec $S_n''(x_1^-) = S_n'''(x_1^-) = 0$ (de Gauche à Droite)

ii) x_n^+ avec $S_n''(x_n^+) = S_n'''(x_n^+) = 0$ (de Droite à Gauche)

Simplifiant les deux relations de chaque point x_i on obtient une relation linéaire entre λ_i et β_i, donné par les équations de récurrences suivantes :

De Gauche à Droite

$$(1.3)\quad \begin{cases} a_i = 6\,\dfrac{b_{i-1}}{h_{i-1}} - 2a_{i-1} \\ b_i = a_{i-1}\,\dfrac{h_{i-1}}{2} - 2b_{i-1} \\ c_i = c_{i-1} + m_{i-1}\left(6\,\dfrac{b_{i-1}}{h_{i-1}} - 3a_{i-1}\right) \qquad i = 2,\dots,n \end{cases}$$

avec

$$a_1 = 0 \qquad b_1 = 1 \qquad c_1 = 0$$

et

$$\boxed{a_i\lambda_i + b_i\beta_i = c_i}$$

De Droite à Gauche

$$(1.3')\quad \begin{cases} \tilde{a}_i = -6\,\dfrac{\tilde{b}_{i+1}}{h_i} - 2\,\tilde{a}_{i+1} \\ \tilde{b}_i = -\tilde{a}_{i+1}\,\dfrac{h_i}{2} - 2\tilde{b}_{i+1} \\ \tilde{c}_i = \tilde{c}_{i+1} - m_i\left(6\,\dfrac{\tilde{b}_{i+1}}{h_i} + 3\,\tilde{a}_{i+1}\right) \qquad i = n-1,\dots,1 \end{cases}$$

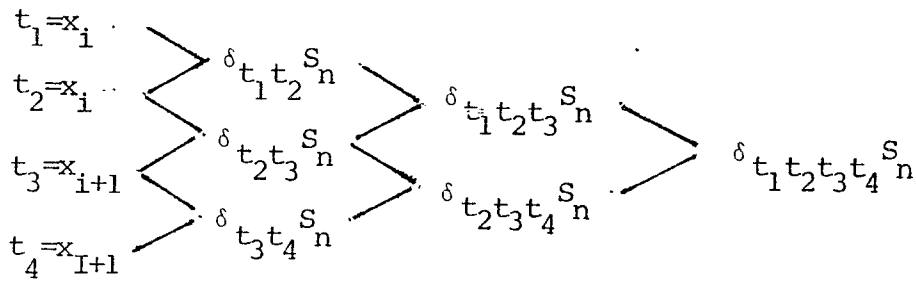
avec

$$\tilde{a}_n = 0 \qquad \tilde{b}_n = 1 \qquad \tilde{c}_n = 0$$

et

$$\boxed{\tilde{a}_i\lambda_i + \tilde{b}_i\beta_i = \tilde{c}_i}$$

La méthode (B) considère S_n sous la forme de Newton avec noeuds confondus (voir [7])



Alors on a :

$$S_n(x) = \begin{cases} f_1+\lambda_1(x-x_1) & \text{si } x \in [a,x_1] \\ f_i+\lambda_i(x-x_i)+\dfrac{m_i-\lambda_i}{h_i}(x-x_i)^2+\dfrac{\lambda_{i+1}-2m_i+\lambda_i}{h_i^2}(x-x_i)^2(x-x_{i+1}) & \\ & \text{si } x \in [x_i,x_{i+1}] \quad i=1,\dots,n \\ f_n+\lambda_n(x-x_n) & \text{si } x \in [x_n,b] \end{cases}$$

Les inconnues λ_i $i = 1,\dots,n$ sont solutions du système suivant, obtenu en assurant la continuité de S_n'' en x_i

$$(1.4) \qquad \begin{cases} 2\lambda_1+\lambda_2 = 3m_1 \\ \dfrac{h_i}{h_{i-1}}\lambda_{i-1}+2(1+\dfrac{h_i}{h_{i-1}})\lambda_i+\lambda_{i+1} = 3(m_i+\dfrac{h_i}{h_{i-1}}m_{i-1}) \quad . \quad i = 2,\dots,n-1 \\ \lambda_{n-1}+2\lambda_n = 3m_{n-1} \end{cases}$$

Le système est résolu par l'élimination simple de Gauss ([16][31]).

Notons $\lambda_i' = \lambda_i/3 \quad i = 1,\dots,n$

Alors : $\lambda_n' = r_n/d_n$

$\lambda_i' = (r_i-\lambda_{i+1}')/d_i \qquad i = n-1,\dots,1$

où :

$$(1.5) \qquad \begin{cases} d_1 = 2 & r_1 = m_1 \\ d_i = \dfrac{h_i}{h_{i-1}}(2-\dfrac{1}{d_{i-1}})+2 & r_i = \dfrac{h_i}{h_{i-1}}(m_{i-1}-\dfrac{r_{i-1}}{d_{i-1}})+m_i \qquad i=2,\dots,n-1 \\ d_n = 2-\dfrac{1}{d_{n-1}} & r_n = m_{n-1}-\dfrac{r_{n-1}}{d_{n-1}} \end{cases}$$

On a constaté que cette méthode (B) donne des résultats numériques satisfaisants comme la méthode (A). (voir [16]).

I.3 - ETUDE DES SUITES DEFINIES PAR L'ALGORITHME DE LA METHODE (

Proposition 1

Dans le procédé (1.3) (resp. (1.3') on a :

a) $\begin{cases} (-1)^i a_i > 0 \\ (-1)^i b_i < 0 \end{cases}$ (resp. $\begin{cases} (-1)^i \tilde{a}_{n-i} > 0 \\ (-1)^i \tilde{b}_{n-i} > 0 \end{cases}$)

b) $|a_i|$ et $|b_i|$ (resp. $|\tilde{a}_{n-i}|$ et $|\tilde{b}_{n-i}|$) croissent quand i croît.

Démonstration :

a) Comme $a_2 = 6/h_1$ et $b_2 = -2$ on a d'après (1.3)

$$\begin{cases} a_i > 0 \\ b_i < 0 \end{cases} \text{ pour i pair alors } \begin{cases} a_{i+1} < 0 \\ b_{i+1} > 0 \end{cases}$$

$$\begin{cases} a_i < 0 \\ b_i > 0 \end{cases} \text{ pour i impair alors } \begin{cases} a_{i+1} > 0 \\ b_{i+1} < 0 \end{cases}$$

d'où le résultat. Démonstration analogue pour les $\tilde{a}_i$, $\tilde{b}_i$ à partir de (1.3').

b) On développe a_i et b_i d'après (1.3), alors on a :

$$\begin{cases} a_i = \sum_{j=1}^{i-1} (-1)^{j+1} \, 3 \times \frac{2^j}{h_{i-j}} b_{i-j} \\ b_i = -2b_{i-1} + \sum_{j=2}^{i-1} (-1)^j \, 3 \times 2^{j-2} \, \frac{h_{i-1}}{h_{i-j}} b_{i-j} \end{cases}$$

D'autre part, la partie (a) implique :

$$\begin{cases} |a_i| = (-1)^i a_i \\ |b_i| = (-1)^{i+1} b_i \\ (-1)^{i+j+1} b_{i-j} = (-1)^{i-j+1} b_{i-j} > 0 \end{cases}$$

utilisant ces relations dans l'expression de a_i rt b_i on obtient :

$$\begin{cases} |a_i| = \sum_{j=1}^{i-1} (-1)^{i+j+1}\, 3 \times \frac{2^j}{h_{i-j}}\, b_{i-j} \\ |b_i| = (-1)^i 2b_{i-1} + \sum_{j=2}^{i-1} (-1)^{i+j+1} 3 \times 2^{j-2} \frac{h_{i-1}}{h_{i-j}}\, b_{i-j} \end{cases}$$

on a donc

$$\begin{cases} |a_i| = \sum_{j=1}^{i-1} 3 \times \frac{2^j}{h_{i-j}}\, |b_{i-j}| \\ |b_i| = 2|b_{i-1}| + \sum_{j=2}^{i-1} 3 \times 2^{j-2} \frac{h_{i-1}}{h_{i-j}}\, |b_{i-j}| \end{cases}$$

ce qui montre la partie (b).

Remarque :

1) Pour le procédé (1.3') on utilisera

$$\begin{cases} \tilde{a}_i = \sum_{j=1}^{n-i-1} (-1)^j\, 3 \times \frac{2^j}{h_{i+j-1}}\, \tilde{b}_{i+j} \\ \tilde{b}_i = -2\tilde{b}_{i+1} + \sum_{j=2}^{n-i-1} (-1)^{j+1} 3 \times 2^{j-2} \frac{h_i}{h_{i+j-1}}\, \tilde{b}_{i+j} \end{cases}$$

2) Cette proposition montre que l'on ne doit pas utiliser les relations (1.3) (resp. (1.3')) pour calculer les suites $\{a_i\}$, $\{b_i\}$ (resp. $\{\tilde{a}_i\}$, $\{\tilde{b}_i\}$).

Proposition 2

Dans le procédé (1.3) (resp. (1.3')) on a :

$$2 < \left|\frac{a_i\, h_{i-1}}{b_i}\right| < 4 \text{ (resp. } 2 < \left|\frac{\tilde{a}_i h_i}{b_i}\right| < 4)$$

Démonstration :

Comme $b_i = a_{i-1}\frac{h_{i-1}}{2} - 2b_{i-1}$ on a alors $a_{i-1} = \frac{2b_i + 4b_{i-1}}{h_{i-1}}$

remplaçant la valeur de a_{i-1} dans $a_i = \frac{6b_{i-1}}{h_{i-1}} - 2a_{i-1}$ on obtient

$a_i = -\frac{1}{h_{i-1}}(4b_i + 2b_{i-1})$ ce qui implique:

$$\frac{a_i}{b_i} = -\frac{4}{h_{i-1}}\left(1 + \frac{b_{i-1}}{2b_i}\right)$$

Mais la proposition (1) implique :

$$\left|\frac{a_i}{b_i}\right| = -\frac{a_i}{b_i} \quad \text{et} \quad -1 < \frac{b_{i-1}}{b_i} < 0$$

alors

$$\frac{1}{2} < 1 + \frac{b_{i-1}}{2b_i} < 1 \qquad \text{ce qui entraine}$$

$$\frac{2}{h_{i-1}} < -\frac{a_i}{b_i} < \frac{4}{h_{i-1}}.$$

Ce qui est le résultat annoncé. Pour le procédé (1.3') la démonstration est analogue.

La proposition ci-dessus exprime la relation qui existe entre les suites $\{a_i\}$ et $\{b_i\}$ (resp. $\{\tilde{a}_i\}$ et $\{\tilde{b}_i\}$) et nous donne deux idées sur la façon de normaliser en tenant compte du coût des calculs et l'utilisation des mémoires.

Nous suggérons les normalisations suivantes :

NOR-1 : Diviser par a_i (resp. $\tilde{a}_i$) à chaque itération

NOR-2 : Diviser par $-\frac{2b_i}{h_i}$ (resp. $-\frac{2\tilde{b}_i}{h_i}$) à chaque itération.

Remarque :

La deuxième normalisation a été choisie pour démontrer la similitude qui existe dans les algorithmes de la méthode (A) et (B).

I.4 - ALGORITHMES AVEC LES NORMALISATIONS SUGGEREES

NORMALISATION 1 (NOR-1)

L'équation $a_i\lambda_i+b_i\beta_i = c_i$ $i = 2,\dots,n$ est équivalente à $\lambda_i+(\frac{b_i}{a_i})\beta_i = (\frac{c_i}{a_i})$ et notons $p_i = \frac{b_i}{a_i}$ $q_i = \frac{c_i}{a_i}$, de la même façon $\tilde{a}_i\lambda_i+\tilde{b}_i\beta_i = \tilde{c}_i$ $i = n-1,\dots,1$ est équivalent à $\lambda_i+(\frac{\tilde{b}_i}{\tilde{a}_i})\beta_i = (\frac{\tilde{c}_i}{\tilde{a}_i})$ et soit $\tilde{p}_i = \frac{\tilde{b}_i}{\tilde{a}_i}$ $\tilde{q}_i = \frac{\tilde{c}_i}{\tilde{a}_i}$.

Ensuite on explicite les équations de récurrence qui permettent de trouver p_i, q_i, $\tilde{p}_i$, $\tilde{q}_i$.

Evidemment $a_2 = \frac{6}{h_1}$ $b_2 = -2$, $c_2 = m_1\frac{6}{h_1}$ implique $p_2 = -h_1/3$ $q_2 = m_1$

maintenant :

$$p_i = \frac{b_i}{a_i} = \frac{b_i/a_{i-1}}{a_i/a_{i-1}} = \frac{\frac{h_{i-1}}{2} - 2(\frac{b_{i-1}}{a_{i-1}})}{\frac{6}{h_{i-1}}(\frac{b_{i-1}}{a_{i-1}} - 2)}$$

$$\boxed{p_i = \frac{\frac{h_{i-1}}{2} - 2\,p_{i-1}}{\frac{6}{h_{i-1}}(p_{i-1}-2)}} \qquad i = 3,\dots,n$$

$$q_i = \frac{c_i}{a_i} = \frac{c_i/a_{i-1}}{a_i/a_{i-1}} = \frac{\frac{c_{i-1}}{a_{i-1}} + m_{i-1}(\frac{6\,b_{i-1}}{h_{i-1}a_{i-1}} - 3)}{\frac{6}{h_{i-1}}(p_{i-1}-2)}$$

$$\boxed{q_i = \frac{q_{i-1}+m_{i-1}(6\frac{p_{i-1}}{h_{i-1}} - 3)}{\frac{6}{h_{i-1}}(p_{i-1}-2)}} \qquad i = 3,\dots,n$$

$$\tilde{a}_{n-1} = -\frac{6}{h_{n-1}}, \quad \tilde{b}_{n-1} = -2, \quad \tilde{c}_{n-1} = -m_{n-1}\frac{6}{h_{n-1}} \quad \text{implique}$$

$$\tilde{p}_{n-1} = h_{n-1}/3 \qquad \tilde{q}_{n-1} = m_{n-1}$$

et

$$\tilde{p}_i = \frac{\tilde{b}_i}{\tilde{a}_i} = \frac{\tilde{b}_i/\tilde{a}_{i+1}}{\tilde{a}_i/\tilde{a}_{i+1}} = \frac{-\frac{h_i}{2} - 2\frac{\tilde{b}_{i+1}}{\tilde{a}_{i+1}}}{\frac{-6\,\tilde{b}_{i+1}}{h_i\,\tilde{a}_{i+1}} - 2}$$

$$\boxed{\tilde{p}_i = \frac{\frac{h_i}{2} + 2\tilde{p}_{i+1}}{\frac{6\,\tilde{p}_{i+1}}{h_i} - 2}} \qquad i = n-2,\ldots,1$$

$$\tilde{q}_i = \frac{\tilde{c}_i}{\tilde{a}_i} = \frac{\tilde{c}_i/\tilde{a}_{i+1}}{\tilde{a}_i/\tilde{a}_{i+1}} = \frac{\frac{\tilde{c}_{i+1}}{\tilde{a}_{i+1}} - m_i\left(\frac{6}{h_i}\frac{\tilde{b}_{i+1}}{\tilde{a}_{i+1}} + 3\right)}{-\frac{6}{h_i}\tilde{p}_{i+1} - 2}$$

$$\boxed{\tilde{q}_i = \frac{-\tilde{q}_{i+1} + m_i\left(\frac{6}{h_i}\tilde{p}_{i+1} + 3\right)}{\frac{6}{h_i}\tilde{p}_{i+1} + 2}} \qquad i = n-2,\ldots,1$$

Donc l'algorithme avec cette normalisation devient :

<u>NOR-1</u>

<u>De Gauche à Droite</u>

$$(1.6) \quad \begin{cases} a'_i = \frac{6}{h_{i-1}} p_{i-1} - 2 \\ p_i = \left(\frac{h_{i-1}}{2} - 2p_{i-1}\right)/a'_i \\ q_i = (q_{i-1} + m_{i-1}(a'_i - 1))/a'_i \qquad i = 3,\ldots,n \end{cases}$$

avec $\quad p_2 = -h_1/3 \quad q_2 = m_1 \quad$ et $\quad \boxed{\lambda_i + p_i\beta_i = q_i}$

De Droite à Gauche

$$(1.6')\quad \begin{cases} \tilde{a}'_i = \dfrac{6}{h_i}\,\tilde{p}_{i+1} + 2 \\ \tilde{p}_i = \left(\dfrac{h_i}{2} + 2\tilde{p}_{i+1}\right)/\tilde{a}'_i \\ \tilde{q}_i = \left(-\tilde{q}_{i+1} + m_i(\tilde{a}'_i + 1)\right)/\tilde{a}'_i \qquad i = n-2,\dots,1 \end{cases}$$

avec $\tilde{p}_{n-1} = h_{n-1}/3 \quad \tilde{q}_{n-1} = m_{n-1}$ et $\boxed{\lambda_i + \tilde{p}_i\beta_i = \tilde{q}_i}$

On a alors :

$$\begin{cases} \beta_i = \dfrac{q_i - \tilde{q}_i}{p_i - \tilde{p}_i},\ \lambda_i = \tilde{q}_i - \tilde{p}_i\beta_i \qquad i = 2,\dots,n-1 \\ \lambda_1 = \tilde{q}_1 \quad \lambda_n = q_n \quad \beta_1 = \beta_n = 0 \end{cases}$$

NORMALISATION 2 (NOR-2)

$a_i\lambda_i + b_i\beta_i = c_i$ est équivalent à $\left(-\dfrac{h_i\,a_i}{2b_i}\right)\lambda_i - \dfrac{h_i}{2}\,\beta_i = \left(-\dfrac{h_i c_i}{2b_i}\right)$ et soit

$p_i = -\dfrac{h_i a_i}{2b_i} \qquad q_i = -\dfrac{h_i c_i}{2b_i}$

$\tilde{a}_i\lambda_i + \tilde{b}_i\beta_i = \tilde{c}_i \iff \left(-\dfrac{h_i a_i}{2\tilde{b}_i}\right)\lambda_i - \dfrac{h_i}{2}\,\beta_i = \left(-\dfrac{h_i\tilde{c}_i}{2\tilde{b}_i}\right)$

Comme $a_2 = \dfrac{6}{h_1} \quad b_2 = -2 \quad c_2 = m_1\dfrac{6}{h_1}$ on a :

$$p_2 = 1.5\,\frac{h_2}{h_1} \qquad q_2 = 1.5\,m_1\,\frac{h_2}{h_1}$$

et

$$p_i = -\frac{h_i a_i}{2b_i} = -\frac{h_i a_i/2b_{i-1}}{b_i/b_{i-1}} = -\frac{\dfrac{h_i}{2}\left(\dfrac{6}{h_{i-1}} - 2\,\dfrac{a_{i-1}}{b_{i-1}}\right)}{\dfrac{a_{i-1}h_{i-1}}{2b_{i-1}} - 2} = -\frac{\dfrac{h_i}{h_{i-1}}(3+2p_{i-1})}{-p_{i-1} - 2}$$

$$\boxed{p_i = \frac{h_i}{h_{i-1}}\left(2 - \frac{1}{p_{i-1}+2}\right)} \qquad i = 3,\dots,n \quad (\text{avec } h_n = h_{n-1})$$

$$q_i = -\frac{h_i c_i}{2b_i} = \frac{-h_i c_i/2b_{i-1}}{b_i/b_{i-1}} = \frac{-\frac{h_i c_{i-1}}{2b_{i-1}} - m_{i-1}\left(3b_{i-1} - 3\frac{a_{i-1}h_{i-1}}{2b_{i-1}}\right)\frac{h_i}{h_{i-1}}}{-p_{i-1}-2}$$

$$\boxed{q_i = \frac{h_i}{h_{i-1}} \frac{(-q_{i-1}+3m_{i-1}(1+p_{i-1}))}{p_{i-1}+2}} \qquad i = 3,\ldots,n$$

(avec $h_n = h_{n-1}$)

$$\tilde{p}_i = -\frac{h_i\tilde{a}_i}{2\tilde{b}_i} = \frac{-h_i\tilde{a}_i/2\tilde{b}_{i+1}}{\tilde{b}_i/\tilde{b}_{i+1}} = \frac{\left(3+\frac{\tilde{a}_{i+1}h_i}{\tilde{b}_{i+1}}\right)}{-\frac{h_i\tilde{a}_{i+1}}{2\tilde{b}_{i+1}}-2}$$

$$\boxed{\tilde{p}_i = -\left(2+\frac{h_{i+1}/h_i}{\tilde{p}_{i+1}-2h_{i+1}/h_i}\right)} \qquad i = n-2,\ldots,1$$

$$\tilde{q}_i = -\frac{h_i\tilde{c}_i}{2\tilde{b}_i} = \frac{-h_i\tilde{c}_i/2\tilde{b}_{i+1}}{\tilde{b}_i/\tilde{b}_{i+1}} = \frac{\frac{-h_i\tilde{c}_{i+1}}{\tilde{b}_{i+1}} + m_i\left(3+3\frac{\tilde{a}_{i+1}h_i}{\tilde{b}_{i+1}}\right)}{\frac{-h_i\tilde{a}_{i+1}}{2\tilde{b}_{i+1}} - 2}$$

$$\boxed{\tilde{q}_i = \frac{\tilde{q}_{i+1}}{\tilde{p}_{i+1}-2h_{i+1}/h_i} - 3m_i\left(1+\frac{h_{i+1}/h_i}{\tilde{p}_{i+1}-2h_{i+1}/h_i}\right)} \qquad i = n-2,\ldots,1$$

Donc l'algorithme avec la normalisation 2 est :

NOR-2

De Gauche à Droite

$$(1.7)\quad \begin{cases} p_i = \dfrac{h_i}{h_{i-1}}\left(2-\dfrac{1}{p_{i-1}+2}\right) \\ q_i = \dfrac{h_i}{h_{i-1}}\left(\dfrac{-q_{i-1}+3m_{i-1}(1+p_{i-1})}{p_{i-1}+2}\right) \end{cases} \qquad i = 2,\dots,n$$

avec

$$p_1 = 0 \quad q_1 = 0 \quad (h_n = h_{n-1}) \text{ et } \boxed{p_i\lambda_i - \frac{h_i}{2}\beta_i = q_i}$$

De Droite à Gauche

$$(1.7')\quad \begin{cases} \tilde{p}_i = -\left(2+\dfrac{h_{i+1}/h_i}{\tilde{p}_{i+1}-2h_{i+1}/h_i}\right) \\ \tilde{q}_i = \dfrac{\tilde{q}_{i+1}}{\tilde{p}_{i+1}-2h_{i+1}/h_i} - 3m_i\left(1+\dfrac{h_{i+1}/h_i}{\tilde{p}_{i+1}-2h_{i+1}/h_i}\right) \end{cases} \qquad i = n-1,\dots,1$$

avec

$$\tilde{p}_n = 0 \quad \tilde{q}_n = 0 \quad (h_n = h_{n-1}) \text{ et } \boxed{\tilde{p}_i\lambda_i - \frac{h_i}{2}\beta_i = \tilde{q}_i}$$

On a alors :

$$\begin{cases} \lambda_i = \dfrac{q_i-\tilde{q}_i}{p_i-\tilde{p}_i} \qquad \beta_i = (\tilde{p}_i\lambda_i-\tilde{q}_i)\dfrac{2}{h_i} \qquad i = 2,\dots,n-1 \\ \lambda_1 = \tilde{q}_1/\tilde{p}_1 \qquad \lambda_n = q_n/p_n \qquad \beta_1 = \beta_n = 0 \end{cases}$$

Proposition 3 :

Dans le procédé (1.6) (resp. (1.6')) on a :

a) $p_i < 0$ (resp. $\tilde{p}_i > 0$)

b) $-\frac{1}{3}h_{i-1} < p_i < -\frac{1}{4}h_{i-1}$ (resp. $\frac{1}{4}h_i < \tilde{p}_i < \frac{1}{3}h_i$)

c) $|q_i| \le 3\bar{m}$ (resp. $|\tilde{q}_i| \le 3\bar{m}$) où $\bar{m} = \max\limits_{j=1,\dots,n-1} |m_j|$

Démonstration

a) Découle de la proposition 1

b) $$p_i = \frac{\frac{h_{i-1}}{2} - 2p_{i-1}}{\frac{6}{h_{i-1}} p_{i-1} - 2} = -\frac{h_{i-1}}{3} \cdot \left(1 + \frac{1}{2\left(\frac{6p_{i-1}}{h_{i-1}} - 2\right)}\right)$$

Comme $p_{i-1} < 0$ on a $\frac{6p_{i-1}}{h_{i-1}} - 2 < -2$,alors $\frac{3}{4} < 1 + \frac{1}{2\left(\frac{6p_{i-1}}{h_{i-1}} - 2\right)} < 1$

d'où $$-\frac{1}{3} h_{i-1} < p_i = -\frac{h_{i-1}}{3} \left(1 + \frac{1}{2\left(\frac{6p_{i-1}}{h_{i-1}} - 2\right)}\right) < -\frac{1}{4} h_{i-1}$$

Ce qui montre :

c) $p_{i-1} < 0$ implique $\left| \frac{6p_{i-1}}{h_{i-1}} - 2 \right| > 2$ et on a

$$\left| \frac{\frac{6p_{i-1}}{h_{i-1}} - 3}{\frac{6}{h_{i-1}} p_{i-1} - 2} \right| = \left| 1 - \frac{1}{\frac{6}{h_{i-1}} p_{i-1} - 2} \right| \leq \frac{3}{2}$$

Comme on a

$$q_i = \frac{q_{i-1} + m_{i-1}\left(\frac{6}{h_{i-1}} p_{i-1} - 3\right)}{\frac{6}{h_{i-1}} p_{i-1} - 2}$$ en tenant compte de l'inégalité ci-dessu on obtient :

$$|q_i| \leq |m_{i-1}| \frac{3}{2} + \frac{1}{2} |q_{i-1}|$$

ce qui entraîne

$$|q_i| \leq \frac{3}{2} |m_{i-1}| + 3 \frac{1}{2^2} |m_{i-2}| + \ldots + 3 \frac{1}{2^{i-2}} |m_2| + \frac{1}{2^{i-2}} |q_2|$$

comme $q_2 = m_1$

On a alors :

$$|q_i| \leq 3 \sum_{j=1}^{i-1} \frac{1}{2^j} |m_{i-j}| \leq \frac{3}{2} \bar{m} \left(\sum_{j=0}^{i-2} \frac{1}{2^j} \right)$$

Donc :

$$|q_i| \leq 3\bar{m} \qquad \text{avec } \bar{m} = \max_{j=1,\dots,n-1} |m_j|$$

Démonstration semblable pour ce qui concerne les suites $\tilde{p}_i$ et $\tilde{q}_i$.

Cette proposition montre qu'avec la normalisation 1 les coefficients restent bornés et à priori on aura moins de risque d'erreurs dans le calcul des inconnues λ_i, β_i.

<u>Proposition 4</u>

Dans le procédé (1.7) (resp. (1.7')) on a :

a) $p_i > 0$ (resp. $\tilde{p}_i < 0$)

b) $\frac{3}{2} \left(\frac{h_i}{h_{i-1}}\right) < p_i < 2\left(\frac{h_i}{h_{i-1}}\right)$ (resp. $-2 < \tilde{p}_i < -\frac{3}{2}$)

c) $|q_i| \leq 6 \frac{\bar{h}}{\underline{h}} \bar{m}$ (resp. $|\tilde{q}_i| \leq 6 \frac{\bar{h}}{\underline{h}} \bar{m}$)

où

$\bar{h} = \max_{j=1,\dots,n-1} \{h_j\}$ $\quad \underline{h} = \min_{j=1,\dots,n-1} \{h_j\}$ $\quad \bar{m} = \max_{j=1,\dots,n-1} |m_j|$

<u>Démonstration</u>

a) Conséquence directe de la proposition 1.

b) $p_{i-1} > 0$ on a alors $p_{i-1} + 2 > 2$ d'où

$$\frac{3}{2} < 2 - \frac{1}{p_{i-1}+2} < 2 \quad \text{et comme} \quad p_i = \frac{h_i}{h_{i-1}} \left(2 - \frac{1}{p_{i-1}+2}\right)$$

on déduit $\frac{3}{2} \frac{h_i}{h_{i-1}} < p_i < 2 \frac{h_i}{h_{i-1}}$.

c) $$q_i = \frac{h_i}{h_{i-1}} \left(\frac{-q_{i-1}+3m_{i-1}(1+p_{i-1})}{p_{i-1}+2}\right), \quad p_{i-1} > 0 \text{ entraîne}$$

$$|q_i| \leq 3|m_{i-1}|\frac{h_i}{h_{i-1}} + \frac{1}{2}\frac{h_i}{h_{i-1}}|q_{i-1}|$$ et on en déduit de façon récursiv

$$|q_i| \leq 3|m_{i-1}|\frac{h_i}{h_{i-1}} + \frac{3}{2}\frac{h_i}{h_{i-2}}|m_{i-2}| + \frac{1}{2^2}\frac{h_i}{h_{i-2}}|q_{i-2}|$$

$$|q_i| \leq 3|m_{i-1}|\frac{h_i}{h_{i-1}} + \frac{3}{2}|m_{i-2}|\frac{h_i}{h_{i-2}} + \ldots + \frac{3}{2^{i-3}}|m_2|\frac{h_i}{h_2} + \frac{1}{2^{i-2}}\frac{h_i}{h_2}|q_2|$$

et comme $|q_2| = \frac{3}{2}\frac{h_2}{h_1}|m_1|$ on a :

$$|q_i| \leq 3\sum_{j=1}^{i-1}\frac{1}{2^{j-1}}|m_{i-j}|\frac{h_i}{h_{i-j}} \leq 3\,\bar{m}\,\frac{\bar{h}}{\underline{h}}\left(\sum_{j=1}^{i-1}\frac{1}{2^{j-1}}\right)$$

$$|q_i| \leq 6\,\bar{m}\,\frac{\bar{h}}{\underline{h}} .$$

La proposition ci-dessus exprime que l'algorithme de la méthode (A) avec la normalisation 2 est lui aussi stable en un certain sens si les rapports $\left(\frac{h_i}{h_{i-1}}\right)$ restent bornés.

<u>Proposition 5</u> :

Si p_i et q_i sont calculés au moyen de la relation (1.7) et si d_i et r_i sont évalués au moyen de (1.5) alors on a :

$$(1.8) \quad \begin{cases} d_i = p_i+2 & \\ 3r_i = c_i+3m_i & i = 1,\ldots,n-1 \\ d_n = p_n & \\ 3r_n = q_n & \end{cases}$$

Démonstration

Par récurrence.

On constate $d_1 = 2 = p_1+2$ et si $d_i = p_i+2$ alors

$$a_{i+1} = \frac{h_{i+1}}{h_i}\left(2-\frac{1}{d_i}\right) + 2 = \underbrace{\frac{h_{i+1}}{h_i}\left(2-\frac{1}{p_i+2}\right) + 2}_{p_{i+1}}$$

d'où $d_{i+1} = p_{i+1}+2$.

Maintenant

$$d_n = 2-\frac{2}{d_{n-1}} = 2-\frac{1}{p_{n-1}+2} = p_n.$$

De la même façon $3r_1 = 3m_1 = q_1+3m_1$ et si $3r_i = q_i+3m_i$ on a :

$$3r_{i+1} = 3\,\frac{h_{i+1}}{h_i}\left(m_i-\frac{r_i}{d_i}\right) + 3m_{i+1} = \frac{h_{i+1}}{h_i}\left(3m_i-\frac{3r_i}{d_i}\right) + 3m_{i+1}$$

$$= \frac{h_{i+1}}{h_i}\left(3m_i-\frac{q_i+3m_i}{p_i+2}\right) + 3m_{i+1}$$

$$= \frac{h_{i+1}}{h_i}\left(\frac{-q_i}{p_i+2} + 3m_i\left(1-\frac{1}{p_i+2}\right)\right) + 3m_{i+1}$$

donc

$$3r_{i+1} = q_{i+1} + 3m_{i+1}$$

et

$$3r_n = 3m_{n-1} - \frac{3r_{n-1}}{d_{n-1}} = 3m_{n-1} - \frac{q_{n-1}+3m_{n-1}}{p_{n-1}+2}$$

$$= -\frac{q_{n-1}}{p_{n-1}+2} + 3m_{n-1}\left(1-\frac{1}{p_{n-1}+2}\right) = q_n.$$

Cette proposition montre que la méthode (A) avec une normalisation bien choisie (NOR-2) dans sa première étape (1.7) de Gauche à Droite est équivalente(à une addition près) à la triangularisation dans la méthode (B).

Alors la deuxième étape (1.7') (de Droite à Gauche) peut être remplacée par :

(NOR-2 + MODIFICATION)

$$(1.9)\quad \begin{cases} \lambda_n = q_n/p_n \\ \lambda_i = (q_i+3m_i-\lambda_{i+1})/(p_i+2) & i = n-1,\dots,1 \\ \beta_i = \dfrac{2}{h_i}\,(\lambda_i p_i - q_i) \end{cases}$$

où p_i et q_i sont calculés au moyen de (1.7).

I.5 - PAS EQUIDISTANTS (PAS H)

Les relations (1.3) et (1.3') peuvent s'écrire

De Gauche à Droite

$$(1.10)\quad \begin{cases} a_i = 6b_{i-1}-2a_{i-1} \\ b_i = \dfrac{a_{i-1}}{2} - 2b_{i-1} \\ c_i = c_{i-1}+m_{i-1}(6b_{i-1}-3a_{i-1}) & i = 2,\dots,n \end{cases}$$

avec $a_1 = 0 \quad b_1 = 1 \quad c_1 = 0.$

De Droite à Gauche

$$(1.10')\quad \begin{cases} \tilde{a}_i = -2\tilde{a}_{i+1} - 6\tilde{b}_{i+1} \\ \tilde{b}_i = -\dfrac{\tilde{a}_{i+1}}{2} - 2\tilde{b}_{i+1} \\ \tilde{c}_i = \tilde{c}_{i+1}-m_i(3\tilde{a}_{i+1}+6\tilde{b}_{i+1}) & i = n-1,\dots,1 \end{cases}$$

avec $\tilde{a}_n = 0 \quad \tilde{b}_n = 1 \quad \tilde{c}_n = 0.$

et on a alors :

$$\begin{cases} a_i\lambda_i + b_i\ h\ \beta_i = c_i \\ \tilde{a}_i\lambda_i + \tilde{b}_i\ h\ \beta_i = \tilde{c}_i \end{cases}$$

NOR-1

De Gauche à Droite

$$(1.11) \quad \begin{cases} p_i = -\frac{1}{6}(2+\frac{1}{6p_{i-1}-2}) \\ q_i = \frac{q_{i-1}}{6p_{i-1}-2} + m_{i-1}(1-\frac{1}{6p_{i-1}-2}) \end{cases} \qquad i = 3,\dots,n$$

où $p_2 = -1/3 \qquad q_2 = m_1$

De Droite à Gauche

$$(1.11') \quad \begin{cases} \tilde{p}_i = \frac{1}{6}\ (2-\frac{1}{6\tilde{p}_{i+1}+2}) \\ \tilde{q}_i = \frac{\tilde{q}_{i+1}}{6\tilde{p}_{i+1}+2} + m_i\ (1+\frac{1}{6\tilde{p}_{i+1}+2}) \end{cases} \qquad i = n-2,\dots,1$$

où $\tilde{p}_{n-1} = 1/3 \qquad \tilde{q}_{n-1} = m_{n-1}$

et alors

$$\begin{cases} \lambda_i + p_i\ h\ \beta_i = q_i \\ \lambda_i + \tilde{p}_i\ h\ \beta_i = \tilde{q}_i \end{cases}$$

d'où :

$$\begin{cases} \beta_i = \frac{1}{h}\ (\frac{q_i-\tilde{q}_i}{p_i-\tilde{p}_i}) \qquad \lambda_i = \tilde{q}_i - \tilde{p}_i\ h\ \beta_i \qquad i = 2,\dots,n-1 \\ \lambda_1 = \tilde{q}_1 \qquad \lambda_n = q_n \qquad \beta_1 = \beta_n = 0 \end{cases}$$

NOR-2

De Gauche à Droite

$$(1.12)\quad \begin{cases} p_i = 2 - \dfrac{1}{p_{i-1}+2} \\[2ex] q_i = -\dfrac{q_{i-1}}{p_{i-1}+2} + 3m_{i-1}\left(1 - \dfrac{1}{p_{i-1}+2}\right) \end{cases} \qquad i = 2,\dots,n$$

avec $p_1 = 0 \quad q_1 = 0.$

De Droite à Gauche

$$(1.12')\quad \begin{cases} \tilde{p}_i = -\left(2 + \dfrac{1}{\tilde{p}_{i+1}-2}\right) \\[2ex] \tilde{q}_i = \dfrac{\tilde{q}_{i+1}}{\tilde{p}_{i+1}-2} - 3m_i\left(1 + \dfrac{1}{\tilde{p}_{i+1}-2}\right) \end{cases} \qquad i = n-1,\dots,1$$

avec $\tilde{p}_n = 0 \quad \tilde{q}_n = 0$

et
$$\begin{cases} p_i\lambda_i - \dfrac{h}{2}\beta_i = q_i \\[2ex] \tilde{p}_i\lambda_i - \dfrac{h}{2}\beta_i = \tilde{q}_i \end{cases}$$

d'où
$$\begin{cases} \lambda_i = \dfrac{q-\tilde{q}_i}{p-\tilde{p}_i} \quad \beta_i = \dfrac{2}{h}\left(\tilde{p}_i\lambda_i - \tilde{q}_i\right) \qquad i = 2,\dots,n-1 \\[2ex] \lambda_1 = \tilde{q}_1/\tilde{p}_1 \quad \lambda_n = q_n/p_n \qquad \beta_1 = \beta_n = 0 \end{cases}$$

De plus, on sait que (1.12') peut être remplacée par :

$$(1.13)\quad \begin{cases} \lambda_n = q_n/p_n \\[2ex] \lambda_i = (q_i + 3m_i - \lambda_{i+1})/(p_i+2) \qquad i = n-1,\dots,1 \\[2ex] \beta_i = \dfrac{2}{h}\left(\lambda_i\, p_i - q_i\right) \end{cases}$$

Proposition 6 :

On suppose n grand.

a) Dans (1.11) (resp. (1.11') on a :

$$p_i \to \frac{-1}{2\sqrt{3}} \quad (\text{resp. } \tilde{p}_{n-i} \to \frac{1}{2\sqrt{3}}) \text{ quand i croît}$$

b) Dans (1.12) (resp. (1.12')) on a :

$$p_i \to \sqrt{3} \quad (\text{resp. } \tilde{p}_{n-i} \to -\sqrt{3}\ ) \text{ quand i croît}$$

Proposition 7 :

Si p_i et $\tilde{p}_i$ sont obtenus au moyen de (1.11) et (1.11') ou (1.12) et (1.12') alors

$$p_i = -\tilde{p}_{n-i+1} \qquad i = 2,\ldots,n-1$$

Remarque :

1) Il n'est donc plus nécessaire de calculer les deux suites $\{p_i\}$ $\{\tilde{p}_i\}$ mais seulement l'une d'entre elles.

2) Dans le procédé (1.11) (resp. 1.12)) la convergence de la suite $\{p_i\}$ vers $-\frac{1}{2\sqrt{3}}$ (resp. $\sqrt{3}$) est très rapide, ceci nous permet de calculer une fois pour toutes les premiers termes, et de considérer la suite comme stationnaire, comme on le voit dans le tableau suivant calculé en précision double, donc l'algorithme se réduit au calcul de q_i et $\tilde{q}_i$.

Tableau 1

i	p_i (1.11)	$p_i - (-\frac{1}{2\sqrt{3}})$
3 *	-0.291666666666667	-0.00299153207185374
4 *	-0.288888888888889	-0.000213754294075968
5 *	-0.288690476190476	-1.53415956632708'-05
6 *	-0.288676236044657	-1.10144984419303'-06
7 *	-0.288675213675214	-7.90804007555490'-08
8 *	-0.288675140272529	-5.67771656079241'-09
9 *	-0.288675135002455	-4.07641670419068'-10
10 *	-0.288675134624080	-2.92673385526854'-11
11 *	-0.288675134596914	-2.10129136313242'-12
12 *	-0.288675134594964	-1.50851553470943'-13
13 *	-0.288675134594824	-1.08107967022875'-14
14 *	-0.288675134594814	-7.49400541621981'-16
15 *	-0.288675134594813	0

Tableau 2

i	p_i (1.12)	$p_i - \sqrt{3}$
3 *	1.71428571428571	-0.0177650932831630
4 *	1.73076923076923	-0.00128157679964658
5 *	1.73195876288660	-9.20446822794130'-05
6 *	1.73204419889503	-6.60867384971731'-06
7 *	1.73205033308660	-4.74482274803734'-07
8 *	1.73205077350258	-3.40662988929097'-08
9 *	1.73205080512303	-2.44585018904786'-09
10 *	1.73205080739327	-1.75604197849566'-10
11 *	1.73205080755627	-1.26079147122482'-11
12 *	1.73205080756797	-9.05275854279353'-13
13 *	1.73205080756881	-6.50590692430342'-14
14 *	1.73205080756887	-4.66293670342566'-15
15 *	1.73205080756888	-4.44089209850063'-16
16 *	1.73205080756888	0

$$-\frac{1}{2\sqrt{3}} = -0.288675134594813$$

$$\sqrt{3} = 1.73205080756888$$

I.6 - COMPARAISON DES ALGORITHMES

On compare la méthode (A) avec les différentes normalisations (usuelle, NOR-1, NOR-2) et la méthode (B).

Mémoire :

Pour les deux normalisations suggérées la méthode (A) utilise le même nombre de vecteurs que la méthode (B), à savoir 4 vecteurs : $\begin{cases} x_i \\ f_i \\ \lambda_i \\ \beta_i \end{cases}$

Coût :

Comparaison en nombre d'opérations, des algorithmes présentés ainsi que de la méthode (A) avec la normalisation usuelle.

Pas quelconque

Tableau 3

	Méthode A Usuelle (1.3)_(1.3')	Méthode A NOR-1 (1.6)_(1.6')	Méthode A NOR-2 (1.7)_(1.7')	Méthode A NOR-2 MOD. (1.7)_(1.9)	Méthode B [7] (1.5)
Sust. Add.	23 n - 34	15 n - 30	15 n - 35	12 n - 20	12 n - 20
Mult. Div.	33 n - 53	18 n - 39	19 n - 44	13 n - 22	12 n - 13
$\sqrt{\ }$	2 n - 4	——	——	——	——

Pas équidistants

Tableau 4

	Méthode A Usuelle (1.10) (1.10')	Méthode A NOR-1 (1.11) (1.11')	Méthode A NOR-2 (1.12) (1.12')	Méthode A NOR-2 MOD. (1.12) (1.13)	Méthode B [7]	
Sust. Add.	15 n - 30	14 n - 27	14 n - 26	10 n - 15	8 n - 8	$n \leq 15$
		8 n + 14	8 n + 14	7 n + 20	7 n + 3	$n > 15$
Mult. Div.	21 n - 40	17 n - 32	13 n - 20	10 n - 12	8 n - 5	$n \leq 15$
		11 n + 8	11 n - 11	8 n + 2	7 n - 6	$n > 15$
$\sqrt{\ }$	2 n - 4	1	1	1	1	

Conclusions :

1) La méthode du transport des relations avec les deux normalisations suggérées reste stable même pour n grand.
De façon expérimentale on a constaté que les résultats numériques (λ_i, β_i $i = 1,\ldots,n$) sont pratiquement les mêmes, le maximum des noeuds considérés étant 3000 . Les fonctions sur lesquelles on a testé sont (entre autres) :

a) $f(x) = 2+ \dfrac{3(x-1)}{x^2-x+1}$ sur $[-4,5]$

b) $f(x) = 10 + \dfrac{50(x-10)}{x^2-x+1}$ sur $[-40,50]$

c) $f(x) = xe^{1/x}$ sur $[0.1, 5.1]$

d) $f(x) = \exp(x(x-2)(x-4)(6-x))$ sur $[-0.5, 6.5]$

2°) Le coût par rapport à la Méthode A avec la normalisation usuelle a été réduit comme l'on voit dans les tableaux comparatifs, on a donc des algorithmes plus performants.

```
C
      SUBROUTINE SPLIN1(N,X,F,D1,D2)
C
C     ON CALCULE LA FONCTION SPLINE S D'ORDRE 2 VERIFIANT
C        S(X(I))=F(I) , I=1,...,N , AVEC  4<= N
C
C     METHODE:TRANSPORT DES RELATIONS NORMALISATION 1
C     D1,D2 CONTIENT EN SORTIE: D1(I)=S'(X(I))
C                               D2(I)=S''(X(I))
C     X,F,D1,D2 EN DOUBLE PRECISION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),F(1),D1(1),D2(1)
C     GAUCHE - DROITE
      H=X(2)-X(1)
      D2(2)=-H/3.D0
      D1(2)=(F(2)-F(1))/H
      N1=N-1
      DO 1 I=3,N1
      H=X(I)-X(I-1)
      P=6.D0*D2(I-1)/H -2.D0
      D2(I)=(H/2.D0 -2.D0*D2(I-1))/P
      D1(I)=(D1(I-1)+(F(I)-F(I-1))*(P-1.D0)/H)/P
    1 CONTINUE
      H=X(N)-X(N-1)
      V=(F(N)-F(N-1))/H
      P=6.D0*D2(N1)/H -2.D0
      D1(N)=(D1(N1)+V*(P-1.D0))/P
      D2(N)=0.D0
C     DROITE - GAUCHE
      BA=H/3.D0
      DA=V
      D2(N1)=(D1(N1)-DA)/(D2(N1)-BA)
      D1(N1)=DA-BA*D2(N1)
      N2=N-2
      DO 2 J=2,N2
      I=N-J
      H=X(I+1)-X(I)
      P=-2.D0-6.D0*BA/H
      BA=(-H/2.D0 -2.D0*BA)/P
      DA=(DA+(F(I+1)-F(I))*(P-1.D0)/H)/P
      D2(I)=(D1(I)-DA)/(D2(I)-BA)
      D1(I)=DA-BA*D2(I)
    2 CONTINUE
      H=X(2)-X(1)
      P=-2.D0-6.D0*BA/H
      V=DA+(F(2)-F(1))*(P-1.D0)/H
      D1(1)=V/P
      D2(1)=0.D0
      RETURN
      END
```

```
C
      SUBROUTINE SPLIN2(N,X,F,D1,D2)
C
C     ON CALCULE LA FONCTION SPLINE S D'ORDRE 2 VERIFIANT
C        S(X(I))=F(I) , I=1,...,N  , AVEC  4<= N
C
C     METHODE:TRANSPORT DES RELATIONS NORMALISATION 2
C     D1,D2 CONTIENT EN SORTIE: D1(I)=S'(X(I))
C                               D2(I)=S''(X(I))
C     X,F,D1,D2 EN DOUBLE PRECISION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),F(1),D1(1),D2(1)
C     GAUCHE - DROITE
      H=X(2)-X(1)
      R=X(3)-X(2)
      Q=R/H
      D1(2)=1.5D0*Q
      D2(2)=D1(2)*(F(2)-F(1))/H
      N1=N-1
      DO 1 I=3,N1
      H=X(I+1)-X(I)
      Q=R/H
      P=(D1(I-1)+2.D0)*Q
      V=3.D0 +2.D0 *D1(I-1)
      D2(I)=(-D2(I-1)+(V+D1(I-1))*(F(I)-F(I-1))/R)/P
      D1(I)=V/P
      R=H
1     CONTINUE
      V=-3.D0 -2.D0*D1(N1)
      P=(F(N)-F(N1))/H
      D1(N)=(D2(N1)+P*(V-D1(N1)))/V
      D2(N)=0.D0
C     DROITE - GAUCHE
      DA=-1.5D0
      BA=DA*P
      D1(N1)=(D2(N1)-BA)/(D1(N1)-DA)
      D2(N1)=(DA*D1(N1)-BA)*2.D0/R
      N2=N-2
      DO 2 J=2,N2
      I=N-J
      H=X(I+1)-X(I)
      Q=R/H
      P=DA-2.D0 *Q
      V=-2.D0 *DA+3.D0 *Q
      BA=(BA+(V-DA)*(F(I+1)-F(I))/H)/P
      DA=V/P
      D1(I)=(D2(I)-BA)/(D1(I)-DA)
      D2(I)=(DA*D1(I)-BA)*2.D0/H
      R=H
2     CONTINUE
      H=X(2)-X(1)
      Q=R/H
      V=-2.D0*DA+3.D0*Q
      P=BA+(V-DA)*(F(2)-F(1))/H
      D1(1)=P/V
      D2(1)=0.D0
      RETURN
      END
```

```
C
      SUBROUTINE SPLIN3(N,X,F,D1,D2)
C
C     ON CALCULE LA FONCTION SPLINE S D'ORDRE 2 VERIFIANT
C        S(X(I))=F(I) , I=1,...,N  ,AVEC  4<=N
C
C     METHODE:TRANSPORT DES RELATIONS NORMALISATION 2
C                     + MODIFICATION
C     D1,D2 CONTIENT EN SORTIE: D1(I)=S'(X(I))
C                               D2(I)=S''(X(I))
C     X,F,D1,D2 EN DOUBLE PRECISION
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1),F(1),D1(1),D2(1)
C     GAUCHE - DROITE
      H=X(2)-X(1)
      R=X(3)-X(2)
      Q=R/H
      D1(2)=1.5D0*Q
      D2(2)=D1(2)*(F(2)-F(1))/H
      N1=N-1
      DO 1 I=3,N1
      H=X(I+1)-X(I)
      Q=R/H
      P=(D1(I-1)+2.D0)*Q
      V=3.D0 +2.D0*D1(I-1)
      D2(I)=(-D2(I-1)+(V*D1(I-1))*(F(I)-F(I-1))/R)/P
      D1(I)=V/P
      R=H
    1 CONTINUE
      V=-3.D0 -2.D0*D1(N1)
      P=(F(N)-F(N1))/H
      D1(N)=(D2(N1)+P*(V-D1(N1)))/V
      D2(N)=0.D0
C     RETOUR EN ARRIERE UTILISANT L'EQUIVALENCE MONTRE
      N2=N-2
      DO 2 J=1,N2
      I=N-J
      H=X(I+1)-X(I)
      P=D1(I)
      D1(I)=(D2(I)+3.D0*(F(I+1)-F(I))/H -D1(I+1))/(D1(I)+2.D0)
      D2(I)=(P*D1(I)-D2(I))*2.D0/H
    2 CONTINUE
      D1(1)=(3.D0*(F(2)-F(1))/(X(2)-X(1))-D1(2))/2.D0
      D2(1)=0.D0
      RETURN
      END
```

CHAPITRE - II

CALCUL DE LA FONCTION SPLINE D'INTERPOLATION POUR UN GRAND NOMBRE DE NOEUDS

II.1 - INTRODUCTION

Soient G et D deux ensembles de noeuds.

On notera σ_G, σ_D, $\sigma_{G\cup D}$ les splines d'interpolation construites respectivement sur les ensembles de noeuds G, D, $G \cup D$.

Nous nous intéressons ici aux relations entre ces fonctions splines. Peut-on obtenir σ_{GUD} à partir de σ_G et σ_D ? Nous allons plus particulièrement étudier le cas où $G \cap D = \emptyset$ (utilisant la Méthode (B)).

Calculer une spline globale (σ_{GUD}) à l'aide de fonctions-spline "Locales" (σ_G et σ_D) peut être particulièrement intéressant lorsque le nombre de noeuds est très grand. En effet, on peut alors, bien que les méthodes (A) et (B) soient très stables, avoir des problèmes d'encombrement, mais aussi des problèmes d'accumulation d'erreurs d'arrondi [18] [32] [27].

II.2 - DECOUPAGE DE L'ENSEMBLE DES NOEUDS EN DEUX SOUS-ENSEMBLES DISJOINTS

Soit $N = n+k$ et $x_i < x_j$ si $i < j$

On prend

$$G = \{x_1,\dots,x_n\} \quad \text{et} \quad D = \{x_{n+1},\dots,x_N\}$$

Le système à résoudre pour trouver σ_{GUD} est (Système (1.4), où on a multipl la n-ème equation par h_{n-1}/h_n).

$$(2.1)\begin{cases} 2\lambda_1+\lambda_2 = 3m_1 \\ \dfrac{h_i}{h_{i-1}}\lambda_{i-1}+2(1+\dfrac{h_i}{h_{i-1}})\lambda_i+\lambda_{i+1} = 3(m_i+\dfrac{h_i}{h_{i-1}}m_{i-1}) & i = 2,\dots,n-1 \\ \lambda_{n-1}+2(\dfrac{h_{n-1}}{h_n}+1)\lambda_n+\dfrac{h_{n-1}}{h_n}\lambda_{n+1} = 3(\dfrac{h_{n-1}}{h_n}m_n+m_{n-1}) \\ \dfrac{h_i}{h_{i-1}}\lambda_{i-1}+2(1+\dfrac{h_i}{h_{i-1}})\lambda_i+\lambda_{i+1} = 3(m_i+\dfrac{h_i}{h_{i-1}}m_{i-1}) & i = n+1,\dots,N-1 \\ \lambda_{N-1}+2\lambda_N = 3m_{N-1} \end{cases}$$

Notons :

$$\Lambda^G = [\sigma_G'(x_i)]_{i=1,\dots,n} \quad \Lambda^D = [\sigma_D'(x_i)]_{i=n+1,\dots,N} \quad \Lambda = [\sigma'_{GUD}(x_i)]_{i=1,\dots,N}$$

Le calcul des trois fonctions spline d'interpolation à étudier σ_{GUD}, σ_G, σ_D nécessite la résolution de trois systèmes linéaires suivants :

$$(2.2) \quad \begin{cases} \text{a)} & M\,\Lambda = Y \quad \text{(système (2.1))} \\ \text{b)} & M_G\,\Lambda^G = Y_G \\ \text{c)} & M_D\,\Lambda^D = Y_D \end{cases}$$

Où les matrices et seconds membres sont :

$$M = \begin{bmatrix} 2 & 1 & & & & & & \\ \frac{h_2}{h_1} & 2(1+\frac{h_2}{h_1}) & 1 & & & & 0 & \\ & \frac{h_{n-1}}{h_{n-2}} & 2(1+\frac{h_{n-1}}{h_{n-2}}) & 1 & & & & \\ & & 1 & 2(\frac{h_{n-1}}{h_n}+1) & \frac{h_{n-1}}{h_n} & & & \\ & & & \frac{h_{n+1}}{h_n} & 2(1+\frac{h_{n+1}}{h_n}) & 1 & & \\ & 0 & & & \frac{h_{N-1}}{h_{N-2}} & 2(1+\frac{h_{N-1}}{h_{N-2}}) & 1 & \\ & & & & & 1 & 2 & \end{bmatrix} \quad Y = 3 \begin{bmatrix} m_1 \\ m_2 + \frac{h_2}{h_1} m_1 \\ \vdots \\ m_{n-1} + \frac{h_{n-1}}{h_{n-2}} m_{n-2} \\ \frac{h_{n-1}}{h_n} m_n + m_{n-1} \\ m_{n+1} + \frac{h_{n+1}}{h_n} m_n \\ \vdots \\ 1\ m_{N-1} + \frac{h_{N-1}}{h_{N-2}} m_{N-2} \\ m_{N-1} \end{bmatrix}$$

$M \in \mathcal{M}_{N,N}(\mathbb{R}) \qquad Y \in \mathbb{R}^N$

$$M_G = \begin{bmatrix} 2 & 1 & & & 0 \\ \frac{h_2}{h_1} & 2(1+\frac{h_2}{h_1}) & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{h_{n-1}}{h_{n-2}} & 2(1+\frac{h_{n-1}}{h_{n-2}}) & 1 \\ 0 & & & 1 & 2 \end{bmatrix} \qquad Y_G = 3 \begin{bmatrix} m_1 \\ \frac{h_2}{h_1} m_1 + m_2 \\ \vdots \\ \frac{h_{n-1}}{h_{n-2}} m_{n-2} + m_{n-1} \\ m_{n-1} \end{bmatrix}$$

$$M_G \in \mathcal{M}_{n,n} \qquad Y_G \in \mathbb{R}^n$$

$$M_D = \begin{bmatrix} 2 & 1 & & & 0 \\ \frac{h_{n+2}}{h_{n+1}} & 2(1+\frac{h_{n+2}}{h_{n+1}}) & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{h_{N+1}}{h_{N-2}} & 2(1+\frac{h_{N-1}}{h_{N-2}}) & 1 \\ 0 & & & 1 & 2 \end{bmatrix} \qquad Y_D = 3 \begin{bmatrix} m_{n+1} \\ \frac{h_{n+2}}{h_{n+1}} m_{n+1} + m_{n+2} \\ \vdots \\ \frac{h_{N-1}}{h_{N-2}} m_{N-2} + m_{N-1} \\ m_{N-1} \end{bmatrix}$$

$$M_D \in \mathcal{M}_{k,k} \qquad Y_G \in \mathbb{R}^k$$

Posons :

$$A = \begin{bmatrix} M_G & 0 \\ 0 & M_D \end{bmatrix} \qquad b = \begin{bmatrix} Y_G \\ Y_D \end{bmatrix} \qquad A \in \mathcal{M}_{N,N}(\mathbb{R}) \qquad b \in \mathbb{R}^N$$

$$e_n^T = [0 \ldots 0 \ 1] \quad e_n \in \mathbb{R}^n$$

$$e_1^T = [1, 0 \ldots 0] \quad e_1 \in \mathbb{R}^k$$

$$Z = 3m_n \begin{bmatrix} p & e_n \\ q & e_1 \end{bmatrix} \qquad U = \begin{bmatrix} e_n & 0 \\ 0 & e_1 \end{bmatrix} \qquad V = \begin{bmatrix} 2pe_n & qe_n \\ pe_1 & 2qe_1 \end{bmatrix}$$

avec

$$p = \frac{h_{n-1}}{h_n} \qquad q = \frac{h_{n+1}}{h_n}$$

$Z \in \mathbb{R}^N$ $\quad U$ et $V \in \mathcal{M}_{N,2}(\mathbb{R})$

On vérifie que $M = 1+UV^T$ $\quad Y = b+Z$ et d'après [8] on sait que :

$$M^{-1} = A^{-1} - A^{-1}U(I_2+V^TA^{-1}U)^{-1}V^TA^{-1}$$

ceci implique :

(2.3) $$\Lambda = M^{-1}Y = A^{-1}b+A^{-1}Z-A^{-1}U(I_2+V^TA^{-1}U)^{-1}V^TA^{-1}(b+Z)$$

appelons :

(2.4) $$\Delta = A^{-1}Z-A^{-1}U(I_2+V^TA^{-1}U)^{-1}V^TA^{-1}(b+Z)$$

Nous allons expliciter le vecteur Δ

Soit $\quad \gamma = M_G^{-1}e_n \in \mathbb{R}^n$

$\theta = M_D^{-1}e_1 \in \mathbb{R}^k$

alors :

$$A^{-1}U = \begin{bmatrix} \gamma & 0 \\ 0 & \theta \end{bmatrix} \qquad V^TA^{-1}U = \begin{bmatrix} 2p\gamma_n & p\theta_1 \\ q\gamma_n & 2q\theta_1 \end{bmatrix}$$

d'où :

(2.5) $$(I_2+V^TA^{-1}U)^{-1} = \frac{1}{r}\begin{bmatrix} 1+2q\theta_1 & -p\theta_1 \\ -q\gamma_n & 1+2p\gamma_n \end{bmatrix}$$

avec $r = 1+2(p\gamma_n+q\theta_1)+3pq\theta_1\gamma_n$

$$A^{-1}Z = 3m_n \begin{bmatrix} p & \gamma \\ q & \theta \end{bmatrix} \qquad \text{et } A^{-1}b = \begin{bmatrix} \Lambda^G \\ \Lambda^D \end{bmatrix}$$

alors on a :

$$(2.6) \qquad V^T A^{-1}(b+Z) = \begin{bmatrix} p(2\Lambda_n^G + 3m_n(2p\gamma_n + q\theta_1) + \Lambda_1^D) \\ q(\Lambda_n^G + 3m_n(2q\theta_1 + p\gamma_n) + 2\Lambda_1^D) \end{bmatrix}$$

(2.5) et (2.6) implique :

$$A^{-1}U(I_2 + V^T A^{-1}U)^{-1}V^T A^{-1}(b+Z) = \frac{1}{r}\begin{bmatrix} p(2+3q\theta_1)\Lambda_n^G + 3m_n[q\theta_1 + p\gamma_n(2+3q\theta_1)] + \Lambda_1^D)\gamma \\ q(\Lambda_n^G + 3m_n[p\gamma_n + q\theta_1(2+3p\gamma_n)] + (2+3p\gamma_n)\Lambda_1^D)\theta \end{bmatrix}$$

Notons $\quad \Lambda = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \end{bmatrix} \quad \Delta = \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} \quad$ avec $\quad \Lambda_1$ et $\Delta_1 \in \mathbb{R}^n$
Λ_2 et $\Delta_2 \in \mathbb{R}^k$

alors

$$\begin{cases} \Delta_1 = p[3m_n - \frac{1}{r}([2+3q\theta_1]\Lambda_n^G + 3m_n[q\theta_1 + p\gamma_n(2+3q\theta_1)] + \Lambda_1^D]\gamma \\ \Delta_2 = q[3m_n - \frac{1}{r}(\Lambda_n^G + 3m_n[p\gamma_n + q\theta_1(2+3p\gamma_n)] + (2+3p\gamma_n)\Lambda_1^D]\theta \end{cases}$$

ce qui peut encore s'écrire

$$(2.7) \begin{cases} \Delta_1 = \frac{p}{r}[3m_n(1+q\theta_1) - (2+3q\theta_1)\Lambda_n^G - \Lambda_1^D]\gamma \\ \Delta_2 = \frac{q}{r}[3m_n(1+p\gamma_n) - \Lambda_n^G - (2+3p\gamma_n)\Lambda_1^D]\theta \end{cases}$$

où $\qquad r = 1 + 2(p\gamma_n + q\theta_1) + 3pq\theta_1\gamma_n$

alors

$$(2.8) \begin{cases} \Lambda_1 = \Lambda^G + \Delta_1 \\ \Lambda_2 = \Lambda^D + \Delta_2 \end{cases}$$

Remarque :

Le vecteur Δ sera appelé vecteur correction, comme on le verra dans le paragraphe suivant Δ_1 et Δ_2 sont facilement obtenus et alors la relation (2.8) nous permet de conclure : "Pour trouver la spline d'interpolation σ_{GUD} il suffit de résoudre deux systèmes (2.2(a) et 2.2(b)) de taille plus petite et faire une correction adéquate" (relation (2.7)).

II.3 - CALCUL DU VECTEUR CORRECTION Δ

Pas Quelconque

Comme $\gamma = M_G^{-1} e_n$, l'élimination de Gauss fait pour résoudre (2.2(a) va nous permettre le calcul de γ .

Alors prenant la notation de (1.5) on a :

$$(2.9) \quad \left\{ \quad \gamma_i = (-1)^{n-i} \prod_{j=i}^{n} \frac{1}{d_j} \qquad i = n,\dots,1 \right.$$

De la même façon $\theta = M_D^{-1} e_1$ peut être trouvé en utilisant l'éliminati de Gauss pour résoudre 2.2(c) on suggère de le faire de bas en haut (triangulaire inférieur), pour avoir la valeur explicite de θ on fait un changement de variable :

$$\tilde{x}_i = x_{N-i+1}$$

$$\tilde{f}_i = f_{N-i+1} \qquad i = 1,\dots,k$$

alors 2.2(c) est équivalent à :

$$(2.10) \quad \left\{ \begin{array}{ll} 2\tilde{\lambda}_1+\tilde{\lambda}_2 = 3\tilde{m}_1 & \\ \dfrac{\tilde{h}_i}{\tilde{h}_{i-1}} \tilde{\lambda}_{i-1}+2(1+\dfrac{\tilde{h}_i}{\tilde{h}_{i-1}})\tilde{\lambda}_i+\tilde{\lambda}_{i+1} = 3(\tilde{m}_i+\dfrac{\tilde{h}_i}{\tilde{h}_{i-1}} \tilde{m}_{i-1}) & i=2,\dots,k-1 \\ \tilde{\lambda}_{k-1}+2\tilde{\lambda}_k = 3\tilde{m}_{k-1} & \end{array} \right.$$

où $\tilde{h}_i = \tilde{x}_{i+1} - \tilde{x}_i$ $\quad \tilde{m}_i = \dfrac{\tilde{f}_{i+1} - \tilde{f}_i}{\tilde{h}_i}$ $\quad i = 1,\ldots,k-1$

$\lambda_i^D = \tilde{\lambda}_{k-i+1}$ $\quad i = 1,\ldots,k$

Si $\tilde{d}_i$ $i = 1,\ldots,k$ sont calculées comme dans (1.5) (en changeant x_i $i = 1,\ldots,n$ en $\tilde{x}_i$ $i = 1,\ldots,k$)

$$(2.11) \quad \left\{ \quad \theta_i = (-1)^{i+1} \prod_{j=k-i+1}^{k} \frac{1}{\tilde{d}_j} \qquad i = 1,\ldots,k \right.$$

<u>Proposition 8</u> :

Si d_i est calculé en utilisant (1.5) on a :

$$d_i > 2 + \frac{3}{2}\left(\frac{h_i}{h_{i-1}}\right) \qquad i = 2,\ldots,n-1$$

$$d_n > \frac{3}{2}$$

$$d_1 = 2.$$

<u>Démonstration</u>

Découle de la proposition (4) et (5); pour montrer $d_n > \frac{3}{2}$ on utilise $d_n = 2 - \frac{1}{d_{n-1}}$ et $d_{n-1} > 2$.

Remarquons que si $\tilde{d}_j$ est calculé au moyen de (1.5) et utilisant $\tilde{x}_i$ $i = 1,\ldots,k$ on a aussi :

$$\left\{ \begin{array}{ll} \tilde{d}_j > 2 + \dfrac{3}{2}\dfrac{\tilde{h}_i}{\tilde{h}_{j-1}} & j = 2,\ldots,k-1 \\ \tilde{d}_k > \dfrac{3}{2} & \end{array} \right.$$

Si on note $\underline{h} = \min\limits_{i=1,\ldots,N-1} \{h_i\}$ $\qquad \alpha = \dfrac{\underline{h}}{\bar{h}}$

$\bar{h} = \max\limits_{i=1,\ldots N-1} \{h_i\}$

alors $d_i > 2 + \frac{3}{2}\alpha$ $\qquad i = 2,\ldots,n-1$

$\tilde{d}_j > 2 + \frac{3}{2}\alpha$ $\qquad j = 2,\ldots,k-1$

Par conséquent on a :

$$(2.12)\quad \begin{cases} |\gamma_i| < \frac{2}{3}\left(\frac{2}{4+3\alpha}\right)^{n-i} & i = n,\dots,2 \\ |\gamma_1| < \frac{1}{3}\left(\frac{2}{4+3\alpha}\right)^{n-2} & \\ |\theta_j| < \frac{2}{3}\left(\frac{2}{4+3\alpha}\right)^{j-1} & j = 1,\dots,k-1 \\ |\theta_k| < \frac{1}{3}\left(\frac{2}{4+3\alpha}\right)^{k-2} & \end{cases}$$

Alors pour n et k grand les premiers composants de γ et les derniers composants de θ sont pratiquement nuls, d'autre part, si les coefficients de γ et θ dans la relation (2.7) restent bornés, le vecteur correction Δ a des composants significatifs seulement "AUTOUR" de la n-ième composante.

Soit

$$(2.13)\quad \begin{cases} c_1 = \frac{p}{r}\,[3m_n(1+q\theta_1)-(2+3q\theta_1)\Lambda_n^G-\Lambda_1^D] \\ c_2 = \frac{q}{r}\,[3m_n(1+p\gamma_n)-\Lambda_n^G-(2+3p\gamma_n)\Lambda_1^D] \end{cases}$$

où $\quad r = 1+2(p\gamma_n+q\theta_1)+3pq\theta_1\gamma_n$

Et supposons vérifié

$$(2.14)\qquad \frac{1}{\alpha} = \frac{\bar{h}}{\underline{h}} \le \beta$$

alors on aura :

$$(2.15)\quad \begin{cases} |c_1| < \beta[(2+3\beta\theta_1)|m_n-\Lambda_n^G|+|m_n-\Lambda_1^D|] \\ |c_2| < \beta[(2+3\beta\gamma_n)|m_n-\Lambda_1^D|+|m_n-\Lambda_n^G|] \end{cases}$$

donc on a la proposition suivante

Proposition 9

Si (2.14) est vérifié alors :

$$\begin{cases} |c_1| \to 0 \\ |c_2| \to 0 \end{cases} \quad \text{quand } \bar{h} \to 0$$

Remarque

Sous l'hypothèse (2.14), les coefficients c_1 et c_2 des vecteurs γ et θ sont bornés et ils convergent même vers zéro quand $\bar{h} \to 0$.

Par conséquent le vecteur correction Δ aura des composantes significatives autour de la n-ème composante seulement.

Expérimentalement on a constaté qu'il suffit de corriger les 30 dernières composantes de Λ_1^G et les 30 premières de Λ^D. Le nombre de composants à corriger est lié à la précision de la machine utilisée, dans notre cas IBM 360

Avant d'expliciter le vecteur correction pour le cas Pas-équidistants considérons la matrice M tridiagonale suivante :

$$M = \begin{bmatrix} \alpha & 1 & & & \\ c & a & 1 & & \\ & \ddots & \ddots & & \\ & & c & a & 1 \\ & & & c & \beta \end{bmatrix} \qquad c > 0 \qquad M \in \mathcal{M}_{n,n}(\mathbb{R})$$

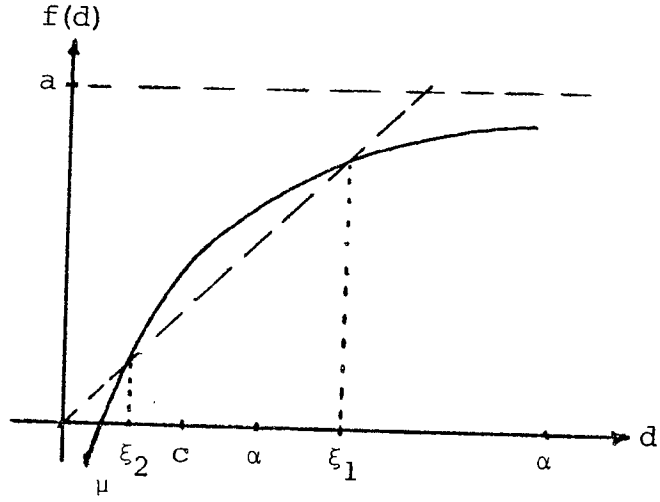
Si on suppose $a > c+1$ $\alpha > c$ et $\alpha > 1$ alors on peut factoriser M (cf. [31])

$$M = \begin{bmatrix} 1 & & & \\ \ell_2 & 1 & & \\ & \ddots & \ddots & \\ & & \ell_n & 1 \end{bmatrix} \qquad \begin{bmatrix} d_1 & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ & & & d_n \end{bmatrix}$$

où : $\ell_i = \dfrac{1}{d_{i-1}}$ $\quad d_i = a - \dfrac{1}{d_{i-1}}$ $\quad i = 2,\dots,n-1$

$\ell_n = \dfrac{1}{d_{n-1}}$ $\quad d_n = \beta - \dfrac{c}{d_{n-1}}$ $\quad d_1 = \alpha$

La suite d_i converge vers un point fixe de $f(d) = a - \dfrac{c}{d}$.



Ces points fixes sont : $\begin{cases} \xi_1 = \dfrac{a+\sqrt{a^2-4c}}{2} \\ \xi_2 = \dfrac{a-\sqrt{a^2-4c}}{2} \end{cases}$

et $\mu = \dfrac{a}{c}$.

$a > c+1$ peut s'écrire $-1 > c-a$ ce qui implique $-4c > 4c^2-4ca$ de même $a^2-4c > a^2+4c^2-4ca$ alors $\sqrt{a^2-4c} > |a-2c|$

donc $\xi_2 = \dfrac{a-\sqrt{a^2-4c}}{2} < \dfrac{a-|a-2c|}{2} \leq c$

Utilisant le fait que f est croissante on voit que

$\begin{cases} d_i \text{ est croissante si } \alpha < \xi_1 \\ d_i \text{ est décroissante si } \alpha > \xi_1 \end{cases}$

Dans les deux cas $\{d_i\}$ converge vers le point fixe ξ_1.
De plus on a :

$$d_{i+1}-d_i = \frac{c}{d_i-d_{i-1}}\,(d_i-d_{i-1}) \text{ on a donc}$$

$$(2.16)\quad \begin{cases} d_{i+1}-d_i = \dfrac{c}{d_i d_1}\left[\displaystyle\prod_{j=2}^{i-1}\left(\frac{c}{d_j^2}\right)\right](d_2-d_1) & i = 3,\ldots \\ |d_{i+1}-d_i| \le \left(\dfrac{c}{\alpha^2}\right)^{i-1}|d_2-d_1| \end{cases}$$

Ceci implique qu'à partir d'une certaine valeur i on aura pratiquement $d_{i+1} = d_i$, et comme en réalité la suite d_i est obtenue avec une erreur due à l'arithmétique de la machine (précision limitée)

<u>Propriété</u> :

"A partir d'une certaine valeur i^* on peut considérer la suite d_i comme stationnaire", c'est-à-dire $d_i = d_{i^*}$ $i > i^*$ et la valeur de i^* peut être estimée à l'aide de (2.16).

<u>Remarque</u> :

Cette propriété et la relation (1.8) justifient les tableaux 1 et 2 : I-5.

<u>Quelques résultats numériques</u> :

1. $a = 3.1 \quad c = 2$

$\xi_2 = (3.1-\sqrt{1.61})/2$

$\xi_1 = (3.1+\sqrt{1.61})/2$

$d_{i^*} = 2.18442887702248 \simeq \xi_1$

α	-3	0.5	1.5	3
i^*	42	43	43	41

2. $a = 3.5 \quad c = 2$

$\xi_2 = (3.5-\sqrt{4.25})/2$

$\xi_1 = (3.5+\sqrt{4.25})/2$

$d_{i^*} = 2.78077640640442 \simeq \xi_1$

α	-3	0.5	1.5	4
i^*	29	30	28	28

3. $a = 4 \qquad c = 2$

$\xi_2 = 2-\sqrt{2}$

$\xi_1 = 2+\sqrt{2}$

$d_{i^*} = 3.41421356237309 \approx \xi_1$

α	-3	0.3	2.5	5
i^*	23	24	21	22

4. $a = 4 \qquad c = 1$

$\xi_2 = 1-\sqrt{3}$

$\xi_1 = 2+\sqrt{3}$

$d_{i^*} = 3.73205080756888 \approx \xi_1$

α	-3	0.1	2	7
i^*	16	16	15	15

Remarques :

1) Même pour $\alpha \in]-\infty,\mu[\cup]\xi_1,c]$ la suite $\{d_i\}$ reste encore stationnaire (2 ou 3 itérations de plus).

2) Si $\alpha \in]\mu,\xi_1[$ il y a le risque d'avoir un certain $d_j = 0$. Dans ce cas la décomposition considérée n'est plus possible, par exemple si $\alpha = \dfrac{ac}{a^2-c}$ on a $d_3 = 0$.

3) La convergence est lente si $(c+1)/a$ est proche de 1, rapide sinon.

PAS EQUIDISTANTS : (Pas h $\quad n > 15 \quad k > 15$)

Les matrices 2.2(a) et (b) sont :

$$M_G = \begin{bmatrix} 2 & 1 & & \\ 1 & 4 & 1 & \\ & & \ddots & \\ & 1 & 4 & 1 \\ & & 1 & 2 \end{bmatrix} \qquad M_D = \begin{bmatrix} 2 & 1 & & \\ 1 & 4 & 1 & \\ & & \ddots & \\ & 1 & 4 & 1 \\ & & 1 & 2 \end{bmatrix} \qquad \begin{array}{l} M_G \in \mathcal{M}_{n,n}(\mathbb{R}) \\ M_D \in \mathcal{M}_{k,k}(\mathbb{R}) \end{array}$$

soit $\{d_i\}$ la suite obtenue au moyen de l'équation récurrente résultant de

$$\begin{cases} d_1 = 2 \\ d_i = 4 - \dfrac{1}{d_{i-1}} & i = 2,\dots,n-3 \\ d_n = 2 - \dfrac{1}{d_{n-1}} \end{cases}$$

Ceci correspond à l'exemple 4 (avec $\beta = 1$).

Dans la pratique (voir [16]) les valeurs de la suite sont :

$$(2.17)\quad \begin{cases} d_1 = 2 & \\ d_i = 4 - \dfrac{1}{d_{i-1}} & i = 2,\dots,15 \\ d_i = d_{15} = 3.73205080756888 \approx 2+\sqrt{3} & i = 16,\dots,n-1 \\ d_n = 2 - \dfrac{1}{d_{15}} \approx \sqrt{3} & \end{cases}$$

et d'après (2.9) et (2.11) les vecteurs γ et θ seront calculés au moyen de :

$$(2.18)\quad \begin{cases} \gamma_i = \dfrac{(-1)^{n-i}}{(2d_{15}-1)} \left(\dfrac{1}{d_{15}}\right)^{n-i-1} & i = n,\dots,15 \\ \gamma_i = \dfrac{(-1)^{n-i}}{(2d_{15}-1)} \left(\dfrac{1}{d_{15}}\right)^{n-16} \prod_{j=i}^{14} \dfrac{1}{d_j} & i = 14,\dots,1 \end{cases}$$

$$(2.18')\quad \begin{cases} \theta_j \doteq \dfrac{(-1)^{j+1}}{(2d_{15}-1)} \left(\dfrac{1}{d_{15}}\right)^{j-2} & j = 1,\dots,k-14 \\ \theta_j = \dfrac{(-1)^{j+1}}{(2d_{15}-1)} \left(\dfrac{1}{d_{15}}\right)^{k-16} \prod_{l=k-j+1}^{14} \left(\dfrac{1}{d_l}\right) & j = k-13,\dots,k \end{cases}$$

Remarquons que d_{15} est le point fixe machine de $f(d) = 4 - \frac{1}{d}$, et en arithmétique exacte on aura :

$$\begin{cases} d_{15} < d_i < 2+\sqrt{3} \quad \text{(point fixe théorique)} & i > 15 \\ 2 - \dfrac{1}{d_{15}} < d_n < 2 - \dfrac{1}{2+\sqrt{3}} = \sqrt{3} & \end{cases}$$

De ces relations on peut tirer une borne pour chaque composante de γ et θ (ce qui nous intéresse le plus, ce sont les dernières composantes de γ et les premières de θ) :

$$(2.19)\quad \begin{cases} |\gamma_i| < \dfrac{1}{2d_{15}-1}\left(\dfrac{1}{d_{15}}\right)^{n-i-1} & i = n,\dots,15 \\[2ex] |\theta_j| < \dfrac{1}{(2d_{15}-1)}\left(\dfrac{1}{d_{15}}\right)^{j-2} & j = 1,\dots,k-14 \end{cases}$$

Maintenant les coefficients respectifs de γ et θ. Pour avoir le vecteur correction Δ soit d'après (2.13) :

$$(2.20)\quad \begin{cases} c_1 = \dfrac{1}{r}\,[3m_n(1+\theta_1)-(2+3\theta_1)\Lambda_n^G-\Lambda_1^D] \\[2ex] c_2 = \dfrac{1}{r}\,[3m_n(1+\gamma_n)-\Lambda_n^G-(2+3\gamma_n)\Lambda_1^D] \end{cases}$$

où $\quad r = 1+2(\gamma_n+\theta_1)+3\theta_1\gamma_n$.

<u>Remarque</u> :

Supposons que $n \gg 15, k \gg 15$.

La suite γ_{n-i+1} (resp. θ_i) décroît assez vite quand i croît ($i = 1,2,\dots$) et comme les valeurs des c_1 et c_2 (2.20) sont petites pour n et k grand, le vecteur correction Δ a des composantes significatives autour de la n-ième composante seulement ; expérimentalement on a vu qu'il suffit de corriger les 15 dernières composantes de Λ^G et les 15 premieres de Λ^D.

Pour $n > 30$ et $k > 30$ dans la pratique on a $\gamma_{n-i+1} = \theta_i$ $i=1,\dots,15$ (d'après (2.18) et (2.18')) alors les 15 dernières composantes de γ (ou les 15 premières de θ) peuvent être calculées une fois pour toutes sans tenir compte de la valeur de n et k (c'est-à-dire du pas h), en conclusion on a une correction automatique : "Correction qui ne dépend plus que de $m_n = \dfrac{f_{n+1}-f_n}{h}$ ".

II.4 - STRATEGIE SUGGEREE POUR TROUVER LA SPLINE POUR UN GRAND NOMBRE DE NOEUDS

Admettons qu'on reçoive les données par paquets de n_i valeurs, appelons S_i la spline construire à partir des $N_i = n_1+\dots+n_i$ premiers noeuds et $\tilde{S}_i$ la spline construite avec les données du i-ème paquet.

Notons

$$\Lambda_i = [S_i'(x_j)]_{j=1,\dots,N_i}$$

$$\tilde{\Lambda}_i = [\tilde{S}_i'(x_j)]_{j=n_{i-1}+1,\dots,n_i}$$

CAS : PAS EQUIDISTANT

Supposons $n_i > 30 \quad \forall i$

Nous suggérons alors le procédé suivant :

1°) on calcule $\Lambda_1 = \tilde{\Lambda}_1$ (c'est-à-dire $S_1 = \tilde{S}_1$)et soit $i = 2$

2°) On prend les n_i valeurs suivantes (les f_i) et on calcule $\tilde{\Lambda}_2$ (c'est-à-dire $\tilde{S}_2$).

3°) On corrige Λ_{i-1} et $\tilde{\Lambda}_i$ pour cela nous utilisons les relations : (2.7), (2.8), (2.18) (2.18') (2.20) et nous tenons compte des remarques précédentes : c'est-à-dire "On corrige les 15 dernieres composantes de Λ_{i-1} et les 15 premières de $\tilde{\Lambda}_i$". Alors on aura Λ_i (spline S_i).

4°) On fait $i := i+1$ et retourne à la deuxième étape jusqu'à avoir saisi toutes les données.

Remarque :

On peut utiliser la méthode(A) ou (B) dans le calcul de $\tilde{\Lambda}_i$.

CAS : PAS QUELCONQUE

Essayons d'appliquer le même procédé que dans le cas de Pas équidistants.

Avec le premier paquet on calcule $\Lambda_1 = \tilde{\Lambda}_1$

Avec le deuxième paquet " " $\tilde{\Lambda}_2$

On trouve facilement le vecteur correction Δ et on peut utiliser la conséquence de la proposition (8) et (9). c'est-à-dire "corriger seulement les 30 dernières composantes de Λ_1 et les 30 premières de $\tilde{\Lambda}_2$", pour avoir Λ_2.

Soulignons que Λ_2 a été trouvé sans résoudre le système (2.2(a)) avec $N = n_1+n_2$, c'est-à-dire sans évaluer les d_i $i = 2,\ldots,N$ au moyen de (1.5).

Ensuite si on prend le paquet suivant on pourra calculer $\tilde{\Lambda}_3$ et le vecteur $\theta = M_D^{-1}e_1$ où $D = \{x_{N_2+1},\ldots,x_{N_2+n_3}\}$ $N_2 = n_1+n_2$

Mais pour calculer Λ_3 il faut trouver le vecteur correction Δ et pou cela il nous faut γ , solution du système $M_G\gamma = e_n$ où $G = \{x_1,\ldots,x_{n_1+n_2}\}$ qui peut être résolu en calculant les d_i $i = 1,\ldots,n_1+n_2$.
Mais avoir ces d_i signifierait que l'on calcule Λ_2 directement, ce qui est contraire à notre but d'utiliser les splines locales. Alors nous ne pouvons p utiliser un procédé évolutif avec les corrections étudiées.

Malgré ce problème, utilisant la conséquence de la proposition (8) et (9) on suggère le procédé suivant :

Admettons $n_1 > 60$ et soit $\hat{S}_i$ la spline relative au 60 derniers noeuds du (i-1)-ème paquet et les n_i noeuds du i-ème paquet.

Notons $\hat{\Lambda}_i = [\hat{S}_i(x_j)]_{j=n_{i-1}-59,\ldots,n_{i-1}+n_i}$

1°) On calcule Λ_1 et soit $i = 2$.

2°) On prend les 60 derniers noeuds et les n_i suivantes et calculons $\hat{\Lambda}_i$.

3°) En prenant les $(N_{i-1}-30)$ premières composantes de Λ_{i-1} et les (n_i+30) dernières de $\hat{\Lambda}_i$ on aura les composantes de Λ_i.

4°) On met $i := i+1$ et on retourne à la deuxième étape jusqu'à avoir toutes les données.

CONCLUSIONS :

Nous avons trouvé que le calcul de la fonction spline σ_{GUD} (spline globale) est possible à l'aide de fonctions spline "locales" σ_G et σ_D (relation (2.7) et (2.8)).

Quand le nombre de noeuds N est grand, expérimentalement on a constaté que les coefficients λ_i $i = 1,\ldots,N$ de la fonction spline globale sont les mêmes que ceux qu'on trouve en utilisant la stratégie suggérée (selon le cas).

Alors on peut conclure que le calcul de la fonction spline globale (N grand) est possible en résolvant des systèmes de taille faible, la stratégie suggérée constitue un moyen de calcul.

CHAPITRE - III

FONCTION SPLINE D'AJUSTEMENT D'ORDRE 2

III.1 - INTRODUCTION

Reprenons les notations du chapitre I, et soit :

$$a_i = [a^i_{jl}]_{2\times 4} \qquad u^i = \begin{bmatrix} u^i_1 \\ u^i_2 \end{bmatrix} \qquad \tilde{a}^i = [\tilde{a}^i_{jl}]_{2\times 4} \qquad \tilde{u}^i = \begin{bmatrix} \tilde{u}^i_1 \\ \tilde{u}^i_2 \end{bmatrix}$$

$$v^i = [f_i \lambda_i \beta_i \gamma_i^+]^T \qquad\qquad i = 1,\dots,n.$$

La méthode (A) au moyen de deux procédés qui seront explicités ensuite construit les suites $\{a^i\}$ $\{u^i\}$ $\{\tilde{a}^i\}$ $\{\tilde{u}^i\}$ et qui vérifient (voir [12])

$$(3.1) \qquad \begin{cases} a^i v^i = u^i \\ \tilde{a}^i v^i = \tilde{u}^i \end{cases} \qquad i = 1,\dots,n$$

Comme dans le cas d'interpolation le problème est de contrôler les suites $\{a^i\}$ et $\{\tilde{a}^i\}$, c'est-à-dire normaliser chaque équation de (3.1) de façon à ce que ces coefficients ne puissent pas croître vers l'infini ou décroître vers zéro en même temps ; nous allons seulement changer la phase deux de la méthode (A) de la manière suivante :

Admettons que $\{a^i\}$ et $\{u^i\}$ soient connus, nous remplaçons dans le système $a^i v^i = u^i$ la valeur de β_i et γ_i^+ (en fonctions des f_i et λ_i) pour obtenir deux relations en f_i, λ_i, f_{i+1} et λ_{i+1}.

De cette façon nous obtenons un système diagonal par bloc qui peut être résolu par retour en arrière ; nous allons aussi étudier en détail la méthode (B).

Nous donnerons aussi les relations existants entre les splines construires sur les ensembles des noeuds G, D et G $\cup$ D.

III.2 - PRESENTATION DES ALGORITHMES

Soit $\rho > 0$ Z_i $i = 1,\dots,n$ réels et σ_ρ la spline d'ajustement relative à ρ et $Z = [Z_1,\dots,Z_N]^T$

La méthode (A) considère σ_ρ sous la forme :

$$\sigma_\rho(x) = \begin{cases} f_1+\lambda_1(x-x_1) & \text{si } x \in [a,x_1] \\ f_i+\lambda_i(x-x_i)+\frac{\beta_i}{2}(x-x_i)^2+\frac{\gamma_i^+}{6}(x-x_i)^3 & \text{si } x \in [x_i,x_{i+1}] \quad i=1,..n \\ f_n+\lambda_n(x-x_n) & \text{si } x \in [x_n,b] \end{cases}$$

L'algorithme pour trouver les suites $\{a^i\}$ $\{u^i\}$ est : (voir [12])

De Gauche à Droite

$$(3.2) \quad \begin{cases} a_{j1}^{i+1} = \rho\, a_{j4}^{i+1}+a_{j1}^{i} \\ a_{j2}^{i+1} = -h_i\, a_{j1}^{i}+a_{j2}^{i} \\ a_{j3}^{i+1} = \frac{h_i^2}{2}\, a_{j1}^{i}-h_i\, a_{j2}^{i} + a_{j3}^{i} \\ a_{j4}^{i+1} = -\frac{h_i^3}{6}\, a_{j1}^{i} + \frac{h_i^2}{2}\, a_{j2}^{i} - h_i\, a_{j3}^{i}+ a_{j4}^{i} \\ u_j^{i+1} = \rho\, z_{i+1}\, a_{j4}^{i+1}+ u_j \end{cases} \qquad \begin{array}{l} i = 1,\dots,n-1 \\ j = 1,2 \end{array}$$

avec

$$a^1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \rho & 0 & 0 & 1 \end{bmatrix} \qquad u^1 = \begin{bmatrix} 0 \\ \rho z_1 \end{bmatrix}$$

et

$$a^i v^i = u^i \qquad i = 2,\dots,n$$

De Droite à Gauche

$$(3.2')\quad \begin{cases} \tilde{a}^i_{j1} = \tilde{a}^{i+1}_{j1} - \rho \tilde{a}^{i+1}_{j4} \\ \tilde{a}^i_{j2} = h_i\, \tilde{a}^i_{j1} + \tilde{a}^{i+1}_{j2} \\ \tilde{a}^i_{j3} = \frac{h_i^2}{2}\, \tilde{a}^i_{j1} + h_i\, \tilde{a}^{i+1}_{j2} + \tilde{a}^{i+1}_{j3} \\ \tilde{a}^i_{j4} = \frac{h_i^3}{6}\, \tilde{a}^i_{j1} + \frac{h_i^2}{2}\, \tilde{a}^{i+1}_{j2} + h_i\, \tilde{a}^{i+1}_{j3} + \tilde{a}^{i+1}_{j4} \\ \tilde{u}^i_j = -\rho z_{i+1}\, \tilde{a}^{i+1}_{j4} + \tilde{u}^{i+1}_j \end{cases} \qquad \begin{matrix} i = n-1,\dots,1 \\ j = 1,2 \end{matrix}$$

avec

$$\tilde{a}^n = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \tilde{u}^n = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

et

$$\tilde{a}^i v^i = \tilde{u}^i \qquad i = n-1,\dots,1$$

La méthode (B) considère σ_ρ sous la forme :

$$\sigma_\rho(x) = \begin{cases} f_1 + \lambda_1 (x - x_1) & \text{si } x \in [a, x_1] \\ f_i + \lambda_i (x - x_i) + \frac{m_i - \lambda_i}{h_i} (x - x_i)^2 + \frac{\lambda_{i+1} - m_i + \lambda_i}{h_i^2} (x - x_i)^2 (x - x_{i+1}) & \text{si } x \in [x_i, x_{i+1}] \\ & i = 1,\dots,n-1 \\ f_n + \lambda_n (x - x_n) & \text{si } x \in [x_n, b] \end{cases}$$

Les inconnues f_i, λ_i sont solution du système tridiagonal par blocs obtenu utilisant la continuité de σ''_ρ , et la relation entre $z_i - f_i$ et la discontinuité de σ''_ρ ; remarquons que $f_i = \sigma_\rho(x_i)$ voir [16].

$$
(3.3)\quad \begin{cases}
(2+\alpha_1)f_1 + h_1\lambda_1 - 2f_2 + h_1\lambda_2 = \alpha_1 Z_1 \\[2ex]
3f_1 + 2h_1\lambda_1 - 3f_2 + h_1\lambda_2 = 0 \\[2ex]
-2\left(\dfrac{h_i}{h_{i-1}}\right)^3 f_{i-1} - \left(\dfrac{h_i}{h_{i-1}}\right)^2 h_i\lambda_{i-1} + \left[2+\alpha_i + 2\left(\dfrac{h_i}{h_{i-1}}\right)^3\right] f_i \\
\qquad + \left[1 - \left(\dfrac{h_i}{h_{i-1}}\right)^2\right] h_i\lambda_i - 2f_{i+1} + h_i\lambda_{i+1} = \alpha_i Z_i \\[2ex]
3\left(\dfrac{h_i}{h_{i-1}}\right)^2 f_{i-1} + \left(\dfrac{h_i}{h_{i-1}}\right) h_i\lambda_{i-1} + 3\left[1 - \left(\dfrac{h_i}{h_{i-1}}\right)^2\right] f_i \\
\qquad + 2\left[1 + \dfrac{h_i}{h_{i-1}}\right] h_i\lambda_i - 3f_{i+1} + h_i\lambda_{i+1} = 0 \qquad i=2,\dots,n-1 \\[2ex]
-2f_{n-1} - h_{n-1}\lambda_{n-1} + (2+\alpha_{n-1}) f_n - h_{n-1}\lambda_n = \alpha_{n-1} Z_n \\[2ex]
3f_{n-1} + h_{n-1}\lambda_{n-1} - 3f_n + 2h_{n-1}\lambda_n = 0
\end{cases}
$$

avec

$$
\alpha_i = \frac{\rho h_i^3}{6} \qquad i = 1,\dots,n-1
$$

Pour l'obtention de a^i au moyen de (3.2) on peut suggérer la normalisation suivante : diviser à chaque itération par $t_j^i = |a_{j1}^i| + |a_{j2}^i| + |a_{j3}^i| + |a_{j4}^i|$ $\quad j = 1,2$ et substituons les relations :

$$
(3.4)\quad \begin{cases}
\beta_i = \dfrac{2}{h_i}\left[3\left(\dfrac{f_{i+1}-f_i}{h_i}\right) - 2\lambda_i - \lambda_{i+1}\right] \\[2ex]
\gamma_i^+ = \dfrac{6}{h_i^2}\left[\lambda_{i+1} - 2\left(\dfrac{f_{i+1}-f_i}{h_i}\right) + \lambda_i\right]
\end{cases}
$$

Alors on a :

$$
(3.5)\quad \begin{cases}
\left(a_{j1}^i - \dfrac{6}{h_i^2} a_{j3}^i + \dfrac{12}{h_i^3} a_{j4}^i\right) f_i + \left(a_{j2}^i - \dfrac{4}{h_i} a_{j3}^i + \dfrac{6}{h_i^2} a_{j4}^i\right)\lambda_i \\[2ex]
\quad + \left(\dfrac{6}{h_i^2} a_{j3}^i - \dfrac{12}{h_i^3} a_{j4}^i\right) f_{i+1} + \left(-\dfrac{2}{h_i} a_{j3}^i + \dfrac{6}{h_i^2} a_{j4}^i\right)\lambda_{i+1} = u_j^{i+1}
\end{cases}
$$

$$
i = 1,\dots,n-1
$$

$$
j = 1,2
$$

Comme $\gamma_n^+ = 0$ et $\beta_n = 0$ la dernière équation bloc nous donne la solution de f_n et λ_n et utilisant (3.5) on fera le retour en arrière au lieu d procédé (3.2')

III-3 - ETUDE DE LA METHODE (B) - CAS PAS EQUIDISTANTS

Le système à résoudre d'après (3.3) est :

(3.6) $\quad M_n \Lambda = Y$

où

$$M_n = \begin{bmatrix} A_1 & C & & & \\ D & A & C & & \\ & \ddots & \ddots & \ddots & \\ & & D & A & C \\ & & & D & A_n \end{bmatrix} \quad A_1 = \begin{bmatrix} 2+\hat{\rho} & 1 \\ 3 & 2 \end{bmatrix} \qquad C = \begin{bmatrix} -2 & 1 \\ -3 & 1 \end{bmatrix}$$

$$A_n = \begin{bmatrix} 2+\hat{\rho} & -1 \\ -3 & 2 \end{bmatrix} \qquad D = \begin{bmatrix} -2 & -1 \\ 3 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 4+\hat{\rho} & 0 \\ 0 & 4 \end{bmatrix} \qquad \hat{\rho} = \frac{\rho h^3}{6}$$

$$\Lambda^T = [\Lambda_1^T, \ldots, \Lambda_n^T] \qquad Y^T = \hat{\rho}[Y_1^T, \ldots, Y_n^T]$$

$$\Lambda_i = \begin{bmatrix} f_i \\ h\lambda_i \end{bmatrix} \qquad Y_i \begin{bmatrix} z_i \\ 0 \end{bmatrix} \qquad i = 1, \ldots, n$$

La résolution de ce système sera faite par l'élimination simple de Gauss par bloc (voir [11],[25],[26])

$$(3.7) \quad \begin{cases} U_1 = A_1 & b_1 = Y_1 \\ U_i = A - DU_{i-1}^{-1}C & b_i = Y_i - DU_{i-1}^{-1} b_{i-1} \qquad i = 2, \ldots, n-1 \\ U_n = A_n - DU_{n-1}^{-1}C & b_n = Y_n - DU_{n-1}^{-1} b_{n-1} \end{cases}$$

alors : $\Lambda_i = \hat{\rho}\ \Lambda'_i \qquad i = 1,\dots,n$

$$(3.8)\quad \begin{cases} \Lambda'_n = U_n^{-1} b_n \\ \Lambda'_i = U_i^{-1}(b_i - C\Lambda_{i+1}) & i = n-1,\dots,1 \end{cases}$$

Ce qui reste à voir est si (3.7) est stable (voir [25]).

Remarquons que dans (3.6) si on multiplie chaque ligne bloc par la matrice $V = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$ on obtient un système symétrique, où si $\bar{A}_1 = V\,A_1$ $\bar{A}_n = VA$ $\bar{C} = VC$, $\bar{D} = VD$, $\bar{A} = VA$ $Y_i = V\,Y_i$ alors (3.7) devient :

$$(3.9)\quad \begin{cases} W_1 = \bar{A}_1 & T_1 = \bar{Y}_1 \\ W_i = \bar{A} - \bar{C}^T W_{i-1}^{-1} \bar{C} & T_i = \bar{Y}_i - \bar{C}^T W_{i-1}^{-1}\, T_{i-1} \qquad i=2,\dots,n-1 \\ W_n = \bar{A}_n - \bar{C}^T W_{n-1}^{-1} \bar{C} & T_n = \bar{Y}_n - \bar{C}^T W_{n-1}^{-1}\, T_{n-1} \end{cases}$$

où $W_i = VU_i$ (symétrique).

Si nous notons $W_i = \begin{vmatrix} a_i & b_i \\ b_i & c_i \end{vmatrix}$ on a :

$$(3.10)\quad \begin{cases} a_i = 12+3\hat{\rho} + \dfrac{36(b_{i-1}-c_{i-1})+9a_{i-1}}{a_{i-1}c_{i-1}-b_{i-1}^2} \\ b_i = \dfrac{18c_{i-1}-15b_{i-1}+3a_{i-1}}{a_{i-1}c_{i-1}-b_{i-1}^2} \\ c_i = 4+\dfrac{6b_{i-1}-9c_{i-1}-a_{i-1}}{a_{i-1}c_{i-1}-b_{i-1}^2} & i = 2,\dots,n-1 \end{cases}$$

avec

$$a_1 = 6+3\hat{\rho} \qquad b_1 = 3 \qquad c_1 = 2$$

Ensuite, on montre la stabilité numérique du procédé (3.9) sur certain condition sur ρ.

Proposition 10 :

Si on pose $\theta = \begin{bmatrix} 1 & 0 \\ 0 & \theta \end{bmatrix}$, $E = \theta \bar{C} \theta \qquad G = \theta A \theta$

alors pour $\hat{\rho} \geq 6$ on peut trouver $\theta > 0$ tel que

$$\|G^{-1}E^T\|_2 + \|G^{-1}E\|_2 \leq 1$$ (ici $\| \ \|_2$ norme matircielle associé à la norme euclidienne)

Démonstration :

$$G^{-1} = \begin{bmatrix} \frac{1}{12+3\hat{\rho}} & 0 \\ 0 & \frac{1}{4\theta^2} \end{bmatrix} \qquad E = \begin{bmatrix} -6 & 3\theta \\ -3\theta & \theta^2 \end{bmatrix}$$

$$\|E\|_2 = \left\| \begin{matrix} -6 & 3\theta \\ 3\theta & -\theta^2 \end{matrix} \right\|_2 = \frac{6+\theta^2+\sqrt{(6+\theta^2)\ +12\theta^2}}{2}$$

mais $\exists \theta > 0$ tel que $\frac{1}{12+3\hat{\rho}} \leq \frac{1}{4\theta^2}$ de façon que $\|G^{-1}\|_2 = \frac{1}{4\theta^2}$
alors on a

(3.11) $$\boxed{4\theta^2 \leq 12+3\hat{\rho}}$$

Pour avoir $\|G^{-1}E\|_2 \leq \frac{1}{2}$ il suffit que $\|G^{-1}\|_2 \ \|E_2\| \leq \frac{1}{2}$

c'est-à-dire $6+\theta^2+\sqrt{(6+\theta^2)^2+12\ \theta^2} \leq 4\theta^2$

si on suppose $\theta^2 > 2$ on aura

$$(6+\theta)^2+12\theta^2 \leq (3\theta^2-6)^2$$

en conséquence

(3.12) $$\boxed{\frac{15}{2} \leq \theta^2}$$

L'hypothèse $\hat{\rho} \geq 6$ entraîne $\frac{15}{2} \leq 3 + \frac{3}{4}\hat{\rho}$
alors on peut choisir $\theta \in [\frac{15}{2}, 3+\frac{3}{4}\hat{\rho}]$.

Ce qui montre la proposition.

Alors le système (3.6) au moyen d'un changement de variable $\bar{\Lambda}_i = [f_i \ \frac{h\Lambda_i}{\theta}]^T$ et d'une pré-multiplication matricielle convenable $\left(\begin{bmatrix} 3 & 0 \\ 0 & \theta \end{bmatrix}\right)$ peut être ramené à un système symétrique diagonale dominante par bloc (avec l'hypothèse $\hat{\rho} \geq 6$). Donc la méthode (B) est stable (voir [25], [19]).

De plus, l'itération (3.9) converge assez vite vers un point fixe. Malgré que l'on n' ait pas démontré la stabilité numérique du procédé (3.9) pour ρ quelconque de façon expérimentale, on a constaté qu'on converge vers un point fixe pour tout $\forall \hat{\rho} > 0$, la convergence étant très rapide pour $\hat{\rho} > 1/30$, comme on peut le voir dans le tableau suivant

Tableau 5

		Point fixe machine		
$\hat{\rho}$	Nb I.T	a	b	c
20/3	11	28.5628137093041	1.31887873314319	3.46702058125867
4	12	20.2158383625775	1.50000000000000	3.36930639376292
10/3	13	18.0853746856522	1.56929746660809	3.33111012187857
8/3	14	15.9242079783350	1.65594457356122	3.28262865163513
5/3	15	12.5903088383716	1.83975705258920	3.17669178784430
1	16	10.2552156443463	2.03165713322155	3.06048299595919
1/3	20	7.69495138322441	2.37889332125586	2.82788660332637
1/6	24	6.95869501535260	2.54283469919252	2.70142615637800
2/15	26	6.79919175698793	2.58699727719334	2.66440293821918
0.1	30	6.63267704129088	2.63830271236066	2.61927493271320
1/30	36	6.26155880620900	2.78485878681858	2.47207573910297
1/300	64	6.04260739726503	2.93012733443137	2.26586907266718
1/3000	100	6.00718742661432	2.97772280652663	2.14953228981799

D'après le tableau 5, pour $3\hat{\rho} < 10^{-16}$ dans la pratique, on a $W_n = 0$ qui implique $U_n = 0$. En conséquence on ne peut pas trouver Λ au moyen de (3.8). Dans ce cas on suggère comme solution la droite de moindre carré.

Ensuite, on donne quelques caractéristiques du point fixe que dans la pratique on constate, pour cela soit $\bar{a}_i = a_i - 3\hat{\rho}$ alors (3.10) devient :

$$\begin{cases} \bar{a}_i = 12 + \dfrac{36(h_{i-1}-c_{i-1})+9(\bar{a}_{i-1}+3\hat{\rho})}{(\bar{a}_{i-1}+3\hat{\rho})c_{i-1}-b_{i-1}^2} \\[2ex] b_i = \dfrac{18c_{i-1}-15b_{i-1}+3(a_{i-1}+3\hat{\rho})}{(\bar{a}_{i-1}+3\hat{\rho})c_{i-1}-b_{i-1}^2} \\[2ex] c_i = 4 + \dfrac{6b_{i-1}-9c_{i-1}-(\bar{a}_{i-1}+3\hat{\rho})}{(\bar{a}_{i-1}+3\hat{\rho})c_{i-1}-b_{i-1}^2} \end{cases}$$

avec

$$\bar{a}_1 = 6 \quad b_1 = 3 \quad c_1 = 2.$$

En utilisant le signe de la dérivée de $\bar{a}_i$, b_i, c_i par rapport à $\hat{\rho}$ on peut voir que $\bar{a}_i \nearrow$ $b_i \searrow$ et $c_i \nearrow$ quand $\hat{\rho} \to +\infty$ et :

$$\lim_{\hat{\rho}\to\infty} \bar{a}_i = 12 - \frac{q}{c_{i-1}} = \bar{a}$$

$$\lim_{\hat{\rho}\to\infty} b_i = \frac{3}{c_{i-1}} = \bar{b}$$

$$\lim_{\hat{\rho}\to\infty} c_i = 4 - \frac{1}{c_{i-1}} = \bar{c}$$

mais

$$c_i < 4 \Rightarrow \bar{c} \leq 2+\sqrt{3}$$

on a alors $\bar{a} \leq 9\sqrt{3}-6$

$$\bar{b} \geq 3(2-\sqrt{3})$$

En conclusion, le point fixe du procédé (3.10) pour $\hat{\rho}$ grand est proche de $(9\sqrt{3}-6 + 3\hat{\rho},\ 3(2-\sqrt{3}),\ 2+\sqrt{3})$ et pour $\hat{\rho}$ petit est proche de $(6+3\hat{\rho}, 3, 2)$

Cette méthode est actuellement utilisé par Utreras, qui étudie le problème de "choix du paramètre de lissage ρ . voir [24], on peut trouver aus des études de ce problème dans [6], [28], [29], [30].

On présente des graphes des fonctions-spline d'ajustement calculées au moyen de la méthode (B) avec trois valeurs de ρ.

III.4 - CALCUL DE LA SPLINE D'AJUSTEMENT AU MOYEN DE DEUX SPLINE D'AJUSTEMENT LOCALES

Nous donnons un moyen de calcul d'une telle spline de manière analogue à celui de l'interpolation, pour cela nous prendrons le système correspondant à (3.9) c'est-à-dire :

(3.15) $\quad M_N \Lambda = Y$

où

$$\begin{bmatrix} A_1 & C & & & \\ C^T & A & C & & \\ & \ddots & \ddots & \ddots & \\ & & C^T & A & C \\ & & & C^T & A_N \end{bmatrix} \qquad A_1 = \begin{bmatrix} (6+3\hat{\rho}) & 3 \\ 3 & 2 \end{bmatrix} \qquad C = \begin{bmatrix} -6 & 3 \\ -3 & 1 \end{bmatrix}$$

$$A_n = \begin{bmatrix} (6+3\hat{\rho}) & -3 \\ -3 & 2 \end{bmatrix} \qquad A = \begin{bmatrix} (12+3\hat{\rho}) & 0 \\ 0 & 4 \end{bmatrix}$$

$$M_N \in \mathcal{M}_{2N,2N}(\mathbb{R})$$

$$\Lambda^T = [\Lambda_1^T, \ldots, \Lambda_N^T] \qquad Y^T = 3\hat{\rho}\ [Y_1^T, \ldots, Y_N^T]$$

$$\Lambda_i^T = [f_i, h\lambda_i] \qquad Y_i^T = [Z_i \quad 0]$$

Soit $N = n+k$ et notons :

$$b_1^T = 3\hat{\rho}[Y_1^T, \ldots, Y_n^T] \qquad b_1 \in \mathbb{R}^{2n}$$

$$b_2^T = 3\hat{\rho}[Y_{n+1}^T, \ldots, Y_N^T] \qquad b_2 \in \mathbb{R}^{2k}$$

$$e_{2n}^T = [0, \ldots, 1] \quad e_{2n-1}^T = [0, \ldots, 1, 0] \quad e_{2n} \text{ et } e_{2n-1} \in \mathbb{R}^{2n}$$

$$e_1 = [1, \ldots, 0] \quad e_2^T = [0, 1, 0, \ldots, 0] \quad e_1 \text{ et } e_2 \in \mathbb{R}^{2k}$$

$$U = \begin{bmatrix} e_{2n-1} & e_{2n} & 0 & 0 \\ 0 & 0 & e_1 & e_2 \end{bmatrix} \qquad V=UT \qquad T = \begin{bmatrix} 6 & 2 & -6 & 3 \\ 3 & 2 & -3 & 1 \\ -6 & -3 & 6 & -3 \\ 3 & 1 & -3 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} M_n & 0 \\ 0 & M_k \end{bmatrix} \qquad A \in \mathcal{M}_{2N,2N}(\mathbb{R})$$

$$V \text{ et } U \in \mathcal{M}_{2N,4}(\mathbb{R})$$

$\alpha = M_n^{-1} e_{2n-1}$ $\quad \gamma = M_k^{-1} e_1$ $\qquad \Lambda_1, \alpha, \beta$ et $\Lambda^G \in \mathbb{R}^{2n}$

$\beta = M_n^{-1} e_n$ $\quad \theta = M_k^{-1} e_2$ $\qquad \Lambda_2, \gamma, \theta$ et $\Lambda^D \in \mathbb{R}^{2k}$

$$\Lambda^G = M_n^{-1} b_1 \qquad \Lambda = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \end{bmatrix}$$

$$\Lambda^D = M_k^{-1} b_2$$

On a avec ces notations :

$$M_N = A+UV^T$$

$$Y = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

d'où

$$\Lambda = M^{-1}Y = A^{-1}Y-A^{-1}U(I_4+V^TA^{-1}U)^{-1}V^TA^{-1}Y \qquad \text{(voir [8])}$$

ce qui peut encore s'écrire :

$$(3.16) \qquad \begin{cases} \Lambda_1 = \Lambda^G+\Delta_1 \\ \Lambda_2 = \Lambda^G+\Delta_2 \end{cases}$$

avec

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = -A^{-1}U(I_4+V^TA^{-1}U)^{-1}V^TA^{-1}Y$$

Calcul de Δ

$$A^{-1}U = \begin{bmatrix} \alpha & \beta & 0 & 0 \\ 0 & 0 & \gamma & \theta \end{bmatrix} \qquad V^TA^{-1}U = T \begin{bmatrix} \alpha_{2n-1} & \beta_{2n-1} & 0 & 0 \\ \alpha_{2n} & \beta_{2n} & 0 & 0 \\ 0 & 0 & \gamma_1 & \theta_1 \\ 0 & 0 & \gamma_2 & \theta_2 \end{bmatrix}$$

$$V^T A^{-1} Y = T \begin{bmatrix} \Lambda^G_{2n-1} \\ \Lambda^G_{2n} \\ \Lambda^D_1 \\ \Lambda^D_2 \end{bmatrix}$$

α et β seront calculés au moyen des relations :

$$(3.17) \quad \begin{cases} W_1 = A_1 \\ W_i = A - C^T W_{i-1} C & i = 2,\dots,n-1 \\ W_n = A_n - C^T W_{n-1}^{-1} C \end{cases}$$

d'où :

$$(3.18) \quad \begin{cases} \begin{bmatrix} \alpha_{2n-1} \\ \alpha_{2n} \end{bmatrix} = W_n^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} \alpha_{2i-1} \\ \alpha_{2i} \end{bmatrix} = -W_i^{-1} C \begin{bmatrix} \alpha_{2i+1} \\ \alpha_{2i+2} \end{bmatrix} & i = n-1,\dots,1 \end{cases}$$

$$(3.19) \quad \begin{cases} \begin{bmatrix} \beta_{2n-1} \\ \beta_{2n} \end{bmatrix} = W_n^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} \beta_{2i-1} \\ \beta_{2i} \end{bmatrix} = W_i^{-1} C \begin{bmatrix} \beta_{2i+1} \\ \beta_{2i+2} \end{bmatrix} & i = n-1,\dots,1 \end{cases}$$

γ et θ seront calculés à l'aide des relations

$$(3.20) \quad \begin{cases} \bar{W}_1 = A_n \\ \bar{W}_i = A - C \bar{W}_{i-1}^{-1} C^T & i = 2,\dots,k-1 \\ \bar{W}_k = A_1 - C \bar{W}_{k-1}^{-1} C^T \end{cases}$$

alors

$$(3.21)\quad \begin{cases} \begin{bmatrix}\gamma_1\\ \gamma_2\end{bmatrix} = \bar{W}_k^{-1} \begin{bmatrix}1\\ 0\end{bmatrix} \\ \begin{bmatrix}\gamma_{2i-1}\\ \gamma_{2i}\end{bmatrix} = -\bar{W}_{k-i+1}^{-1}\ C^T \begin{bmatrix}\gamma_{2i-3}\\ \gamma_{2i-2}\end{bmatrix} \qquad i = 2,\dots,k \end{cases}$$

$$(3.22)\quad \begin{cases} \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \bar{W}_k^{-1} \begin{bmatrix}1\\ 0\end{bmatrix} \\ \begin{bmatrix}\theta_{2i-1}\\ \theta_{2i}\end{bmatrix} = -\bar{W}_{k-i+1}^{-1}\ C^T \begin{bmatrix}\theta_{2i-3}\\ \theta_{2i-2}\end{bmatrix} \qquad i = 2,\dots,k \end{cases}$$

<u>Remarques</u> :

1°) Comme $A_1 = S\ A_n\ S \qquad C = S\ C^T\ S$

avec $S = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$ on a : $\begin{cases} \bar{W}_i = S\ W_i\ S \\ \bar{W}_i^{-1}\ C^T = D\ W_i^{-1}\ C\ D \end{cases}$

$i = 1,\dots,\min\ (k,n)$

ces relations peuvent être utiliser pour trouver γ et θ .

2°) En général $\|W_i^{-1}\ C\|_2$ $i = n-1,\dots,$ n'est pas plus petit que 1, et on doit calculer toutes les composantes de Δ pour calculer Λ par la formule (3.16).

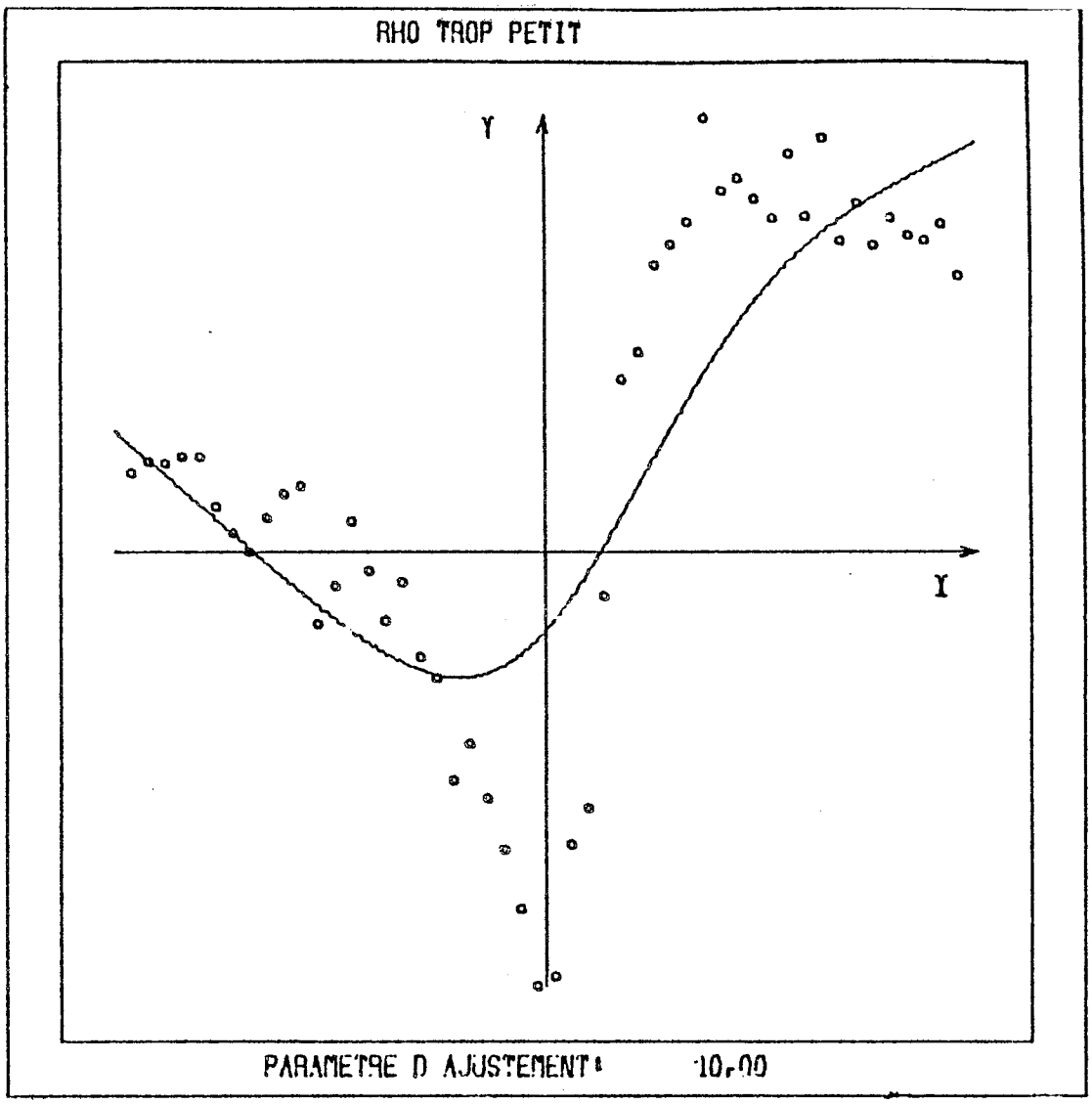
CONCLUSION

Le calcul de la fonction spline d'ajustement à partir de deux splines d'ajustement locale est possible en utilisant (3.16).

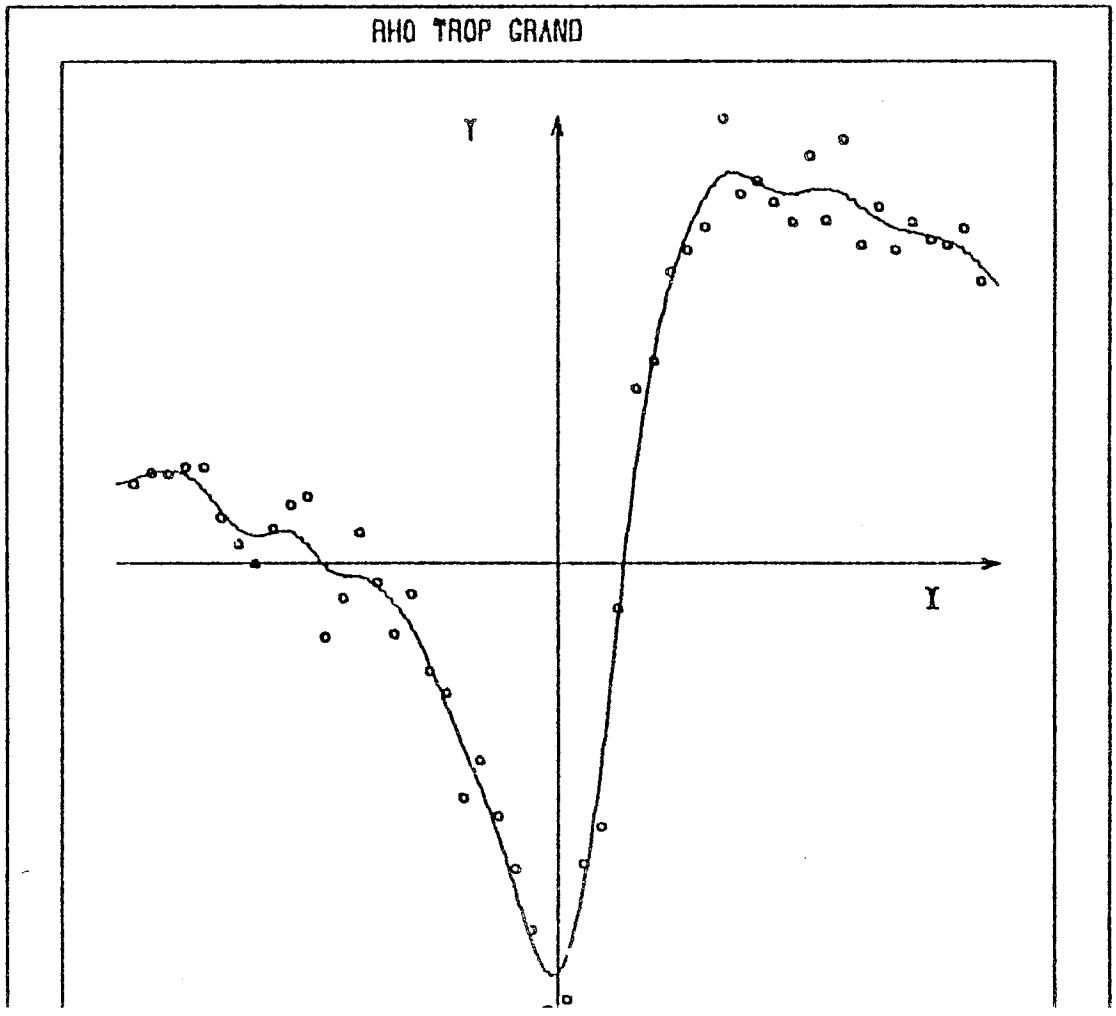
Toutes les composantes du vecteur correction sont à calculer, parce qu'elles sont significatives. Ceci implique qu'il faut corriger toutes les composantes de Λ^G et Λ^D.

Bien qu'il soit possible d'éviter la résolution par moyen de deux systèmes de taille faible, il est préférable de résoudre globalement (3.15) car la méthode (B) est stable.

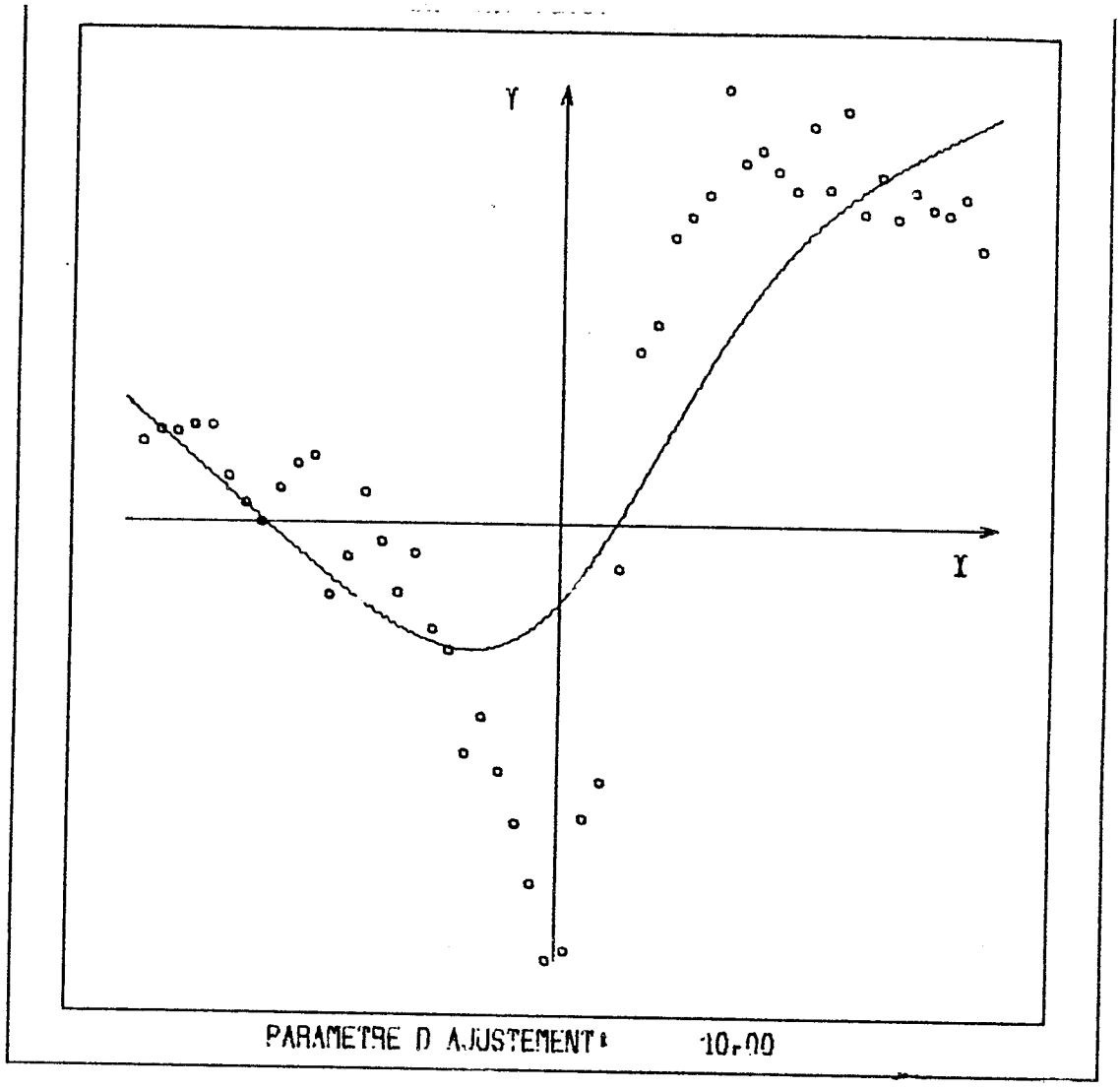
(figure 1)



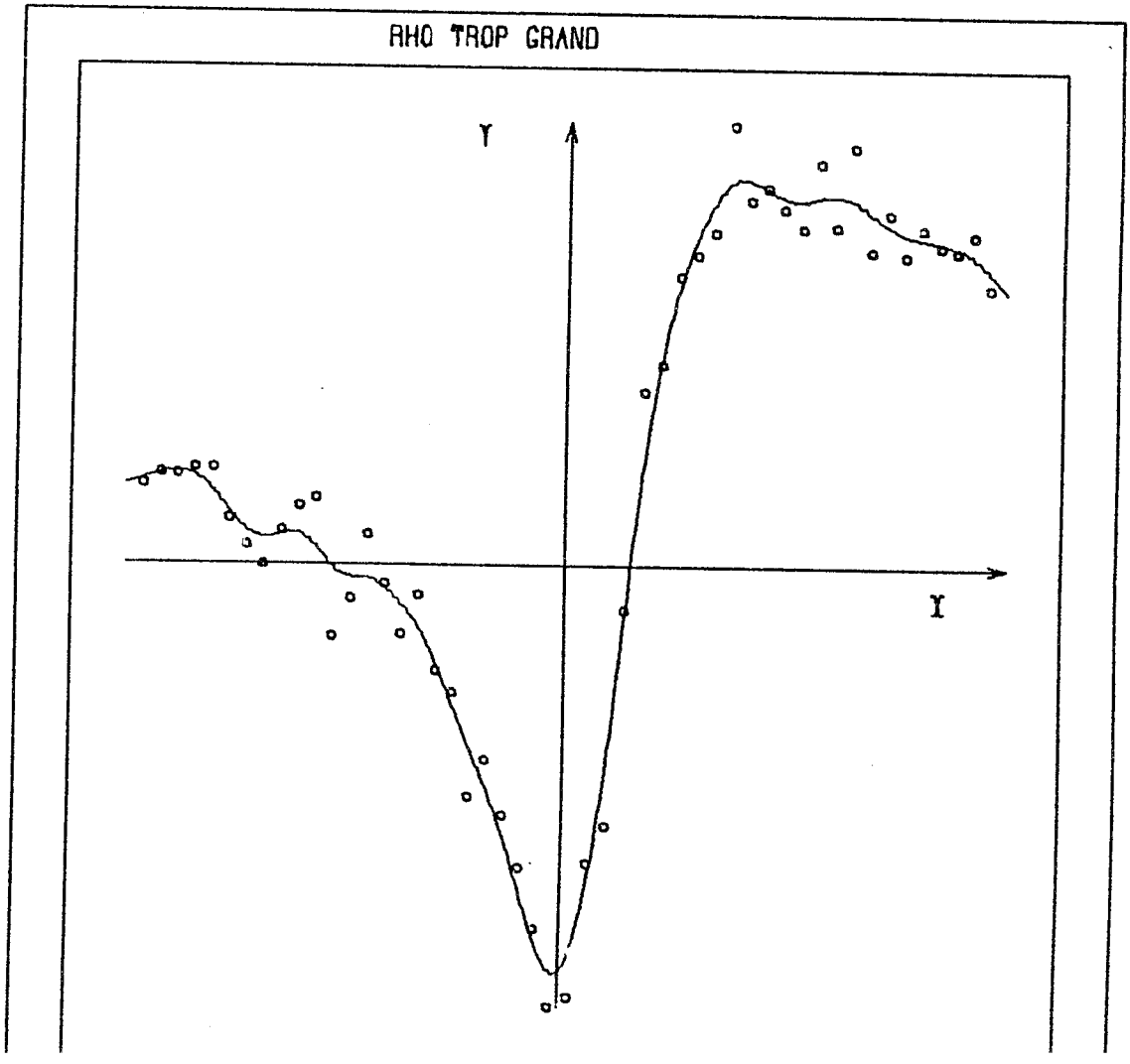
(figure 2)



(figure 1)



(figure 2)



La fonction utilisée est :

$$f(x) = 2+\frac{3(x-1)}{x^2-x+1} \quad \text{sur } [-5,5]$$

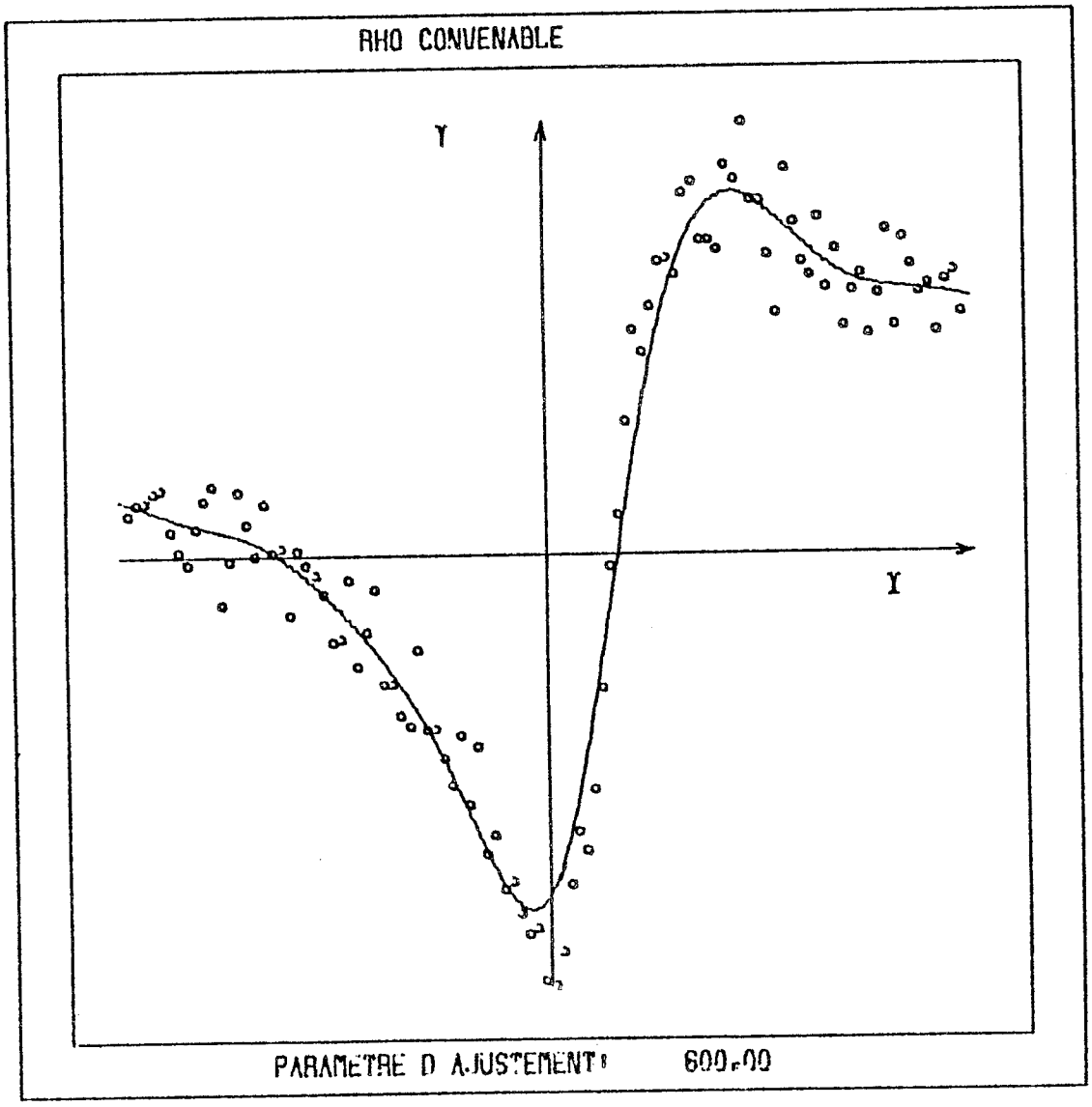
$n = 50$ (50 points équidistants)

L'erreur sur la valeur de f aux noeuds a pour écart type $\sigma = 0.2$

La figure 4 est le graphique de la fonction f ($\sigma = 0$)

La figure 5 est le spline d'ajustement considarant $n = 100$.

(Figure 5)



```
      SUBROUTINE SADJPE(N,RO,H,Z,D,A)
C
C
C     ON CALCULE LA FONCTION SPLINE D'AJUSTEMENT D'ORDRE 2, S
C     RELATIVE AUX N POINTS EQUIDISTANTS ET N VALEURS Z(I) I=1,...,N >2
C     RO PARAMETRE DE LISSAGE :RO>10'-16
C     Z(I) CONTIENT EN SORTIE LA VALEUR EN X(I) DE LA SPLINE S
C     D(I)                                      X(I) DE S'
C     A CONTIENT LA SUR DIAGONALE (BLOC) D'APRES L'ELIMINATION.
C
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION A(N,2,2),Z(1),D(1)
        ALFA=H*H*H*RO/6.D0
        DET=1.D0+ALFA+ALFA
        Z1=ALFA*Z(1)/DET
        BETA=4.D0+ALFA
        Z(1)=2.D0*Z1
        D(1)=-3.D0*Z1
        A11=1.D0/DET
        A(1,1,1)=-A11
        A(1,1,2)=A11
        A(1,2,1)=-3.D0 *ALFA/DET
        A(1,2,2)=(ALFA-1.D0)/DET
C     ELIMINATION  PAR BLOC
        N1=N-1
        DO 1 I=2,N1
        A1=2.D0*A(I-1,1,1)+A(I-1,2,1)
        A2=2.D0*A(I-1,1,2)+A(I-1,2,2)
        B1=2.D0*Z(I-1)+D(I-1)
C
        A22=BETA+A1
        A12=A2
        A21=-A1-A(I-1,1,1)
        A11=4.D0-A2-A(I-1,1,2)
        Z1=ALFA*Z(I)+B1
        D1=-(B1+Z(I-1))
C
        DET=A11*A22-A12*A21
        A11=A11/DET
        A12=-A12/DET
        A21=-A21/DET
        A22=A22/DET
        A(I,1,1)=-2.D0*A11-3.D0*A12
        A(I,1,2)=A11+A12
        A(I,2,1)=-2.D0*A21-3.D0*A22
        A(I,2,2)=A21+A22
        Z(I)=A11*Z1+A12*D1
        D(I)=A21*Z1+A22*D1
    1   CONTINUE
C
        A1=2.D0*A(N1,1,1)+A(N1,2,1)
        A2=2.D0*A(N1,1,2)+A(N1,2,2)
        B1=2.D0*Z(N1)+D(N1)
        Z1=ALFA*Z(N)+B1
        A22=ALFA+2.D0+A1
```

```
      A12=-1.D0+A2
      A21=-3.D0-A1-A(N1,1,1)
      A11=2.D0-A2-A(N1,1,2)
      D1=-(B1+Z(N1))
C     RESOLUTION DU SYSTEME TRIANGULAIRE SUPERIEUR PAR BLOC
      DET=A11*A22-A12*A21
      A11=A11/DET
      A12=-A12/DET
      A21=-A21/DET
      A22=A22/DET
      A(N,1,1)=A11
      A(N,1,2)=A12
      A(N,2,1)=A21
      A(N,2,2)=A22
      Z(N)=A11*Z1+A12*D1
      D(N)=A21*Z1+A22*D1
      DO 2 J=1,N1
      I=N-J
      Z(I)=Z(I)-(A(I,1,1)*Z(I+1)+A(I,1,2)*D(I+1))
      D(I)=D(I)-(A(I,2,1)*Z(I+1)+A(I,2,2)*D(I+1))
2     CONTINUE
      DO 3 I=1,N
      D(I)=D(I)/H
3     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SADJPQ(N,RO,X,Z,D,LOUP,A,H)
C
C
C     ON CALCULE LA FONCTION SPLINE D'AJUSTEMENT D'ORDRE 2,S
C     RELATIVE AUX NOEUDS X(I) ET AUX VALEURS Z(I)  I=1,...,N >2
C     RO PARAMETRE DE LISSAGE
C
C     Z(I) CONTIENT EN SORTIE LA VALEUR EN X(I) DE LA SPLINE  S
C     D(I)
C     LOUP=.FALSE. SI CETTE METHODE NE PEUT ETRE MENEE A SON TERME
C     A CONTIENT LA SUR DIAGONALE (BLOC) D'APRES L'ELIMINATION
C     H VECTEUR AUXILIAIRE H(I)=X(I+1)-X(I)
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL LOUP
      DIMENSION A(N,2,2),X(1),Z(1),D(1),H(1)
      N1=N-1
      DO 1 I=1,N1
      H(I)=X(I+1)-X(I)
1     CONTINUE
      H(N)=H(N1)
      ALFA=(RO*H(1)**3)/6.D0
      DET=1.D0+ALFA+ALFA
      Z1=ALFA*Z(1)/DET
      Z(1)=2.D0*Z1
      D(1)=-3.D0*Z1
      P1=H(2)/H(1)
      A1=P1*DET
      A(1,1,1)=-1.D0/DET
      A(1,1,2)=1.D0/A1
      A(1,2,1)=-3.D0*ALFA/DET
      A(1,2,2)=(ALFA-1.D0)/A1
C     ELIMINATION PAR BLOC
      DO 2 I=2,N1
      A1=2.D0*A(I-1,1,1)+A(I-1,2,1)
      A2=2.D0*A(I-1,1,2)+A(I-1,2,2)
      B1=2.D0*Z(I-1)+D(I-1)
      P2=P1*P1
      P3=P2*P1
      ALFA=(RO*H(I)**3)/6.D0
      A22=ALFA+2.D0+(2.D0+A1)*P3
      A3=1.D0-P2
      A12=A3+A2*P3
      A21=3.D0*A3-(A1+A(I-1,1,1))*P2
      A11=2.D0*(1.D0+P1)-(A2+A(I-1,1,2))*P2
      Z1=ALFA*Z(I)+B1*P3
      D1=-(B1+Z(I-1))*P2
C
      DET=A11*A22-A12*A21
      IF(DET.EQ.0.D0)GO TO 5
      A11=A11/DET
      A12=-A12/DET
      A21=-A21/DET
      A22=A22/DET
C
```

```
      P1=H(I+1)/H(I)
      A(I,1,1)=-2.D0*A11-3.D0*A12
      A(I,1,2)=(A11+A12)/P1
      A(I,2,1)=-2.D0*A21-3.D0*A22
      A(I,2,2)=(A21+A22)/P1
      Z(I)=A11*Z1+A12*D1
      D(I)=A21*Z1+A22*D1
2     CONTINUE
C
      A1=2.D0*A(N1,1,1)+A(N1,2,1)
      A2=2.D0*A(N1,1,2)+A(N1,2,2)
      B1=2.D0*Z(N1)+D(N1)
      A22=ALFA+2.D0+A1
      A12=-1.D0+A2
      A21=-3.D0-(A1+A(N1,1,1))
      A11=2.D0-(A2+A(N1,1,2))
      Z1=ALFA*Z(N)+B1
      D1=-(B1+Z(N1))
C     RESOLUTION DU SYSTEME TRIANGULAIRE SUPERIEUR PAR BLOC
      DET=A11*A22-A12*A21
      IF (DET.EQ.0.D0)GO TO 5
      A11=A11/DET
      A12=-A12/DET
      A21=-A21/DET
      A22=A22/DET
      A(N,1,1)=A11
      A(N,1,2)=A12
      A(N,2,1)=A21
      A(N,2,2)=A22
      Z(N)=A11*Z1+A12*D1
      D(N)=A21*Z1+A22*D1
      DO 3 J=1,N1
      I=N-J
      Z(I)=Z(I)-(A(I,1,1)*Z(I+1)+A(I,1,2)*D(I+1))
      D(I)=D(I)-(A(I,2,1)*Z(I+1)+A(I,2,2)*D(I+1))
3     CONTINUE
      DO 4 I=1,N
      D(I)=D(I)/H(I)
4     CONTINUE
      LOUP=.TRUE.
      RETURN
5     LOUP=.FALSE.
      RETURN
      END
```

```
      SUBROUTINE AJUSP2(N,X,RO,F,FP,BOOL)
C
C     CE PROGRAMME CALCULE LA FONCTION-SPLINE D'AJUSTEMENT D'ORDRE 2,S,
C     RELATIVE AUX POINTS X(I) I=1,...N QUI LISSE LES VALEURS ZI . CES
C     VALEURS SONT INTRODUITES PAR LE TABLEAU F .
C
C     N <= 1000
C
C     X(I),F(I),FP(I)  I=1,...N .
C     RO          PARAMETRE DE LISSAGE . DOIT ETRE > 1
C     F(I)        CONTIENT EN ENTREE LA VALEUR ZI
C                 CONTIENT EN SORTIE LA VALEUR EN X(I) DE LA FONCTION-
C                 SPLINE S .
C     FP(I)       VALEUR EN X(I) DE S' .
C     BOOL=.FALSE.  SI CETTE METHODE NE PEUT ETRE MENEE A SON TERME
C                   (DETERMINANT=0 EN COURS D'EXECUTION) .
      DIMENSION X(1),F(1),FP(1)
      LOGICAL BOOL
      REAL MO2,MJ1,MJ2,MJ3,MJ4,MJ5
      REAL M(2,5)
      DIMENSION E(2,2),G(2,2),R11(1000),R12(1000),R21(1000),R22(1000)
C
C     INITIALISATION
C
      DO 1 J=1,5
      M(1,J)=0.
1     M(2,J)=0.
      M(1,3)=1.
      M(2,1)=1.
      M(2,4)=1./RO
      M(2,5)=F(1)
C
C     TRANSPORT
C
      N1=N-1
      DO 2 I=1,N1
      H1=X(I+1)-X(I)
      H2=H1*H1/2.
      H3=H1*H2/3.
      DO 3 J=1,2
      G(J,1)=H1*M(J,3)-2*M(J,4)
      E(J,1)=H3*M(J,1)-G(J,I)
      P=2*H2*M(J,3)/3.
      G(J,2)=H1*M(J,4)-P
3     E(J,2)=H3*M(J,2)+G(J,2)-P
      FI=H3*M(1,5)
      FPI=H3*M(2,5)
      P=E(1,1)
      Q=E(2,2)
      DET=P*Q-E(1,2)*E(2,1)
      IF (DET.EQ.0.) GOTO 20
      E(1,1)=Q/DET
      E(2,2)=P/DET
      E(1,2)=-E(1,2)/DET
      E(2,1)=-E(2,1)/DET
```

```
      R11(I)=F(1,1)*G(1,1)+F(1,2)*G(2,1)
      R12(I)=F(1,1)*G(1,2)+F(1,2)*G(2,2)
      R21(I)=F(2,1)*G(1,1)+F(2,2)*G(2,1)
      R22(I)=F(2,1)*G(1,2)+F(2,2)*G(2,2)
      F(I)=F(1,1)*FI+F(1,2)*FPI
      FP(I)=F(2,1)*FI+F(2,2)*FPI
      DO 4 J=1,2
      MJ4=M(J,4)-H1*M(J,3)+H2*M(J,2)-H3*M(J,1)
      P=RO*MJ4
      MJ3=M(J,3)-H1*M(J,2)+H2*M(J,1)
      MJ2=M(J,2)-H1*M(J,1)
      MJ1=M(J,1)+P
      MJ5=M(J,5)+F(I+1)*P
      NOR=ABS(MJ1)+ABS(MJ2)+ABS(MJ3)+ABS(MJ4)
      M(J,1)=MJ1/NOR
      M(J,2)=MJ2/NOR
      M(J,3)=MJ3/NOR
      M(J,4)=MJ4/NOR
      M(J,5)=MJ5/NOR
 4    CONTINUE
 2    CONTINUE
C
C   RESOLUTION DU SYSTEME TRIANGULAIRE SUPERIEUR PAR BLOC
C
      DET=M(1,1)*M(2,2)-M(1,2)*M(2,1)
      F(N)=(M(1,5)*M(2,2)-M(2,5)*M(1,2))/DET
      FP(N)=(M(2,5)*M(1,1)-M(1,5)*M(2,1))/DET
      DO 5 I=1,NI
      NI=N-I
      P=R11(NI)*F(NI+1)+R12(NI)*FP(NI+1)
      Q=R21(NI)*F(NI+1)+R22(NI)*FP(NI+1)
      F(NI)=F(NI)-P
 5    FP(NI)=FP(NI)-Q
      BOOL=.TRUE.
      RETURN
 20   BOOL=.FALSE.
      RETURN
      END
```

REFERENCES

[1] ALBERT J.H., NILSON E.N. and Walsh J.L.
"The theory of splines and their applications"
Academic Press N.Y. (1967)

[2] ALBERT J.H.
"Cardinal of odd degree on uniforms meshes"
Journal of approximation theory, 5, 428-437 (1972)

[3] ASTOR P.H. and DURIS C.S.
"Discrete L-splines"
Numerische Mathematik, 22, 393-402 (1974)

[4] BARRAR R.B. and LOIB H.L.
"Analytic extended monosplines"
Numerische Mathematik, 22, 119-125 (1974)

[5] CARL DE BOOR
"On calculating with B-splines"
Journal of approximation theory, 6, 50-60 (1972)

[6] DUCHON J.
"Functions-spline et vecteurs aléatoires"
Séminaire d'Analyse Numérique n° 213, USMG Grenoble 9-1-1975

[7] DUC-JACQUET M.
"Une propriété de convergence des fonctions-spline d'interpolation d'ordre 2 basée sur des noeuds équidistants"
Séminaire d'Analyse Numérique n° 219 - USMG Grenoble 20-2-75

[8] DUC-JACQUET M
"Résolution de quelques problèmes d'analyse numérique linéaire à l'aide de perturbations par des matrices antiscalaires".
Thèse de 3ème Cycle USMG Grenoble 1968

[9] GREVILLE T.N.E
"Spline functions, interpolation and numerical quadrature"
Mathematics Research Center, US Army - University of Wisconsin.

[10] JERONE J.W. and SHUMAKER L.L.
"Local support bases for a class of splines functions".
Journal of approximation theory 16, 16-27 (1976)

[11] HALLER D.
"Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems"
Siam J. Numerical Analysis Vol. 13, n° 4, september 1976.

[12] LAURENT P.J.
"Approximation et Optimisation".
Hermann Paris 1972

[13] LYCHE J. and SUMAKER L.L.
"Local spline approximation methods"
Journal of approximation theory 15, 294-325 (1975)

[14] MUND E.H., HALLET P. and HENNART
"Algorithm for the interpolation of arising quintic splines"
Journal of computational and Appli. Mathematics Vol I, n° 4 1975

[15] MUNTEANU M.J. and SHUMAKER L.L.
"On a method of Carasso and Laurent constructing interpolation splines'
Mathematics of computation, vol. 27, n° 122, April 1973

[16] PAIHUA L.
"Comparaison numérique de trois méthodes d'obtention des fonctions-spline d'ordre deux"
Rapport D.E.A. Analyse numérique USMG Grenoble juin 1976

[17] PASSOW E.
"Piecewice monotone spline interpolation"
Journal of Approximation Theory 12, 240-241 (1974)

[18] PICHAT M.
"Contribution à l'étude des erreurs d'arrondi en arithmétique à virgule flottante".
Thèse d'Etat USMG Grenoble (1976)

[19] ROBERT F.
"Matrices non négatives et normes vectorielles"
Cours D.E.A. Analyse Numérique USMG Grenoble 1975-1976

[20] SCHOENBERG I.J.
"Cardinal interpolation and spline functions"
Journal of approximation theory 2, 167-206 (1969)

[21] SHISHA O.
"Note on saturation with splines"
Journal of approximation theory 13, 491-494 (1975)

[22] SHULTZ M.H.
"Spline Analysis"
Printice Hall international 1973

[23] SPATH H.
"Spline Algorithmen zurkonstouktions glatter kurven und flächen"
R. Oldenbourg Verlang München Wien (1973)

[24] UTRERAS F.
"Sur le choix du paramètre d'ajustement dans le lissage par fonctions-splines".
Séminaire d'Analyse Numérique USMG Granoble 9-3-1978

[25] VARAH J.M.
"On the solution of block-tridiagonal systems arising from certain finite difference equations"
Mathematics of computation, Vol. 26 n° 120, cotober 1972

[26] VARGA R.S.
"Matrix Iterative Analysis"
Printice Hall Englewood Cliff. N.J. 1962

[27] VIGNES J.
"Erreurs dues à l'ordinateur"
Congrès d'Analyse Numérique de Port-Bail, mai 1976

[28] WAHBA G.
"Smothing Noisy data with spline functions"
Numerische Mathematik, 24, 383-393 (1975)

[29] WAHBA G.
"A survey of some smoothing problems and the method of generalized validation for solving them".
Technical report n° 457 University of Wisconsin Madison, July 1976

[30] WAHBA G.
"Optimal smoothing of density estimates"
Tec hnical report n° 469 University of Wisconsin Madison, October 1976

[31] WENDROFF B.
"Theorical Numerical Analysis"
Academic Press N.Y. 1966

[32] WILKINSON J. H.
"Rounding errors in algebraic processes"
Prentice Hall 1963.

DEUXIEME PARTIE

ALGORITHMES D'OBTENTION NUMERIQUE DES FONCTIONS SPLINE A DIMENSION $d \geq 2$

CHAPITRE - I

FONCTION SPLINE DU TYPE PLAQUE MINCE

EN DIMENSION DEUX

I.1 - INTRODUCTION

Les fonctions splines du type plaque mince en dimension deux, ont été étudiées par Atteia [2], Duchon [11] [12] et Thomann [40].

Parmi les fonctions v qui prennent des valeurs imposées Z_i $i=1,\dots,n$ aux points $t^i = (x_i, y_i)$ du plan on cherche celle qui minimise la fonctionnelle

$$(1.1) \qquad l(v) = \int_\Omega |\nabla^2 v|^2 = \int_\Omega |D_x^2 v + 2D_x D_y v + D_y^2 v|^2$$

Le cas Ω rectangulaire ou circulaire a été développé par Thomann [40] et pour le cas $\Omega = \mathbb{R}^2$ Duchon [11] obtient la caractérisation suivante :

<u>Théorème 1</u>

Parmi les fonctions continues sur $\mathbb{R}^2$ prenant des valeurs $Z_1,\dots,Z_n$ en des points $t^1,\dots,t^n$ données (non tous alignés) il existe une et une seule fonction σ, qui réalise le minimum de la fonctionnelle (1.1) (avec $\Omega = \mathbb{R}^2$) et σ s'écrit d'une manière unique sous la forme :

$$(1.2) \qquad \sigma(t) = \sum_{i=1}^{n} \lambda_i \, \mathcal{K}(t^i,t) + \alpha_1 x + \alpha_2 y + \beta$$

avec

$$\begin{cases} \sum_{i=1}^{n} \lambda_i = 0 \\ \sum_{i=1}^{n} \lambda_i t^i = 0 \end{cases}$$

où

$$t = (x,y) \qquad \mathcal{K}(t^i,t) = |t-t^i|^2 \operatorname{Log} |t-t^i|^2 \qquad (*)$$

$$t^i = (x_i,y_i)$$

(*) $|.|$ norme euclidienne dans $\mathbb{R}^2$.

On peut donc obtenir les coefficients $\lambda_1,\dots,\lambda_n,\alpha_1,\alpha_2,\beta$ en résolvant le système

$$(1.3) \qquad \begin{bmatrix} K & E \\ E^T & O \end{bmatrix} \begin{bmatrix} \Lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} Z \\ O \end{bmatrix}$$

où

$$K = [k_{ij}]_{n\times n} \qquad k_{ij} = \mathcal{K}(t^i,t^j) \qquad k_{ii} = 0$$

$$E^T = \begin{bmatrix} 1,\dots,1 \\ x_1,\dots,x_n \\ y_1,\dots,y_n \end{bmatrix} \qquad Z^T = [z_1,\dots,z_n]$$

$$\Lambda^T = [\lambda_1,\dots,\lambda_n] \qquad \alpha^T = [\beta,\alpha_1,\alpha_2]$$

La matrice K sera appelée matrice du noyau reproduisant. (voir [12][44])

Nous noterons $\|\cdot\|$ la norme euclidienne dans $\mathbb{R}^n$ (ou $\mathbb{R}^{n-3}$) ou norme matricielle associée, selon le cas.

Soient $\mathcal{E} \subset \mathbb{R}^n$ le sous-espace engendré par les trois vecteurs colonnes de E, $\mathcal{F} = \mathcal{E}^\perp \subset \mathbb{R}^n$ le supplémentaire orthogonal (par rapport au produit scalaire habituel).

Notre but est de mettre au point des algorithmes de résolution du système (1.3) en utilisant au maximum les propriétés de la matrice K, et en étudiant les deux cas suivants :

1°) Aucune hypothèse n'est faite sur la numérotation des données $t^i \quad i = 1,\dots,n$

2°) On admet une hypothèse faible sur la numérotation des données $t^i \quad i = 1,\dots,n$ hypothèse non contraignante dans la pratique.

En utilisant les propriétés de K que nous étudions dans le paragraphe suivant, et le système (1.3) qui est assez particulier, nous allons proposer deux méthodes directes (Méthodes (A) et (B) et deux méthodes itératives (Méthodes (C) et (D)).

Les deux méthodes directes utilisent le fait que $\mathcal{F}$ est isomorphe à $\mathbb{R}^{n-3}$ et que la solution Λ cherchée est orthogonale à $\mathcal{E}$ (c'est-à-dire $\Lambda \in \mathcal{F}$).

Dans la méthode (A) on construit une base $r_1, \ldots, r_{n-3}$ de $\mathcal{F}$ qui nous permet de faire un changement de variable dans (1.3), ce qui ajouté à la condition d'orthogonalité de Λ à $\mathcal{E}$, nous ramène d'un système à (n+3) inconnues (1.3) à un système à (n-3) inconnues (cf. 1.5, I-3) avec une matrice définie positive ; la partie délicate est la construction de la base.

En effet, pour cela il nous faut une hypothèse sur la numérotation des données, hypothèse qui sera discutée en détail, mais il faut souligner que l'avantage essentiel de cette méthode est "la définie positivité" de la matrice du nouveau système.

La méthode (B) construit l'application projection orthogonale de $\mathbb{R}^n$ sur $\mathcal{F}$ qui nous permet de ramener le système (1.3) à un système à n inconnues (cf. 1.11, I-3) avec une matrice semi-définie positive de rang égal à (n-3). Ce qui est alors intéressant c'est que nous n'avons aucune hypothèse sur la numérotation des données.

De plus, l'encombrement est du même ordre que dans la méthode (A), cependant la résolution va être un peu délicate (voir [14] [15] [22] [24], comme nous le verrons.

La méthode (C) est une méthode de descente dans des directions fixées de façon cyclique (type Gauss-Seidel) voir [21][32]. Ces directions sont celles des r_i $i = 1, \ldots, n-3$ construits comme dans la méthode (A).

La méthode (D) est la méthode du gradient avec contraintes linéaires (Gradient sur $\mathcal{F}$ où gradient projeté) ; nous citons aussi quelques accélérations de chacune des méthodes itératives.

Nous indiquons enfin comment les quatre méthodes directes sont adaptées dans le cas des splines d'ajustement de type plaque mince.

I.2 - QUELQUES PROPRIETES DE LA MATRICE K

Comme conséquence directe des propriétés du noyau reproduisant $\mathcal{H}$ de $X^{2,0}$ (voir [11], cf. chapitre II) nous avons :

Propriété 1 :

$$\forall x \in \mathcal{F} \quad x^T K x > 0$$
$$x \neq 0$$

Autrement dit K est définie positive sur $\mathcal{F}$, mais si $x \notin \mathcal{F}$ la valeur $x^T K x$ peut être négative, de plus la matrice K peut être singulière.

Soit $M \in \mathcal{M}_{n,n}(\mathbb{R})$ (M symétrique)

Définitions : (voir [21] [32])

1- M est coercive sur $\mathcal{F}$ s'il existe un réel $\mu > 0$ tel que :

$$\forall \Lambda \in \mathcal{F} \qquad \mu \Lambda^T \Lambda \leq \Lambda^T M \Lambda$$

2- On appelle conditionnement de M sur $\mathcal{F}$ la valeur $\gamma_{\mathcal{F}}(M) = \frac{\mu}{\beta}$ où

$$\mu = \underset{\substack{\Lambda \in \mathcal{F} \\ \neq 0}}{\mathrm{Min}} \frac{\Lambda^T M \Lambda}{\Lambda^T \Lambda} \qquad \beta = \|M\|_{\mathcal{F}} = \underset{\substack{\Lambda \in \mathcal{F} \\ \neq 0}}{\mathrm{Sup}} \frac{\|M\Lambda\|}{\|\Lambda\|}$$

Proposition 1 :

K est coercive sur $\mathcal{F}$.

Démonstration :

Considérons $c = \{x \in \mathbb{R}^n \,/\, x = \frac{\Lambda}{\|\Lambda\|} \quad \Lambda \underset{\neq 0}{\in} \mathcal{F}\}$.

L'ensemble c est fermé et borné. Comme K est définie positive sur $\mathcal{F}$, la fonctionnelle $\mu(\Lambda) = \frac{\Lambda^T K \Lambda}{\Lambda^T \Lambda}$ est positive et continue. De plus son minimum sur $\mathcal{F}$ est atteint sur c.

Cela démontre la proposition.

Ensuite nous allons utiliser la propriété de K et étudier le système (1.3)

Soit $G = \begin{bmatrix} K & E \\ E^T & O \end{bmatrix}$

<u>Proposition 2</u> :

G est définie positive sur $\mathcal{F} \times \mathbb{R}^3$

<u>Démonstration</u> :

Si $x \in \mathcal{F} \times \mathbb{R}^3$ soit $x = \begin{bmatrix} \Lambda \\ \alpha \end{bmatrix}$ avec $\Lambda \in \mathcal{F}$ $\alpha \in \mathbb{R}^3$

on voit que l'on a $x^T G x = \Lambda^T K$ et d'après la proposition 1, on obtient $x^T G x >$

Maintenant si $\{r_1, \ldots, r_{n-3}\}$ est une base (quelconque) de $\mathcal{F}$, formons $R = [r_1, \ldots, r_{n-3}]$.
(R matrice base) $R \in \mathcal{M}_{n \times (n-3)}(\mathbb{R})$

Considérons : $R : \mathbb{R}^{n-3} \to \mathcal{F}$

$$x \rightsquigarrow \Lambda = Rx$$

Alors si

$$A = R^T K R \qquad A \in \mathcal{M}_{(n-3)\times(n-3)}(\mathbb{R}) \text{ on constate que}$$

$$\forall x \in \mathbb{R}^{n-3}_{\neq 0} \qquad x^T A x = \Lambda^T K \Lambda > 0 \qquad \text{où} \quad \Lambda = Rx$$

On a donc la propriété suivante :

<u>Propriété 2</u> :

Quelque soit la base prise dans $\mathcal{F}$, la matrice A formée comme ci-dessus, est définie positive.

Alors comme conséquence on a (voir [17] [32])

$\exists \mu > 0$ et $\beta > 0$ tels que quelque soit $x \in \mathbb{R}^{n-3}$

$$\begin{cases} x^T A x \leq \beta x^T x & \text{(continuité)} \\ x^T A^{-1} x \leq \frac{1}{\mu} x^T x & \text{(coercivité)} \end{cases}$$

$$\|A\| \leq \beta$$

$$\|A^{-1}\| \leq \frac{1}{\mu}$$

Ensuite on exhibe une borne pour le conditionnement de A, quand la base de $\mathcal{F}$ orthonormée.

<u>Proposition 3</u> :

Si la base $\{r_1,\ldots,r_{n-3}\}$ de $\mathcal{F}$ est orthonormée, alors on a :

i) $\|A\| \leq \|K\|_{\mathcal{F}}$

ii) $\|A^{-1}\| = \frac{1}{\mu}$ avec $\mu = \underset{\substack{\Lambda \in \mathcal{F} \\ \neq 0}}{\mathrm{Min}} \frac{\Lambda^T K \Lambda}{\Lambda^T \Lambda}$

et donc

$$\boxed{\gamma(A) \geq \gamma_{\mathcal{F}}(K)}$$

<u>Démonstration</u> :

i) $$\|A\| = \sup_{\substack{x \in \mathbb{R}^{n-3} \\ \neq 0}} \frac{\|Ax\|}{\|x\|} = \sup_{\substack{x \in \mathbb{R}^{n-3} \\ \neq 0}} \frac{\|R^T K R x\|}{\|x\|}$$

si $\Lambda = Rx$ alors $x = R^T\Lambda$ car R est orthogonal par colonnes, de plus on a $\|R\| = \|R^T\| = 1$

Comme $\|x\| = \|\Lambda\|$ alors $\frac{\|R^T K R x\|}{\|x\|} \leq \frac{\|K\Lambda\|}{\|\Lambda\|}$

d'où $$\|A\| \leq \sup_{\substack{\Lambda \in \mathcal{F} \\ \neq 0}} \frac{\|K\Lambda\|}{\|\Lambda\|} = \|K\|_{\mathcal{F}}$$

ii) La coercivité de A implique

$\forall x \in \mathbb{R}^{n-3}$ $\quad \mu x^T x \leq x^T A x$ $\quad$ (μ est la plus petite valeur propre de A)

mais

$x^T A x = \Lambda^T K \Lambda$ $\quad$ avec $\Lambda = Rx$ $\quad$ ($x = R^T\Lambda$)

alors

$$\mu \Lambda^T \Lambda = \mu x^T x \leq x^T A x = \Lambda^T K \Lambda \qquad \forall \Lambda \in \mathfrak{F}$$

ce qui implique

$$\mu = \underset{\substack{\Lambda \in \mathfrak{F} \\ \neq 0}}{\text{Min}} \frac{\Lambda^T K \Lambda}{\Lambda^T \Lambda}$$

on a donc

$$\|A^{-1}\| = \frac{1}{\mu} \qquad \text{avec } \mu \text{ vérifiant l'égalité ci-dessus}$$

I.3 - METHODES NUMERIQUES

METHODE (A)

Considérons $R = [r_1, \ldots, r_{n-3}]$ r_i $i = 1, \ldots, n-3$ étant une base quelconque de $\mathfrak{F}$. Si $\Lambda \in \mathfrak{F}$ alors il existe $\gamma_1, \ldots, \gamma_{n-1}$ réels (uniques) tel que :

$$(1.4) \qquad \Lambda = R\gamma \qquad \gamma^T = [\gamma_1, \ldots, \gamma_{n-3}]$$

et on a alors

$$K\Lambda + E\alpha = Z \qquad R^T K \Lambda + \underbrace{R^T E \alpha}_{0} = R^T Z$$

d'où finalement on a :

$$(1.5) \qquad \boxed{A\gamma = b} \qquad \text{avec } A = R^T K R \qquad b = R^T Z$$

Ainsi pour résoudre le problème (1.3) à (n+3) inconnues on peut résoudre le système (1.5) à (n-3) inconnues puis la solution Λ^* cherchée sera obtenue au moyen de la relation (1.4) et enfin pour trouver α^* on utilisera trois relations indépendantes de $K\Lambda + E\alpha = Z$ (ce qui sera explicité plus loin).

Remarquons aussi que la matrice A est définie positive, alors que G ne l'était pas, donc on a un système plus favorable, dans ces conditions on peut utiliser soit une méthode directe soit une méthode itérative ; on va utiliser la décomposition de Choleski pour résoudre (1.5).

Jusqu'à maintenant nous avons considéré une base quelconque pour $\mathfrak{F}$, mais dans la pratique nous devons construire une base qui tienne compte au moins :

i) du coût d'obtention de A et de R

ii) de la mémoire nécessaire pour stocker R

iii) de la numérotation des données t^i $\quad i = 1,\dots,n$

Si on fait l'hypothèse que la numérotation est telle que trois points consécutifs sont non alignés alors on peut construire la base de la façon suivante :

$$r_1 = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad r_i = \begin{bmatrix} 0 \\ 0 \\ c_1^i \\ c_2^i \\ c_3^i \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \\ \rightarrow (i+3)\text{ème composante} \\ \\ \\ \\ \end{array} \quad i = 2,\dots,n-3$$

avec c_1^i, c_2^i, c_3^i vérifiant la relation $E^T r_i = 0 \qquad i = 1,\dots,n-3.$

Ceci correspond à la méthode présentée par Thomann dans [40], mais l'hypothèse faite est ennuyeuse ; de plus, il y a des cas où elle n'est pas réaliste.

Faisons plus simplement l'hypothèse :

(H) $\boxed{\text{les trois premiers points } (t^1,t^2,t^3) \text{ sont non alignés}}$

Sous l'hypothèse H on peut fabriquer la base suivante :

$$(1.6)\quad r_i = \begin{bmatrix} c_1^i \\ c_2^i \\ c_3^i \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \\ \\ \rightarrow \text{ (i+3) ème composante} \\ \\ \\ \\ \end{array} \qquad i = 1,\dots,n-3$$

c_1^i, c_2^i, c_3^i étant déterminés à partir de $E^T r_i = 0$, c'est-à-dire étant solution du système :

$$(1.7)\qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & x_2-x_1 & x_3-x_1 \\ 0 & y_2-y_1 & y_3-y_1 \end{bmatrix} \begin{bmatrix} c_1^i \\ c_2^i \\ c_3^i \end{bmatrix} = \begin{bmatrix} -1 \\ x_1-x_{i+3} \\ y_1-y_{i+3} \end{bmatrix} \qquad i=1,\dots,n-3$$

"$-c_1^i, -c_2^i, -c_3^i$ sont les coefficients barycentriques du point t^{i+3} par rapport aux trois premiers points".

Explicitons A et b pour le choix ci-dessus

$$A = R_1^T K_{1,1} R_1 + K_{2,1} R_1 + R_1^T K_{1,2} + K_{2,2}$$

où

$$R = \begin{bmatrix} R_1 \\ \\ I_{n-3} \end{bmatrix} \qquad K = \begin{bmatrix} K_{1,1} & K_{1,2} \\ \\ K_{2,1} & K_{2,2} \end{bmatrix}$$

$R_1 \in \mathcal{M}_{3,n-3}(\mathbb{R})$ $\quad K_{1,1} \in \mathcal{M}_{3,3}(\mathbb{R})$ $\quad K_{1,2} = K_{2,1}^T \in \mathcal{M}_{3,(n-3)}(\mathbb{R})$

$K_{2,2} \in \mathcal{M}_{(n-3),(n-3)}(\mathbb{R})$

si $A = [a_{ij}]$, on a

$$a_{ij} = c_1^i[c_2^j k_{1,2}+c_3^j k_{1,3}+k_{1,j+3}]+$$
$$c_2^i[c_1^j k_{1,2}+c_3^j k_{2,3}+k_{2,j+3}]+$$
$$c_3^i[c_1^j k_{1,3}+c_2^j k_{2,3}+k_{3,j+3}]+$$
$$c_1^i k_{1,i+1}+c_2^i k_{2,i+3}+k_{j+3,i+3} \qquad i,j = 1,\dots,n-3$$

avec $k_{ii} = 0$

$$b = R_1^T Z + \sum_{i=1}^{n-3} Z_{i+3} e_i \qquad e_i \in \mathbb{R}^{n-3}$$

si $b = [b_i]$ alors la valeur de b_i est

$$b_i = c_1^i Z_1+c_2^i Z_2+c_3^i Z_3+Z_{i+3} \qquad i = 1,\dots,n-3$$

De plus si $\gamma^* = [\gamma_1^*,\dots, \gamma_{n-3}^*]^T$ est solution du système (1.5) alors la solution Λ^* vaut :

$$(1.8) \quad \begin{cases} \lambda_i^* = \sum_{j=1}^{n-3} c_1^j \gamma_j^* & i = 1,2,3 \\ \lambda_i^* = \gamma_{i-3}^* & i = 4,\dots,n \end{cases}$$

On en déduit que la solution cherchée dépend du choix des trois premiers points; changer l'énumération des autres (n-3) points est permuter les (n-3) dernieres composants de λ_i^*

Pour trouver $\alpha^* = [\beta^*,\alpha_1^*,\alpha_2^*]$ on utilise les trois premières relations c'est-à-dire α^* est solution de :

$$(1.9) \quad \begin{bmatrix} 1 & x_1 & y_1 \\ 0 & x_2-x_1 & y_2-y_1 \\ 0 & x_3-x_1 & y_3-y_1 \end{bmatrix} \begin{bmatrix} \beta^* \\ \alpha_1^* \\ \alpha_2^* \end{bmatrix} = \begin{bmatrix} Z_1 - \sum_{i=1}^{n} \lambda_i k_{1,i} \\ Z_2-Z_1 + \sum_{i=1}^{n} \lambda_i (k_{1,i}-k_{2,i}) \\ Z_3-z_1 + \sum_{i=1}^{n} \lambda_i (k_{1,i}-k_{3,i}) \end{bmatrix}$$

Proposition 6 :

Soit $A = R^T K R$ (matrice du système (1.5)) R matrice $n \times (n-3)$ dont les vecteurs colonnes sont de la forme (1.6) et construite en utilisant (1.7) alors

(1.10) $\gamma(A) \geq \frac{1}{\rho^2} \gamma_{\mathcal{F}}(K)$ où ρ est le rayon spectral de (RR^T)

Démonstration :

Soit $x \in \mathbb{R}^{n-3}$ alors $\|Ax\| = \|R^T K R x\| \leq \|R^T\| \, \|K\Lambda\|$

avec

$\Lambda = Rx \in \mathcal{F}$ mais $\|\Lambda\| \leq \|R\| \, \|x\|$

par conséquent

$$\frac{\|Ax\|}{\|x\|} \leq \frac{\|R^T\| \, \|K\Lambda\| \, \|R\|}{\|\Lambda\|}$$

d'où $\|A\| \leq \|R\| \, \|R^T\| \, \|K\|_{\mathcal{F}}$ mais $\|R\| = \|R^T\| \leq \rho$

alors on obtient

$$\|A\| \leq \rho^2 \|K\|_{\mathcal{F}}$$

Posons $\|A^{-1}\| = \frac{1}{\mu_A}$, où μ_A est la plus petite valeur propre de A qui vérifie $\mu_A x^T x \leq x^T A x$ $\forall x \in \mathbb{R}^{n-3}$; d'après la proposition 1 $\exists \mu > 0$ tel que

$$\mu = \underset{\substack{\Lambda \in \mathcal{F} \\ \neq 0}}{\mathrm{Min}} \frac{\Lambda^T K \Lambda}{\Lambda^T \Lambda}$$

en faisant toujours $\Lambda = Rx$ on a

$$\mu \leq \frac{\Lambda^T K \Lambda}{\Lambda^T \Lambda} \leq \frac{x^T A x}{x^T x} \qquad (\|x\| \leq \|\Lambda\| \text{ cf. } (1.8))$$

alors $\mu \leq \mu_A$

ce qui entraine

$$\|A^{-1}\| \leq \frac{1}{\mu}$$

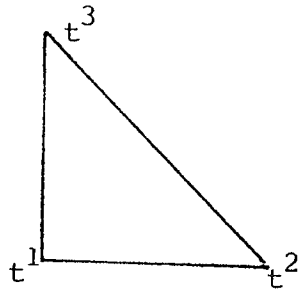
on a donc $\gamma(A) = \frac{1}{\|A\| \, \|A^{-1}\|} \geq \frac{\mu}{\rho^2 \|K\|_{\mathcal{F}}}$ $\gamma(A) \geq \frac{1}{\rho^2} \gamma_{\mathcal{F}}(K)$

Comme la solution numérique du système (1.5) dépend du choix des trois premiers points et non de la numérotation des points t^i $i=4,\ldots,n$, le choix idéal des trois premiers points serait celui qui rend $\gamma(A)$ le plus grand possible mais ce choix est presque impossible dans la pratique.

D'autre part, la partie délicate est de savoir si les coefficients C_1^i, C_2^i, C_3^i seront bien calculés, c'est-à-dire si A est correctement calculée. choisirons ces points de façon que le conditionnement du système (1.7) (en réalité le système formé par les deux dernières relations) soit bon.

Par exemple, si le triangle formé par ces points est :

1) Rectangle



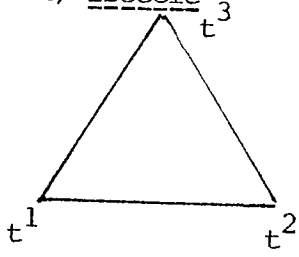
Après une rotation on peut avoir la situation suivante :

$$\begin{bmatrix} x_1-x_1 & x_3-x_1 \\ y_2-y_1 & y_3-y_1 \end{bmatrix} = \begin{bmatrix} x_2-x_1 & 0 \\ 0 & y_3-y_1 \end{bmatrix} = D$$

$$\gamma(D) = \frac{\min(|x_2-x_1|, |y_3-y_1|)}{\max(|x_2-x_1|, |y_3-y_1|)}$$

Si de plus on a même côté alors $\gamma(D) = 1$

2) Isocèle



$$D=\begin{bmatrix} x_2-x_1 & x_3-x_1 \\ y_2-y_1 & y_3-y_1 \end{bmatrix} = \begin{bmatrix} x_2-x_1 & \frac{x_2-x_1}{2} \\ 0 & y_3-y_1 \end{bmatrix}$$

soit $d_{11} = x_2-x_1 \quad d_{12} = \frac{1}{2} d_{11} \quad d_{22} = y_3-y_1 \quad d_{21} = 0$

$d_{11} > 0 \quad d_{22} > 0$

alors $$DD^T = \begin{bmatrix} d_{11}^2+d_{12}^2 & d_{12}d_{22} \\ d_{12}d_{22} & d_{22}^2 \end{bmatrix}$$

Les valeurs propres de DD^T sont

$$\lambda = \frac{u \pm \sqrt{u^2-4d_{11}^2 d_{22}^2}}{2} \qquad \text{où} \quad u = d_{11}^2+d_{12}^2+d_{22}^2$$

$$= \frac{5}{4}\, d_{11}^2+d_{22}^2$$

et

$$u^2-4d_{11}^2 d_{22}^2 = [\,(d_{11}-d_{22})^2+d_{12}^2][\,(d_{11}+d_{22})^2+d_{12}^2] > 0$$

posons $d_{22} = \theta\, d_{11} \qquad \theta > 0$

$$\lambda = \frac{d_{11}^2[(\frac{5}{4}+\theta^2) \pm \sqrt{(\frac{5}{4}+\theta^2)^2-4\theta^2}\ ]}{2}$$

si on note γ_θ le conditionnement de D on a :

$$[\gamma_\theta]^2 = \frac{\frac{5}{4}+\theta^2-\sqrt{(\frac{5}{4}+\theta^2)\ -4\theta^2}}{\frac{5}{4}+\theta^2+\sqrt{(\frac{5}{4}+\theta^2)^2-4\theta^2}} = \frac{(\frac{5}{4}+\theta^2-\sqrt{(\frac{5}{4}+\theta^2)^2-4\theta^2})^2}{4\theta^2}$$

d'où finalement :

$$\boxed{\gamma_\theta = \frac{\frac{5}{4}+\theta^2-\sqrt{(\frac{5}{4}+\theta^2)^2-4\theta^2}}{2\theta}}$$

La valeur de θ que rend maximum γ_θ est $\theta = \frac{\sqrt{5}}{2}$

$$\gamma_{\frac{\sqrt{5}}{2}} = \frac{\sqrt{5}-1}{2}$$

si $\theta = \frac{\sqrt{3}}{2}$ (c'est-à-dire triangle équilatéral) on a $\gamma_{\frac{\sqrt{3}}{2}} = \frac{1}{\sqrt{3}}$

Remarque

La méthode (A) est en fait (à des différences près de notation ou de présentati exactement la même que la "méthode générale" décrite dans [45] (voir aussi [46]

Dans cette méthode générale pour la construction de l'élément solution de $\min_{A(x)=z} \|T(x)\|$ (voir [11]) avec :

$$A(x) = [\langle K_1,x\rangle,\dots,\langle K_n,x\rangle] \qquad \langle K_i,x\rangle = \delta_{t^i}(x) = x(t^i)$$

T "bigradient" dans notre cas $\dim(\mathcal{N}(T)) = q = 3$

$x \in X =$ {fonctions continues sur $\mathbb{R}^2$ dont les dérivées secondes (au sens de distribution) sont dans $L^2(\mathbb{R}^2)$}

et soit $Y = \{y_{ij} \quad i,j = 1,2$ où $y_{ij} \in L^2(\mathbb{R}^2)\}$

On détermine :

1°) $h_j = \sum_{i=1}^{n} b_j^i k_i \qquad j = 1,\dots,n-q$ tels que $h_j \in \mathcal{N}(T)^{\perp}$

Les $b_j \quad j = 1,\dots,n-q$ sont les r_j base de $\mathcal{F}$.

2°) $f_j = T^{*-1}(h_j) \qquad j = 1,\dots,n-q$

On sait que la fonction $H_i(t) = \frac{1}{2\pi} |t-t^i|^2 \operatorname{Log} |t-t^i|$ vérifie $T^*TH_i = k_i$ c'est-à-dire $\Delta^2 H_i = \delta_{t^i}$

alors $\langle TH_i,TH_j\rangle = \langle T^*TH_i,H_j\rangle$

$= \langle K_i,H_j\rangle = K_{ji}$ termes de la matrice K

et :

$\forall x$ on a $\langle f_j,Tx\rangle = \langle T^*f_j,x\rangle = \langle h_j,x\rangle$

$$= \langle \sum_{i=1}^{n} b_j^i k_i,x\rangle = \sum_{i=1}^{n} b_j^i \langle T^*TH_i,x\rangle$$

$$= \sum_{i=1}^{n} b_j^i \langle TH_i,Tx\rangle$$

$$= \langle \sum_{i=1}^{n} b_j^i TH_i,Tx\rangle$$

donc $$f_j = \sum_{i=1}^{n} b_j^i \, TH_i$$

ainsi

$$\langle f_j, f_i \rangle = \langle \sum_{i=1}^{n} b_j^i \, TH_i, \sum_{i=1}^{n} b_j^i \, TH_i \rangle = R^T K R$$

où R a pour terme b_j^i

et K a pour terme $\langle TH_i, TH_j \rangle = K_{ij}$

donc la matrice $A = R^T K R$ du système (1.4) est en fait la même que la matrice F de terme $\langle f_j, f_i \rangle$.

Maintenant si $I_Z = \{x / A(x) = Z\}$ on peut alors décrire

$\Delta_Z = T(I_Z)$ par :

$$\Delta_Z = \{g \in Y \mid \langle g, f_j \rangle = \delta_j \quad j = 1, \ldots, n-q\}$$

où $$\delta_j = \sum_{i=1}^{n} b_j^i \, Z_i \qquad (\delta \text{ est le vecteur } b \text{ du système } (1.5))$$

alors $\bar{p} = T(\sigma)$ est la projection orthogonale de O sur Δ_Z et l'on a $\bar{p} = \sum_{j=1}^{n-q} \mu_j \, f_j$ avec μ_j solution du système linéaire $\sum_{j=1}^{n-q} \mu_j \, \langle f_j, f_i \rangle = \delta_i$ $i = 1, \ldots, n-q$.

Et donc μ est le vecteur γ du système (1.5)).

3°) On sait aussi que σ est solution si et seulement si $T^* T\sigma = \sum_{i=1}^{n} \lambda_i k_i$.(les λ_i sont les mêmes que dans (1.3) au 4π près)

ou encore

$$\langle T\sigma, Tx \rangle = \Sigma \lambda_i \langle K_i, x \rangle$$

alors

$$T^* T\sigma = T^*\left(\sum_{j=1}^{n-q} \mu_j - f_j\right) = \sum_{j=1}^{n-q} \mu_j \, T^* f_j = \sum_{j=1}^{n-q} \mu_j h_j$$

$$= \sum_{j=1}^{n-q} \mu_j \left(\sum_{i=1}^{n} b_j^i \, k_i\right) = \sum_{i=1}^{n} \lambda_i k_i$$

donc

$$\lambda_i = \sum_{j=1}^{n-q} b_j^i \, \mu_j$$

ce qui correspond à la relation $\Lambda = R\gamma$ dans la méthode (A).

```
      SUBROUTINE SMETA(N,RO,X,Y,Z,S,T,R,A,IER)
C
C     SPLINES D INTERPOLATION A 2 DIMENSIONS
C     N=NOMBRES DE POINTS  N<= 150
C     X,Y=COORDONNEES DES N POINTS
C     Z=VALEURS DE LA FONCTION AUX POINTS X,Y
C     RO FACTEUR DE PONDERATION EST 4*PI/RO
C  S(I) I=1,...,N ; T(1),T(2),T(3) SONT LES
C  COEFFICIENTS DE LA SPLINE
C  LA PARTIE TRIAGULAIRE SUPERIEUR DE LA MATRICE NOYAU EST
C  RANGE DANS  R  DIMENSION NR=N*(N-1)/2.
C  A : VECTEUR AUXILIAIRE   DIMENSION NA=(N-3)*(N-2)/2
C  SI IER<0 OU IER > 0 CHANGER LES TROIS PREMIERS POINTS
C
      IMPLICIT REAL*8(A-G,O-Z)
      DIMENSION X(1),Y(1),Z(1),S(1),A( 1 ),R( 1 ),T(3)
      DIMENSION C1(150),C2(150),C3(150),T(3)
      N3=N-3
C     CALCUL DE K(I,J) I<J
      DO 2 J=2,N
       IL=J-1
       IJ=(IL*(J-2))/2
       DO 3 I=1,IL
      A3=(X(I)-X(J))**2+(Y(I)-Y(J))**2
       R(IJ+I)=A3*DLOG(A3)
3     CONTINUE
2     CONTINUE
C     CALCULE C1,C2,C3(I)
      V1=Y(3)-Y(1)
      V4=X(2)-X(1)
      V2=X(1)-X(3)
      V3=Y(1)-Y(2)
      A1=V1*V4-V2*V3
      V1=V1/A1
      V2=V2/A1
      V3=V3/A1
      V4=V4/A1
      DO 4 I=1,N3
      A1=X(I)-X(I+3)
      A2=Y(I)-Y(I+3)
      C3(I)=V3*A1+V4*A2
      C2(I)=V1*A1+V2*A2
      C1(I)=-(1.D0+C2(I)+C3(I))
4     CONTINUE
C     ON CALCULE A(I,J) I<=J
      DO 5 I=1,N3
      I1=(I*(I+1))/2
      IJ=I1+I+2
      A1=C2(I)*R(1)+C3(I)*R(2)+R(IJ)
      A2=C3(I)*R(3)+R(IJ+1)
      A3=RO*(C1(I)*C1(I)+C2(I)*C2(I)+C3(I)*C3(I)+1.D0)
      A(I1)=2.D0*(C1(I)*A1+C2(I)*A2+C3(I)*R(IJ+2))+A3
5     CONTINUE
      DO 6 J=2,N3
      IL=J-1
```

```
      IJ=(J*(J+3)+4)/2
       IT=(J*IL)/2
      DO 7 I=1,IL
       I1=IT+I
      J1=(I*(I+3)+4)/2
      A1=C2(J)*R(1)+C3(J)*R(2)+R(IJ)
      A2=C1(J)*R(1)+C3(J)*R(3)+R(IJ+1)
      A3=C1(J)*R(2)+C2(J)*R(3)+R(IJ+2)
      A4=C1(J)*R(J1)+C2(J)*R(J1+1)+C3(J)*R(J1+2)+R(IJ+I+2)
      A5=R0*(C1(I)*C1(J)+C2(I)*C2(J)+C3(I)*C3(J))
      A(I1)=C1(I)*A1+C2(I)*A2+C3(I)*A3+A4+A5
 7    CONTINUE
 6    CONTINUE
 C    CALCUL DE B
      DO 8 I=1,N3
      S(I)=C1(I)*Z(1)+C2(I)*Z(2)+C3(I)*Z(3)+Z(I+3)
 8    CONTINUE
      HP=1.E-16
      CALL DMESD(A,N3,HP,IER)
       IF(IER)20,21,20
 21   S(1)=S(1)/A(1)
      DO 11 I=2,N3
      A1=0.000
      IL=I-1
      I1=(IL*I)/2
     DO 12 J=1,IL
 12   A1=A1+A(I1+J)*S(J)
      J1=I1+I
      S(I)=(S(I)-A1)/A(J1)
 11   CONTINUE
      I1=(N3*(N3+1))/2
      S(N3)=S(N3)/A(I1)
      NK=N3-1
      DO 13 IK=1,NK
      I=N3-IK
      I1=I1-I-1
      A1=0.000
      J1=I1
      IL=I+1
      DO 14 J=IL,N3
      J1=J1+J-1
      A1=A1+A(J1)*S(J)
 14   CONTINUE
      S(I)=(S(I)-A1)/A(I1)
 13   CONTINUE
 C    EVALUATION DE L(N+1).....L(N+3)
      A1=0.000
      A2=0.000
      A3=0.000
      DO 15 I=1,N3
      A1=A1+C1(I)*S(I)
      A2=A2+C2(I)*S(I)
      A3=A3+C3(I)*S(I)
 15   CONTINUE
      N1=N+1
```

```
      N2=N-2
      DO 16 I=1,N3
      S(N1-I)=S(N2-I)
16    CONTINUE
      S(1)=A1
      S(2)=A2
      S(3)=A3
      A1=S(2)*R(1)+S(3)*R(2)+RC*S(1)
      A2=S(1)*R(1)+S(3)*R(3)+RC*S(2)
      A3=S(1)*R(2)+S(2)*R(3)+RC*S(3)
      DO 17 I=4,N
      I1=(I*(I-3)+4)/2
      A1=A1+S(I)*R(I1)
      A2=A2+S(I)*R(I1+1)
      A3=A3+S(I)*R(I1+2)
17    CONTINUE
      A1=Z(1)-A1
      A2=Z(2)-(A2+A1)
      A3=Z(3)-(A3+A1)
      T( 1 )=V1*A2+V3*A3
      T( 2 )=V2*A2+V4*A3
      T( 3 )=A1-(X(1)*T(1)+Y(1)*T(2))
20    CONTINUE
      RETURN
      END
```

METHODE (B)

Soit V_1, V_2, V_3 une base orthonormée de $\mathcal{E}$ et considérons

$$V = [V_1, V_2, V_3] \qquad V \in \mathcal{M}_{n,3}(\mathbb{R})$$

$$Q = VV^T \qquad Q \in \mathcal{M}_{n,n}(\mathbb{R})$$

$$P = I - VV^T \qquad P \in \mathcal{M}_{n,n}(\mathbb{R})$$

On vérifie facilement que Q est la projection orthogonale de $\mathbb{R}^n$ sur $\mathcal{E}$ et donc P est la projection orthogonale de $\mathbb{R}^n$ sur $\mathcal{F}$.

Maintenant prenons le système (1.3)

$$K\Lambda + E\alpha = Z \text{ alors } PK\Lambda + \underbrace{PE}_{0}\alpha = PZ$$

et comme $P\Lambda = \Lambda$ on obtient $PKP\Lambda = PZ$, en notant

(1.10) $\quad A = PKP \qquad b = PZ$

On obtient :

(1.11) $\quad \boxed{A\Lambda = b}$

Si Λ^* est la solution de (1.3) on voit que l'ensemble de solutions de (1.11) est $\Lambda^* + \mathcal{E}$, alors il suffit de connaître une solution particulière de (1.11) pour calculer Λ^* au moyen de la projection P, donc notre but est de préciser une façon de calculer une solution particulière de (1.11) (par exemple celle de norme minimale).

La propriété de K implique la proposition :

Proposition 7 :

Soit A définie comme ci-dessus alors :

i) A est semi-définie positive (définie positive sur $\mathcal{F}$)

ii) $\gamma_{\mathcal{F}}(A) \geq \gamma_{\mathcal{F}}(K)$
et rang $(A) = n-3$

Démonstration :

i) Conséquence de la propriété de K et du fait que P est la projection orthogonale sur $\mathcal{F}$.

ii) Comme $\mathcal{F} = \mathcal{E}^{\perp}$ alors il existe une base $R = [r_1,\ldots,r_{n-3}]$ de $\mathcal{F}$, orthonor telle que $RR^T = P$.

$A = RR^TKRR^T$, comme l'ensemble de valeurs propres non nulles de A est égal à l'ensemble des valeurs propres de

$$R^TKRR^TR = R^TKR = \bar{A} \in \mathcal{M}_{(n-3)\times(n-3)}(\mathbb{R})$$

Cette matrice est définie positive (proposition 2) alors rang(A) = n-3 de plus on a $\gamma_{\mathcal{F}}(A) = \gamma(\bar{A})$.
Mais d'après la proposition 3 ($\gamma(A) \geq \gamma_{\mathcal{F}}(K)$) d'où $\gamma_{\mathcal{F}}(A) \geq \gamma_{\mathcal{F}}(K)$

La résolution du système (1.11) peut être menée de plusieurs façons par exemple voir [14][15][18] et [22][34][37] où on utilisera la propriété de semi-définie positivité de A. Nous allons utiliser la méthode de factorisation Choleski. Voir [35] [42] mais avec permutation de lignes et de colonnes (dans cas de pivot nul), ensuite on explique comment on utilise le résultat de la proposition 7 pour résoudre (1.11).

Comme conséquence de la semi-définie positivité de A on a :

i) $a_{ii} \geq 0 \qquad i = 1,\ldots,n$

ii) $|a_{ij}| = |e_i^TAe_j| \leq (e_i^TAe_i)^{1/2}(e_j^TAe_j)^{1/2} \leq \underset{i=1,\ldots,n}{\text{Max}} \{a_{ii}\}$

(1.12) $\Rightarrow \quad |a_{ij}| \leq \underset{i=1,\ldots,n}{\text{Max}}\{a_{ii}\} \qquad \forall i,j = 1,\ldots,n \qquad e_i,e_j \in \mathbb{R}^n$

et si $a_{11} > 0$ alors A admet la décomposition suivante (unique si le signe est positif) voir [9] [17]

$$(1.12) \qquad A = \begin{bmatrix} t & 0 \\ V^T & I_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} t & V \\ 0 & I_{n-1} \end{bmatrix}$$

Il suffit de prendre $t = \sqrt{a_{11}}$ $\quad V = \frac{1}{t} A_{12}$ $\quad D = A_{22} - V^T V$ où

$$A = \begin{bmatrix} a_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \qquad \begin{matrix} A_{21} = A_{12}^T \in \mathbb{R}^{n-1} \\ A_{22} \in \mathcal{M}_{(n-1),(n-1)} \end{matrix}$$

La matrice D est semi-définie positive voir [17] [21] [42], en tenant compte du rang de A et (1.12) on sait qu'il existe des éléments diagonaux positifs (sinon (1.12) implique D = 0). Supposons que

$$D = \begin{bmatrix} d_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \text{ avec } d_{11} > 0 \qquad \begin{matrix} D_{21} = D_{12}^T \in \mathbb{R}^{n-2} \\ D_{22} \in \mathcal{M}_{(n-2),(n-2)} \end{matrix}$$

Utilisant (1.12) on a : $D = \begin{bmatrix} s & 0 \\ U^T & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & D' \end{bmatrix} \begin{bmatrix} s & U \\ 0 & I \end{bmatrix}$ $\quad U^T \in \mathbb{R}^{n-2}$

alors

$$A = \begin{bmatrix} t & & 0 \\ V^T & s & \\ & U^T & I_{n-2} \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & D' \end{bmatrix} \begin{bmatrix} t & V \\ & s \quad U \\ 0 & I_{n-2} \end{bmatrix}$$

si $V = [v_1 v_2]$ $\quad v_1$ réel $\quad v_2^T \in \mathbb{R}^{n-2}$

soit $T = \begin{bmatrix} t & v_1 \\ 0 & s \end{bmatrix}$ $\quad W = \begin{bmatrix} v_2 \\ U \end{bmatrix}$ $\quad W \in \mathcal{M}_{2,(n-2)}(\mathbb{R})$

d'où $A = \begin{bmatrix} T^T & \\ W^T & I_{n-2} \end{bmatrix} \begin{bmatrix} I_2 & 0 \\ 0 & D' \end{bmatrix} \begin{bmatrix} T & W \\ 0 & I_{n-2} \end{bmatrix}$

Ce procédé on le fera (n-3) fois (rang de A) sous la réserve que chaque pivot soit positif , dans le cas contraire il faudrait permuter la ligne et colonne de façon convenable ; en conséquence on a :

"Si A est définie comme dans (1.10), il existe une matrice de permutatio $\mathcal{P}$ de façon que $\tilde{A} = \mathcal{P} A \mathcal{P}$ soit décomposable sous la forme"

$$(1.13) \qquad \tilde{A} = \begin{bmatrix} T^T & O \\ V^T & I_3 \end{bmatrix} \begin{bmatrix} I_{n-3} & O \\ O & O \end{bmatrix} \begin{bmatrix} T & V \\ O & I_3 \end{bmatrix}$$

avec T triangulaire supérieure non singulière.

Maintenant si on pose : $\tilde{b} = \mathcal{P} b$

$\tilde{\Lambda} = \mathcal{P} \Lambda$

et $\tilde{b} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \end{bmatrix}$ $\tilde{\Lambda} = \begin{bmatrix} \tilde{\Lambda}_1 \\ \tilde{\Lambda}_2 \end{bmatrix}$ $\begin{matrix} b_1 \text{ et } \tilde{\Lambda}_1 \in \mathbb{R}^{n-3} \\ \tilde{b}_2 \text{ et } \tilde{\Lambda}_2 \in \mathbb{R}^3 \end{matrix}$

alors (1.11) devient $\tilde{A}\tilde{\Lambda} = \tilde{b}$ et utilisant (1.13) on a :

$$\begin{cases} T^T T\tilde{\Lambda}_1 + T^T V\tilde{\Lambda}_2 = \tilde{b}_1 \\ V^T T\tilde{\Lambda}_1 + V^T V\tilde{\Lambda}_2 = \tilde{b}_2 \end{cases} \qquad \begin{cases} T^T T(\tilde{\Lambda}_1 + U\tilde{\Lambda}_2) = \tilde{b}_1 \\ U^T T^T T\tilde{\Lambda}_1 + U^T T^T T U\tilde{\Lambda}_2 = \tilde{b}_2 \end{cases} \quad \text{avec } U = T^{-1}V$$

on en déduit

$$(1.13') \quad \begin{cases} \tilde{\Lambda}_1 = (T^T T)^{-1}\tilde{b}_1 - U\tilde{\Lambda}_2 \\ U^T\tilde{b}_1 = \tilde{b}_2 \end{cases}$$

(dans notre cas, système consistant).

Alors on peut trouver une solution particulière en fixant $\tilde{\Lambda}_2$. Nous allons considérer la solution de norme (euclidienne) minimale (voir [42])

$$(1.14) \quad \begin{cases} \tilde{\Lambda}_1 = (T^T T)^{-1}\tilde{b}_1 - U\tilde{\Lambda}_2 \\ \tilde{\Lambda}_2 = (I + U^T U)^{-1} U^T (T^T T)^{-1}\tilde{b}_1 \end{cases}$$

Après avoir trouvé $\tilde{\Lambda}_1$ et $\tilde{\Lambda}_2$ on réordonne les composantes de cette solution particulière qui est projetée sur $\mathfrak{S}$ au moyen de P et on obtient la solution Λ^* cherchée.

Nous avons alors $E\alpha = K\Lambda - Z$ et en multipliant à gauche par E^T

on a $\quad E^T E\alpha = E^T(K\Lambda - Z)$

alors $\quad \boxed{\alpha = (E^T E)^{-1} E^T (K\Lambda - Z)}$

Il faut souligner que cette méthode ne nécessite aucune condition sur la numérotation des données $t^i \quad i = 1,\ldots,n$, et il n'est pas nécessaire de connaître la matrice P pour pouvoir calculer A, on utilise :

$$A = PKP = (I - VV^T)K(I - VV^T)$$

$$= K + V[\tfrac{1}{2} VV^T KV - KV]^T + [\tfrac{1}{2} VV^T KV - KV]V^T$$

soit

$$W = \frac{1}{2} V^T KV \text{ et } U = KV \qquad W \in \mathcal{M}_{3,3}(\mathbb{R}) \quad V \in \mathcal{M}_{n,3}(\mathbb{R})$$

on a donc $\quad A = K + V[VW - U]^T + [VW - U]V^T$

```
      SUBROUTINE SMETP(N,RC,X,Y,Z,S,T,R,A)
C
C     SPLINES  A 2 DIMENSIONS
C     N NOMBRE DE POINTS  N<=150
C     X,Y  COORDONNEES DES N POINTS
C     Z VALEUR DE LA FONCTION AUX POINTS X,Y
C     RO FACTEUR DE PONDERATION EST 4*PI/RC
C     S(I) I=1,...,N ; T(1),T(2),T(3)  SONT LES
C      COEFFICIENTS DE LA SPLINE
C     LA PARTIE TRIANGULAIRE SUPERIEUR DE LA MATRICE NOYAU EST
C      RANGE DANS  R  DIMENSION NR=N*(N-1)/2
C     A  VECTEUR AUXILIAIRE  DIMENSION NA=N*(N+1)/2
C
      IMPLICIT REAL*8(A-G,O-Z)
      DIMENSION X(1),Y(1),Z(1),S(1),A(1 ),R(1 ),T(3)
      DIMENSION U1(150),U2(150),U3(150),V1(150),V2(150),V3(150)
      DIMENSION TRAC(150)
C
      DO 1 J=2,N
      J1=J-1
      IT=(J1*(J-2))/2
      DO 2 I=1,J1
      IJ=IT+I
      A1= (X(I)-X(J))**2+(Y(I)-Y(J))**2
      R(IJ)=A1*DLOG(A1)
    2 CONTINUE
    1 CONTINUE
C     CALCUL DE TROIS VECTEURS ORTHONORMAUX
      XM=0.D0
      YM=0.D0
      DO 3 I=1,N
      XM=XM+X(I)
      YM=YM+Y(I)
    3 CONTINUE
      XM=XM/N
      YM=YM/N
      DO 4 I=1,N
      V2(I)=X(I)-XM
      V3(I)=Y(I)-YM
    4 CONTINUE
      V2N=PROVEC(N,V2,V2)
      ALF=(PROVEC(N,V2,V3))/V2N
      DO 5 I=1,N
      V3(I)=V3(I)-ALF*V2(I)
    5 CONTINUE
      A2=DSQRT(V2N)
      V3N=PROVEC(N,V3,V3)
      A3=DSQRT(V3N)
      DO 6 I=1,N
      V1(I)=1.D0
      V2(I)=V2(I)/A2
      V3(I)=V3(I)/A3
    6 CONTINUE
C     CALCUL  DE W1,W2,W3,W4,W5,W6
      CALL PROMAT(N,RO,R,V1,U1)
```

```
      CALL PROMAT(N,RO,R,V2,U2)
      CALL PROMAT(N,RO,R,V3,U3)
      W1=(PROVEC(N,V1,U1))/2.D0
      W2=(PROVEC(N,V1,U2))/2.D0
      W3=(PROVEC(N,V2,U2))/2.D0
      W4=(PROVEC(N,V1,U3))/2.D0
      W5=(PROVEC(N,V2,U3))/2.D0
      W6=(PROVEC(N,V3,U3))/2.D0
      A1=W1/N
      A2=W2/N
      A3=W4/N
      DO 15 I=1,N
      U1(I)=(A1+V2(I)*W2+V3(I)*W4-U1(I))/N
      U2(I)=A2+V2(I)*W3+V3(I)*W5-U2(I)
      U3(I)=A3+V2(I)*W5+V3(I)*W6-U3(I)
   15 CONTINUE
      DO 7 I=1,N
      IJ=((I+1)*I)/2
      A(IJ)=2*(U1(I)+V2(I)*U2(I)+V3(I)*U3(I))+PO
    7 CONTINUE
      DO 8 J=2,N
      J1=J-1
      I1=(J1*J)/2
      DO 9 I=1,J1
      IJ=I1+I
      JI=IJ-J1
      A(IJ)=R(JI)+U1(I)+V2(I)*U2(J)+V3(I)*U3(J)
     2            + U1(J)+V2(J)*U2(I)+V3(J)*U3(I)
    9 CONTINUE
    8 CONTINUE
C     APPELLE SUBROUTINE DMFSS
      HPS=1.E-5
      CALL DMFSS(A,N,HPS,IRANK,TRAC)
C     APPELLE SUBROUTINE DMLSS POUR TROUVER LA SOLUTION MINIMALE
      DO 10 I=1,N
      U2(I)=Z(I)
      S(I)=0.D0
   10 CONTINUE
      IK=-3
 41   CONTINUE
      CALL PROJEC(N,V1,V2,V3,U2)
      II=0
      CALL DMLSS(A,N,IRANK,TRAC,II,U2,TER)
      CALL PROJEC(N,V1,V2,V3,U2)
      DO 43 I=1,N
      S(I)=U2(I)+S(I)
   43 CONTINUE
C
C     CALCUL DES TROIS DERNIERS COMPOSANTS
      CALL PROMAT(N,RO,R,S,U1)
      DO 11 I=1,N
      U1(I)=Z(I)-U1(I)
   11 CONTINUE
      A1=  (PROVEC(N,V1,U1))/N
      A2=  (PROVEC(N,X,U1))/N -XM*A1
```

```
      A3=   (PROVEC(N,Y,U1))/N -YM*A1-ALF*A2
      A3=N*(A3/V3N)
      A2=N*(A2/V2N)-ALF*A3
      A1=A1-XM*A2-YM*A3
      DO 40 I=1,N
      U2(I)=U1(I)-(A2*X(I)+A3*Y(I)+A1)
   40 CONTINUE
      IK=IK+1
      IF(IK)41,41,42
   42 CONTINUE
      T(1)=A2
      T(2)=A3
      T(3)=A1
      RETURN
      END
C
      SUBROUTINE PROMAT(N,RO,R,V,RV)
      REAL * 8 R,V,RV,A1,RO
      DIMENSION V(1),RV(1),R(1)
C     CALCUL DE RV=(R+RO*I)*V
      N1=N-1
      N2=N-2
      DO 1 I=1,N1
      I1=(I*(I+1))/2
      J1=I1
      A1=R(I1)*V(I+1)+RO*V(I)
      IF(I.EQ.N1)GO TO 6
      DO 2 J=I,N2
      J1=J1+J
      A1=A1+R(J1)*V(J+2)
    2 CONTINUE
      IF(I.EQ.1)GO TO 5
    6 J1=I1-I+1
      I1=I-1
      DO 3 K=1,I1
      J=I-K
      J1=J1-1
      A1=A1+R(J1)*V(J)
    3 CONTINUE
    5 RV(I)=A1
    1 CONTINUE
      I1=(N1*N2)/2
      A1=RO*V(N)
      DO 4 J=1,N1
      A1=A1+R(I1+J)*V(J)
    4 CONTINUE
      RV(N)=A1
      RETURN
      END
C
      SUBROUTINE PROJEC(N,V1,V2,V3,S)
      REAL*8 V1,V2,V3,S,A1,A2,A3
      DIMENSION S(1),V1(1),V2(1),V3(1)
C     CALCUL DE LA PROJECTION SUR F
      A1=(PROVEC(N,V1,S))/N
```

```
      A2=PROVEC(N,V2,S)
      A3=PROVEC(N,V3,S)
      DO 18 I=1,N
      S(I)=S(I)-(A1+V2(I)*A2+V3(I)*A3)
   18 CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION PROVEC(N,X,Y)
      REAL * 8 X,Y,PROVEC,V
      DIMENSION X(1),Y(1)
      V=0.D0
      DO 1 I=1,N
      V=V+X(I)*Y(I)
    1 CONTINUE
      PROVEC=V
      RETURN
      END
```

METHODES ITERATIVES

Soit $f(\Lambda) = \frac{1}{2}\Lambda^T K\Lambda - \Lambda^T Z$

Proposition 8

f admet dans $\mathcal{F}$ un minimum unique, atteint en Λ^* solution du système (1.3)

Démonstration :

Soit le problème (P) $\underset{\Lambda \in \mathcal{F}}{\text{Inf}}\ (f(\Lambda))$ et considérons le lagrangien (associé aux contraintes) :

$$\mathcal{L}(\Lambda,\alpha) = \frac{1}{2}\Lambda^T K\Lambda - Z^T\Lambda + \alpha^T E^T\Lambda \qquad \Lambda \in \mathbb{R}^n \quad \alpha \in \mathbb{R}^3$$

si $(\bar{\Lambda},\bar{\alpha})$ est un point de selle du Lagrangien, c'est-à-dire

$$(*) \qquad \mathcal{L}(\bar{\Lambda},\alpha) \leq \mathcal{L}(\bar{\Lambda},\bar{\alpha}) \leq \mathcal{L}(\Lambda,\bar{\alpha}) \qquad \forall\Lambda,\ \forall\alpha$$

alors $\bar{\Lambda}$ réalise le minimum de f sur $\mathcal{F}$, pour avoir (*) il suffit d'annuler les dérivées de $\mathcal{L}$ par rapport à Λ et α en $(\bar{\Lambda},\bar{\alpha})$ ce qui donne :

$$K\bar{\Lambda} - Z + E\bar{\alpha} = 0 \qquad \text{(par rapport à } \Lambda\text{)}$$

$$E^T\bar{\Lambda} = 0 \qquad \text{(par rapport à } \alpha\text{)}$$

ce qui correspond bien au système (1.3)

La solution du problème (P) est unique du fait que K est définie positive sur $\mathcal{F}$.

Remarque :

Le vecteur $\bar{\alpha}$ est solution du problème dual associé :

$$(Q) \qquad \underset{\alpha \in \mathbb{R}^3}{\text{Inf}}\ \underset{X \in \mathbb{R}^n}{\text{sup}}\ (-\mathcal{L}(X,\alpha))$$

Le problème perturbé de (P) est :

$$(P_\alpha) \qquad \inf_{\Lambda \in \mathbb{R}^n} \varphi(\Lambda,\alpha) \quad \text{où } \varphi(\Lambda,\alpha) = \begin{cases} f(\Lambda) & \text{si } \Lambda \in \mathcal{F}_\alpha \\ +\infty & \text{sinon} \end{cases}$$

$$\mathcal{F}_\alpha = \{\Lambda \in \mathbb{R}^n \,/\, E^T\Lambda+\alpha = 0\}$$

Donc on peut chercher Λ^* résolvant le problème :

$$(1.15) \qquad f(\Lambda^*) = \min_{\Lambda \in \mathcal{F}} f(\Lambda)$$

utilisant une méthode de descente voir [3] [6] [23] [28] [39].

METHODE (C)

Descente dans des directions imposées.

Supposons que l'hypothèse (H) soit vérifiée et reprenons $R = [r_1,\dots,r_{n-3}]$ base construite comme pour la méthode (A) et considérons l'itération suivante :

$$(1.16) \qquad \begin{cases} X_1 = \Lambda_i & i = 1,\dots \\ X_{j+1} = X_j+\theta_j r_j & j = 1,\dots,n-3 \\ \Lambda_{i+1} = X_{n-2} & \end{cases}$$

où $\Lambda_1 \in \mathcal{F}$ donné

$$\theta_j = -\frac{\rho_j^T r_j}{r_j^T K r_j} \qquad \rho_j = KX_j - Z$$

Donc une itération consiste en (n-3) déplacements dans des directions $r_1,\dots,r_{n-3}$, comme ces (n-3) vecteurs constituent une base de $\mathcal{F}$ il existe au moins un r_{i_o} tel que $\rho_j^T r_{i_o} \neq 0$, dans le cas contraire on a $\rho_j \in \mathcal{E}$ alors $\Lambda^* = X_j$, et pour trouver α^* on utilise (1.7).

Par construction

$$\Lambda_i \in \mathfrak{F} \ \forall i$$

$$f(\Lambda_{i+1}) \leq f(\Lambda_i) \text{ parce que } \theta_j \text{ minimise } f(X_j+\theta r_j)$$

Pour la convergence rapprochons cette méthode de la méthode de Gauss-Seidel sur le système (1.5) voir [17] [32]

$$(1.17) \quad \begin{cases} Y_1 = \gamma_i & i = 1,\ldots \\ Y_{j+1} = Y_j + \bar{\theta}_j e_j & j = 1,\ldots,n-3 \\ \gamma_{i+1} = Y_{n-2} & \end{cases}$$

où $\gamma_1 \in \mathbb{R}^{n-3}$ (donné)

$$\bar{\theta}_j = -\frac{e_j^T \bar{\rho}_j}{a_{jj}} \qquad \bar{\rho}_j = AY_j - b$$

Si on fait l'identification suivante :

$$X_j = RY_j \quad \Lambda_i = R\gamma_i \qquad j = 1,\ldots,n-3 \quad i = 1,\ldots$$

On obtient l'algorithme (1.16) et :

$$\|\Lambda_i - \Lambda^*\| \leq \|R\| \ \|\gamma_i - \gamma^*\| \to 0$$

$$\{\Lambda_i\} \to \Lambda^* \text{ en norme} \quad \text{voir [21][32]}$$

<u>Remarques</u>

1- Pour la base $r_1,\ldots,r_{n-3}$ utiliséedans cette méthode on a $\|R\| \geq 1$ alors la convergence est moins rapide que la méthode de Gauss-Seidel mais elle ne nécessite que le seul stockage de $C_1^i \ C_2^i \ C_3^i$ $i = 1,\ldots,n-3$ (trois vecteurs de dimension n-3 au lieu de $\frac{(n-3)(n-2)}{2}$ termes).

2 - Pour trouver X_{j+1} il n'est pas nécessaire d'évaluer toutes les composantes de ρ_j, il suffit des trois premières et de la (j+3)ième composante, de plus les valeus $r_j^T K r_j$ $j = 1,\ldots,n-3$ peuvent être calculées une fois pour toutes (au prix de n-3 places mémoires de plus).

3 - Le coût de chaque itération est de l'ordre de $4n^2-5n$ opérations.

4 - Cette méthode permet aussi de ne pas stocker la matrice K, qui ne dépend que des données t^i $i = 1,\ldots,n$, bien sûr le coût dans chaque itération va être plus élevé, car il faut calculer K à chaque itération, néanmoins ceci est important quand on dispose d'une petite machine.

5 - Si on prend une base orthonormée on a $\|R\| = 1$ et la méthode est au moins aussi rapide que la méthode de Gauss-Seidel, mais dans la pratique, on ne gagne rien car l'évaluation de R et chaque itération seront chères, et de plus, il faut stocker la matrice R.

6 - Cette méthode de descente est en quelque sorte un procédé du type Gauss-Seidel.

On peut modifier le procédé (1.16) afin de générer des suites qui convergent "plus vite" que celles obtenues avec l'algorithme original. Voir [3] [13].

METHODE : C-1

Dans le procédé (1.16) au lieu de prendre le déplacement dans la direction r_j $(j = 1,\ldots,n-3)$ on le fera dans la direction $U_j = r_j+\beta_j U_{j-1}$ où β_i et U_{j-1} seront explicités ensuite :

Soit X_1 et $U_o \in \mathcal{F}$ (donné) telle que

$(KX_1-Z)^T U_o = 0$ (on peut prendre $X_1 = U_o = 0 \in \mathbb{R}^n$)

Définissons $X_{j+1} = X_j+\theta_j U_j$ $\qquad j = 1,\ldots,n-3$

avec $U_j = r_j+\beta_j U_{j-1}$

Pour chaque direction U_j (avec donc β_j fixé) la valeur minimum de $f(X_{j+1})$ est atteinte pour

$$\theta_j = -\frac{\rho_j^T U_j}{u_j^T K U_j} \quad \text{où} \quad \rho_j = KX_j - Z$$

et vaut : $f(X_{j+1}) = f(X_j) - \frac{1}{2}\frac{(\rho_j^T U_j)^2}{u_j^T K U_j}$

Soit $\alpha = \frac{(\rho_j^T U_j)^2}{u_j^T K U_j}$ et on cherche β_j de façon que α soit maximum :

$$\alpha = \frac{[\rho_j^T(r_j+\beta_j U_{j-1})]^2}{r_j^T K r_j + \beta_j^2 U_{j-1}^T K U_{j-1} + 2\beta_j r_j^T K U_{j-1}} = \frac{(\rho_j^T r_j)^2}{r_j^T K r_j + \beta_j^2 U_{j-1}^T K U_{j-1} + 2\beta_j r_j^T K U_{j-1}}$$

car $\rho_j^T U_{j-1} = 0$

donc α sera maximum pour $\boxed{\beta_j = -\frac{r_j^T K U_{j-1}}{U_{j-1}^T K U_{j-1}}}$

Alors on obtient l'algorithme accélérée de la méthode c

$$(1.18) \quad \begin{cases} X_1 = \Lambda_i \quad U_o = V_{i-1} \quad i = 1,\ldots \quad \Lambda_1 \in \mathcal{F} \text{ donné} \quad V_o^1 = 0 \\ X_{j+1} = X_j + \theta_j U_j \quad U_j = r_j + \beta_j U_{j-1} \quad j = 1,\ldots,n-3 \\ \Lambda_{i+1} = X_{n-2} \quad V_{i+1} = U_{n-3} \end{cases}$$

où

$$\beta_j = -\frac{r_j^T K U_{j-1}}{U_{j-1}^T K U_{j-1}} \qquad \theta_j = -\frac{U_j^T \rho_j}{U_j^T K U_j}$$

METHODE (D) GRADIENT PROJETE

Cette méthode présentée par Rosen [3] et développée par d'autres auteurs (voir [6] [23] [28]) pour résoudre différents types de problèmes, est facilement applicable dans cette minimisation, car c'est une méthode de descente dans la direction du gradient sur $\mathcal{F}$, c'est-à-dire c'est la méthode du gradient restreint au sous-espace $\mathcal{F}$.

L'algorithme :

$$(1.20)\quad \begin{cases} \Lambda_1 \in \mathcal{F} \quad \text{(donné)} \\ \Lambda_{j+1} = \Lambda_j + \theta_j \tilde{\rho}_j \qquad j = 1,\ldots \\ \text{ou} \quad \theta_j = -\dfrac{\tilde{\rho}_j^T \tilde{\rho}_j}{\tilde{\rho}_j^T K \tilde{\rho}_j} \qquad \tilde{\rho}_j = \mathrm{Proj}_{\mathcal{F}}(\rho_j) \qquad \rho_j = K\Lambda_j - Z \end{cases}$$

Evidemment on a :

i) $\Lambda_j \in \mathcal{F} \quad \forall j = 1,\ldots$

ii) $f(\Lambda_{j+1}) \leq f(\Lambda) \leq f(\Lambda_j)$

$$\Lambda = \varphi \Lambda_{j+1} + (1-\varphi)\Lambda_j \quad \forall \varphi \in [0,1]$$

car θ_j minimise $f(\Lambda_j + \theta\tilde{\rho}_j)$

iii) $\theta_j = -\dfrac{\tilde{\rho}_j^T \tilde{\rho}}{\tilde{\rho}_j^T K \tilde{\rho}_j} \qquad (\tilde{\rho}_j \in \mathcal{F})$

Remarque :

1) Pour $\Lambda \in \mathcal{F}$

$$f(\Lambda) = \frac{1}{2}\Lambda^T K\Lambda - \Lambda^T Z = \frac{1}{2}\Lambda^T PKP\Lambda - \Lambda^T PZ$$

avec $P = I - VV^T$, V définie comme dans la Méthode (B).

On a alors

(1.21) $$f(\Lambda) = \frac{1}{2}\Lambda^T A\Lambda - \Lambda^T b \qquad \text{avec } A = PKP, \quad b = PZ$$

Alors cette méthode n'est autre que la méthode de la plus profonde descente pour résoudre le système (1.11) et par conséquent $\tilde{\rho}_j$ peut être calculé de deux façons (équivalentes)

a) $$\tilde{\rho}_j = \rho_j - E\beta_j \qquad \text{avec } \beta_j = (E^TE)^{-1}E^T\rho_j$$

b) $$\tilde{\rho}_j = P\rho_j$$ P définie comme dans la méthode (B)

2) La suite Λ_i converge vers la solution Λ^* et de plus on a : voir [21] [32]

(1.22) $$\|\Lambda_{j+1}-\Lambda^*\| \leq (1-\frac{\mu}{\beta})^{j/2}\sqrt{\frac{(\Lambda_1-\Lambda^*)^TK(\Lambda_1-\Lambda^*)}{\mu}}$$

avec μ coefficient de coercivité de K

$$\beta = \|K\|_{\mathcal{F}}$$

3) Cette méthode est indépendante de la numérotation des données t^i $i = 1,\ldots,n$ et permet aussi de ne pas stocker K.

4) Le coût pour chaque itération est de l'ordre de $2n^2+5n$

5) Si on a $\tilde{\rho}_j^TK\tilde{\rho}_j = 0$ θ_j n'est pas définie mais cette égalité implique $\tilde{\rho}_j = 0$ (propriété de K) alors $\Lambda^* = \Lambda_j$

d'où finalement : $$\alpha^* = \beta_j = (E^TE)^{-1}E^T\rho_j$$

De la même façon que pour la méthode (C) on peut accélérer cette méthode :

$$(1.24)\quad \begin{cases} \Lambda_1 \in \mathcal{F} \text{ (donné)} & U_1 = \tilde{\rho}_1 \\ \Lambda_{j+1} = \Lambda_j + \theta_j U_j & U_j = \tilde{\rho}_{j+1} + \mu_j U_j \qquad j = 1,\dots \end{cases}$$

où

$$\theta_j = - \frac{U_j^T \tilde{\rho}_j}{U_j^T K U_j}$$

$$\mu_j = - \frac{U_j^T K \rho_{j+1}}{U_j^T K U_j}$$

Cet algorithme n'est autre que la méthode du gradient conjugué voir [13] [17] [22][32]
Il converge donc en un nombre fini de pas.

I.4 - FONCTION SPLINE D'AJUSTEMENT DU TYPE PLAQUE MINCE

Soient t^i $i = 1,\dots,n$ n points du $\mathbb{R}^2$ non tous alignés.
Z_i. $i = 1,\dots,n$ n nombres réels et $\rho > 0$.

Dans [11] [12] on trouve la caractérisation suivante :

Théorème 2

Parmi les fonctions continues sur $\mathbb{R}^2$ il existe une et une seule fonction σ_ρ qui minimise la fonctionnelle :

$$\int_{\mathbb{R}^2} |\nabla^2 v|^2 + \rho \sum_{i=1}^{n} |v(t^i) - Z_i|^2$$

et σ_ρ s'écrit d'une manière unique sous la forme :

$$(1.25)\quad \sigma_\rho(t) = \sum_{i=1}^{n} \lambda_i \mathcal{K}(t^i, t) + \alpha_1 x + \alpha_2 y + \beta$$

avec $\sum_{i=1}^{n} \lambda_i = 0$ $\qquad \sum_{i=1}^{n} \lambda_i t^i = 0$

Les coefficients $\lambda_1,\dots,\lambda_n$, α_1,α_2,β sont solutions du système :

$$(1.26) \qquad \begin{bmatrix} \tilde{K} & E \\ E & O \end{bmatrix} \begin{bmatrix} \Lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} Z \\ O \end{bmatrix}$$

où $\tilde{K} = K+\tilde{\rho} I_n \qquad \tilde{\rho} = 4\pi/\rho$

Propriété 3 :

$\tilde{K}$ est définie positive sur $\mathcal{F}$.

Pour résoudre le système (1.26) on peut utiliser les quatre méthodes décrites en changeant K en $\tilde{K}$.

I.5 - RESULTATS NUMERIQUES ET CONCLUSION

Présentons d'abord l'exemple suivant.

On prend t^i $i = 1,\dots,50$, 50 points aléatoires dans $[0,1] \times [0,1]$ générés par la procédure Randu (voir [30]) et la numérotation choisie est l'ordre d'obtention, soit $Z_i = \exp(x_i y_i)$ et le test d'arrêt util pour la méthode (C) (procédé (1.16)) et la méthode (D) (procédé 1.20) est :

$$\frac{\varphi_1(\Lambda_{j+1}-\Lambda_j)}{\varphi_1(\Lambda_{j+1})} < \varepsilon = 10^{-3} \qquad \Lambda_1 = 0 \text{ où } \varphi_1(\Lambda) = \sum_{i=1}^{50} |\lambda_i|$$

Les temps d'exécution obtenus ont été

Tableau 1

A	B	C	D
0.95	1.74	20.06	15.44

secondes

Le temps d'exécution en secondes de chaque itération est en moyenne :

C	D
.5	.36

Les splines trouvées pour chaque méthode ont été représentées en perspective au moyen du terminal graphique IBM 2250 (voir [30]).Ainsi que la différence de ces splines avec $f(x,y)=\exp(xy)$.On trouve que les quatre surfaces sont proches de $f(x,y)$,les meilleures étant celles qui sont trouvées par les deux méthodes directes.

Les méthodes itératives etant très couteuses et moins precises, nous allons comparer numeriquement les deux méthodes directes sur les exemples reels suivants:

a) Relevé de pluie dans la region des Cevennes dessin N° 1 et N° 2 , avec 73 points((stations).

b) Altitude de la region des Cevennes dessin N° 3, 73 mesures.

c) Relevé du bruit dans une usine de G.D.F.

1) 70 points dessin N° 4.

2) 140 points dessin N° 5.

3) 139 points dessin N° 7.

Notons σ_A, σ_B les splines obtenues par les méthodes A et B,alors on peut écrire:

$$\sigma_A(t) = \sum_{i=1}^{n} \lambda_i^A \, \mathcal{K}(t,t^i) + \lambda_{n+1}^A x + \lambda_{n+2}^A y + \lambda_{n+3}^A$$

$$\sigma_B(t) = \sum_{i=1}^{n} \lambda_i^B \, \mathcal{K}(t,t^i) + \lambda_{n+1}^B x + \lambda_{n+2}^B y + \lambda_{n+3}^B$$

Dans chaque exemple consideré,le calcul des coefficients λ_i^A et λ_i^B $i=1,\dots,n+3$ a eté fait en double précision ; la méthode (A) utilise la décomposition de Choleski,de même que la méthode (B) mais avec permutation des lignes et des colonnes.Nous avons constaté que:

$$(1.27) \quad |\lambda_i^A - \lambda_i^B| \leq 10^{-8}$$

$$(1.28) \quad |\sigma_A(t^i) - z_i| \leq 10^{-11}$$

$$(1.29) \quad |\sigma_B(t^i) - z_i| \leq 10^{-7}$$

Comme on le voit la méthode (B) est "moins exacte" que la méthode (A), car les erreurs d'arrondi comises sont telles que la deuxième relation de (1.13') n'est pas vraie,mais $(U^T\tilde{b}_1 - \tilde{b}_2)$ est très petit.En conséquence il est préferable de corriger la solution,en utilisant la factorisation dejà faite voir [24], [29],[41] où on étudie en detail problèmes de ce type.

De manière experimentale on s'est aperçu que pour avoir pratiquement la même solution que celle de la méthode (A), il suffit de corriger trois fois de la façon suivante:

(1.30) $\Lambda^{k+1} = \Lambda^k + \varepsilon^k$ $\qquad k = 1,2,3$

où ε^k est la solution de $A\varepsilon^k = (b - A\Lambda^k)$ voir [22]

Λ^1 est la solution à corriger

Le tableau suivant est le temps d'execution en secondes pour les deux méthodes directes, considérant (1.30) et les algorithmes présentés .

Tableau 2

n	Méthode A	Méthode B	n	Méthode A	Méthode B	n	Méthode A	Méthode B
10	0.022	0.093	55	1.199	2.120	100	5.431	7.868
15	0.057	0.174	60	1.485	2.550	105	6.160	8.798
20	0.111	0.292	65	1.816	3.025	110	6.957	9.784
25	0.186	0.438	70	2.184	3.555	115	7.818	10.848
30	0.279	0.618	75	2.599	4.133	120	8.738	11.970
35	0.404	0.836	80	3.060	4.759	125	9.884	13.391
40	0.557	1.094	85	3.571	5.451	130	11.009	14.577
45	0.735	1.396	90	4.134	6.191	135	12.177	15.899
50	0.948	1.735	95	4.756	6.993	140	13.331	17.352

Les dessins qu'on exhibesont les tracés des courbes de niveaux obtenues au moyen de la méthode décrit en [31] (voir aussi [25]), l'algorithme se trouve dans la Bibliothèque des Procedures Harwell.

Pour obtenir ces résultats numériques on procéde de la manière suivante:

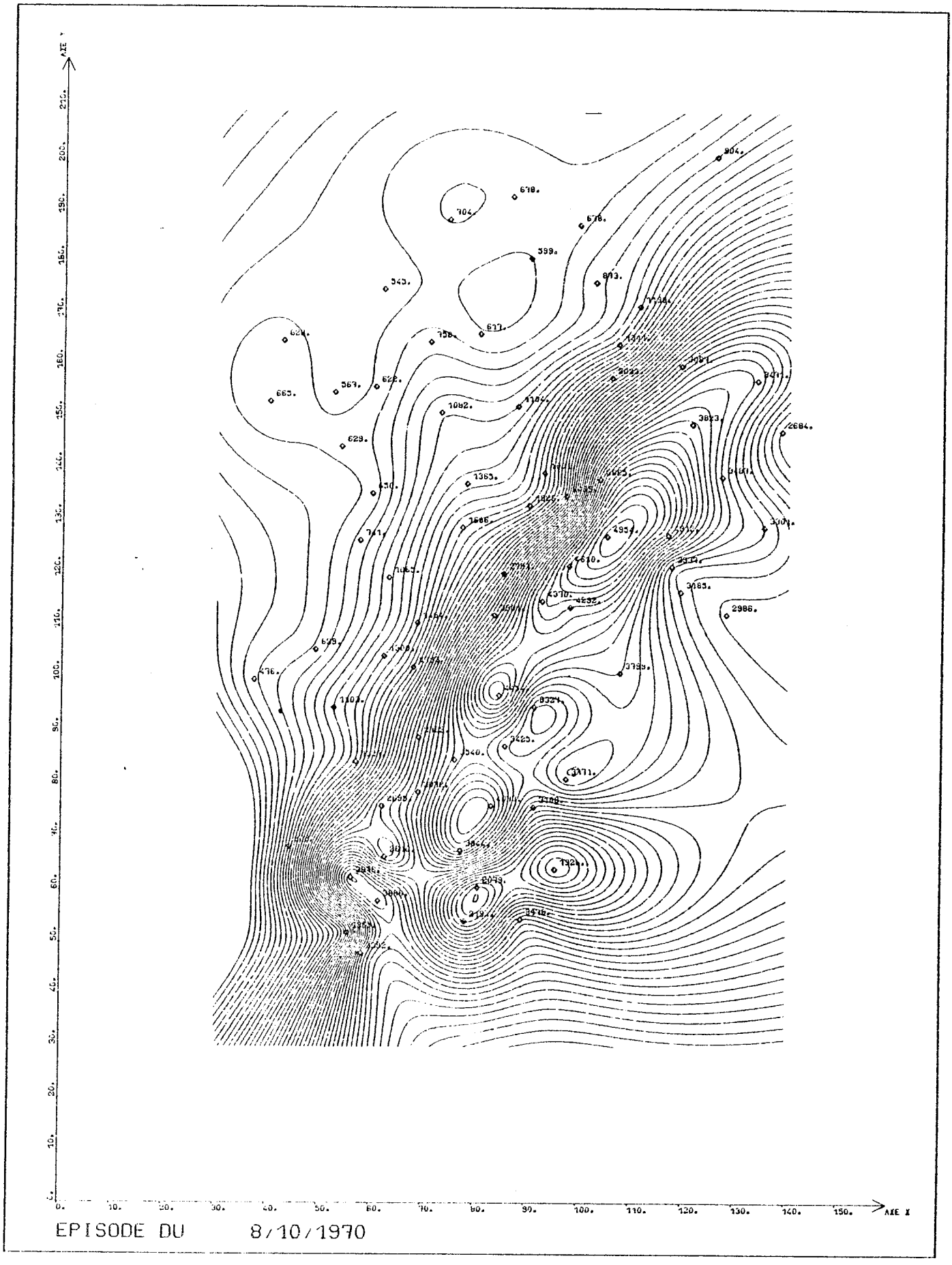
1°) Utilisant les données $t^i=(x_i,y_i)$ et z_i $i=1,\dots,n$ on calcule les coefficients de la spline d'interpolation.

2°) On fixe une grille régulière fine dans la partie à tracer du domaine, et on trouve la valeur de la spline au noeuds de la grille.

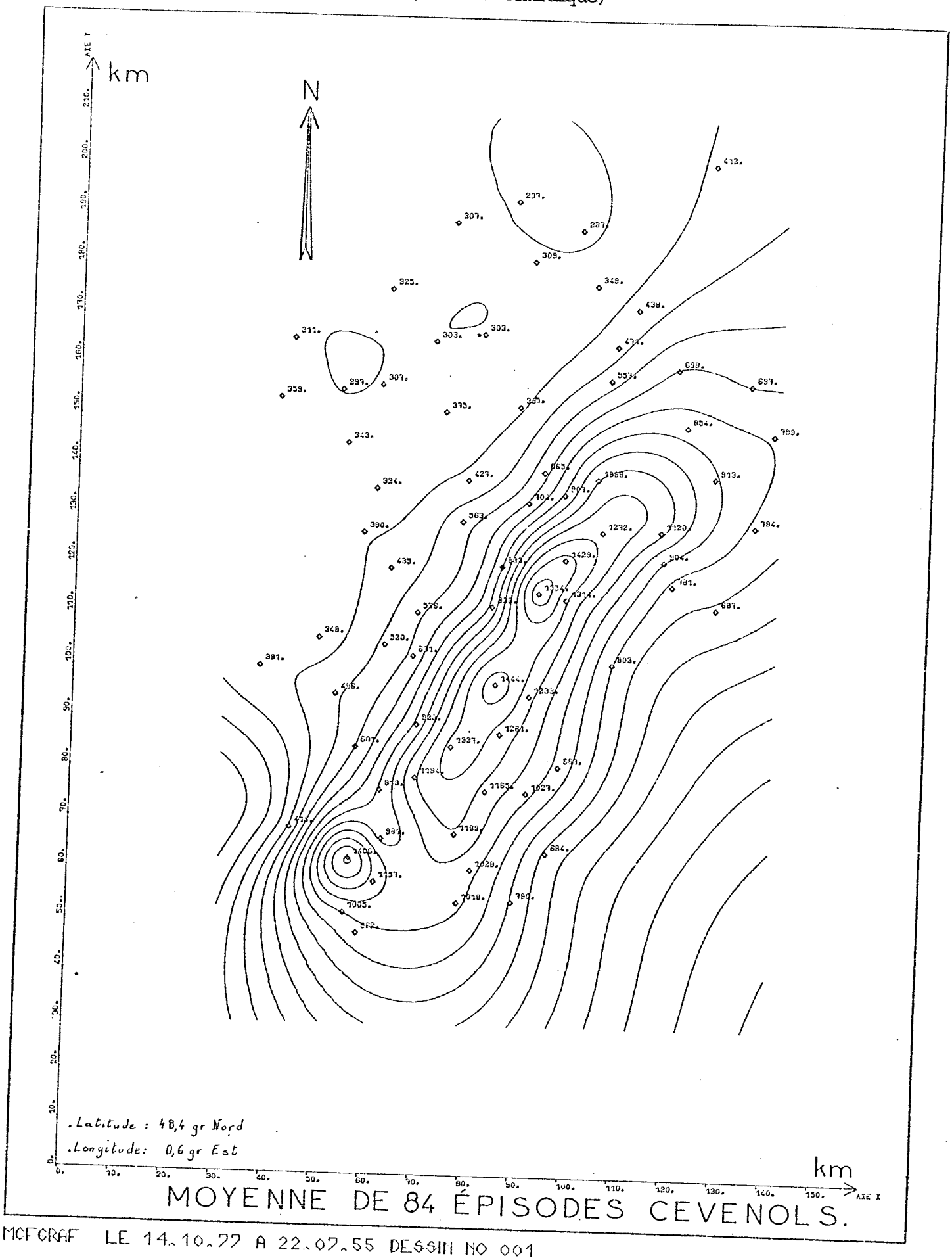
3°) Les valeures de la spline ainsi obtenues sont utilisées dans la procedure de tracé des courbes de niveaux.

Dessin n° 1

(tracé automatique)

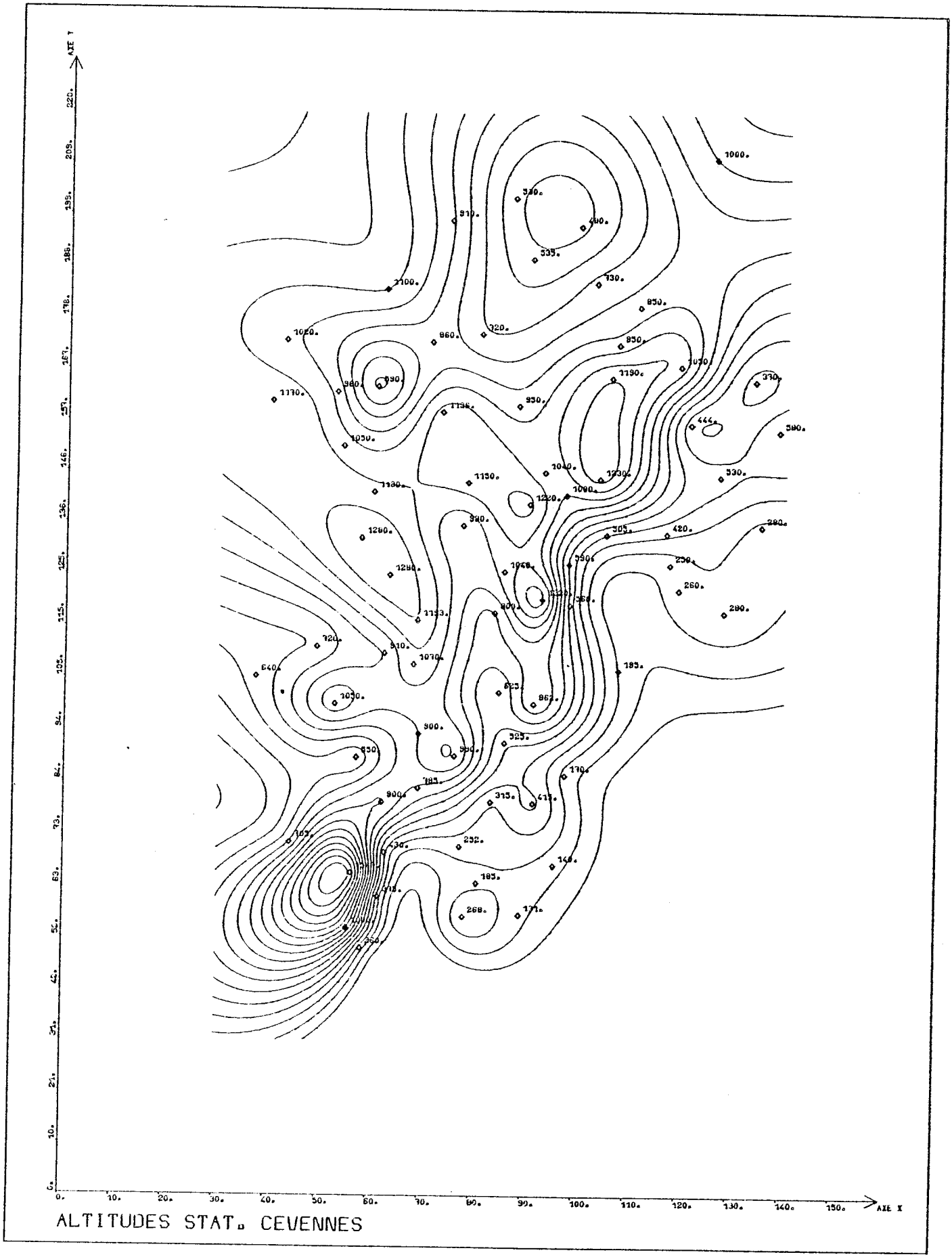


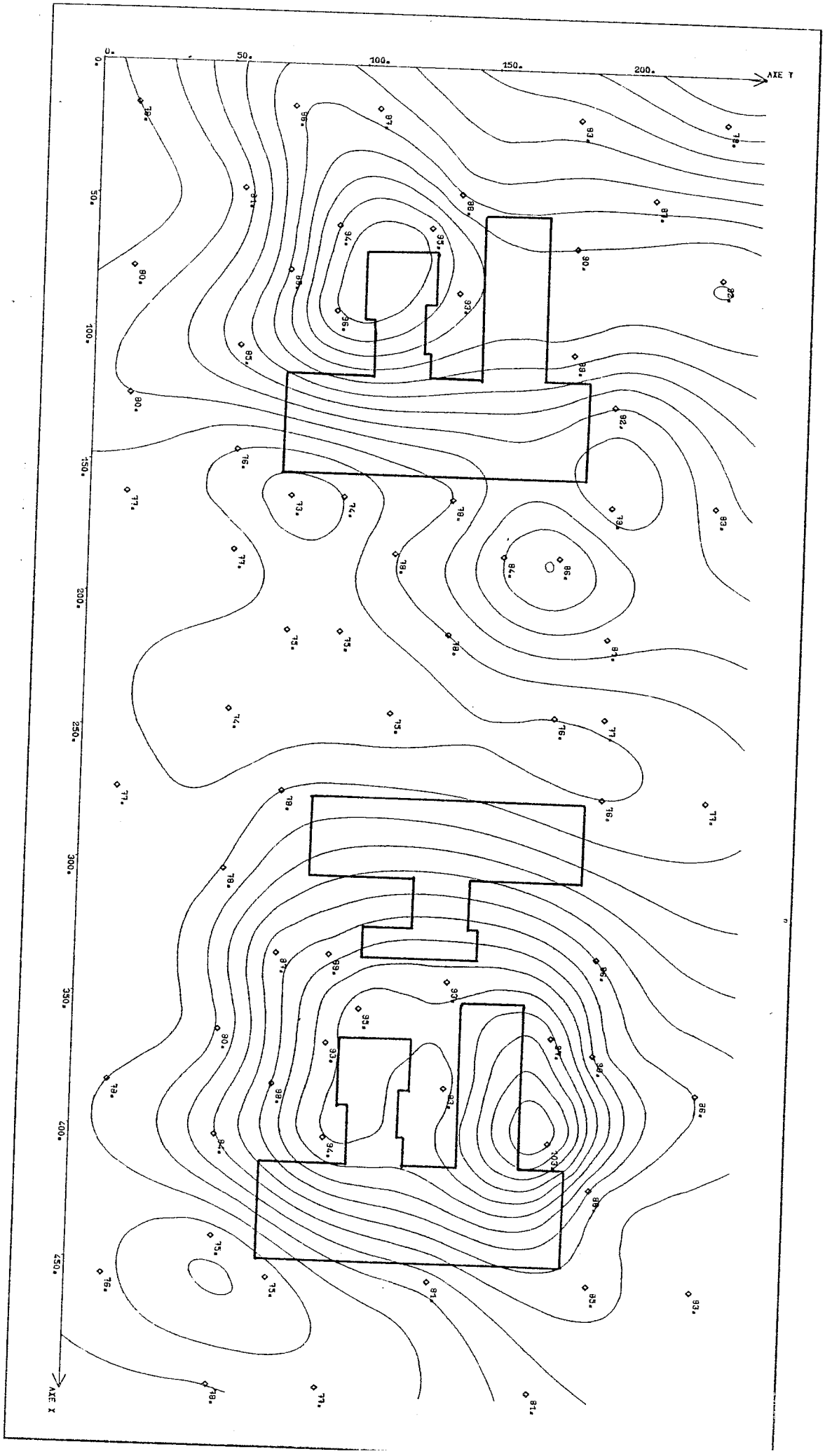
Dessin n° 2 (tracé automatique)



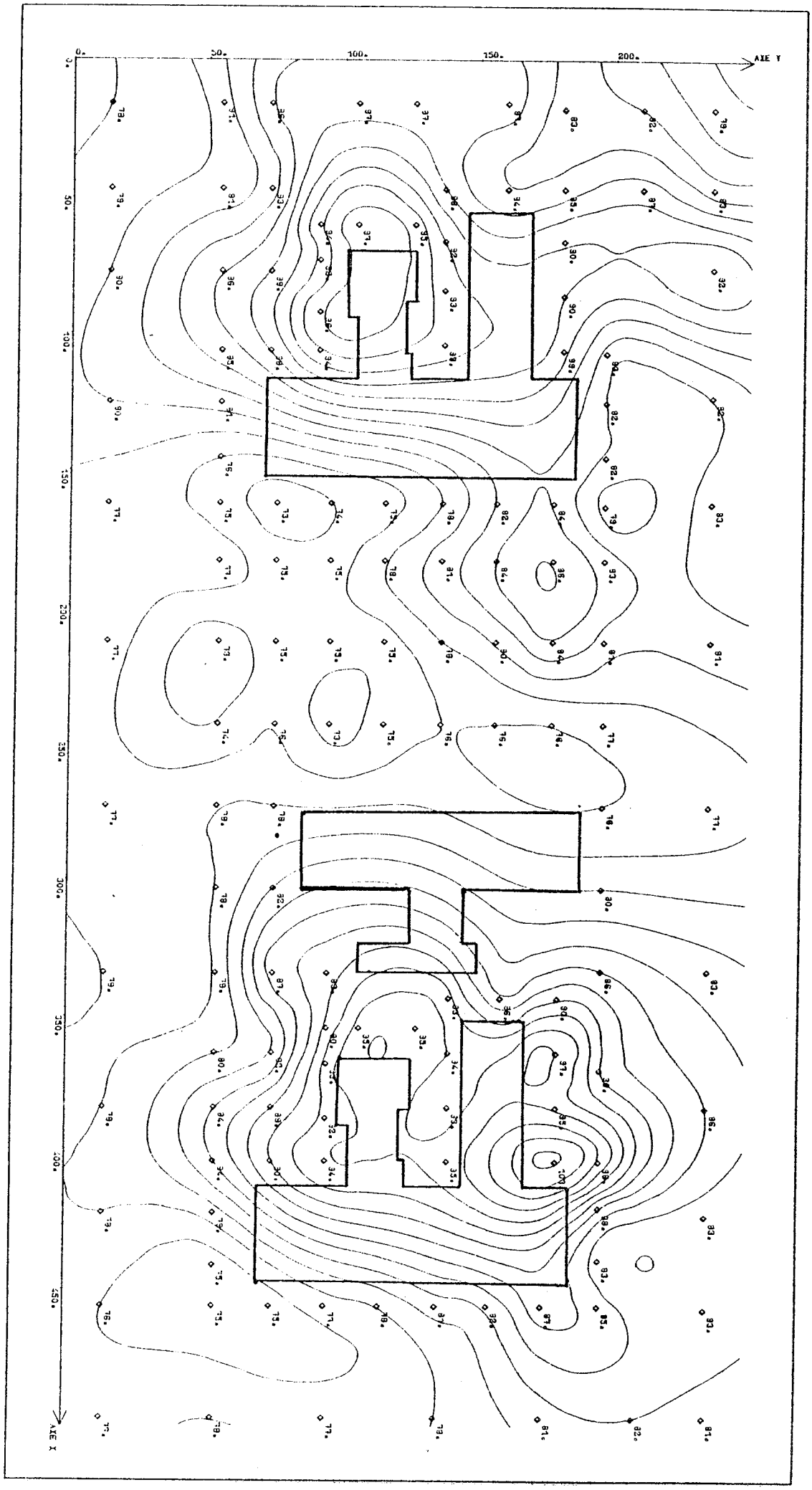
MCFGRAF LE 14.10.77 A 22.07.55 DESSIN NO 001

Dessin n° 3 (tracé automatique)

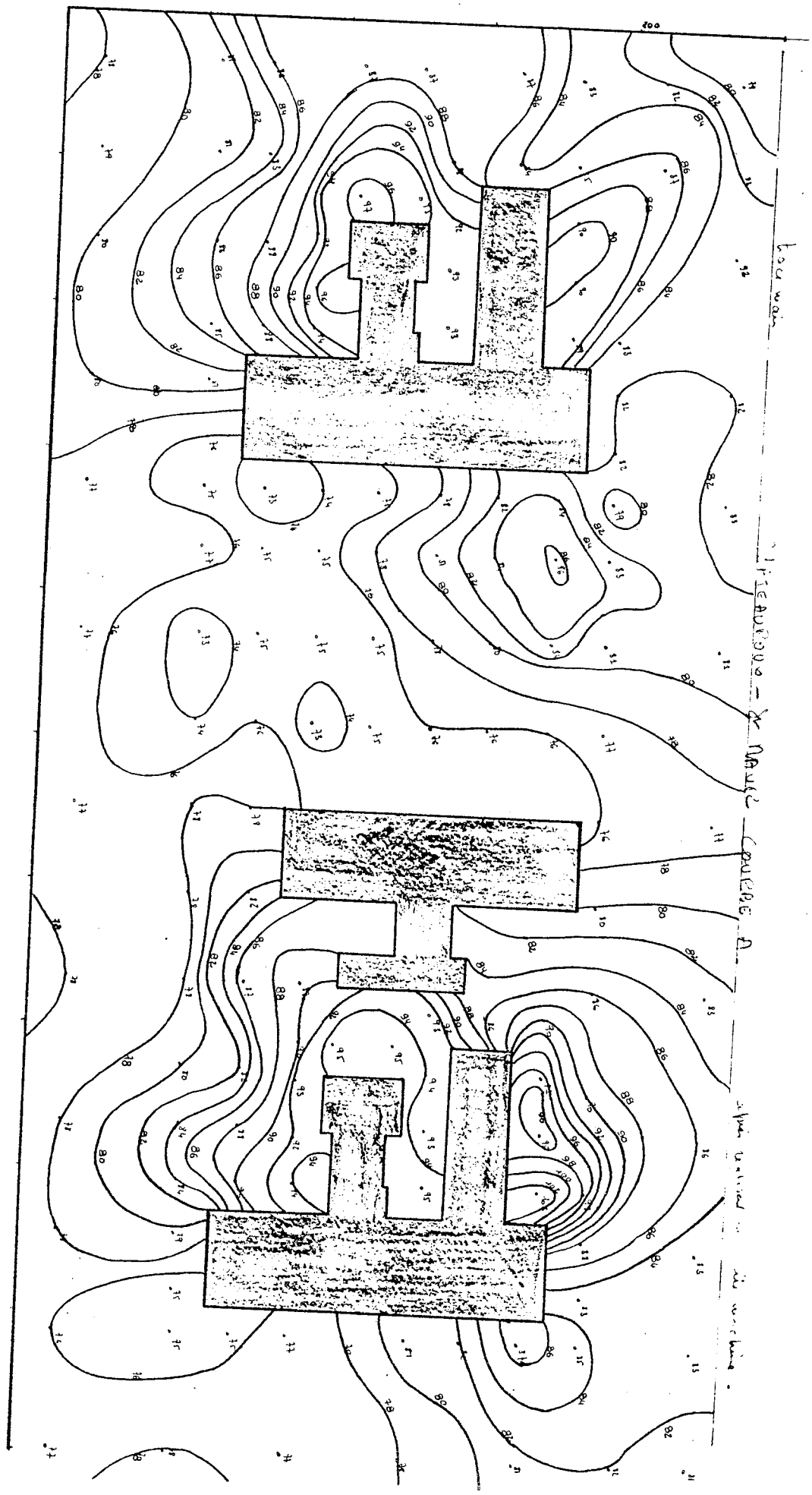




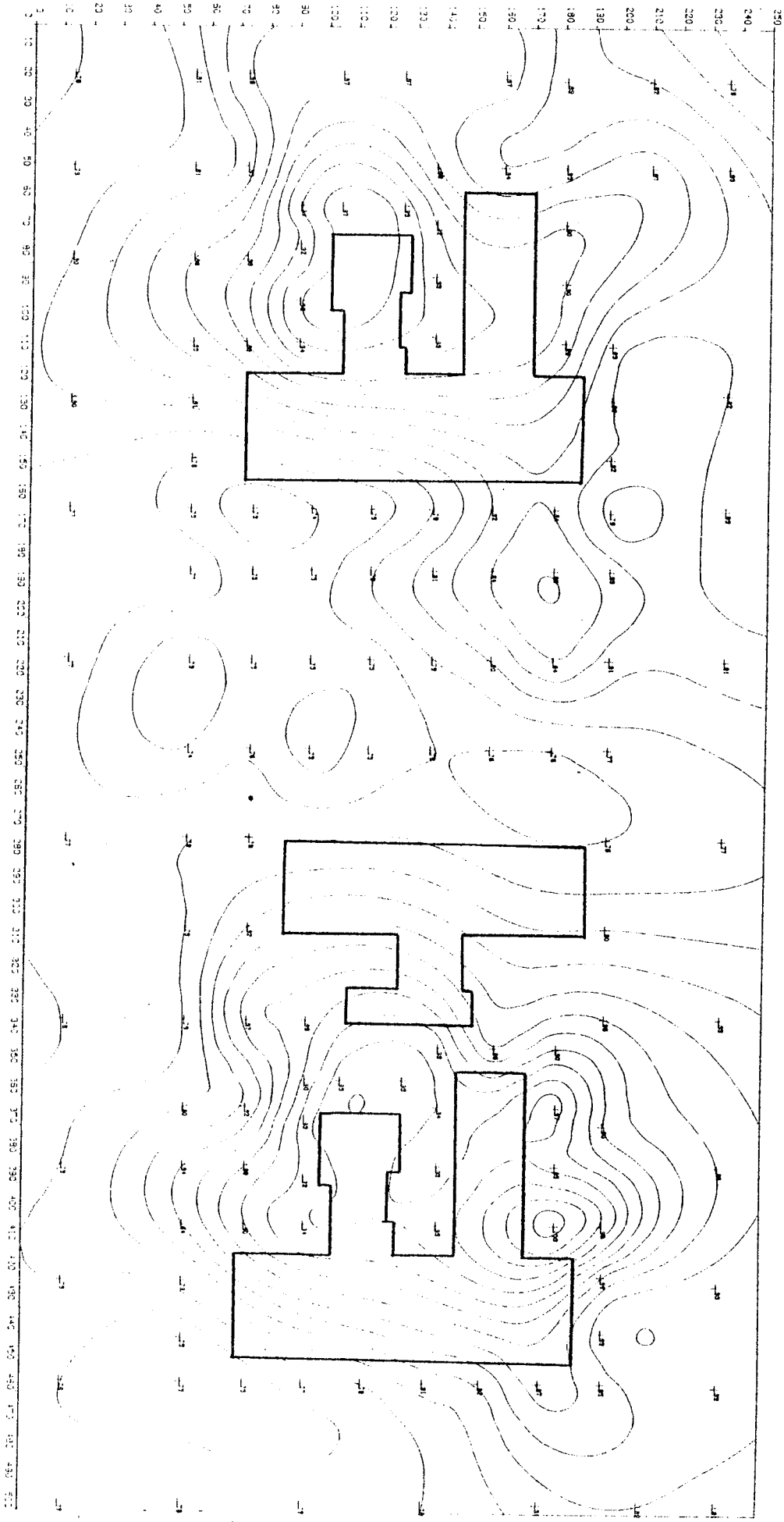
Dessin
(tracé
automat.



Dessin n° 5
(tracé
automatique



Dessin n°
(tracé à
main)



Dessin n° 7
(tracé
automatique)

On souligne que le calcul des coefficients de la spline et les évaluationes dans la grille sont faits en double précision,et le tracé en précision simple.

Les tableau suivante nous donne les caracteristiques des résultats présentés (sans compter le calcul de la spline).

Tableau 3

Dessin	Nombre de Points	Nombre de Courbes de Niveaux	Domaine	Grille	Temps calcul de Grille(sec.)	Temps Tracé (sec.)
1	73	50 isohyetes	[30,160]×[30,220] Km^2	60 90	86.12	906.23
2	73	20 isohyetes	"	"	"	328.63
3	73	17	"	"	"	320.45
4	70	20 isobruits	[0,500]×[0,250] M^2	51 21	16.3	86.3
5	140	20 isobruits	"	"	33.4	90.2

Remarques:

1) Le dessin Nº 6 a eté fait à la main (EDF) et on voit une différence enorme avec le dessin Nº 5 , au voisinage du point (76.,236.) où la valeur donnée est 92. (décibels).

Le dessin Nº 7 est obtenu sans considérer le point (76.,236.) (donnée erronée),et on voit qu'il est semblable au dessin Nº 6 .

2) En général, les méthodes de tracé d'isoligne, utilisent l'interpolation par polynômes ou triangularisation à cause de la simplicité des calculs, voir par exemple [8],[10],[27],et on peut considerer deux cas:

a) Si les données sont disposées en grille régulière le plus simple à faire et soit:

- Une interpolation locale par polynômes voir [25] .
- Une interpolation par spline bicubique voir [4],[5],[20] .

b) Si les données sont disposées de manière quelconque (cas plus fréquent), d'abord on estime les valeurs dans une grille fixé à l'avance, ces valeurs on peut les estimer au moyen d'une interpolation polynômiale en prenent des sous ensembles des données voir [8],[16],[26];et on se trouve alors dans le cas (a) .

CONCLUSION

1) Il est difficile bien sûr de comparer une méthode itérative à une méthode directe (choix de la précision) , l'avantage essentielle des deux méthodes itératives présentées ici est qu'elles n'ont pas le problème d'encombrement,malheuresement la matrice K est pratiquement pleine et en conséquence chaque itération est couteuse.

2) D'après les experiences faites , l'unique limitation qu'on ait trouvée pour les deux méthodes directes est "l'encombrement",dû à la matrice K et A, les résultats numériques étant eux très satisfaisants.

3) L'hypothèse (H) est fondamentale pour la méthode A ,et on conseille que les trois premiers points soient assez éloignés l'un à l'autre et à peu près à égale distance les uns des autres; si les trois premièrs points sont presque alignés il-y-a le risque de mal determiner les coefficients barycentriques ($-C_1^i$,$-C_2^i$,$-C_3^i$ $i=1,\ldots,n-3$),c'est-à-dire mal determiner le système (1.5) .Par contre dans la méthode B "aucune hypothèse n'est necessaire" sur la numerotation des données; mais la méthode A est moins couteuse que la méthode B .

4) D'après l'opinion des personnes qui nous ont fournis les données on peut conclure que:"Les dessins obtenus reflètent la realité du phénoméne traté ".

5) On touve des isolignes correspondants à certains valeures des données qui ne passent pas par ces points,bien que la valeur de la spline

est la même. La raison en est qu'à partir de la grille construite,on utilise une méthode de tracé qui ne tient pas compte de la spline obtenue.

CHAPITRE - II

ADAPTATION DES METHODES DECRITES PRECEDEMMENT AUX FONCTIONS SPLINES INVARIANTES PAR ROTATION EN DIMENSION $d > 2$

I.1 - INTRODUCTION

La théorie des fonctions splines invariantes par rotation a été développée par Duchon [12] ; pour trouver ces splines on verra qu'il faut résoudre un système semblable à (1.3) avec K une matrice $n \times n$ liée au noyau reproduisant (type de spline) et E une matrice $n \times p$ où p est la dimension de l'espace nul de X^{ins} (explicité plus loin). L'adaptation des méthodes décrites sera immédiate et nous citerons les différences dans l'utilisation des méthodes.

Dans la suite nous prenons les notations et résultats de Duchon [12]. (voir aussi [43], [44]).
Si X est un espace semi- hilbertien nous appellerons espace nul de X l'ensemble N des vecteurs de semi-norme nulle.

Soit T un ensemble quelconque $\mathbb{R}^T$ l'espace de fonctions réellesdéfinies sur T, $\mathbb{R}^{(T)}$ l'espace des fonctions réelles définies sur T à support fini.

Si N est un sous-espace vectoriel de $\mathbb{R}^T$, nous noterons N° l'ensemble des $\lambda \in \mathbb{R}^{(T)}$ telles que $\Sigma\lambda(t) \quad \nu(t) = 0 \quad \forall\nu \in N$.

Nous appellerons espace semi-hilbertien de fonction sur T (ou sous-espace semi-hilbertien de $\mathbb{R}^T$), tous espace semi-hilbertien X inclus dans $\mathbb{R}^T$ ayant un noyau N de dimension finie, et tel que la convergence $\|f\| \to 0$ (au sens de la semi-norme de X) implique

$$\Sigma\lambda(t)\ f(t) \to 0 \quad \forall\lambda \in N^\circ.$$

Nous dirons que X ($\subset \mathbb{R}^T$) sépare les points de T si, quels que soient $a_1,\ldots,a_n \in T$ et $z_1,\ldots,z_n$ réels, il existe $v \in X$ vérifiant $v(a_i) = z_i$ $i = 1,\ldots,n$.

Soit X un espace semi-hilbertien de fonctions sur T, d'espace nul N. Nous appellerons noyau reproduisant de X, tout noyau $\mathcal{K}(t,t')$ défini sur $T \times T$, à valeurs réelles tel que :

pour tout $\lambda \in N^0 \subset \mathbb{R}^{(T)}$ on ait

$\Sigma\lambda(t)\ \mathcal{K}(.,t) \in X$ avec

$\langle\langle \Sigma\lambda(t)\ \mathcal{K}(.,t), v\rangle\rangle = \Sigma\lambda(t)\ v(t) \qquad \forall v \in X$

<u>Théorème 3</u> (voir [12] [44])

Soit X un espace semi-hilbertien de fonctions sur T, séparant les points de T ; $a_1,\dots,a_n$ sont des points (distincts) de T tels qu'aucune fonction de semi-norme nulle ne puisse s'annuler en a_i $i = 1,\dots,n$ sans être identiquement nulle et $z_1,\dots,z_n$ sont des réels quelconques, alors :

1°) Il existe $\sigma \in X$ unique (fonction spline d'interpolation) vérifiant $\sigma(a_i) = z_i \quad i = 1,\dots,n$
et

$$\|\sigma\| = \inf_{v\in X} \{\|v\|,\ v(a_i) = z_i \quad i = 1,\dots,n\}$$

de plus si $\mathcal{K}(t,t')$ est un noyau reproduisant de X

(2.1) $$\boxed{\sigma(t) = \sum_{i=1}^{n} \lambda_i\ \mathcal{K}(t,a_i) + \sum_{k=1}^{p} \alpha_k \nu_k(t)}$$

où $\Sigma\lambda_i\ v_j(a_i) = 0 \quad \forall v \in N \quad j = 1,\dots,p$
avec $(\nu_1,\dots,\nu_p)$ une base de N (dim N = p).

2°) Quel que soit $\rho > 0$ il existe $\sigma_\rho \in X$ unique (fonction spline d'ajustemer vérifiant :

$$\|\sigma_\rho\|^2 + \rho \sum_{i=1}^{n} |\sigma_\rho(a_i) - z_i|^2 = \inf_{v\in X} \{\|v\|^2 + \rho \sum_{i=1}^{n} |v(a_i) - z_i|^2\}$$

et

(2.2) $$\boxed{\sigma_\rho(t) = \sum_{i=1}^{n} \lambda_i^\rho\ \mathcal{K}(t,a_i) + \sum_{i=1}^{p} \alpha_k^\rho\ \nu_k(t)}$$

avec : $\sum_{j=1}^{n} \nu_k(a_j)\lambda_j^\rho = 0 \qquad k = 1,\dots,p$

$\lambda_i^\rho = \rho(z_i - \sigma_\rho(a_i)) \qquad i = 1,\dots,n$

Considérons maintenant $T = \mathbb{R}^d$ muni de la norme euclidienne usuelle $|t|^2 = t_1^2 + \dots + t_d^2$.

Pour tout $\theta > 0$ posons :

$$A_\theta(t) = |t|^\theta \qquad \text{si } \theta \text{ n'est pas un entier pair}$$

$$A_\theta(t) = |t|^\theta \operatorname{Log}|t| \qquad \text{si } \theta \text{ est entier pair} \tag{2.3}$$

Appelons $\tilde{H}^s$ l'ensemble des distributions tempérées u dont la transformée de Fourier $\hat{u}$ est une fonction vérifiant :

$$\int |\hat{u}(\tau)|^2 |\tau|^{2s} d\tau < +\infty$$

$$\|u\|_s = \| |\tau|^2 \hat{u} \|_{L^2}$$

X^{ms} étant l'ensemble des distributions u dont toutes les dérivées d'ordre m ($D^\alpha u$, $|u| = m$) sont dans $\tilde{H}^s$, munissons X^{ms} de la semi-norme $\|u\|_{ms} = (\Sigma \|D_{i_1} \dots D_{i_m} u\|_s^2)^{1/2}$ et du semi-produit scalaire correspondant.

$$\langle\langle u,v \rangle\rangle_{ms} = \Sigma \int_{\mathbb{R}^d} (\mathcal{F} D_{i_1} \dots D_{i_m} u)(\tau) \overline{(\mathcal{F} D_{i_1} \dots D_{i_m} v)}(\tau) |\tau|^{2s} d\tau \tag{2.4}$$

<u>Théorème 4</u> (voir [12], [44])

Si $-m + \frac{d}{2} < s < \frac{d}{2}$, X^{ms} est un espace semi-hilbertien de fonctions continues sur $\mathbb{R}^d$, d'espace nul P_{m-1} (Polynômes de degré $\leq m-1$), séparant les points de $\mathbb{R}^d$, alors un noyau reproduisant de X^{ms} est proportionnel à $A_{2m+2s-d}(t-t')$.

Alors en posant $\mathcal{K}(t,t') = A_{2m+2s-d}(t-t')$ on obtient la fonction spline d'interpolation (ou d'ajustement).

II.2 - CALCUL DE LA FONCTION SPLINE D'INTERPOLATION ET D'AJUSTEMENT AU MOYEN DES METHODES DECRITES AU CHAPITRE I.

Définition 1

Nous dirons que l'ensemble fini $\mathcal{A}$ de points distincts de $\mathbb{R}^d$ est P_{m-1} unisolvent si aucun polynôme de degré $\leq (m-1)$ ne peut s'annuler sur $\mathcal{A}$ sans être identiquement nul.

Supposons que t^i $i = 1,\dots,n$ $t^i \in \mathbb{R}^d$ (points distincts) contient un sous-ensemble P_{m-1} unisolvent ; et soient z_i $i = 1,\dots,n$ réels. De plus admettons que m, d et s vérifient la condition du théorème 4.

En considérant ν_j $j = 1,\dots,p$ une base de P_{m-1} on a alors :

1°) La fonction spline d'interpolation s'écrit comme (2.1) où les paramètres $\lambda_1,\dots,\lambda_n$, $\alpha_1,\dots,\alpha_p$ seront calculés à partir du système :

$$\begin{bmatrix} K & E \\ E^T & O \end{bmatrix} \begin{bmatrix} \Lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} Z \\ O \end{bmatrix} \tag{2.5}$$

avec

$$K = [K_{ij}]_{n\times n} \qquad K_{ij} = \mathcal{K}(t^i,t^j)$$

$$E = [E_1,\dots,E_p]_{n\times p} \qquad E_j = [e_{ij}] \in \mathbb{R}^n \qquad e_{ij} = \nu_j(t^i)$$

$$Z = [Z_i] \quad \Lambda = [\lambda_i] \quad \alpha = [\alpha_j] \qquad i = 1,\dots,n \quad j = 1,\dots,p$$

2°) La fonction spline d'ajustement s'écrit comme (2.2) et les paramètres $\lambda_1^\rho,\dots,\lambda_n^\rho$, $\alpha_1^\rho,\dots,\alpha_p^\rho$ seront déterminées à partir du système :

$$(2.6) \qquad \begin{bmatrix} \tilde{K} & E \\ E^T & O \end{bmatrix} \begin{bmatrix} \Lambda^\rho \\ \alpha^\rho \end{bmatrix} = \begin{bmatrix} Z \\ O \end{bmatrix}$$

où

$$\tilde{K} = K + \frac{1}{\rho} I_n \qquad (\rho > 0).$$

Remarque

1) On ne discutera que du système (2.5), car le système (2.6) peut être résolu de la même façon en changeant la matrice K en $\tilde{K}$.

2) L'hypothèse (H) du chapitre I peut être formulée aussi :

"Les trois premiers points forment un ensemble P_1-unisolvent".

Cette nouvelle forme d'expression de l'hypothèse (H) nous indique que pour utiliser une démarche semblable à la méthode (A) il faudrait faire l'hypothèse suivante :

$(\tilde{H})$ "Les p premiers points (t^i $i=1,\dots,p$) forment un ensemble P_{m-1} unisolvent".

Reprenons la notation du chapitre I c'est-à-dire $\mathcal{E} \subset \mathbb{R}^n$ (espace engendré par les p vecteurs de E) $\mathcal{F} = \mathcal{E}^\perp \subset \mathbb{R}^n$.

METHODE A

Sous l'hypothèse $\tilde{H}$ on peut construire la base suivante de $\mathcal{F}$.

$$(2.7) \qquad r_i = \begin{bmatrix} C_i \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \\ \rightarrow (i+p)\text{ème composante} \\ \\ \\ \\ \end{array} \qquad \begin{array}{l} r_i \in \mathbb{R}^n \\ C_i \in \mathbb{R}^p \end{array}$$

$$i = 1,\dots,n-p$$

C_i est déterminée à partir de la relation d'orthogonalité :

(2.8) $$\boxed{E^T r_i = 0}$$

Si $R = [r_1, \ldots, r_{n-p}]$ on a en utilisant (2.5)

$$R^T K R \gamma = R^T Z \quad \text{avec} \quad \Lambda = R\gamma$$

soit

$$A = R^T K R \qquad b = R^T Z$$

on a donc

(2.9) $$\boxed{A\gamma = b}$$

Remarque

a) A matrice (n-p) × (n-p) est définie positive et le système (2.9) peut être résolu de la même façon que dans le cas de plaque mince, mais l'hypothèse $(\tilde{H})$ qui est à la base de cette méthode est délicate, car au moins il nous faut un bon conditionnement du système (2.8).
De plus, il est souhaitable que pour le choix de ces p points, le systè (2.9) soit bien conditionné.

b) Dès que m > 2, il est difficile de trouver un système de p points P_{m-1}-unisolvant.

Par exemple si m = 3, d = 2 on aura p = 6 et trouver 6 points formant un système P_2-unisolvant et qui tienne compte de la remarque (a) est ch très difficile.

Alors la limitation de cette méthode (A) pour trouver la spline d'inter polation (ou ajustement) est l'hypothèse $\tilde{H}$.

METHODE B

De la même façon que pour la plaque mince on peut construire $V_1,\dots,V_p$ p vecteurs orthonormés de $\mathcal{E}$ et si $V = [V_1,\dots,V_p]$ posons :

$$Q = VV^T$$
$$P = I-Q$$

La relation (2.5) peut être ramenée à :

(2.10) $$A\Lambda = b \quad \text{avec} \quad \begin{cases} A = PKP \\ b = PZ \end{cases}$$

Remarque

a) A est une matrice $n \times n$ semi-définie positive de rang égal à $(n-p)$, (2.10) peut être alors résolu de la même façon que (1.11), c'est-à-dire au moyen de (1.14).

Cette méthode (B) n'exige aucune condition sur les données t^i $i=1,\dots,n$ facilement utilisable pour n'importe quelle valeur de m et d.

METHODES ITERATIVES

Pour les méthodes itératives on peut voir facilement que si on définit :

$$f(\Lambda) = \frac{1}{2}\Lambda^T K\Lambda - \Lambda^T Z \qquad \Lambda \in \mathbb{R}^n$$

f admet dans $\mathcal{F}$ un minimum unique Λ^* solution du système (2.5) et :

$$f(\Lambda^*) = \min_{\Lambda \in \mathcal{F}} f(\Lambda)$$

Alors les deux méthodes itératives (C) et (D) sont données par les procédés :

METHODE (C)

$$(1.11)\quad \begin{cases} X_1 = \Lambda_i & i = 1,\dots \quad (\Lambda_1 \in \mathcal{F} \text{ don} \\ X_{j+1} = X_j + \theta_j r_j & j = 1,\dots,n-p \\ \Lambda_{i+1} = X_{n-p+1} & \end{cases}$$

où

$$\theta_j = -\frac{\rho_j^T r_j}{r_j^T K r_j} \qquad \rho_j = KX_j - Z$$

METHODE (D)

$$(1.12)\quad \begin{cases} \Lambda_1 \in \mathcal{F} \text{ donné} \\ \Lambda_{j+1} = \Lambda_j + \theta_j \tilde{\rho}_j \end{cases}$$

où

$$\theta_j = -\frac{\tilde{\rho}_j^T \tilde{\rho}_j}{\tilde{\rho}_j^T K \tilde{\rho}_j} \qquad \tilde{\rho}_j = \mathrm{Proj}_{\mathcal{F}}(\rho_j) \qquad \rho_j = K\Lambda_j - Z$$

Remarque :

1) Chacune de ces méthodes peut être accélérée de façon analogue à (1.18) ou (1.24).

2) Dans le procédé (1.12) le vecteur $\tilde{\rho}_j$ peut être calculé par :

a) soit $\tilde{\rho}_j = \rho_j - E\beta_j$ avec $\beta_j = (E^T E)^{-1} E^T \rho_j$

b) soit $\tilde{\rho}_j = P\rho_j$ avec $P = I - VV^T$

V définie par la méthode (B).

CHAPITRE - III

CONSTRUCTION D'UNE SURFACE D'INTERPOLATION POUR UN GRAND NOMBRE DE NOEUDS D'INTERPOLATION

III-1 - INTRODUCTION

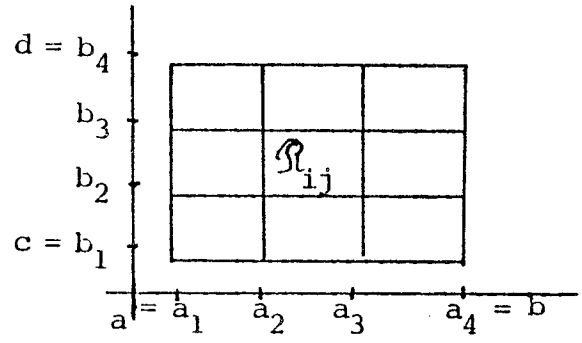
Supposons qu'on ait t^i $i = 1,\dots,n$ n points de $\mathbb{R}^2$ non tous alignés

La construction d'une fonction spline, par exemple du type plaque mince (ou Pseudo-Cubique) va être pratiquement impossible car pour les méthodes directes le problème d'encombrement va jouer le rôle decisif et pour les méthodes itératives les coûts vont être élevés. De plus, l'évaluation de la spline en un seul point devient alors difficile et coûteux.

Ainsi nous allons proposer le calcul d'une fonction S qui n'est plus une spline globale mais une combinaison de splines locales construites en des sous-ensembles de noeuds donnés, ces sous-ensembles contenant un nombre raisonnable de noeuds de façon à pouvoir utiliser les méthodes directes pour le calcul de splines locales.

Pour trouver S on procède de la façon suivante :
le domaine $[a,b] \times [c,d]$ est divisé en sous rectangles (q dans le cas de la figure 1)

(3.1)



Notons $\mathfrak{R}_{ij} = [a_i, a_{i+1}[\times [b_j, b_{j+1}[$ $\quad i = 1,2,3$ $\quad j = 1,2,3$

n_{ij} = nombre de points dans $\mathfrak{R}_{ij}$

Faisons l'hypothèse que $n_{ij} \leq 35$ et que chaque $\mathfrak{R}_{ij}$ contient à peu près le même nombre de points.

Soit $\sigma_{i,j}$ $i = 1,2$ $j = 1,2$ la fonction spline locale, calculée e utilisant les noeuds qui se trouvent dans

$$\mathcal{R}_{ij} \cup \mathcal{R}_{i+1,j} \cup \mathcal{R}_{i+1,j+1} \cup \mathcal{R}_{i,j+1} = [a_i, a_{i+2}[\times [b_j, b_{j+2}[.$$

Alors nous pouvons considérer l'application S de $\mathbb{R}^2$ dans $\mathbb{R}$ restrei au rectangle $[a,b] \times [c,d]$ telle que, si $(x,y) \in \mathcal{R}_{i,j}$ la valeur de S en ce point est une combinaison convexe "convenable" (explicité plus loin) de toutes les splines qui utilisent les noeuds de $\mathcal{R}_{ij}$ pour leur détermination.

Par exemple si :

$i = 1$ $j = 1$ il y a une seule spline locale σ_{11}

$i = 1$ $j = 2$ il y a deux splines locales σ_{11} et σ_{12}

$i = 2$ $j = 2$ il y a quatre splines locales σ_{11}, σ_{21}, σ_{22} et σ_{12}

Ainsi il y aura une combinaison soit de quatre, soit de deux splines (sauf pour les points qui se trouvent dans les quatre petits rectangles des coins), et donc il faut avoir un jeu de quatre fonctions positives comme pondération de manière que S ait la même régularité que le type de la spline utilisée.

III-2 - CONSTRUCTION DE S POUR LE RECTANGLE CITE

Nous allons chercher des fonctions poids qui permettent les combinaisons des splines envisagées parmi les polynômes à deux dimensions de degré convenable, de manière que S ait la régularité de la spline.

CAS - Régularité C^1

Considérons α et β deux polynômes à une variable, à déterminer, de degré le plus bas possible, et tels que :

$$(3.2)\quad \begin{cases} \alpha(x) + \beta(x) = 1 & \forall x \in [0,1] \\ \alpha(x) \geq 0 & \beta(x) \geq 0 \end{cases}$$

On peut supposer $\alpha(0) = 1$, $\alpha(1) = 0$.

Nous avons deux conditions pour α et il suffit d'admettre la même régularité au voisinage de 0 et 1 (deux conditions de plus) pour déterminer un polynôme de degré 3.

Posons $\alpha'(0) = \alpha'(1) = 0$ alors on a :

$$\boxed{\alpha(x) = (x-1)^2\ (2x+1)}$$

On vérifie que $0 \leq \alpha(x) \leq 1$ $\forall x \in [0,1]$, et β peut être pris comme :

$\beta(x) = 1-\alpha(x)$

$$\boxed{\beta(x) = x^2(3-2x)}$$

alors on a :

$$(3.3)\quad \begin{cases} \alpha(x) = (x-1)^2\ (2x+1) \\ \beta(x) = x^2(3-2x) \end{cases}$$

α et β vérifient (3.2)

Notons maintenant :

$$(3.4)\quad \begin{cases} \hat{\alpha}_{11}(x,y) = \alpha(x)\ \alpha(x) \\ \hat{\alpha}_{21}(x,y) = \beta(x)\ \alpha(y) \\ \hat{\alpha}_{22}(x,y) = \beta(x)\ \beta(y) \\ \hat{\alpha}_{12}(x,y) = \alpha(x)\ \beta(y) \end{cases}$$

Comme α et β vérifient (3.2) alors :

$$(3.5)\quad \begin{cases} \hat{\alpha}_{11}(t)+\hat{\alpha}_{21}(t)+\hat{\alpha}_{22}(t)+\hat{\alpha}_{12}(t) = 1 & \forall t \in [0,1] \times [0,1] \\ \hat{\alpha}_{ij}(t) \geq 0 & i = 1,2 \\ & j = 1,2 \end{cases}$$

Considérons $t = (x,y) \in \mathcal{R}_{ij}$ (3.1) et soient

$$(3.6)\qquad \varphi_{ij} : \bar{\mathcal{R}}_{ij} \to [0,1] \times [0,1] \qquad i = 1,2,3$$

$$(x,y) \rightsquigarrow \left(\frac{x-a_i}{a_{i+1}-a_i}, \frac{y-b_j}{b_{j+1}-b_j}\right)$$

$$(3.7)\qquad \alpha_{gl}/\bar{\mathcal{R}}_{ij} = \hat{\alpha}_{gl} \circ \varphi_{ij} \qquad \begin{matrix} g = 1,2 \\ l = 1,2 \end{matrix}$$

Définissons S de la manière suivante :

$$(3.8)\quad S(t)=\begin{cases} \sigma_{11}(t) & \text{si } t \in \mathcal{R}_{11} \\ \sigma_{21}(t) & \text{si } t \in \mathcal{R}_{31} \\ \sigma_{22}(t) & \text{si } t \in \mathcal{R}_{33} \\ \sigma_{12}(t) & \text{si } t \in \mathcal{R}_{13} \\ \alpha_{11}(x,b_1)\sigma_{11}(t)+\alpha_{21}(x,b_1)\sigma_{21}(t) & \text{si } t \in \mathcal{R}_{21} \\ \alpha_{12}(x,b_4)\sigma_{12}(t)+\alpha_{22}(x,b_4)\sigma_{22}(t) & \text{si } t \in \mathcal{R}_{23} \\ \alpha_{21}(a_4,y)\sigma_{21}(t)+\alpha_{22}(a_4,y)\sigma_{22}(t) & \text{si } t \in \mathcal{R}_{32} \\ \alpha_{11}(a_1,y)\sigma_{11}(t)+\alpha_{12}(a_1,y)\sigma_{12}(t) & \text{si } t \in \mathcal{R}_{12} \\ (\alpha_{11}\sigma_{11}+\alpha_{21}\sigma_{21}+\alpha_{22}\sigma_{22}+\alpha_{12}\sigma_{12})(t) & \text{si } t \in \mathcal{R}_{22} \end{cases}$$

<u>Proposition 8</u> :

L'application S définie comme ci-dessus est continue dans $[a,b] \times [c,d]$.

<u>Démonstration</u> :

Soit $t_o = (a_2,y_o)$ avec $y_o \in [b_2,b_3]$ $\quad (t_o \in \mathcal{R}_{22})$

alors $S(t_o) = (\alpha_{11}\sigma_{11}+\alpha_{21}\sigma_{21}+\alpha_{22}\sigma_{22}+\alpha_{12}\sigma_{12})(t_o)$

Comme $\alpha_{gl}(t_o) = (\tilde{\alpha}_{gl} \circ \varphi_{22})(t_o) = \tilde{\alpha}_{gl}(0, \frac{y_o-b_2}{b_3-b_2})$

on a $S(t_o) = \alpha(\frac{y_o-b_2}{b_3-b_2})\sigma_{11}(t_o) + \beta(\frac{y_o-b_2}{b_3-b_2})\ \sigma_{12}(t_o)$

Prenons $t \in \mathcal{R}_{12} \Rightarrow$

$$S(t) = \alpha_{11}(a_1,y)\sigma_{11}(t)+\alpha_{12}(a_1,y)\sigma_{12}(t)$$

$$= (\hat{\alpha}_{11} \circ \varphi_{12})(a_1,y)\sigma_{11}(t)+(\hat{\alpha}_{12} \circ \varphi_{12})(a_1,y)\sigma_{12}(t)$$

$$= \hat{\alpha}_{11}(0, \frac{y-b_2}{b_3-b_2})\sigma_{11}(t)+\hat{\alpha}_{12}(0, \frac{y-b_2}{b_3-b_2})\sigma_{12}(t)$$

$$= \alpha(\frac{y-b_2}{b_3-b_2})\sigma_{11}(t)+\beta(\frac{y-b_2}{b_3-b_2})\sigma_{12}(t)$$

alors $\lim\limits_{t\to t_o} S(t) = \alpha(\frac{y-b_2}{b_3-b_2})\sigma_{11}(t_o)+\beta(\frac{y-b_2}{b_3-b_2})\sigma_{12}(t_o) = S(t_o)$

Pour les autres points de la frontière des $\mathcal{R}_{ij}$ la démonstration est analogue.

<u>Proposition 9</u> :

$\frac{\partial S}{\partial x}$ et $\frac{\partial S}{\partial y}$ sont continuessur $[a,b] \times [c,d]$ (on admet que σ_{ij} est différentiable).

<u>Démonstration</u> :

Prenons $t_o = (a_2,y_o)$ avec $y_o \in [b_2,b_3]$ pour $t = (x,y)$ point à l'intérie de $\mathcal{R}_{22}$ on a :

$$\frac{\partial S}{\partial x} = \sum_{\substack{i=1\\j=1}}^{2} (\frac{\partial \alpha_{ij}}{\partial x})\sigma_{ij} + \sum_{\substack{i=1\\j=1}}^{2} \alpha_{ij} (\frac{\partial \sigma}{\partial x})$$

Notons $(\tilde{x},\tilde{y}) = \varphi_{22}(t)$ alors

$$\frac{\partial S}{\partial x}(t)=[\alpha'(\tilde{x})\alpha(\tilde{y})\sigma_{11}(t)+\beta'(\tilde{x})\alpha(\tilde{y})\sigma_{21}(t)+\beta'(\tilde{x})\beta(x)\sigma_{22}(t)+\alpha'(\tilde{x})\beta(\tilde{y})\sigma_{12}(t)]\frac{1}{a_3-a_2}$$

$$+\alpha_{11}(t)\frac{\partial}{\partial x}\sigma_{11}(t)+\alpha_{21}(t)\frac{\partial}{\partial x}\sigma_{21}(t)+\alpha_{22}\frac{\partial}{\partial x}\sigma_{22}(t)+\alpha_{12}(t)\frac{\partial\sigma_{12}}{\partial x}(t)$$

d'où $\lim\limits_{t\to t_o} \frac{\partial\sigma}{\partial x}(t) = \alpha(\frac{y_o-b_2}{b_3-b_2})\frac{\partial}{\partial x}\sigma_{11}(t_o)+\beta(\frac{y_o-b_2}{b_3-b_2})\frac{\partial}{\partial x}\sigma_{12}(t_o)$

D'autre part si $t = (x,y)$ est à l'intérieur de $\mathcal{R}_{12}$ on a :

$$\frac{\partial}{\partial x}\sigma(t) = \alpha_{11}(a_1,y)\frac{\partial}{\partial x}\sigma_{11}(t)+\alpha_{12}(a_1,y)\frac{\partial}{\partial x}\sigma_{12}(t) +$$

$$[\sigma_{11}(t)\alpha'(\tilde{y})+\sigma_{12}(t)\ \beta'(\tilde{y})]\ \frac{1}{b_3-b_2}$$

où

$$\tilde{y} = \frac{y-b_2}{b_3-b_2}$$

on a donc :

$$\lim_{t\to t_o}\frac{\partial S}{\partial x}(t) = \alpha\left(\frac{y_o-b_2}{b_3-b_2}\right)\frac{\partial}{\partial x}\sigma_{11}(t_o) + \beta\left(\frac{y_o-b_2}{b_3-b_2}\right)\frac{\partial}{\partial x}\ \sigma_{12}(t_o)$$

Pour les autres points de la primitive des $\mathcal{R}_{ij}$ et la dérivée par rapport à y la démonstration est analogue.

<u>CAS</u> : <u>Régularité C^2</u>

Si aux conditions $\alpha(0) = 1$

$\alpha(1) = 0$

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

on ajoute deux conditions aux points 0 et 1, et on obtient un polynôme de degré cinq.

Ainsi si

$\alpha''(0) = \alpha''(1) = 0$ on a :

$$\alpha(x) = (1-x)^3(6x^2+3x+1)$$

$$0 \le \alpha(x) \text{ si } x \in [0,1]$$

comme $\alpha'(x) = -3\ 10x^2(1-x)^2 \le 0$, α est décroissant

alors on a : $0 \le \alpha(x) \le 1 \qquad x \in [0,1]$

et on peut prendre $\beta(x) = 1-\alpha(x)$

donc on a :

$$(3.9)\quad \begin{cases} \alpha(x) = (1-x)^3(6x^2+3x+1) \\ \beta(x) = x^3(6x^2-15x+10) \end{cases}$$

Considérons $\hat{\alpha}_{ij}$, φ_{ij}, α_{ij} construits comme précédemment en changeant (3.3) en (3.9) et soit S définie comme en (3.8).

<u>Proposition 10</u> :

1°) S est continue sur $[a,b] \times [c,d] = R$

2°) $\frac{\partial S}{\partial x}$ $\frac{\partial S}{\partial y}$ sont continues sur R

3°) $\frac{\partial^2 S}{\partial x^2}$ $\frac{\partial^2 S}{\partial xy}$ $\frac{\partial^2 S}{\partial y^2}$ sont continues sur R

(évidemment on admet σ_{ij} deux fois dérivable).

III.3 - CONSTRUCTIONS DE S QUAND LES NOMBRES DE NOEUDS SONT ASSEZ GRANDS

Soit le rectangle $[a,b] \times [c,d]$ et supposons que :

$a = a_1 < a_2 < \ldots < a_{n_x+1} = b$ $\qquad (n_x > 2)$

$c = b_1 < b_2 < \ldots < b_{n_y+1} = d$ $\qquad (n_y > 2)$

Notons $\mathcal{R}_{ij} = [a_i, a_{i+1}[\times [b_j, b_{j+1}[$ $\qquad i = 1,\ldots,n_x$, $j = 1,\ldots,n_y$

n_{ij} nombre de points dans $\mathcal{R}_{ij}$

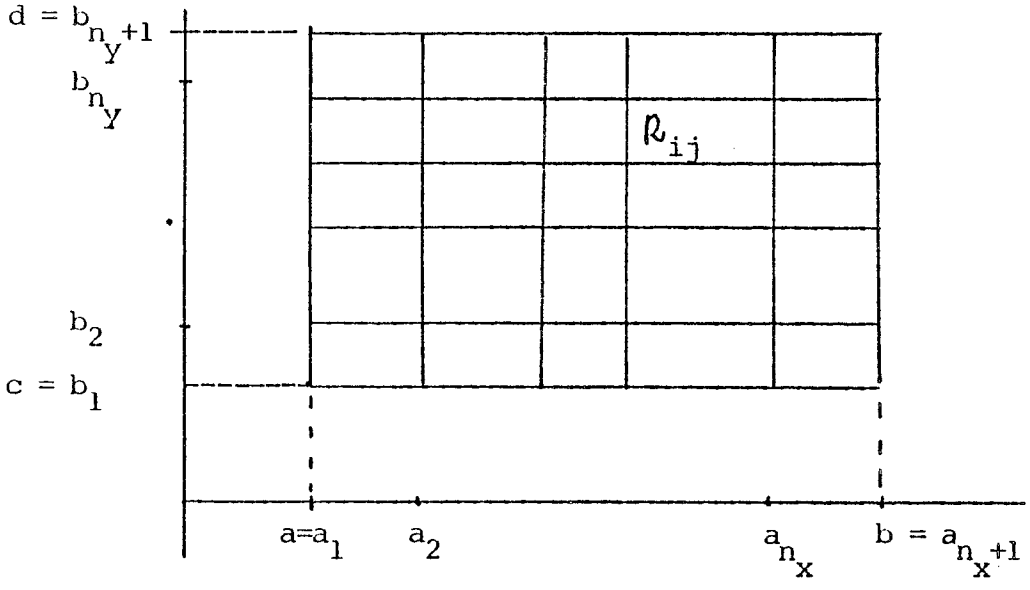
σ_{ij} spline construite avec les noeuds appartenant à l'ensembl

$$[a_i, a_{i+2}[\times [b_j, b_{j+2}[\qquad i = 1,\dots,n_x-1 \quad j = 1,\dots,n_y-1$$

$$N_{ij} = n_{ij}+n_{i+1,j}+n_{i+1,j+1}+n_{i,j+1}$$

Imposons $n_{ij} \leq 35$ de façon que N_{ij} soit au plus 140 (tous les n_{ij} étant du même ordre)

(3.9)



Prenons:

α, β définie par (3.3) ou (3.9) selon le cas

$\hat{\alpha}_{11}$, $\hat{\alpha}_{21}$, $\hat{\alpha}_{22}$, $\hat{\alpha}_{12}$ définies par (3.4)

φ_{ij} comme (3.6) $\qquad i = 1,\dots,n_x$

$\alpha_{gl/\bar{R}_{ij}}$ comme (3.7) $\qquad j = 1,\dots,n_y$

$g = 1,2$

$l = 1,2$

et soit S :

(3.10)

$$S(t) = \begin{cases} \sigma_{11}(t) & \text{si } t \in \mathcal{R}_{11} \\ \sigma_{n_x-1,1}(t) & \text{si } t \in \mathcal{R}_{n_x,} \\ \sigma_{n_x-1,n_y-1}(t) & \text{si } t \in \mathcal{R}_{n_x,} \\ \sigma_{1,n_y-1}(t) & \text{si } t \in \mathcal{R}_{1,n} \\ \alpha_{11}(x,b_1)\sigma_{i-1,1}(t)+\alpha_{21}(x,b_1)\sigma_{i,1}(t) & \text{si } t \in \mathcal{R}_{i,1} \quad i = 2,\dots,n_x \\ \alpha_{12}(x,d)\sigma_{i-1,n_y-1}(t)+\alpha_{22}(x,d)\sigma_{i,n_y-1}(t) & \text{si } t \in \mathcal{R}_{i,n} \quad i = 2,\dots,n_x \\ \alpha_{21}(b,y)\sigma_{n_x-1,j-1}(t)+\alpha_{22}(b,y)\sigma_{n_x-1,j}(t) & \text{si } t \in \mathcal{R}_{n_x} \quad j = 2,\dots, n \\ \alpha_{11}(a_1,y)\sigma_{1,j-1}(t)+\alpha_{12}(a_1,y)\sigma_{1,j}(t) & \text{si } t \in \mathcal{R}_{1,j} \quad j = 2,\dots,n_y \\ (\alpha_{11}\sigma_{i-1,j-1}+\alpha_{21}\sigma_{i,j-1}+\alpha_{22}\sigma_{ij}+\alpha_{12}\sigma_{i-1,j})(t) & \text{si } t \in \mathcal{R}_{ij} \quad i = 2,\dots,n_x \quad j = 2,\dots,n_y \end{cases}$$

<u>Remarque</u> :

Comme

$$(3.11). \begin{cases} \hat{\alpha}_{11}(0,y) = \hat{\alpha}_{21}(1,y) \\ \hat{\alpha}_{22}(1,y) = \hat{\alpha}_{12}(0,y) \\ \hat{\alpha}_{11}(x,1) = \hat{\alpha}_{12}(x,0) \\ \hat{\alpha}_{22}(x,0) = \hat{\alpha}_{21}(x,1) \end{cases}$$

alors on a :

$$\left.\begin{array}{l}\alpha_{11}(a_1,y) = \alpha_{21}(a_2,y)\\ \alpha_{12}(a_1,y) = \alpha_{22}(a_2,y)\end{array}\right\} \quad \text{sur } \mathcal{R}_{1,j} \qquad j = 2,\ldots,n_y-1$$

$$\left.\begin{array}{l}\alpha_{11}(a_{n_x},y) = \alpha_{21}(b,y)\\ \alpha_{12}(a_{n_x},y) = \alpha_{22}(b,y)\end{array}\right\} \quad \text{sur } \mathcal{R}_{n_x,j} \qquad j = 2,\ldots,n_y-1$$

$$\left.\begin{array}{l}\alpha_{12}(x,d) = \alpha_{11}(x,b_{n_x})\\ \alpha_{22}(x,d) = \alpha_{21}(x,b_{n_x})\end{array}\right\} \quad \text{sur } \mathcal{R}_{i,n_y} \qquad i = 2,\ldots,n_x-1$$

$$\left.\begin{array}{l}\alpha_{11}(x,b_1) = \alpha_{12}(x,b_2)\\ \alpha_{21}(x,b_1) = \alpha_{22}(x,b_2)\end{array}\right\} \quad \text{sur } \mathcal{R}_{i,1} \qquad i = 2,\ldots,n_x-1$$

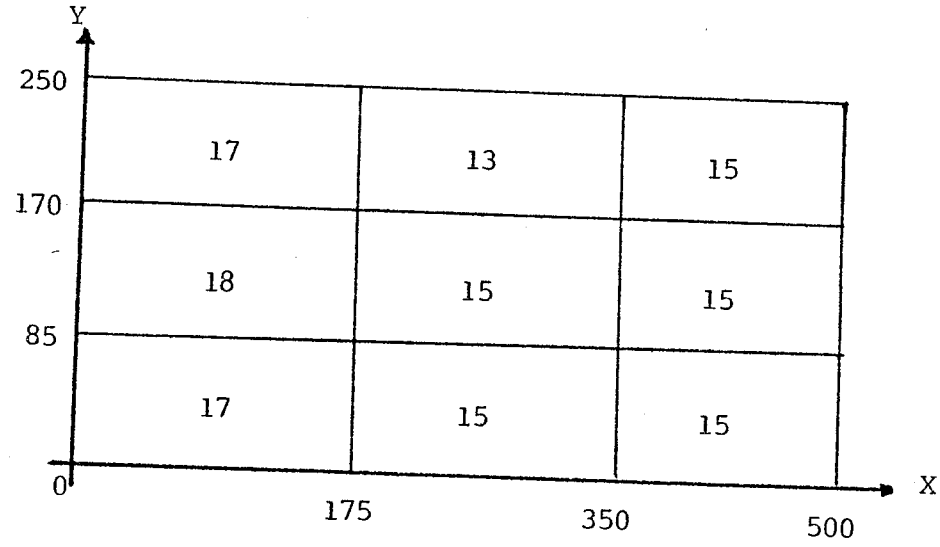
III.4 - RESULTATS NUMERIQUES ET CONCLUSIONS

Comme application on va présenter deux dessins des courbes de niveaux de la surface S définie par (3.10) ,les splines locales sont calculées au moyen de la méthode (B) ,la combinaison des splines utilisant (3.3) ,et le tracé de la même façon que dans le chapitre I .

Exemple 1 : Dessin N° 8

On prend les 140 points utilisés pour tracer le dessin N° 5, le découpage fait est de trois intervalles dans chaque axe du domaine.

Figure 1



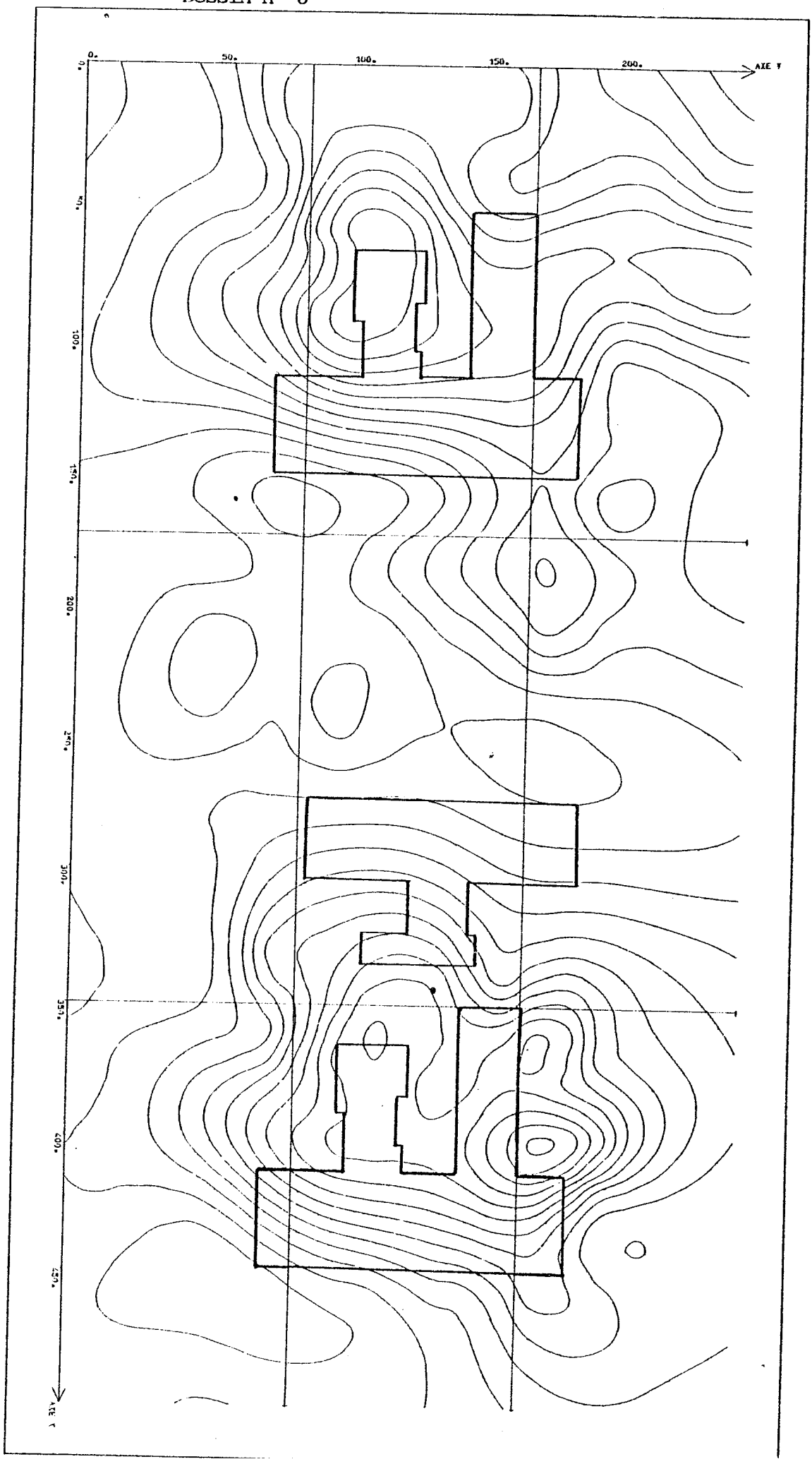
Les chiffres dans chaque rectangle $R_{i,j}$ de la figure 1 sont le nombre des points $n_{i,j}$ i=1,2,3 j=1,2,3 .La grille construite pour tracer les 20 isobruits est de 51x21 ; et comme temps d'éxecution on a:

Tableau 4

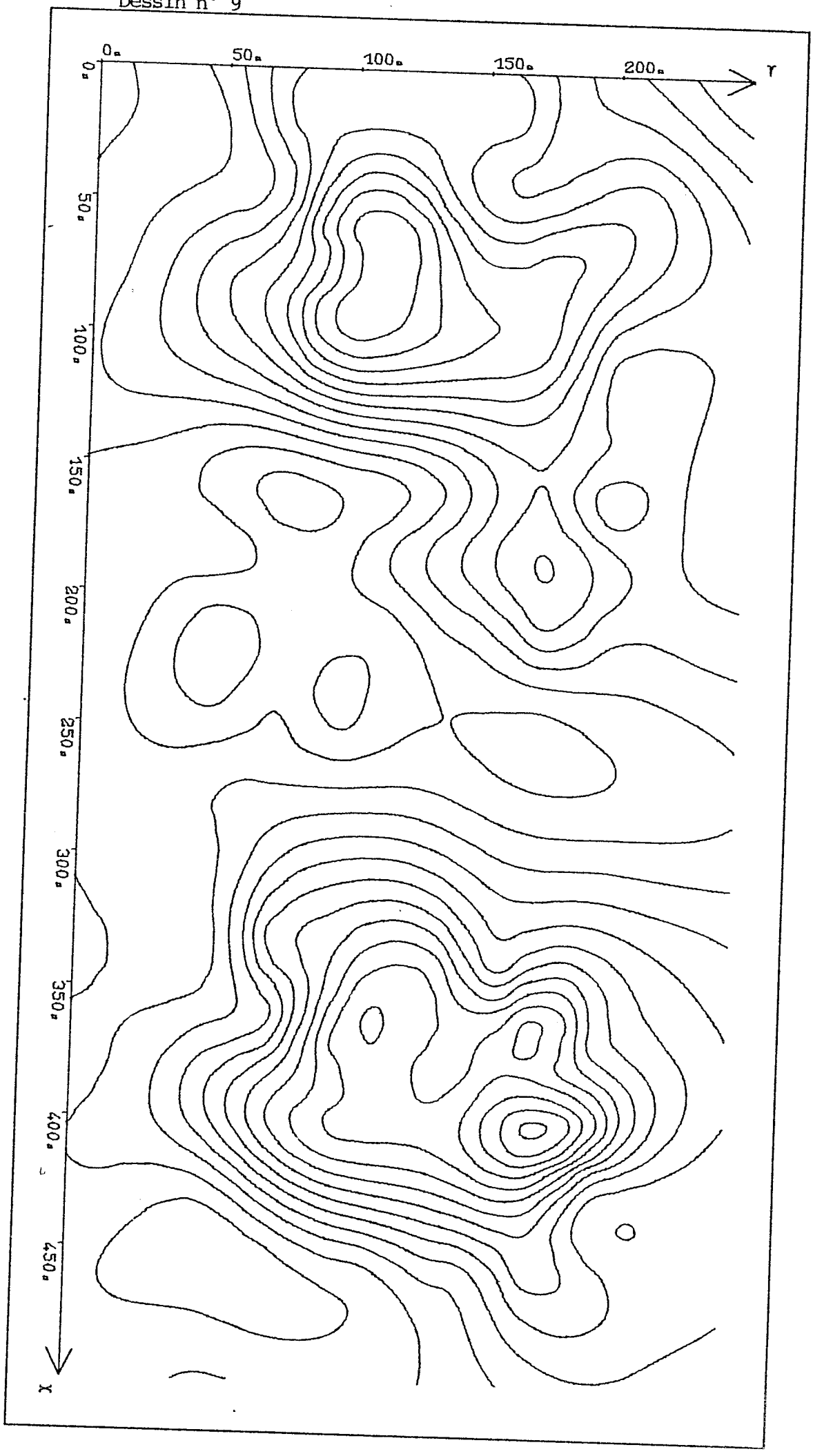
Calcul de S (4 Splines locales)	Calcul de la grille	Tracé
11.77	31.54	83.55

(secondes)

Dessin n° 8



Dessin n° 9


0m
50m
100m
150m
200m
250m
300m
350m
400m
450m
Y
0m
50m
100m
150m
200m
X

Remarques:

1) Utiliser la méthode (A) pour calculer les splines locales est dejà problematique à cause de l'hypothèse (H);car, pour le calcul de chacune des splines locales il nous faut une bonne disposition des trois premiers points, d'où l'importance de la méthode (B).

2) Par rapport à l'obtention de la spline globale (dessin N° 5) on a :

a.- Moins d'encombrement dans le calcul des splines locales , d'après la figure 1 on voit que $n_{i,j} \leq 65$ $i=1,2$ $j=1,2$.

b.- Le temps d'obtention des splines (quatre) et l'évaluation en chaque noeud de la grille ont eté réduites (tableaux 3 et 4).

c.- Peu des différences entre les dessins N° 5 et N° 8,c'est-à-dire que l'on a un résultat satisfaisant.

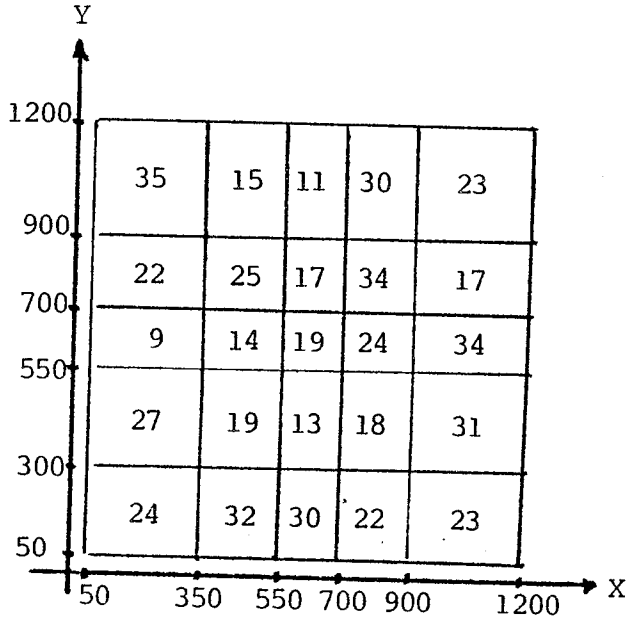
3) Le dissin N° 9 est obtenu sans tenir compte du point (76.,236.) (mesure fausse) , le découpage est la même que la figure 1 ($n_{1,3} = 16$ au lieu de 17).

Exemple 2 : Dessin N° 10

Relevé d'altitude dans la région de NOIRETABLE , 568 points dans le domaine $[50,1200] \times [50,1200]\ u^2$ ($u = \frac{1}{2}$ Km.) figure 3, le découpage qu'on a fait est de cinq intervalles dans chaque axe (figure 2), le nombre des isovaleurs à tracer est de 24 à partir d'une grille de 61×61, construite dans le domaine $[150,1050] \times [150,1050]\ u^2$

Le nombre de points $n_{i,j}$ dans $\mathcal{R}_{i,j}$ $i=1,\ldots,5$ $j=1,\ldots,5$ est:

Figure 2



Le temps d'éxecution est donné dans le tableau suivant

Tableau 5

Calcul de S (16 Splines locales)	Calcul de la grille	Tracé
96.82	283.94	301.83

(secondes)

Remarques:

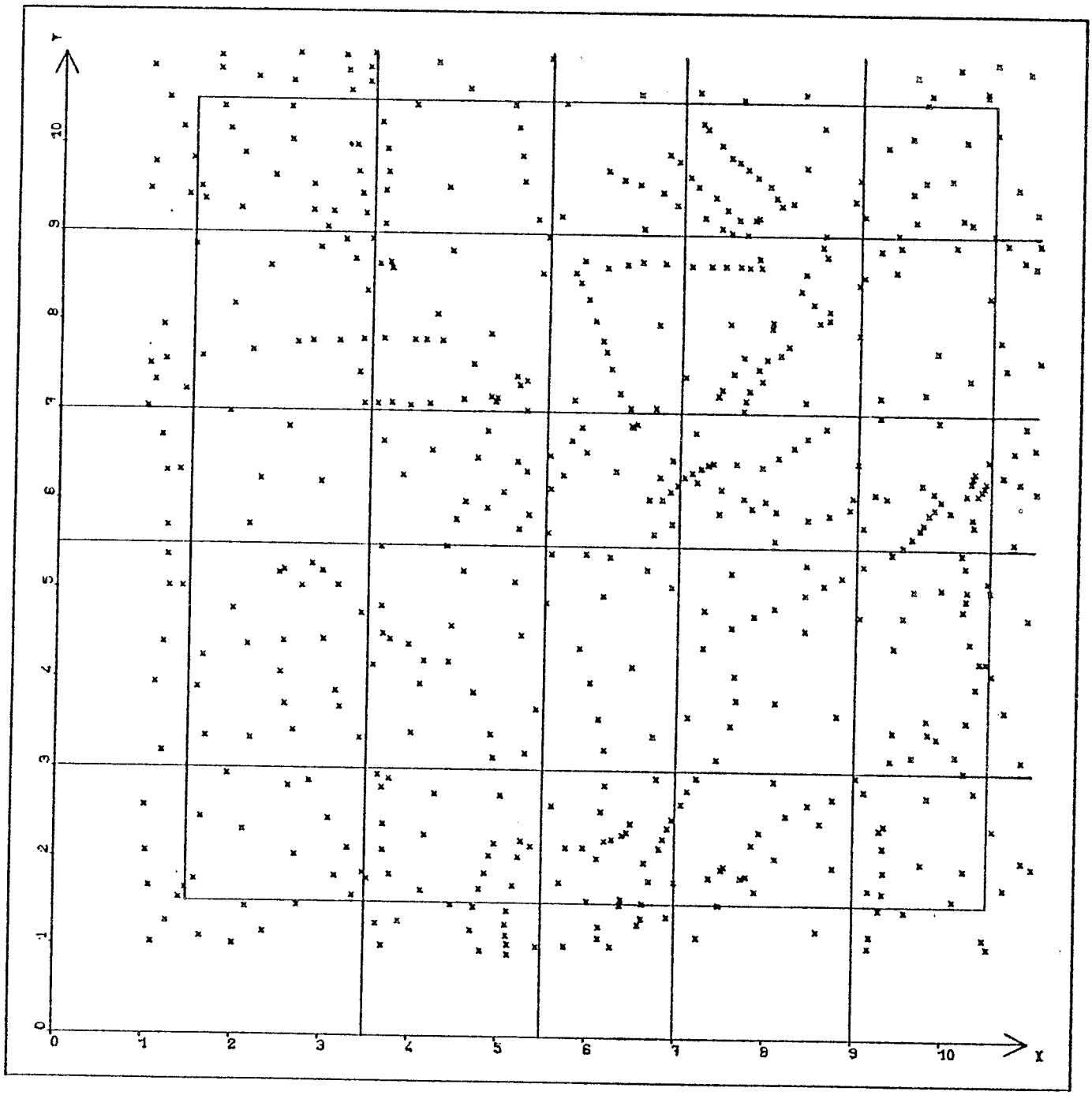
1) Le calcul de la spline globale est dejà impossible.

2) L'hypothèse (H) pour utiliser la méthode A est difficile à realiser pour chacune des splines locales,en tennant compte de la bonne disposition des trois premiers poits.

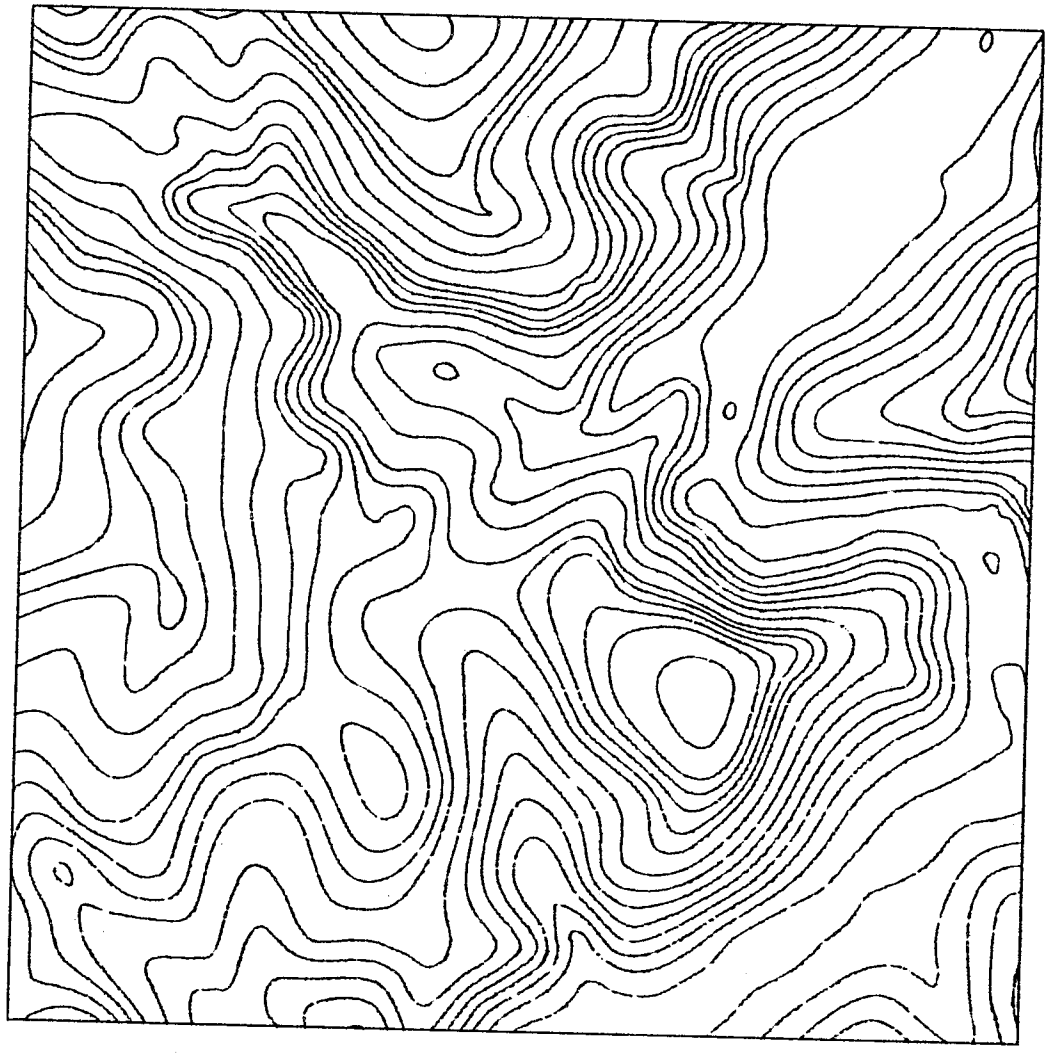
3) Si on compare les dessins Nº 10 et Nº 11 ,on voit qu'il y a certaines différences,sur tout dans le rectangle $\mathcal{R}_{1,3}$(niveau 605).

4) Les mêmes données ont eté utilisées pour tracer le dessin Nº12 avec une découpage en 6 intervalles dans chaque axe (figure 4)

Figure 3



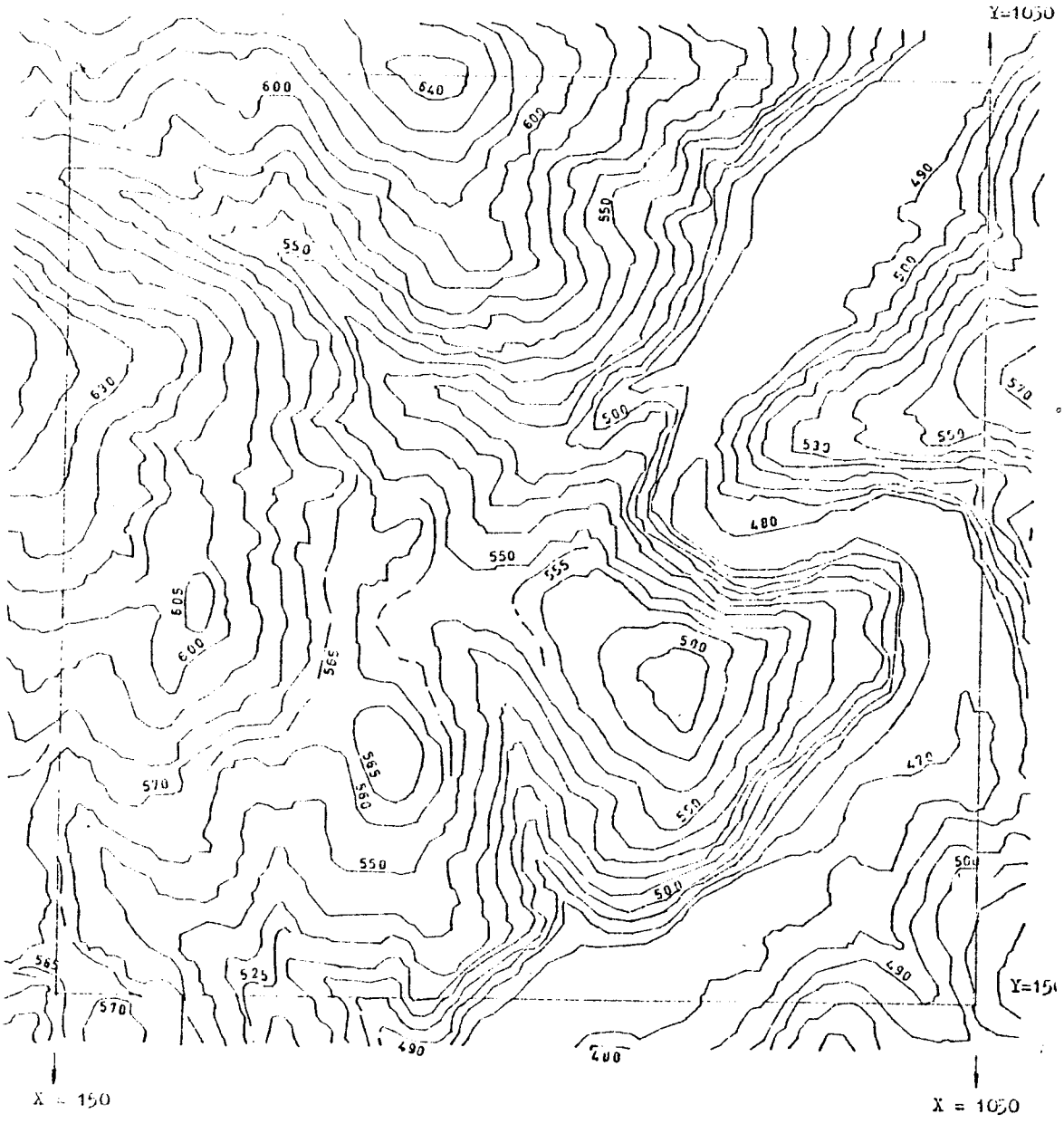
Dessin n° 10 (tracé automatique)



Dessin n° 11

Noirétable

Courbes de niveau obtenues par photogrammétrie classique



Dessin n° 12 (tracé automatique)

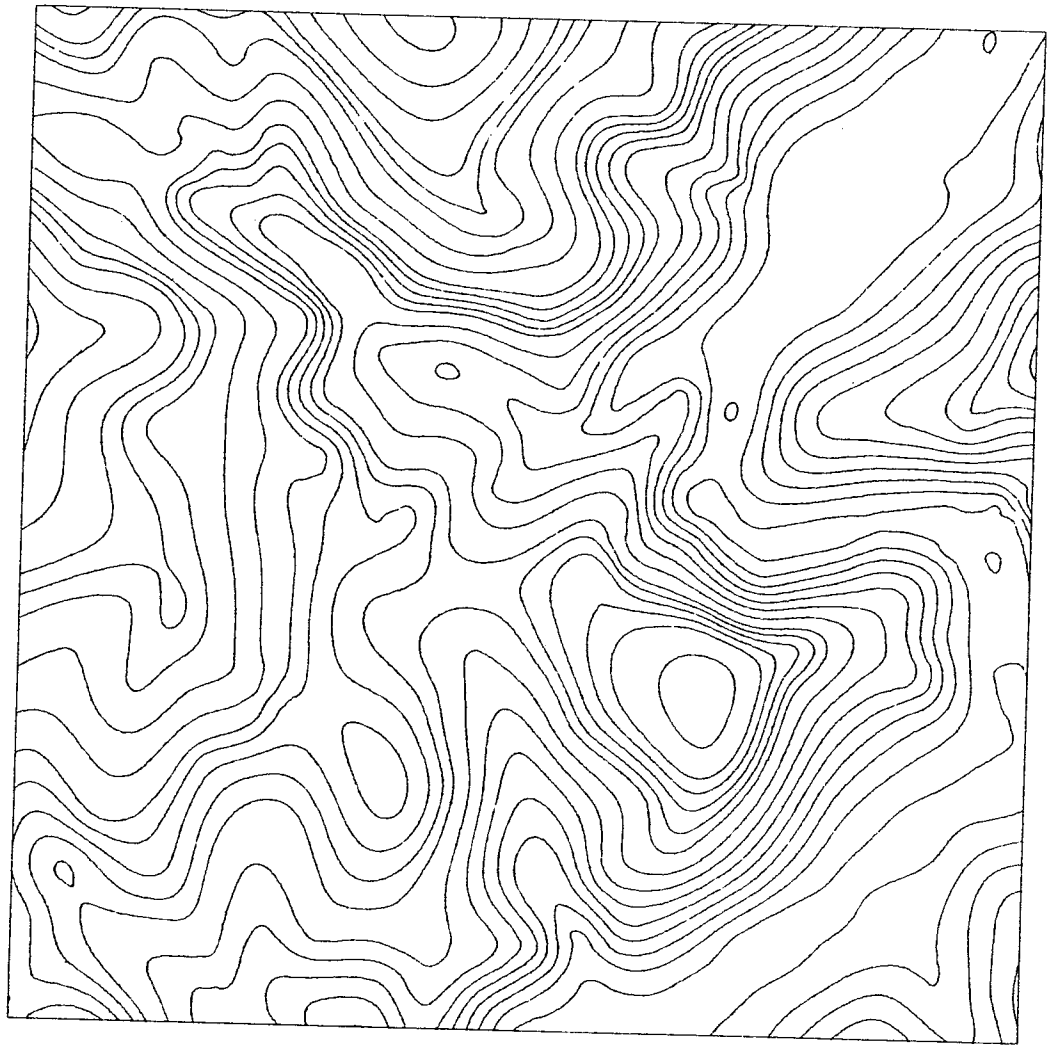
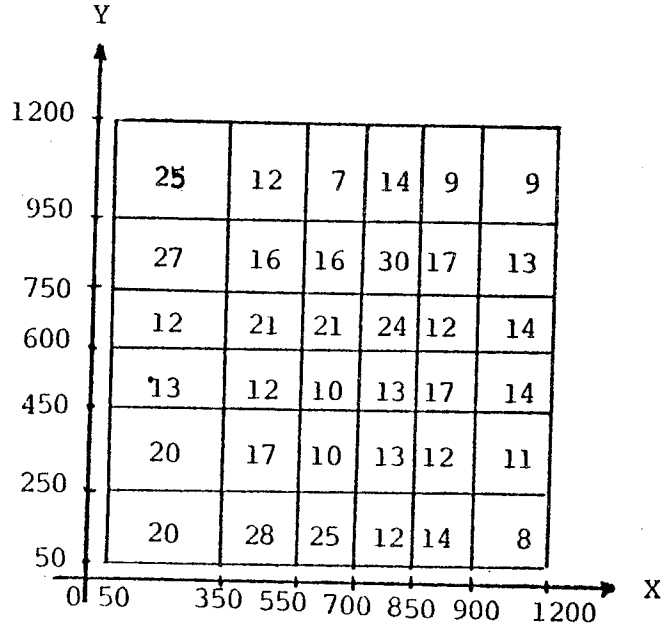


Figure 4



Le temps d'éxecution obtenu est:

Tableau 6

Calcul de S (25 Splines locales)	calcul de la grille	Tracé
75.2	194.3	300

(secondes)

1) Quand le nombre des points est assez grand,il est préférable d'utiliser les splines locales,en utilisant les pondérations citées,et la méthode B pour l'obtention des splines.

2) Le temps de calcul des splines locales par rapport aux évaluations de S aux noeuds de la grille regulière et le tracé est faible; de même soulignons que le découpage fait,jeu en rôle important ,pour une part dans le temps de calcul des splines et la grille (tableaux 5 et 6), et pour autre part la bonne réprésentation du phénomène.

3) La technique de découpage énoncé ,nous donne des résultats satisfaisants,bien que S est utilisé que pour le calcul des valeurs aux noeuds de la grille regulière. L'idéal serait de tracer les isolignes utilisant S de manière directe.

```
C     PROGRAMME DE CALCUL DES VALEURS D'UNE FONCTION Z=F(X,Y) SUR UN
C     MAILLAGE RECTANGULAIRE ETANT DONNEES LES VALEURS DE CETTE FONCTION
C     EN UN NOMBRE DE POINTS ARBITRAIREMENT DONNEES
C     DEFINITION DES VARIABLES UTILISEES
C     A(I),I=1,NX+1 INDIQUE LES FRONTIERES DES RECTANGLES SELON L'AXE X
C     B(J),J=1,NY+1 INDIQUE LES FRONTIERES DES RECTANGLES SELON L'AXE Y
C     ON DELIMITE DONC NX*NY RECTANGLES R(I,J)
C     VX,VY,VZ DESIGNENT UN TRIPLET DE DONNEES
C     ON RANGE CES DONNEES DANS X(I,J,K),Y(I,J,K),Z(I,J,K)
      REAL*8 S,T,R0
      REAL*8 V1,V2,V3,V4,VALSP
      REAL*8 ALPHA,ALPHAR,ALPHAY,BETA,BETAR,BETAY
      DIMENSION F(65,65)
      DIMENSION A(10),B(10),NBX(10),NBY(10)
      DIMENSION IX(568),IY(568),IZ(568)
        COMMON /CVALSP/S(2,9,140),T(2,9,3)
C     LECTURE DU NOMBRE DE FRONTIERES DE RECTANGLES EN X,EN Y
C     LECTURE DU NOMBRE TOTAL DE POINTS,DU NOMBRE DE POINTS DU
C     MAILLAGE SELON L'AXE DES X ET SELON L'AXE DES Y
      COMMON /CSPL/X(10,10,35),Y(10,10,35),Z(10,10,35),N(10,10),NYM1
C     LECTURE DES BORNES DES RECTANGLES R(I,J)
      DATA NX/5/,NY/5/,NPX/61/,NPY/61/,NT/568/
      DATA AA/150./,BB/1050./,CC/150./,DD/1050./,R0/0./
      HX=(BB-AA)/(NPX-1)
      HY=(DD-CC)/(NPY-1)
      NYM1=NY-1
      NX1=NX+1
      NY1=NY+1
      READ(5,2)(A(I),I=1,NX1)
      READ(5,2)(B(J),J=1,NY1)
C       FORMATS
    2 FORMAT(10F8.2)
    4 FORMAT(3(3I5,10X))
C     LECTURE DES DONNEES,ET REMPLISSAGE DES TABLEAUX X,Y,Z,N
      READ(5,4)((IX(I),IY(I),IZ(I)),I=1,NT)
      DO 10 I=1,NX
      DO 11 J=1,NY
      N(I,J)=0
   11 CONTINUE
   10 CONTINUE
      DO 3 II=1,NT
      VX=IX(II)
      VY=IY(II)
      VZ=IZ(II)
      I1=1
5     I1=I1+1
      IF(VX.GE.A(I1))GOTO 5
      I=I1-1
      I1=1
    6 I1=I1+1
```

```
      IF (VY.GE.B(I1))GOTO 6
      J=I1-1
      N(I,J)=N(I,J)+1
      K=N(I,J)
      X(I,J,K)=VX
      Y(I,J,K)=VY
      Z(I,J,K)=VZ
    3 CONTINUE
      WRITE(6,9)((N(I,J),I=1,NX),J=1,NY)
    9 FORMAT(6I5)
C
C       LECTURE DES NIVEAUX DES COURBES DE NIVEAU
C       LECTURE DE LA DISTANCE ENTRE POINTS DE LA GRILLE SELON X ET Y
C     LECTURE DU NOMBRE DE POINTS DU MAILLAGE SELON L'AXE DES X ET
C     SELON L'AXE DES Y
C     CALCUL DE PAS DU MAILLAGE SELON L'AXE DES X ET SELON L'AXE DES Y
C     NOMBRE DE POINTS SELON L'AXE DES X,ET SELON L'AXE DES Y
      A(1)=AA
      A(NX1)=BB
      B(1)=CC
      B(NY1)=DD
      ORX=AA
      DO 100 I=1,NX
      NBX(I)=(A(I+1)-ORX)/HX+1
  100 ORX=ORX+NBX(I)*HX
      ORY=CC
      DO 200 J=1,NY
      NBY(J)=(B(J+1)-ORY)/HY+1
  200 ORY=ORY+NBY(J)*HY
C       IMPRESSION DES DONNEES
        WRITE(6,2000)(NBX(I),I=1,NX)
2000    FORMAT(1H ,'NOMBRE DE POINTS SELON X PAR RECTANGLE',6I4)
        WRITE(6,2001)(NBY(J),J=1,NY)
2001    FORMAT(1H ,'NOMBRE DE POINTS SELON Y PAR RECTANGLE',6I4)
C     ON CALCULE LES VALEURS DE Z=F(X,Y) SUR LE MAILLAGE COLONNE PAR
C     COLONNE(A CHAQUE ETAPE DU CALCUL ON CONSERVE LES VALEURS DES
C     COEFFICIENTS DES FONCTIONS SPLINE CORRESPONDANT A DEUX COLONNES
C     CONSECUTIVES
C     NTX;NTY SONT LES INDICES DE NUMEROTATION SELON L'AXE DES X ET SELON
C     L'AXE DES Y
      CALL CHRONO
        CALL SPLCOL(1,1,S,T,R0)
      NTX=0
      NTY=0
      ORX=AA-HX
      ORY=CC-HY
      OY=ORY
C     CAS L=1
C     CALCUL DE LA VALEUR DE LA SPLINE SUR LES POINTS DU MAILLAGE DEP(1,1)
      NBY1=NBY(1)
```

```
      NBX1=NBX(1)
      DO 398 J=1,NBY1
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      DO 399 I=1,NBX1
      OX=OX+HX
      F(I,NTY)=VALSP(1,1,1,OX,OY)
  399 CONTINUE
  398 CONTINUE
C     CALCUL POUR DES POINTS DU MAILLAGE DE LA PREMIERE COLONNE
C     SANS APPARTENIR AU RECTANGLE SUPERIEUR
      NBX1=NBX(1)
      DO 400 L=2,NYM1
        NBYL=NBY(L)
      B1=B(L+1)-B(L)
      DO 500 J=1,NBYL
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
C     ZR VARIABLE REDUITE
      ZR=(OY-B(L))/B1
      ALPHAR=ALPHA(ZR)
      BETAR=BETA(ZR)
      DO 600 I=1,NBX1
      OX=OX+HX
      V2=VALSP(1,1,L,OX,OY)
      V1=VALSP(1,1,L-1,OX,OY)
      F(I,NTY)=ALPHAR*V1+BETAR*V2
  600 CONTINUE
  500 CONTINUE
  400 CONTINUE
C     L=NY
      NBYNY=NBY(NY)
      DO 501 J=1,NBYNY
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      DO 601 I=1,NBX1
      OX=OX+HX
      F(I,NTY)=VALSP(1,1,NYM1,OX,OY)
  601 CONTINUE
  501 CONTINUE
      NXM1=NX-1
      DO 700 II=2,NXM1
                NTX=NTX+NBX(II-1)
      CALL SPLCOL(2,II,S,T,R0)
      ORX=OX
      NTY=0
      OY=ORY
```

```
      NBXII=NBX(II)
C     CAS L=1
      A1=A(II+1)-A(II)
      DO 798 J=1,NBY1
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      DO 799 I=1,NBXII
      K=NTX+I
      OX=OX+HX
C     ZR VARIABLE REDUITE
      ZR=(OX-A(II))/A1
      ALPHAR=ALPHA(ZR)
      BETAR=BETA(ZR)
      V1=VALSP(1,II-1,1,OX,OY)
      V2=VALSP(2,II,1,OX,OY)
      F(K,NTY)=ALPHAR*V1+BETAR*V2
  799 CONTINUE
  798 CONTINUE
      DO 800 L=2,NYM1
      B1=B(L+1)-B(L)
      NBYL=NBY(L)
      DO 900 J=1,NBYL
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      ZRR=(OY-B(L))/B1
      ALPHAY=ALPHA(ZRR)
      BETAY=BETA(ZRR)
      DO 1000 I=1,NBXII
      K=NTX+I
      OX=OX+HX
      ZR=(OX-A(II))/A1
      ALPHAR=ALPHA(ZR)
      BETAR=BETA(ZR)
      V1=VALSP(1,II-1,L-1,OX,OY)
      V2=VALSP(2,II,L-1,OX,OY)
      V3=VALSP(2,II,L,OX,OY)
      V4=VALSP(1,II-1,L,OX,OY)
      F(K,NTY)=ALPHAR*ALPHAY*V1+BETAR*ALPHAY*V2+BETAR*BETAY*V3+
     1ALPHAR*BETAY*V4
 1000 CONTINUE
  900 CONTINUE
  800 CONTINUE
C     CAS L=NY
      DO 901 J=1,NBYNY
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      DO 1001  I=1,NBXII
```

```
      K=NTX+I
      OX=OX+HX
C     VARIABLE REDUITE
      ZR=(OX-A(II))/A1
      ALPHAR=ALPHA(ZR)
      BETAR=BETA(ZR)
      V1=VALSP(1,II-1,NYM1,OX,OY)
      V2=VALSP(2,II,NYM1,OX,OY)
      F(K,NTY)=ALPHAR*V1+BETAR*V2
 1001 CONTINUE
  901 CONTINUE
      DO 1100 J=1,NYM1
      N4=N(II,J)+N(II+1,J)+N(II+1,J+1)+N(II,J+1)
      DO 1200 K=1,N4
 1200 S(1,J,K)=S(2,J,K)
      T(1,J,1)=T(2,J,1)
      T(1,J,2)=T(2,J,2)
      T(1,J,3)=T(2,J,3)
 1100   CONTINUE
  700 CONTINUE
        NTX=NTX+NBX(NXM1)
C     CALCUL POUR LA DERNIERE COLONNE
      ORX=OX
      NTY=0
      OY=ORY
C     CAS L=1
        NBXNX=NBX(NX)
      DO 748 J=1,NBY1
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      DO 759 I=1,NBXNX
      K=NTX+I
      OX=OX+HX
      F(K,NTY)=VALSP(1,NXM1,1,OX,OY)
  759 CONTINUE
  748 CONTINUE
C     L COMPRIS ENTRE 2 ET NY-1
      DO 750 L=2,NYM1
      B1=B(L+1)-B(L)
      NBYL=NBY(L)
      DO 760 J=1,NBYL
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      ZR=(OY-B(L))/B1
      ALPHAR=ALPHA(ZR)
      BETAR=BETA(ZR)
      DO 770 I=1,NBXNX
      K=NTX+I
```

```
      OX=OX+HX
      V2=VALSP(1,NXM1,L,OX,OY)
      V1=VALSP(1,NXM1,L-1,OX,OY)
      F(K,NTY)=ALPHAR*V1+BETAR*V2
770     CONTINUE
  760 CONTINUE
  750 CONTINUE
C     L=NY
      DO 761 J=1,NBYNY
      NTY=NTY+1
      OY=OY+HY
      OX=ORX
      DO 771 I=1,NBXNX
      K=NTX+I
      OX=OX+HX
      F(K,NTY)=VALSP(1,NXM1,NYM1,OX,OY)
  771 CONTINUE
  761 CONTINUE
      CALL CHRONO(IT1)
      WRITE(6,9876)IT1
 9876 FORMAT(' TEMPS DE CALCUL SPLINE ET GRILLE = ',I9)
      WRITE(6,3001)NPX,NPY,AA,BB,CC,DD
 3001 FORMAT(I5,' X ',I5,' POINTS DANS (',F8.2,',',F8.2,') X (',F8.2,','
     2,F8.2,')')
      WRITE(6,6789)(((I,J,F(I,J)),J=1,NPY),I=1,NPX)
 6789 FORMAT(7(2I3,1XF8.3,'*'))
      PUNCH 4000,NPX,NPY
      PUNCH 4001,AA,BB,CC,DD,HX,HY
      PUNCH 4002,((F(I,J),J=1,NPY),I=1,NPX)
 4000 FORMAT(2I4)
 4001 FORMAT(6F10.4)
 4002 FORMAT(10F8.3)
      STOP
      END


      DOUBLE PRECISION FUNCTION ALPHA(XX)
      REAL*8 ALPHA,X
      X=XX
      ALPHA=(1.-X)*(X+1.-2.*X*X)
      RETURN
      END


      DOUBLE PRECISION FUNCTION BETA(XX)
      REAL*8 BETA,X
      X=XX
      BETA=X*X*(3.-2.*X)
      RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION VALSP(IJ,I,J,OOX,OOY)
C     CALCUL DE LA VALEUR DE LA FONCTION SPLINE CALCULEEA L'AIDE DES
C     POINTS DES RECTANGLES R(I,J),R(I+1,J),R(I+1,J+1),R(I,J+1) ET
C     EVALUEE AU POINT OX,OY
      REAL*8 VALSP,S,T,OX,OY,D,EPS
      COMMON /CVALSP/S(2,9,140),T(2,9,3)
      COMMON /CSPL/X(10,10,35),Y(10,10,35),Z(10,10,35),N(10,10),NYM1
C     LE PARAMETRE IJ INDIQUE QUELS PARAMETRES DE SPLINE ON UTILISE
C     IJ=1 OU IJ=2
      EPS=0.1D-39
      OX=OOX
      OY=OOY
      N1=N(I,J)
      N2=N(I+1,J)
      N3=N(I+1,J+1)
      N4=N(I,J+1)
        I1=N1
      VALSP=OX*T(IJ,J,1)+OY*T(IJ,J,2)+T(IJ,J,3)
      IF (N1.EQ.0)GOTO 200
      DO 100 K=1,N1
      D=(OX-X(I,J,K))**2+(OY-Y(I,J,K))**2
        IF(D.LT.EPS)GOTO 100
      VALSP=VALSP+S(IJ,J,K)*D*DLOG(D)
100     CONTINUE
  200 IF (N2.EQ.0)GOTO 300
      DO 110 K=1,N2
      I1=I1+1
      D=(OX-X(I+1,J,K))**2+(OY-Y(I+1,J,K))**2
        IF(D.LT.EPS)GOTO 110
      VALSP=VALSP+S(IJ,J,I1)*D*DLOG(D)
110     CONTINUE
  300 IF (N3.EQ.0)GOTO 400
      DO 120 K=1,N3
      I1=I1+1
      D=(OX-X(I+1,J+1,K))**2+(OY-Y(I+1,J+1,K))**2
        IF(D.LT.EPS)GOTO 120
      VALSP=VALSP+S(IJ,J,I1)*D*DLOG(D)
120     CONTINUE
  400 IF (N4.EQ.0) RETURN
      DO 130 K=1,N4
      I1=I1+1
      D=(OX-X(I,J+1,K))**2+(OY-Y(I,J+1,K))**2
        IF(D.LT.EPS)GOTO 130
      VALSP=VALSP+S(IJ,J,I1)*D*DLOG(D)
130     CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION ALPHA(XX)
      REAL*8 ALPHA,X
      X=XX
      ALPHA=(1.-X)*(X+1.-2.*X*X)
      RETURN
```

```
      SUBROUTINE SPLCOL(IJ,I,S,T,R0)
C     SOUS PROGRAMME DE CALCUL DES COEFFICIENTS S ET T DE TOUTES LES
C     FONCTIONS SPLINE FAISANT INTERVENIR LES COLONNES I ET I+1
      REAL*8 XA,YA,ZA,SA,TA,A,R,S,T
      DIMENSION A(9870),R(9730)
      COMMON /CSPL/X(10,10,35),Y(10,10,35),Z(10,10,35),N(10,10),NYM1
      DIMENSION S(2,9,140),T(2,9,3)
C     TABLEAUX AUXILIAIRES POUR CONSTRUIRE LES FONCTIONS SPLINE
      DIMENSION XA(140),YA(140),ZA(140),SA(140),TA(3)
C     REMPLISSAGE DES TABLEAUX AUXILIAIRES
        I5=I+1
      DO 1 J=1,NYM1
        J5=J+1
      I1=N(I,J)
      IF(I1.EQ.0)GOTO 3
      DO 2 K=1,I1
      XA(K)=X(I,J,K)
      YA(K)=Y(I,J,K)
      ZA(K)=Z(I,J,K)
    2 CONTINUE
    3 I2=I1+N(I+1,J)
      IF(I2.EQ.I1)GOTO 4
      K1=I1+1
      DO 5 K=K1,I2
      K2=K-I1
      XA(K)=X(I5,J,K2)
      YA(K)=Y(I5,J,K2)
      ZA(K)=Z(I5,J,K2)
    5 CONTINUE
    4 I3=I2+N(I+1,J+1)
      IF(I3.EQ.I2)GOTO 6
      K1=I2+1
      DO 7 K=K1,I3
      K2=K-I2
      XA(K)=X(I5,J5,K2)
      YA(K)=Y(I5,J5,K2)
      ZA(K)=Z(I5,J5,K2)
    7 CONTINUE
    6 I4=I3+N(I,J+1)
      IF(I4.EQ.I3)GOTO 8
      K1=I3+1
      DO 9 K=K1,I4
      K2=K-I3
      XA(K)=X(I,J5,K2)
      YA(K)=Y(I,J5,K2)
      ZA(K)=Z(I,J5,K2)
    9 CONTINUE
C     CALCUL DE LA FONCTION SPLINE CALCULEE A L'AIDE DE  XA,YA,ZA
C     (SOIT AU TOTAL I4 POINTS)
    8 CALL SMETP(I4,R0,XA,YA,ZA,SA,TA,R,A)
      DO 10 K=1,I4
   10 S(IJ,J,K)=SA(K)
      DO 11 K=1,3
   11 T(IJ,J,K)=TA(K)
    1 CONTINUE
      RETURN
      END
```

REFERENCES

[1] ARTHUR D.W. .
"Multivariate spline functions. Construction, proprieties and computation"
Journal of Approximation theory 12, 396-411 (1974)

[2] ATTEIA M.
"Existence et détermination des fonctions spline à plusieurs variables"
C.R. Acad. Sc. Paris, t. 262 série A, p. 575-578, 1966

[3] BERIZIN I.S. and SHIDKOV N.P.
"Computing methods"
Pergamon Press Ltd. Vol. II 1965

[4] BOLONDI G. ROCCA F. and ZANOLETTI S.
"Automatic contouring of faulted subsurfaces"
Geophisics vol 41, n° 6, 1377-1393 december (1976)

[5] BRIGGS I.C.
"Machine contouring using minimum curvature"
Geophisics, vol 39 n° 1, 39-48 February (1974)

[6] CEA J.
"Optimisation théorie et algorithmes"
Dunod Paris 1971

[7] COLE A.J.
"Algorithm for the production of contour maps from scattered data"
Nature, vol. 220, 92-92 october 5 (1968)

[8] CRAIN I.K.
"Computer interpolation and contouring of two dimensional data : A review"
Geoexploration, 8, 71-86 (1970)

[9] DAX A. and KANILL S.
"Pivoting techniques for symetric gaussian elimination"
Numerische Mathematik, 28, 221-242, 1977

[10] DOOLEY J.C.
"Two-dimensional interpolation of irregularly spaced data using polynomial splines"
Physics of the earth and planetary interiors 12, 180-187 (1976)

[11] DUCHON J.
"Fonctions-spline du type plaque mince en dimension 2"
Séminaire d'Analyse Numérique n° 231 USMG Grenoble 9-10-1975

[12] DUCHON J.
"Fonctions-spline à énergie invariante par rotation"
Rapport de recherche n° 27 Mathématiques Appliquées, Grenoble 1976

[13] ESPINOZA C.
"Contribution à la résolution numérique de certains systèmes d'équations'
Thèse 3ème Cycle, USMG Grenoble 27 mai 1977

[14] FLETCHER R. and POWELL M.J.D.
"On the modification of LDL^T factorizations"
Mathematics of computation, Vol. 28, n° 128, 1067-1087 october 1974

[15] FLETCHER R.
"Factorizing symetric indefinite matrices"
Sincer algebra and its applications, 14, 257-272 (1976)

[16] FRANKE R.
"Locally determined smooth interpolation at irregularity spaced points in several variables"
J. Inst. Maths. Applies 19, 471-482 (1977)

[17] GASTINEL N.
"Analyse numérique linéaire"
Hermann Paris (1966)

[18] GILL P.E., MORRAY W. and SAUNDERS M.A.
"Methods for computing and modifying the LDV factors of a matrix"
Mathematics of Computation, Vol. 19, n° 132, 1051-1077, october 1975

[19] HALL C.A.
"Natural cubic and bicubic spline interpolation"
SIAM J. Num. Anal. vol. 10, n° 6, december 1973

[20] HESSING R.C., LEE H.K., PERCE A. and POWERS E.N.
"Automatic contouring using bicubic functions"
Geophysics, vol. 37, n° 4, 669-674 August 1972

[21] HOUSEHOLDER A.S.
"The theory of matrices in Numerical Analysis"
Blaisdell Publishing Company 1965

[22] JENNING A. and MALIK G.M.
"Partial elimination"
J. Inst. Maths. Applies. 20, 307-316 (1977)

[23] KOWALIK J. and OSBORNE M.R.
"Inconstrained optimization problems"
American Elsevier Publisching company Inc. N.Y. 1968

[24] LA PORTE M. and VIGNES J.
"Etude statistique des erreurs dans l'arithmétique des ordinateurs ; application au contrôle des résultats d'algorithmes numériques"
Numerische Mathematik, 23, 63, 82 (1974)

[25] MARKOW S. and POWELL M.J.D.
"A fortran subroutine for plating the part of a conic, that is inside a given triangl"
AERE-R8336 Harwell, oxfordshire, England, april 1976

[26] Mc LAIN D.H.
"Drawing contours from arbitrary data points"
The computer Journal, vol. 17, n° 4, 318-324.

[27] Mc. LAIN D.H.
"Two dimensional interpolation from vaudom data"
The computer Journal, Vol. 19, n° 2, 178-181.

[28] ORTEGA J. and RHEINBOLD W.
"Iteration solution of non linear equations un several variables"
Academic Press, N.Y. 1970

[29] PAIGE L.C.
"An error analysis of a method for solving matrix equations"
Mathematics of Computation, vol 27, n° 122, april 1973

[30] PAIHUA L.
"Méthodes numériques pour l'obtention des fonctions-spline du type plaque mince en dimension 2".
Séminaire Anal. Num. n° 273 USMG Grenoble 24-3-77

[31] POWELL M.J.D.
"Piecewise quadratique surface fitting for contour plotting" in "software for numerical mathematics"
D.J. Evans, Academic Press (London) 1974

[32] ROBERT F.
"Analyse Numérique Iterative"
Cours INPG-USMG 1975-1976

[33] ROSEN J.B.
"The gradient projection method for non linear programming"
J. SIAM vol 8 n° 1 march (1960)

[34] RAO C.R. and MITRA S.K
"Generalized inverse of matrices and its applications"

[35] SAUNDERS M.A.
"Large-scale linear programming using the Choleski factorization"
Computer science department SU 326, p. 30-14 stanford University

[36] SHULTZ M.H.
"Multivariate L-spline interpolation"
Journal of approximation theory 2, 127-135 (1969)

[37] STEWART G.W.
"Perturbation bounds for the QR factorization of a matrix"
SIAM J. Numr. Anal. vol 14, n° 3, June 1977

[38]
"An algorithm for arawing the courbe f(x,y) = O"
The computer journal vol. 19, n° 3

[39] SZEGO G.P.
"Minimization algorithms"
Academic Press .N.Y. London 1972

[40] THOMANN J.
"Détermination et construction de fonctions spline à deux variables définies sur un domaine rectangulaire ou circulaire"
Thèse 3ème Cycle, Lille 5-2-1970

[41] WILKINSON J.H.
"The algebraic lingenvalue problem"
Clarendon Press Oxford 1965

[42] WILKINSON J.H. and REINSH C.
"Linear algebra"
Springer-Verlag Berlin Heidelberg N.Y. 1971

[43] DUCHON J.
"Fonctions-spline et esperances conditionnelles de champs gaussiens"
Annales Scientifiques de l'Université de Clermont. n° 61 14ème fas., Clermont-Ferrand 1976

[44] DUCHON J.
"Splines minimizing rotation-invariant semi-norms in soldev spaces"
Lecture notes in mathematics, 571 constructive theory of functions of several variables, oberwolfach 1976, p. 85-100
Springer-Verlag 1977

[45] ANSELONE P.M. and LAURENT J.P.
"A general method for the construction of interpolating or smoothing spline-functions"
Numerische Mathematik 12, 66-82 (1968)

[46] LAURENT P.J.
"Approximation et Optimisation"
Hermann Paris 1972

[47] ROCKAFELLAR R.
"Convex analysis"
Princeton 1972

Dernière page d'une thèse

VU

Grenoble, le 20 Avril 1978

p. Le Président de la thèse empêché

F. ROBERT

F. ROBERT
Responsable 3e cycle
analyse numérique

Vu, et permis d'imprimer,

Grenoble, le 21 Avril 1978

Le Président de l'Université
Scientifique et Médicale

Dr G. CALI