



HAL
open science

Amélioration de maillages par des méthodes de sous-gradient

Charles Cua

► **To cite this version:**

Charles Cua. Amélioration de maillages par des méthodes de sous-gradient. Modélisation et simulation. Université Joseph-Fourier - Grenoble I; Institut National Polytechnique de Grenoble - INPG, 1985. Français. NNT: . tel-00318480

HAL Id: tel-00318480

<https://theses.hal.science/tel-00318480>

Submitted on 4 Sep 2008

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

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

THESE

présentée à

l'Université Scientifique et Médicale de Grenoble

et à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR de 3ème Cycle

Mathématiques appliquées

option recherche opérationnelle

par

Charles Cua

Amélioration de Maillages

par

des Méthodes de Sous-Gradient

Thèse soutenue le 23 Octobre 1985 devant la Commission d'Examen

J. FONLUPT Président
J.L. COULOMB Examineurs
P. GENOUX
P.J. LAURENT
J. PERIAUX
P. PERRIER

THESE

présentée à

l'Université Scientifique et Médicale de Grenoble

et à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR de 3ème Cycle

Mathématiques appliquées

option recherche opérationnelle

par

Charles Cua

Amélioration de Maillages

par

des Méthodes de Sous-Gradient

Thèse soutenue le 23 Octobre 1985 devant la Commission d'Examen

J. FONLUPT	Président
J.L. COULOMB	Examineurs
P. GENOUX	
P.J. LAURENT	
J. PERIAUX	
P. PERRIER	

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : M. TANCHE

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

(RANG A)

SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE

PROFESSEURS DE 1ère CLASSE

ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

.../...

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDO Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie

PROFESSEURS DE 2^{ème} CLASSE

ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

.../...

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : D. BLOCH

Vice-Président : René CARRE

Hervé CHERADAME

Marcel IVANES

PROFESSEURS DES UNIVERSITES :

ANCEAU François	E.N.S.I.M.A.G.
BARRAUD Alain	E.N.S.I.E.G.
BAUDELET Bernard	E.N.S.I.E.G.
BESSON Jean	E.N.S.E.E.G.
BLIMAN Samuel	E.N.S.E.R.G.
BLOCH Daniel	E.N.S.I.E.G.
BOIS Philippe	E.N.S.H.G.
BONNETAIN Lucien	E.N.S.E.E.G.
BONNIER Etienne	E.N.S.E.E.G.
BOUVARD Maurice	E.N.S.H.G.
BRISSONNEAU Pierre	E.N.S.I.E.G.
BUYLE BODIN Maurice	E.N.S.E.R.G.
CAVAIGNAC Jean-François	E.N.S.I.E.G.
CHARTIER Germain	E.N.S.I.E.G.
CHENEVIER Pierre	E.N.S.E.R.G.
CHERADAME Hervé	U.E.R.M.C.P.P.
CHERUY Arlette	E.N.S.I.E.G.
CHIAVERINA Jean	U.E.R.M.C.P.P.
COHEN Joseph	E.N.S.E.R.G.
COUMES André	E.N.S.E.R.G.
DURAND Francis	E.N.S.E.E.G.
DURAND Jean-Louis	E.N.S.I.E.G.
FELICI Noël	E.N.S.I.E.G.
FOULARD Claude	E.N.S.I.E.G.
GENTIL Pierre	E.N.S.E.R.G.
GUERIN Bernard	E.N.S.E.R.G.
GUYOT Pierre	E.N.S.E.E.G.
IVANES Marcel	E.N.S.I.E.G.
JAUSSAUD Pierre	E.N.S.I.E.G.
JOUBERT Jean-Claude	E.N.S.I.E.G.
JOURDAIN Geneviève	E.N.S.I.E.G.
LACOUME Jean-Louis	E.N.S.I.E.G.
LATOMBE Jean-Claude	E.N.S.I.M.A.G.

LESSIEUR Marcel	E.N.S.H.G.
LESPINARD Georges	E.N.S.H.G.
LONGEQUEUE Jean-Pierre	E.N.S.I.E.G.
MAZARE Guy	E.N.S.I.M.A.G.
MOREAU René	E.N.S.H.G.
MORET Roger	E.N.S.I.E.G.
MOSSIERE Jacques	E.N.S.I.M.A.G.
PARIAUD Jean-Charles	E.N.S.E.E.G.
PAUTHENET René	E.N.S.I.E.G.
PERRET René	E.N.S.I.E.G.
PERRET Robert	E.N.S.I.E.G.
PIAU Jean-Michel	E.N.S.H.G.
POLOJADOFF Michel	E.N.S.I.E.G.
POUPOT Christian	E.N.S.E.R.G.
RAMEAU Jean-Jacques	E.N.S.E.E.G.
RENAUD Maurice	U.E.R.M.C.P.P.
ROBERT André	U.E.R.M.C.P.P.
ROBERT François	E.N.S.I.M.A.G.
SABONNADIERE Jean-Claude	E.N.S.I.E.G.
SAUCIER Gabrielle	E.N.S.I.M.A.G.
SCHLENKER Claire	E.N.S.I.E.G.
SCHLENKER Michel	E.N.S.I.E.G.
SERMET Pierre	E.N.S.E.R.G.
SILVY Jacques	U.E.R.M.C.P.P.
SOHM Jean-Claude	E.N.S.E.E.G.
SOUQUET Jean-Louis	E.N.S.E.E.G.
VEILLON Gérard	E.N.S.I.M.A.G.
ZADWORNY François	E.N.S.E.R.G.

PROFESSEURS ASSOCIES

BASTIN Georges	E.N.S.H.G.
BERRIL John	E.N.S.H.G.
CARREAU Pierre	E.N.S.H.G.
GANDINI Alessandro	U.E.R.M.C.P.P.
HAYASHI Hirashi	E.N.S.I.E.G.

PROFESSEURS UNIVERSITE DES SCIENCES SOCIALES (Grenoble II)

BOLLIET Louis
Chatelin Françoise

PROFESSEURS E.N.S. Mines de Saint-Etienne

RIEU Jean
SOUSTELLE Michel

CHERCHEURS DU C.N.R.S.

FRUCHART Robert
VACHAUD Georges

Directeur de Recherche
Directeur de Recherche

.../...

ALLIBERT Michel	Maître de Recherche
ANSARA Ibrahim	Maître de Recherche
ARMAND Michel	Maître de Recherche
BINDER Gilbert	
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DEPORTES Jacques	
DRIOLE Jean	Maître de Recherche
GIGNOUX Damien	
GIVORD Dominique	
GUELIN Pierre	
HOPFINGER Emil	Maître de Recherche
JOUD Jean-Charles	Maître de Recherche
KAMARINOS Georges	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Dore	Maître de Recherche
LASJAUNIAS J.C.	
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche
PIAU Monique	
PORTESEIL Jean-Louis	
THOLENCE Jean-Louis	
VERDILLON André	

CHERCHEURS du MINISTERE de la RECHERCHE et de la TECHNOLOGIE (Directeurs et Maîtres de Recherches, ENS Mines de St. Etienne)

LESBATS Pierre	Directeur de Recherche
BISCONDI Michel	Maître de Recherche
KOBYLANSKI André	Maître de Recherche
LE COZE Jean	Maître de Recherche
LALAUZE René	Maître de Recherche
LANCELOT Francis	Maître de Recherche
THEVENOT François	Maître de Recherche
TRAN MINH Canh	Maître de Recherche

PERSONNALITES HABILITEES à DIRIGER des TRAVAUX de RECHERCHE (Décision du Conseil Scientifique)

ALLIBERT Colette	E.N.S.E.E.G.
BERNARD Claude	E.N.S.E.E.G.
BONNET Rolland	E.N.S.E.E.G.
CAILLET Marcel	E.N.S.E.E.G.
CHATILLON Catherine	E.N.S.E.E.G.
CHATILLON Christian	E.N.S.E.E.G.
COULON Michel	E.N.S.E.E.G.
DIARD Jean-Paul	E.N.S.E.E.G.
EUSTAPOPOULOS Nicolas	E.N.S.E.E.G.
FOSTER Panayotis	E.N.S.E.E.G.

.../...

GALERIE Alain	E.N.S.E.E.G.
HAMMOU Abdelkader	E.N.S.E.E.G.
MALMEJAC Yves	E.N.S.E.E.G. (CENG)
MARTIN GARIN Régina	E.N.S.E.E.G.
NGUYEN TRUONG Bernadette	E.N.S.E.E.G.
RAVAINE Denis	E.N.S.E.E.G.
SAINFORT	E.N.S.E.E.G. (CENG)
SARRAZIN Pierre	E.N.S.E.E.G.
SIMON Jean-Paul	E.N.S.E.E.G.
TOUZAIN Philippe	E.N.S.E.E.G.
URBAIN Georges	E.N.S.E.E.G. (Laboratoire des ultra-réfractaires ODEILLON)
GUILHOT Bernard	E.N.S. Mines Saint Etienne
THOMAS Gérard	E.N.S. Mines Saint Etienne
DRIVER Julien	E.N.S. Mines Saint Etienne
BARIBAUD Michel	E.N.S.E.R.G.
BOREL Joseph	E.N.S.E.R.G.
CHOVET Alain	E.N.S.E.R.G.
CHEHIKIAN Alain	E.N.S.E.R.G.
DOLMAZON Jean-Marc	E.N.S.E.R.G.
HERAULT Jeanny	E.N.S.E.R.G.
MONLLOR Christian	E.N.S.E.R.G.
BORNARD Guy	E.N.S.I.E.G.
DESCHIZEAU Pierre	E.N.S.I.E.G.
GLANGEAUD François	E.N.S.I.E.G.
KOFMAN Walter	E.N.S.I.E.G.
LEJEUNE Gérard	E.N.S.I.E.G.
MAZUER Jean	E.N.S.I.E.G.
PERARD Jacques	E.N.S.I.E.G.
REINISCH Raymond	E.N.S.I.E.G.
ALEMANY Antoine	E.N.S.H.G.
BOIS Daniel	E.N.S.H.G.
DARVE Félix	E.N.S.H.G.
MICHEL Jean-Marie	E.N.S.H.G.
OBLÉD Charles	E.N.S.H.G.
ROWE Alain	E.N.S.H.G.
VAUCLIN Michel	E.N.S.H.G.
WACK Bernard	E.N.S.H.G.
BERT Didier	E.N.S.I.M.A.G.
CALMET Jacques	E.N.S.I.M.A.G.
COURTIN Jacques	E.N.S.I.M.A.G.
COURTOIS Bernard	E.N.S.I.M.A.G.
DELLA DORA Jean	E.N.S.I.M.A.G.
FONLUPT Jean	E.N.S.I.M.A.G.
SIFAKIS Joseph	E.N.S.I.M.A.G.
CHARUEL Robert	U.E.R.M.C.P.P.
CADET Jean	C.E.N.G.
COEURE Philippe	C.E.N.G. (LETI)

.../...

DELHAYE Jean-Marc	C.E.N.G. (STT)
DUPUY Michel	C.E.N.G. (LETI)
JOUBE Hubert	C.E.N.G. (LETI)
NICOLAU Yvan	C.E.N.G. (LETI)
NIFENECKER Hervé	C.E.N.G.
PERROUD Paul	C.E.N.G.
PEUZIN Jean-Claude	C.E.N.G. (LETI)
TAIEB Maurice	C.E.N.G.
VINCENDON Marc	C.E.N.G.

LABORATOIRES EXTERIEURS

DEMOULIN Eric	C.N.E.T.
DEVINE	C.N.E.T. (R.A.B.)
GERBER Roland	C.N.E.T.
MERCKEL Gérard	C.N.E.T.
PAULEAU Yves	C.N.E.T.
GAUBERT C.	I.N.S.A. Lyon

J'exprime toute ma reconnaissance à Monsieur le Professeur J. FONLUPT qui a dirigé mes recherches pour cette thèse. Son attention, ses conseils et sa patience ont contribué énormément à l'accomplissement de ce travail. Je le remercie aussi pour avoir accepté de présider le jury de cette thèse.

Que Messieurs P. PERRIER et J. PERIAUX soient vivement remerciés pour l'intérêt qu'ils ont apporté à ce travail, pour leurs conseils et leur accueil chaleureux durant les stages que j'ai effectués à AVIONS MARCEL DASSAULT/ BREGUET AVIATION. Ces stages m'ont permis de me familiariser avec le problème et certains des logiciels le concernant. Monsieur PERRIER est chef de département de l'aérodynamique théorique; Monsieur PERIAUX est chef de service, analyse numérique.

Je remercie aussi Messieurs P.J. LAURENT, professeur à l'USMG, J.L. COULOMB, maître de conférence à l'ENSIEG, et P. GENOUX, ingénieur principal de l'armement, pour avoir accepté de faire partie de ce jury.

Tout mes remerciements également aux membres de l'équipe de Recherche Operationnelle et de l'ARTEMIS et, de manière particulière, à Y. PAILLET, D. CHEVROLET, A. ZEMERLINE, M. BURLET, J.C. LACOTE, G. KUNTZ, A. ELNACHEF et à l'équipe DGT/DEA d'AVIONS MARCEL DASSAULT/ BREGUET AVIATION notamment C. POULETTY, pour les divers conseils et qu'ils m'ont donné.

Que soient remerciés Mr. D. IGLESIAS et le service de reprographie de l'IMAG qui ont assuré la réalisation matérielle de ce travail.

Que trouvent ici l'expression de ma profonde gratitude tous ceux qui m'ont apporté un soutien moral indispensable pour faire mes études en France, notamment Mr. G. DE NALE, Mr. D. PHILIPPE, M. et Mme. P. FOUILLET, Mme. G. VELLOTT, et mes chers amis du groupe DJ Grenoble.

TABLE DES MATIERES

1. INTRODUCTION	1
2. RAPPELS	4
2.1 Rappels sur la Nature des Maillages Eléments Finis	4
2.2 Rappels sur les Méthodes des Maillages Eléments Finis	4
2.2.1 Méthodes de Génération de Maillage	4
2.2.2 Méthodes de l'Amélioration de Maillage	12
2.3 Rappels sur les Méthodes de Gradient et de Sous-Gradient en Optimisation	14
3. SPECIFICATION DU PROBLEME ET DE LA METHODE DE RESOLUTION	15
3.1 Présentation Sommaire du Problème	15
3.2 Etude des Contraintes du Problème	15
3.3 Etude des Critères d'Amélioration	16
3.4 Formulation Mathématique du Problème	17
3.5 Représentation de $V_j(P_0)$, le Volume du $j^{\text{ème}}$ Tétraèdre Entourant P_0	18
3.6 Présentation Sommaire de la Méthode de Sous-Gradient Utilisée en tant que Méthode de Résolution du Problème	19
4. PRESENTATION DETAILLEE DES ALGORITHMES - ALGO1 ET ALGOTOUT	20
4.1 ALGO1 - Algorithme pour Bouger un Seul Noeud P_0	20
4.1.1 Le Choix de la Direction de Descente en tant que Direction d'Amélioration	20
4.1.2 Le Choix du Pas à Prendre dans la Direction d'Amélioration	21
4.1.3 Résumé de la Méthode de Résolution	24
4.1.4 Un Exemple de l'Exécution de l'ALGO1	25
4.1.5 Trois Exemples de l'Exécution de l'ALGO1 en Faisant Varier la Fonction Objective	28

4.2	ALGOTOUT - Algorithme pour Bouger un Ensemble de Noeuds en les Bougeant Un par Un	29
4.2.1	Le Choix de la Fonction Objective à Minimiser	29
4.2.2	Le Choix du Nombre d'Itérations Successives pour Bouger un Noeud et du Nombre de Balayages pour Bouger un Ensemble de Noeuds	30
4.2.3	Le Choix des Noeuds à Bouger, et de l'Ordre dans Lequel Ils Seront Bougés	39
5.	RESULTATS NUMERIQUES	43
5.1	Résultats sur Quatre Cas d'Ecole et Trois Cas Réels	43
5.1.1	Quelques Résultats en Dimensions 2 et 3 sur l'Ordinateur HB-68 (Système d'Exploitation MULTICS)	43
5.1.2	Courbes de Convergences de POURAME, le Pourcentage d'Amélioration Par Rapport au Nombre de Balayages de l'Ensemble de Noeuds	64
5.2	Autres Résultats	68
6.	CONCLUSIONS	83
	ANNEXES INFORMATIQUES	85
	Annexe 1 : Description des Variables et Programmes Principaux Utilisés dans les Programmes Informatiques de l'ALGO1 et de l'ALGOTOUT en dimension 3 (en dimension 2)	85
	Annexe 2.1 : Programmes de l'ALGO1 en dimension 2	87
	Annexe 2.2 : Programmes de l'ALGOTOUT en dimension 2	95
	Annexe 3 : Programmes de l'ALGOTOUT en dimension 3	100
	Annexe 4.1 : La Déroulement de l'ALGO1 en Bougeant Une Fois le Noeud 6	111
	Annexe 4.2 : Bouger le Noeud 10 Une Fois par $C_1(P_0)$	112
	Annexe 4.3 : Bouger le Noeud 10 Une Fois par $C_2(P_0)$	114

Annexe 5.1 : Bouger le Noeud 6 en 5 Itérations	118
Annexe 5.2 : Bouger le Noeud 10 en 5 Itérations	120
Annexe 5.3 : Bouger le Noeud 10 en 5 Itérations par $C_3(P_0)$ avec une Itération Non-Améliorante	122
Annexe 6 : Bouger 12 Noeuds avec 3 Itérations par Noeud et 3 Tours pour l'Ensemble de Noeuds	123
Annexe 7 : Maillage à Trois Etapes	124

BIBLIOGRAPHIE	127
----------------------	------------

1. INTRODUCTION

Un grand nombre de phénomènes physiques en mécanique, en électrodynamique, en thermodynamique, par exemple, peuvent s'étudier par la résolution d'équations aux dérivées partielles. Cette résolution se fait, en général, par des méthodes d'éléments finis, c'est à dire à partir d'un découpage de l'espace étudié en tétraèdres en dimension 3 ou triangles en dimension 2. Ainsi, dans l'aéronautique, le problème de l'écoulement d'air autour des parois d'un avion peut-il s'aborder de cette manière. L'obtention d'un bon découpage de l'espace autour d'une forme d'avion que nous appellerons également maillage de l'espace est un problème difficile pour lequel aucune méthodologie systématique n'existe réellement, même si en pratique un grand nombre de méthodes plus ou moins adaptées au type de problème étudié permet de répondre à ces questions mais d'une manière qui n'est pas toujours satisfaisante.

C'est à partir d'une question soulevée par la DGT/DEA, Avions Marcel Dassault/ Breguet Aviation que nous avons abordé cette recherche. Résumons les problèmes soulevés par le maillage de l'espace autour d'une forme d'avion.

On peut distinguer deux grandes questions à poser sur les maillages :

- 1) la création d'un maillage
- 2) l'amélioration d'un maillage.

Le problème de la création d'un maillage initial peut être excessivement complexe surtout quand les formes autour desquelles s'effectue ce maillage sont complexes, ce qui est le cas en aéronautique. Nous n'avons pas abordé ici ce problème.

Un bon maillage doit satisfaire certaines conditions liées au problème étudié, par exemple,

- des conditions géométriques du genre: les tétraèdres ne doivent pas être trop aplatis, deux tétraèdres voisins doivent avoir des volumes aussi proches que possible,
- des conditions liées au problème physique étudié: dans le cas d'avions, des phénomènes physiques se produiront à des endroits bien connus (à l'avant du fuselage, au bord des ailes), et dans ces endroits il faut une forte densité de points, tandis que dans d'autres endroits loin de la surface de l'avion, on peut avoir une plus faible densité de points.

Quand on s'intéresse au problème d'amélioration d'un maillage, on peut se poser deux types de questions :

- 1) D'abord, comment améliorer le maillage par changement de topologie actuelle. Cette question facile dans le cadre de deux dimensions est plus délicate en trois dimensions, car il n'est pas facile de passer d'une structure de tétraèdres à une structure voisine.

2) Sans changer la structure actuelle de tétraèdres, comment bouger les points afin d'obtenir une meilleure tétraèdrisation à partir d'un critère géométrique ou physique qu'on se fixe.

C'est ce dernier problème qui est abordé dans cette thèse. Il s'agit donc d'une étape dans le cadre du problème général d'obtention d'un maillage. Dans la littérature, on peut trouver des méthodes spécifiques adaptées à tel ou tel type de problème, mais qui ne peuvent se généraliser.

Notre travail propose une méthode générale et algorithmique fondée sur des méthodes d'optimisation classique en recherche opérationnelle qui permet d'englober un grand nombre des méthodes actuelles et d'en proposer de nouvelles. Nos méthodes ont été testées sur des exemples d'école qui ont permis de bien ajuster les algorithmes choisis. Elles ont aussi été testées sur des cas réels, en collaboration avec la DGT/DEA, Avions Marcel Dassault/ Breguet Aviation.

Notre effort a porté sur la réalisation d'algorithmes efficaces, c'est à dire que nous avons essayé de trouver un compromis entre l'utilisation des méthodes puissantes de l'analyse numérique qui souvent ne sont pas applicables car trop lourdes et le désir d'obtenir un bon maillage à partir d'un maillage donné.

Les algorithmes proposés sont connus sous le nom de méthodes de sous-gradient (cf l'article de Held et Crowder [22]). Leur avantage est d'être rapide au moins dans les premières itérations, c'est à dire de donner rapidement une bonne solution, contrairement aux méthodes classiques, mais elles ne permettent pas d'obtenir une solution optimale pour un problème de minimisation donné. Ceci semble bien adapté dans notre problème car les critères d'optimisation choisis sont toujours plus ou moins subjectifs et chercher la meilleure solution suivant tel ou tel critère n'a pas de véritable intérêt.

On trouve dans le deuxième chapitre de brefs rappels sur la nature des maillages de type éléments finis, sur les méthodes connues de génération et d'amélioration des maillages, et sur la nature du sous-gradient en tant qu'outil pour la résolution des problèmes d'optimisation.

Le chapitre 3 présente le problème en définissant ses contraintes et les objectifs à atteindre. Il propose ensuite des méthodes de résolution connues sous le nom de méthodes de sous-gradient en décrivant les étapes principales à effectuer en utilisant ces méthodes pour résoudre le problème.

Le chapitre 4 décrit en détail les méthodes de résolution proposées au chapitre 3. Ce chapitre peut se diviser en deux parties :

1) On cherche à fixer tous les noeuds du maillage sauf un, et on crée un algorithme qui consiste à replacer ce noeud et à améliorer le critère d'amélioration choisi. Nous appellerons cet algorithme ALGO1.

2) On applique ALGO1 itérativement en balayant tous les noeuds internes, et ce balayage est refait un certain nombre de fois. Nous appellerons ALGOTOUT l'algorithme qui effectue cette application itérative de l'ALGO1 .

On trouve aussi dans le chapitre 4 des applications des algorithmes ALGO1 et ALGOTOUT sur des cas d'école pour montrer le comportement de ces deux algorithmes.

Le chapitre 5 présente l'application de l'ALGOTOUT sur quatre cas d'école et trois cas réels proposés par la DGT/DEA, Avions Marcel Dassault/ Breguet Aviation. On calcule le temps moyen pour bouger une seule fois un noeud dans chacun de ces sept cas. On mesure le degré d'amélioration du maillage résultant par rapport au maillage initial. On trouve aussi d'autres applications de l'ALGO1 par DGT/DEA, Avions Marcel Dassault/ Breguet Aviation en conjonction avec certains de leurs algorithmes.

Dans le chapitre 6 sont données les conclusions de cette étude.

Enfin, en annexes, sont donnés les listings des programmes informatiques de l'ALGO1 et l'ALGOTOUT, précédés par une description préliminaire des variables et des sous-programmes principaux présents dans les programmes informatiques. On trouve aussi une présentation informatique détaillée du déroulement de l'exécution des deux algorithmes sur des cas d'école du chapitre 4.

2. RAPPELS

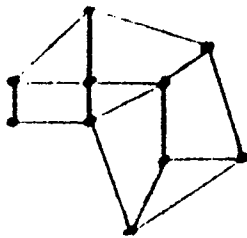
2.1 Rappels sur la Nature des Maillages Eléments Finis

Un maillage de type éléments finis peut se définir de la manière suivante :

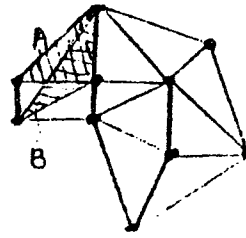
La zone étudiée est découpée uniquement en tétraèdres; chaque tétraèdre est défini par quatre sommets; l'intersection de deux tétraèdres est limitée à un seul sommet, ou à une seule arête (une arête est définie par deux sommets), ou à la seule face commune possible aux deux tétraèdres (une face est définie par trois sommets), ou est vide; chaque face de tétraèdre appartient à deux tétraèdres, à l'exception des faces sur la frontière de la zone étudiée qui appartient à un seul tétraèdre. En dimension 2, la zone est découpée en triangles.

Voici trois exemples en dimension 2 d'une décomposition de l'espace qui n'est pas un maillage de type éléments finis.

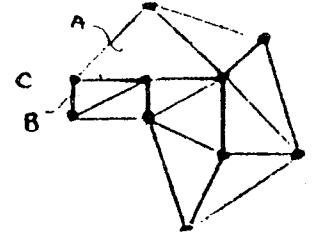
Figure 2.1 (maillage dimension 2)



une décomposition de l'espace mais pas uniquement en triangles



l'intersection des triangles A et B n'est ni un seul sommet ni une seule arête



l'arête C du triangle B n'est pas une arête du triangle A

2.2 Rappels sur les Méthodes des Maillages Eléments Finis

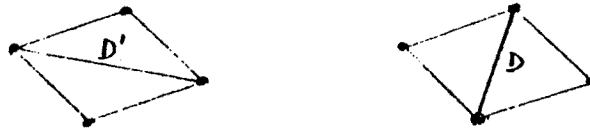
2.2.1 Méthodes de Génération de Maillage

Diverses méthodes existent dans le domaine de génération des maillages éléments finis, c'est à dire, la tétraèdrisation initiale de l'espace à mailler. Un panorama de ces méthodes se trouve dans l'article [18] de W.C.Thacker. Voici une description sommaire de certaines de ces méthodes :

- (1) Parmi les méthodes qui se basent sur la construction un par un des tétraèdres, l'article [13] de Nguyen-van-Phai et l'article [12] de Mclain utilisent des critères géométriques pour le choix du noeud D parmi les noeuds définis sur l'espace à mailler afin de construire le tétraèdre $ABCD$ à partir du triangle ABC qui fait partie du maillage actuel. Ces méthodes sont limitées aux espaces convexes ou presque convexes, et, même dans ces cas, la possibilité d'avoir un maillage non réalisable existe.

En partant d'un maillage qui contient des quadrilatères non maillés (en dimension 2), l'article [10] de Kleinstreuer et Holdeman donne le choix de la diagonale du quadrilatère qui découpe le quadrilatère en deux triangles les plus équilatéraux possibles, comme c'est illustré par la figure 2.2. Il utilise la mesure d'angles de ces triangles pour faire son choix.

Figure 2.2 (en dimension 2)



On préfère la diagonale D à la diagonale D' .

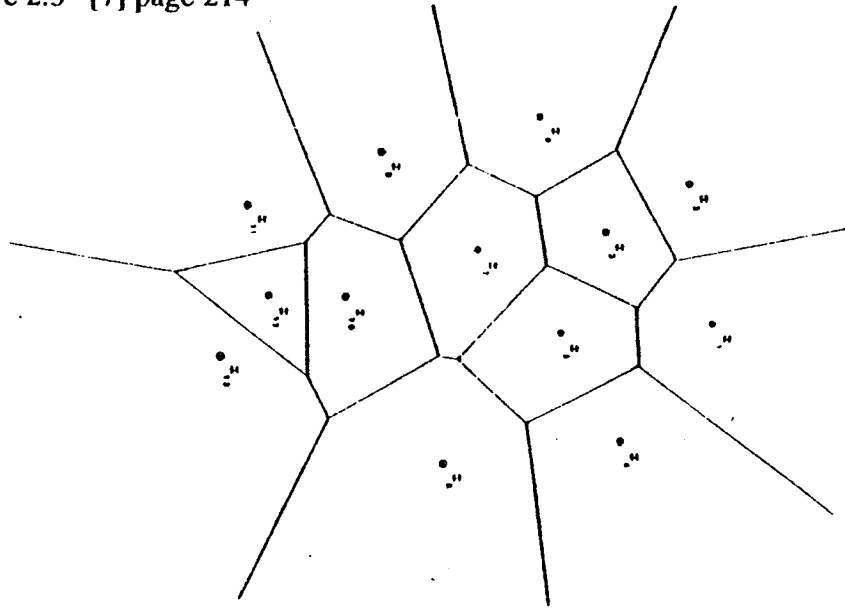
- (2) On trouve des méthodes de découpage de l'espace en polyèdres convexes ou presque convexes dans les articles de Bykat [2] et de Kamel et Eisenstein [9] (en dimension 2) et dans l'article de Zienkiewicz et Phillips [21] (en dimension 3). Un article qui présente plus d'intérêt que les autres car il utilise comme seul outil de génération de maillage des découpages successifs de l'espace en polyèdres est l'article [11] de Lewis et Robinson. Il donne un algorithme récursif sans se soucier de la convexité des régions intermédiaires. Dans le cas de dimension 2, cet algorithme choisit une droite, qui sépare la région en 2, et 2 seulement, parties de surfaces égales. La frontière qui va définitivement découper la région en deux est formée par des noeuds choisis à proximité de cette droite, de plus, deux noeuds consécutifs de cette frontière ne sont pas dans le même demi-plan défini par cette droite. L'algorithme est appliqué récursivement sur les nouvelles régions. Preparata et Hong dans l'article [15] utilise un algorithme semblable à celui de Lewis et Robinson.
- (3) Hermeline dans l'article [7] donne un algorithme (en dimension 2) qui permet de construire une triangulation de Delaunay qui, en fait, est le dual des polygones de Voronoi.

Voici la définition d'un polygone de Voronoi :

Soient n points: x_1, \dots, x_n de l'espace \mathbb{R}^2 . Le polygone de Voronoi associé au point x_i est l'ensemble des points $x \in \mathbb{R}^2$ tel que $d(x, x_i) \leq d(x, x_j)$ pour tout $x_j, j=1, \dots, n; j \neq i$ ($d(x, y)$ désigne la distance euclidienne de x à y).

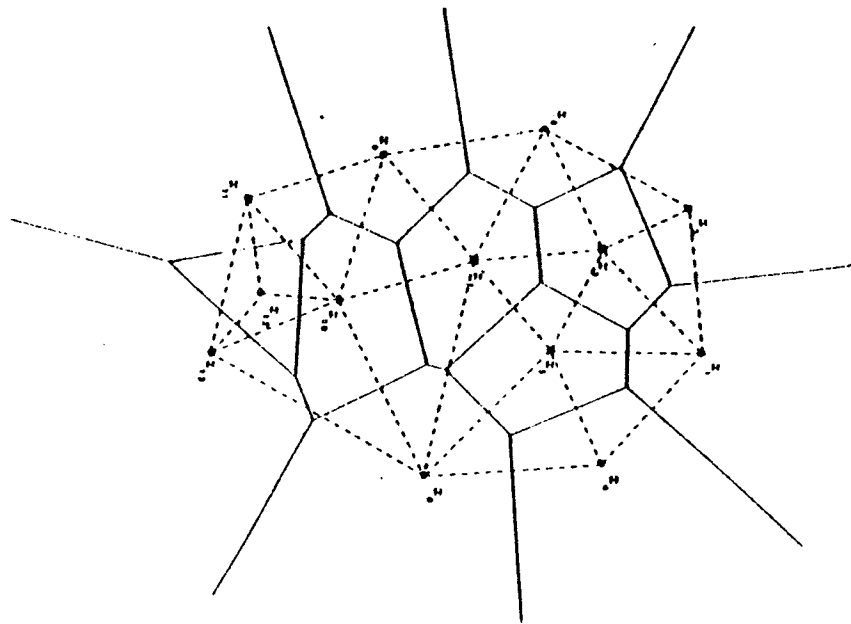
Voici un exemple d'un polygone de Voronoi associé à l'ensemble $E=\{x_1, \dots, x_{13}\}$.

Figure 2.3 [7] page 214



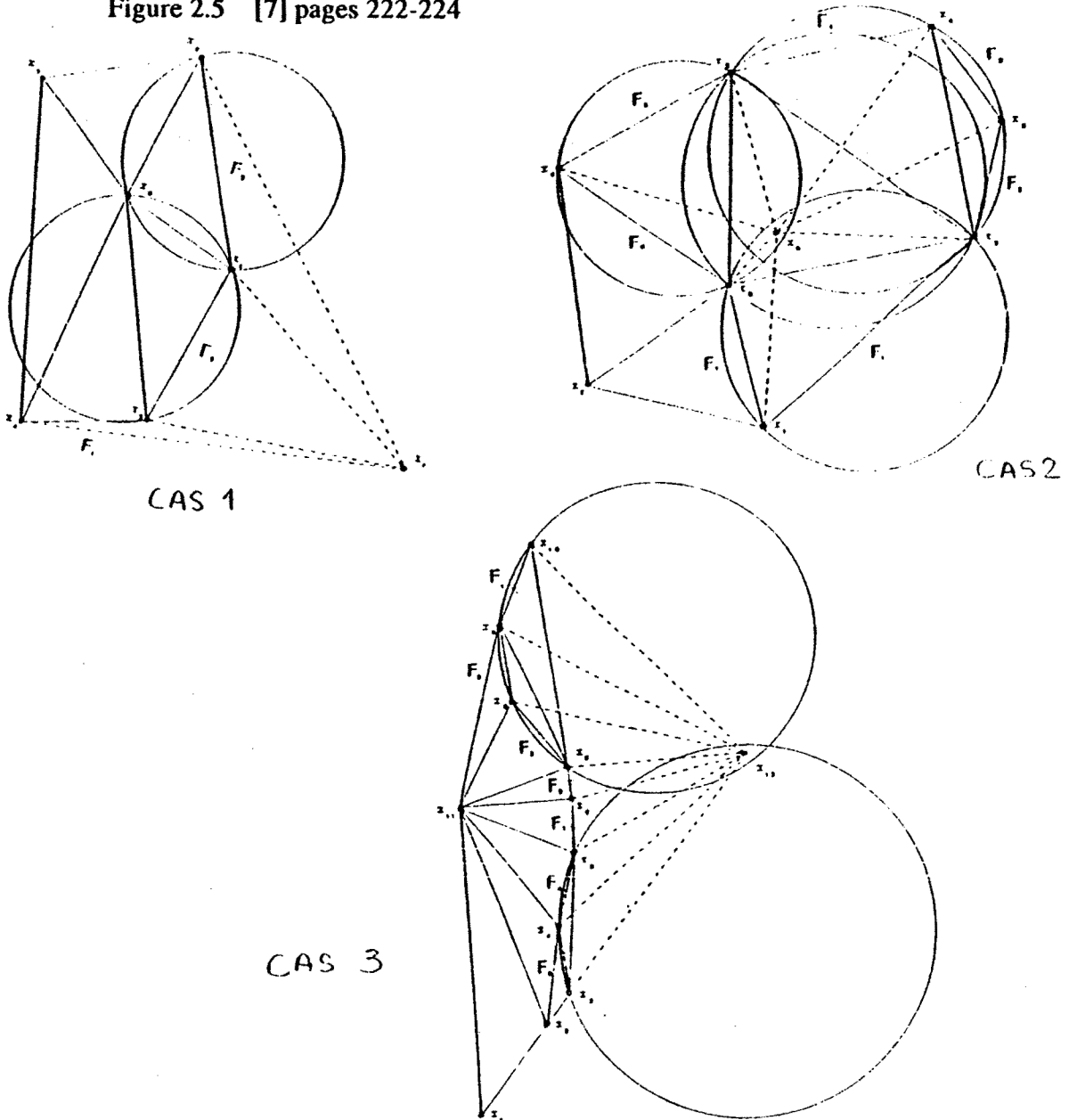
Voici la triangulation de Delaunay associée à l'exemple ci-dessus :

Figure 2.4 [7] page 215



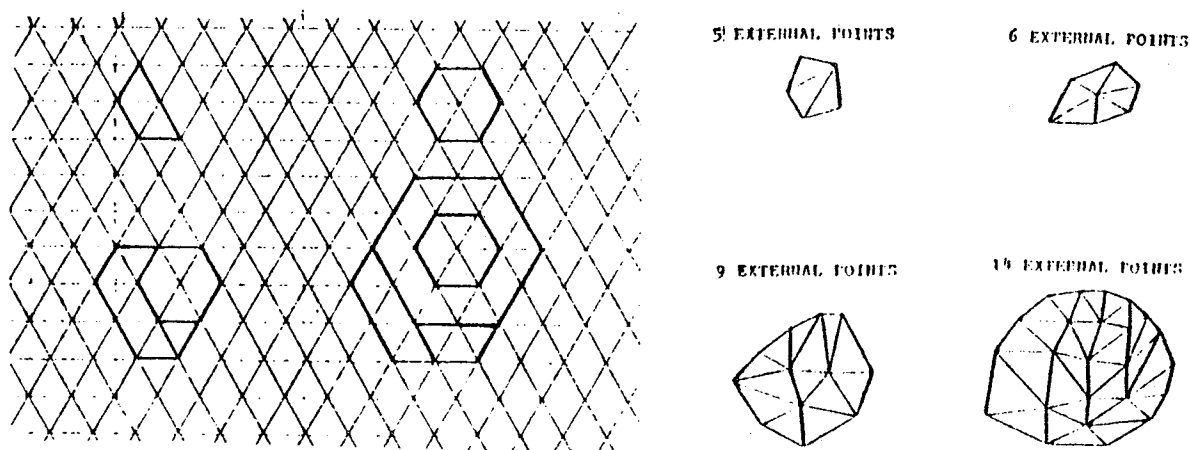
L'algorithme d'Hermeline permet de trianguler un espace non-convexe (en dimension 2) en triangulant d'abord l'enveloppe convexe de cet espace. Ensuite, il effectue la triangulation frontalière et la triangulation des faces singulières en faisant entrer noeud par noeud les noeuds de la frontière et des faces singulières. Pour faire entrer un noeud déjà placé à l'avance dans un maillage connu, il réutilise la triangulation de Delaunay pour la nouvelle triangulation comprenant ce noeud. Voici les trois cas possibles qu'on peut avoir pour remailler avec ce nouveau noeud. Dans le premier cas, aucun triangle actuel (en trait plein) n'a été modifié, et trois nouveaux triangles ont été créés (en pointillé). Dans le second cas, quatre triangles sont remplacés par sept nouveaux, tandis que, dans le troisième, trois triangles sont remplacés par sept nouveaux.

Figure 2.5 [7] pages 222-224



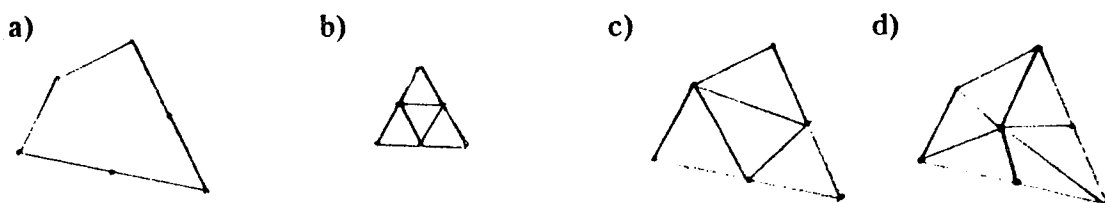
(4) Ayant le souci d'avoir une triangulation telle que les triangles soient le plus équilatéraux possible (en dimension 2), Kamel et Eisenstein dans l'article [9] donne des maillages intéressants en fonction uniquement du nombre de noeuds frontaliers et de la forme de la figure à mailler. Voici un exemple des quatre configurations de base ayant comme nombre de noeuds frontaliers 5; 6; 9 et 14 (cf grille de gauche de la figure 2.6). Ces quatre configurations de base donnent les maillages pour les quatre figures de droite :

Figure 2.6 [9] page 461



S'il veut mailler une figure ayant six noeuds frontaliers comme a) dans la figure 2.7, au lieu d'utiliser la configuration de base donnée dans la grille pour six noeuds frontaliers, il choisirait plutôt la configuration qu'on propose ici en b) qui donne alors comme maillage c) au lieu de d).

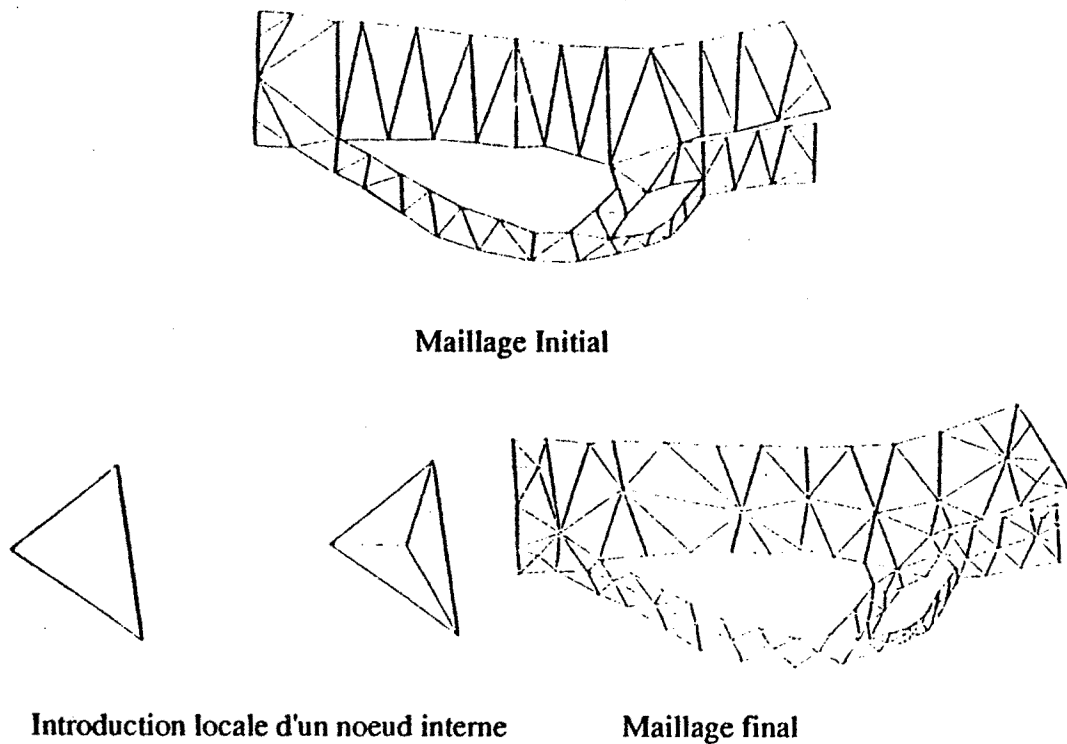
Figure 2.7



Kleinstreuer et Holdeman dans son article [10] présente un algorithme (en deux dimensions) d'un maillage initial des noeuds frontaliers, suivi par l'introduction des noeuds internes. Le maillage initial commence par mailler le plus petit quadrilatère formé

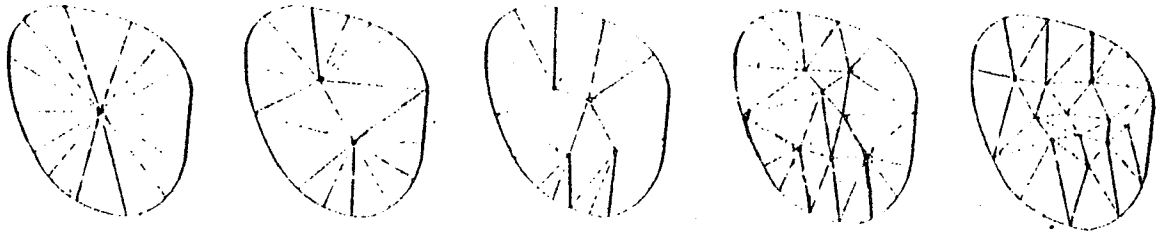
par quatre noeuds frontaliers, en choisissant la diagonale qui forme deux triangles réalisables et les plus équilatéraux possible. En respectant le maillage partiel réalisé à chaque étape, on réitère le processus décrit ci-dessus sur la partie non-maillée de l'espace. A partir de ce maillage initial, un processus d'introduction des noeuds internes et d'un changement de diagonale du quadrilatère affine le maillage de l'étape précédente. pour faire entrer des noeuds dans un maillage en dimension 2 qui ne contient que des noeuds frontaliers.

Figure 2.8 [10] pages 1328-29



L'article de Kamel et Eisenstein [9] donne aussi une technique, celle qu'on appellera "mini-tétraèdrisation" (en dimensions 2 et 3) pour obtenir un maillage simple et facile à construire dans les cas des espaces convexes ou presque convexes. A partir de maillage de la frontière, on crée successivement des noeuds internes en dédoublant le noeud ayant le plus grand nombre de voisins, et en redistribuant ses voisins plus ou moins équitablement entre les deux. Voici un exemple en dimension 2 où le barycentre est le premier noeud interne, suivi par le dédoublement de celui-ci. Ensuite, on dédouble ces deux noeuds internes car ils ont chacun 9 voisins, et ainsi de suite.

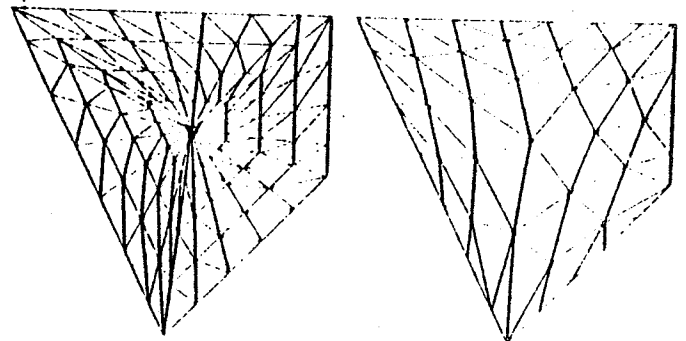
Figure 2.9 [9] page 468



La technique de "mini-tétraèdrisation"

(5) A partir d'un espace à mailler formé entre une surface externe déjà maillée, et une surface interne déjà maillée, la construction des couches se fait en créant des formes entre ces deux qui se déforment progressivement de l'une vers l'autre. La non existence d'une surface interne est un cas particulier. Les maillages s'appuieront sur deux couches voisines. L'utilisation de couches peut beaucoup simplifier la génération initiale d'un maillage. Il n'empêche que cette tétraèdrisation initiale n'est pas acceptable à cause du nombre important de noeuds internes. L'article [9] de Kamel et Eisenstein donne un algorithme assez simple à implémenter dans le cas des espaces convexes ou presque convexes, qu'on appellera "maxi-tétraèdrisation". Il choisit le nombre de couches selon le nombre de noeuds frontaliers, et la valeur proposée = partie entière $((\text{nombre de noeuds frontaliers} + 3)/6)$ en dimension 2. A partir du maillage de la frontière, on relie le barycentre à tous les noeuds frontaliers par des arêtes, et des noeuds internes sont créés par l'intersection de ces arêtes et des couches formées entre le barycentre et le maillage frontalier. Ensuite, on fait une triangulation en choisissant la diagonale de chaque quadrilatère formé (en deux dimensions) et une tétraèdrisation des pentaèdres formés (dimension trois). Pour diminuer le nombre de noeuds internes créés, on confond deux noeuds voisins internes s'ils ont peu de voisins chacun. Voici un exemple en deux dimensions de la "maxi-tétraèdrisation".





Figure 2.10 [9] pages 467,469



La technique de "maxi-tétraèdrisation"

(6) Imafuku, Kodera, Sayawaki et Kono dans l'article [8] présente une méthode pour trianguler une surface en deux dimensions en connaissant seulement les segments frontaliers. Il transforme homomorphiquement la frontière d'une région convexe ou presque convexe en un quadrilatère, et selon le nombre de noeuds sur chaque côté de ce quadrilatère, on génère les noeuds internes et, ensuite, le maillage.

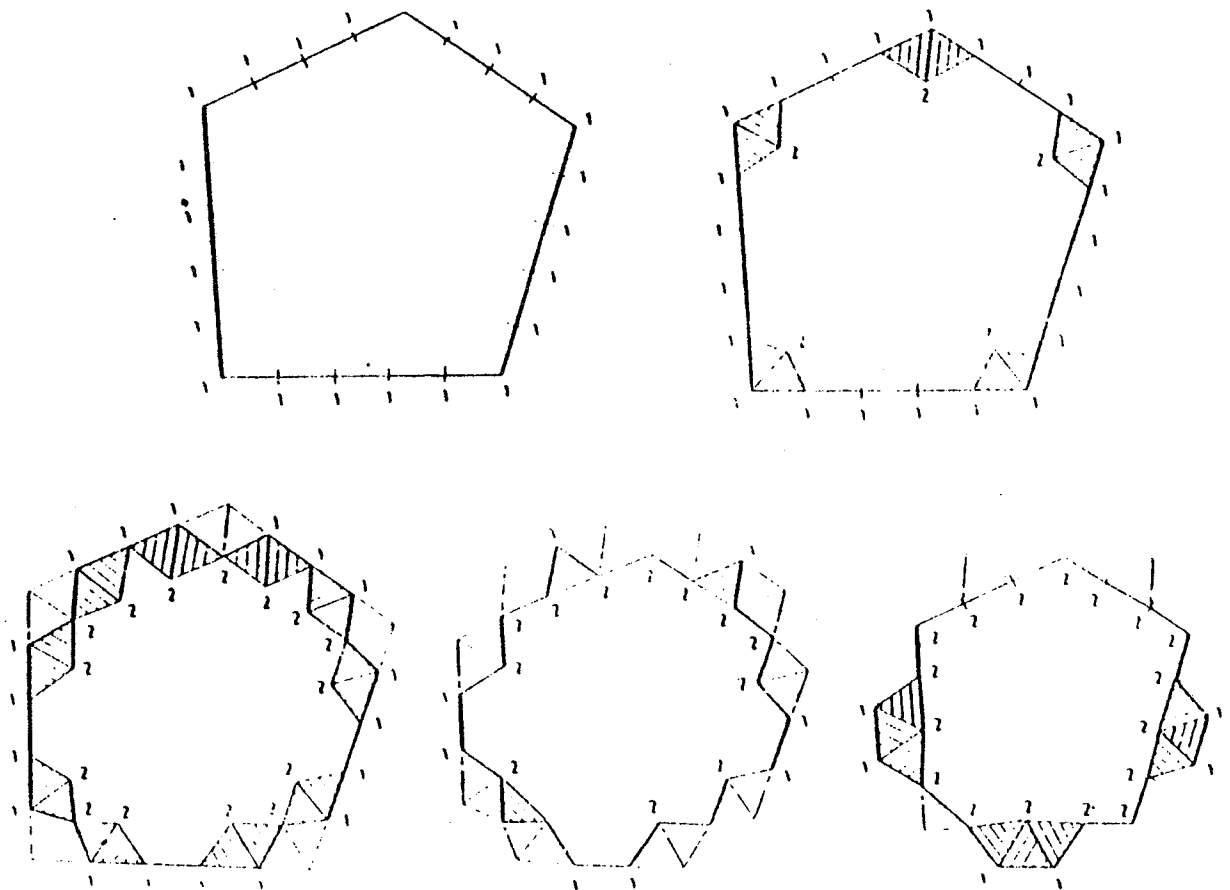
Sadek dans son article [16] part du contour extérieur et en créant des noeuds internes un par un, la région non tétraèdrisée diminue en taille au fur et à mesure que des éléments finis sont créés. En dimension 2, il crée des noeuds internes en utilisant trois noeuds de la frontière de la région non triangulée ayant une de ces deux configurations :

-  qui est maillée de la façon suivante  ;
-  qui est maillée de la façon suivante  .

L'idée de Sadek est illustrée dans la figure suivante :

Figure 2.11 [16] page 1814

(la partie hachurée représente les triangles créés à chaque étape)



2.2.2 Méthodes de l'Amélioration de Maillage

Dans la réalité, on ne peut pas séparer le problème de génération du maillage et celui de l'amélioration du maillage. Ceci se voit surtout dans le domaine de l'amélioration du maillage avec changement de topologie car l'idée de changer de topologie dans une certaine région de l'espace à mailler implique une certaine création de nouveaux éléments, des nouveaux noeuds.

L'amélioration du maillage dépend de ce que doit être un bon maillage. Ce dernier doit satisfaire certaines conditions géométriques, par exemple, les tétraèdres doivent être les plus réguliers possible, comme c'est le cas dans les articles de Kleinstreuer et Holdeman [10], Sadek [16], Nguyen-Van-Phai [13], Mclain [12], Kamel et Eisenstein [9]. Ou bien deux tétraèdres voisins doivent avoir des volumes aussi proches que possible, comme le souhaitent Kleinstreuer et Holdeman [10] et Kamel et Eisenstein [9]. Un bon maillage doit satisfaire aussi certaines conditions liées au problème physique étudié, par exemple on peut souhaiter une forte ou une faible densité de points en des endroits bien précis, ce qui est une condition soulevée par Kleinstreuer et Holdeman [10] et Imafuku, et. al. [8]. Voici quelques méthodes proposées dans le domaine de l'amélioration du maillage.

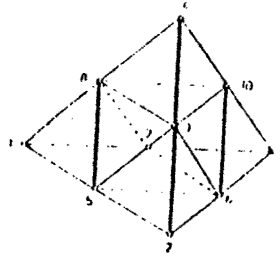
(1) Si on peut augmenter le nombre de noeuds internes, les deux questions qui se posent sont combien de noeuds à ajouter, et où faut-il les situer.

Kamel et Eisenstein dans l'article [9] utilise une procédure itérative pour faire entrer des noeuds internes, en se basant surtout sur le dédoubleage de noeuds entourés par un nombre important de voisins, en limitant, s'il est possible, le nombre de voisins d'un noeud donné à 6 (en dimension 2) et à 10 (en dimension 3).

Kleinstreuer et Holdeman dans l'article [10] propose d'autres règles pour ajouter des noeuds internes afin de mieux restructurer nos éléments dans le maillage.

Nguyen-Van-Phai dans son article [13] propose des choix simples pour l'emplacement des nouveaux noeuds internes. Sur le tétraèdre à modifier, en prenant le milieu de chaque arête du tétraèdre, il construit huit nouveaux tétraèdres à la place de l'ancien. Voici un exemple où l'ancien tétraèdre est décrit par les sommets 1-2-3-4, où 5; 6; 7; 8; 9; 10 sont les milieux de ses arêtes, où les nouveaux tétraèdres sont 8-9-10-7; 1-5-7-8; 2-6-5-9; 3-7-6-10; 5-9-6-7; 8-9-5-7; 6-9-10-7; 9-8-10-7. On peut remarquer que la création de ces nouveaux noeuds entraîne des modifications sur les tétraèdres voisins de l'ancien tétraèdre.

Figure 2.12 [13] page 284



On a déjà vu la méthode proposée par Hermeline dans son article [7] qui propose le voisinage du nouveau noeud à ajouter en utilisant la triangulation de Delaunay. Cf. figure 2.5 page 7.

(2) On peut aussi vouloir diminuer le nombre de noeuds internes, dans le cas où on se trouve avec un nombre de noeuds internes supérieur à celui qu'on souhaite avoir.

La technique de "maxi-tétraèdrisation" de Kamel et Eisenstein fournit une règle simple pour éliminer certains noeuds internes. Notons que pour trouver l'emplacement du nouveau noeud qui remplace les deux anciens, on doit vérifier si on a toujours un maillage de type éléments finis.

(3) Pour détecter les anomalies dans un maillage, Frey, Hall et Porsching dans l'article [5] utilise des plans bien spécifiés qui coupent l'espace maillé (en dimension 3).

(4) La déformation de la forme d'un espace à mailler est utilisé par Kamel dans l'article [9], Imafuku, et. al. dans l'article [8] et Afzali dans l'article [1].

(5) Finalement, il faut aussi noter que certaines méthodes de génération du maillage initial donnent des bons maillages dès le départ, par exemple,

Bykat, [2] pages 1330-35

Kamel et Eisenstein, [9] pages 459-63, 467-71

Mclain, [12] pages 178-81

Kleinstreuer et Holdeman, [10] pages 1325-34

Zienkiewicz et Phillips, [21] pages 522-24

Sadek, [16] pages 1813-22

Enfin, l'étude de toutes ces méthodes montre qu'en général, elles s'appliquent à des cas particuliers et ne permettent pas souvent d'assurer l'obtention de maillage quelle que soit la zone à mailler, en particulier, dans le cas de maillage de l'extérieur d'une forme d'avion. C'est avec le souci de trouver une méthode d'amélioration de maillage applicable à tous les maillages

existants que ce travail a été réalisé, mais rappelons que nous simplifions considérablement le problème en n'autorisant pas de changement de topologie du maillage initial.

2.3 Rappels sur les Méthodes de Gradient et de Sous-Gradient en Optimisation

Les méthodes utilisées pour minimiser une fonction f définie dans l'espace \mathbb{R}^n sont en général des méthodes de descente qui peuvent sommairement se décrire dans la manière suivante:

- A partir d'une solution $v^i \in \mathbb{R}^n$, trouver une direction de descente d^i , c'est à dire, une direction telle que $f(v^i + \lambda d^i) < f(v^i)$ pour λ scalaire positif suffisamment petit.
- Trouver un pas λ^* positif tel que $f(v^i + \lambda^* d^i) < f(v^i)$
- Remplacer v^i par $v^i + \lambda^* d^i$ et itérer ce procédé jusqu'à ce qu'on ne puisse plus améliorer significativement la fonction f .

En général, la direction de descente choisie est l'opposé du gradient ou le gradient conjugué, quand f est dérivable.

Ces méthodes convergent en un nombre infini d'itérations vers l'optimum global dans le cas où la fonction est convexe sous certaines conditions de régularité, sinon on obtient simplement un optimum local de la fonction.

Ces méthodes sont difficiles à mettre en oeuvre dans le cas où la fonction n'est pas différentiable et où le domaine de réalisabilité n'est pas l'espace tout entier. C'est, donc, dans ce cas que la méthode générale connue sous le nom de méthode de sous-gradient donne des résultats rapides et intéressants, si on s'intéresse non pas à trouver l'optimum (global ou local) mais simplement à trouver une solution meilleure que la solution initiale.

Nous ne décrivons pas ici la méthode générale de sous-gradient. Nous décrivons son application dans le cas précis du chapitre 3.

3. SPECIFICATION DU PROBLEME ET DE LA METHODE DE RESOLUTION

3.1 Présentation Sommaire du Problème

Considérons une tétraédrisation donnée de l'espace. En fixant tous les noeuds sauf un, on veut trouver une position nouvelle pour le noeud libre de manière à :

- ne pas remettre en cause la structure de la tétraédrisation actuelle, c'est à dire, obliger le noeud libre à rester dans un certain domaine afin que la structure de la tétraédrisation actuelle soit conservée.
- améliorer la tétraédrisation locale, et, si possible, la tétraédrisation globale selon certains critères d'optimisation.

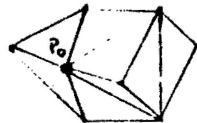
3.2 Etude des Contraintes du Problème

Soit $P_0(x_0, y_0, z_0)$ le noeud qu'on désire bouger, en laissant les autres fixes. Les contraintes que nous imposons sont destinées à ne pas changer la structure de la tétraédrisation existante, c'est à dire que chaque tétraèdre de la tétraédrisation actuelle devra rester comme tétraèdre dans la tétraédrisation améliorée. Ceci impose, bien évidemment, que le noeud reste à l'intérieur du domaine D défini par ses voisins, mais il peut imposer aussi des conditions plus restrictives, comme le montrent les exemples suivants (en dimension 2) :

Si le point n'est plus dans le domaine D défini par ses voisins, alors il faut définir une nouvelle tétraédrisation pour le domaine D auquel appartenait P_0 .

Figure 3.1

- la triangulation initiale



- en déplaçant P_0 hors du domaine D , on trouve



qui n'est pas une triangulation

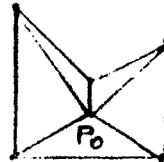


- on peut alors définir une nouvelle triangulation
mais celle ci sera nécessairement distinct
de la triangulation initiale

Si P_0 reste dans le domaine D , il se peut qu'on ne puisse pas garder la même tétraédrisation pour le domaine D .

Figure 3.2

- la triangulation initiale



- Bien que P_0 reste dans D , la triangulation initiale n'est pas réalisable



qui n'est pas une triangulation, même si P_0 est dans D

Si on fait bouger P_0 , on peut montrer que la tétraédrisation initiale n'est plus valable à partir du moment où le volume d'un tétraèdre devient égal à 0.

3.3 Etude des Critères d'Amélioration

Les critères d'amélioration consisteront à maximiser ou minimiser une fonction objective qui dépend entièrement de la position de P_0 , et on note cette fonction $C(P_0)$. Bien entendu, les critères de choix sont purement expérimentaux; il n'y a aucun moyen rigoureux de montrer qu'un critère est meilleur qu'un autre.

Nous nous sommes restrints ici à des critères relatifs aux volumes des tétraèdres entourant P_0 . Néanmoins, la méthode peut être adaptée à n'importe quel type de critère. Il suffit de changer certaines parties précises de nos programmes.

Soient $T_i(P_0)$, $i=1, \dots, m$, les m tétraèdres entourant P_0 , et $V_i(P_0)$, $i=1, \dots, m$, leurs volumes respectifs. Soient $T_{\max}(P_0)$, $T_{\min}(P_0)$ les tétraèdres parmi le $T_i(P_0)$ ayant les volumes max et min, respectivement, et leurs volumes respectifs, $V_{\max}(P_0)$ et $V_{\min}(P_0)$.

Considérons les critères suivant :

(1) minimiser $C_1(P_O) = V_{\max}(P_O) - V_{\min}(P_O)$

(2) minimiser $C_2(P_O) = 1 - (V_{\min}(P_O)/V_{\max}(P_O))$

(3) minimiser $C_3(P_O) = \sum_{i=1}^m |V_i - \bar{V}|$ où $\bar{V} = \sum_{i=1}^m V_i / m$

(4) minimiser $C_4(P_O) = V_{\max}(P_O)$

Ces quatre critères correspondent à la même idée d'essayer d'égaliser tous les volumes des tétraèdres entourant P_O . Cependant, on peut envisager dans les premier, troisième et quatrième critères quelques cas pathologiques où l'un des volumes devient nul. D'autre part, le troisième critère donne une description plus précise de l'idée "d'égaliser tous les volumes des tétraèdres entourant P_O " que les autres critères, mais a-t-on besoin d'une telle précision? L'utilisation du premier et du quatrième critère permet plus de facilité au niveau du calcul de la nouvelle position du P_O car leurs fonctions objectives sont, comme nous le verrons dans 3.5, affines par rapport aux coordonnées de P_O , tandis que le deuxième critère sera une fonction hyperbolique des coordonnées de P_O .

3.4 Formulation Mathématique du Problème

A présent, on peut poser mathématiquement le problème dans la façon suivante:

Trouver P_O tel que $V_i(P_O) \geq 0 \quad i=1, \dots, m$ et minimiser $C(P_O)$.

En utilisant le quatrième critère, la formulation mathématique du problème serait un programme linéaire:

Trouver P_O tel que $0 \leq V_i(P_O) \leq U \quad i=1, \dots, m$ et minimiser U .

De la même façon, pour le premier critère, on aura le programme linéaire:

Trouver P_O tel que $U \geq V_i(P_O) \geq L \geq 0 \quad i=1, \dots, m$ et minimiser $U - L$.

Remarque que le volume d'un tétraèdre ne doit pas être nul, donc $V_i(P_O) > 0$ pour $i=1, \dots, m$.

3.5 Représentation de $V_i(P_0)$, le Volume du $i^{\text{ème}}$ Tétrahédre entourant P_0

Le volume $V_i(P_0)$ dépend paramétriquement des coordonnées de $P_0(x_0, y_0, z_0)$. Soient P_1, P_2, P_3 les trois autres noeuds qui sont fixes du tétraédre $T_i(P_0)$, ayant, respectivement, comme coordonnées $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$. On peut représenter $V_i(P_0)$ de la manière suivante :

$$V_i(P_0) = (1/3!) \det \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \\ z_0 & z_1 & z_2 & z_3 \end{vmatrix} \quad \text{à une permutation de colonnes près. } 31$$

Soient

$$\begin{aligned} r_i &= \det \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{vmatrix} \\ a_i &= -\det \begin{vmatrix} 1 & 1 & 1 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{vmatrix} \\ b_i &= \det \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ z_1 & z_2 & z_3 \end{vmatrix} \\ c_i &= -\det \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \end{aligned}$$

On a donc $V_i(P_0) = r_i + a_i x_0 + b_i y_0 + c_i z_0$ qui est une fonction linéaire en x_0, y_0, z_0 .

On voit que, pour l'intérêt du calcul, on peut calculer une fois pour toutes les mineurs r_i, a_i, b_i, c_i et les stocker pour ne pas les recalculer à chaque fois que l'on a besoin d'eux.

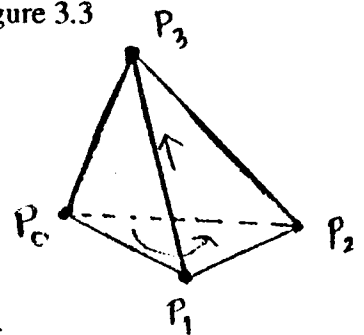
$V_i(P_0)$ est la valeur absolue du déterminant. En dimension 2, en choisissant l'ordre des noeuds du triangle dans le sens contraire des aiguilles d'une montre, en commençant par P_0 , on est assuré de la valeur positive du déterminant.

En dimension 3, l'ordre des noeuds est pris en sens direct en commençant par P_0 ,

(cf figure 3.3). On peut donc énoncer les contraintes du problème énoncé dans 3.4 comme:

$$V_i(P_0) = r_i + a_i x_0 + b_i y_0 + c_i z_0 \geq 0 \quad i = 1, \dots, m.$$

Figure 3.3



3.6 Présentation Sommaire de la Méthode de Sous-Gradient Utilisée en tant que Méthode de Résolution du Problème

On va utiliser une méthode connue sous le nom de sous-gradient qui est rapide mais, qui, en contre partie, même dans le cas de programmation linéaire, ne conduit jamais à une solution optimale en un nombre fini d'itérations.

Cette méthode consiste à trouver une direction d'amélioration δ , c'est à dire, soit P_0 positionné en (x_0, y_0, z_0) , une direction δ sera appelée une direction d'amélioration pour le critère $C(P_0)$, s'il existe un ρ scalaire positif suffisamment petit tel que $C(P_0 + \rho\delta) < C(P_0)$. Ceci veut dire qu'un petit déplacement du point P_0 dans la direction δ assure une amélioration de la valeur du critère d'amélioration.

Prenons le premier critère comme notre critère d'amélioration, δ est une direction d'amélioration au point $P_0(x_0, y_0, z_0)$, s'il existe un ρ scalaire positif suffisamment petit tel que

$$C_1(P_0 + \rho\delta) < C_1(P_0) \quad \text{c'est à dire}$$

$$V_{\max}(P_0 + \rho\delta) - V_{\min}(P_0 + \rho\delta) < V_{\max}(P_0) - V_{\min}(P_0).$$

Dans notre méthode, on se placera toujours aux point où la fonction critère est différentiable, et, par conséquent, la direction d'amélioration sera fournie par l'opposé du gradient de la fonction objective.

Ayant trouvé la direction d'amélioration δ au point $P_0(x_0, y_0, z_0)$, nous prenons un pas ρ scalaire dans la direction δ , telle que le point $P_0^* = P_0 + \rho\delta$ est, d'une part, dans le domaine défini par les contraintes, et, d'autre part, vérifie la relation $C(P_0^*) \leq C(P_0)$ quand la fonction objective est à minimiser.

4. PRESENTATION DETAILLEE DES ALGORITHMES ALGO1 ET ALGOTOUT

4.1 ALGO1 - Algorithme pour Bouger un Seul Noeud P_0

4.1.1 Le Choix de la Direction de Descente en tant que Direction d'Amélioration

Supposons qu'à une certaine itération, un des tétraèdres, par exemple, $T_1(P_0)$, soit le seul tétraèdre ayant le volume minimum, et un autre, par exemple, $T_2(P_0)$, soit le seul tétraèdre ayant le volume maximum. Alors on a

$$V_{\max}(P_0) = V_2(P_0) \text{ et } V_{\min}(P_0) = V_1(P_0)$$

$$\text{où } V_1(P_0) = r_1 + a_1x_0 + b_1y_0 + c_1z_0 \text{ et } V_2(P_0) = r_2 + a_2x_0 + b_2y_0 + c_2z_0.$$

Prenons le premier critère, minimiser $C_1(P_0) = V_{\max}(P_0) - V_{\min}(P_0)$,

$$C_1(P_0) = r_2 - r_1 + (a_2 - a_1)x_0 + (b_2 - b_1)y_0 + (c_2 - c_1)z_0.$$

On a la direction d'amélioration $\delta = (\delta x_0 \ \delta y_0 \ \delta z_0)$ donné par le gradient, c'est à dire, $\delta = -(a_2 - a_1, b_2 - b_1, c_2 - c_1)$.

Prenons le deuxième critère, minimiser $C_2(P_0) = 1 - (V_{\min}(P_0)/V_{\max}(P_0))$,

$$C_2(P_0) = 1 - ((r_1 + a_1x_0 + b_1y_0 + c_1z_0)/(r_2 + a_2x_0 + b_2y_0 + c_2z_0))$$

qui nous donne comme direction d'amélioration $\delta = (\delta x_0 \ \delta y_0 \ \delta z_0)$ le gradient, c'est à dire,

$$\delta x_0 = \frac{r_1 a_2 - r_2 a_1 + (b_1 a_2 - b_2 a_1)y_0 + (c_1 a_2 - c_2 a_1)z_0}{(r_2 + a_2x_0 + b_2y_0 + c_2z_0)^2}$$

$$\delta y_0 = \frac{r_1 b_2 - r_2 b_1 + (b_2 a_1 - b_1 a_2)x_0 + (c_1 b_2 - c_2 b_1)z_0}{(r_2 + a_2x_0 + b_2y_0 + c_2z_0)^2}$$

$$\delta z_0 = \frac{r_1 c_2 - r_2 c_1 + (c_2 a_1 - c_1 a_2)x_0 + (c_2 b_1 - c_1 b_2)y_0}{(r_2 + a_2 x_0 + b_2 y_0 + c_2 z_0)^2}$$

Une autre direction d'amélioration δ est plus simplement: $\delta = (\delta x_0 \ \delta y_0 \ \delta z_0)$ avec $\delta x_0 = -(a_2 - a_1)$, $\delta y_0 = -(b_2 - b_1)$, $\delta z_0 = -(c_2 - c_1)$, car il existe un ρ scalaire positif suffisamment petit qui donne

$$C_2(P_0 + \rho\delta) - C_2(P_0) = \frac{\rho (\delta x_0(a_2 - a_1) + \delta y_0(b_2 - b_1) + \delta z_0(c_2 - c_1))}{r_2 + a_2 x_0 + b_2 y_0 + c_2 z_0} \leq 0.$$

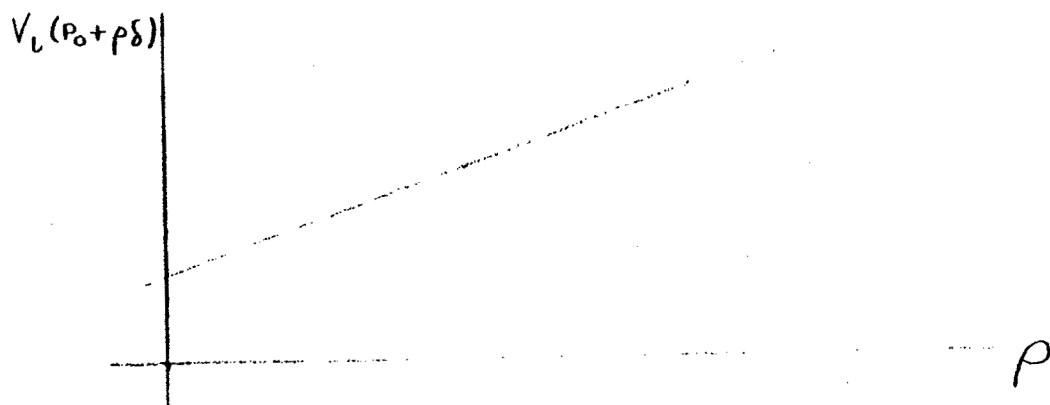
4.1.2 Le Choix du Pas à Prendre dans la Direction d'Amélioration

Soit la direction d'amélioration $\delta = (\delta x \ \delta y \ \delta z)$ trouvée dans 4.1.1. Le choix du pas ρ à prendre dans la direction d'amélioration δ donnée peut se représenter graphiquement de la manière suivante :

$$\begin{aligned} \text{On sait que } V_i(P_0 + \rho(\delta x \ \delta y \ \delta z)) &= V_i(x_0 + \rho\delta x \ y_0 + \rho\delta y \ z_0 + \rho\delta z) \\ &= r_i + (x_0 + \rho\delta x)a_i + (y_0 + \rho\delta y)b_i + (z_0 + \rho\delta z)c_i \\ &= (r_i + a_i x_0 + b_i y_0 + c_i z_0) + \rho(a_i \delta x + b_i \delta y + c_i \delta z) \text{ est affine en } \rho. \end{aligned}$$

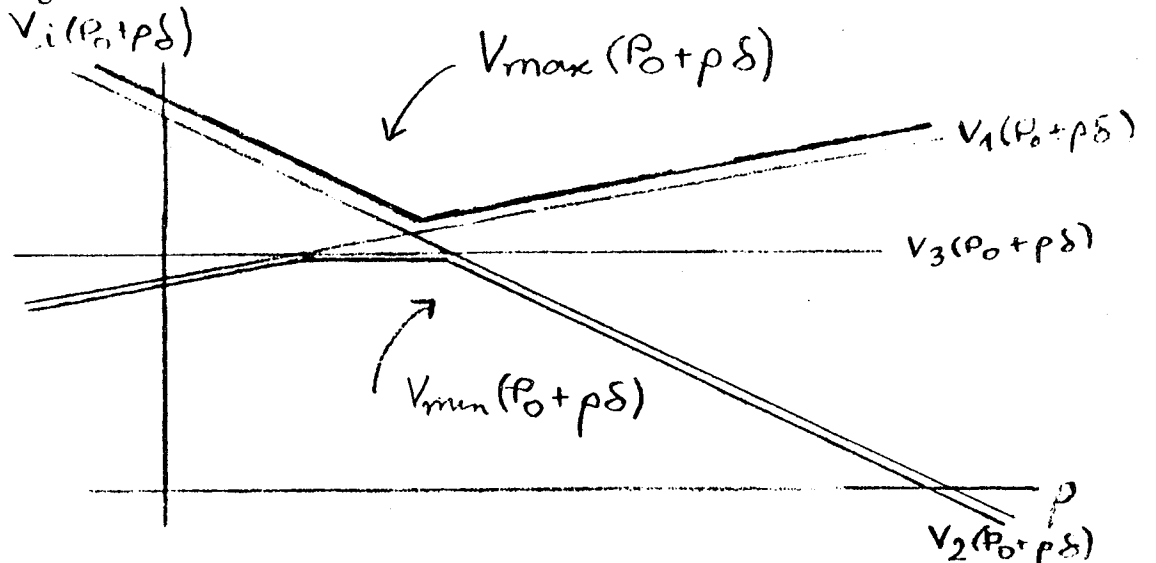
Représentons la fonction $\rho \mapsto V_i(P_0 + \rho\delta)$ sur le diagramme suivant :

Figure 4.1



Représentons tous les $V_i(P_0 + \rho\delta)$ de cette manière sur un seul repère, comme dans la figure suivante où il y a trois tétraèdres qui entourent P_0 .

Figure 4.2

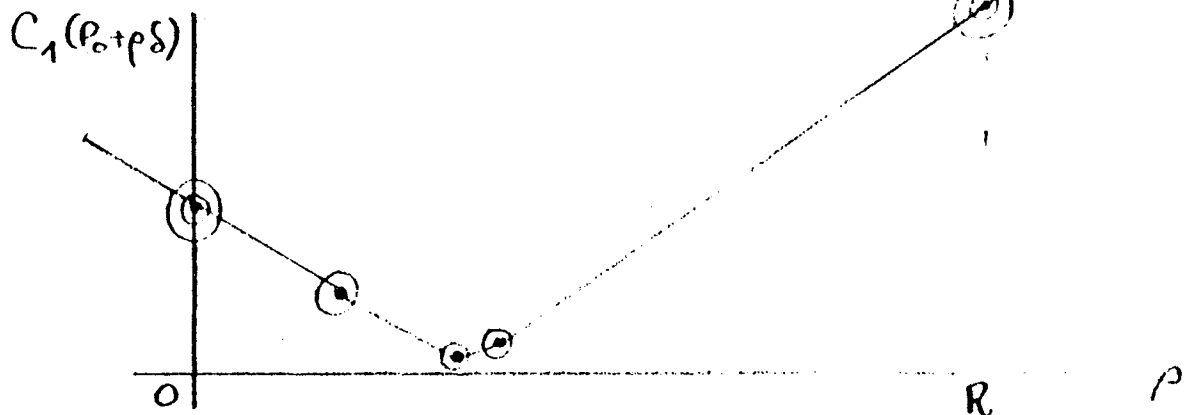


On voit que $V_{\max}(P_0 + \rho\delta)$ et $V_{\min}(P_0 + \rho\delta)$ sont des fonctions continues affines par morceaux. On voit dans la figure 4.2 que les fonction $\rho \mapsto V_{\max}(P_0 + \rho\delta)$ et $\rho \mapsto V_{\min}(P_0 + \rho\delta)$ sont les enveloppes supérieure et inférieure respectivement d'un nombre fini de fonctions affines.

Pour le critère $C_1(P_0)$, $C_1(P_0 + \rho\delta) = V_{\max}(P_0 + \rho\delta) - V_{\min}(P_0 + \rho\delta)$.

Représentons la fonction $\rho \mapsto C_1(P_0 + \rho\delta)$ de l'exemple de la figure 4.2 :

Figure 4.3



On voit que $C_1(P_0 + \rho\delta)$ est aussi une fonction continue affine par morceaux (car c'est la différence de deux fonctions continues affines par morceaux) et elle se déduit comme l'écart entre les deux enveloppes mentionnés ci-dessus. $C_1(P_0)$ est une fonction différentiable presque partout. Elle est non dérivable en un nombre fini de points et ces points de cassure sont encerclés une fois dans la figure 4.3. En plus, le domaine de réalisabilité n'est pas l'espace tout entier car $\rho \geq 0$ et $V_{\max}(P_0 + \rho\delta), V_{\min}(P_0 + \rho\delta) > 0$, donc le domaine de définition pour ρ est $[0, R[$ où R est un nombre réel positif. Notons que les méthodes de gradient ne sont pas applicables pour les points de "cassure" c'est à dire pour les points où la fonction est non différentiable. C'est donc pour cette raison qu'on emploie des méthodes de sous-gradient.

Dans les méthodes de sous-gradient, le choix du pas est crucial. Des résultats théoriques existent (voir l'article [22] de Held et Crowder) garantissant la convergence en un nombre infini d'itérations mais en pratique ces résultats ne s'appliquent guère.

Comment, donc, choisir le pas ρ pour un δ donné?

On doit faire un bon choix du pas ρ , et une manière de le faire nous est tout à fait évidente :

Soit $\rho = \rho_0$, la valeur du ρ qui optimise le critère d'amélioration. En général, pour $\rho = \rho_0$, on tombe dans la situation où il y a plusieurs tétraèdres dont le volume est minimum (ou maximum). Si notre fonction critère est en fonction de $V_{\max}(P_0)$ et $V_{\min}(P_0)$, il est nécessaire d'avoir un seul tétraèdre ayant le volume maximum et un seul ayant le volume minimum. Cette condition n'est pas satisfaite pour le pas ρ_0 ; on suggère une manière pour remédier cette situation : on va chercher une valeur pour ρ à proximité de ρ_0 . Cette valeur va nous permettre d'avoir la valeur de la fonction critère à proximité de la valeur optimale de la fonction critère qui correspond à ρ_0 . Afin d'éviter que cette valeur de ρ retombe dans la situation décrite précédemment, on va la situer entre ρ_0 et la première valeur de $\rho > \rho_0$ pour laquelle cette situation se renouvelle. Soit $\rho = \rho^*$ la première valeur de $\rho > \rho_0$ pour laquelle cette situation se renouvelle. Comme on veut que la valeur du pas ρ à prendre dans la direction d'amélioration est

à proximité de ρ_0 , notre choix du pas ρ sera :

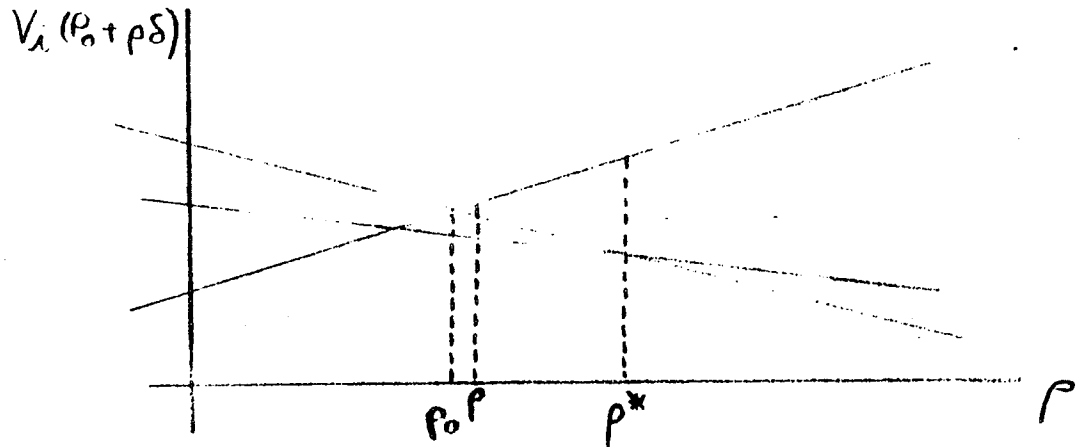
$$\rho = \rho_0 + \varepsilon \times (\rho^* - \rho_0) \quad \text{où } \varepsilon \text{ est un scalaire positif entre 0 et 1;}$$

dans le cas où ρ^* n'existe pas, $\rho = \rho_0 + \varepsilon \times \rho_0$.

Le choix de ε est arbitraire; une petite valeur est souhaitée, par exemple, 0,01.

Voici une illustration graphique de choix du pas ρ .

Figure 4.5



On peut déduire que pour cette valeur de ρ , $C_1(P_0 + \rho\delta)$ est continue et dérivable car ce n'est pas un point de "cassure" pour $C_1(P_0 + \rho\delta)$. L'utilisation de la direction de plus forte descente, c'est à dire du gradient comme direction d'amélioration est justifiable.

4.1.3 Résumé de la Méthode de Résolution

Préliminaires :

- choisir le noeud $P_0(x_0, y_0, z_0)$ à bouger.
- chercher les tétraèdres $T_i(P_0)$ qui entourent P_0 .
- calculer les mineurs r_i, a_i, b_i, c_i du déterminant qui représentent le volume $V_i(P_0)$ pour chacun de ces tétraèdres.
- calculer les volumes initiaux pour chacun de ces tétraèdres.
- chercher les tétraèdres ayant le plus grand volume et le plus petit volume.

Appellons ces volumes, respectivement, $V_{\max}(P_0)$ et $V_{\min}(P_0)$.

Déroulement d'une Itération :

- trouver la direction d'amélioration $\delta = (\delta x_0 \delta y_0 \delta z_0)$.
 - trouver le pas ρ à prendre dans cette direction :
 - chercher ρ_0 .
 - si $\rho_0 = 0$ ou $V_{\min}(P_0 + \rho_0 \delta) \leq 0$, s'arrêter ou aller au prochain noeud à bouger.
 - sinon, calculer la nouvelle position de P_0 , c'est à dire, $(x_0 + \rho \delta x_0; y_0 + \rho \delta y_0; z_0 + \rho \delta z_0)$ et la valeur de la fonction critère pour P_0 .
 - calculer les nouveaux volumes $V_i(P_0)$.
 - notons que $T_{\max}(P_0)$, $T_{\min}(P_0)$ sont connus à la fin de l'itération, et que les mineurs r_i , a_i , b_i , c_i du déterminant qui représentent le volume $V_i(P_0)$ sont constantes.
- On va appeler le programme informatique de la méthode de résolution proposée ici ALGO1.

4.1.4 Un Exemple de l'Exécution de l'ALGO1 (en dimension 2) (pour les détails informatique, cf Annexe 4.1)

On désire bouger le noeud $P_0=6$

Figure 4.6 Figure Initiale

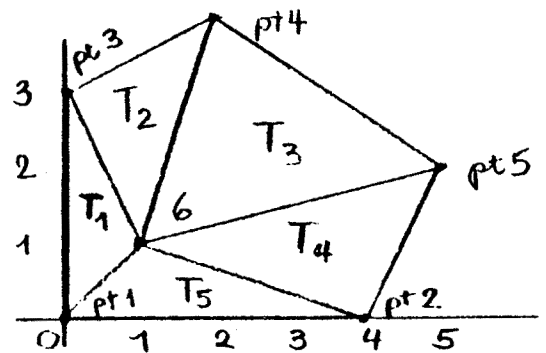
dont les coordonnées (x_0, y_0) sont $(1;1)$.

Ce noeud est entourés par les triangles T_1, T_2, \dots, T_5 , comme on le voit sur la figure 4.6. La fonction objective est

à minimiser $C_1(P_0) = V_{\max}(P_0) - V_{\min}(P_0)$,

la valeur de ϵ utilisée pour

trouver le pas ρ est 0,01.



Pour calculer $V_1(P_0)$, le volume de $T_1(P_0)$, le triangle ayant les sommets $(1;3;6)$, on a

$$V_1(P_0) = \det \begin{vmatrix} 1 & 1 & 1 \\ x_0 & 0 & 0 \\ y_0 & 0 & 3 \end{vmatrix} \quad \text{dont les mineurs sont}$$

$$r_1 = \det \begin{vmatrix} 0 & 0 \\ 0 & 3 \end{vmatrix} = 0 \quad a_1 = -\det \begin{vmatrix} 1 & 1 \\ 0 & 3 \end{vmatrix} = -3 \quad b_1 = \det \begin{vmatrix} 1 & 1 \\ 0 & 0 \end{vmatrix} = 0.$$

$$\text{On a } V_1(P_0) = 1 \times 0 + -3 x_0 + 0 y_0 = -3 x_0$$

Car $x_0 = 1$, pour que $V_1(P_0) > 0$, il faut changer les signes de r_1 , a_1 , b_1 . On a donc $r_1=0$, $a_1=3$, $b_1=0$ et $V_1(P_0) = 3x_0$. De la même manière, on trouve $V_2(P_0) = 6+x_0-2y_0$, $V_3(P_0) = 16-2x_0-3y_0$, $V_4(P_0) = 8-2x_0+y_0$, $V_5(P_0) = 4y_0$ et pour $P_0(1;1)$, $V_1(P_0)=3$, $V_2(P_0)=5$, $V_3(P_0)=9$, $V_4(P_0)=7$, $V_5(P_0)=4$. Le $V_{\max}(P_0)$ initial est 11 pour le triangle T_3 ; le $V_{\min}(P_0)$ initial est 3 pour le triangle T_1 , et la valeur initiale de $C_1(P_0) = V_{\max}(P_0) - V_{\min}(P_0)$ est $11 - 3 = 8$. La direction de descente, donnée par le gradient de $C_1(P_0)$, $\delta = -(a_2 - a_1, b_2 - b_1, c_2 - c_1)$, est $-((-2) - 3; (-3) - 0) = (5; 3)$. Les applications $V_i(P_0 + \rho\delta)$ affines en ρ sont trouvées de la façon suivante:

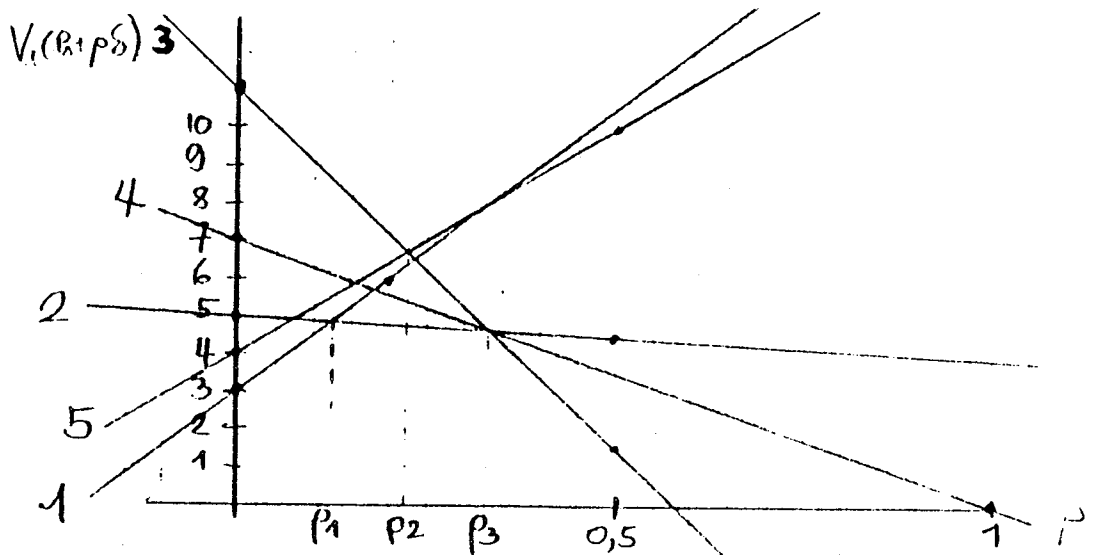
$$\text{Pour } V_1(P_0 + \rho\delta), \text{ on a } V_1(1 + 5\rho; 1 + 3\rho) = 0 + 3(1+5\rho) + 0(1+3\rho) = 3 + 15\rho.$$

$$\text{De la même manière, } V_2(P_0 + \rho\delta) = 5 - \rho, \quad V_3(P_0 + \rho\delta) = -11 + 19\rho, \quad V_4(P_0 + \rho\delta) = 7 - 7\rho,$$

$$V_5(P_0 + \rho\delta) = 4 + 12\rho.$$

Voici la représentation graphique de $V_i(P_0 + \rho\delta)$ en ρ :

Figure 4.7

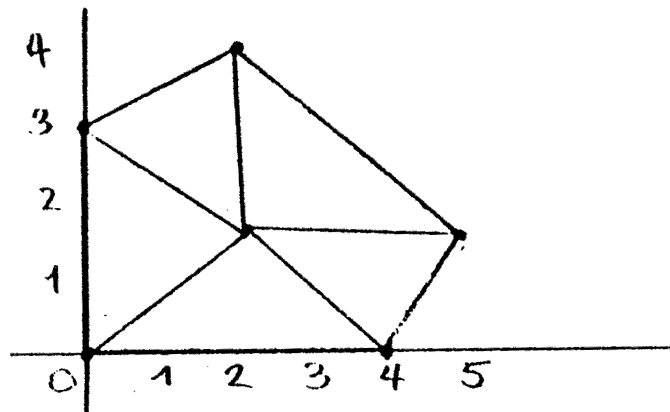


Les pas intermédiaires à la recherche du pas ρ à prendre dans la direction d'amélioration $\delta=(5;3)$ sont $\rho_1=0,125$; $\rho_2=0,22580645$; $\rho_3=1/3$.

Pour arriver à ρ_1 on a $T_{\max}=T_3$, $T_{\min}=T_1$; à ρ_2 on a $T_{\max}=T_3$, $T_{\min}=T_2$; à ρ_3 on a $T_{\max}=T_5$, $T_{\min}=T_2$. Le pas optimal ρ_0 est ρ_2 ; ρ^* est ρ_3 ; et, en prenant pour valeur de $\epsilon=0,01$, $\rho = \rho_0 + 0,01 \times (\rho^* - \rho_0) = 0,2258... + 0,01(0,1075) = 0,2268...$ Le nouveau plus grand triangle T_{\max} est T_5 , et le plus petit T_{\min} est T_2 . On trouve les nouvelles superficies des triangles en substituant la valeur de ρ trouvé dans les $V_i(P_0 + \rho\delta)$: $V_1(P_0 + \rho\delta) = 3 + 15\rho = 3 + 15(0,2268...) = 6,4032$, $V_2(P_0 + \rho\delta) = 4,7731$, $V_3(P_0 + \rho\delta) = 6,689$, $V_4(P_0 + \rho\delta) = 5,4118$, $V_5(P_0 + \rho\delta) = 6,7225$, et on trouve $V_{\max} - V_{\min} = 6,7225 - 4,7731 = 1,9494$ comme nouvelle valeur de notre fonction critère $C_1(P_0)$. L'ancienne valeur de $C_1(P_0)$ était 8, donc on voit qu'il y a une amélioration de la fonction critère $C_1(P_0)$. La nouvelle position du noeud 6 est trouvée de la manière suivante : $P_0 + \rho\delta = (1 + 5\rho; 1 + 3\rho) = (2,1344; 1,6806)$.

Voici la figure finale:

Figure 4.8



4.1.5 Trois Exemples de l'Exécution de l'ALGO1 en Faisant Varier la Fonction Objective

La figure initiale qui correspond au noeud à bouger numeroté 10 et à ses voisins numerotés 4;6;9;11;14;15;17 est figure 4.9. Les figure finales sont les figures 4.10, 4.11, 4.12 pour les fonctions objectives $C_1(P_0)$, $C_2(P_0)$ et $C_3(P_0)$, respectivement (cf 3.3). Les détails de l'exécution de l'ALGO1 se trouve dans les Annexes 4.2, 4.3, et 4.4, respectivement.

Figure 4.9

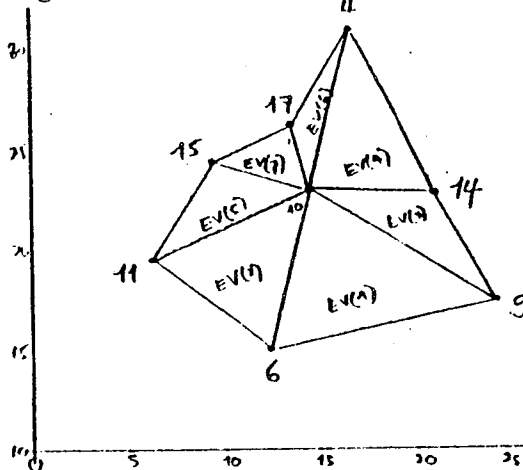


Figure 4.11

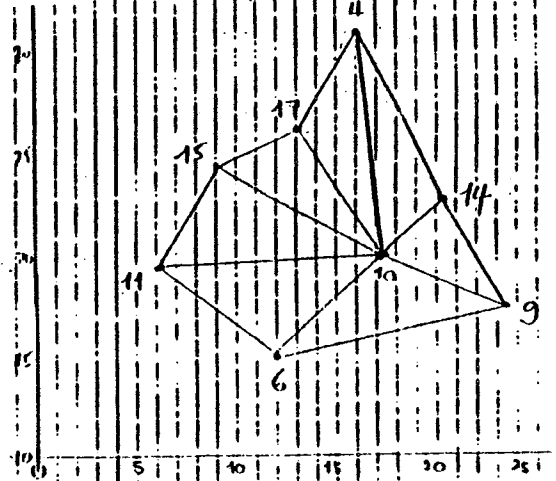


Figure 4.10

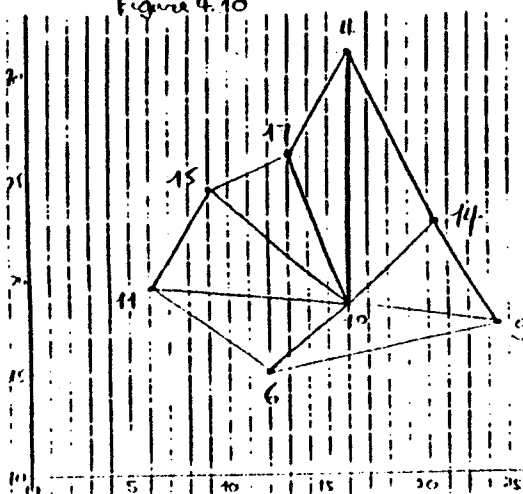
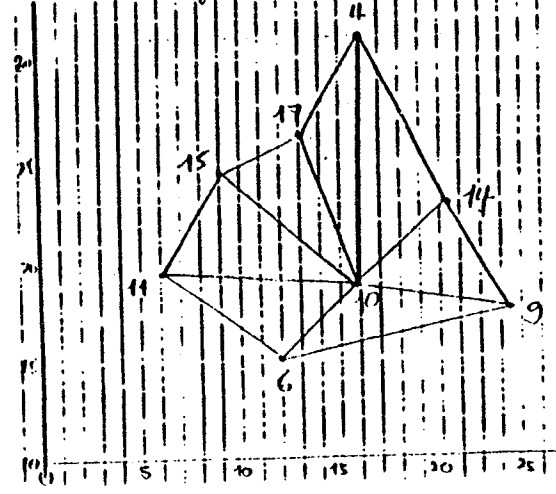


Figure 4.12



Voici les valeurs initiales et finales de la fonction critère utilisée :

- $C_1(P_0)$, initiale=80,75, finale=16,58
- $C_2(P_0)$, initiale=0,85, finale=0,40
- $C_3(P_0)$, initiale=22,07, finale=6,15

4.2 ALGOTOUT - Algorithme pour Bouger un Ensemble de Noeuds en les Bougeant Un par Un

On a utilisé la méthode de résolution proposée dans un autre cadre plus général pour tenter d'améliorer localement et globalement le maillage par un processus itératif qui applique un certain nombre de fois l'ALGO1, l'algorithme qui bouge un seul noeud une seule fois. Ceci comporte deux parties :

- on désire bouger un seul noeud plusieurs fois et on appelle ce nombre de fois ITERATION. Sachant que les méthodes de sous-gradient sont rapides dans les premières itérations, plusieurs itérations successives de l'ALGO1 sont à appliquer à un noeud P_0 pour avoir une meilleure amélioration locale;

- on désire bouger un ensemble de noeuds plusieurs fois (en bougeant les noeuds un par un selon un certain ordre) et on appelle ce nombre de fois TOUR. Cette partie nous apportera-t-elle une amélioration globale au niveau de la qualité du maillage de l'ensemble des noeuds?

Appelons l'algorithme qui bouge un ensemble de noeuds un certain nombre de fois en les bougeant un par un et chacun plusieurs fois de suite ALGOTOUT.

4.2.1 Le Choix de la Fonction Objective à Minimiser

La recherche du pas à prendre dans la direction d'amélioration va dépendre uniquement de la fonction objective $C_1(P_0) = V_{\max}(P_0) - V_{\min}(P_0)$ à cause de la facilité d'obtenir les pas intermédiaires et le pas optimal par $C_1(P_0)$. Ceci ne nous empêche pas de tester si la nouvelle position du noeud P_0 améliore aussi d'autres fonctions objectives telles que

$$(1) \text{ minimiser } C_2(P_0) = 1 - (V_{\min}(P_0)/V_{\max}(P_0)),$$

$$(2) \text{ minimiser } C_3(P_0) = \sum_{i=1}^m |V_i - V| \quad \text{où } V = \sum_{i=1}^m V_i / m$$

$$(3) \text{ minimiser } C_4(P_0) = V_{\max}(P_0)$$

ou certains autres critères, par exemple, la régularité des tétraèdres.

Deux exemples ont été donnés en utilisant $C_2(P_0)$ et $C_3(P_0)$ (cf 4.1.5), où, pour $C_2(P_0)$, la figure finale est la figure 4.11; et, pour $C_3(P_0)$, la figure finale est la figure 4.12.

4.2.2 Le Choix du Nombre d'itérations Successives pour Bouger un Noeud et du Nombre de Balayages pour Bouger un Ensemble de Noeuds

Souhaitant que le nombre d'itérations successives pour bouger un seul noeud ne soit pas trop élevé, on peut imposer une valeur fixe, par exemple 2; 3; 4 ou 5. Un travail préliminaire effectué montre que la grande part de l'amélioration est due à la première itération, les autres servent plutôt à raffiner la première. Dans ce cas, le choix de la valeur 3 pour le nombre d'itérations semble assez suffisant. Voici deux exemples en dimension 2 qui confirment l'importance de la première itération :

Figure 4.13 Bouger le noeud 6 avec ITERATION = 5 et la fonction objective

$$C_1(P_0) = V_{\max} - V_{\min} \text{ (pour les détails informatiques, cf Annexe 5.1)}$$

(Les fleches indiquent la direction d'amélioration pour l'itération suivante)

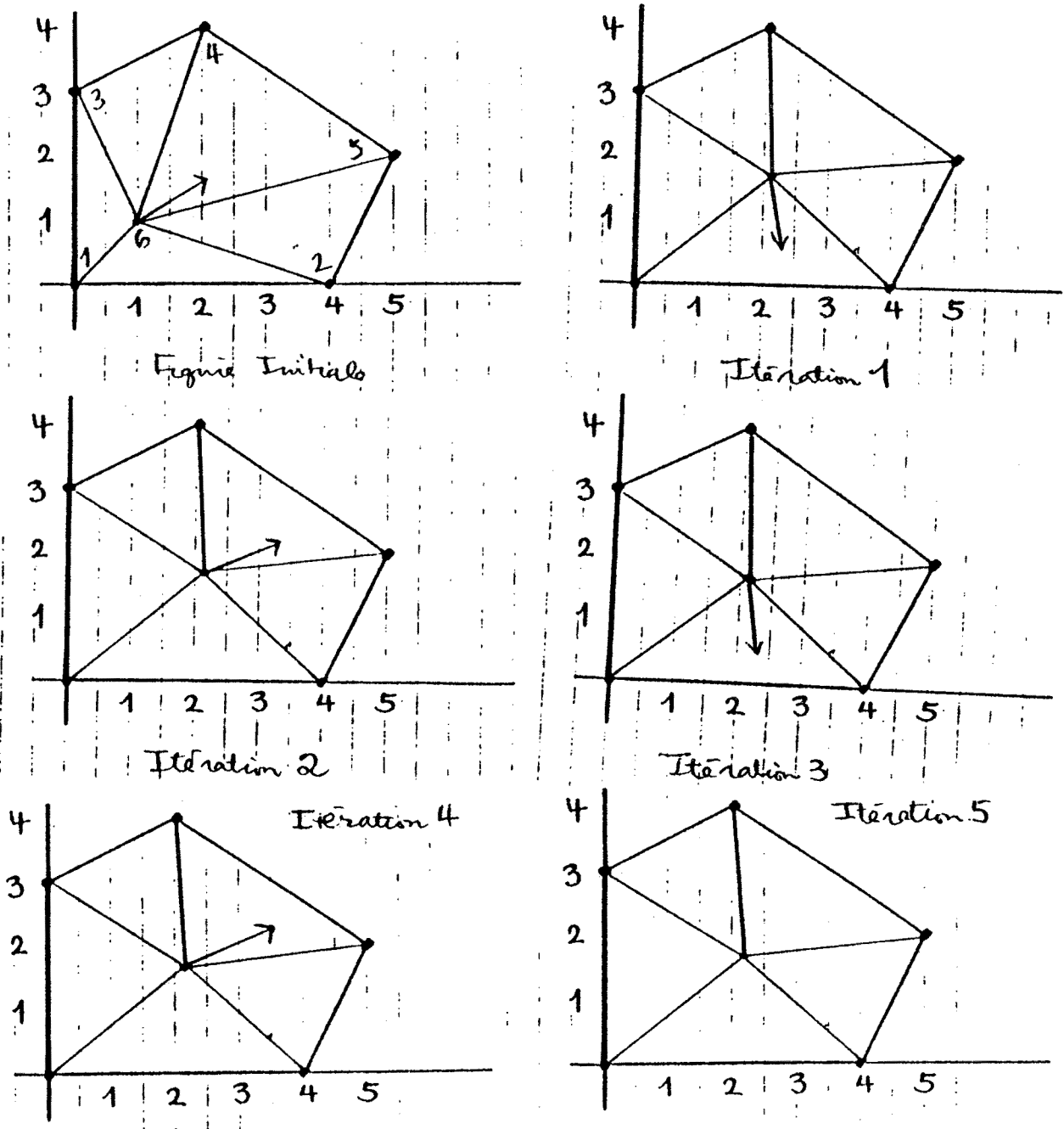
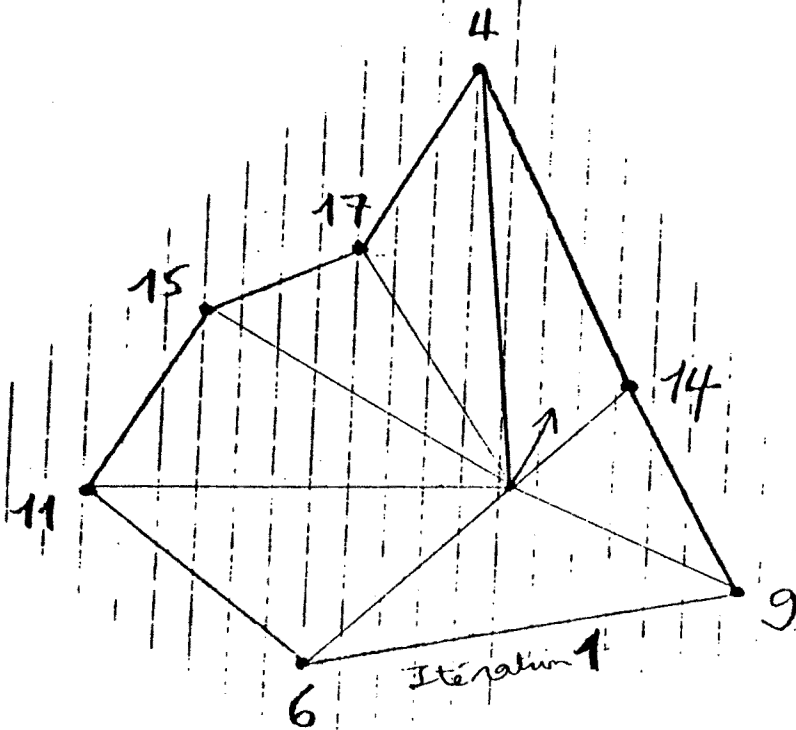
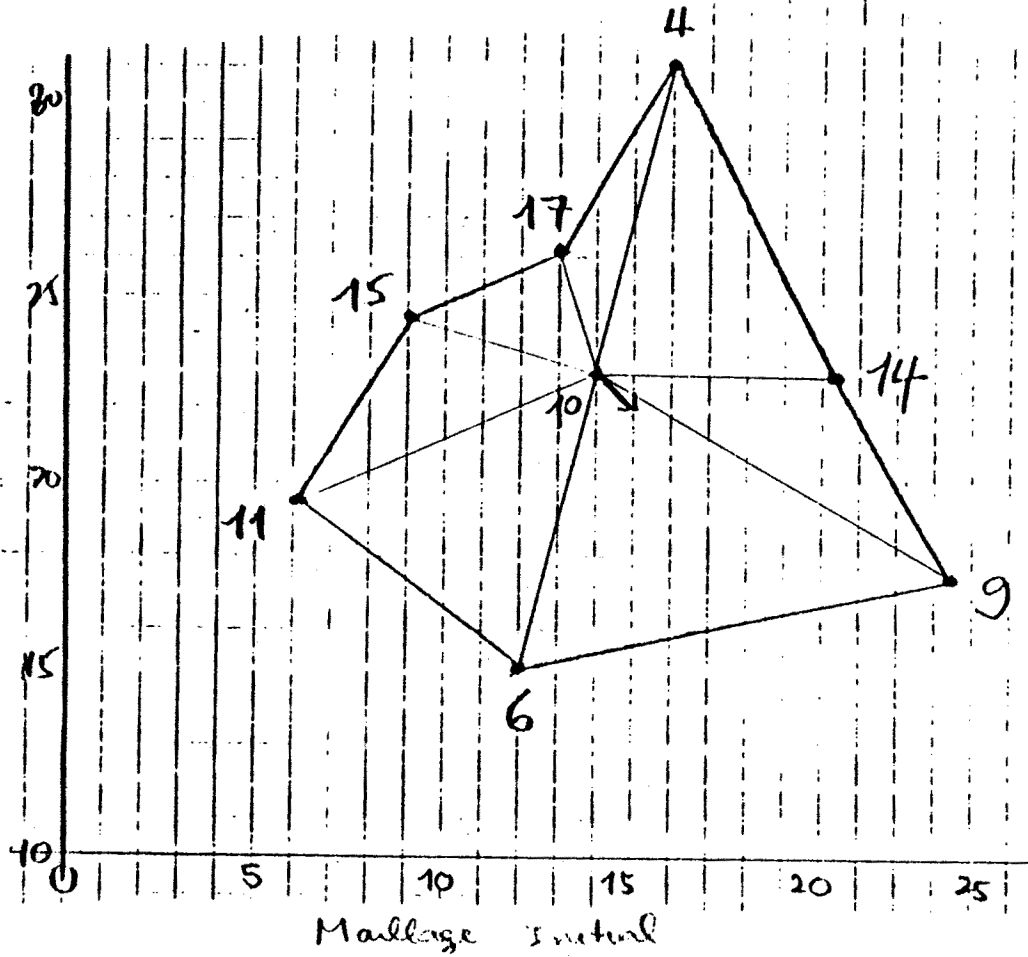


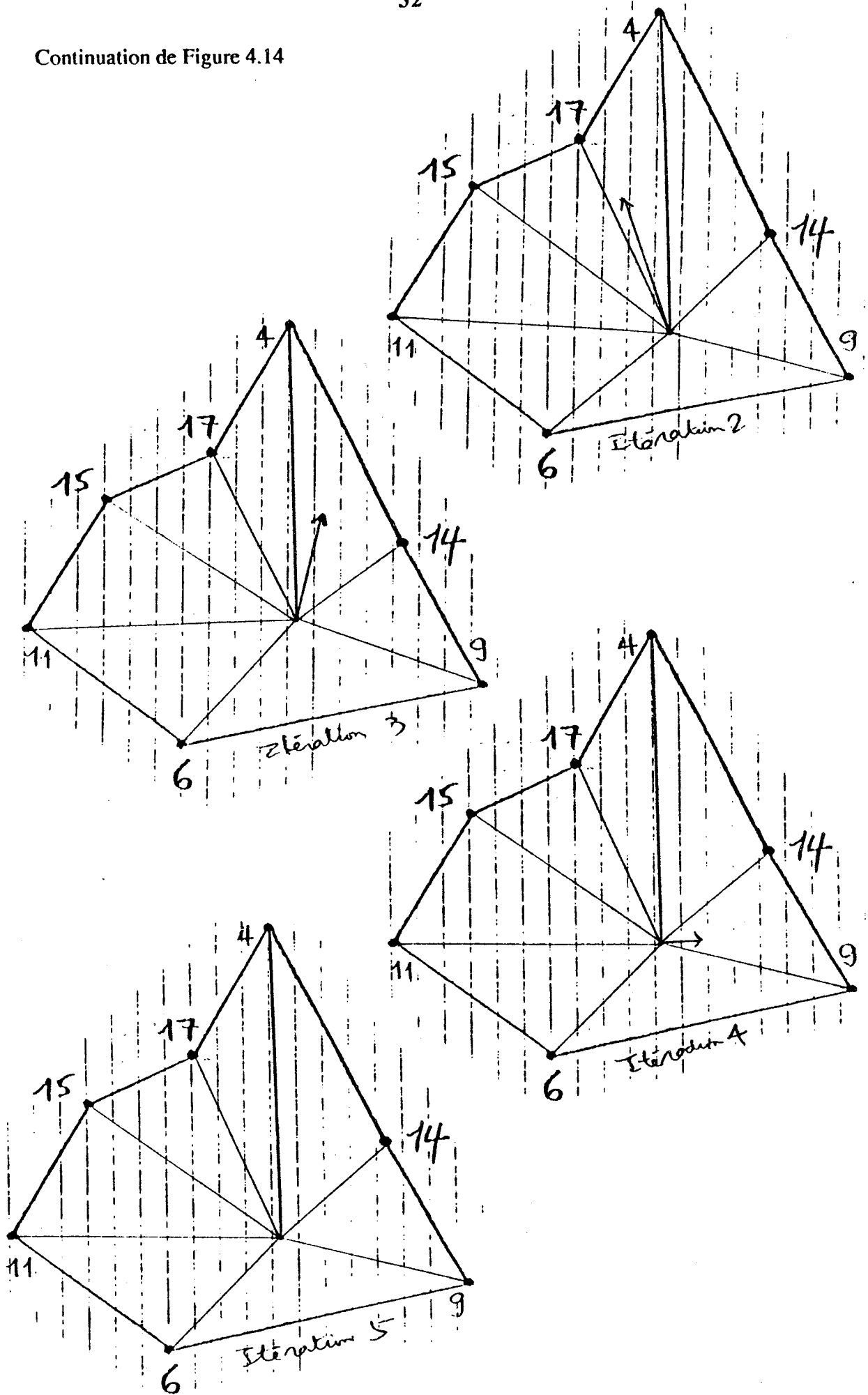
Figure 4.14 Bouger le noeud 10 avec ITERATION = 5 et la fonction objective

$$C_2(P_0) = 1 - (V_{\min}/V_{\max}) \quad (\text{pour les détails informatiques, cf Annexe 5.2})$$

(Les fleches indiquent la direction d'amélioration pour l'itération suivante)



Continuation de Figure 4.14

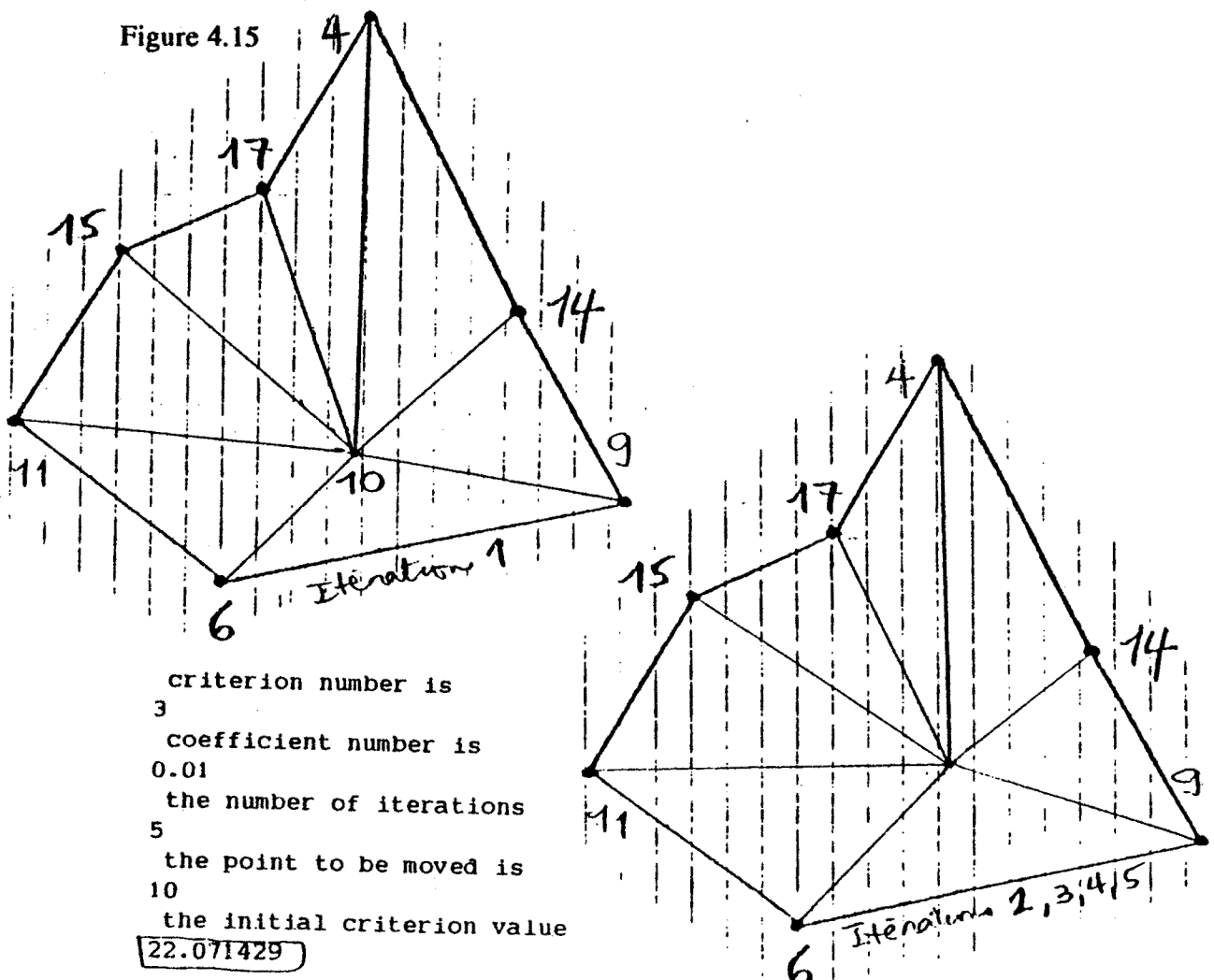


Voici les valeurs de la fonction objective respectivement à chaque itération:

Pour bouger le noeud 6 - initiale:	8,0	et le noeud 10 :	0,85
itération 1:	1,95		: 0,4
2:	1,92		: 0,32
3:	1,912		: 0,30
4:	1,906		: 0,284
5:	1,898		: 0,280

Dans tous les deux cas, il y a une très forte amélioration à la suite de la première itération, et de très faibles améliorations pour les quatre autres. On peut voir ceci sur les figures 4.13 et 4.14 par un très faible déplacement du noeud à bouger pour les 4 dernières itérations.

Puisque la méthode de sous-gradient utilisée donne rapidement une bonne solution, on se permet de temps en temps une itération non améliorante. Un exemple d'un tel cas se trouve en bougeant le noeud 10 dans la figure suivante avec 5 itérations par $C_3(P_0)$ (pour les détails informatiques, cf Annexe 5.13; Figure 4.14 pour la figure initiale; Figure 4.15 pour les figures intermédiaires). Les valeurs de la fonction objective sont - Initiale : 22,07; Itération 1: 6,15; Itération 2: 6,59; Itération 3: 6,56; Itération 4: 6,54; Itération 5: 6,52. La deuxième itération n'est pas améliorante; cependant, la valeur 6,59 n'est pas très éloignée de 6,15. Donc on l'accepte et on continue de faire encore quatre itérations. Notons que la position du noeud pour les quatre dernières itérations n'a pas vraiment changé.



On désire bouger l'ensemble des noeuds internes en les bougeant un par un. Sur un exemple donné ci-dessous, on a bougé 12 noeuds internes un par un, où chaque noeud a été bougé trois fois de suite (ITERATION=3). On aperçoit qu'il y a globalement un meilleur maillage. Mais, localement, certains noeuds ne sont pas très bien placés. Ceci nous amène à réitérer ce processus plusieurs fois de suite. On se limite à un nombre de fois pas trop grand, par exemple TOUR = 2; 3; 4 ou 5. En fait, la valeur 3 nous est largement suffisante.

Voici l'exemple où on bouge les 12 noeuds internes, avec ITERATION = 3, et TOUR = 3. Le maillage initial se trouve sur la page 35 (Figure 4.16).

Les résultats sur les trois tours sont -----

TOUR 1 :

l'ordre des noeuds à bouger est 9; 10; 8; 13; 11; 6; 15; 7; 16; 12; 14; 17

Le maillage après le premier tour est celui de la page 36 (Figure 4.17).

TOUR 2 :

l'ordre des noeuds à bouger est 10; 15; 9; 11; 16; 7; 8; 13; 14; 12; 6; 17

Le maillage après le deuxième tour est celui de la page 37 (Figure 4.18).

TOUR 3 :

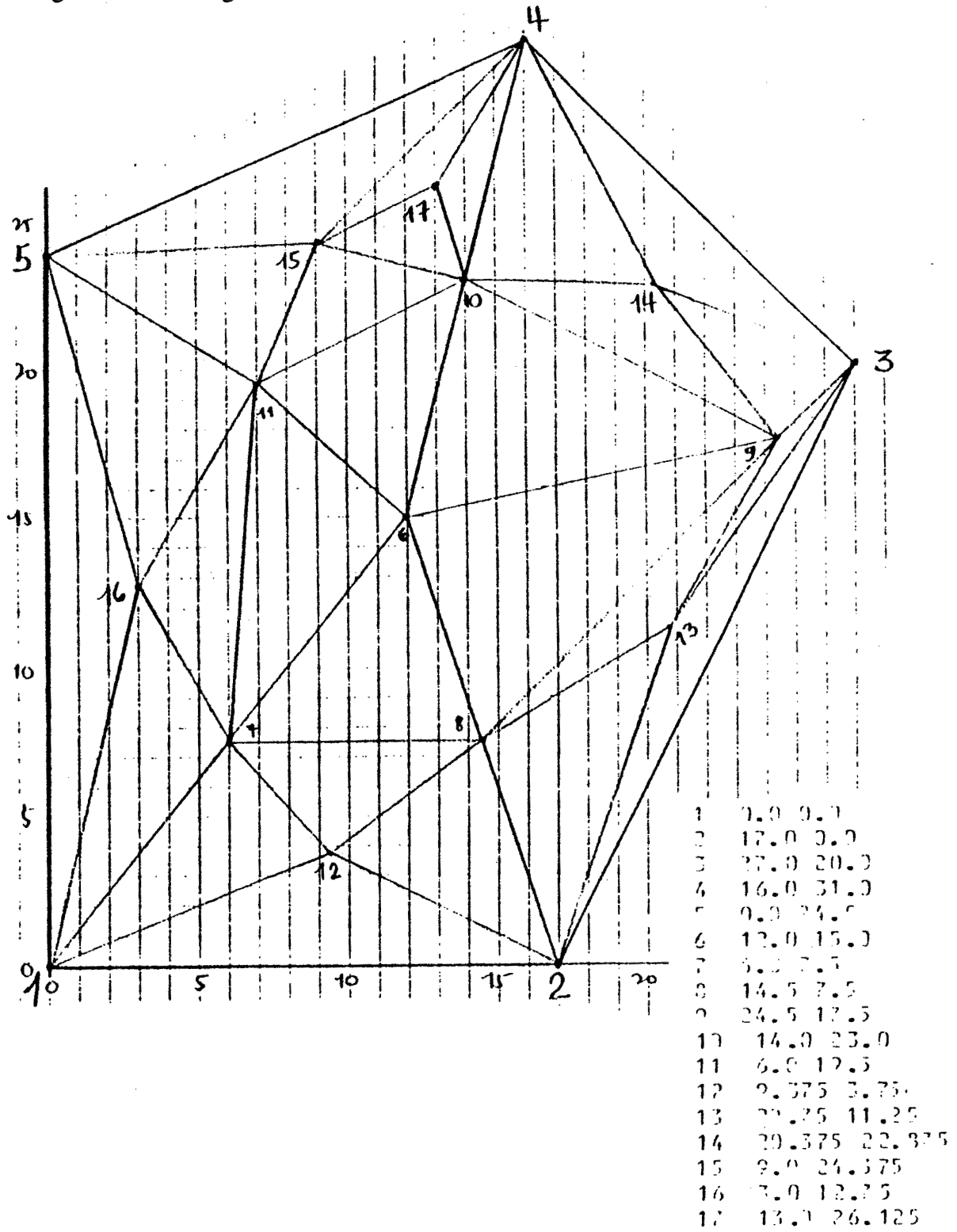
l'ordre des noeuds à bouger est 11; 15; 10; 9; 16; 14; 7; 13; 8; 12; 6; 17

Le maillage après le troisième tour est celui de la page 38 (Figure 4.19). C'est aussi le maillage final.

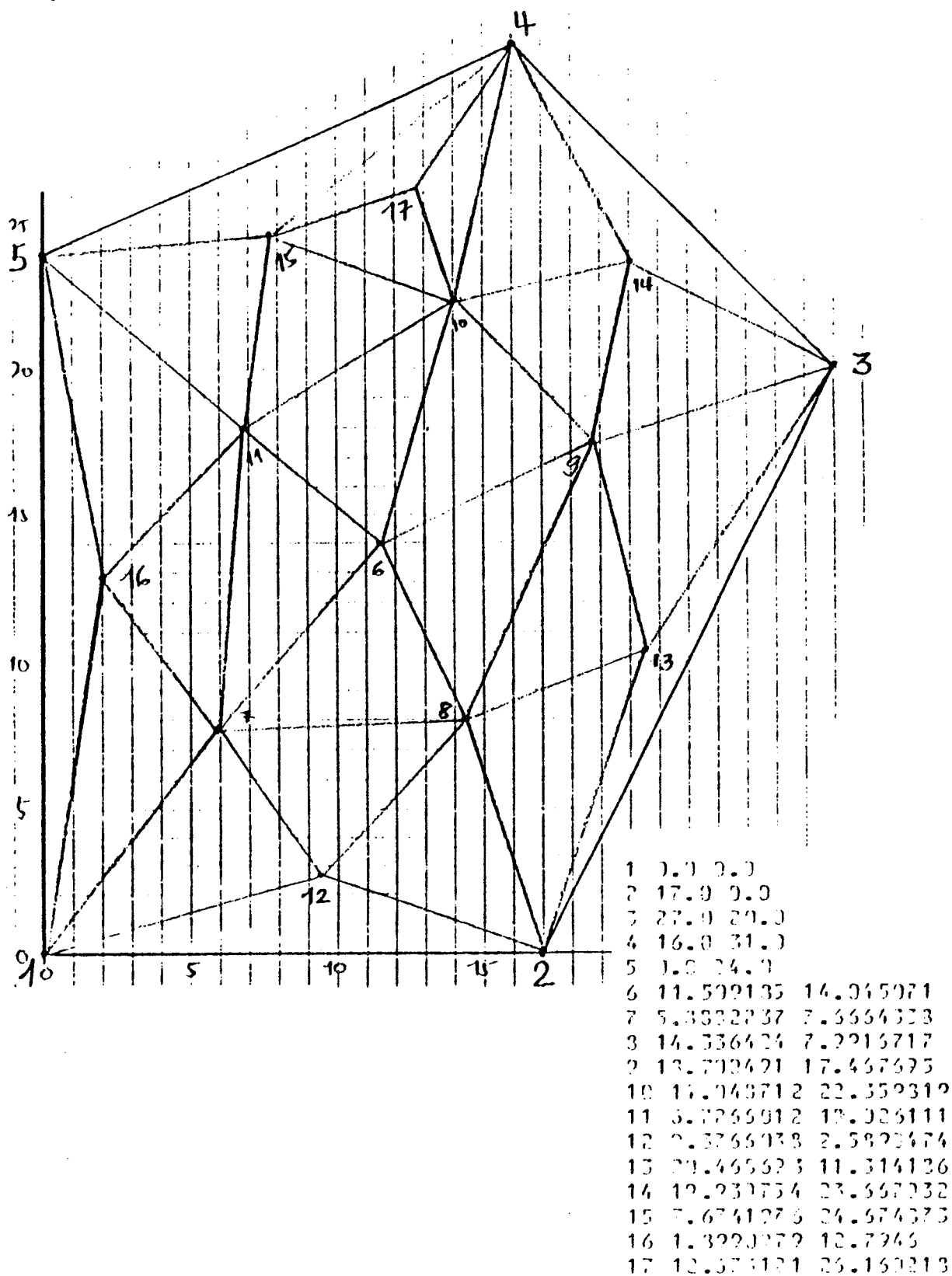
(Pour des détails informatiques, cf. Annexe 6)

Néanmoins, on peut toujours augmenter le nombre de fois qu'on va bouger un noeud successivement au cas où les améliorations résultantes des trois premières itérations ne nous satisfont pas; cette modification peut être appliquée aussi sur le nombre de tours.

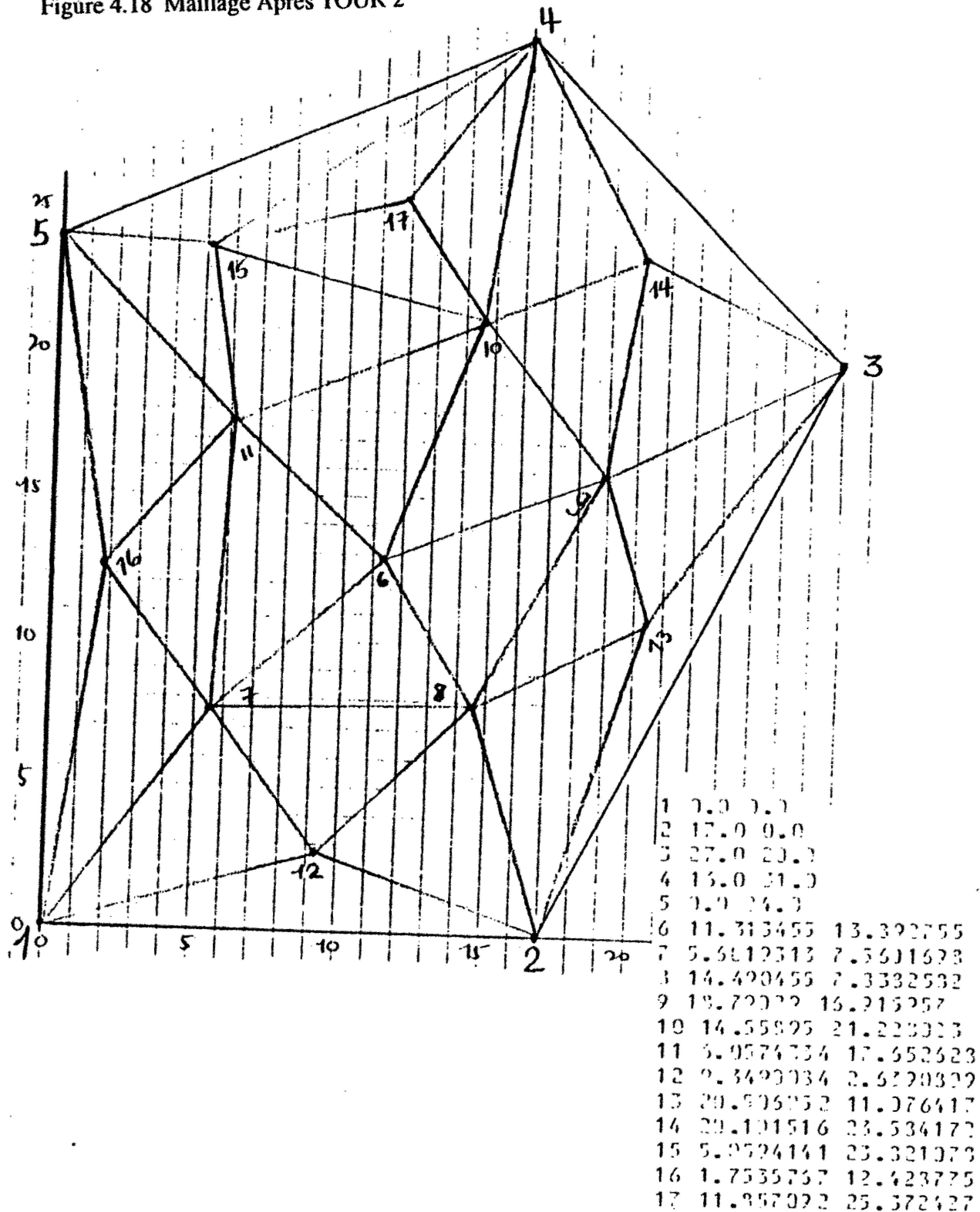
Bouger 12 noeuds avec 3 ITERATIONS par noeud et 3 TOURS pour l'ensemble de noeuds
 Figure 4.16 Maillage Initial



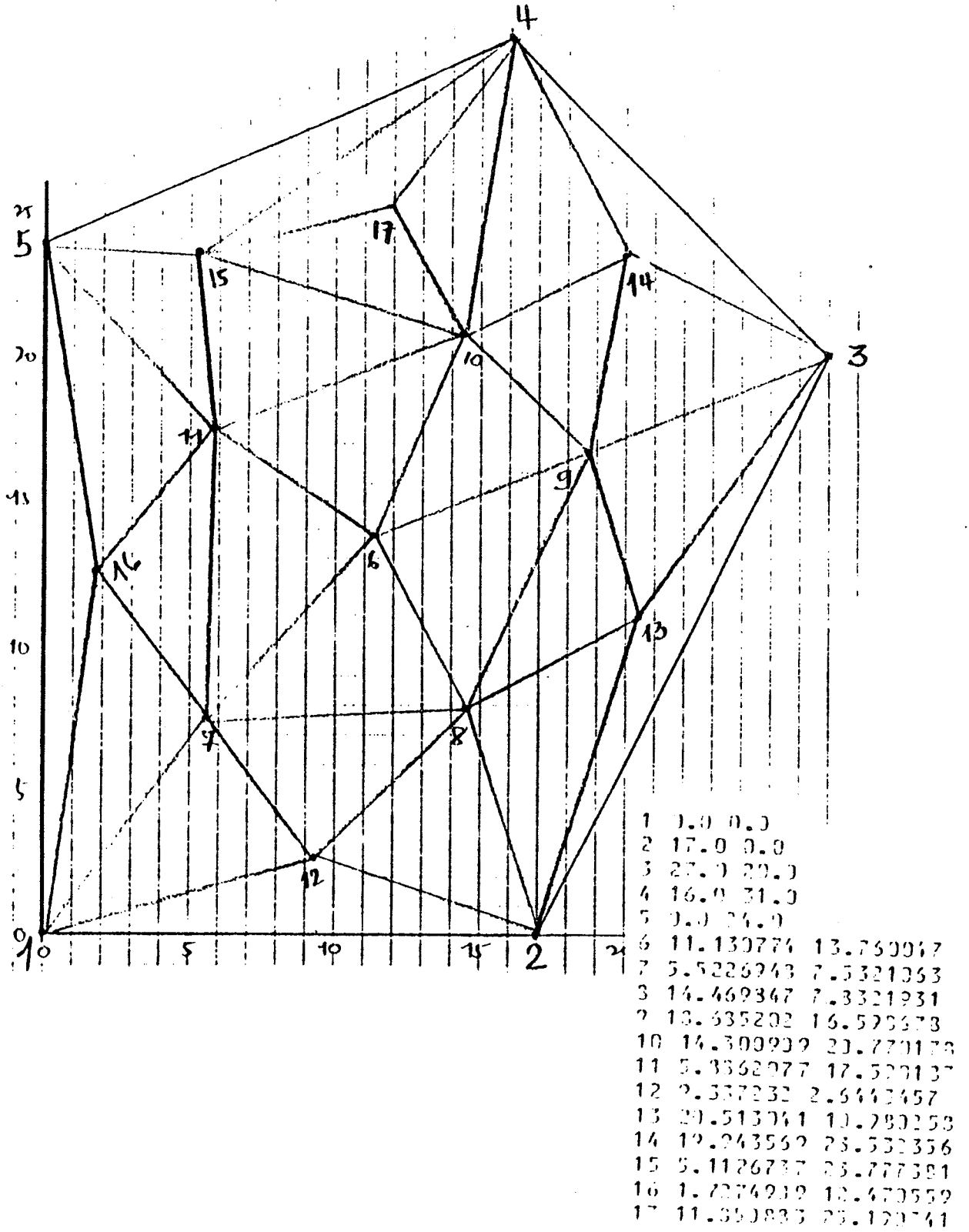
Bouger 12 noeuds avec 3 ITERATIONS par noeud et 3 TOURS pour l'ensemble de noeuds
 Figure 4.17 Maillage Après TOUR 1



Bouger 12 noeuds avec 3 ITERATIONS par noeud et 3 TOURS pour l'ensemble de noeuds
 Figure 4.18 Maillage Apres TOUR 2



Bouger 12 noeuds avec 3 ITERATIONS par noeud et 3 TOURS pour l'ensemble de noeuds
 Figure 4.19 Maillage Après TOUR 3 et Maillage Final



4.2.3 Le Choix des Noeuds à Bouger, et de l'Ordre dans Lequel Ils Seront Bougés

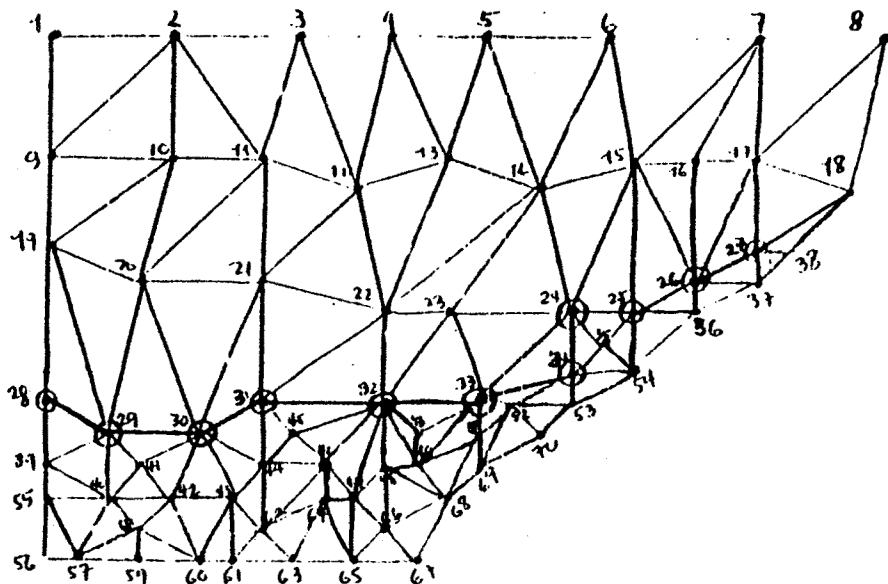
Pour un tour donné, il nous faut choisir les noeuds à bouger, et l'ordre dans lequel ils seront bougés. Si on voit qu'il y a certaines régions qui sont déjà assez bien maillées, on peut se limiter à celles qui ne le sont pas.

Si, sur une certaine région, on souhaite avoir des petits éléments bien réguliers, tandis qu'aux autres régions, on préfère avoir des éléments bien réguliers de taille plus grandes, alors on peut faire notre travail en deux étapes :

- d'abord, améliorer le maillage de l'ensemble des petits éléments, et, ensuite, l'ensemble des grands éléments.
- remailler la frontière entre ces deux ensembles pour que la différence des volumes de tétraèdres voisins ne soit pas trop grande

Voici un exemple en dimension 2, où les noeuds frontaliers (noeuds encerclés et numérotés 28; 29; 30; 31; 32; 33; 34; 24; 25; 26; 27; 18) entre les deux régions décrites ci-dessus séparent la région en deux ensembles d'éléments: les grands éléments se trouvent au-dessus de ces noeuds frontaliers, et les petits éléments se trouvent au-dessous de ces noeuds frontaliers.

Figure 4.20



Ayant bougé l'ensemble de noeuds internes de la figure 4.20 dans l'ordre décrit ci-dessus avec ITERATION = 3, et TOUR = 3, et la fonction objective $C_1(P_0) = V_{\max} - V_{\min}$, on trouve les maillages suivants: (pour les détails informatiques, cf. Annexe 7)

Figure 4.21 Le maillage sur l'ensemble de petits éléments

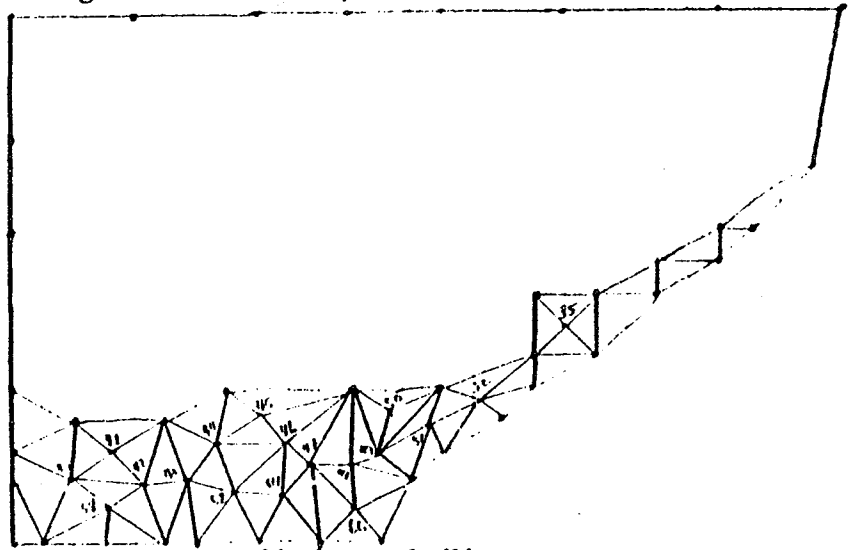


Figure 4.22 Le maillage sur l'ensemble de grands éléments

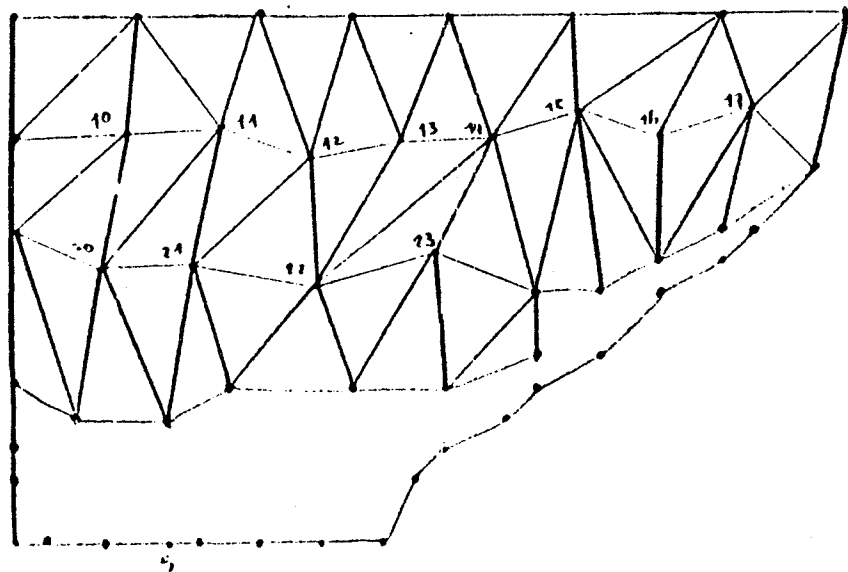
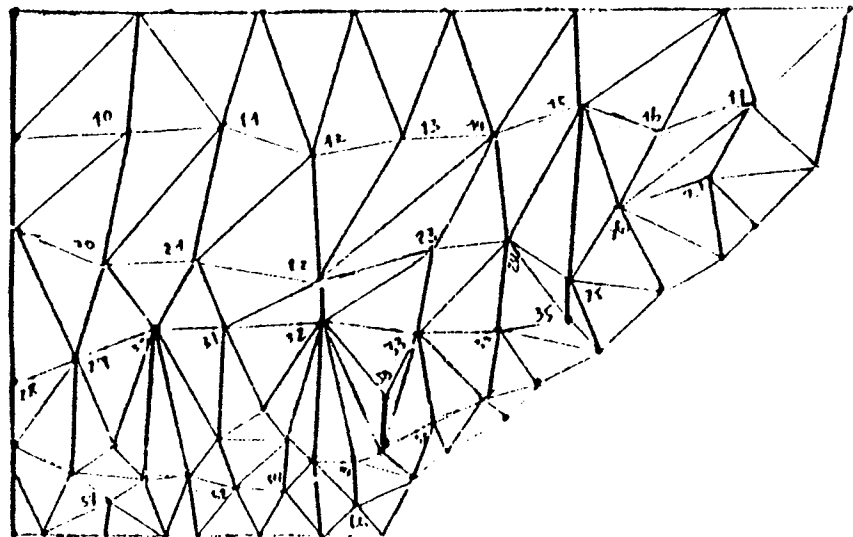


Figure 4.23 Le maillage sur l'ensemble des éléments autour des noeuds frontaliers



Des choix divers existent pour l'ordre dans lequel on va bouger un ensemble des noeuds. On peut penser à améliorer d'abord la position des noeuds de plus mauvaise qualité, selon un certain critère, bien entendu. On peut aussi penser à améliorer d'abord la position des noeuds près de la frontière de la région dont le maillage est à améliorer, et, ensuite, progressivement, aller vers l'intérieur.

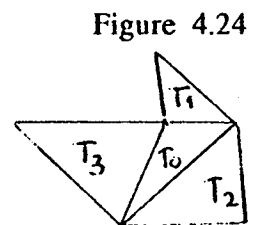
Un critère de qualité qui rend les tétraèdres les plus réguliers possible est défini ci-dessous:

En dimension 3 (en dimension 2), soit S_T = le volume (la superficie) d'un tétraèdre (triangle); S_C = le volume (superficie) de la plus grande sphère (cercle) inscrite dans le tétraèdre (triangle); $S_R = S_C/S_T$, et S_{RE} la valeur de S_R pour un tétraèdre régulier (triangle équilatéral). Notre critère de qualité est de minimiser $C_5(T_0) = \max \{ T \in \Delta \} (1 - S_R/S_{RE})$ où Δ est l'ensemble des tétraèdres (triangles) dans la région à étudier.

Puisqu'on désire bouger un ensemble des noeuds plusieurs fois, on peut choisir un ordre, par exemple, des noeuds les plus mal placés aux mieux placés, en utilisant des critères différents chaque fois qu'on veut rebouger l'ensemble des noeuds. On peut aussi prendre comme critère un mélange de ces critères. Par exemple, en prenant $C_5(T_0)$ comme critère de qualité d'un tétraèdre T_0 , et minimiser $C_6(T_0)$ comme critère de qualité de voisinage d'un élément T_0 , où $C_6(T_0)$ est défini ci-dessous:

Soit T_1, T_2, T_3, T_4 les tétraèdres voisins de T_0 (en dimension 3),
soit $V_M = \max(V_0, \dots, V_4)$ et $V_m = \min(V_0, \dots, V_4)$,
alors $C_6(T_0) = 1 - V_m/V_M$.

Dans Figure 4.24 (en dimension 2), T_1, T_2, T_3 sont les triangles voisins de T_0 , et $V_M = V_3$ et $V_m = V_1$



Un mélange de ces deux critères qui sont deux fonctions objectives à minimiser, telles que $0 \leq C_5(T_0), C_6(T_0) \leq 1$ peut être exprimé comme: $C_7(T_0) = \alpha C_5(T_0) + \beta C_6(T_0)$ à minimiser où $\alpha + \beta = 1$, et $0 \leq \alpha, \beta \leq 1$.

On peut aussi appliquer en sens inverse l'ordre de traitement précédent, par exemple, au lieu de prendre l'ordre allant du noeud le plus mal placé au noeud le mieux placé, on peut choisir d'aller du noeud le mieux placé au noeud le plus mal placé; ou, au lieu d'aller de la frontière vers l'intérieur, on peut commencer avec les noeuds à l'intérieur vers les noeuds frontaliers.

L'amélioration globale du maillage par ALGOTOUT, l'algorithme qui bouge un ensemble des noeuds, peut être définie de plusieurs façons. L'idée utilisée ici est la valeur moyenne de toutes les améliorations locales, c'est à dire l'amélioration de chaque noeud par rapport au critère utilisé. Regardons l'exemple de la page 34, ayant comme maillage initial la Figure 4.16 de la page 35. Le maillage final après un tour est la Figure 4.17 de la page 36. Le tableau ci-dessous donne le numéro de noeud, la valeur de la fonction objective (ici $V_{\max} - V_{\min}$) pour le noeud dans le maillage initial, celle dans le maillage final, et la différence de ces deux dernières, c'est à dire, la valeur initiale moins la valeur finale car on veut minimiser.

```

the points, the initial and present criterion values
at each point, and the difference of the 2 values
1 24.1875 16.765045 7.4224553
2 26.25 14.088861 12.161139
3 35.0 18.124385 16.875615
4 42.75 29.019019 13.730981
5 29.25 21.108376 8.1416245
6 43.0 3.3324785 39.667521
7 40.125 16.749521 23.375479
8 75.0 14.246999 60.753001
9 93.75 20.692186 73.057814
10 80.75 31.801026 48.948975
11 43.5 19.453426 24.046574
12 31.875 12.626416 19.248584
13 50.0 14.246998 35.753002
14 27.5 13.479509 14.020491
15 42.75 31.13227 11.61773
16 36.0 18.453603 17.546397
17 0.0 1.7889946 -1.7889946
the initial and intermediate global criterion values
and the global improvement at this moment
721.6875 424.57839 0.58831335

```

Cette différence est l'amélioration locale du noeud. Par exemple, pour le noeud 1, la valeur initiale est 24,1875; la valeur finale est 16,765045; donc l'amélioration pour le noeud 1 est 7,4224553. On aperçoit qu'il y a une amélioration locale supérieure à 50% de la valeur initiale pour les noeuds 6; 7; 8; 9; 10; 11; 12; 13; 14 et 16; à 25% pour les noeuds 1; 2; 3; 4; 5 et 15. Seul le noeud 17 a été détérioré mais de très peu. La dernière ligne donne la somme des valeurs de la deuxième et de la quatrième colonne respectivement, c'est à dire la valeur globale initiale de la fonction objective (721,6875) et la somme des améliorations locales (424,57839), ensuite elle donne le rapport du second sur le premier multiplié par 100 (58,83%) qui est notre formule pour déterminer le pourcentage d'amélioration globale. De la même façon, après deux tours, on a 59,68% d'amélioration globale (cf Figure 4.18, page 37, et le tableau correspondant, cf Annexe 6), et, après trois tours, on a 60,77% d'amélioration globale (cf Figure 4.19, page 38, et le tableau correspondant, cf Annexe 6).

5. RESULTATS NUMERIQUES

5.1 Résultats sur Quatre Cas d'Ecole et Trois Cas Réels

5.1.1 Quelques Résultats en dimension 2 et en dimension 3 sur l'Ordinateur HB-68 (Système d'Exploitation Multics)

Voici les données communes à l'ensemble des exemples de dimension 2 et de dimension 3 sur lesquels on a appliqué ALGOTOUT, l'algorithme qui bouge un ensemble des noeuds :

- chaque noeud est bougé successivement par un certain nombre d'itérations et ce nombre d'itérations est ITERATION; ici, ITERATION=3
- le processus de bouger un ensemble des noeuds est réitéré un certain nombre de fois, et ce nombre est TOUR; ici TOUR=3
- la valeur pour ϵ utilisée pour déterminer le pas ρ , on appelle XCOEF; ici XCOEF= 0,01
- l'ordre dans lequel on bouge les noeuds pour chaque tour est le même pour tous et il est décrit par le mot ORDRE;

Nous avons abrégé les intitulés des colonnes des tableaux dans ce chapitre de la façon suivante:

NOM	le nom de l'exemple
PTFILE	nom du fichier contenant les coordonnées initiales de chaque noeud
TRFILE	nom du fichier contenant la description de chaque élément par ses sommets
NBNF	nombre de noeuds frontaliers
NBNI	nombre de noeuds internes
NBCRI	la fonction objective utilisée, soit $C_1(P_0)$, $C_2(P_0)$, $C_3(P_0)$ (cf. page 17)
ORDRE	si on bouge d'abord les noeuds les plus mal placés, on désigne ORDRE par MB; si on bouge d'abord les noeuds les mieux placés, on désigne ORDRE par BM; si on bouge d'abord l'ensemble des noeuds contenus dans les petits éléments, ensuite l'ensemble des noeuds contenus dans les grands éléments, et, finalement, les noeuds internes frontaliers par rapport aux deux ensembles décrits ci-dessus, on désigne ORDRE par SB
POURAME	le pourcentage d'amélioration globale de la fonction objective
TGEXC	le temps global d'exécution de du programme informatique (en secondes)
TMEXC	le temps moyen d'exécution de l'ALGO1, c'est à dire le temps moyen pour bouger un seul noeud une seule fois

NEWPTF	nom du fichier contenant les coordonnées finales de chaque noeud; aussi le nom de la figure finale
PGINF	la page où se trouve la figure initiale
PGFIF	la page où se trouve la figure finale

Pour les tableaux des trois pages suivantes, vous devez consulter la liste ci-dessus. Pour trouver les figures qui correspondent à chaque maillage initial et à chaque maillage final, consultez les deux dernières colonnes.

Résultats en dimension 2

(les deux premiers cas sont des cas d'école et les cas derniers sont des cas réels proposés par la DGT/DEA, Avions Marcel Dassault/ Breguet Avion)

NOM	PTFILE	TRFILE	NBNF	NBNI	NBCRI	ORDRE	POURAME	TGEXC	NEWPTF	PGINF	PGFIF
Essai1	tpt	ttr	5	12	C1(Po)	MB	60,77	1,4984	tpt3	47	48
Essai1	tpt	ttr	5	12	C1(Po)	BM	64,38	1,47	tpt12	47	48
Essai1	tpt	ttr	5	12	C2(Po)	MB	48,47	1,4905	tpt6	47	49
Essai1	tpt	ttr	5	12	C2(Po)	BM	52,24	1,4587	tpt15	47	49
Essai1	tpt	ttr	5	12	C3(Po)	MB	53,25	1,4931	tpt9	47	50
Essai1	tpt	ttr	5	12	C3(Po)	BM	58,49	1,4795	tpt18	47	50
Essai2	sbpt	sbtr	30	40	C1(Po)	SB-MB	19,46		sbpt2	51	
étape 1	sbpt			30			25,43	4,485	sbpt1		52
étape 2	sbpt1			10			-13,48	3,5925	sbpt2		52
Essai2	sbpt	sbtr	30	40	C1(Po)	SB-BM	20,57		sbpt4	51	
étape 1	sbpt			30			21,72	4,413	sbpt3		53
étape 2	sbpt3			10			-5,31	3,554	sbpt4		53
Essai2	sbpt	sbtr	30	40	C1(Po)	MB	36,30	4,94	sbpt5	51	54
Essai2	sbpt	sbtr	30	40	C1(Po)	BM	29,73	4,89	sbpt6	51	54
Essai3	smrp	smrt	32	53	C1(Po)	MB	38,55	6,93	smrp1	55	55
Essai3	smrp	smrt	32	53	C1(Po)	BM	29,07	6,831	smrp3	55	55
Essai3	smrp	smrt	32	53	C3(Po)	MB	46,22	7,13	smrp2	55	55
Essai3	smrp	smrt	32	53	C3(Po)	BM	37,30	7,097	smrp4	55	55

NOM	PTFILE	TRFILE	NBNF	NBNI	NBCRI	ORDRE	POURAME	TGEXC	NEWPTF	PGINF	PGFIF	
Essai4	mncp	mnct	17	58	C1(Po)	MB	36,73	6,58	mncp1	56	57	
Essai4	mncp	mnct	17	58	C1(Po)	BM	34,55	6,66	mncp4	56	58	
Essai4	mncp	mnct	17	58	C2(Po)	MB	11,69	6,60	mncp2	56	57	
Essai4	mncp	mnct	17	58	C2(Po)	BM	-0,148	6,76	mncp5	56	59	
Essai4	mncp	mnct	17	58	C3(Po)	MB	38,95	6,83	mncp3	56	58	
Essai4	mncp	mnct	17	58	C3(Po)	BM	28,24	6,97	mncp6	56	59	
Essai5	patpt	patt	39	81	patsurf	C1(Po)	MB	33,54	12,08	patpt1	60	61
Essai5	patpt	patt	39	81	patsurf	C1(Po)	BM	29,12	11,81	patpt4	60	62
Essai5	patpt	patt	39	81	patsurf	C2(Po)	MB	31,61	12,02	patpt2	60	61
Essai5	patpt	patt	39	81	patsurf	C2(Po)	BM	27,97	12,08	patpt5	60	63
Essai5	patpt	patt	39	81	patsurf	C3(Po)	MB	37,09	12,43	patpt3	60	62
Essai5	patpt	patt	39	81	patsurf	C3(Po)	BM	35,54	12,27	patpt6	60	63

Résultats en dimension 3 (tous les deux sont des cas d'école)

NOM	PTFILE	TRFILE	NBNF	NBNI	FRONT	NBCRI	ORDRE	POURAME	TGEXC	NEWPTF
Essai6	dapt	datet	13	2	ptdata	C1(Po)	MB	8,57	1,4927	dapt1
Essai6	dapt	datet	13	2	ptdata	C1(Po)	BM	8,94	1,4904	dapt2
Essai6	dapt	datet	13	2	ptdata	C2(Po)	MB	14,05	1,4993	dapt3
Essai6	dapt	datet	13	2	ptdata	C2(Po)	BM	14,05	1,4858	dapt4
Essai6	dapt	datet	13	2	ptdata	C3(Po)	MB	1,58	1,5767	dapt5
Essai6	dapt	datet	13	2	ptdata	C3(Po)	BM	1,68	1,51	dapt6
Essai7	inp	intt	13	137	ptdata	C1(Po)	MB	40,13	69,99	inp1
Essai7	inp	intt	13	137	ptdata	C1(Po)	BM	36,81	69,67	inp2
Essai7	inp	intt	13	137	ptdata	C2(Po)	MB	13,55	70,57	inp3
Essai7	inp	intt	13	137	ptdata	C2(Po)	BM	9,22	70,41	inp4
Essai7	inp	intt	13	137	ptdata	C3(Po)	MB	20,15	70,34	inp5
Essai7	inp	intt	13	137	ptdata	C3(Po)	BM	19,21	70,73	inp6

Le programme informatique comporte trois grandes parties: la première partie est la partie initialisation; la deuxième partie est l'application proprement dite de l'ALGOTOUT qui bouge un ensemble des noeuds en les bougeant un par un; la troisième partie est la mise à jour des coordonnées des noeuds après chaque tour et le choix de l'ordre dans lequel les noeuds seront bougés. Voici les temps moyens d'exécution du programme informatique et de chacune de trois

parties sur chacun de nos exemples :

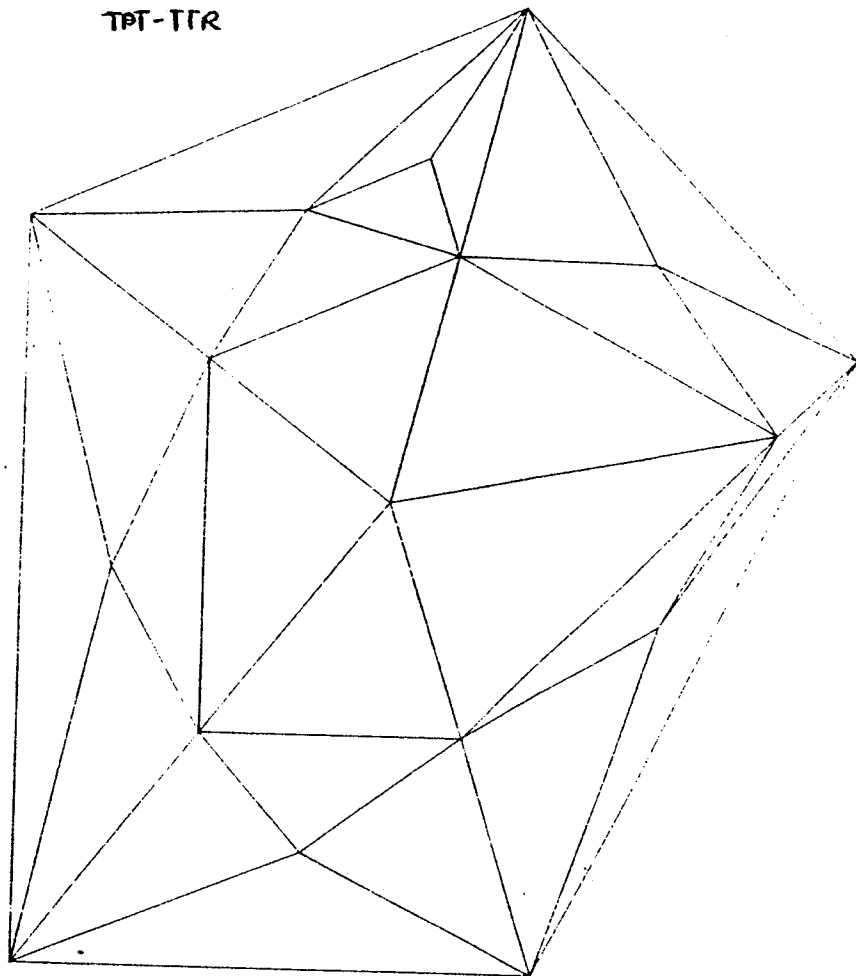
(Résultats en dimension 2)				
NOM (nombre total de noeuds; nombre total d'éléments; NBNI)	TGEXC (secondes) (TOUR=3)	1e partie (secondes)	2e partie (secondes) (TOUR=1)	3e partie (secondes) (TOUR=1)
Essai1 (17; 27; 12)	1,485	0,537	0,088	0,217
Essai2 (70; 108; 30)	4,450	1,106	0,343	0,781
Essai2 (70; 108; 10)	3,570	1,115	0,120	0,725
Essai2 (70; 108; 40)	4,915	1,117	0,407	0,900
Essai3 (85; 136; 53)	7,000	1,437	0,573	1,221
Essai4 (75; 131; 58)	6,732	1,307	0,610	1,195
Essai5 (120; 199; 81)	12,117	2,075	1,018	2,130
(Résultats en dimension 3)				
NOM (nombre total de noeuds; nombre total des éléments; NBNI)	TGEXC (secondes) (TOUR=3)	1e partie (secondes)	2e partie (secondes) (TOUR=1)	3e partie (secondes) (TOUR=1)
Essai6 (15; 25; 2)	1,5091	0,5375	0,0619	0,2547
Essai7 (150; 638; 137)	70,285	8,7051	7,4112	13,1212

Les première et troisième parties dépendent de la structure des données, du choix de la fonction critère ($C(P_0)$), du choix de l'ordre dans lequel on bouge les noeuds, et tous ceux-ci dépendent de l'utilisateur. Calculons maintenant le temps moyen pris pour bouger un noeud une seule fois sur chaque exemple.

(Résultats en dimension 2)				
NOM	NBNI	2e partie (secondes)	NBNI×CNT (où CNT=3)	TMEXC (sec/noeud/ITERATION/TOUR)
Essai1	12	0,088	36	0,002444
Essai2	30	0,343	90	0,003811
Essai2	10	0,120	30	0,004
Essai2	40	0,407	120	0,003391
Essai3	53	0,573	159	0,003603
Essai4	58	0,610	174	0,003505
Essai5	81	1,018	243	0,004189
(Résultats en dimension 3)				
NOM	NBNI	2e partie (secondes)	NBNI×CNT (où CNT=3)	TMEXC (sec/noeud/ITERATION/TOUR)
Essai6	2	0,0619	6	0,010316
Essai7	137	7,4112	411	0,018032

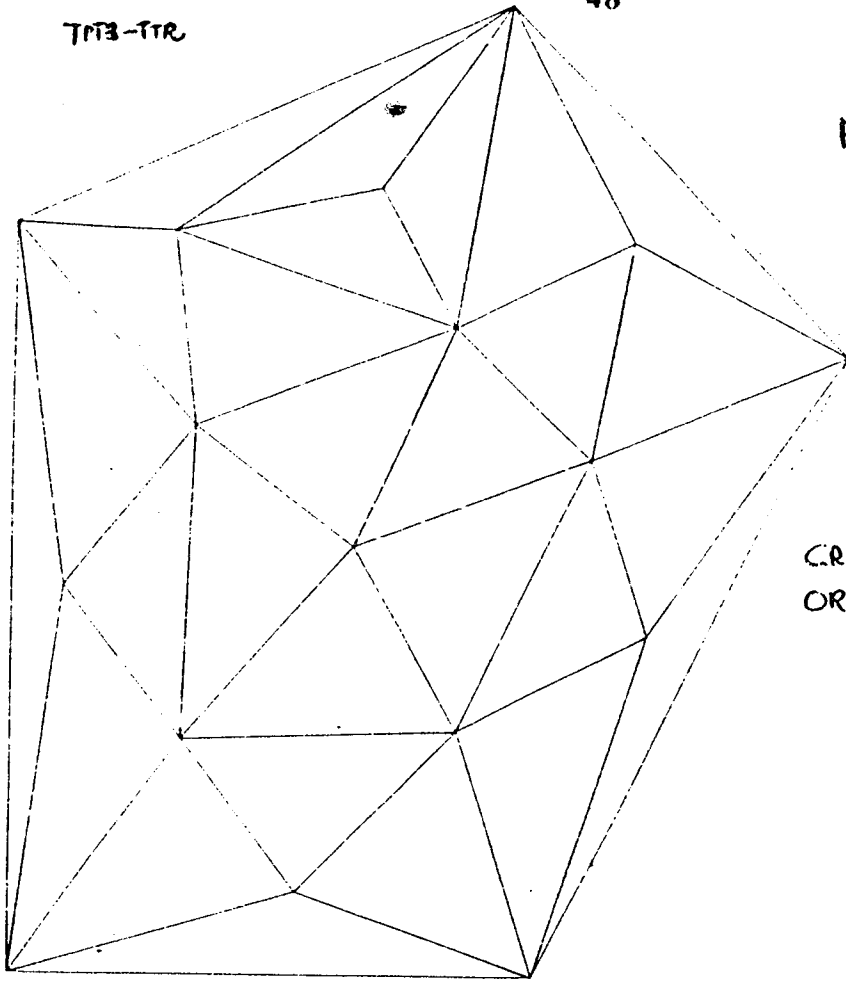
ESSAI 1
(CAS D'ECOLE)

TPT-TTR



MAILLAGE INITIAL

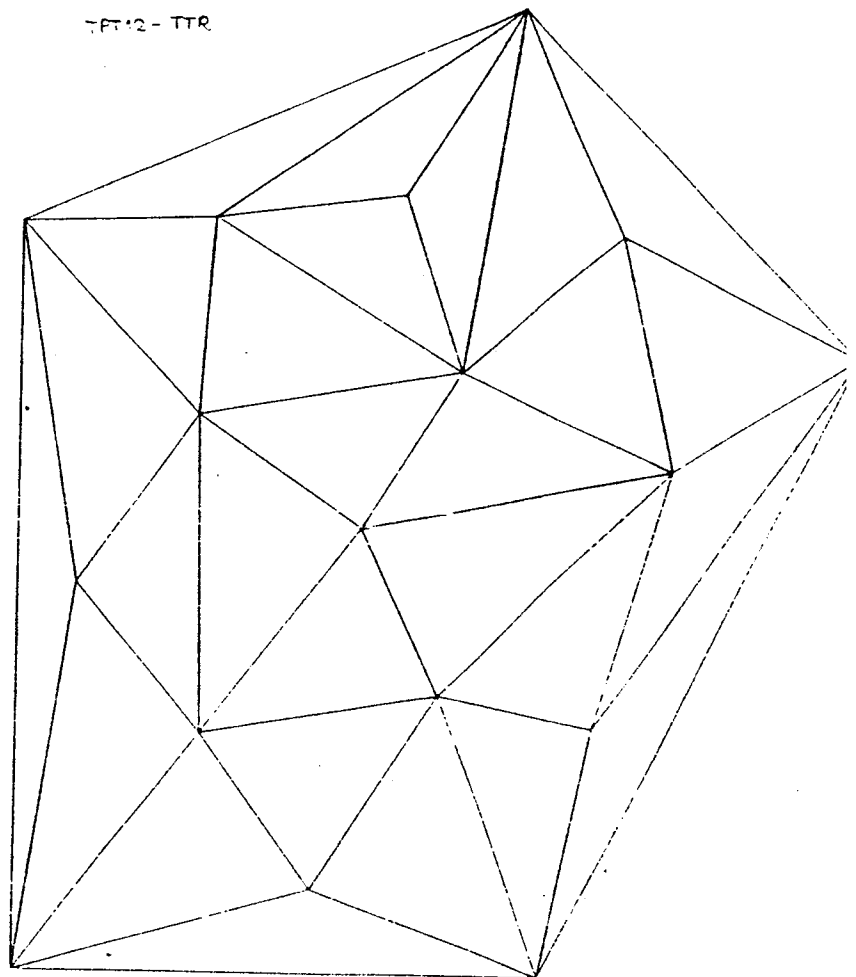
TFT3-TTR



ESSAI 1
MAILLAGES FINALS

CRITERE : $C_1(P_0)$
ORDRE : MEILLEUR AU
PIRE

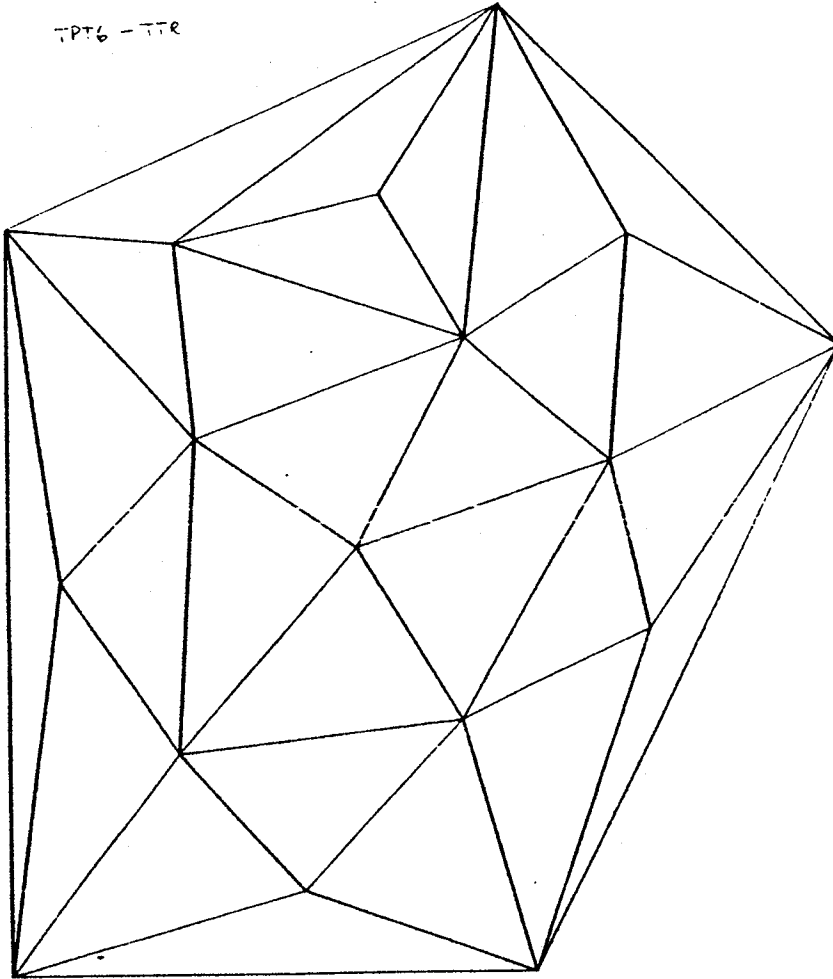
TFT12-TTR



CRITERE :
 $C_1(P_0)$
ORDRE :
PIRE AU
MEILLEUR

49

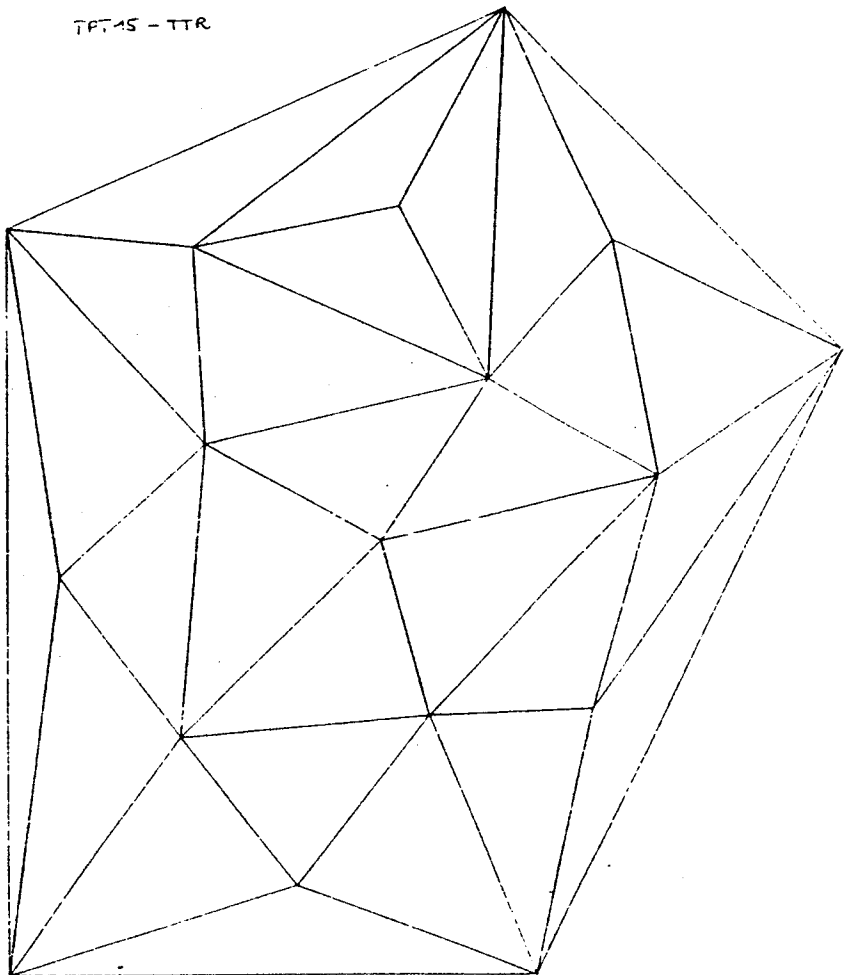
TPT6 - TTR



ESSAI 1
MAILLAGES
FINALS
CRITERE: $C_2(P_2)$

ORDRE:
MEILLEUR
AU PIRE

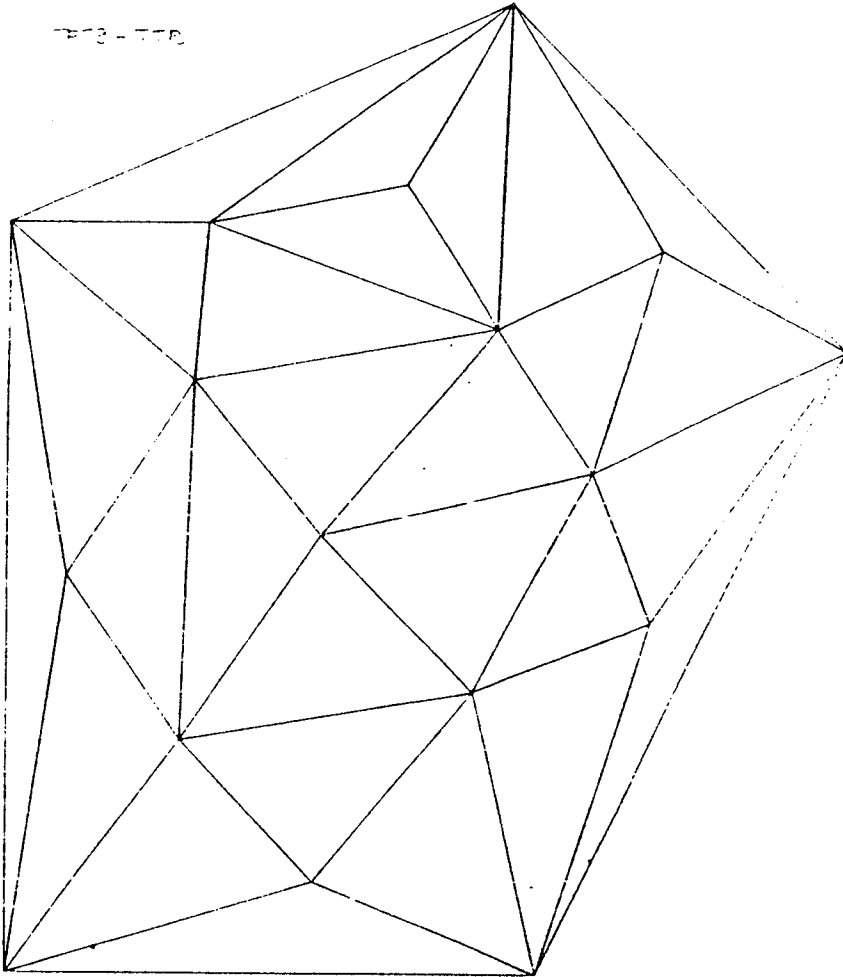
TPT45 - TTR



ORDRE:
PIRE
AU
MEILLEUR

50

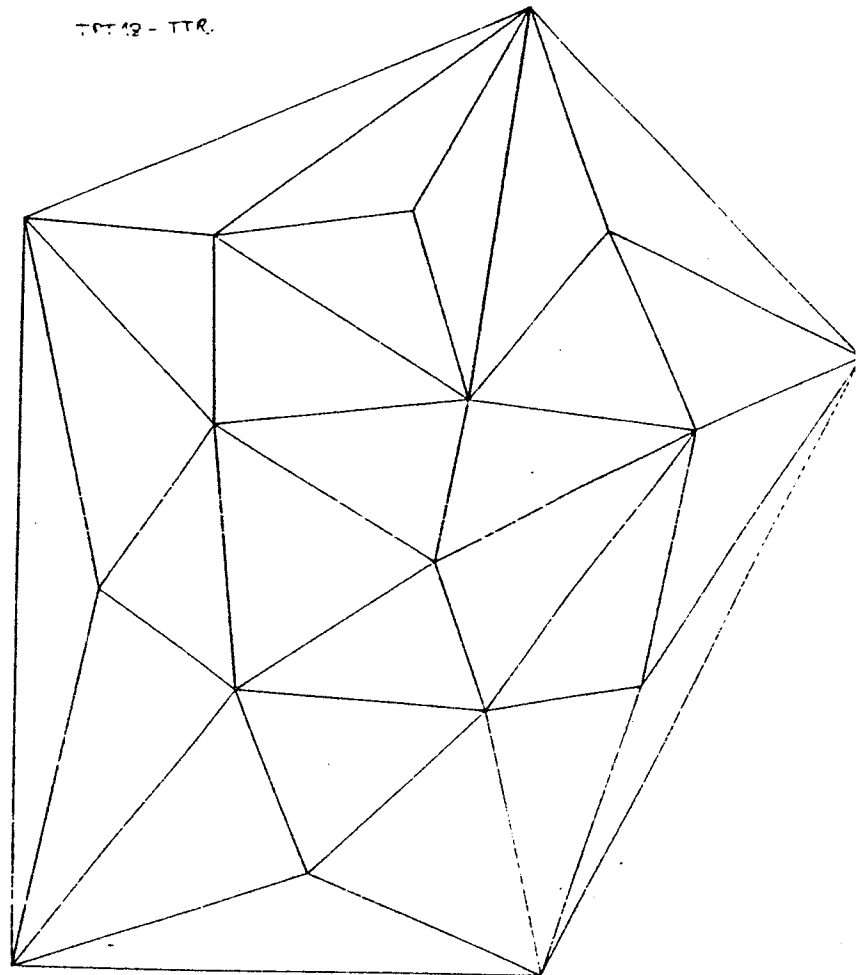
TET3 - TTR



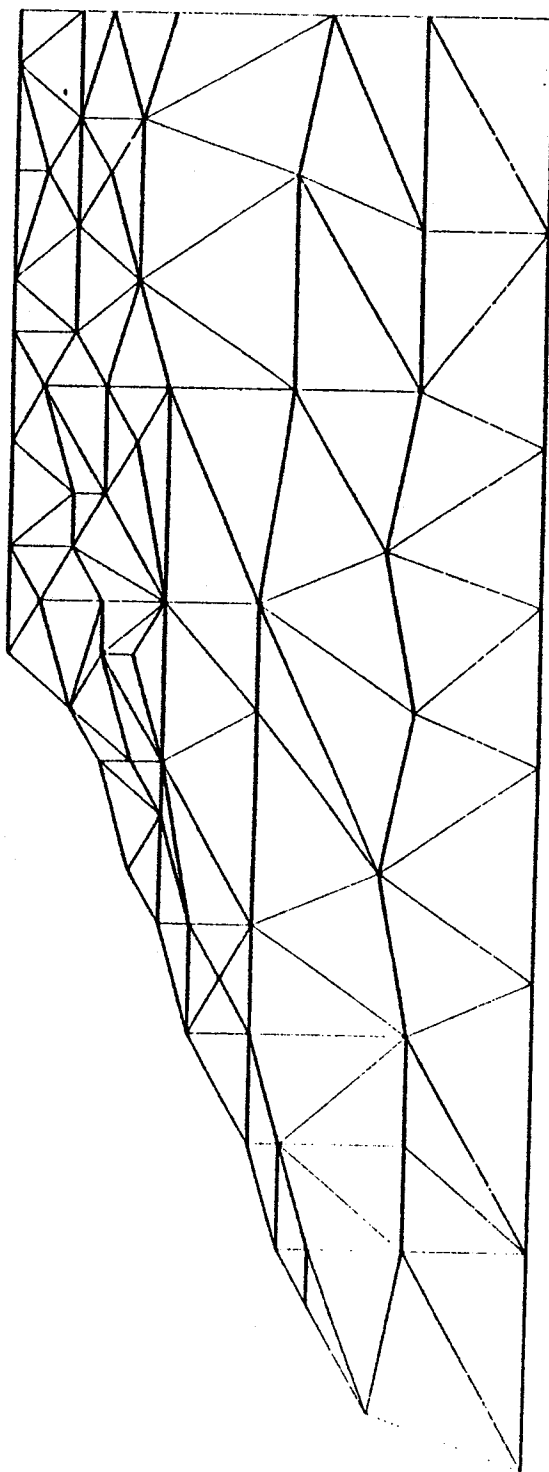
ESSAI 1
MAILLAGES FIN
CRITÈRE : $C_3(P_0)$

ORDRE :
MEILLEUR
AU PIRE

TET12 - TTR



ORDRE :
PIRE AU
MEILLEUR

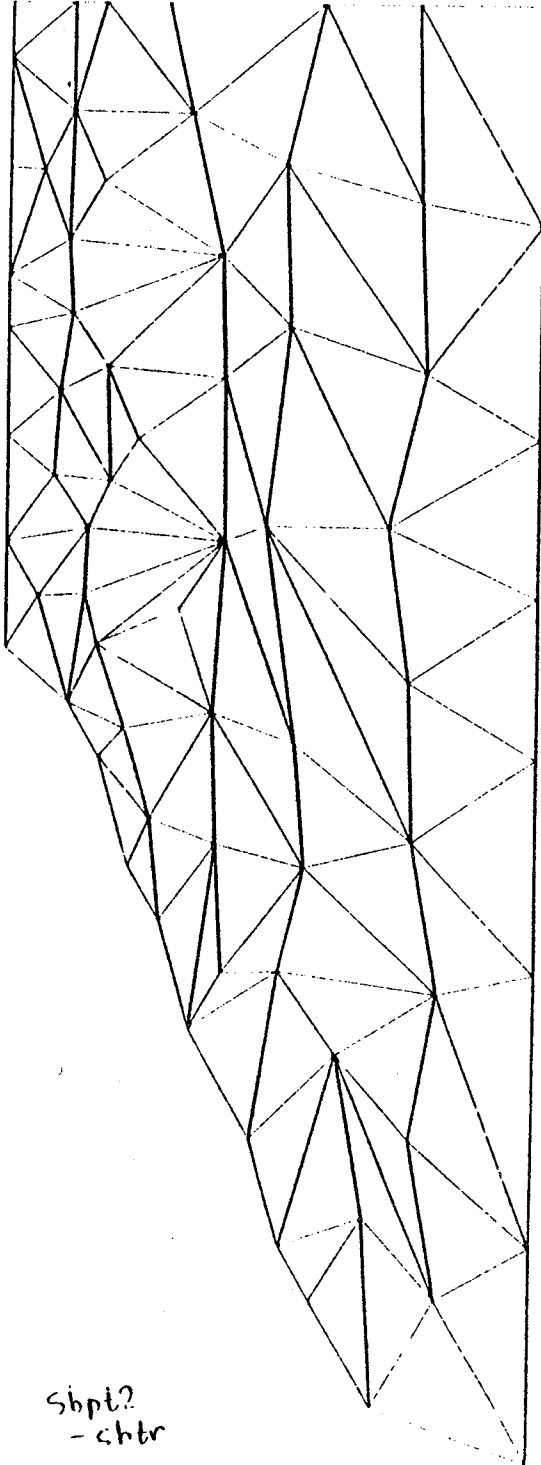


ESSAI 2
(CAS D'ECOLE)
MAILLAGE
INITIAL

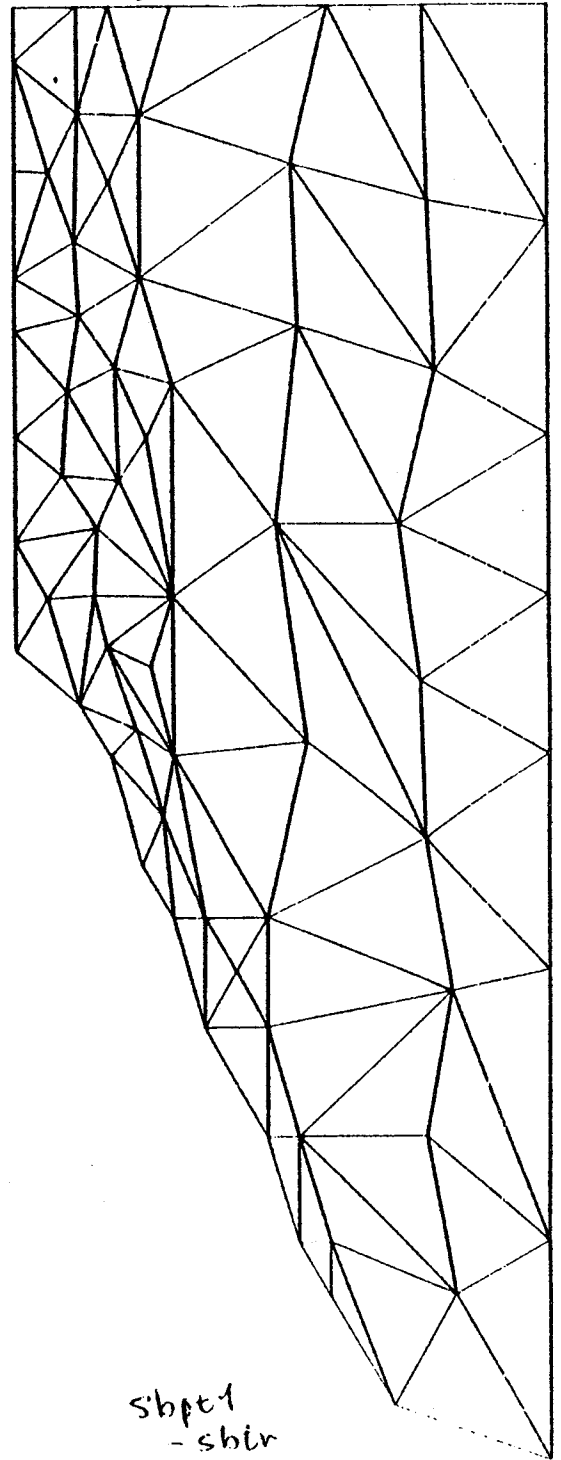
sbpt-sbtr

ESSAI 2
 CRITERE: $C_1(F_0)$
 ORDRE: MEILLEUR AU PIRE

MAILLAGE FINAL:
 - MAILLER LES NŒUDS FRONTALIERS
 ENTRE LES DEUX ENSEMBLES

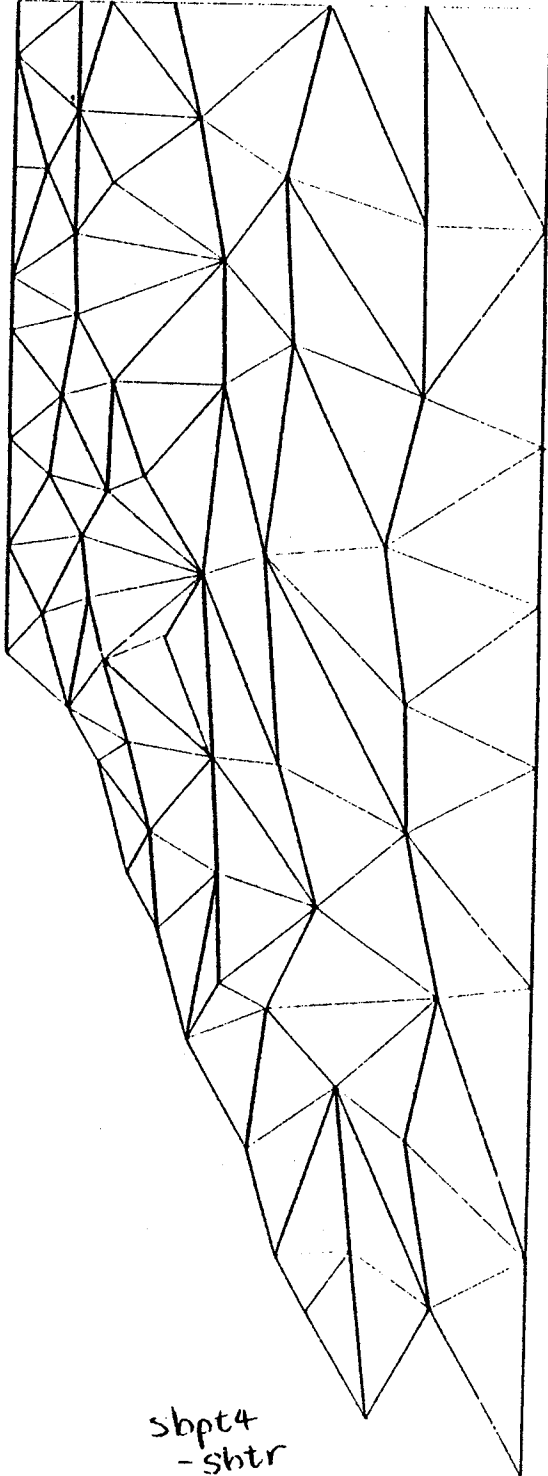


MAILLAGE INTERMÉDIAIRE:
 MAILLER SEPARÉMENT L'ENSEMBLE
 DES PETITS ÉLÉMENTS ET L'ENSEMBLE
 DES GRANDS ÉLÉMENTS

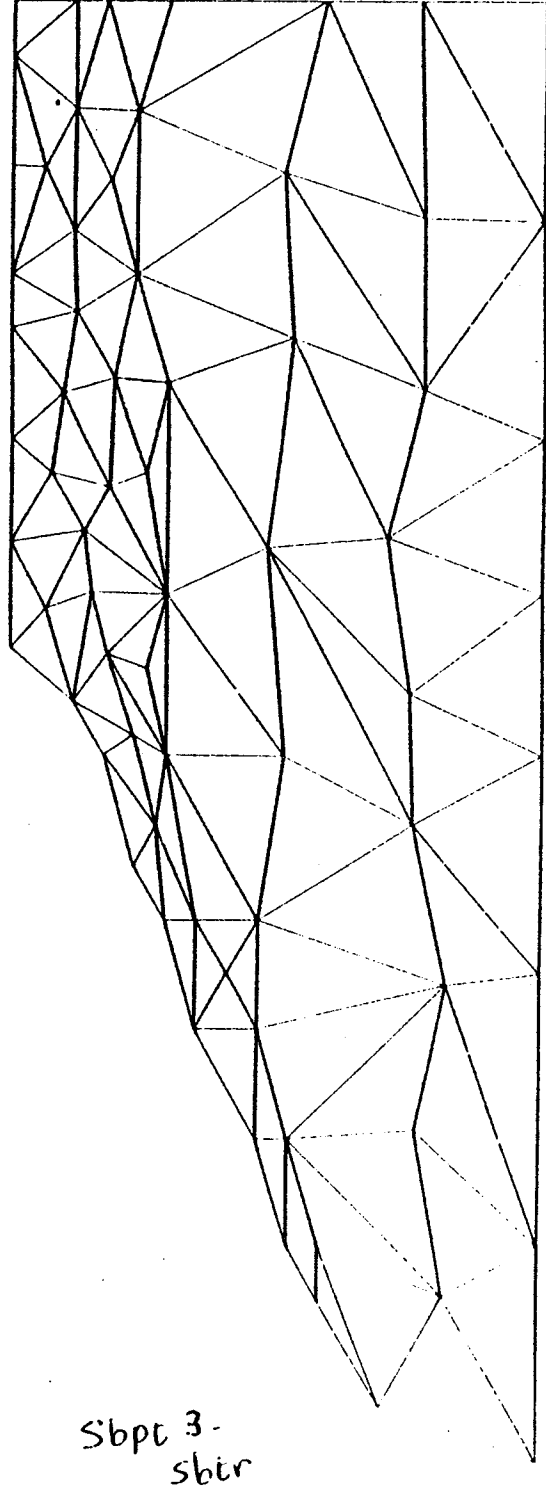


ESSAI 2
 CRITERE : $C_1(C_2)$
 ORDRE : PIRE AU MEILLEUR

MAILLAGE FINAL :
 MAILLER LES NOEUDS FRONTALIERS
 ENTRE LES DEUX ELEMENTS

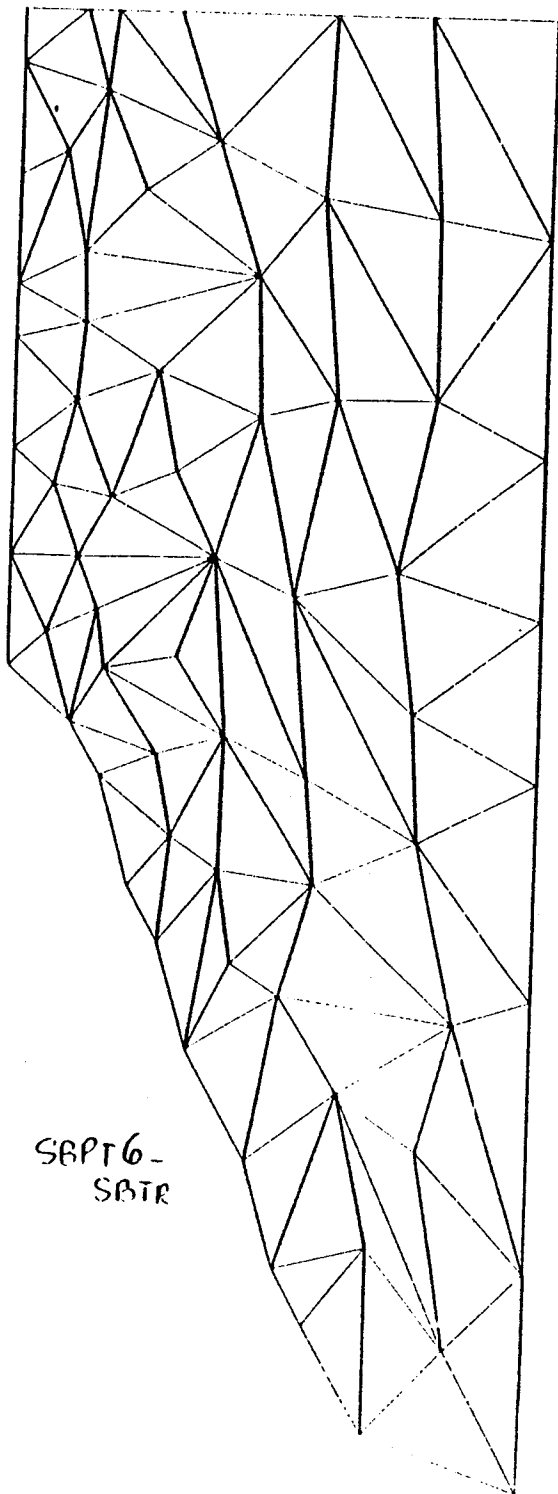


MAILLAGE INTERMEDIAIRE :
 MAILLER SEPAREMENT L'ENSEMBLE
 DES PETITS ELEMENTS ET L'ENSEMBLE
 DES GRANDS ELEMENTS



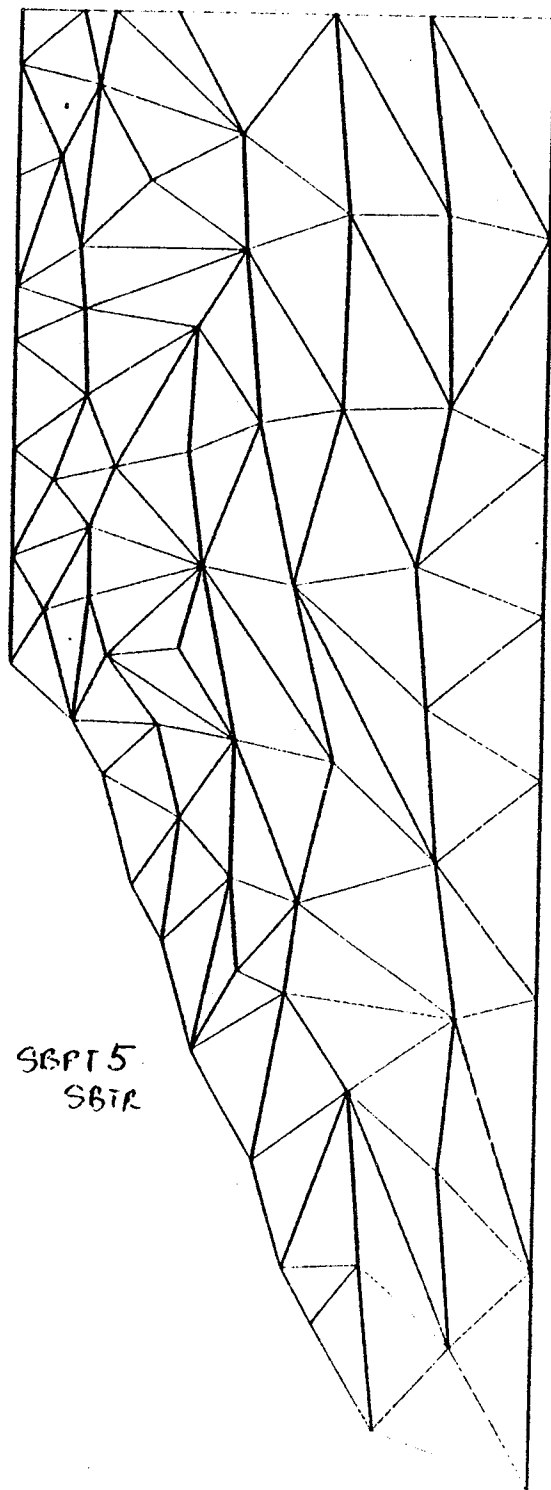
ESSAI 3
MAILLAGES FINALS
CRITERE : $C_1(P_0)$

ORDRE : PIRE AU
MEILLEUR



SBPT6-
SBTR

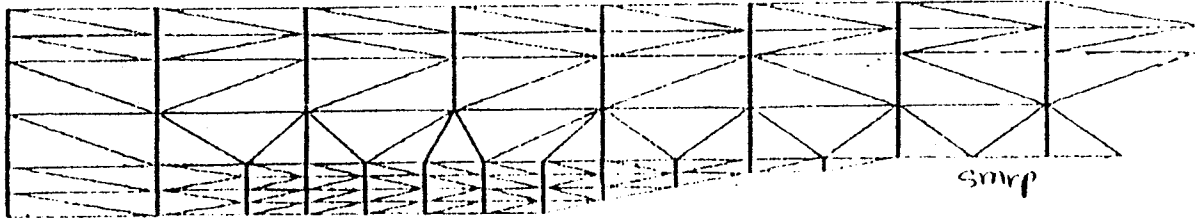
ORDRE : MEILLEUR
AU PIRE



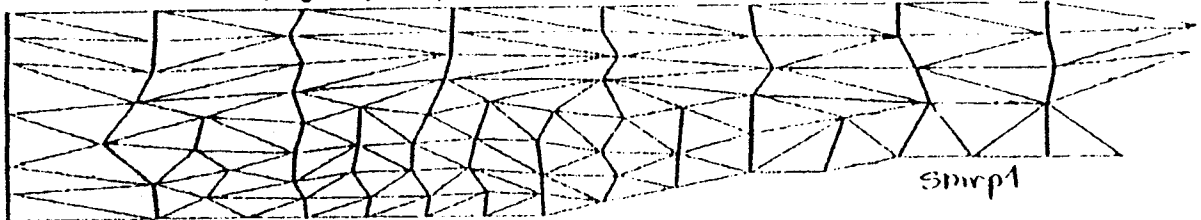
SBPT5
SBTR

ESSAI 3
(CAS REEL)

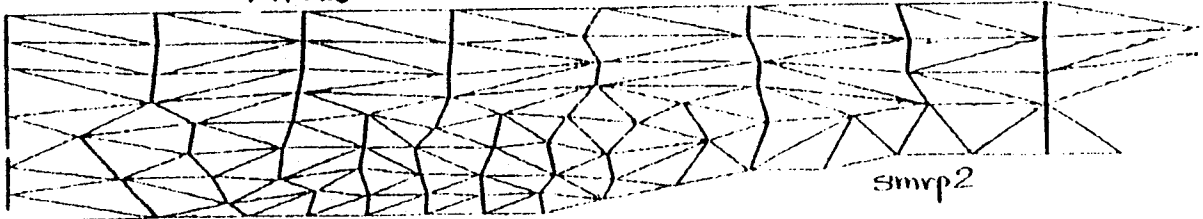
MAILLAGE INITIAL



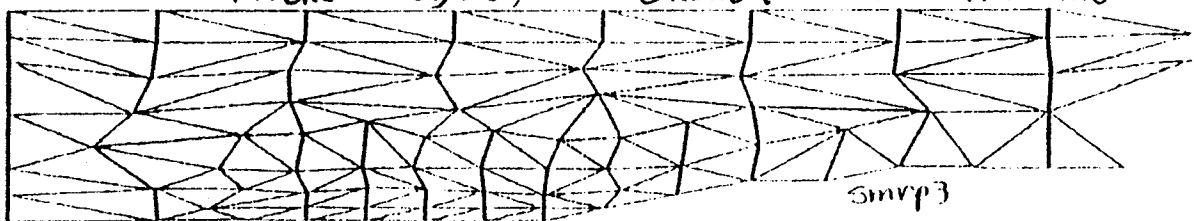
MAILLAGES FINALS CRITERE : $C_1(P_0)$ ORDRE : MEILLEUR AU PIRE



CRITERE : $C_1(P_0)$ ORDRE : PIRE AU MEILLEUR



CRITERE : $C_2(P_0)$ ORDRE : MEILLEUR AU PIRE

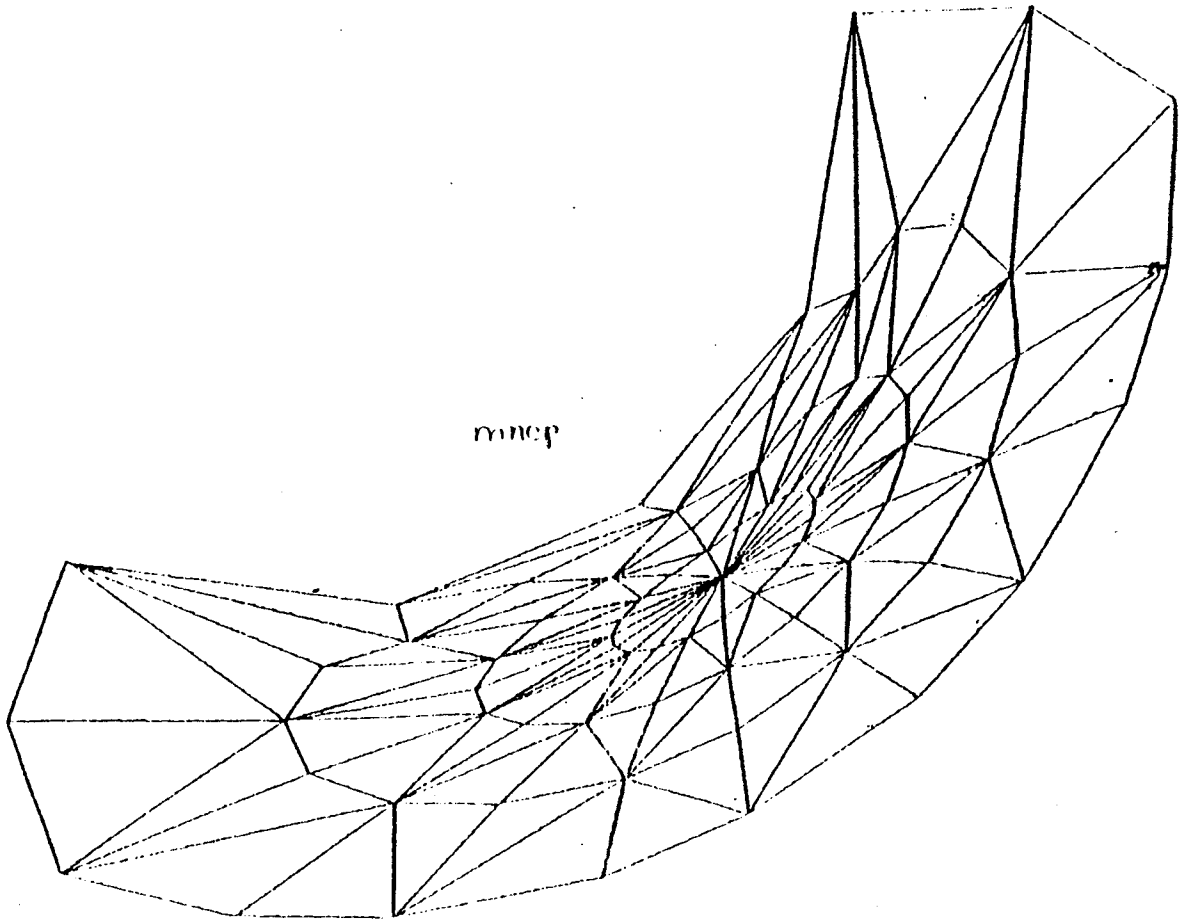


CRITERE : $C_3(P_0)$ ORDRE : PIRE AU MEILLEUR

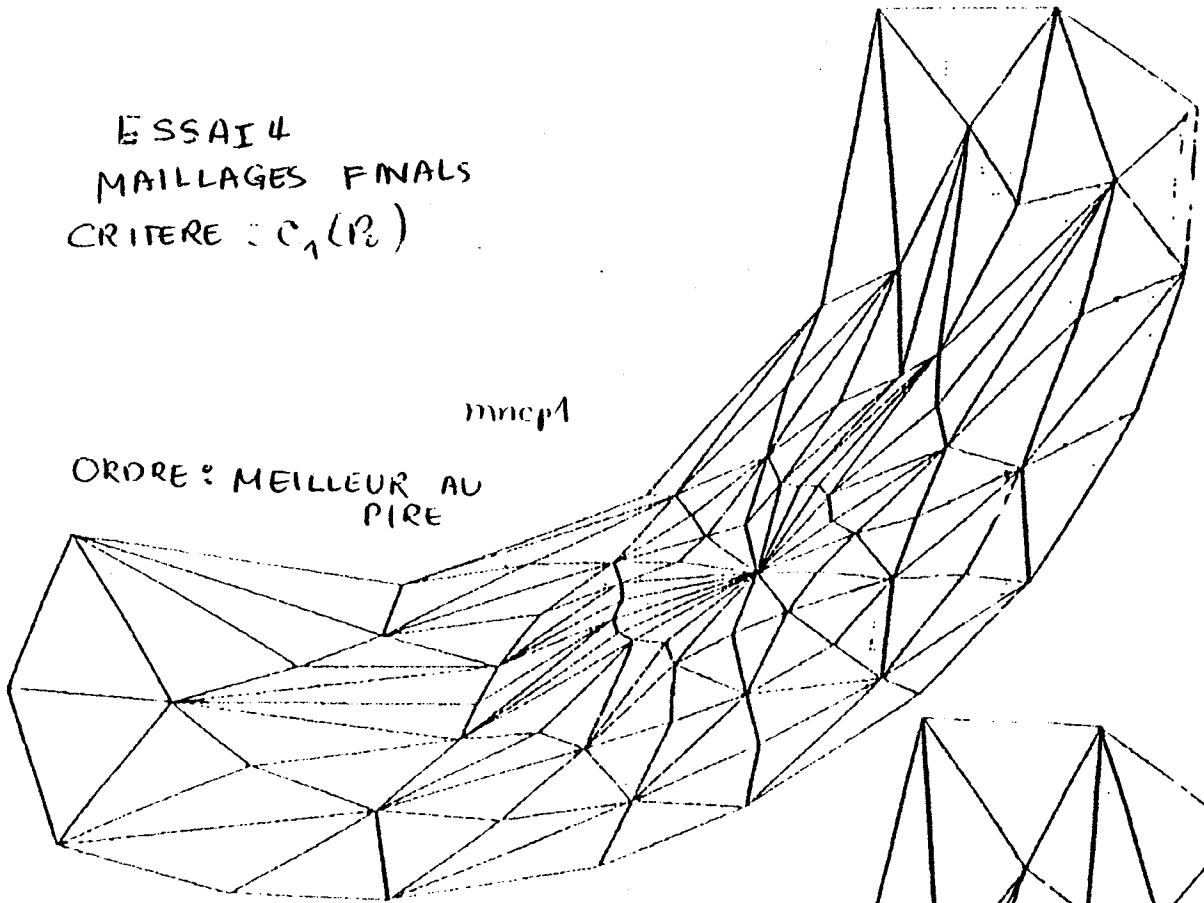


ESSAI 4
(CAS REEL)

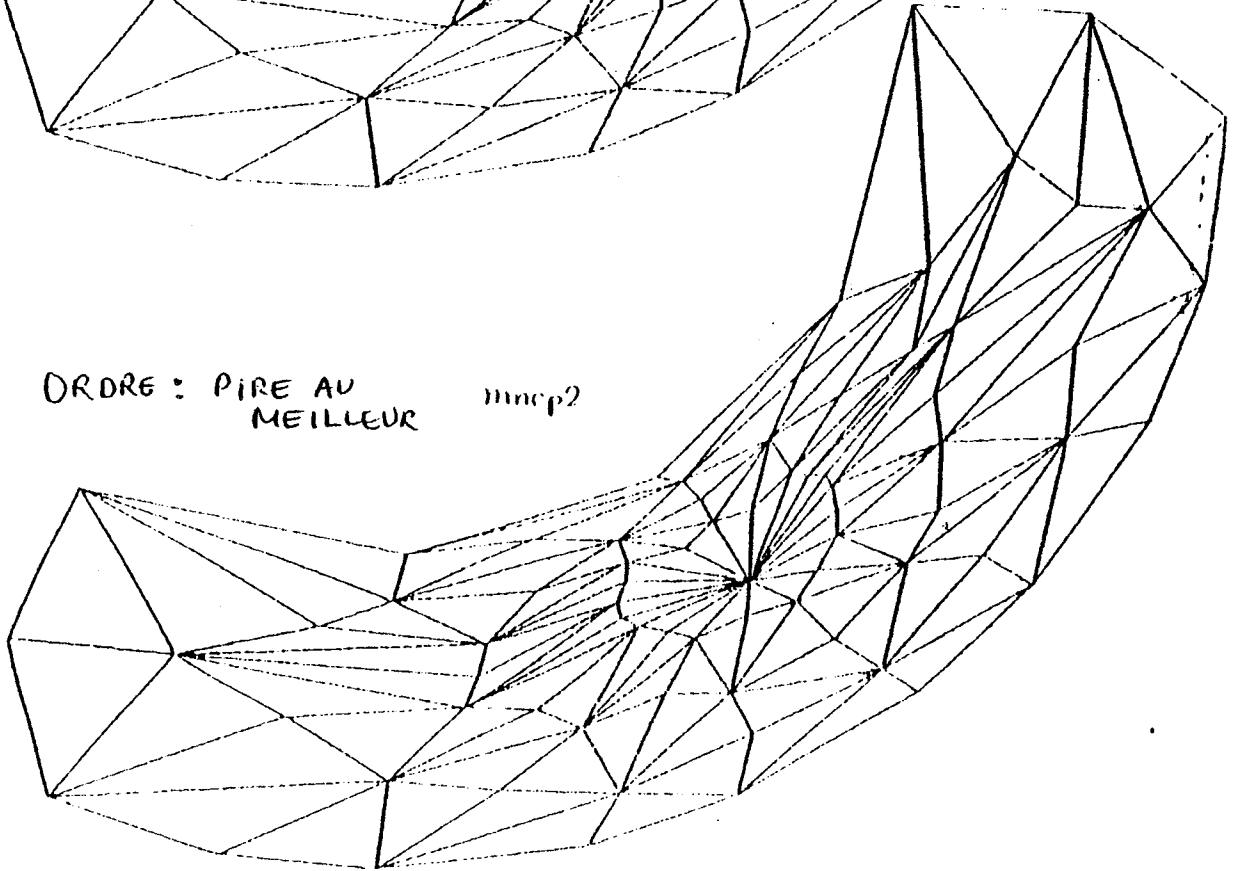
MAILLAGE INITIAL



ESSAI 4
MAILLAGES FINALS
CRITERE : $C_1(P_2)$



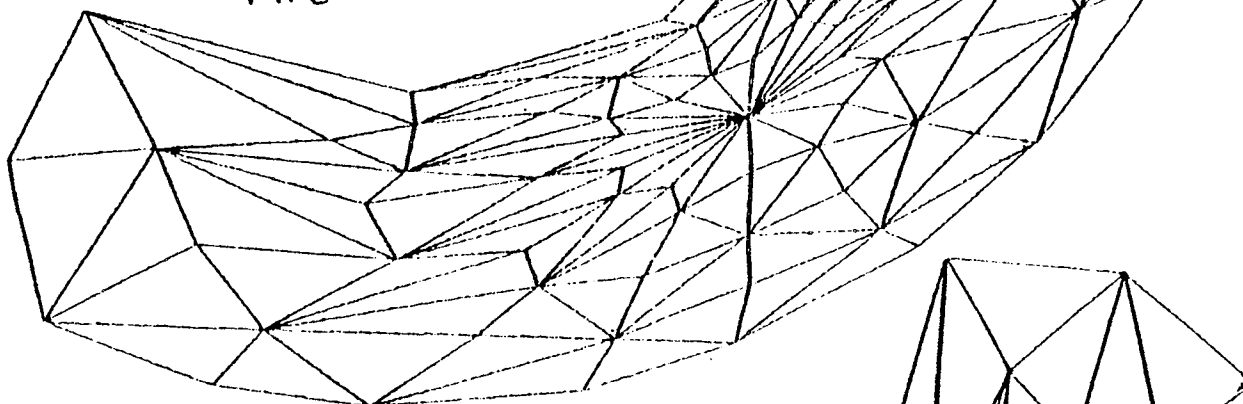
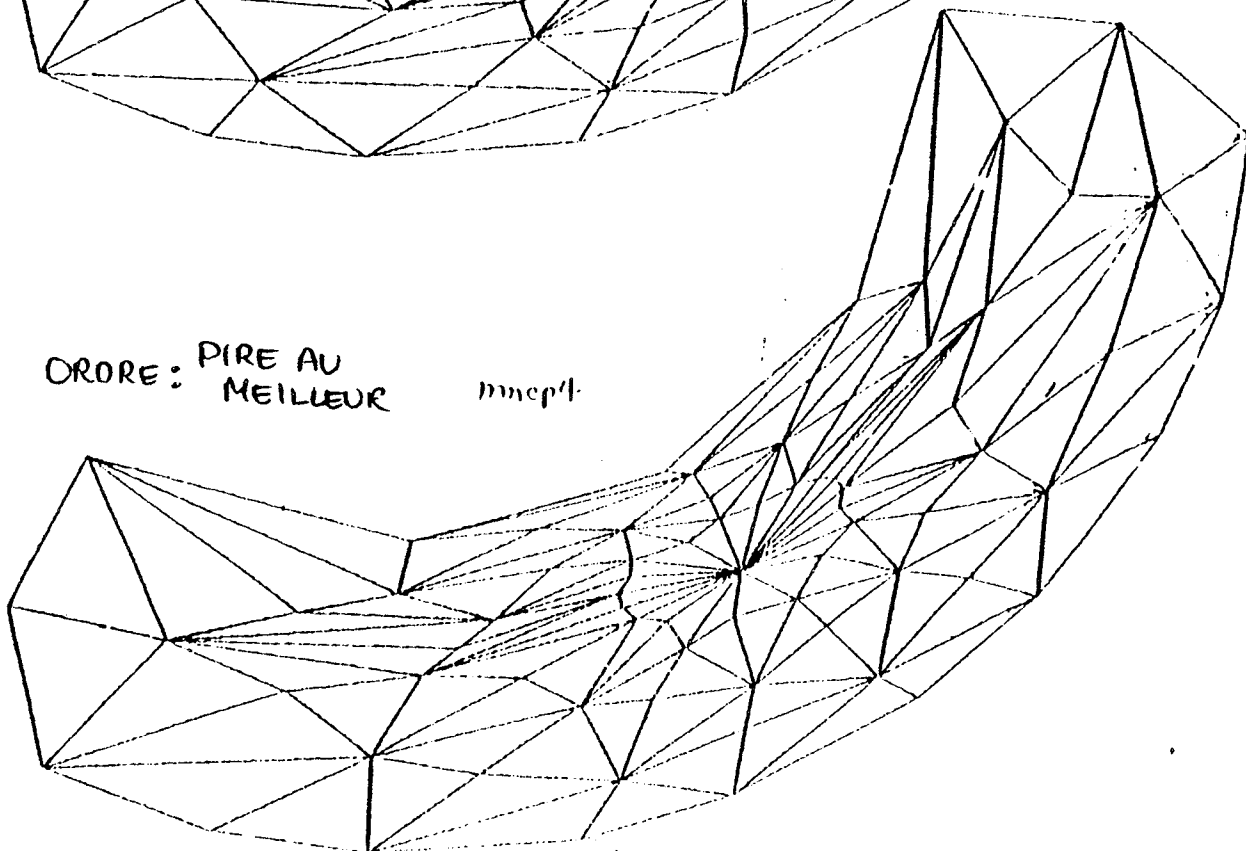
mncp1
ORDRE : MEILLEUR AU
PIRE



ORDRE : PIRE AU
MEILLEUR mncp2

ESSAI 4

MAILLAGES FINALS

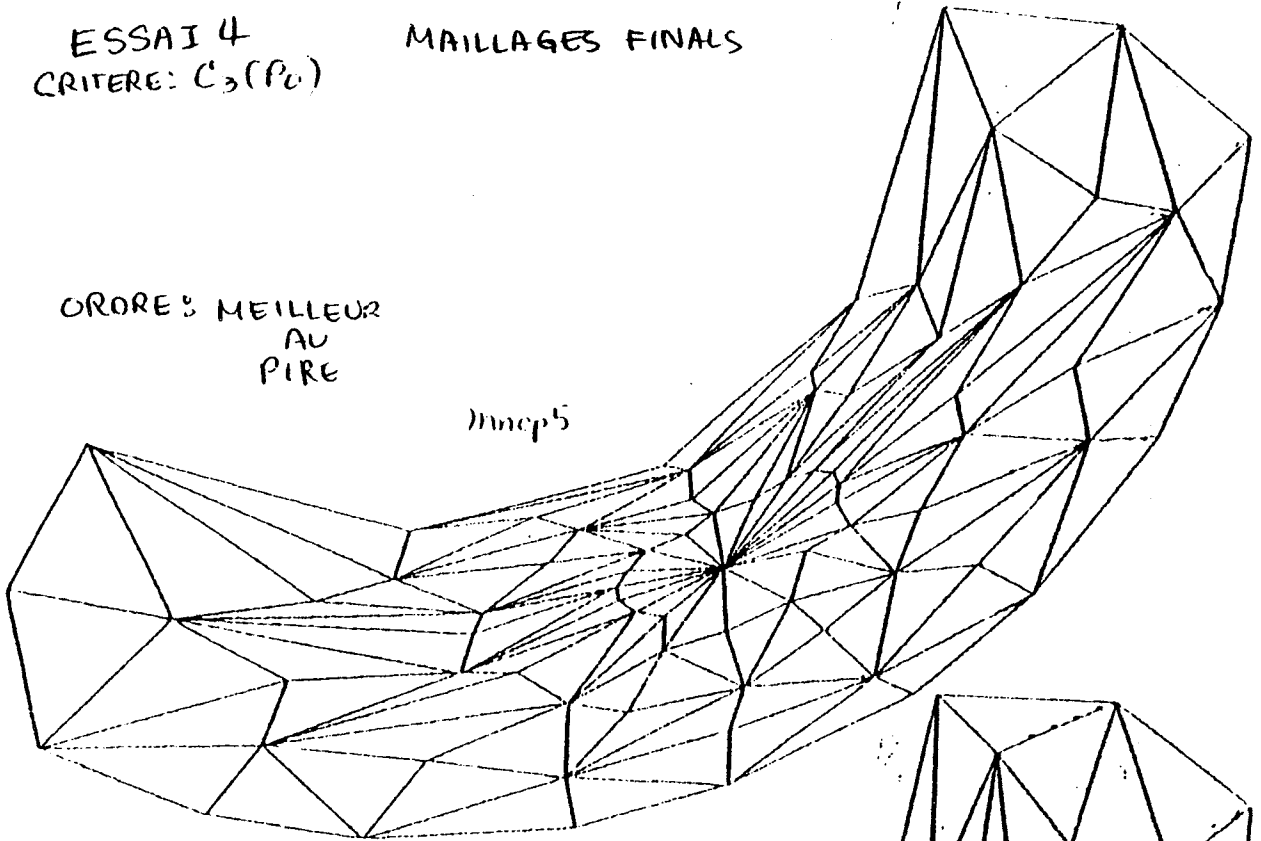
CRITERE : $C_2(P_0)$ ORDRE : MEILLEUR
AU
PIRE h_{mep^3} ORDRE : PIRE AU
MEILLEUR h_{mep^4} 

ESSAI 4
CRITERE: $C_3(P_0)$

MAILLAGES FINALS

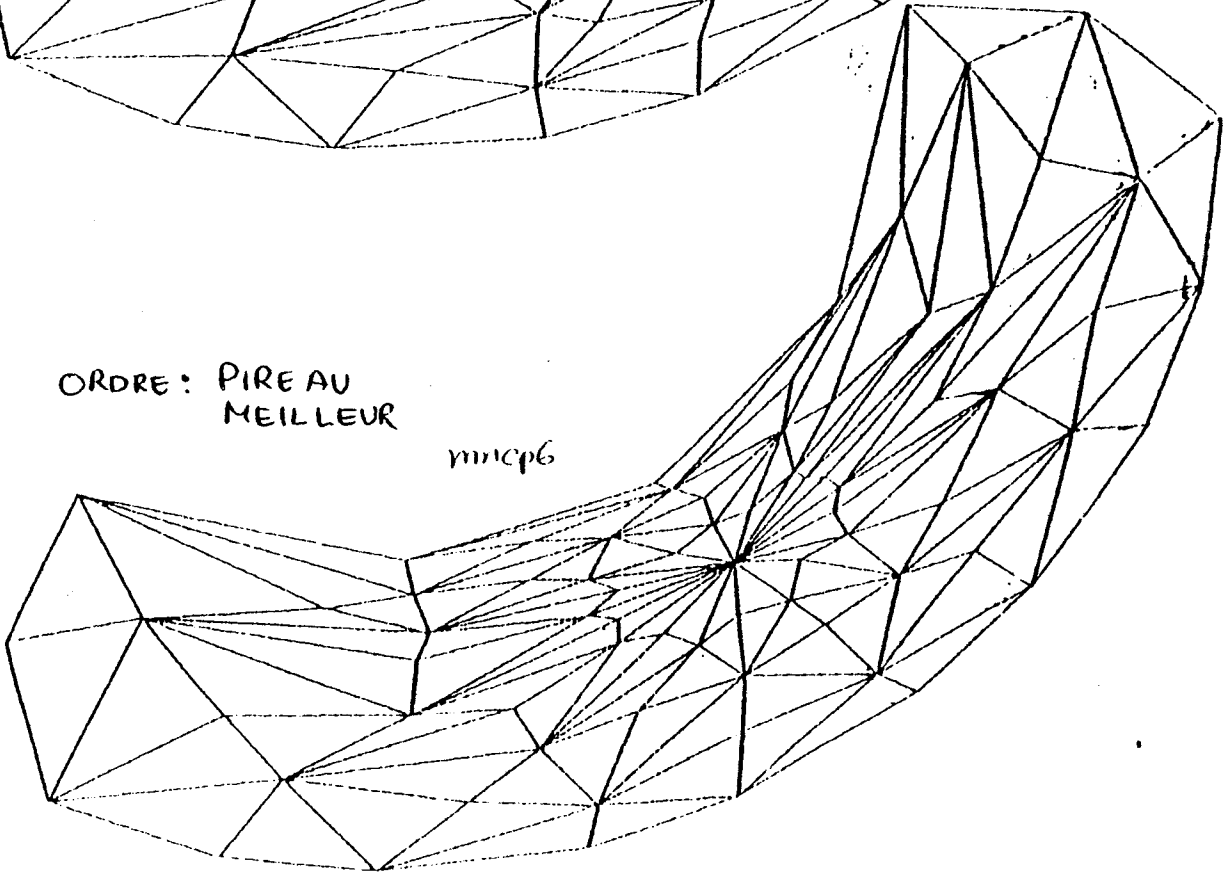
ORDRE: MEILLEUR
AU
PIRE

mncp5

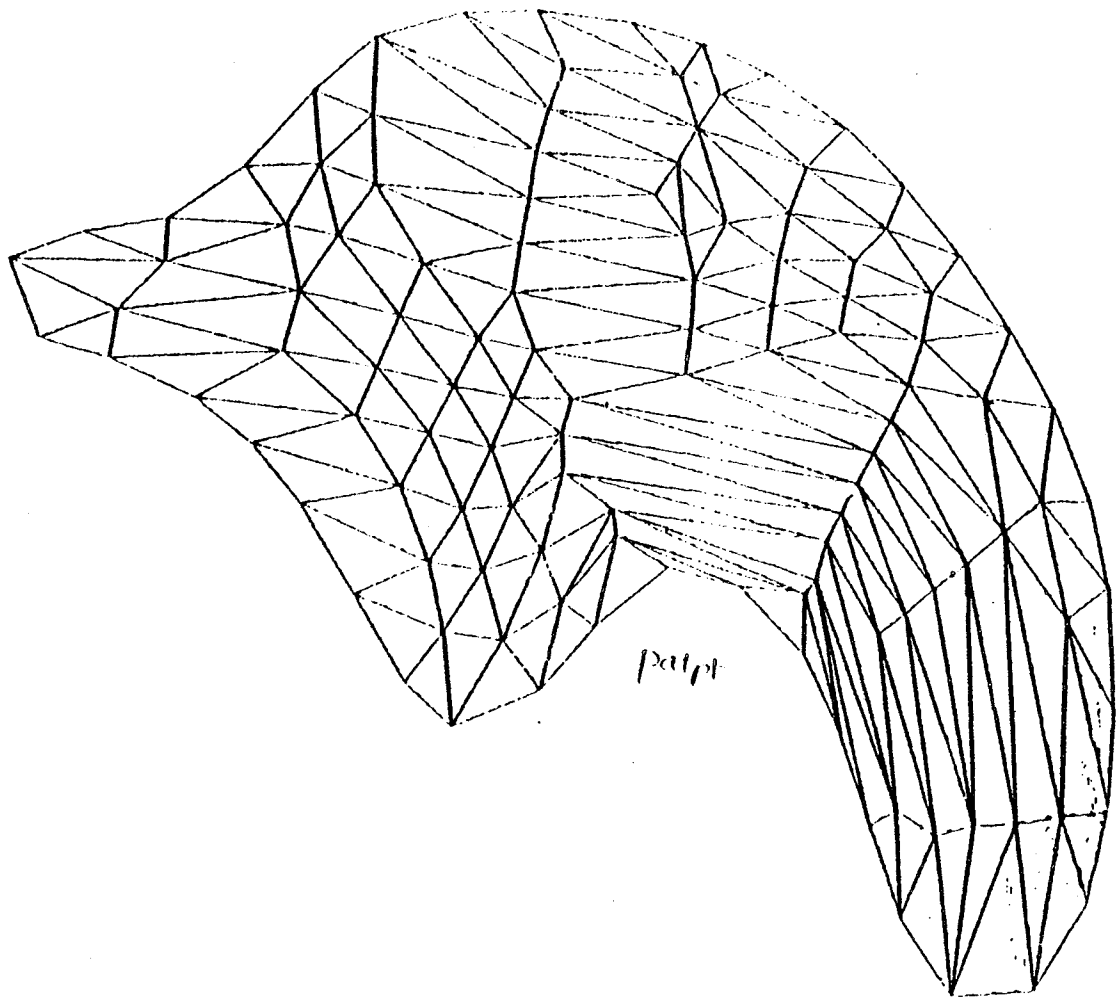


ORDRE: PIRE AU
MEILLEUR

mncp6

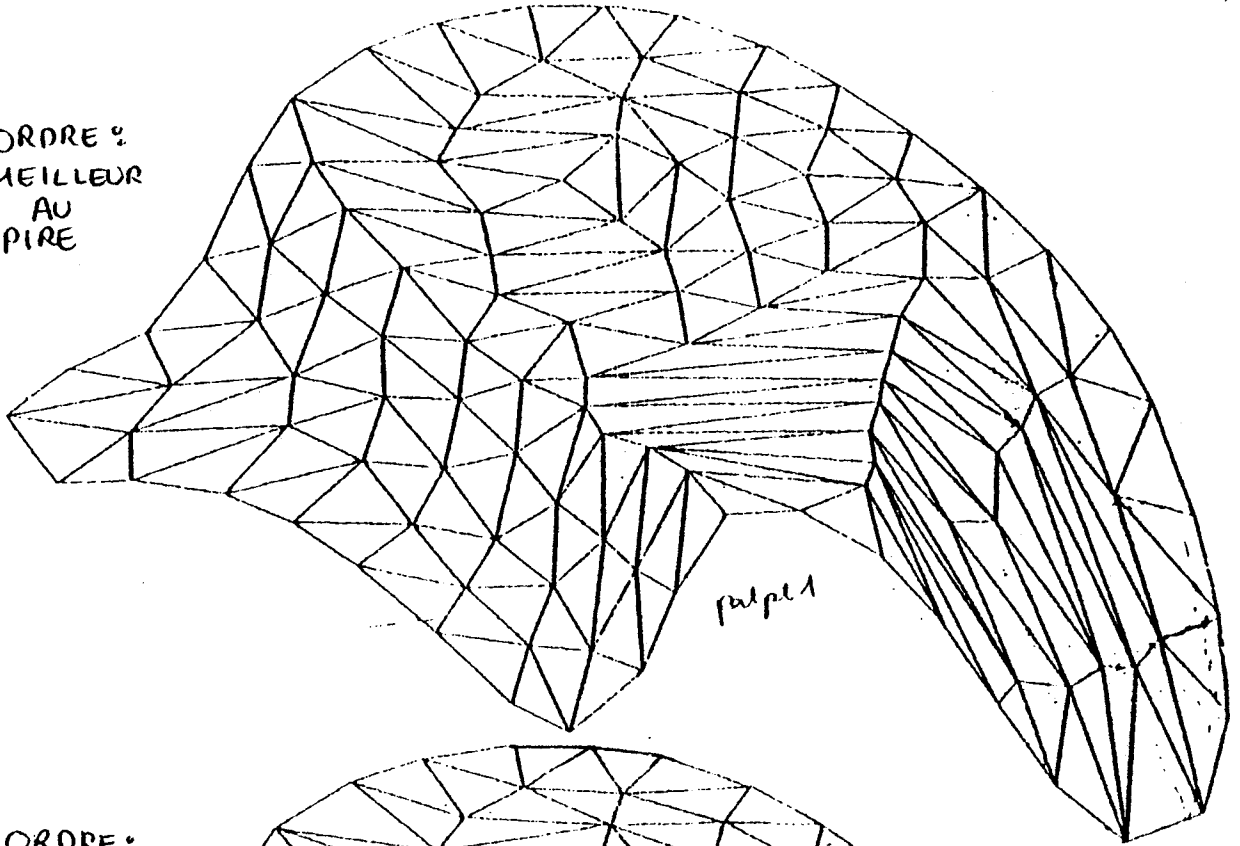


ESSAIS (CAS REEL)
MAILLAGE INITIAL

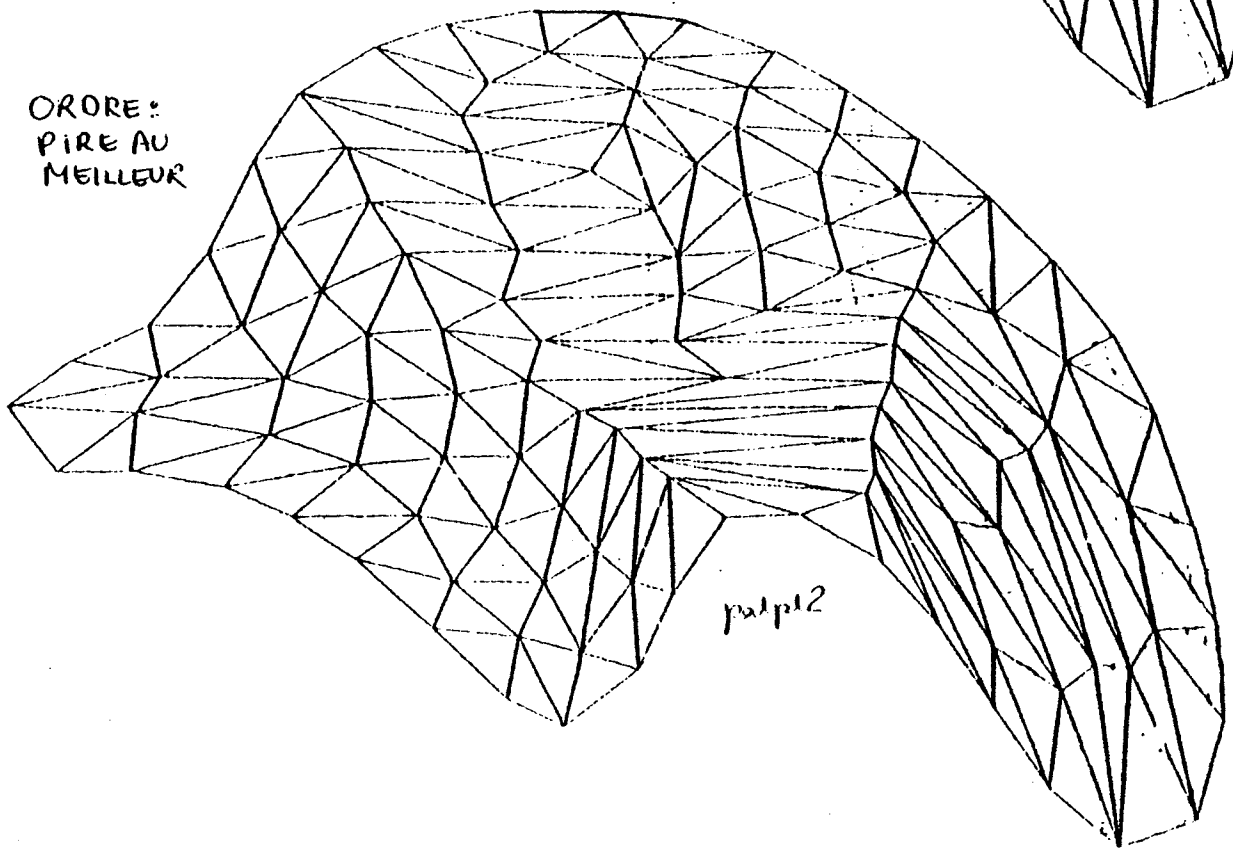


ESSAIS
MAILLAGES FINALS
CRITERE : $C_1(P_0)$

ORDRE :
MEILLEUR
AU
PIRE

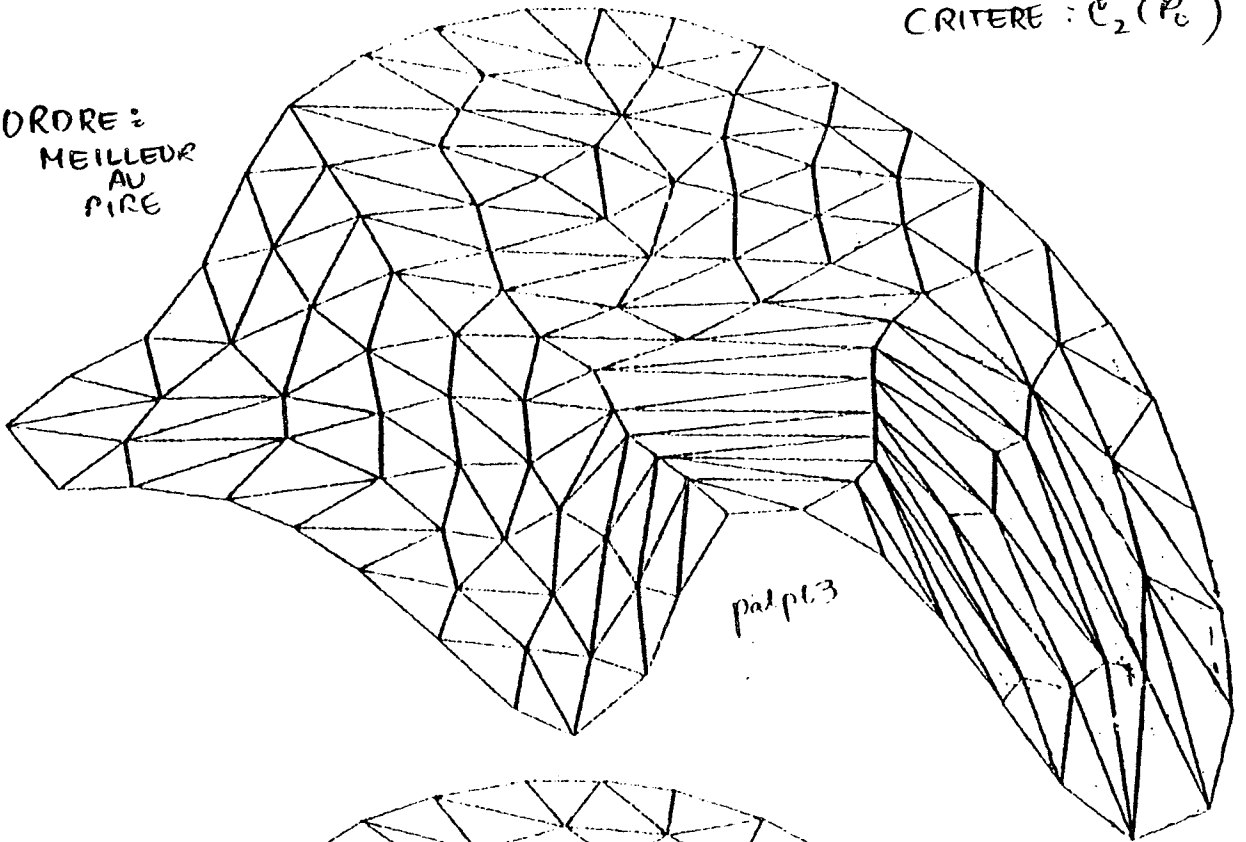


ORDRE :
PIRE AU
MEILLEUR

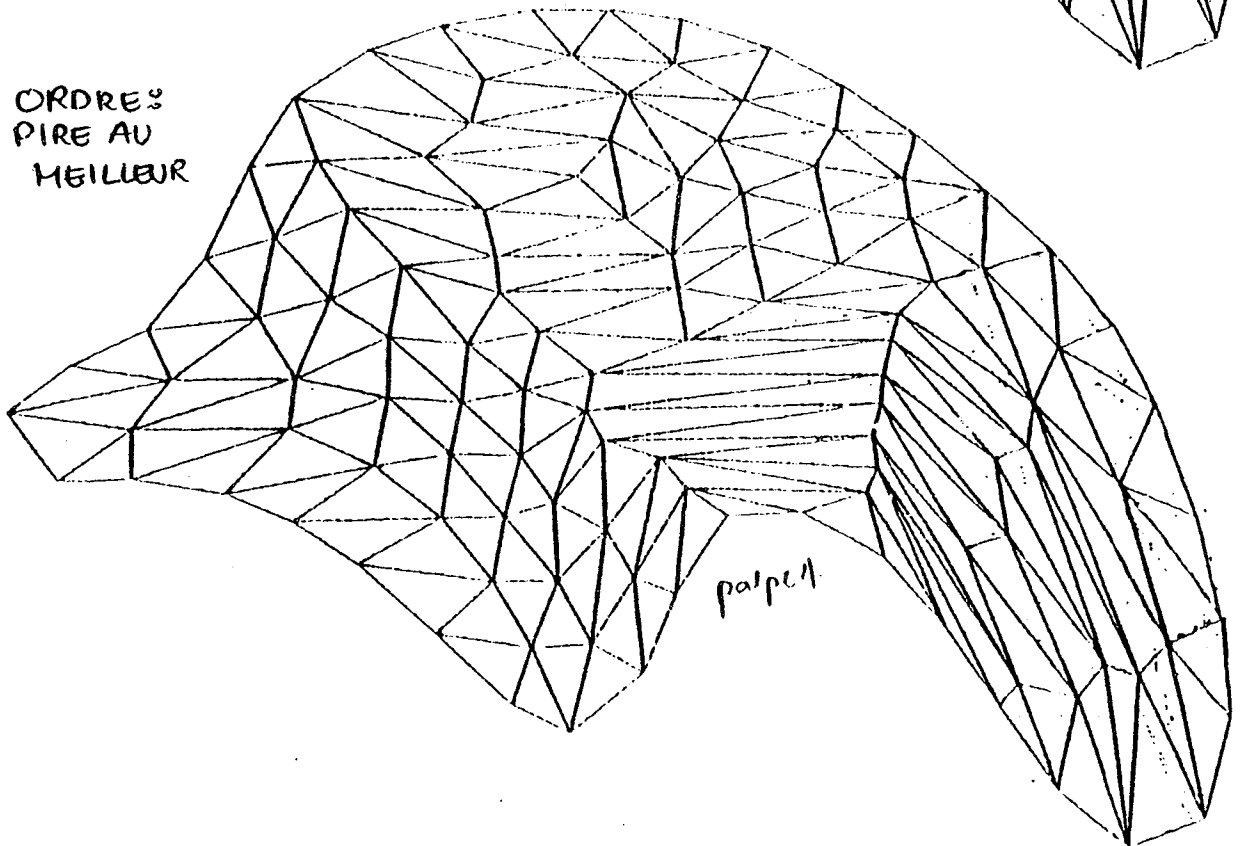


ESSAI 5
MAILLAGES FINALS
CRITERE : $C_2(P_0)$

ORDRE :
MEILLEUR
AU
PIRE

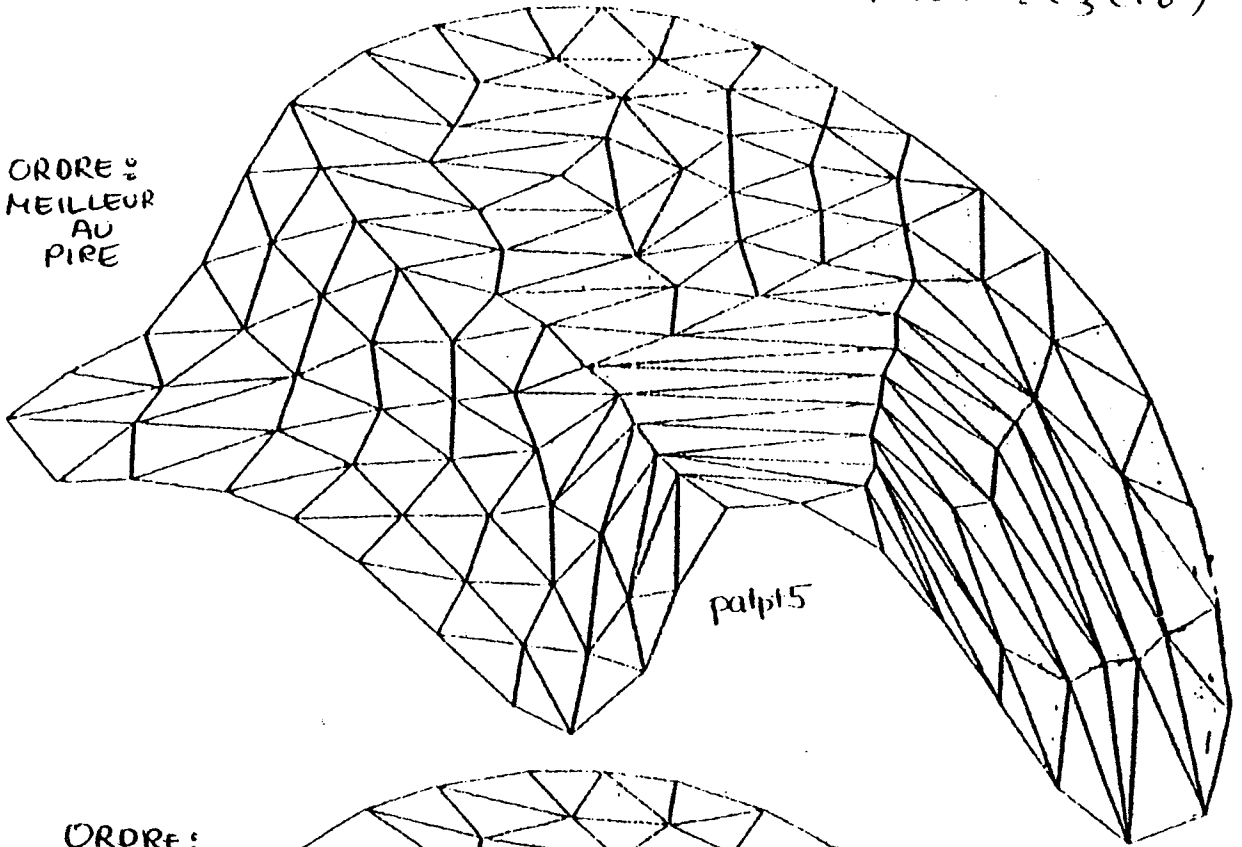


ORDRE :
PIRE AU
MEILLEUR

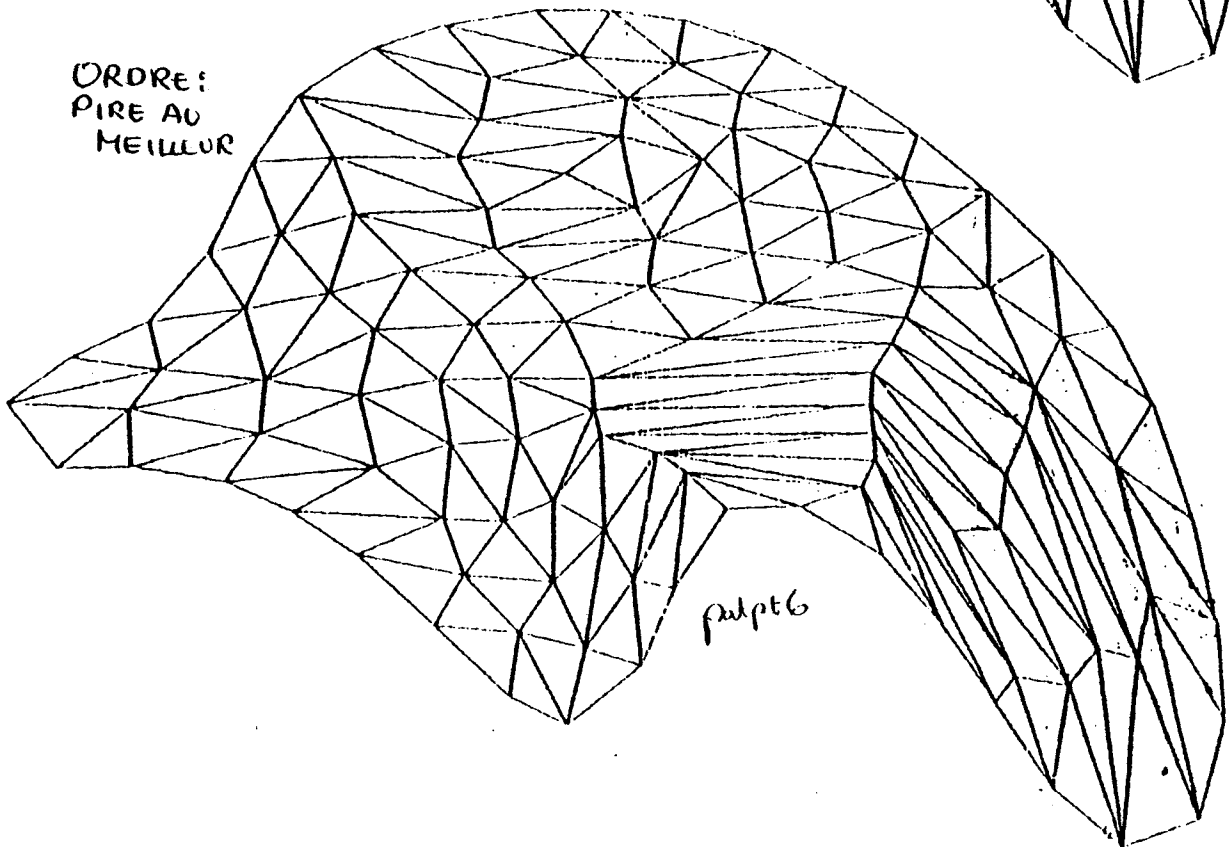


ESSAI 5
MAILLAGES FINALS
CRITERE : $C_3(P_0)$

ORDRE :
MEILLEUR
AU
PIRE



ORDRE :
PIRE AU
MEILLEUR



5.1.2 Courbes de Convergence de POURAME, le Pourcentage d'Amélioration du Maillage Par Rapport au Nombre de Balayages de l'Ensemble de Noeuds

Nous rappelons ici la définition de POURAME :

$$\text{POURAME} = 100 \times \frac{\text{la somme des améliorations locales pour chaque noeud}}{\text{la somme des valeurs initiales de la fonction critère pour chaque noeud}}$$

On a calculé les valeurs de POURAME pour certains de nos exemples de 5.1.1.

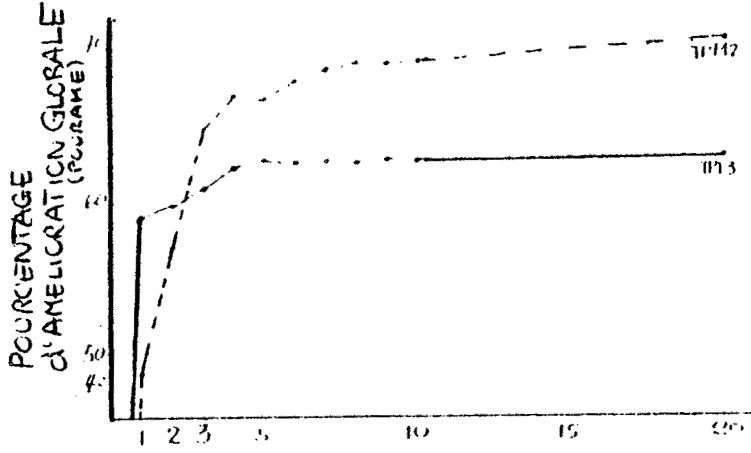
Voici les valeurs de POURAME pour les TOUR=1,...,10 et celles pour TOUR=20:

(NOM est le nom de l'exemple; NEWPTF désigne le nom de la figure finale.)

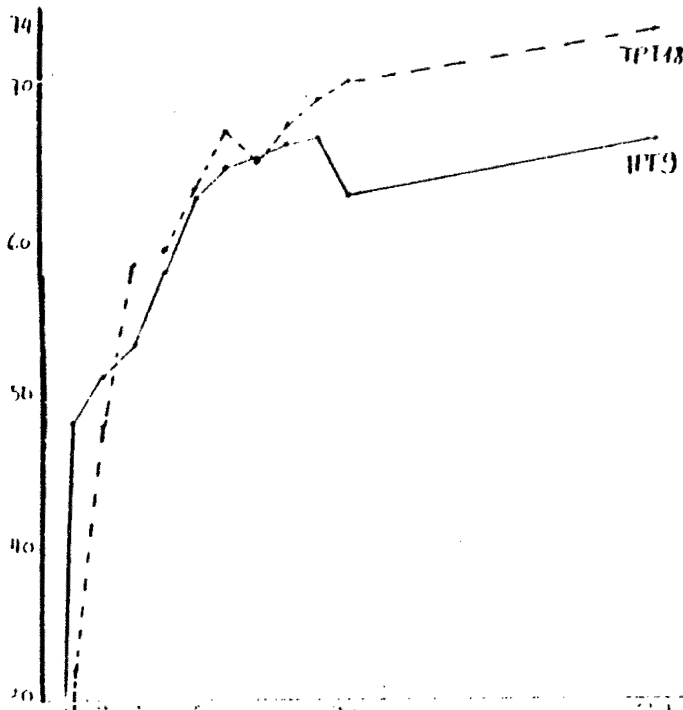
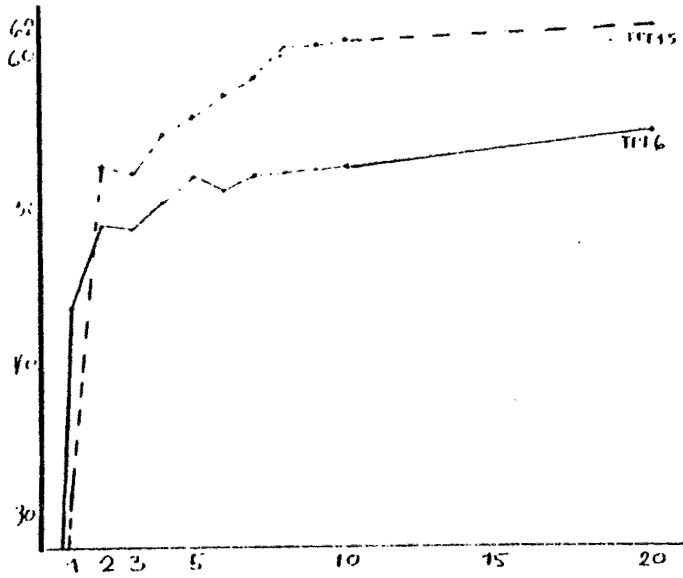
NOM		Essai1	Essai1	Essai1	Essai1	Essai1	Essai1	Essai5
NEWPTF		tpt3	tpt12	tpt6	tpt15	tpt9	tpt18	patpt3
POURAME	1	58,83	48,86	43,41	31,25	48,15	31,89	24,31
	2	59,68	56,92	48,85	52,87	51,32	47,91	30,91
	3	60,77	64,38	48,47	52,24	53,25	58,49	37,09
	4	62,03	66,58	50,26	54,61	57,98	59,32	39,81
	5	62,47	66,43	52,01	55,92	63,08	63,69	40,83
	6	62,34	67,60	51,21	57,42	64,88	67,27	41,80
	7	62,39	68,17	52,10	58,50	65,46	65,36	44,67
	8	62,28	68,66	52,37	60,85	66,31	67,50	45,56
	9	62,36	68,60	52,58	60,88	66,69	69,19	46,24
	10	62,38	68,78	52,84	61,15	62,97	70,37	47,56
	20	62,80	70,13	55,23	62,17	66,62	73,52	51,44
NOM		Essai2	Essai2	Essai3	Essai3	Essai4	Essai4	Essai5
NEWPTF		sbpt5	sbpt6	smrp1	smrp2	mncp1	mncp3	patpt1
POURAME	1	20,71	8,49	11,85	20,60	29,45	18,55	19,04
	2	32,52	25,71	31,92	34,44	34,67	32,17	29,74
	3	36,30	29,73	38,55	46,22	36,73	38,95	33,54
	4	36,31	32,02	42,88	51,61	38,70	39,56	37,03
	5	36,17	32,60	47,04	53,66	39,91	41,99	39,02
	6	36,30	34,91	48,81	58,32	37,85	45,81	40,24
	7	36,38	35,94	51,67	59,32	40,04	46,60	40,91
	8	40,01	37,89	52,95	61,87	41,94	48,28	41,71
	9	40,44	37,86	54,82	62,76	43,10	49,95	42,10
	10	40,67	39,14	55,10	63,71	43,61	50,37	42,56
	20	41,86	41,04	58,22	65,06	48,01	51,25	45,39

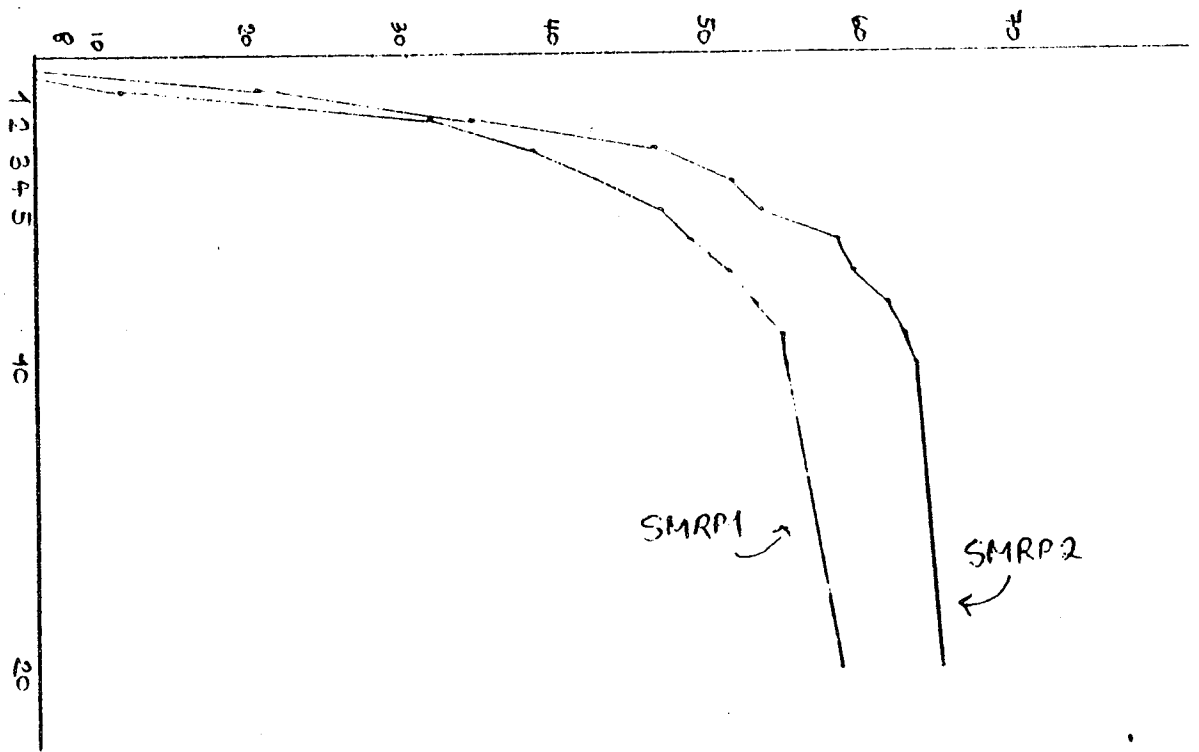
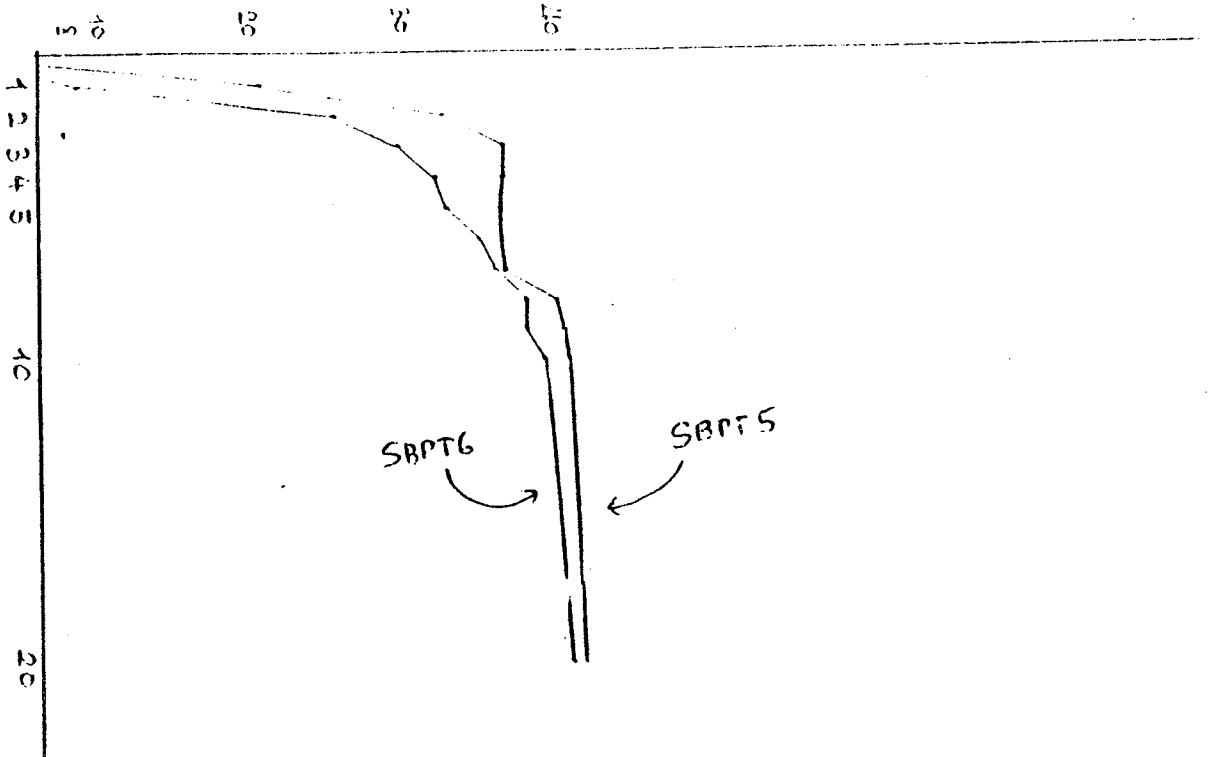
Remarque que POURAME = 0% pour TOUR=0! Dans les pages suivantes, on va tracer les courbes de convergence du pourcentage d'amélioration globale qui correspondent au tableau

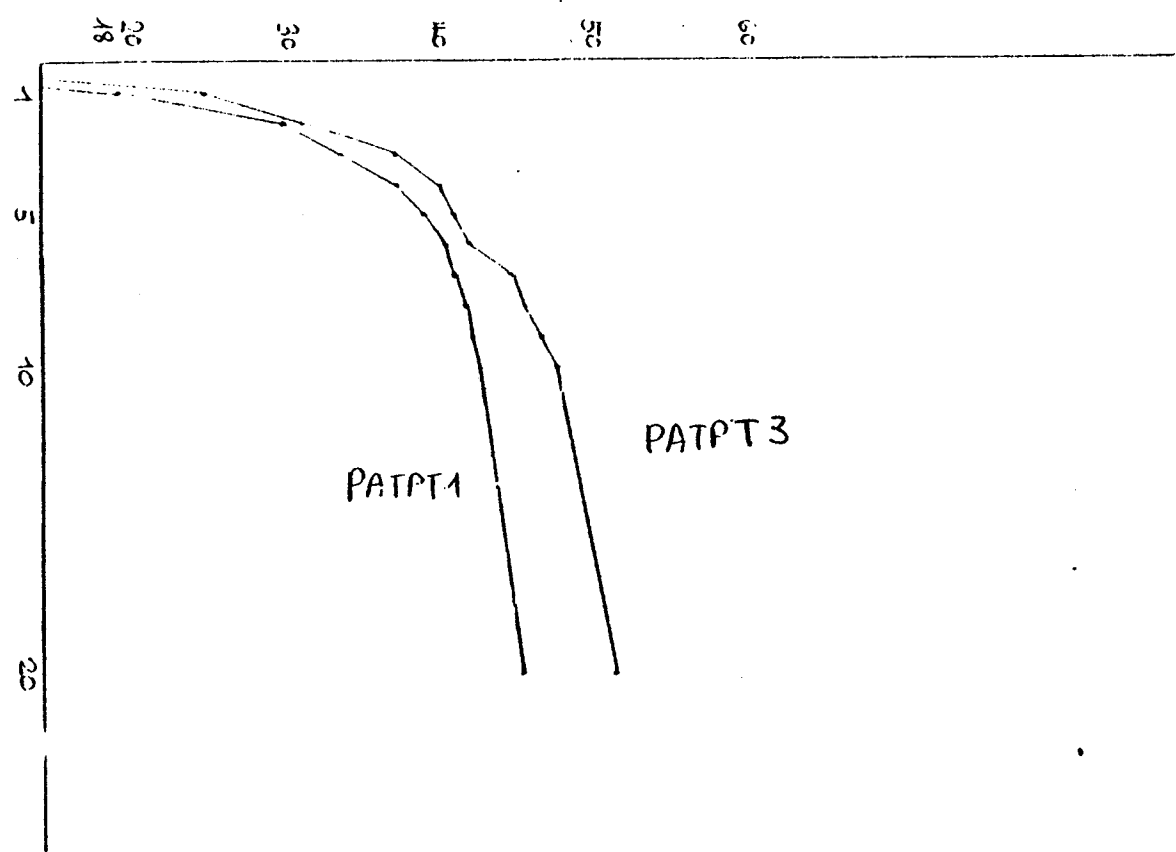
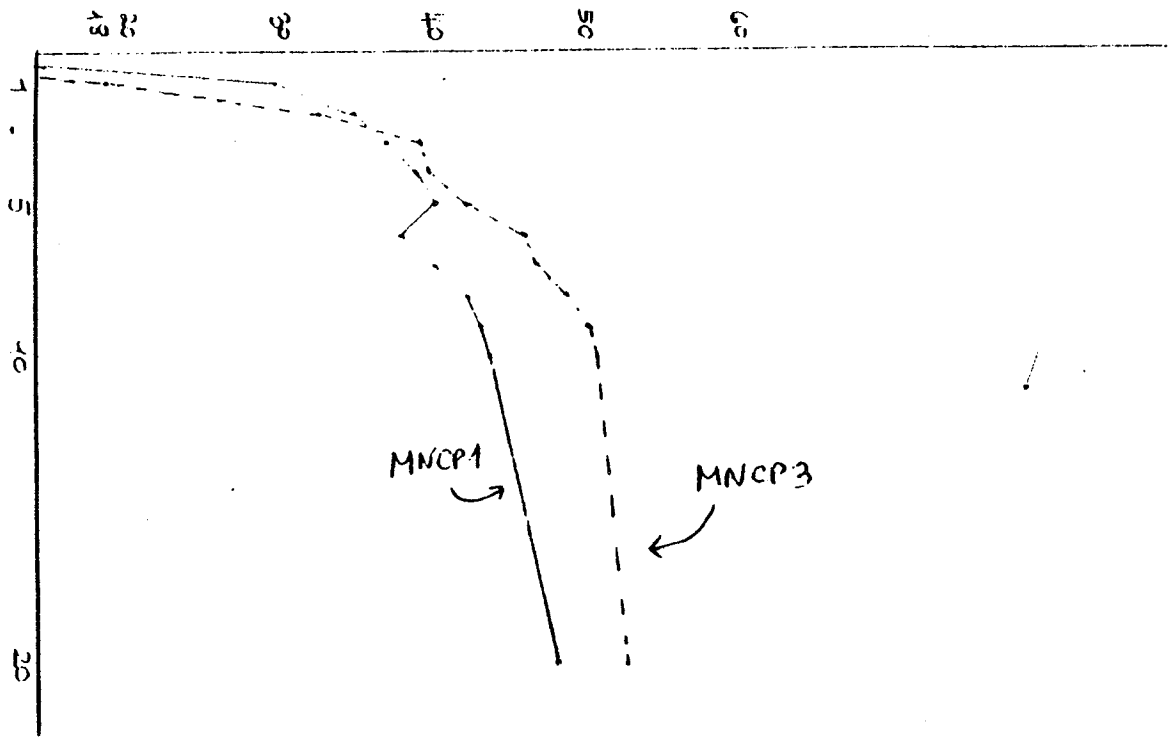
Voici les courbes de convergence tracées :



NOMBRE de
BALAYAGES de l'ensemble
de nœuds







On a tracé les courbes de convergence afin de connaître l'amélioration possible en augmentant le nombre de tours. Sachant qu'on ne peut pas obtenir une solution optimale, on peut observer le comportement de l'amélioration globale en fonction du nombre de tours.

On aperçoit que pour toutes les courbes tracées, la progression du pourcentage d'amélioration devient négligeable à partir du 7^e tour. C'est pour cette raison qu'on analyse la progression moyenne du pourcentage d'amélioration de chaque tour entre le 1^e et le 7^e; le 2^e et le 7^e; le 3^e et le 7^e. Voici les valeurs recueillies :

- entre 1^e et 7^e : 6,5; 5,5; 4,8; 4,7; 4,5; 4,5; 3,5; 3,5; 3,2; 2,9; 2,5; 1,8; 1,3; 0,6.
- entre 2^e et 7^e : 4,1; 3,0; 2,9; 2,8; 2,4; 2,2; 2,0; 1,9; 1,7; 1,0; 1,0; 0,6; 0,6; 0,5.
- entre 3^e et 7^e : 2,2; 2,2; 2,0; 1,3; 1,3; 1,3; 1,1; 1,0; 1,0; 0,7; 0,6; 0,6; 0,3; 0,0.

Les résultats obtenus montrent que la progression est très rapide entre les trois premiers tours et peu sensible par la suite.

Afin de donner support à cette supposition, on compte combien de fois dans les quatorze exemples, il y a une progression supérieure à 2,5% entre le 1^e tour et le 2^e tour, entre le 2^e tour et le 3^e tour, entre le 3^e tour et le 4^e tour, et, ainsi de suite, jusqu'au 7^e tour.

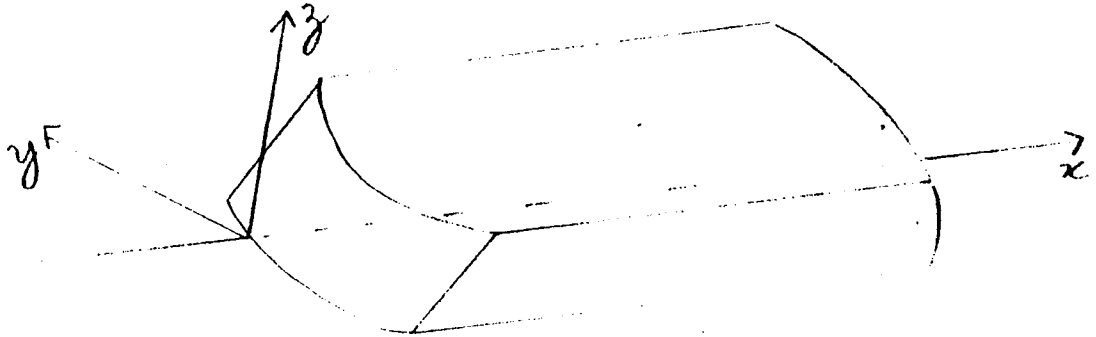
- entre 1^e et 2^e : 13; - entre 2^e et 3^e : 9
- entre 3^e et 4^e : 5; - entre 4^e et 5^e : 3
- entre 5^e et 6^e : 2; - entre 6^e et 7^e : 3.

Ces résultats confirment notre supposition car il y a une large majorité de cas où une progression était assez sensible (supérieure à 2,5%) entre 2 tours parmi les trois premiers tours, et il y a une toute petite minorité de tels cas parmi les 4 derniers. On remarque, quand même, que 5 fois sur 14 entre le 3^e et le 4^e cette progression dépassait 2,5%. Aussi est-il peut-être intéressant de temps en temps d'appliquer l'ALGOTOUT pour TOUR = 4?

5.2 Autres Résultats

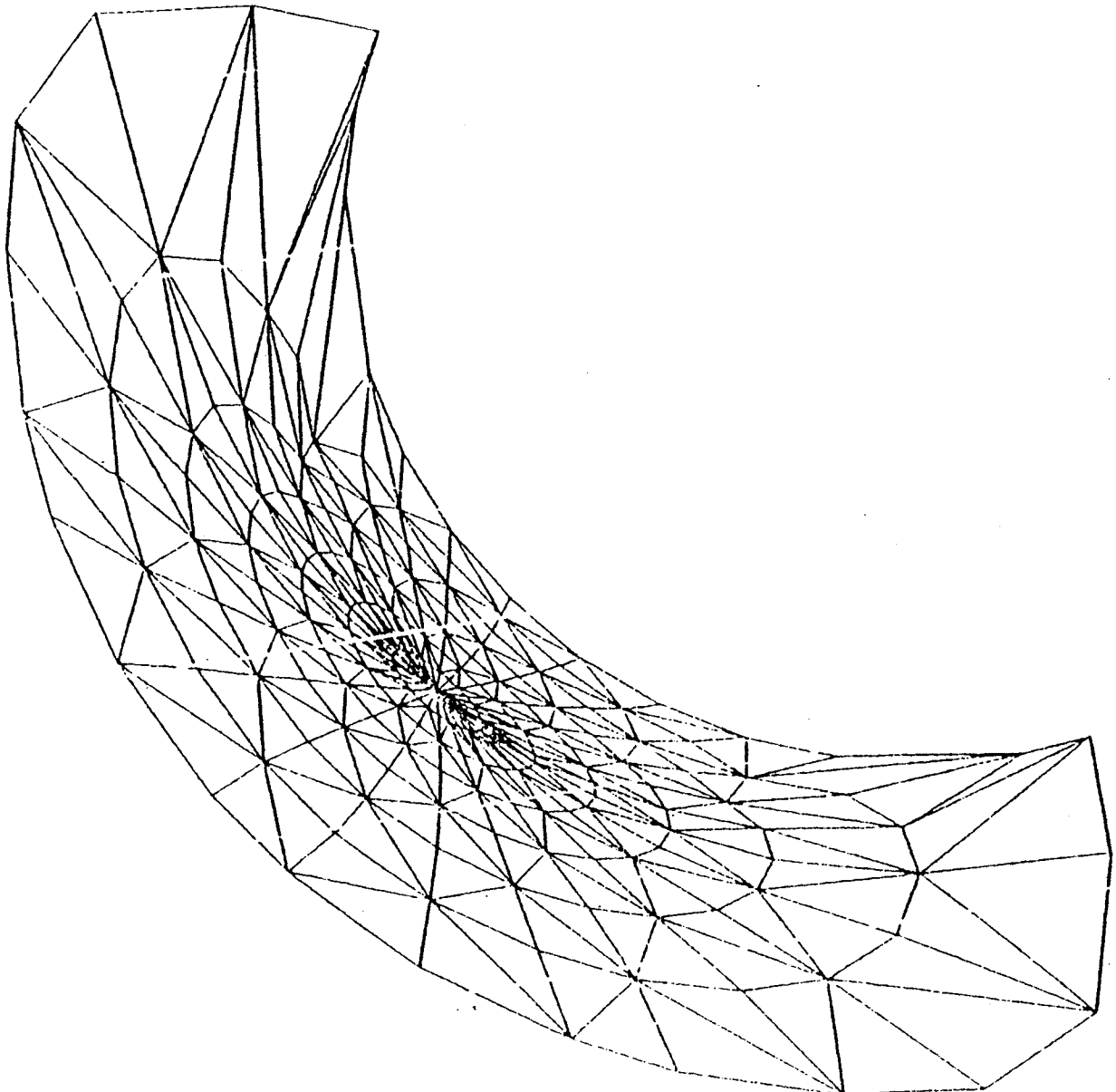
Comme le travail effectué pour aboutir à cette thèse était à la suite d'une demande du DGT/DEA, Avions Marcel Dassault, Breguet Aviation, l'application de notre algorithme sur des cas réels a été effectuée en collaboration avec Mlle C. POULETTY, ingénier stagiaire, Avions Marcel Dassault/ Breguet Aviation qui cite également ces résultats dans sa thèse [23]. Cette intégration de l'algorithme dans un contexte industriel a permis effectivement dans certains cas d'obtenir un maillage réel alors que d'autres méthodes s'étaient révélées infructueuses. Notre algorithme a été utilisé dans ce cadre en dimension 2. Ces essais en vraie grandeur ont permis de montrer l'intérêt de méthodes, en particulier en comparaison avec d'autres algorithmes de logiciel de programmation convexe (logiciel de l'INRIA).

Voici deux exemples pris sur des entrées d'air d'un avion où on montre ici une partie avec les trois axes x , y , z .

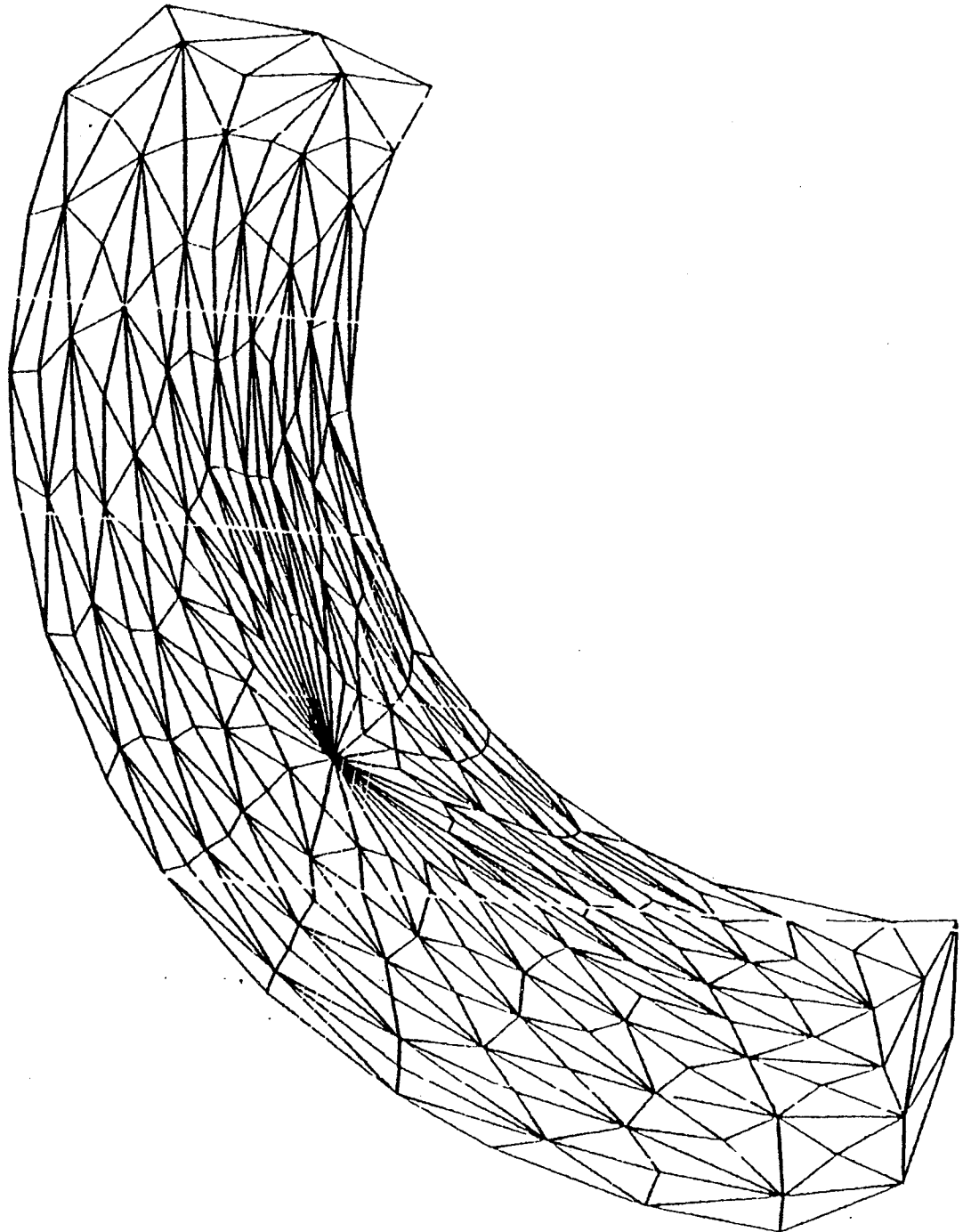


L'une est une coupe en x et l'autre en z . Le critère utilisé est minimiser $C_2(P_0) = 1 - (V_{\min}(P_0)/V_{\max}(P_0))$. L'ordre dans lequel on bouge les noeuds suit la numérotation usuelle des noeuds; trois tours ont été effectués; dans chaque tour, chaque noeud a été bougé une seule fois. Le maillage initial a été obtenu par Mlle C. POULETTY à partir de MODULEF [24].

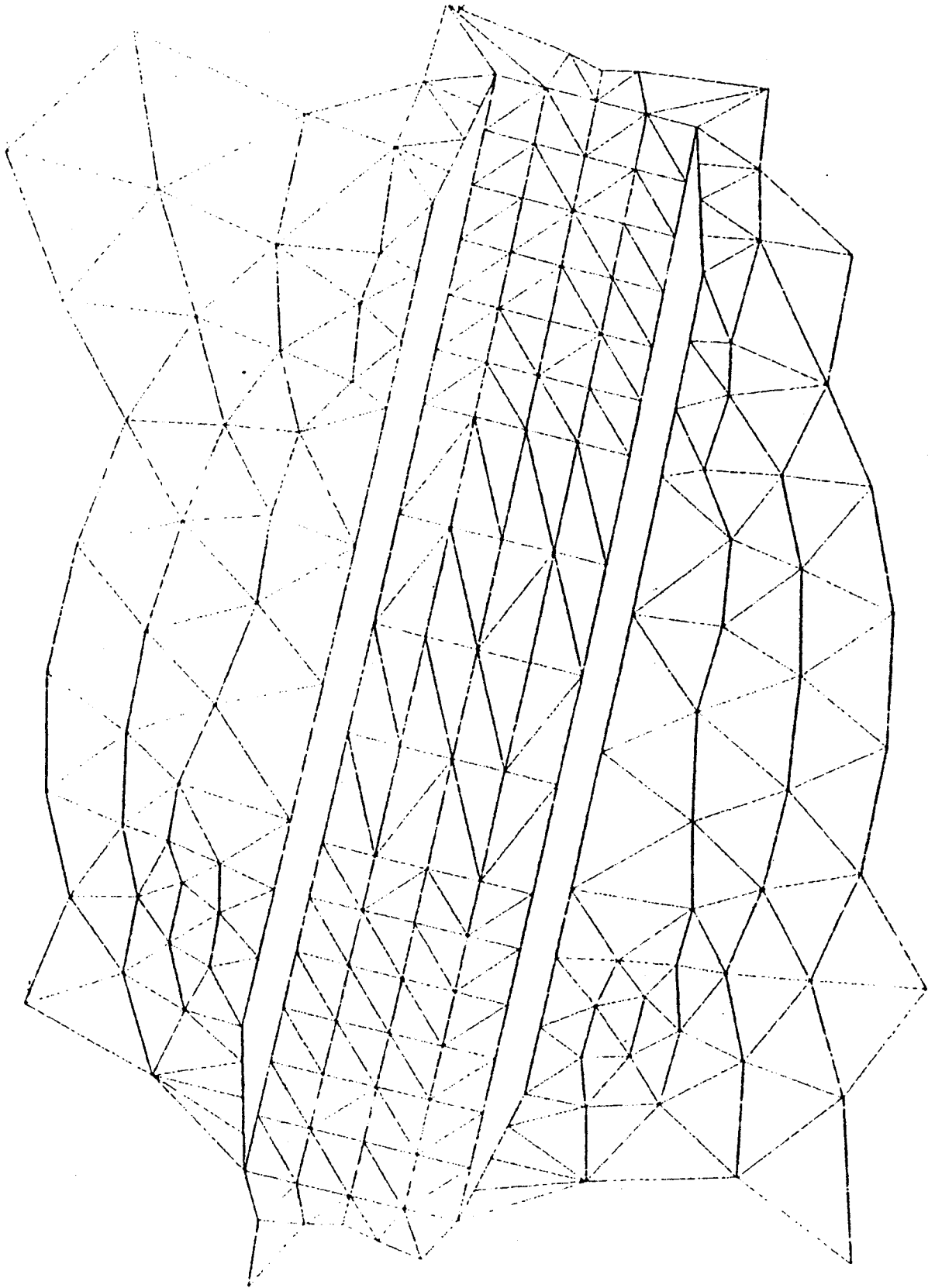
La coupe en z (maillage initial)



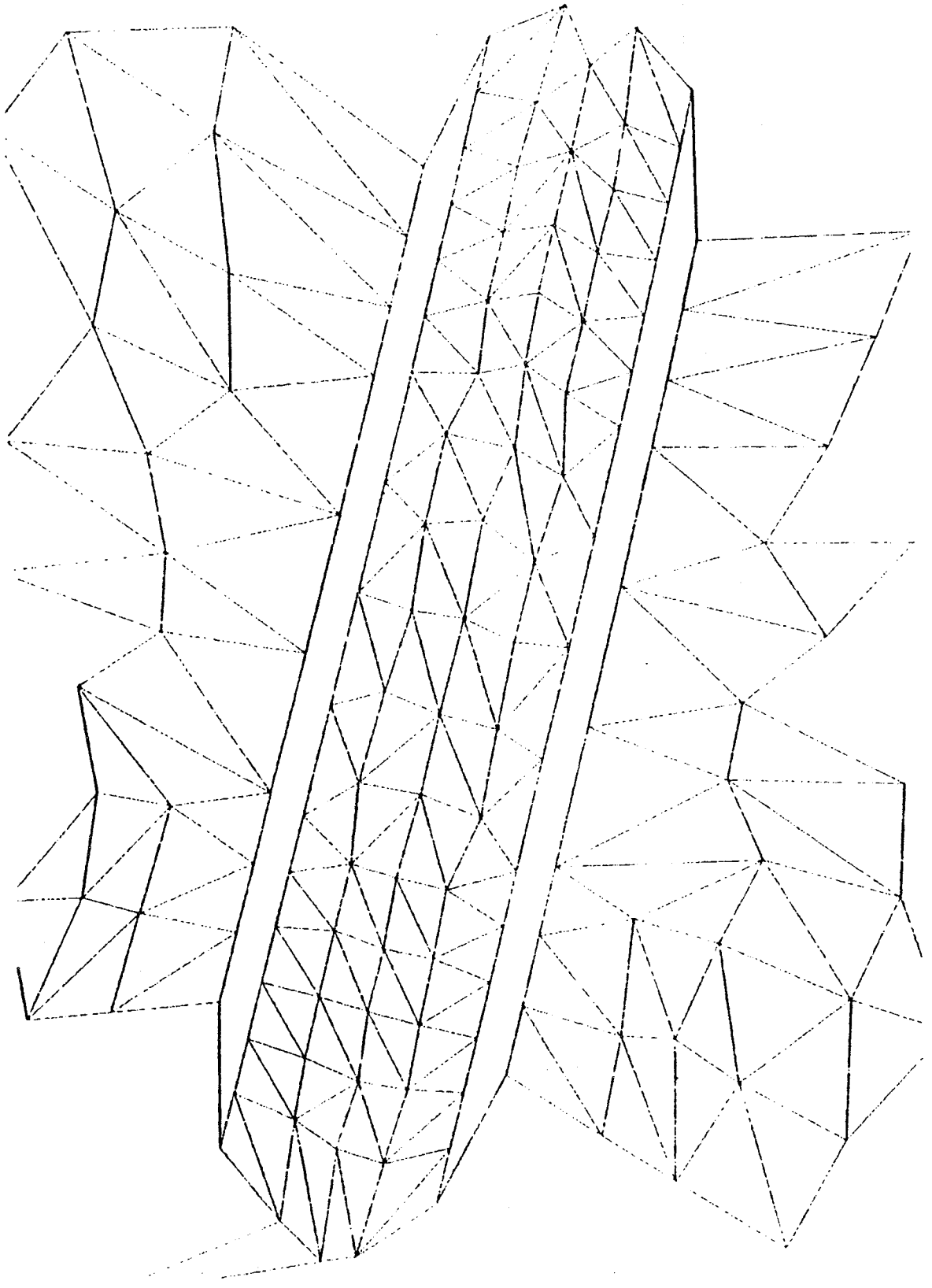
La coupe en z (maillage final)



La coupe en x (maillage initial)

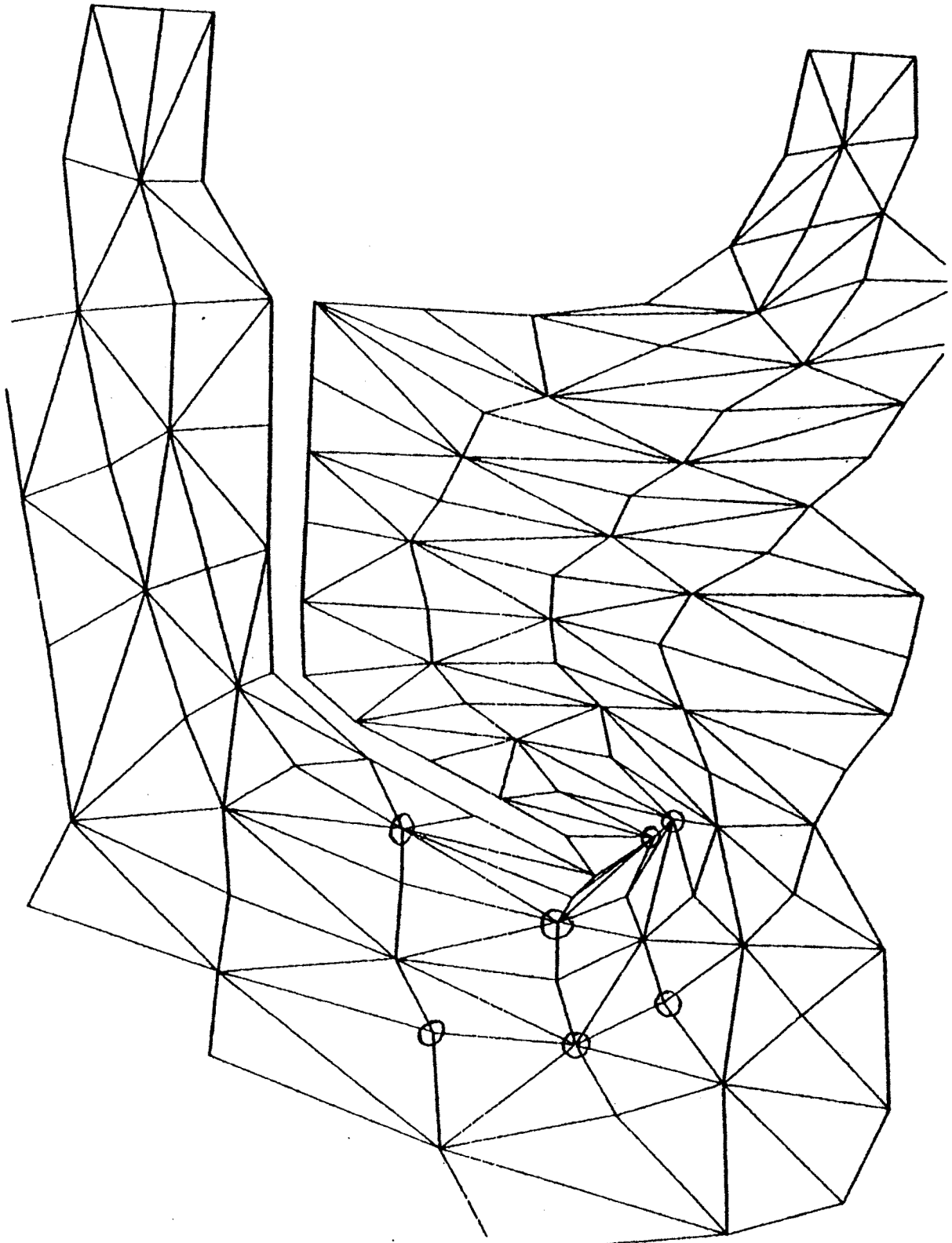


La coupe en x (maillage final)

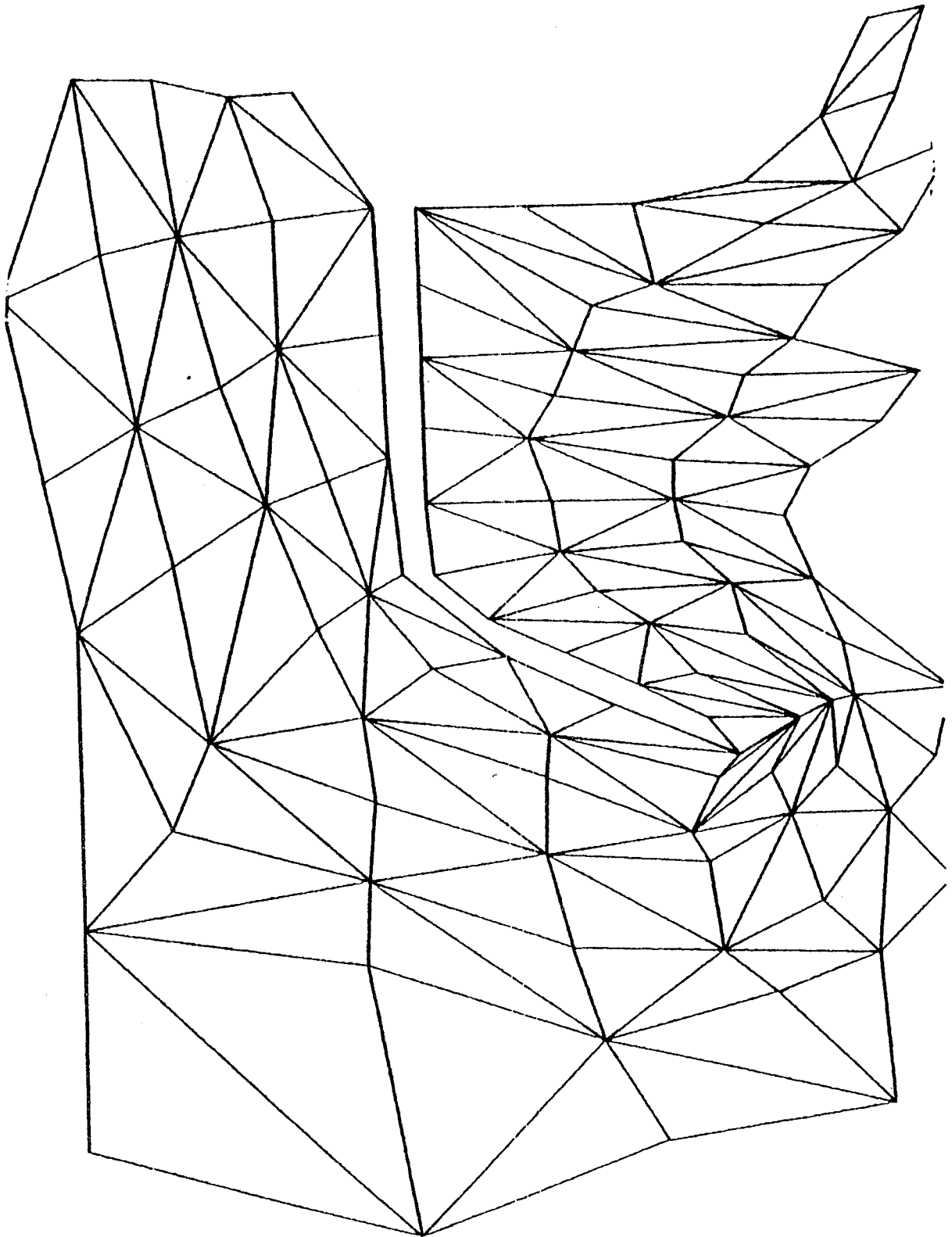


Voici un exemple où les noeuds qui ont bougé dans le maillage final sont encerclés dans le maillage initial. Un agrandissement de cette région est aussi donné. Le maillage initial a été généré par Mlle C. POULETTY à partir de MODULEF [24].

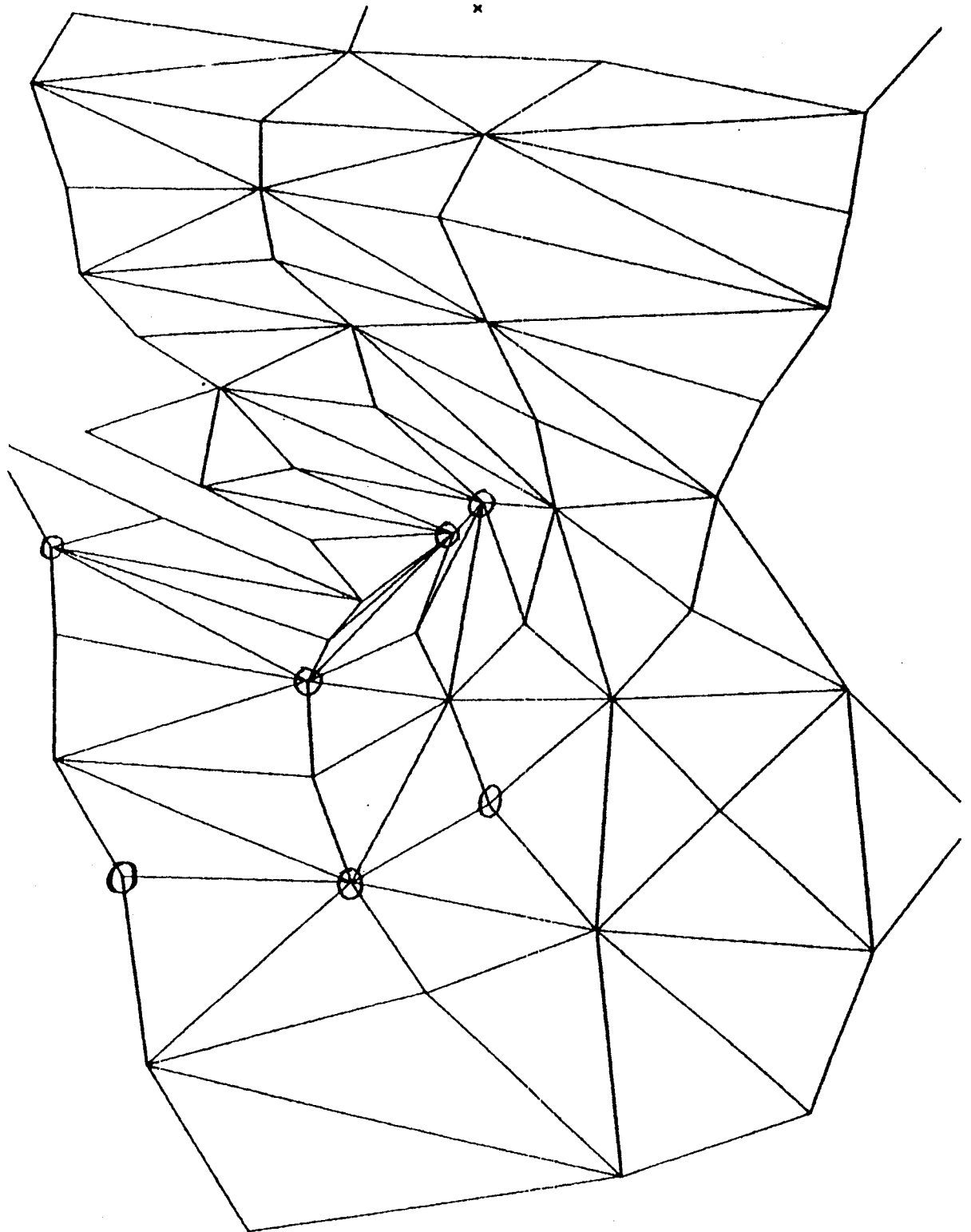
Maillage Initial



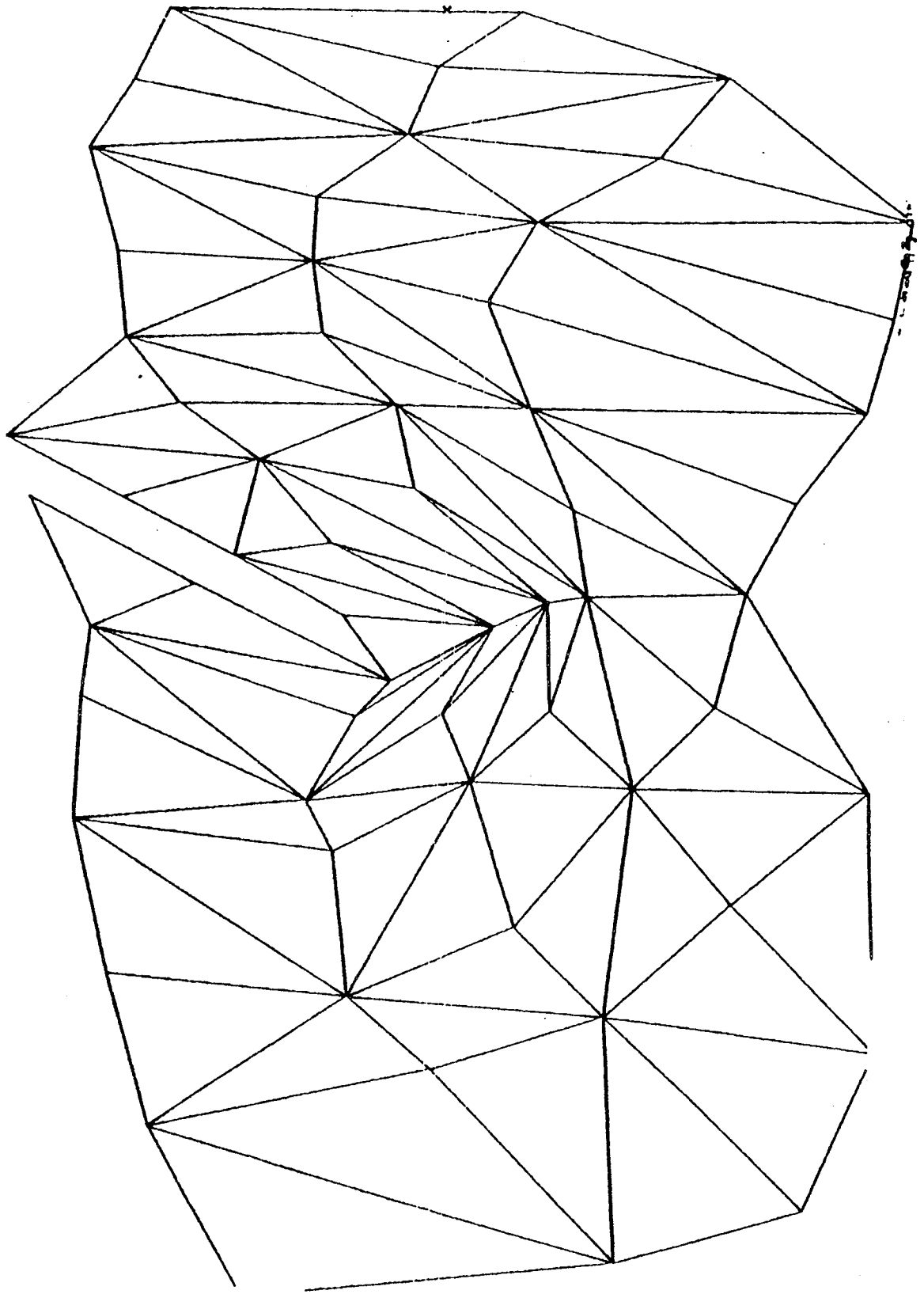
Maillage Final



Agrandissement du maillage initial

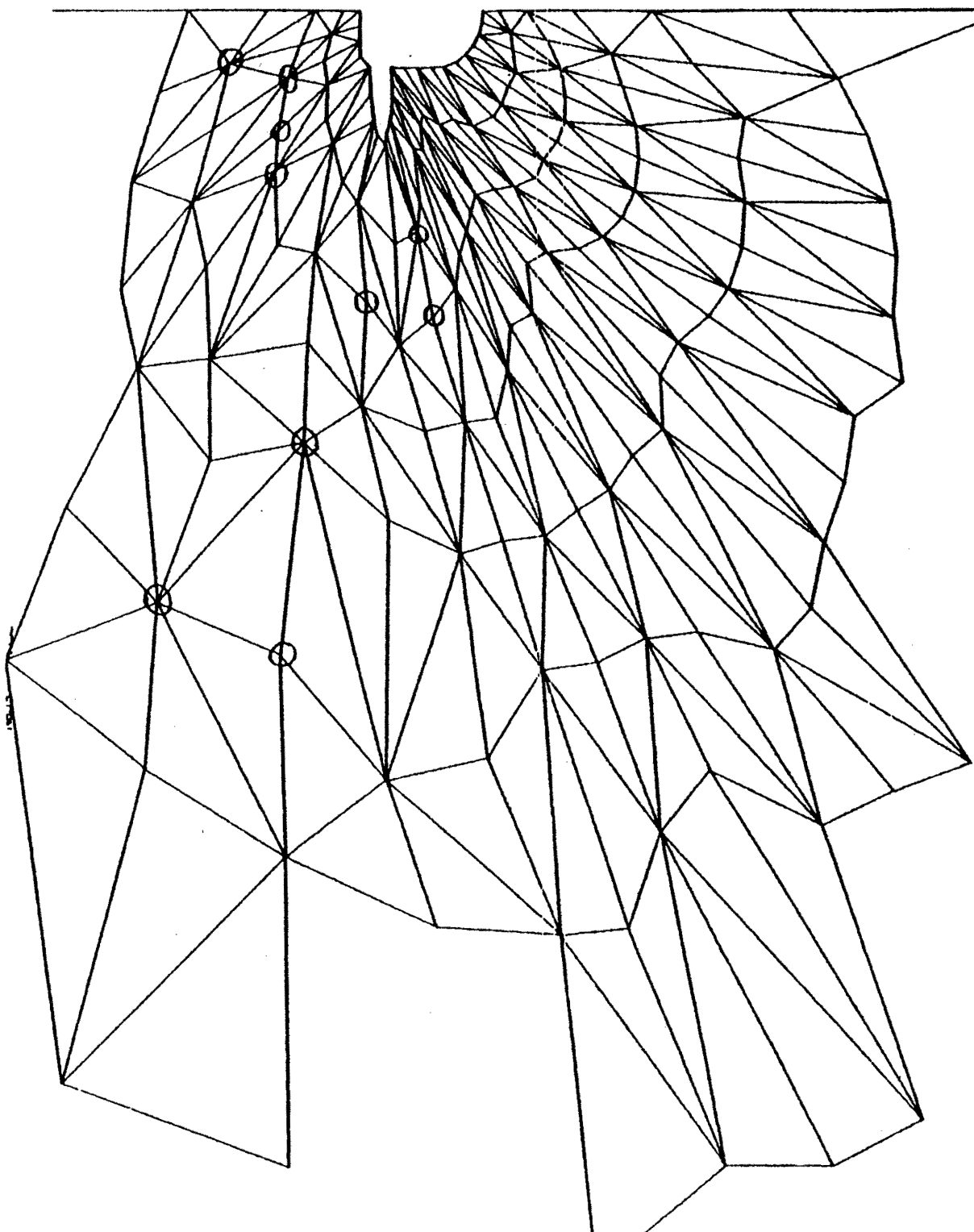


Agrandissement du maillage final

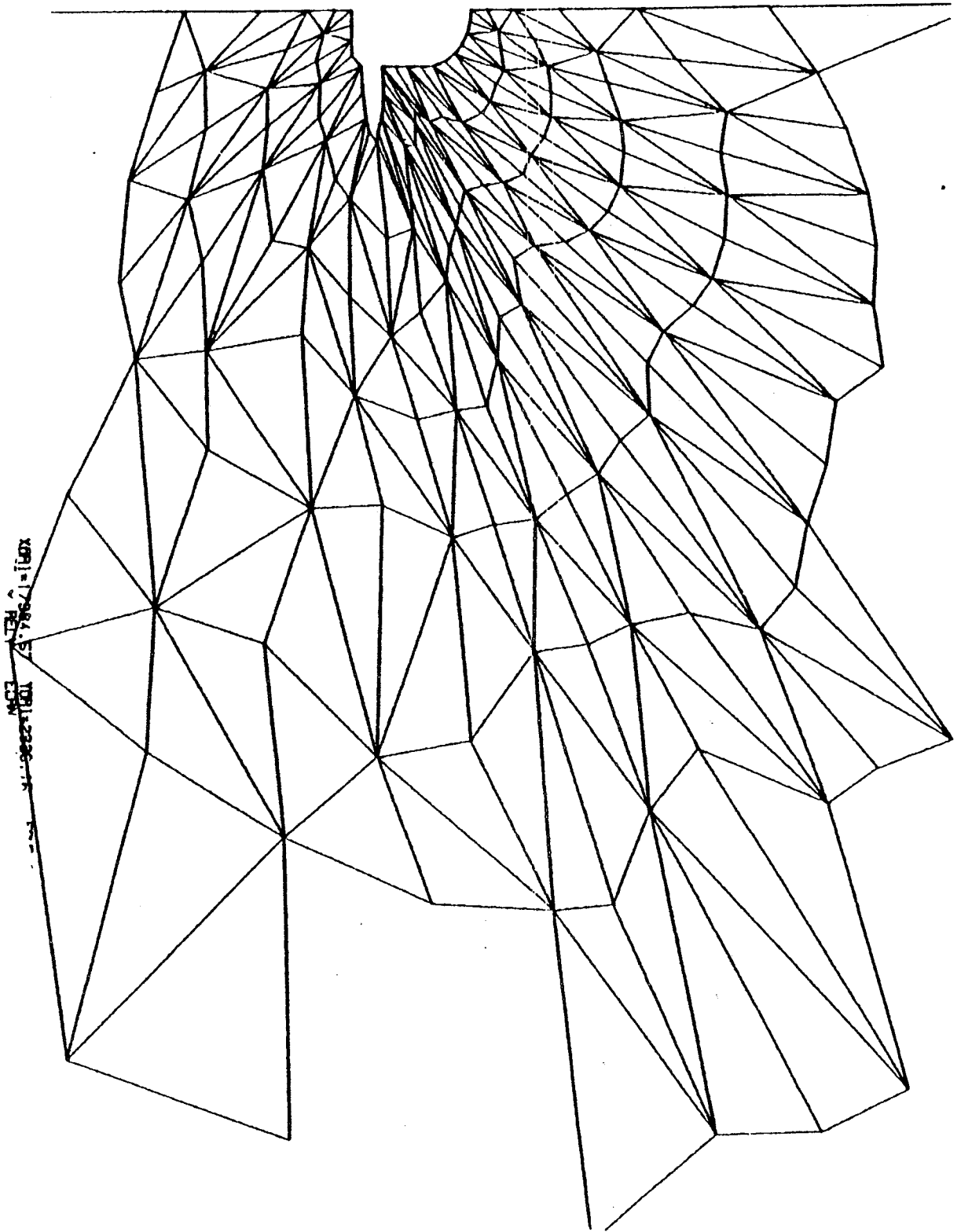


Voici un autre exemple où les noeuds qui ont bougé dans le maillage final sont encerclés dans le maillage initial. Deux agrandissements successifs de cette région sont aussi donnés. Remarque qu'il y a trois noeuds qui ont bougé qu'on n'a pas déceler que dans les agrandissements. Le maillage initial a été généré par Mlle C. POULETTY à partir de MODULEF [24].

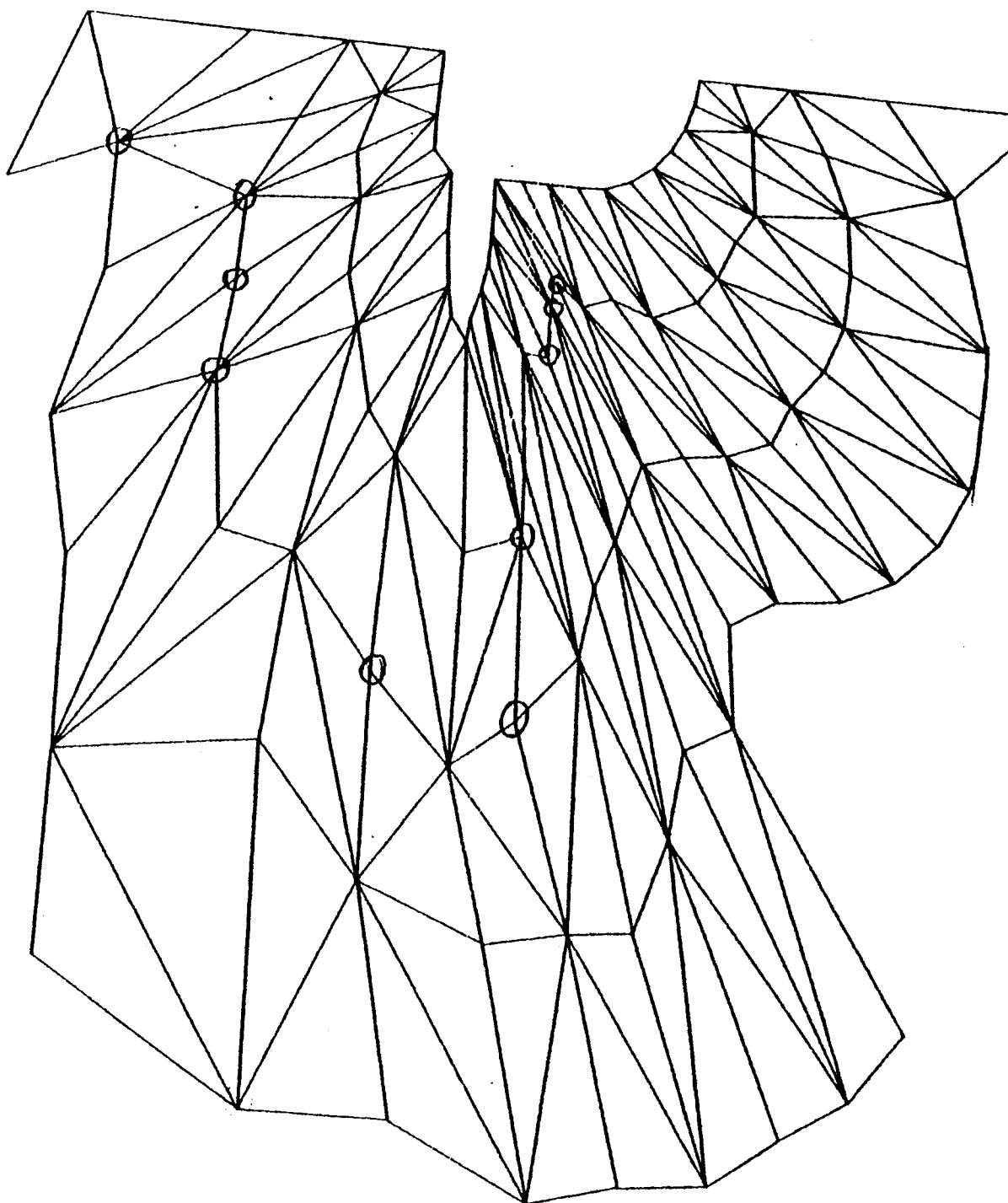
Maillage Initial



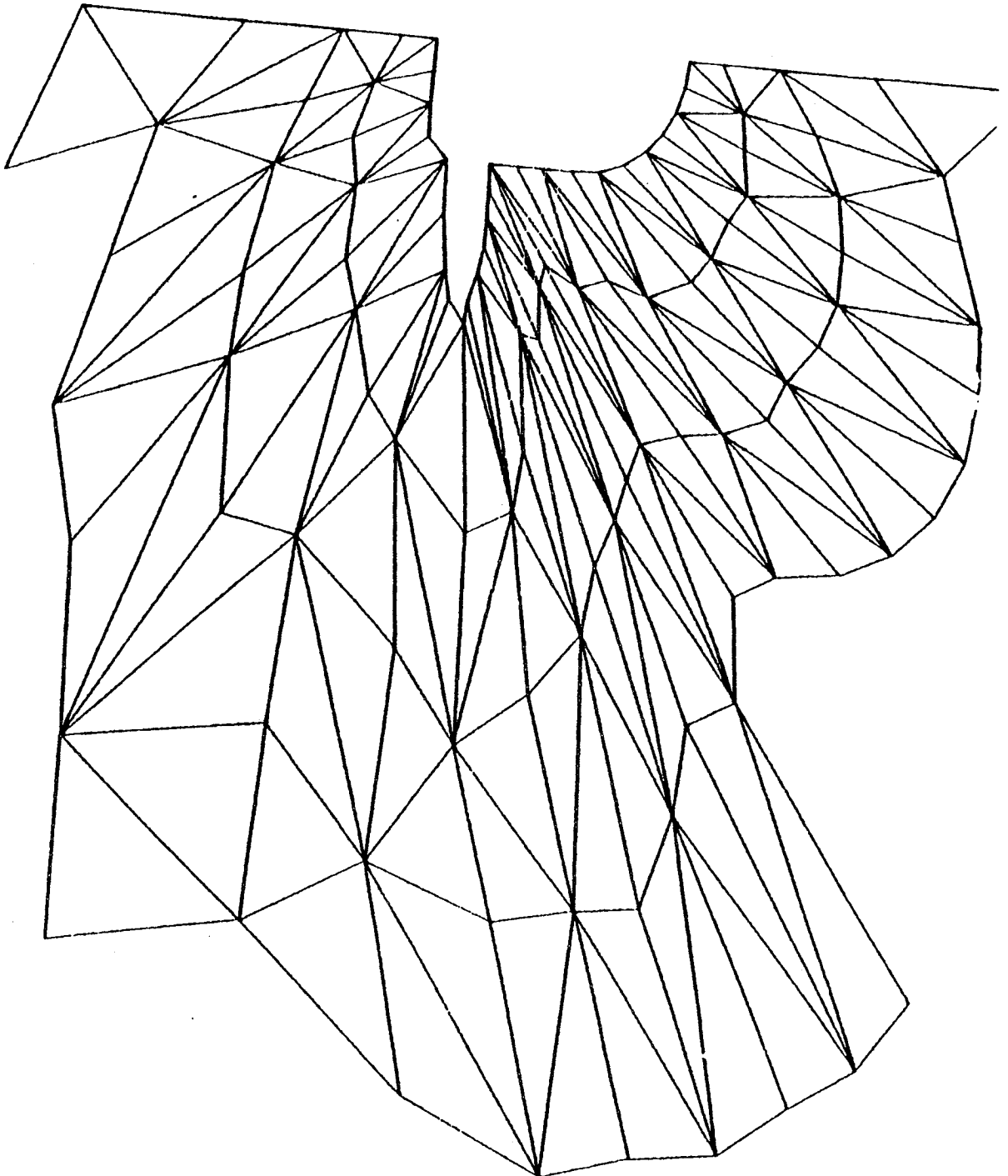
Maillage Final



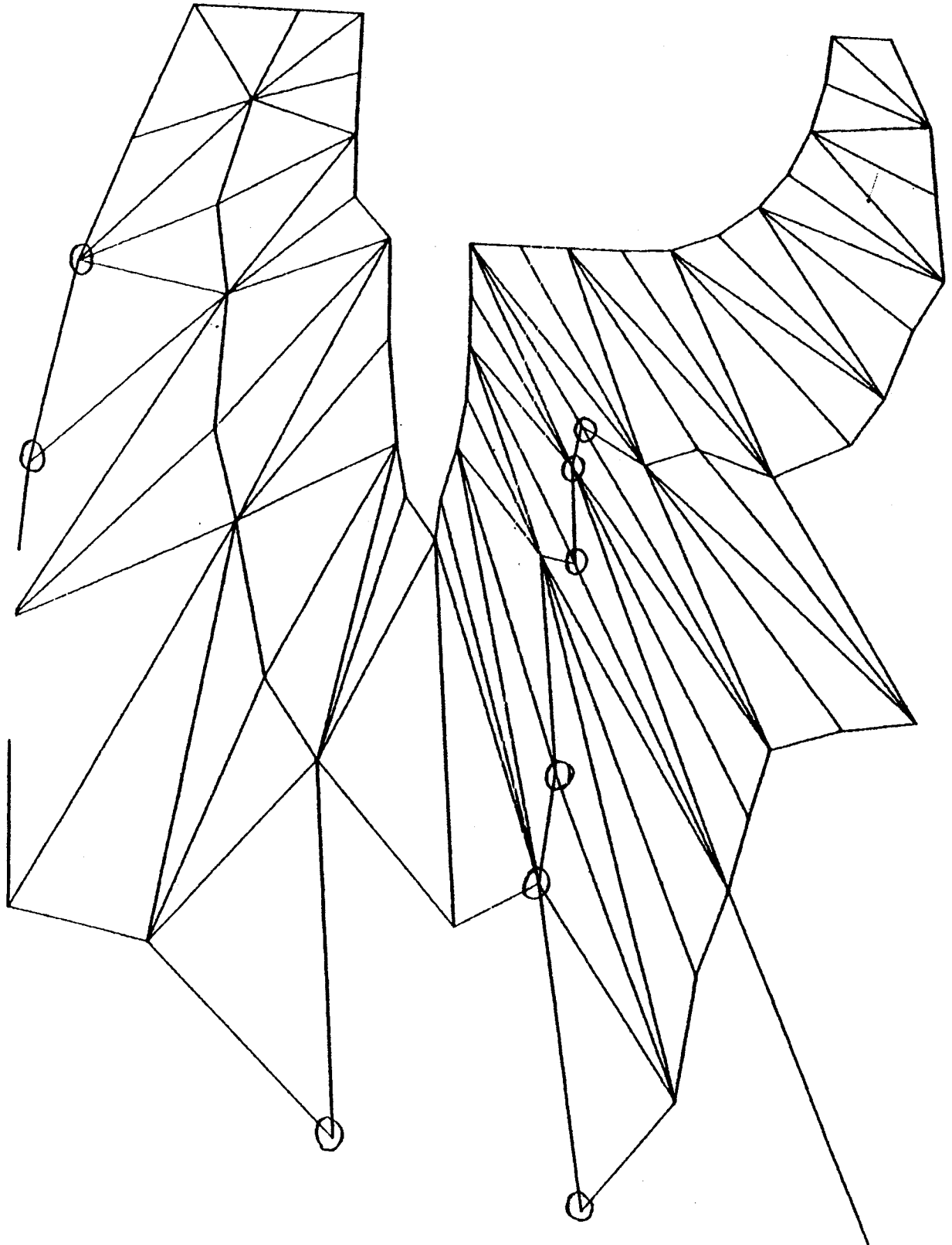
Premier agrandissement du maillage initial



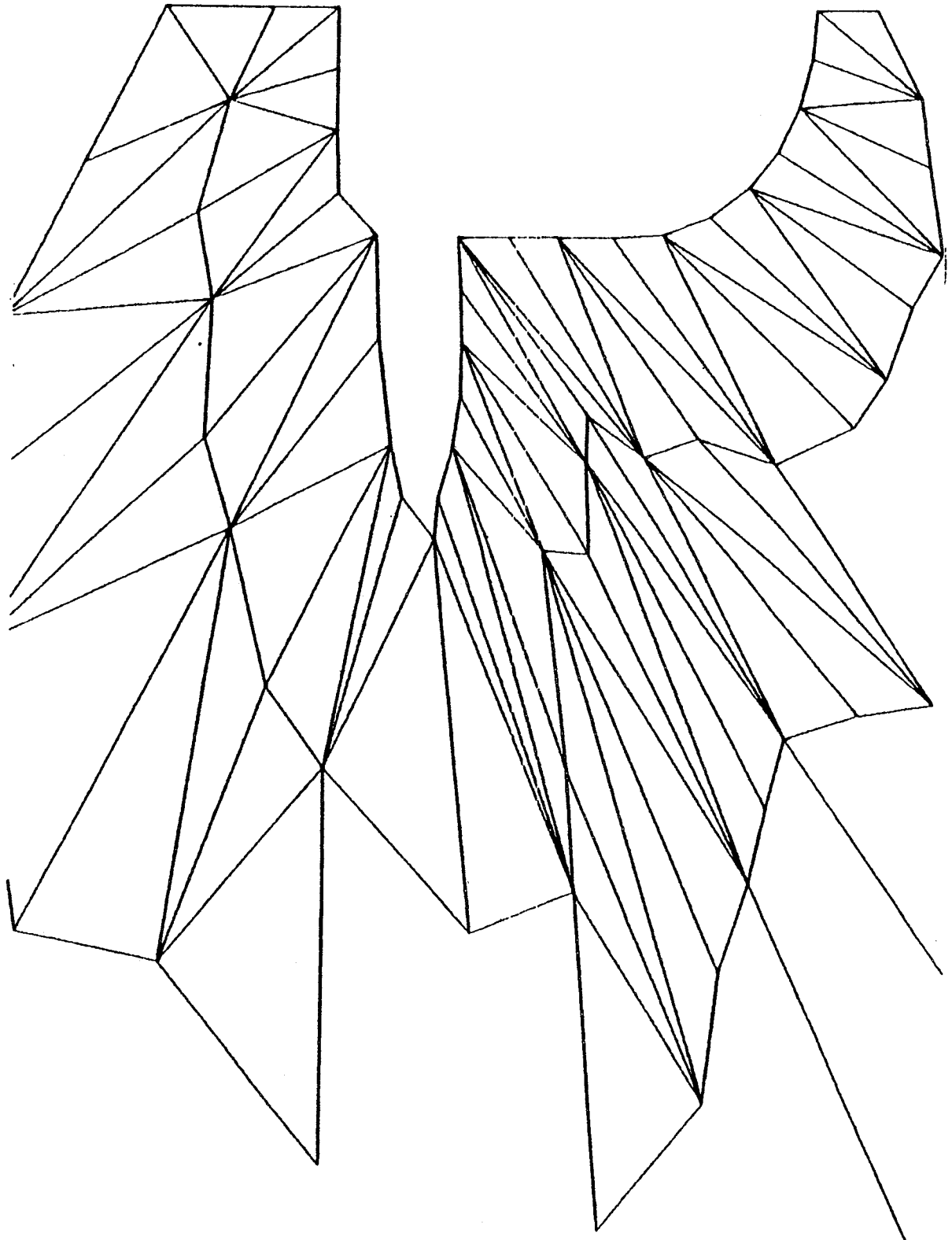
Premier agrandissement du maillage final



Deuxième agrandissement du maillage initial



Deuxième agrandissement du maillage final



6. CONCLUSIONS

L'objectif de ce travail a été d'établir une méthode systématique et algorithmique qui peut servir comme une étape dans le cadre du problème général d'obtention d'un maillage de types éléments finis et qui permet d'englober un certain nombre des méthodes connues pour résoudre ce problème.

Nous avons traité le problème de l'amélioration de maillage par des méthodes d'optimisation classiques en recherche opérationnelle connues sous le nom des méthodes de sous-gradient. Nos algorithmes consistent à trouver une position nouvelle pour un noeud, en fixant tous les autres, sans remettre en cause la topologie existante, afin d'améliorer le maillage local, et si possible, global selon certains critères d'optimisation liés au volumes. On déplace le noeud libre dans la direction d'amélioration fournie par l'opposé du gradient de la fonction critère avec un pas bien choisi.

Nous avons constaté que les algorithmes proposés ont l'avantage d'être rapide au moins dans les premières itérations, c'est à dire de donner rapidement une bonne solution, même s'ils ne permettent pas d'obtenir une solution optimale pour le problème de minimisation d'un critère donné.

Notre souci était d'établir un outil efficace à partir de ces méthodes et aussi de trouver un compromis entre la théorie et les applications industrielles. Ce travail montre donc la possibilité d'appliquer des méthodes d'optimisation pour répondre au problème de maillage.

Dans le même esprit, nous pouvons envisager deux directions possibles pour de futures recherches :

- (1) la prise en compte dans nos méthodes d'autres critères géométriques et de critères physiques en relation avec le problème étudié,
- (2) une variation des méthodes proposées qui permettrait de bouger plusieurs noeuds simultanément.

ANNEXES INFORMATIQUES

Annexe 1 : Description des Variables et Programmes Principaux Utilisés dans les Programmes Informatiques de l'ALGO1 et de l'ALGOTOUT en dimension 3 (en dimension 2)

Les coordonnées de tous les noeuds de la région considérée sont stockées dans XYZ (XY).

Les éléments de maillage décrits par leurs sommets sont stockés dans SOMTET (SOMRI).

Le noeud à bouger est stocké dans SPPT (SPPT).

La fonction objective à minimiser est stockée dans NBCRI (NBCRI).

Le valeur de ϵ pour trouver le pas est stockée dans XCOEF (XCOEF).

Le programme pour bouger le noeud SPPT est MOVEPT (MVPT2).

Le sous-programme pour trouver les éléments entourant le noeud SPPT, leurs volumes (superficies) initiales, et les premiers élément ayant le volume (superficie) max et min, respectivement, est AREA (AREA2).

Le sous-programme pour trouver la valeur des mineurs du déterminant utilisé pour trouver les volumes (superficies) initiales des éléments est XNOR (XNOR2).

Le sous-programme qui décrit la fonction objective choisie est XCRIT (XCRIT2). Ce sous-programme a besoin de NBCRI.

Le sous-programme qui calcule la direction d'amélioration est DIREC (DIREC2). Ce sous-programme a besoin de NBCRI.

Le sous-programme pour calculer les coefficients de ρ dans l'application $V_i(P_0 + \rho\delta)$ affine en ρ est XFIVE (XFIVE2).

Le sous-programme pour trouver le pas à prendre dans la direction d'amélioration est NEWEPS (NEWEPS2). Ce sous-programme a besoin de XCOEF.

AMETRI est le programme de l'ALGO1 en dimension 2. Ce programme appelle directement MVPT2 et dépend de NBCRI, XCOEF, ET SPPT.

Les descriptions qui suivent ne concernent que l'ALGOTOUT.

Le nombre d'itérations successives pour bouger un noeud est CNT, et, ici, CNT=3.

Le nombre de fois qu'on bouge tout l'ensemble des noeuds est TOUR, et, ici, TOUR=3.

Le choix de l'ordre dans lequel on bouge les noeuds se trouve dans BWST; BWST = 0 si on choisit le sens "pire au meilleur", et 1 si on choisit le sens "meilleur au pire". On peut aussi

décider si on suit le même sens ou si on alterne le sens à chaque tour, et on peut construire d'autres sous-programmes pour d'autres ordres possibles dans lesquels on désire bouger nos noeuds à chaque tour.

Le calcul des valeurs initiales et finales de la fonction objective et du pourcentage d'amélioration est faite dans un sous-programme CRIT (CRIT2).

AMTETCHX (ATRCHX) est le programme de l'ALGOTOUT en dimension 3 (en dimension 2). Ce programme appelle directement MOVEPT (MVPT2) et CRIT(CRIT2) et dépend de NBCRI, et BWST.

Annexe 2.1 Programmes de l'ALGO1 en dimension 2

```

        subroutine movept2(sppt)
%include data2
c
        call area2(sppt)
        abc=4
        xd1=xcrit2(abc)
        print, "the initial criterion value"
        write(6,*) xd1
        call direc2
        call xfive2
        call neweps2
        do 20 i=1,nvoit
        xtri(i,6)=xtri(i,4)+xeps1*xtri(i,5)
20      continue
        do 50 i=1,2
50      xspt(i)=xeps1*xspt(i)
        do 60 i=1,2
60      xy(sppt,i)=xy(sppt,i)+xspt(i)
        print, "the new criterion value"
        abc=6
        xd2=xcrit2(abc)
        write(6,*) xd2
        print, "the new areas of the triangles"
        write(6,*) (xtri(i,6),i=1,nvoit)
        print, "the new coordinates of our point"
        write(6,*) (xy(sppt,i),i=1,2)
130     return
        end

        subroutine area2(sppt)
c
c        we want to find the triangles surrounding a point, their initial
c        areas; we fill in the first 2 columns of tri with the vertices of
c        the opposite edge, the 3rd is the triangle number.
c
%include data2
c
        print, "the three minors and the initial area"
        j=0
        do 80 i=1,nbt
        do 5 k=1,3
        if (somri(i,k).eq.sppt) go to 10
5      continue
        go to 80
10     j=j+1
        m=1
        l=1
12     if (m.eq.4) go to 20
        if*(m.eq.k) go to 15
        tri(j,1)=somri(i,m)
        l=l+1
15     m=m+1
        go to 12
20     tri(j,3)=i
c
c        the area of triangle pqr is defined as the determinant
c        of the 3*3 matrix shown below; we use xab to store the determi-
c        nant minors of this matrix

```

```

c          1  1  1      where p(x,y) and edge qr form our triangle
c          x q1 r1      we'll put in advance the proper sign of each
c          y q2 r2      minor, i.e., +--, (respectively for 1,x,y)
c          stored in xtri(i,1-3); note that we have a subroutine xnor2 to
c          find the determinant of a 2*2 matrix (here,xab)
c
c          do 30 k=1,2
c          do 25 l=1,2
c          xab(1,k)=xy(tri(j,k),1)
25      continue
30      continue
c          xtri(j,1)=xnor2(xab)
c          do 40 l=1,2
40      xab(1,l)=1.0
c          xtri(j,2)=-xnor2(xab)
c          do 50 l=1,2
50      xab(2,l)=xy(tri(j,l),1)
c          xtri(j,3)=xnor2(xab)
c
c          xtri(j,4) contains the present area of the jth triangle,
c          and there are nvoit triangles.
c          if xtri(j,4) is negative, we change all the signs of all minors
c          as well as that of xtri(j,4).
c
c          xtri(j,4)=xtri(j,1)
c          do 70 k=1,2
70      xtri(j,4)=xtri(j,4)+(xy(sppt,k)*xtri(j,k+1))
c          if (xtri(j,4).lt.0.0) then
c          do 75 k=1,4
75      xtri(j,k)=-xtri(j,k)
c          end if
c          do 77 k=1,4
77      write(6,*) (xtri(j,k),k=1,4)
80      continue
c          nvoit=j
c
c          among our nvoit triangles, we'll choose that with maximum
c          area(vmax) and that with minimum area(vmin)
c
c          if (xtri(1,4).ge.xtri(2,4)) then
c          vmax=1
c          vmin=2
c          else
c          vmax=2
c          vmin=1
c          end if
c          do 90 i=3,nvoit
c          if (xtri(i,4).gt.xtri(vmax,4)) then
c          vmax=i
c          else
c          if (xtri(i,4).lt.xtri(vmin,4)) vmin=i
c          end if
90      continue
c          print, "the initial vmax and vmin"
c          write(6,*) vmax,vmin
c          return
c          end

```

```

function xnor2(xba)
c   this function finds the value of a 2*2 determinant
c
dimension xba(2,2)
c
xnor2=xba(1,1)*xba(2,2)-xba(1,2)*xba(2,1)
return
end

function xcrit2(cab)
c   this function is the criterion to be used
%include data2
c
if (nbcri.eq.1) then
xcrit2=1.0-xtri(vmin,cab)/xtri(vmax,cab)
else
if (nbcri.eq.2) then
xcrit2=xtri(vmax,cab)-xtri(vmin,cab)
print, "the criterion used is vmax-vmin"
else
if (nbcri.eq.3) then
xv=0.0
do 10 i=1,nvoit
10  xv=xv+xtri(i,cab)
xv=xv/nvoit
xcrit2=0.0
do 20 i=1,nvoit
yv=xtri(i,cab)-xv
if (yv.lt.0.0) yv=-yv
20  xcrit2=xcrit2+yv
xcrit2=xcrit2/nvoit
end if
end if
end if
return
end

subroutine xfive2
c   we compute xtri(i,5) in this subroutine
c
%include data2
do 20 i=1,nvoit
xtri(i,5)=0.0
do 10 j=1,2
10  xtri(i,5)=xtri(i,5)+xspt(j)*xtri(i,j+1)
20  continue
print, "the area linear equations as a function of the step"
print, "the constant and the slope"
do 30 i=1,nvoit
30  write(6,*) (xtri(i,j),j=4,5)
return
end

```

```

      subroutine direc2
c      this subroutine determines the direction vector using the subgradient
c      method
c
c      %include data2
c
      if (nbcri.eq.1) then
      do 10 k=2,3
10      xspt(k-1)=-((xtri(vmin,4)*xtri(vmax,k)-xtri(vmax,4)*xtri(vmin,k))/
      &              (xtri(vmax,4)**2))
      else
      if (nbcri.eq.2) then
      do 20 k=2,3
20      xspt(k-1)=-((xtri(vmax,k)-xtri(vmin,k))
      else
      if (nbcri.eq.3) then
      xv=0.0
      yv=0.0
      zv=0.0
      do 30 i=1,nvoit
      xv=xv+xtri(i,2)
      yv=yv+xtri(i,3)
30      zv=zv+xtri(i,4)
      xv=xv/nvoit
      yv=yv/nvoit
      zv=zv/nvoit
      xspt(1)=0.0
      xspt(2)=0.0
      do 40 i=1,nvoit
      xvl=xtri(i,2)-xv
      yvl=xtri(i,3)-yv
      if ((xtri(i,4)-zv).lt.0.0) then
      xvl=-xvl
      yvl=-yvl
      end if
      xspt(1)=xspt(1) - xvl
40      xspt(2)=xspt(2) - yvl
      end if
      end if
      end if
      print, "the gradient vector to be used"
      write(6,*) (xspt(k),k=1,2)
      return
      end

```

```

      subroutine neweps2
c      we look for the (xeps,min(vmax-vmin)) and the xeps1 a function of
c      of xeps and xcoef
%include data2
c      we look for the intersections at max and min, and the increasing
c      values of xeps at these intersections
c
      print, "the intermediate step values with corresponding vmax,vmin"
      xdif=xtri(vmax,4)-xtri(vmin,4)
      xeps=0.0
      vmax1=vmax
      vmin1=vmin
      vmax3=vmax
      vmin3=vmin
      first=0
8      if (first.eq.2) go to 12
      xmx=1.e15
      do 10 i=1,nvoit
      ya=xtri(i,4)-xtri(vmax1,4)
      yb=xtri(vmax1,5)-xtri(i,5)
      if ((ya*yb).le.0.0) go to 10
      yab=ya/yb
      if ((xeps.lt.yab).and.(yab.lt.xmx)) then
      xmx=yab
      vmax2=i
      end if
10     continue
12     if (first.eq.1) go to 17
      xmn=1.e15
      do 15 i=1,nvoit
      za=xtri(i,4)-xtri(vmin1,4)
      zb=xtri(vmin1,5)-xtri(i,5)
      if ((za*zb).le.0.0) go to 15
      zab=za/zb
      if ((xeps.lt.zab).and.(zab.lt.xmn)) then
      xmn=zab
      vmin2=i
      end if
15     continue
17     if ((xmx.eq.1.e15).and.(xmn.eq.1.e15)) go to 70
      vmax3=vmax1
      vmin3=vmin1
      if (xmx-xmn) 20,20,30
20     vmax1=vmax2
      first=1
      if (xmx.lt.xmn) go to 35
      first=0
30     vmin1=vmin2
      if (xmx.gt.xmn) first=2
35     if (xmx.ge.xmn) then
      xeps1=xmn

```



```

else
xeps1=xmx
end if
write(6,*) xeps1,vmax3,vmin3
C when smallest volume is negative
if (xtri(vmin1,4)+xeps1*xtri(vmin1,5).le.0.0) go to 75
xdif1=xtri(vmax1,4)-xtri(vmin1,4)+xeps1*(xtri(vmax1,5)-xtri(vmin1,5))
if (xdif1.gt.xdif) go to 60
xdif=xdif1
xeps=xeps1
go to 8
60 print, "this is the optimal step taken"
write(6,*) xeps
print, "this is the next step"
write(6,*) xeps1
xeps1=xeps+(xcoef)*(xeps1-xeps)
ginsid=.false.
go to 80
70 vmax3=vmax1
vmin3=vmin1
75 if (xeps.eq.0.0) ginsid=.true.
print, "the optimal step taken with no next step"
write(6,*) xeps
xeps1=xeps*(1+0.1*xcoef)
80 vmax=vmax3
vmin=vmin3
xdif2=xtri(vmax,4)-xtri(vmin,4)+xeps1*(xtri(vmax,5)-xtri(vmin,5))
print, "this is the real step taken"
write(6,*) xeps1
print, "the new vmax,vmin"
write(6,*) vmax,vmin
return
end

```

```

subroutine ametri
#include data2
call stri2
print, "criterion number is"
read(5,*) nbcri
print, "coefficient number is"
read(5,*) xcoef
spct=6
print, "the point to be moved is"
read(5,*) spct
call mvpt2(spct)
return
end

```

```

        subroutine stri2
c        structure of input files (point,triangle)
c
c        %include data2
c
c        character*12 efipt,efitet
c
c        read into xy data from entry point file
c        print, "entry point file is"
c        read(5,*) efipt
c        open(10,file=efipt,form='formatted')
10      read(10,*,end=20) nbp,(xy(nbp,i),i=1,2)
c        write(6,*) nbp,(xy(nbp,i),i=1,2)
c        go to 10
20      close(10)
c
c        read into somri data from entry triangle file
c        print, "entry triangle file is"
c        read(5,*) efitet
c        open(10,file=efitet,form='formatted')
50      read(10,*,end=60) nbt,(somri(nbt,i),i=1,3)
c        write(6,*) nbt,(somri(nbt,i),i=1,3)
c        go to 50
60      close(10)
c        return
c        end

        subroutine stro2
c        structure of output(point,criterion value)
c
c        %include data2
c        write into exit point file data from xy with pt number indicated
c        print, "exit point file"
c        do 10 i=1,nbp
c        write(6,*) i,(xy(i,j),j=1,2)
10      continue
c        return
c        end

```

data2.incl.fortran

```

c      the letters xyz for real, g for logical, the rest integer
c      implicit integer (a-f,h-w),logical (g)
c      coordinates of a pt
c      common/point2/xy(100,2),nbp,nbpmax,nbi,nsp,xy2(100,5),ixy(100,2)
c      vertices of a boundary edge
c      common/edge2/someg(50,2),nbe,nbemax
c      vertices of a triangle
c      common/tri2/somri(150,3),nbt,nbtmax
c      coordinates of a special point
c      vertices of a special edge
c      vertices of a special triangle
c      a logical indicating whether it's inside or not
c      common/spec2/xspt(2),speg(2),spri(3),ginsid
c      for use in programs included in ametri.fortran and amtrchoix.fortran
c      common/mpt2/tri(80,3),xab(2,2),xtri(80,7),nvoit,nbcri,xepsi,count,
&      xnbl,coef,vmin,vmax,xcoef,col,ptset,ptnb(100),nptnb,cnt,xquo,xsum

```

Annexe 2.2 Programmes de l'ALGOTOUT en dimension 2

(cf Annexe 2.1 pour les sous-programmes AREA2, DIREC2, NEWEPS2, STRI2, XCRIT2, XFIVE2, XNOR2)

```

      subroutine atrchx
%include data2
      call stri2
      print, "criterion number is"
      read(5,*) nbcri
      write(6,*) nbcri
      print, "best to worst?"
      read(5,*) bwst
      write(6,*) bwst
      xcoef=0.01
      cnt=3
      tour=3
      col=1
      xquo=0.0
      do 10 i=1,nbp
      spct=i
      call crit2(spct)
      xquo=xquo+xy2(i,1)
10      continue
      call prepsurf2
      do 15 i=1,nbi
15      xy2(i,4)=xy2(ixy(i,2),1)
      call tricrit2
      if (bwst.eq.1) then
      fst=nbi
      sec=1
      thd=-1
      else
      fst=1
      sec=nbi
      thd=1
      end if
17      do 30 j=1,tour
      do 20 i=fst,sec,thd
      spct=ixy(i,2)
      call movept2(spct)
20      continue
      do 25 i=1,nbi
      spct=ixy(i,2)
      col=5
      call crit2(spct)
      xy2(i,4)=xy2(spct,5)
25      continue
      call tricrit2
      col=2
      xsum=0.0
      do 45 i=1,nbp
      spct=i
      call crit2(spct)
      xy2(i,3)=xy2(i,1)-xy2(i,2)
      xsum=xsum+xy2(i,3)
45      continue
      xsum=xsum/xquo
      call stro2
30      continue
      return
      end

```

```

subroutine crit2(sppt)
c
c we want to find the triangles surrounding a point and their
c areas; we fill in the first 2 columns of tri with the vertices of
c the opposite edge, the 3rd is the triangle number.
c we're also interested in the criterion value depending on nbcri.
c
%include data2
c
call area2(sppt)
abc=4
xd1=xcrit2(abc)
xy2(sppt,col)=xd1
return
end

```

data2.incl.fortran

```

c the letters xyz for real, g for logical, the rest integer
c implicit integer (a-f,h-w),logical (g)
c coordinates of a pt
c common/point2/xy(200,2),nbp,nbpmax,nbi,nsp,xy2(200,5),ixy(200,2)
c vertices of a boundary edge
c common/edge2/someg(50,2),nbe,nbemax
c vertices of a triangle
c common/tri2/somri(250,3),nbt,nbtmax
c coordinates of a special point
c vertices of a special edge
c vertices of a special triangle
c a logical indicating whether it's inside or not
c common/spec2/xspt(2),speg(2),spri(3),ginsid
c for use in programs included in ametri.fortran and amtrchoix.fortran
c common/mpt2/tri(80,3),xab(2,2),xtri(80,7),nvoit,nbcri,xeps1,count,
& xnb1,coef,vmin,vmax,xcoef,col,ptset,ptnb(100),nptnb,cnt,xquo,xsum

```

```

subroutine movept2(sppt)
c
c   we want to reposition a point without changing its neighbors;
c   in order to optimize a certain criterion (nbcrit) by using the
c   subgradient method.
c
%include data2
c
   call area2(sppt)
   abc=4
   xd1=xcrit2(abc)
c   we want to reduce the gap between the areas ;
c   the direction vector xspt is used in v(i,xspt,xeps)=xtri(i,4)
c   +xeps*(xspt(1)*xtri(i,2)+xspt(2)*xtri(i,3))
c   and xspt is found in the subroutine direc.
c
   do 80 jj=1,cnt
10  call direc2
c
c   volume is now simply xtri(i,4)+xeps*xtri(i,5) and xtri(i,5) will
c   be computed in the subroutine xfive
   call xfive2
c
c   on r*r, we plot the functions f(xeps)=a+b*xeps for each triangle
c   and we find the xeps value giving the minimum difference between
c   vmax and vmin in the subroutine eps; moreover, we look for the xeps!
c   value depending on the xcoef.
c
   call neweps2
c
   if (ginsid) go to 130
   xtri(vmin,6)=xtri(vmin,4)+xeps1*xtri(vmin,5)
   if (xtri(vmin,6).le.0.0) go to 130
   do 20 i=1,nvoit
20  xtri(i,6)=xtri(i,4)+xeps1*xtri(i,5)
   continue
   abc=6
   xd3=xcrit2(abc)
c   xspt will now contain our real gradient vector
   do 50 i=1,2
50  xspt(i)=xeps1*xspt(i)
c   we'll move our pt in the direction of xspt
   do 60 i=1,2
60  xy(sppt,i)=xy(sppt,i)+xspt(i)
c   we'll revise the new areas in xtri(i,4)
   do 70 i=1,nvoit
70  xtri(i,4)=xtri(i,6)
80  continue
130 return
end

```

```

      subroutine prepsurf2
c      prepares the use of the file of boundary points in ametri
%include data2
      character*12 bndpt
c
      do 10 i=1,nbp
10      ixy(i,1)=0
      print, "boundary point file is"
      read(5,*) bndpt
      write(6,*) bndpt
      open(10,file=bndpt,form='formatted')
20      read(10,*,end=30) i
      ixy(i,1)=1
      go to 20
30      close(10)
      j=0
      do 40 i=1,nbp
      if (ixy(i,1).eq.0) then
      j=j+1
      ixy(j,2)=1
      end if
40      continue
      nbi=j
      return
      end

      subroutine stro2
c      structure of output(point,criterion value)
c
%include data2
      character*12 sfipt,sfitet
c
c      write into exit point file data from xy with pt number indicated
      read(5,*) sfipt
      open(10,file=sfipt,form='formatted')
      do 10 i=1,nbp
10      write(10,*) i,(xy(i,j),j=1,2)
      continue
      close(10)
      print, "the initial and intermediate global criterion values"
      print, "and the global improvement at this moment"
      write(6,*) xquo,xquo*xsum,xsum
      return
      end

```

```
      subroutine tricrit2
c      arranges xy2(*,2) in decreasing order and subsequently ixy(*,2)
%include data2
c
      5      fini=0
           do 10 i=1,nbi-1
           x=xy2(i,4)
           y=xy2(i+1,4)
           if (x.lt.y) then
           xy2(i,4)=y
           xy2(i+1,4)=x
           j=ixy(i,2)
           ixy(i,2)=ixy(i+1,2)
           ixy(i+1,2)=j
           fini=1
           end if
      10      continue
           if (fini.eq.1) go to 5
           return
           end
```


Annexe 3 Programmes de l'ALGOTOUT en dimension 3

```

subroutine area(sppt)
c
c   we want to find the tetrahedrons surrounding a point, their initial
c   areas; we fill in the first 3 columns of tat with the vertices of
c   the opposite triangle, the 4th is the tet nb.
c
%include data3
c
      j=0
      do 80 i=1,nbtet
      do 5 k=1,4
      if (sometet(i,k).eq.sppt) go to 10
5      continue
      go to 80
10     j=j+1
      m=1
      l=1
12     if (m.ε3.5) go to 20
      if (m.eq.k) go to 15
      tat(j,l)=sometet(i,m)
      l=l+1
15     m=m+1
      go to 12
20     tat(j,4)=i
c
c   the volume of tetrahedron pqrs is defined as the determinant
c   of the 4*4 matrix shown below; we use xa to store the determi-
c   nant minors of this matrix
c       1   1   1   1   where p(x,y,z) and triangle qrs form
c       x q1 r1 s1   our tetrahedron; we'll put in advance
c       y q2 r2 s2   the proper sign of each minor, i.e., +---,
c       z q3 r3 s3   (respectively, for 1,x,y,z) stored in
c   xtat(i,1-4); note that we have a special subroutine xnor to
c   find the determinant of a 3*3 matrix (here,xa)
c
      do 30 k=1,3
      do 25 l=1,3
      xa(1,k)=xyz(tat(j,k),1)
25     continue
      continue
30     xtat(j,1)=xnor(xa)
      do 40 l=1,3
40     xa(1,l)=1.0
      xtat(j,2)=-xnor(xa)
      do 50 l=1,3
50     xa(2,l)=xyz(tat(j,l),1)
      xtat(j,3)=xnor(xa)
      do 60 l=1,3
60     xa(3,l)=xyz(tat(j,l),2)
      xtat(j,4)=-xnor(xa)
c

```

```

c      xtat(j,5) contains the present volume of the jth tetrahedron,
c      and there are nvoit tetrahedrons.
c      if xtat(j,5) is negative, we change all the signs of all minors
c      as well as that of xtat(j,5).
c
c      xtat(j,5)=xtat(j,1)
c      do 70 k=1,3
70     xtat(j,5)=xtat(j,5)+(xyz(sppt,k)*xtat(j,k+1))
c      if (xtat(j,5).lt.0.0) then
c      do 75 k=1,5
75     xtat(j,k)=-xtat(j,k)
c      end if
c      continue
80     nvoit=j
c
c      among our nvoit tetrahedrons, we'll choose that with maximum
c      volume(vmax) and that with minimum volume(vmin)
c
c      if (xtat(1,5).ge.xtat(2,5)) then
c      vmax=1
c      vmin=2
c      else
c      vmax=2
c      vmin=1
c      end if
c      do 90 i=3,nvoit
c      if (xtat(i,5).gt.xtat(vmax,5)) then
c      vmax=i
c      else
c      if (xtat(i,5).lt.xtat(vmin,5)) vmin=i
c      end if
90     continue
c      return
c      end

subroutine crit(sppt)
c
c      we want to find the tetrahedrons surrounding a point, their initial
c      areas; we fill in the first 3 columns of tat with the vertices of
c      the opposite triangle, the 4th is the tet nb.
c      we're also interested in the criterion value dependent on nbcri.
c
%include data3
c
c      call area(sppt)
c      abc=5
c      xd1=xcrit(abc)
c      xyz2(sppt,col)=xd1
c      return
c      end

```

data3.incl.fortran

```

c      the letters xyz for real, g for logical, the rest for integer
c      implicit integer (a-f,h-w),logical (g)
c      three coordinates of a point
c      common/point/xyz(200,3),nbp,nbpmax,nsp,nbi,xyz2(200,5),ixyz(200,2)
c      the three vertices of a triangle
c      common/tria/somtri(100,3),nbt,nbtmax
c      the four vertices of a tetrahedron
c      common/tetra/somt(800,4),nbtet,ntetmx
c      3 coordinates of one single special point
c      3 vertices of one single special triangle
c      4 vertices of one single special tetrahedron
c      a logical indicating whether it's inside or not
c      common/spec/xsppt(3),sptri(3),sptet(4),ginsid
c      for use in programs included in ametet.fortran and amtetchx.fortran
&     common/mpt/tat(100,4),xa(3,3),xtat(100,8),nvoit,nbcric,xepsi,count
&     xnbl,coef,vmin,vmax,xcoef,col,ptset,ptnb(150),nptnb,cnt,xquo,xsum

```

```

c      function xnor(xb)
c      this function finds the value of a 3 by 3 determinant
c      dimension xb(3,3),xpha(3)
c      m=1
c      k=2
c      l=3
5     xpha(m)=xb(m,1)*(xb(k,2)*xb(l,3)-xb(l,2)*xb(k,3))
c      if (m.eq.3) go to 30
c      if (k.eq.2) go to 10
c      if (k.eq.1) go to 20
10    k=1
c      m=2
c      go to 5
20    l=2
c      m=3
c      go to 5
30    xnor=(xpha(1)-xpha(2))+xpha(3)
c      return
c      end

```

```

subroutine direc
c      this subroutine determines the direction vector using the subgradient
c      method
c
%include data3
c
      if (nbcri.eq.1) then
do 10 k=2,4
10      xsppt(k-1)=-((xtat(vmin,5)*xtat(vmax,k)-xtat(vmax,5)*xtat(vmin,k))/
&          (xtat(vmax,5)**2))
      else
      if (nbcri.eq.2) then
do 20 k=2,4
20      xsppt(k-1)=- (xtat(vmax,k)-xtat(vmin,k))
      else
      if (nbcri.eq.3) then
xv=0.0
yv=0.0
zv=0.0
xyv=0.0
do 30 i=1,nvoit
xv=xv+xtat(i,2)
yv=yv+xtat(i,3)
zv=zv+xtat(i,4)
30      xyv=xyv+xtat(i,5)
xv=xv/nvoit
yv=yv/nvoit
zv=zv/nvoit
xyv=xyv/nvoit
xsppt(1)=0.0
xsppt(2)=0.0
xsppt(3)=0.0
do 40 i=1,nvoit
xv1=xtat(i,2)-xv
yv1=xtat(i,3)-yv
zv1=xtat(i,4)-zv
if ((xtat(i,5)-xyv).lt.0.0) then
xv1=-xv1
yv1=-yv1
zv1=-zv1
end if
40      xsppt(1)=xsppt(1) - xv1
xsppt(2)=xsppt(2) - yv1
xsppt(3)=xsppt(3) - zv1
end if
end if
end if
return
end

```

```

subroutine movept(sppt)
c
c   we want to reposition a point without changing its neighbors;
c   in order to optimize a certain criterion (nbcrit) by using the
c   subgradient method.
c
%include data3
c
c   call area(sppt)
c   abc=5
c   xd1=xcrit(abc)
c   we want to reduce the gap between the volumes ;
c   the direction vector xsppt is used in v(i,xsppt,xeps)=xtat(i,5)
c   +xeps*(xsppt(1)*xtat(i,2)+xsppt(2)*xtat(i,3)+xsppt(3)*xtat(i,4))
c   and xsppt is found in the subroutine direc.
c
c   do 80 jj=1,cnt
10  call direc
c
c   volume is now simply xtat(i,5)+xeps*xtat(i,6) and xtat(i,6) will
c   be computed in the subroutine xfive
c   call xfive
c
c   on r*r, we plot the functions f(xeps)=a+b*xeps for each tetrahedron
c   and we find the xeps value giving the minimum difference between
c   vmax and vmin in the subroutine neweps; moreover, we look for the
c   xeps1 value depending on the xcoef.
c
c   call neweps
c
c   if (ginsid) go to 130
c   xtat(vmin,7)=xtat(vmin,5)+xeps1*xtat(vmin,6)
c   if (xtat(vmin,7).le.0.0) go to 130
c   do 20 i=1,nvoit
20  xtat(i,7)=xtat(i,5)+xeps1*xtat(i,6)
c   continue
c   abc=7
c   xd3=xcrit(abc)
c   xsppt will now contain our real gradient vector
c   do 50 i=1,3
50  xsppt(i)=xeps1*xsppt(i)
c   we'll move our pt in the direction of xsppt
c   do 60 i=1,3
60  xyz(sppt,i)=xyz(sppt,i)+xsppt(i)
c   we'll revise the new volumes in xtat(i,5)
c   do 70 i=1,nvoit
70  xtat(i,5)=xtat(i,7)
80  continue
130 return
end

```

```

      subroutine stri
c      structure of input files (point,tetrahedron)
c
c      %include data3
c
c      character*12 efipt,efitet
c
c      read into xyz data from entry point file
c      print, "name of entry point file is"
c      read(5,*) efipt
c      write(6,*) efipt
c      open(10,file=efipt,form='formatted')
10      read(10,*,end=20) nbp,(xyz(nbp,i),i=1,3)
c      go to 10
20      close(10)
c
c      read into somtet data from entry tetrahedron file
c      print,"name of entry tetrahedron file is"
c      read(5,*) efitet
c      write(6,*) efitet
c      open(10,file=efitet,form='formatted')
50      read(10,*,end=60) nbtet,(sometet(nbtet,i),i=1,4)
c      go to 50
60      close(10)
c
c      return
c      end

      subroutine stro
c      structure of output(point,criterion value)
c
c      %include data3
c
c      character*12 sfipt,sfitet
c
c      write into exit point file data from xyz with pt number indicated
c      read(5,*) sfipt
c      open(10,file=sfipt,form='formatted')
c      do 10 i=1,nbp
c      write(10,*) i,(xyz(i,j),j=1,3)
10      continue
c      close(10)
c
c      print, "the initial and intermediate global criterion values"
c      print, "and the global improvement value at this moment"
c      write(6,*) xquo,xquo*xsum,xsum
c
c      return
c      end

```

```

        function xcrit(cab)
c      this function is the criterion to be used
%include data3
c
        if (nbcrit.eq.1) then
xcrit=1.0-xtat(vmin,cab)/xtat(vmax,cab)
        else
        if (nbcrit.eq.2) then
xcrit=xtat(vmax,cab)-xtat(vmin,cab)
        else
        if (nbcrit.eq.3) then
xcv=0.0
        do 10 i=1,nvoit
. 10  xcv=xcv+xtat(i,cab)
        xcv=xcv/nvoit
xcrit=0.0
        do 20 i=1,nvoit
        yv=xtat(i,cab)-xcv
        if (yv.lt.0.0) yv=-yv
20  xcrit=xcrit+yv
        xcrit=xcrit/nvoit
        end if
        end if
        end if
        return
        end

```

```

        subroutine xfive
c      we compute xtat(i,6) in this subroutine
c
%include data3
        do 20 i=1,nvoit
        xtat(i,6)=0.0
        do 10 j=1,3
10  xtat(i,6)=xtat(i,6)+xsppt(j)*xtat(i,j+1)
20  continue
        return
        end

```

```

subroutine neweps
c      we look for the (xeps,min(vmax-vmin)) and the xeps1 a function of
c      of xeps and xcoef
%include data3
c      the initial vmax and vmin may have others at the max and min values,
c      so we have to choose that for which the slope is biggest for the max,
c      and the smallest for the min
c
      if (count.eq.0) then
      do 7 i=1,nvoit
      if (xtat(i,5).lt.xtat(vmax,5)) go to 3
      if (xtat(i,6).gt.xtat(vmax,6)) vmax=i
      go to 7
3     if (xtat(i,5).gt.xtat(vmin,5)) go to 7
      if (xtat(i,6).lt.xtat(vmin,6)) vmin=i
7     continue
      end if
c
c      we now look for the intersections at max and min, and the increasing
c      values of xeps at these intersections
c
      xdif=xtat(vmax,5)-xtat(vmin,5)
      xeps=0.0
      vmax1=vmax
      vmin1=vmin
      vmax3=vmax
      vmin3=vmin
      first=0
8     if (first.eq.2) go to 12
      xmx=1.e15
      do 10 i=1,nvoit
      ya=xtat(i,5)-xtat(vmax1,5)
      yb=xtat(vmax1,6)-xtat(i,6)
      if ((ya*yb).le.0.0) go to 10
      yab=ya/yb
      if ((xeps.lt.yab).and.(yab.lt.xmx)) then
      xmx=yab
      vmax2=i
      end if
10    continue
12    if (first.eq.1) go to 17
      xmn=1.e15
      do 15 i=1,nvoit
      za=xtat(i,5)-xtat(vmin1,5)
      zb=xtat(vmin1,6)-xtat(i,6)
      if ((za*zb).le.0.0) go to 15
      zab=za/zb
      if ((xeps.lt.zab).and.(zab.lt.xmn)) then
      xmn=zab
      vmin2=i
      end if

```



```

15     continue
17     if ((xmx.eq.1.e15).and.(xmn.eq.1.e15)) go to 70
        vmax3=vmax1
        vmin3=vmin1
        if (xmx-xmn) 20,20,30
20     vmax1=vmax2
        first=1
        if (xmx.lt.xmn) go to 35
        first=0
30     vmin1=vmin2
        if (xmx.gt.xmn) first=2
35     if (xmx.ge.xmn) then
        xeps1=xmn
        else
        xeps1=xmx
        end if
c     when smallest volume is negative
        if (xtat(vmin1,5)+xeps1*xtat(vmin1,6).le.0.0) go to 75
        xdif1=xtat(vmax1,5)-xtat(vmin1,5)+xeps1*(xtat(vmax1,6)-xtat(vmin1,6))
        if (xdif1.gt.xdif) go to 60.
        xdif=xdif1
        xeps=xeps1
        go to 8
60     xeps1=xeps+(xcoef)*(xeps1-xeps)
        ginsid=.false.
        go to 80
70     vmax3=vmax1
        vmin3=vmin1
75     if (xeps.eq.0.0) ginsid=.true.
        xeps1=xeps*(1+0.1*xcoef)
80     vmax=vmax3
        vmin=vmin3
        xdif2=xtat(vmax,5)-xtat(vmin,5)+xeps1*(xtat(vmax,6)-xtat(vmin,6))
        return
        end

```

```

subroutine amtetchx
%include data3
  call stri
  print, "criterion number is"
  read(5,*) nbcrit
  write(6,*) nbcrit
  print, "best to worst?"
  read(5,*) bwst
  write(6,*) bwst
  xcoef=0.01
  cnt=3
  tour=3
  col=1
  xquo=0.0
  do 10 i=1,nbp
    spct=i
    call crit(spct)
    xquo=xquo+xyz2(i,1)
10  continue
    call prepsurf
    do 15 i=1,nbi
15  xyz2(i,4)=xyz2(ixyz(i,2),1)
    call tricrit
    if (bwst.eq.1) then
      fst=nbi
      sec=1
      thd=-1
    else
      fst=1
      sec=nbi
      thd=1
    end if
17  do 30 j=1,tour
    do 20 i=fst,sec,thd
      spct=ixyz(i,2)
      call movept(spct)
20  continue
    do 25 i=1,nbi
      spct=ixyz(i,2)
      col=5
      call crit(spct)
      xyz2(i,4)=xyz2(spct,5)
25  continue
      call tricrit
      col=2
      xsum=0.0
      do 45 i=1,nbp
        spct=i
        call crit(spct)
        xyz2(i,3)=xyz2(i,1)-xyz2(i,2)
        xsum=xsum+xyz2(i,3)
45  continue
      xsum=xsum/xquo
      call stro
30  continue
  return
end

```

```

      subroutine tricrit
c      arranges xyz2(*,2) in decreasing order and subsequently ixyz(*,2)
%include data3
c
      5      fini=0
            do 10 i=1,nbi-1
              x=xyz2(i,4)
              y=xyz2(i+1,4)
              if (x.lt.y) then
                xyz2(i,4)=y
                xyz2(i+1,4)=x
                j=ixyz(i,2)
                ixyz(i,2)=ixyz(i+1,2)
                ixyz(i+1,2)=j
                fini=1
              end if
            10      continue
            if (fini.eq.1) go to 5
            return
            end

```

```

      subroutine prepsurf
c      prepares the use of the file of boundary points in amtetchx
%include data3
      character*12 bndpt
c
            do 10 i=1,nbp
      10      ixyz(i,1)=0
            print, "boundary point file is"
            read(5,*) bndpt
            write(6,*) bndpt
            open(10,file=bndpt,form='formatted')
      20      read(10,*,end=30) i
            ixyz(i,1)=1
            go to 20
      30      close(10)
            j=0
            do 40 i=1,nbp
              if (ixyz(i,1).eq.0) then
                j=j+1
                ixyz(j,2)=i
              end if
      40      continue
            nbi=j
            return
            end

```

Annexe 4.1 :

La Déroulement de l'ALGO 1 en Bougeant Une Fois le Noeud 6:

COMMENTAIRES

```

entry point file is PTS
  1 |.00000000E+00 |.00000000E+00
  2 |.40000000E+01 |.00000000E+00
  3 |.00000000E+00 |.30000000E+01
  4 |.20000000E+01 |.40000000E+01
  5 |.50000000E+01 |.20000000E+01
  6 |.10000000E+01 |.10000000E+01
entry triangle file is TRS
  1 | 1 | 3 | 6
  2 | 3 | 4 | 6
  3 | 4 | 5 | 6
  4 | 5 | 2 | 6
  5 | 2 | 1 | 6
criterion number is  $C_1(P_0)$   $v_{max} - v_{min}$ 
coefficient number is  $0,01$ 
the point to be moved is 6
the three minors and the initial area
|.00000000E+00 |.30000000E+01 |.00000000E+00 |.30000000E+01
|.60000000E+01 |.10000000E+01 |-.20000000E+01 |.50000000E+01
|.16000000E+02 |-.20000000E+01 |-.30000000E+01 |.11000000E+02
|.80000000E+01 |-.20000000E+01 |.10000000E+01 |.70000000E+01
|.00000000E+00 |.00000000E+00 |.40000000E+01 |.40000000E+01
the initial vmax and vmin
| 3 | 1 |
the criterion used is vmax-vmin
the initial criterion value
|.80000000E+01|
the gradient vector to be used
|.50000000E+01 |.30000000E+01|
the area linear equations as a function of the step
the constant and the slope
|.30000000E+01 |.15000000E+02
|.50000000E+01 |-.10000000E+01
|.11000000E+02 |-.19000000E+02
|.70000000E+01 |-.70000000E+01
|.40000000E+01 |.12000000E+02
the intermediate step values with corresponding vmax,vmin
|.12500000E+00 | 3 | 1
|.22580645E+00 | 3 | 2
|.33333333E+00 | 5 | 2
this is the optimal step taken
|.22580645E+00|
this is the next step
|.33333333E+00|
this is the real step taken
|.22688172E+00|
the new vmax,vmin
| 5 | 2 |
the new criterion value
the criterion used is vmax-vmin
|.19494624E+01|
the new areas of the triangles
|.64032258E+01 |.47731183E+01 |.66892473E+01 |.54118279E+01 |.67225807E+01|
the new coordinates of our point
|.21344086E+01 |.16806452E+01|
R 0.3303 15.824 1653.4

```

Fichier de coordonnées des noeuds

Fichier de sommets des triangles

Fonction objective
- le ϵ pour le pas
- le noeud à bouger

Les Mineurs du déterminant représentant $V_i(P_0)$ avec les superficies initiales

T_{max} initiale; T_{min} initiale

La valeur initiale de la fonction objective

La direction de descente

Les applications affines en p , $V_i(P_0 + \delta)$, ayant p donné, où la première valeur est la constante, et la seconde est

le coefficient de p
Les pas intermédiaires dans la recherche de p optimal et les T_{max} et T_{min} respectives

Le pas optimal

Le pas suivant

Le pas à prendre

Les nouveaux T_{max} et T_{min}

La valeur finale de la fonction objective

Les nouvelles superficies des triangles

La nouvelle emplacement de P_0

Annexe 4.2 :

Bouger le noeud 10 une fois par $C_1(P_0)$ entry point file is *tp*

1 0.0 0.0
 2 17.0 0.0
 3 27.0 20.0
 4 16.0 31.0
 5 0.0 24.0
 6 12.0 15.0
 7 6.0 7.5
 8 14.5 7.5
 9 24.5 17.5
 10 14.0 23.0
 11 6.0 19.5
 12 9.375 3.75
 13 20.75 11.25
 14 20.375 22.875
 15 9.0 24.375
 16 3.0 12.75
 17 13.0 26.125

entry triangle file is *tr*

1 6 7 8
 2 6 8 9
 EV(1) = 3 6 9 10
 EV(2) = 4 6 10 11
 5 6 7 11
 6 1 7 12
 7 7 8 12
 8 2 8 12
 9 1 2 12
 10 8 9 13
 11 2 8 13
 12 2 3 13
 13 3 9 13
 EV(3) = 14 9 10 14
 15 3 9 14
 16 3 4 14
 EV(4) = 17 4 10 14
 EV(5) = 18 10 11 15
 EV(6) = 19 17 4 10
 20 4 5 15
 21 5 11 15
 22 7 11 16
 23 1 7 16
 24 1 5 16
 25 5 11 16
 EV(7) = 26 10 15 17
 27 4 15 17

criterion number is 2 ($C_1(P_0)$)

coefficient number is 0,01

the point to be moved is 10

the three minors and the initial area

-157.5 -2.5 12.5 95.0
 -144.0 4.5 6.0 57.0
 203.875 -5.375 -4.125 33.75
 265.625 -8.125 -4.375 51.25
 29.25 4.875 -3.0 28.5
 15.0 4.875 -3.0 14.25
 81.75 1.75 -4.0 14.25

the initial v_{max} and v_{min} $V_{max} - V_{min}$

Bouger le noeud 10 une fois par $C_1(P_0)$ (continuation) avec la figure finale Figure 4.10

```

1 6
  the criterion used is vmax-vmin
  the initial criterion value
80.75
  the gradient vector to be used
7.375 -15.5
  the area linear equations as a function of the step
  the constant and the slope
95.0 -212.1875
57.0 -59.8125
33.75 24.296875
51.25 7.890625
28.5 82.453125
14.25 82.453125
14.25 74.90625
  the intermediate step values with corresponding vmax,vmin
0.19879304 1 6
0.27406268 4 6
0.28126701 4 1
  this is the optimal step taken
0.27406268
  this is the next step
0.28126701
  this is the real step taken
0.27413473
  the new vmax,vmin
4 1
  the new criterion value
16.581056
  the new areas of the triangles
36.832038 40.603317 40.410617 53.413095 51.103265 36.853265 34.784404
  the new coordinates of our point
16.021744 18.750912
.r 18:46 0.373 1

```

Annexe 4.3 :

Bouger le noeud 10 une fois par $C_2(P_0)$ entry point file is tpf

1 0.0 0.0
 2 17.0 0.0
 3 27.0 20.0
 4 16.0 31.0
 5 0.0 24.0
 6 12.0 15.0
 7 6.0 7.5
 8 14.5 7.5
 9 24.5 17.5
 10 14.0 23.0
 11 6.0 19.5
 12 9.375 3.75
 13 20.75 11.25
 14 20.375 22.875
 15 9.0 24.375
 16 3.0 12.75
 17 13.0 26.125

entry triangle file is tr

1 6 7 8
 2 6 8 9
 $EV(1) =$ 3 6 9 10
 $EV(2) =$ 4 6 10 11
 5 6 7 11
 6 1 7 12
 7 7 8 12
 8 2 8 12
 9 1 2 12
 10 8 9 13
 11 2 8 13
 12 2 3 13
 $EV(3) =$ 13 3 9 13
 14 9 10 14
 15 3 9 14
 16 3 4 14
 $EV(4) =$ 17 4 10 14
 $EV(5) =$ 18 10 11 15
 $EV(6) =$ 19 17 4 10
 20 4 5 15
 21 5 11 15
 22 7 11 16
 23 1 7 16
 24 1 5 16
 25 5 11 16
 $EV(7) =$ 26 10 15 17
 27 4 15 17

criterion number is 1 ($C_2(P_0)$)

coefficient number is 0.01

the point to be moved is 10

the three minors and the initial area

-157.5 -2.5 12.5 95.0
 -144.0 4.5 6.0 57.0
 203.875 -5.375 -4.125 33.75
 265.625 -8.125 -4.375 51.25
 29.25 4.875 -3.0 28.5
 15.0 4.875 -3.0 14.25
 81.75 1.75 -4.0 14.25

the initial v_{max} and v_{min}

$$1 - \frac{v_{min}}{v_{max}}$$

Bouger le noeud 10 une fois par $C_2(P_0)$ (continuation) avec la figure finale Figure 4.11

```

1 6
the initial criterion value
0.85
the gradient vector to be used
0.055263158 -0.05131579
the area linear equations as a function of the step
the constant and the slope
95.0 -0.77960526
57.0 -0.059210527
33.75 -0.085361843
51.25 -0.22450658
28.5 0.42335526
14.25 0.42335526
14.25 0.30197369
the intermediate step values with corresponding vmax,vmin
38.331717 1 6
50.343949 1 3
52.748858 1 7
59.059305 2 7
70.278038 5 7
this is the optimal step taken
59.059305
this is the next step
70.278038
this is the real step taken
59.171492
the new vmax,vmin
5 7
the new criterion value
0.40022602
the new areas of the triangles
48.869594 53.496425 28.699012 37.965611 53.550562 39.300562 32.118234
the new coordinates of our point
17.270004 19.963568

```


Annexe 4.4 :

Bouger le noeud 10 une fois par $C_3(P_0)$

```

    entry point file is tpt
    1 0.0 0.0
    2 17.0 0.0
    3 27.0 20.0
    4 16.0 31.0
    5 0.0 24.0
    6 12.0 15.0
    7 6.0 7.5
    8 14.5 7.5
    9 24.5 17.5
    10 14.0 23.0
    11 6.0 19.5
    12 9.375 3.75
    13 20.75 11.25
    14 20.375 22.875
    15 9.0 24.375
    16 3.0 12.75
    17 13.0 26.125
    entry triangle file is ttr
    1 6 7 8
    2 6 8 9
    EV(1)= 3 6 9 10
    EV(2)= 4 6 10 11
    5 6 7 11
    6 1 7 12
    7 7 8 12
    8 2 8 12
    9 1 2 12
    10 8 9 13
    11 2 8 13
    12 2 3 13
    13 3 9 13
    EV(3)= 14 9 10 14
    15 3 9 14
    16 3 4 14
    EV(4)= 17 4 10 14
    EV(5)= 18 10 11 15
    EV(6)= 19 17 4 10
    20 4 5 15
    21 5 11 15
    22 7 11 16
    23 1 7 16
    24 1 5 16
    25 5 11 16
    EV(7)= 26 10 15 17
    27 4 15 17
    criterion number is 3 (C3(P0))
    coefficient number is 0.01
    the point to be moved is 10
    the three minors and the initial area
    -157.5 -2.5 12.5 95.0
    -144.0 4.5 6.0 57.0
    203.875 -5.375 -4.125 33.75
    265.625 -8.125 -4.375 51.25
    29.25 4.875 -3.0 28.5
    15.0 4.875 -3.0 14.25
    81.75 1.75 -4.0 14.25
    the initial vmax and vmin
  
```

Bouger le noeud 10 une fois par $C_3(P_0)$ (continuation) avec la figure finale Figure 4.12

```

1 6
the initial criterion value
22.071429
the gradient vector to be used
12.25 -28.25
the area linear equations as a function of the step
the constant and the slope
95.0 -383.75
57.0 -114.375
33.75 50.6875
51.25 24.0625
28.5 144.46875
14.25 144.46875
14.25 134.4375
the intermediate step values with corresponding vmax,vmin
0.10727969 1 6
0.15287227 4 6
0.15583162 4 1
this is the optimal step taken
0.15287227
this is the next step
0.15583162
this is the real step taken
0.15290187
the new vmax,vmin
4 1
the new criterion value
6.1482121
the new areas of the triangles
36.323909 39.511849 41.500213 54.929201 50.589541 36.339541 34.805745
the new coordinates of our point
15.873048 18.680522

```

Annexe 5.1 :

Bouger le Noeud 6 en 5 ITERATIONS (cf Figure 4.13)

entry point file is
 pts
 1 0.0 0.0
 2 4.0 0.0
 3 0.0 3.0
 4 2.0 4.0
 5 5.0 2.0
 6 **1.0 1.0**
 entry triangle file is
 trs
 1 1 3 6
 2 3 4 6
 3 4 5 6
 4 5 2 6
 5 2 1 6
 criterion number is $V_{max} - V_{min}$
 2
 coefficient number is
 0.01
 the number of iterations
 5
 the point to be moved is
 6
 the criterion used is $v_{max} - v_{min}$
 the initial criterion value

Iteration I

8.0
 the gradient vector to be used
5.0 3.0
 this is the real step taken
 0.22688172
 the new v_{max}, v_{min}
 5 2
 the new criterion value
 the criterion used is $v_{max} - v_{min}$
1.9494624
 the new areas of the triangles
 6.4032258 4.7731183 6.6892473 5.4118279 6.7225807
 the new coordinates of our point

Iteration II

2.1344086 1.6806452
 the gradient vector to be used
1.0 -6.0
 this is the real step taken
 0.0011291481
 the new v_{max}, v_{min}
 3 2
 the new criterion value
 the criterion used is $v_{max} - v_{min}$
1.9195164
 the new areas of the triangles
 6.4066133 4.7877972 6.7073137 5.4027947 6.6954811
 the new coordinates of our point

Iteration III

2.1355378 1.6738703
 the gradient vector to be used
3.0 1.0
 this is the real step taken
 0.0014788287
 the new v_{max}, v_{min}
 5 2

Bouger le Noeud 6 en 5 ITERATIONS (continuation) (cf Figure 4.13)

the new criterion value
the criterion used is v_{max}-v_{min}
1.9121204
the new areas of the triangles
6.4199228 4.7892761 6.6940042 5.3954006 6.7013965
the new coordinates of our point
2.1399743 1.6753491
Iteration IV the gradient vector to be used
1.0 -6.0
this is the real step taken
0.00047158983
the new v_{max},v_{min}
3 2
the new criterion value
the criterion used is v_{max}-v_{min}
1.9061428
the new areas of the triangles
6.4213375 4.7954068 6.7015496 5.3916278 6.6900783
the new coordinates of our point
2.1404459 1.6725196
Iteration V the gradient vector to be used
3.0 1.0
this is the real step taken
0.001411063
the new v_{max},v_{min}
5 2
the new criterion value
the criterion used is v_{max}-v_{min}
1.8989047
the new areas of the triangles
6.4340371 4.7968178 6.68885 5.3845725 6.6957226
the new coordinates of our point
2.144679 1.6739306
r 15:13 0.323 1

Annexe 5.2 :

Bouger le Noeud 10 en 5 ITERATIONS (cf Figure 4.14)

entry point file is

tpt

1 0.0 0.0
 2 17.0 0.0
 3 27.0 20.0
 4 16.0 31.0
 5 0.0 24.0
 6 12.0 15.0
 7 6.0 7.5
 8 14.5 7.5
 9 24.5 17.5
 10 14.0 23.0
 11 6.0 19.5
 12 9.375 3.75
 13 20.75 11.25
 14 20.375 22.875
 15 9.0 24.375
 16 3.0 12.75
 17 13.0 26.125

entry triangle file is

ttr

1 6 7 8
 2 6 8 9
 3 6 9 10
 4 6 10 11
 5 6 7 11
 6 1 7 12
 7 7 8 12
 8 2 8 12
 9 1 2 12
 10 8 9 13
 11 2 8 13
 12 2 3 13
 13 3 9 13
 14 9 10 14
 15 3 9 14
 16 3 4 14
 17 4 10 14
 18 10 11 15
 19 17 4 10
 20 4 5 15
 21 5 11 15
 22 7 11 16
 23 1 7 16
 24 1 5 16
 25 5 11 16
 26 10 15 17
 27 4 15 17

criterion number is

1

coefficient number is

0.01

the number of iterations

5

the point to be moved is

10

I. the initial criterion value

0.85

$$1 - \frac{V_{\min}}{V_{\max}} \quad C_2(P_0)$$

Bouger le Noeud 10 en 5 ITERATIONS (continuation) (cf Figure 4.14)

the gradient vector to be used

0.055263158 -0.05131579

this is the real step taken

59.171492

the new vmax,vmin

5 7

the new criterion value

0.40022602

the new areas of the triangles

48.869594 53.496425 28.699012 37.965611 53.550562 39.300562 32.118234

the new coordinates of our point

17.270004 19.963568

II the gradient vector to be used

-0.021921303 -0.04109533

this is the real step taken

27.045943

the new vmax,vmin

5 1

the new criterion value

0.3247752

the new areas of the triangles

36.458525 44.159682 36.470535 47.645426 53.994647 39.744647 35.526537

the new coordinates of our point

16.677121 18.852106

III the gradient vector to be used

-0.10726472 0.26902064

this is the real step taken

3.5406104

the new vmax,vmin

1 3

the new criterion value

0.29872496

the new areas of the triangles

49.314197 48.165644 34.582816 46.563984 49.285715 35.035715 31.051929

the new coordinates of our point

16.297338 19.804604

IV the gradient vector to be used

-0.073443604 -0.26140419

this is the real step taken

2.0909891

the new vmax,vmin

4 6

the new criterion value

0.28436982

the new areas of the triangles

42.865705 44.19502 37.662951 50.203084 50.176842 35.926842 32.969555

the new coordinates of our point

16.143769 19.25801

V the gradient vector to be used

0.21292507 0.0026070523

this is the real step taken

0.81739283

the new vmax,vmin

5 3

the new criterion value

0.28029282

the new areas of the triangles

42.457234 44.991002 36.718677 48.779658 51.018911 36.768911 33.265607

the new coordinates of our point

16.317812 19.260141

Annexe 5.3 :

Bouger Noeud 10 en 5 Itérations par $C_3(P_0)$ avec une Itération Non-Améliorante

- I. the gradient vector to be used
 12.25 -28.25
 this is the real step taken
 0.15290187
 the new vmax,vmin
 4 1
 the new criterion value
6.1482121
 the new areas of the triangles
 36.323909 39.511849 41.500213 54.929201 50.589541 36.339541 34.805745
 the new coordinates of our point
 15.873048 18.680522
- II. the gradient vector to be used
 6.5 14.75
 this is the real step taken
 0.062182715
 the new vmax,vmin
 5 3
 the new criterion value
6.586589
 the new areas of the triangles
 46.778378 46.833864 35.544275 47.632448 49.808371 35.558371 31.844293
 the new coordinates of our point
 16.277236 19.597718
- III. the gradient vector to be used
 2.5 -22.25
 this is the real step taken
 0.00030971941
 the new vmax,vmin
 5 3
 the new criterion value
6.5644079
 the new areas of the triangles
 46.690302 46.796 35.56854 47.656306 49.832819 35.582819 31.873213
 the new coordinates of our point
 16.27801 19.590826
- IV. the gradient vector to be used
 2.5 -22.25
 this is the real step taken
 0.00030662221
 the new vmax,vmin
 5 3
 the new criterion value
6.5424489
 the new areas of the triangles
 46.603106 46.758516 35.592562 47.679926 49.857023 35.607023 31.901844
 the new coordinates of our point
 16.278776 19.584004
- V. the gradient vector to be used
 2.5 -22.25
 this is the real step taken
 0.00030355597
 the new vmax,vmin
 5 3
 the new criterion value
6.5207094
 the new areas of the triangles
 46.516782 46.721406 35.616343 47.703309 49.880985 35.630985 31.930188
 the new coordinates of our point 16.279535 19.57725

Annexe 6 :

Bouger 12 noeuds avec 3 ITERATIONS par noeud et 3 TOURS pour l'ensemble de noeuds

TOUR 1 (cf page 42)

TOUR 2

	the points,	the initial	and present	critierion values
	at each point,	and the difference	of the 2 values	
1	24.1875	14.281962	9.9055376	
2	26.25	14.634966	11.615034	
3	35.0	16.774974	18.225026	
4	42.75	18.939167	23.810833	
5	29.25	27.93115	1.31885	
6	43.0	9.8403893	33.159611	
7	40.125	15.310784	24.814216	
8	75.0	14.593881	60.406119	
9	93.75	21.003619	72.746382	
10	80.75	23.645636	57.104364	
11	43.5	27.931149	15.568851	
12	31.875	11.112908	20.762092	
13	50.0	14.634967	35.365033	
14	27.5	16.896925	10.603075	
15	42.75	24.97932	17.77068	
16	36.0	17.904305	18.095695	
17	0.0	0.54745388	-0.54745388	
	the initial and intermediate	global criterion values	and the global improvement	at this moment
	721.6875	430.72395	<u>0.59682889</u>	

TOUR 3

	the points,	the initial	and present	critierion values
	at each point,	and the difference	of the 2 values	
1	24.1875	14.399665	9.7878351	
2	26.25	15.754794	10.495206	
3	35.0	14.607202	20.392798	
4	42.75	20.455956	22.294044	
5	29.25	25.330893	3.919107	
6	43.0	10.301254	32.698746	
7	40.125	15.891786	24.233214	
8	75.0	15.589186	59.410814	
9	93.75	18.611383	75.138618	
10	80.75	21.948954	58.801046	
11	43.5	25.330892	18.169108	
12	31.875	10.848876	21.026124	
13	50.0	15.754791	34.245209	
14	27.5	17.723465	9.7765346	
15	42.75	23.062318	19.687682	
16	36.0	16.824133	19.175867	
17	0.0	0.70554352	-0.70554352	
	the initial and intermediate	global criterion values	and the global improvement	at this moment
	721.6875	438.54641	0.60766802	

Annexe 7 : Maillage à Trois Etapes

Les Données du Maillage Initial (Cf figure 4.20) et les Etapes du Maillage en Maillant
Séparément les Petits Eléments et les Grands Eléments (Cf figures 4.21; figure 4.22)

entry point file is sbpt			entry triangle file is sbtr					
1	0.0	17.0	31	7.0	5.0	1 1 9 2	37 20 30 21	72 32 31 45
2	4.0	17.0	32	11.0	5.0	2 2 9 10	38 21 30 31	73 32 45 46
3	8.0	17.0	33	14.0	5.0	3 2 10 11	39 21 31 22	74 32 46 47
4	11.0	17.0	34	17.0	6.0	4 2 11 3	40 22 31 32	75 46 64 47
5	14.0	17.0	35	18.0	7.0	5 3 11 12	41 22 32 23	76 47 64 65
6	18.0	17.0	36	21.0	8.0	6 3 12 4	42 23 32 33	77 64 63 65
7	23.0	17.0	37	23.0	9.0	7 4 12 13	43 23 33 24	78 47 65 66
8	27.0	17.0	38	24.0	10.0	8 4 13 5	44 24 33 34	79 66 65 67
9	0.0	13.0	39	0.0	3.0	9 5 13 14		80 66 67 68
10	4.0	13.0	40	2.0	2.0	10 5 14 6	45 41 40 42	81 48 47 66
11	7.0	13.0	41	3.0	3.0	11 6 14 15	46 42 40 58	82 48 66 68
12	10.0	12.0	42	4.0	2.0	12 6 15 7	47 42 58 60	83 32 47 48
13	13.0	13.0	43	6.0	2.0	13 7 15 16	48 58 59 60	84 32 48 49
14	16.0	12.0	44	7.0	3.0	14 7 16 17	49 43 42 60	85 49 48 68
15	19.0	13.0	45	8.0	4.0	15 7 17 8	50 43 60 61	86 51 68 69
16	21.0	13.0	46	9.0	3.0	16 8 17 18	51 28 39 29	87 51 49 68
17	23.0	13.0	47	10.0	2.0	17 9 19 10	52 39 40 29	88 33 49 51
18	26.0	12.0	48	11.0	3.0	18 10 19 20	53 39 55 40	89 33 50 49
19	0.0	10.0	49	12.0	3.0	19 10 20 11	54 55 56 57	90 32 49 50
20	3.0	9.0	50	12.0	4.0	20 11 20 21	55 55 57 40	91 33 32 50
21	7.0	9.0	51	14.0	4.0	21 11 21 12	56 57 58 40	92 52 33 51
22	11.0	8.0	52	15.0	5.0	22 12 21 22	57 57 59 58	93 52 51 69
23	13.0	8.0	53	17.0	5.0	23 12 22 13	58 29 40 41	94 70 52 69
24	17.0	8.0	54	19.0	6.0	24 13 22 14	59 30 29 41	95 53 52 70
25	19.0	8.0	55	0.0	2.0	25 14 22 23	60 30 41 42	96 34 52 53
26	21.0	9.0	56	0.0	0.0	26 14 23 24	61 30 42 43	97 34 33 52
27	23.0	10.0	57	1.0	0.0	27 14 24 15	62 30 43 44	98 54 34 53
28	0.0	5.0	58	3.0	1.0	28 15 24 25	63 44 43 62	99 54 35 34
29	2.0	4.0	59	3.0	0.0	29 15 25 26	64 43 61 62	100 35 24 34
30	5.0	4.0	60	5.0	0.0	30 15 26 16	65 62 61 63	101 25 24 35
			61	6.0	0.0	31 16 26 17	66 62 63 64	102 36 25 54
			62	7.0	1.0	32 17 26 27	67 46 62 64	103 37 26 36
			63	8.0	0.0	33 17 27 18	68 46 44 62	104 38 27 37
			64	9.0	2.0	34 19 28 29	69 45 44 46	105 25 35 54
			65	10.0	0.0	35 19 29 20	70 45 31 44	106 26 25 36
			66	11.0	1.0	36 20 29 30	71 31 30 44	107 27 26 37
			67	12.0	0.0			108 18 27 38
			68	13.0	2.0			
			69	14.0	3.0			
			70	16.0	4.0			

criterion number is

2

best to worst?

0

boundary point file is

the initial and intermediate global criterion values
and the global improvement at this moment

.35649392E+03 .65840820E+02 .18468988E+00

the initial and intermediate global criterion values
and the global improvement at this moment

.35649392E+03 .86482026E+02 .24259046E+00

the initial and intermediate global criterion values
and the global improvement at this moment

.35649392E+03 .90654933E+02 .25429587E+00

R 4.4076 84.111 1709.3

Les Coordonnées de Noeuds après avoir Mailler Séparément les Petits Eléments
et les Grands Eléments (cf. figures 4.21; 4.22) et les Etapes du Maillage en
Maillant le Frontier (Cf figures 4.23)

entry point file is	41 3.2291155 2.9981999
sbpt1	42 4.3165899 1.9209197
1 0.0 17.0	43 5.715973 2.0476134
2 4.0 17.0	44 6.6944651 3.1727193
3 8.0 17.0	45 8.0321015 4.1802766
4 11.0 17.0	46 8.842486 3.2979761
5 14.0 17.0	47 9.7504975 2.5985341
6 18.0 17.0	48 10.971449 2.5004078
7 23.0 17.0	49 11.929812 2.8814226
8 27.0 17.0	50 12.309928 4.2938037
9 0.0 13.0	51 13.484187 3.826897
10 3.5941024 13.141349	52 15.145567 4.6516445
11 6.7792336 13.377212	53 17.0 5.0
12 9.7130737 12.191992	54 19.0 6.0
13 12.592204 12.872693	55 0.0 2.0
14 15.543045 13.047782	56 0.0 0.0
15 18.371752 13.887089	57 1.0 0.0
16 21.015757 13.064598	58 3.0393627 1.0761773
17 23.967857 13.991859	59 3.0 0.0
18 26.0 12.0	60 5.0 0.0
19 0.0 10.0	61 6.0 0.0
20 2.8986697 8.8270448	62 7.1475797 1.6619686
21 5.9312255 9.0232998	63 8.0 0.0
22 9.6874131 8.263752	64 8.7408139 1.4830897
23 13.750476 9.2572303	65 10.0 0.0
24 17.0 8.0	66 11.019548 1.0183131
25 19.0 8.0	67 12.0 0.0
26 21.0 9.0	68 13.0 2.0
27 23.0 10.0	69 14.0 3.0
28 0.0 5.0	70 16.0 4.0
29 2.0 4.0	
30 5.0 4.0	
31 7.0 5.0	
32 11.0 5.0	
33 14.0 5.0	
34 17.0 6.0	
35 18.0 7.0	
36 21.0 8.0	
37 23.0 9.0	
38 24.0 10.0	
39 0.0 3.0	
40 1.9509481 2.0202287	

criterion number is

2

best to worst?

0

boundary point file is

the initial and intermediate global criterion values
and the global improvement at this moment

.26633293E+03 -.22894849E+02 -.85963267E-01

the initial and intermediate global criterion values
and the global improvement at this moment

.26633293E+03 -.24533412E+02 -.92115577E-01

the initial and intermediate global criterion values
and the global improvement at this moment

.26633293E+03 -.35897019E+02 -.13478251E+00

R 3.5925 88.489 1710.9

Les Coordonnées de Noeuds du Maillage Final (après avoir mailler les éléments entourant les noeuds frontaliers) (Cf. figure 4.23)

1	0.0	17.0		
2	4.0	17.0		
3	8.0	17.0		
4	11.0	17.0		
5	14.0	17.0		
6	13.0	17.0		
7	23.0	17.0		
8	27.0	17.0		
9	0.0	15.0		
10	3.5941024	13.141549		
11	6.7722336	13.377212		
12	9.7130737	12.191922		
13	12.592204	12.872593		
14	15.543045	13.047732		
15	18.371752	13.987089		
16	21.015757	13.064593		
17	23.967857	13.221852		
18	26.0	12.0		
19	0.0	10.0		
20	2.8986677	8.8270448		
21	5.9312255	9.0232928		
22	9.6874131	8.263752		
23	13.750476	9.2572303		
24	16.009304	9.6024004		
25	17.958756	3.8225107		
26	19.456142	10.637001		
27	22.451177	11.642315		
28	0.0	5.0		
29	1.93317	5.7215231		
30	4.5874405	6.8147208		
31	6.8302677	6.9141577		
32	9.9664414	6.973465		
33	13.178504	6.6384423		
34	15.590594	6.7764057		
35	13.0	7.0		
36	21.0	8.0		
37	23.0	9.0		
38	24.0	10.0		
39	0.0	3.0		
40	1.2509431	2.0202237		
41	3.2291155	2.2281222		
42	4.3165399	1.9209197		
43	5.715973	2.0476134		
44	6.6944651	3.1727193		
45	8.0321015	4.1802766		
46	8.842486	3.2972761		
47	9.7504975	2.5735341		
48	10.971449	2.5004073		
49	11.922812	2.8814226		
50	12.309228	4.2939037		
51	13.484137	3.326827		
52	15.145567	4.6516445		
53	17.0	5.0		
54	19.0	6.0		
55	0.0	2.0		
56	0.0	0.0		
57	1.0	0.0		
58	3.0393627	1.0761773		
59	3.0	0.0		
60	5.0	0.0		
61	6.0	0.0		
62	7.1475727	1.6618686		
63	3.0	0.0		
64	3.7408139	1.4330827		
65	10.0	0.0		
66	11.019548	1.0183131		
67	12.0	0.0		
68	13.0	2.0		
69	14.0	3.0		
70	16.0	4.0		

BIBLIOGRAPHIE

- [1] AFZALI, M., Plane Intersections of a Three-Dimensional Mesh;
Short Communications, January 1982, pages 1253-1259.
- [2] BYKAT, A., Automatic Generation of Triangular Grid :
I. Subdivision of a General Polygon into Convex Subregions
II. Triangulation of Convex Polygons;
International Journal for Numerical Methods in Engineering, 1976,
Volume 10, pages 1329-1342.
- [3] CIARLET, P.G., Introduction à l'Analyse Numérique Matricielle
et à l'Optimisation;
Masson, Paris, 1982.
- [4] EDDY, W. F., A New Convex Hull Algorithm for Planar Sets;
ACM Transactions on Mathematical Software, Volume 3, Number 4,
December 1977, pages 398-403.
- [5] FREY, A.E., C.A. Hall, and T.A. Porsching, An Application
of Computer Graphics to Three Dimensional Finite Element Analysis;
Computers & Structures, Volume 10, May 1978, pages 149-154.
- [6] GREEN, P.J., and R. Sibson, Computing Dirichlet Tessellations in the Plane;
The Computer Journal, Volume 21, Number 2, March 1977, pages 168-173.
- [7] HERMELINE, F., Triangulation Automatique d'un Polyèdre en Dimension N;
R.A.I.R.O. Analyse Numérique / Numerical Analysis, Volume 16, Numero 3,
1982, pages 211-242.
- [8] IMAFUKU, I., Y. Kodera, M. Sayawaki and M. Kono, A Generalized
Automatic Mesh Generation Scheme for Finite Element Method;
International Journal for Numerical Methods in Engineering, Volume 15, 1980,
pages 713-731.
- [9] KAMEL, H.A., and H.K. Eisenstein, Automatic Mesh Generation in Two
and Three Dimensional Inter-Connected Domains;
High Speed Computing of Elastic Structures, Proc. Symp. Int. Union
of Theoretical and Applied Mechanics, Liège, August 23-28, 1970
(Ed. B. Fraeijis de Veubeke), Les Congrès et Colloques de l'Université de Liège,
1971, Volume 61, pages 455-475.
- [10] KLEINSTREUER C. and J.T. Holdeman, A Triangular Finite Element Mesh
Generator for Fluid Dynamic Systems of Arbitrary Geometry;
International Journal for Numerical Methods in Engineering, Volume 15, 1980,
pages 1325-1334.
- [11] LEWIS B.A. and J.S. Robinson,
Triangulation of Planar Regions with Applications;
The Computer Journal, Volume 21, Number 4, November 1976,
pages 324-332.

- [12] MCLAIN, D.H., Two Dimensional Interpolation from Random Data; The Computer Journal, Volume 19, Number 2, March 1974, pages 178-181.
- [13] NGUYEN-Van-Phai, Automatic Mesh Generation with Tetrahedron Elements; International Journal for Numerical Methods in Engineering, Volume 18, 1982, pages 273-289.
- [14] PISSANETZKY, S., KUBIK: An Automatic Three-Dimensional Finite Element Mesh Generator; International Journal for Numerical Methods in Engineering, Volume 17, 1981, pages 255-269.
- [15] PREPARATA, F.P. and S.J. Hong, Convex Hulls of Finite Sets of Points in Two and Three Dimensions; Communications of the ACM, February 1977, Volume 20, Number 2, pages 87-93.
- [16] SADEK, E.A., A Scheme for the Automatic Generation of Triangular Finite Elements; International Journal for Numerical Methods in Engineering, Volume 15, 1980, pages 1813-1822.
- [17] SIBSON, R., Locally Equiangular Triangulations; The Computer Journal, Volume 21, Number 3, March 1977, pages 243-245.
- [18] THACKER, W.C., A Brief Review of Techniques for Generating Irregular Computational Grids; International Journal for Numerical Methods in Engineering, Volume 15, 1980, pages 1335-1341.
- [19] WEGSTEIN, H.J., Algorithm 113 Treesort, Algorithm 271 Quickersort; Communications of the ACM, Volume 8, Number 11, October 1965, pages 434, 669-670.
- [20] YERRY, M.A. and M.S. Shephard, A Modified Quadratic Approach to Finite Element Mesh Generation; IEEE CG & A, January-February 1983, pages 39-46.
- [21] ZIENKIEWICZ, O.C. and D.V. Phillips, An Automatic Mesh Generation Scheme for Plane and Curved Surfaces by "Isoparametric" Coordinates; International Journal for Numerical Methods in Engineering, Volume 3, 1971, pages 519-528.
- [22] HELD, M. and H.P. Crowder, Validation of Subgradient Optimization; Mathematical Programming, Volume 6, 1974, pages 62-88.
- [23] POULETTY, C., Génération et Optimisation de Maillages en Eléments Finis - Application à la Résolution de quelques Equations en Mécanique des Fluides; thèse de 3ème cycle (à paraître).
- [24] PERONNET, A., A Library of Subroutines for Finite Element Analysis (MODULEF); Proceedings of the third INRIA Symposium on Computing.

AUTORISATION DE SOUTENANCE

DOCTORAT 3^{ème} CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USMG

Vu les dispositions de l'arrêté du 16 avril 1974,

Vu les dispositions de l'arrêté du 5 juillet 1984,

Vu les rapports de M. F.O.N.C.U.P.T.

M.

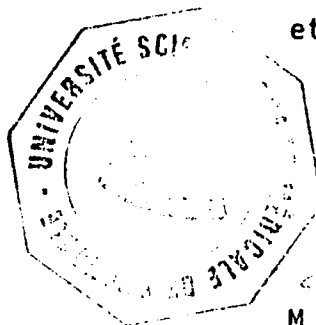
M. CHARLES CoA est autorisé
à présenter une thèse en vue de l'obtention du .. doctorat de 3^{ème} cycle

08 OCT. 1984

Grenoble, le

Le Président de l'Université Scientifique

et Médicale



M. TANCHE

RESUME

Dans cette thèse, on étudie un aspect du problème général d'obtention des maillages de type éléments finis. C'est celui d'amélioration d'un maillage existant sans changer la topologie actuelle de tétraèdres. On propose des méthodes générales et algorithmiques d'optimisation classique en recherche opérationnelle connues sous le nom de méthodes de sous-graadiant. On applique ces méthodes sur des cas d'école et sur des cas réels proposés par la DGT/DEA, AVIONS MARCEL DASSAULT/ BREGUET AVIATION.

MOTS CLES

MAILLAGES DE TYPE ELEMENTS FINIS, TETRAEDRISATION, TRIANGULATION, METHODES DE SOUS-GRADIENT.