

UNIVERSITÉ DU QUÉBEC

MÉMOIRE
PRÉSENTÉ A
L'UNIVERSITÉ DU QUÉBEC A CHICOUTIMI
COMME EXIGENCE PARTIELLE
DE LA MAITRISE EN RESSOURCES ET SYSTÈMES

PAR
RÉJEAN FOURNIER
ING.

UTILISATION D'UN MODÈLE VARIATIONNEL POUR
L'INSPECTION DES DIMENSIONS A L'AIDE DE LA VISION PAR
ORDINATEUR

AVRIL 1992



Mise en garde/Advice

Afin de rendre accessible au plus grand nombre le résultat des travaux de recherche menés par ses étudiants gradués et dans l'esprit des règles qui régissent le dépôt et la diffusion des mémoires et thèses produits dans cette Institution, **l'Université du Québec à Chicoutimi (UQAC)** est fière de rendre accessible une version complète et gratuite de cette œuvre.

Motivated by a desire to make the results of its graduate students' research accessible to all, and in accordance with the rules governing the acceptance and diffusion of dissertations and theses in this Institution, the **Université du Québec à Chicoutimi (UQAC)** is proud to make a complete version of this work available at no cost to the reader.

L'auteur conserve néanmoins la propriété du droit d'auteur qui protège ce mémoire ou cette thèse. Ni le mémoire ou la thèse ni des extraits substantiels de ceux-ci ne peuvent être imprimés ou autrement reproduits sans son autorisation.

The author retains ownership of the copyright of this dissertation or thesis. Neither the dissertation or thesis, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

UTILISATION D'UN MODÈLE VARIATIONNEL POUR L'INSPECTION DES DIMENSIONS A L'AIDE DE LA VISION PAR ORDINATEUR

PAR

RÉJEAN FOURNIER

RÉSUMÉ

Le développement de la vision par ordinateur a amené son utilisation pour l'inspection dimensionnelle. Dans le domaine de la conception assistée par ordinateur (CAO), une nouvelle méthode de modélisation a été développée: la géométrie variationnelle.

Le présent mémoire fait le lien entre un modèle variationnel et l'inspection des dimensions à l'aide de la vision par ordinateur. Le modèle est enregistré dans une base de données paramétriques équivalente à celle que pourrait produire un logiciel de CAO.

La première partie porte sur l'élaboration d'un modèle variationnel simple en deux dimensions. La deuxième partie porte sur l'utilisation du processus de vision par ordinateur permettant d'extraire l'information voulue d'une image. L'image utilisée est celle produite par une caméra CCD. Un progiciel a été développé qui fait le lien entre le modèle et les données obtenues avec la vision par ordinateur. Le système développé est utilisable pour des objets de type polygone à n côtés.

Des résultats expérimentaux ont été obtenus avec différents exemples. Ils ont permis de démontrer que le système est très bien adapté pour la vérification des paramètres du modèle. L'utilisation du modèle lors du processus de vision permet le calcul des paramètres réels et de leur précision. Si le modèle contient les paramètres et leurs tolérances, la comparaison peut être faite avec les données obtenues par la vision.

Le modèle variationnel est utilisable pour la vérification dimensionnelle à l'aide de la vision par ordinateur. Un des avantages est d'utiliser le même modèle pour la vision que celui utilisé pour la conception. La vérification des dimensions se fait alors sur les paramètres du modèle.

Directeur de recherche: Pierre McComber, Ing. Ph.D.
Professeur, département de mécanique, École de
Technologie Supérieure, Montréal, Québec

REMERCIEMENTS

Je tiens à remercier toutes les personnes qui m'ont appuyé et aidé à la réalisation de ce mémoire à l'Université du Québec à Chicoutimi et plus particulièrement, Pierre McComber, ing. Ph.D., mon directeur de recherche et professeur à l'École de technologie Supérieure. Merci aussi à Magdalie Nadeau qui a su m'encourager et me soutenir tout au long de mon labeur.

TABLE DES MATIÈRES

	page
RÉSUMÉ	i
REMERCIEMENTS	ii
TABLE DES MATIÈRES	iii
LISTE DES FIGURES	vii
LISTE DES TABLEAUX	ix
LISTE DES SYMBOLES ET DES ABRÉVIATIONS	x
GLOSSAIRE	xi
1. Introduction	1
1.1 Objectif général	1
1.2 Hypothèses de travail	4
1.3 Travaux pertinents	5
1.3.1 Vision par ordinateur	6
1.3.2 Inspection dimensionnelle	7
1.3.3 Modélisation appliquée à la Conception Assistée par Ordinateur	8
1.3.3.1 Conception paramétrique	9
1.3.3.2 Géométrie variationnelle	9
2. Modélisation	10
2.1 Introduction	10
2.2 Approche paramétrique	14

2.3	Approche variationnelle	15
2.3.1	Topologie	16
2.3.2	Schème	16
2.4	Définition du modèle	19
2.4.1	Topologie	20
2.4.1.1	Tables de définition des noeuds et éléments	23
2.4.1.2	Sens de la numérotation	24
2.4.2	Contraintes	25
2.4.2.1	Ordre des contraintes	26
2.4.2.2	Contraintes qui définissent la référence	26
2.4.2.3	Autres contraintes permettant de fixer complètement la géométrie	28
2.4.3	Tolérances	31
2.4.3.1	Théorie des tolérances	33
2.4.3.2	Tolérances sur les contraintes qui définissent la référence	35
2.4.3.3	Tolérances sur les autres contraintes	37
2.5	Résumé du modèle variationnel	38
3.	Vision par ordinateur	40
3.1	Introduction	40
3.2	Acquisition d'images	43
3.2.1	Introduction	43
3.2.2	Digitalisation de l'image	43

3.2.3	Problèmes d'intensité et leur correction	47
3.2.4	Distorsion et correction de la distorsion	50
3.3	Analyse primaire	52
3.3.1	Introduction	52
3.3.2	Niveau de séparation	53
3.3.3	Suivi de contour et représentation de la forme	54
3.4	Analyse secondaire	57
3.4.1	Présentation des deux méthodes	57
3.4.2	Première méthode: Conditions de matériel	58
3.4.2.1	Description des étapes	58
3.4.2.2	Élaboration des algorithmes	60
3.4.2.3	Avantages et désavantages	61
3.4.3	Deuxième méthode: Modélisation des données réelles	62
3.4.3.1	Description des étapes	62
3.4.3.2	Élaboration des algorithmes	65
3.4.3.3	Avantages et désavantages	70
3.4.3.4	Précision des mesures pour la deuxième méthode	71
3.4.4	Discussion sur les méthodes	76
4.	Réalisation (expérimentation)	77
4.1	Introduction	77
4.2	Réalisation matérielle	77
4.3	Réalisation logicielle	83

4.3.1	Introduction	83
4.3.2	Structure de la base de données du modèle	84
4.3.3	Analyse des programmes	88
4.3.3.1	HISTO	88
4.3.3.2	POLY	91
4.3.3.3	GEO	94
4.4	Exemples	99
4.5	Résultats obtenus avec les exemples	104
4.6	Discussion des résultats obtenus avec les exemples	108
5.	Conclusions et recommandations	110
5.1	Conclusions	110
5.2	Recommandations	112

BIBLIOGRAPHIE

- ANNEXE A - Fonctions de transfert de quelques caméras à technologie CCD disponibles commercialement.**
- ANNEXE B - Description des programmes du Micro-PDP 11/23 utilisés pour l'acquisition d'images.**
- ANNEXE C - Objets utilisés comme exemples.**
- ANNEXE D - Fichiers de données des modèles des exemples.**
- ANNEXE E - Liste des programmes.**
- ANNEXE F - Résultats obtenus pour les exemples.**

LISTE DES FIGURES

	page
1.1 Schéma sommaire du système d'inspection	2
2.1 Processus de modélisation en vue de la fabrication	10
2.2 Système de coordonnées cartésien	21
2.3 Trois types d'éléments à une dimension	22
3.1 Processus de vision par ordinateur	41
3.2 Processus de digitalisation	44
3.3 Fonction de transfert d'une caméra idéale	48
3.4 Plage utilisable de la fonction de transfert réelle	49
3.5 Détermination du niveau de séparation T à partir de l'histogramme	54
3.6 Suivi de contour d'une image binaire	55
3.7 Diagramme de la méthode "conditions de matériel"	59
3.8 Diagramme de la méthode "modélisation des données réelles"	64
3.9 Problème du point de séparation entre deux éléments	66
4.1 Montage pour l'acquisition des images	81
4.2 Organigramme du programme HISTO	89
4.3 Organigramme du programme POLY	92
4.4 Organigramme du programme GEO	94
4.5 Arbre des préalables du programme GEO	96
4.6 Modèle de l'exemple 1: Élément de type 1	100

4.7	Données de la vision simulées pour l'exemple 1	100
4.8	Modèle de l'exemple 2: Rectangle	101
4.9	Données de la vision simulées pour l'exemple 2	101
4.10	Modèle de l'exemple 3: Rectangle	102
4.11	Données obtenues de la vision pour l'exemple 3	102
4.12	Modèle de l'exemple 4: Polygone	103
4.13	Données obtenues de la vision pour l'exemple 4	103

LISTE DES TABLEAUX

	page
3.1 Utilisation de la topologie du modèle	67
3.2 Utilisation d'un modèle topologique général	68
3.3 Exemple de mesures de niveau de gris en fonction de x	72
4.1 Types de contraintes	87
4.2 Résumé des résultats pour les exemples	105

LISTE DES SYMBOLES ET DES ABRÉVIATIONS

ANSI American National Standards Institute

CAO Conception Assistée par Ordinateur

CCD Charged Coupled Device

GKS Graphic Kernel System

ISO International Standards Association

LMC Least Material Condition

MEF Méthode des Éléments Finis

MIT Massachusset Institute of Technology

MMC Maximum Material Condition

MOS Metal-Oxyde-Semiconductor

RAM Random Access Memory

ReGIS Digital graphics protocol

GLOSSAIRE

Comme la bibliographie est constituée en majorité de documents rédigés en langue anglaise, la liste suivante permet au lecteur de faire la correspondance entre les termes français utilisés et les termes anglais originaux.

<u>Français</u>	<u>Anglais</u>
Entité	Feature
Modèle topologique	Shape model
Schème	Scheme
	Dimension tree
Référence	Datum
Objet	Object
Face	Face
Arrête	Edge
Point (de rencontre)	Vertex
Représentation de frontière	Boundary representation
	Brep

CHAPITRE I

INTRODUCTION

1.1 OBJECTIF GÉNÉRAL

Avec l'avancement de l'informatique, l'industrie change de plus en plus ses méthodes de production. Ainsi, dans le domaine de la conception, on utilise maintenant couramment des logiciels de conception assistée par ordinateur. Au niveau de la fabrication, le développement des nouvelles technologies amène une automatisation des méthodes utilisées. Au cours des dernières années, on a donc assisté à un développement important au niveau des logiciels de CAO et en particulier au rapprochement de ceux-ci vers les méthodes de fabrication.

Un autre domaine qui a connu un intense développement est celui de la vision artificielle aussi appelée vision par ordinateur. De domaine de recherche, la vision par ordinateur est devenue un outil applicable à l'industrie.

Le présent mémoire fait le lien entre l'utilisation d'une base de données paramétriques, telle qu'elle pourrait être produite par un logiciel de CAO pour la modélisation d'un objet, et l'inspection des dimensions à l'aide de la vision

par ordinateur. La figure 1.1 présente le schéma sommaire du système. On désire vérifier les avantages d'utiliser une base de données paramétriques pour l'inspection dimensionnelle à l'aide de la vision par ordinateur.

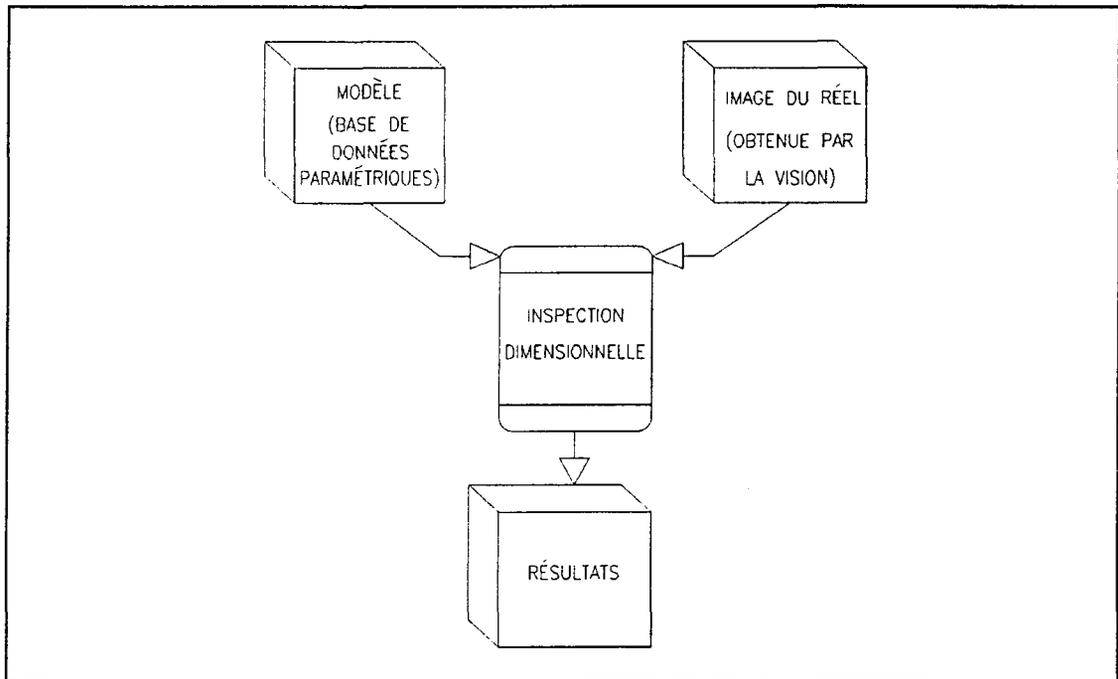


Figure 1.1: Schéma sommaire du système d'inspection

Au niveau de la théorie, la présente thèse se divise en deux parties. La première partie concerne la modélisation des données pour arriver à la définition et l'élaboration d'une base de données paramétriques. La deuxième partie porte sur la vision par ordinateur. La théorie nécessaire aux méthodes utilisées lors de l'expérimentation est présentée. La présentation permettra de connaître l'état actuel des possibilités et des limites du domaine de la vision appliquée à l'inspection dimensionnelle en deux dimensions. Le lien entre le

modèle (contenu dans la base de données) et le monde réel (données obtenues par la vision) est élaboré; une méthode est développée pour l'utilisation de la base de données à des fins de vérification des dimensions de pièces mécaniques.

Les concepts présentés ont été utilisés pour la réalisation d'un système de base afin d'en vérifier la faisabilité et d'obtenir des résultats expérimentaux.

1.2 HYPOTHÈSES DE TRAVAIL

Certaines hypothèses de travail ont dû être posées afin de bien cerner l'étendue de la présente thèse.

En ce qui concerne la vision par ordinateur, celle-ci est limitée à l'acquisition et au traitement d'images en deux dimensions de pièces mécaniques simples. Les conditions d'obtention des images sont contrôlées afin de simplifier le traitement subséquent. L'acquisition est faite en environnement de laboratoire.

En ce qui concerne la base de données, celle-ci utilise l'ensemble complet et suffisant des contraintes géométriques. Les contraintes géométriques utilisées ont dû être limitées en nombre et en complexité afin de ne pas alourdir inutilement les expérimentations.

Finalement, la réalisation a dû se limiter au matériel et aux outils logiciels disponibles pour l'expérimentation.

1.3 TRAVAUX PERTINENTS

Les concepts nécessaires à l'élaboration de la présente thèse couvrent plusieurs domaines. Le premier domaine est celui de la vision par ordinateur. Le deuxième est celui de l'inspection dimensionnelle. De nombreux travaux en ce domaine sont rattachés directement à la vision par ordinateur ou plus généralement aux moyens non-tactiles de mesure. Le troisième domaine est celui de la modélisation particulièrement appliquée lors de la conception par ordinateur. Un secteur de la recherche en conception est directement orienté en vue de la fabrication. Les liens entre les modèles utilisés et l'inspection se retrouvent surtout au niveau des travaux portant spécifiquement sur l'inspection dimensionnelle. La seule référence traitant directement des modèles utilisés pour l'inspection est Woods et al. [5.33].

Les points suivants présentent les travaux pertinents à chacun des domaines. Un bref historique y est présenté, suivi des développements actuels et des différents domaines de recherche reliés à la présente thèse.

1.3.1 VISION PAR ORDINATEUR

Relativement récente, la vision par ordinateur est un domaine de recherche en forte croissance. Les premières expériences furent faites vers la fin des années cinquante et plusieurs des concepts essentiels ont été développés durant les dix dernières années.

Aujourd'hui, la recherche en vision par ordinateur peut être divisée selon les technologies utilisées (composantes et architecture des systèmes), les types de logiciels (méthodes et algorithmes) et les domaines d'application.

La bibliographie présentée couvre tous les domaines. Le livre *Computer Vision* de Ballard et Brown [2.1] est utilisé comme référence générale. Divers articles plus spécialisés couvrent l'acquisition et les divers algorithmes utilisés pour l'analyse d'image. Les références les plus pertinentes sont Rosenfeld [1.1], Gonzalez et Wintz [2.3], Pratt [2.4], Brady [2.5], Wagner [5.1] à Davis [5.13], White [5.16] à Roning et Hall [5.31] et Woods et al. [5.33].

1.3.2 INSPECTION DIMENSIONNELLE

Le domaine d'inspection dimensionnelle qui est pertinent est celui rattaché à la vision par ordinateur. Les développements en ce domaine suivent généralement les développements technologiques de la vision.

Les références utilisées permettent de préciser les limites de l'inspection dimensionnelle faisant usage de la vision. Plusieurs méthodes d'inspection ont été développées selon les types d'équipement utilisé, les méthodes et algorithmes de traitement et selon les besoins d'application spécifique. Les références pertinentes sont [1.2], [1.3], [2.2], [5.1], [5.9], [5.12], [5.13], [5.16], [5.19], [5.20], [5.23], [5.26], [5.28], [5.29], [5.31] et [5.33] à [5.35].

1.3.3 MODÉLISATION APPLIQUÉE A LA CONCEPTION ASSISTÉE PAR ORDINATEUR

La conception assistée par ordinateur ou CAO a connu ses débuts dans les années soixante. Elle a d'abord évolué à partir de la production de dessins réalisés manuellement. Ce domaine est aujourd'hui en pleine expansion et suit l'implantation de plus en plus grande des stations de travail et des ordinateurs personnels dans l'industrie. Plusieurs méthodes de conception développées depuis moins de dix ans sont maintenant disponibles dans des logiciels commerciaux.

Plusieurs bases mathématiques sont utilisées pour la modélisation. Les théories pour la représentation formelle d'objets du monde réel sont encore en développement. Ceci est particulièrement vrai en ce qui concerne les tolérances nécessaires à la fabrication et à l'inspection des dimensions. Les standards existants sont définis de façon informelle. Les références pertinentes sont Baer et al. [5.14], CIME staff report [5.15], Dori et Pnueli [5.32], Woods et al. [5.33], Etesami et Uicker [5.34] et Requicha [5.38].

Deux méthodes particulières de conception sont intéressantes ici car elles utilisent une approche formelle. Les travaux pertinents à ces deux méthodes sont récents et peu nombreux.

1.3.2.1 CONCEPTION PARAMÉTRIQUE

La conception paramétrique consiste à définir les dimensions d'un modèle par des paramètres ou variables qui peuvent prendre par la suite des valeurs définies. Cette approche avait initialement été développée comme un outil de programmation des logiciels de CAO disponibles commercialement. La seule référence abordant directement la conception paramétrique est CIME staff report [5.15].

1.3.2.2 GÉOMÉTRIE VARIATIONNELLE

Cette technologie fut développée au début des années 80 au laboratoire de CAO du MIT. La géométrie variationnelle est une méthode permettant de concevoir et de modifier une géométrie à partir des caractéristiques géométriques, appelées contraintes, tels les distances, angles, etc. La référence la plus élaborée est la thèse de V.C. Lin [3.1]. Les autres travaux se rapportant à cette technologie sont McComber [4.1], Fournier [4.2] et CIME staff report [5.15].

CHAPITRE II

MODÉLISATION

2.1 INTRODUCTION

La modélisation est vue ici comme l'étape au cours de laquelle le concepteur définit le modèle d'un objet. La figure 2.1 présente les étapes du processus de modélisation en vue de la fabrication.

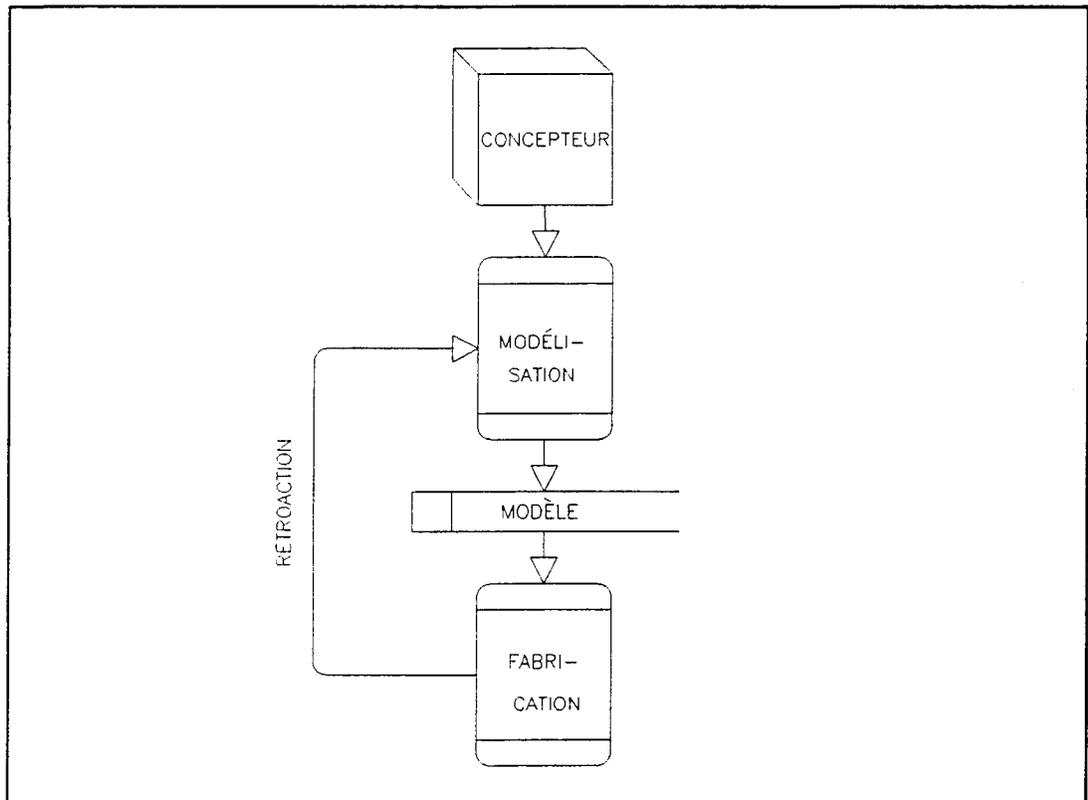


Figure 2.1: Processus de modélisation en vue de la fabrication

L'importance d'avoir un modèle qui répond au besoin voulu n'est plus à démontrer car, les limites du modèle sont les limites qui définissent notre représentation du réel.

La modélisation en ingénierie mécanique s'effectuait auparavant principalement à l'aide de dessins. Il est maintenant reconnu que l'ordinateur est un médium de représentation de la même façon que le sont les dessins. Plutôt que de représenter les dessins dans un ordinateur et d'incorporer leurs limites, l'ordinateur peut représenter un modèle plus réaliste.

Les pratiques courantes en dessin industriel, décrits dans les normes (ANSI, ISO, ...), sont malheureusement encore définies d'une façon informelle, principalement pour les situations spéciales. Les développements au niveau de la conception assistée par ordinateur (CAO) ont fait naître le besoin d'une plus grande formalisation afin que l'ordinateur puisse libérer le plus possible le concepteur dans sa tâche. Cette formalisation permet aussi d'atteindre une plus grande qualité demandée à l'industrie aujourd'hui, ainsi que la représentation complète de modèles de plus en plus complexes.

Le présent travail se limite à définir un modèle tel qu'il pourrait être généré par un logiciel de CAO. L'étape de la génération du modèle par le concepteur ne sera pas examinée. La présentation est limitée au modèle

proprement dit.

Le but recherché est d'avoir un modèle utilisable dans le contexte de la fabrication industrielle et de l'appliquer à la vérification dimensionnelle au moyen de la vision par ordinateur. Les processus de modélisation et de fabrication, tels qu'ils apparaissent à la figure 2.1, ne seront pas abordés.

Le modèle doit servir à vérifier les dimensions de pièces mécaniques en deux dimensions. Ce modèle doit donc couvrir la représentation de structures géométriques bi-dimensionnelles. Ballard et Brown [2.1] consacrent un chapitre complet à ce sujet. Ils classent généralement les représentations en deux types: les représentations de frontière et les représentations de région. Différentes bases mathématiques sont utilisées.

Suite à l'examen de la bibliographie, il faut conclure que les théories pour la représentation formelle d'objets du monde réel sont encore au début de leurs développements. Ceci est particulièrement vrai en ce qui concerne les tolérances nécessaires à la fabrication et, dans notre cas, à l'inspection des dimensions.

Le modèle choisi est le modèle variationnel. Au cours de ce chapitre, ce modèle est élaboré en détail. Auparavant, deux approches sont examinées:

l'approche paramétrique et l'approche variationnelle. Celles-ci servent de base à la définition du modèle choisi. Le modèle servira à la définition d'une base de données contenant l'information complète et suffisante à la représentation de structures géométriques bi-dimensionnelles.

2.2 APPROCHE PARAMÉTRIQUE

Avec le développement des logiciels de CAO, le concepteur a eu à indiquer à l'ordinateur les différentes commandes menant à l'obtention du modèle final. Ce modèle peut aller d'une simple représentation d'une vue d'un objet à une représentation complète d'un objet extrêmement complexe en trois dimensions. Très vite, le besoin est apparu d'automatiser les séquences de commandes. L'utilisateur dans ce cas n'a plus qu'à spécifier certains paramètres, l'ordinateur prenant en charge les opérations qui ne changent pas. Dans cette méthode, on appelle paramètres les valeurs ou dimensions spécifiées par l'utilisateur. Cette méthode est très utile pour la création de familles de pièces semblables par exemple.

A partir de ses débuts, cette approche s'est complexifiée jusqu'à aujourd'hui pour obtenir ce qu'on appelle "l'approche paramétrique". L'approche paramétrique demande de savoir dans quel ordre les séquences de modélisation sont faites. L'approche paramétrique est en fait une technique de conception pour la création du modèle d'un objet. Cette technique est apparue et s'est développée en lien avec la CAO.

2.3 APPROCHE VARIATIONNELLE

L'approche variationnelle de la conception, appelée géométrie variationnelle, est une méthode mathématique de modélisation. Elle permet une conception et une modification aisée de la géométrie car elle se rapproche de la conception traditionnelle utilisée en dessin technique.

La méthode de la géométrie variationnelle a été développée au début des années 1980 au laboratoire CAO du MIT par Lin [3.1]. Depuis lors, seulement quelques articles touchant cette méthode ont été publiés. Des systèmes ont été développés pour des applications commerciales et, c'est pourquoi, l'information détaillée sur la mise en oeuvre de la méthode n'est pas disponible. Les systèmes développés à ce jour sont orientés vers les applications en ingénierie mécanique pour aider au design, à l'évaluation et à la fabrication de pièces mécaniques.

Le modèle utilisé en géométrie variationnelle peut être divisé en deux parties: la topologie de l'objet et les contraintes appliquées fixant la topologie pour définir une géométrie particulière.

2.3.1 TOPOLOGIE

La topologie est l'expression mathématique de la forme de l'objet. La forme ou contour de l'objet est défini par l'assemblage des entités géométriques qui le composent. Chaque entité géométrique peut s'exprimer par un certain nombre de variables ou coordonnées qui sont appelées degrés de liberté. Le modèle topologique est donc l'ensemble des correspondants aux coordonnées de points et les interconnexions entre ces degrés de liberté.

Le présent travail se limite à la topologie d'objets "bien définis" en deux dimensions. Nous ne donnerons pas de définition formelle de ce qu'est un objet "bien défini". Nous posons l'hypothèse que le modèle topologique généré par un logiciel CAO paramétrique représente un objet "bien défini".

2.3.2 SCÈME

Le schème est l'ensemble des contraintes appliquées sur les degrés de liberté. Chaque contrainte est appliquée sur les degrés de liberté au moyen d'une équation algébrique. Les équations algébriques peuvent être linéaires ou non linéaires. Un système d'équation particulier appliqué à une topologie constitue un schème.

Pour que le système d'équation se solutionne, il faut autant de contraintes que la topologie a de degrés de liberté. Dans ce cas, on dit que le schème est admissible. Dans le cas où on a moins de contraintes que de degrés de liberté, le système d'équation n'a pas de solution. On dit que le schème est sous-contraint. En terme de dessin technique, un schème sous-contraint correspond à un dessin sous-dimensionné. Dans le cas où on a plus de contraintes que de degrés de liberté, le système d'équation n'a pas non plus de solution. On dit que le schème est sur-contraint. En terme de dessin technique, un schème sur-contraint correspond à un dessin sur-dimensionné. Pour le présent travail, nous considérons que nous avons toujours affaire à des schèmes admissibles.

Il est important de noter qu'il n'y a pratiquement jamais un seul schème admissible pour une même topologie; il y a plutôt un ensemble de schèmes admissibles. Le choix d'un schème admissible est déterminé lors de l'étape de conception. Celui-ci provient de l'ordre et du choix des contraintes qui seront appliquées à la topologie. Ces contraintes sont déterminées par l'utilisateur d'une façon explicite ou sont prises en charge d'une façon implicite par le système CAO. Comme nous n'aborderons pas l'étape de la conception, nous prenons comme point de départ qu'un schème admissible est déjà déterminé.

La géométrie variationnelle exprime le schème sous forme matricielle.

Elle fait donc appel de façon intensive aux méthodes matricielles de solution d'équations non-linéaires pour solutionner le système d'équation. La solution du système d'équation satisfait les contraintes posées et donne la géométrie de l'objet par la valeur de chacun des degrés de liberté. La modification de la géométrie se fait en changeant les paramètres à l'intérieur des contraintes et en solutionnant à nouveau. Mentionnons qu'en général, le temps de calcul croît au carré du nombre de contraintes. Les méthodes de solution matricielles des équations ne seront pas abordées au cours de ce travail.

2.4 DÉFINITION DU MODÈLE

Un modèle variationnel comprend différentes informations qui, ensembles, définissent le modèle d'un objet quelconque.

Les informations générées à l'aide d'un logiciel de CAO approprié sont enregistrées sous un certain format à l'intérieur d'un ordinateur. Nous devons maintenant définir une telle base de données de façon concrète afin de pouvoir l'utiliser par la suite pour la vision.

La base de données est divisée en trois parties distinctes: la topologie, les contraintes et les tolérances. Chaque partie contient des données reliées entre elles et exprimées sous un format facilement programmable par ordinateur.

Plusieurs rapprochements peuvent être faits entre le modèle géométrique variationnel utilisé ici et la méthode des éléments finis (MEF).

La MEF est une méthode de modélisation permettant de simuler le comportement de systèmes physiques complexes. C'est une méthode très générale qui s'applique à la majorité des problèmes rencontrés dans la pratique: problèmes stationnaires ou non stationnaires, linéaires ou non linéaires, définis

dans un domaine géométrique quelconque à une, deux ou trois dimensions.

En géométrie variationnelle, on a un problème non-linéaire défini dans un domaine géométrique à une ou plusieurs dimensions. La méthode fait aussi appel aux méthodes numériques pour résoudre les équations algébriques ainsi qu'à l'informatique pour exécuter efficacement les calculs sur ordinateur.

La représentation de frontière pour définir la forme d'une pièce peut se faire à l'aide d'éléments finis à une dimension. Ce point est très important et sera utilisé pour l'expression de la topologie des objets.

Mentionnons finalement que plusieurs logiciels de CAO font maintenant le lien entre un modèle "solide" tridimensionnel et un modèle "éléments finis" tridimensionnel.

2.4.1 TOPOLOGIE

Le modèle topologique doit définir les entités géométriques ou composantes de l'objet et la façon dont celles-ci sont interconnectées.

Le système de coordonnées qui est utilisé est le système familier de coordonnées bi-dimensionnel rectangulaire (cartésien). Les coordonnées d'un

point sont les projections perpendiculaires de sa localisation sur les axes de coordonnées. La figure 2.2 illustre ce système. Le type de représentation choisi

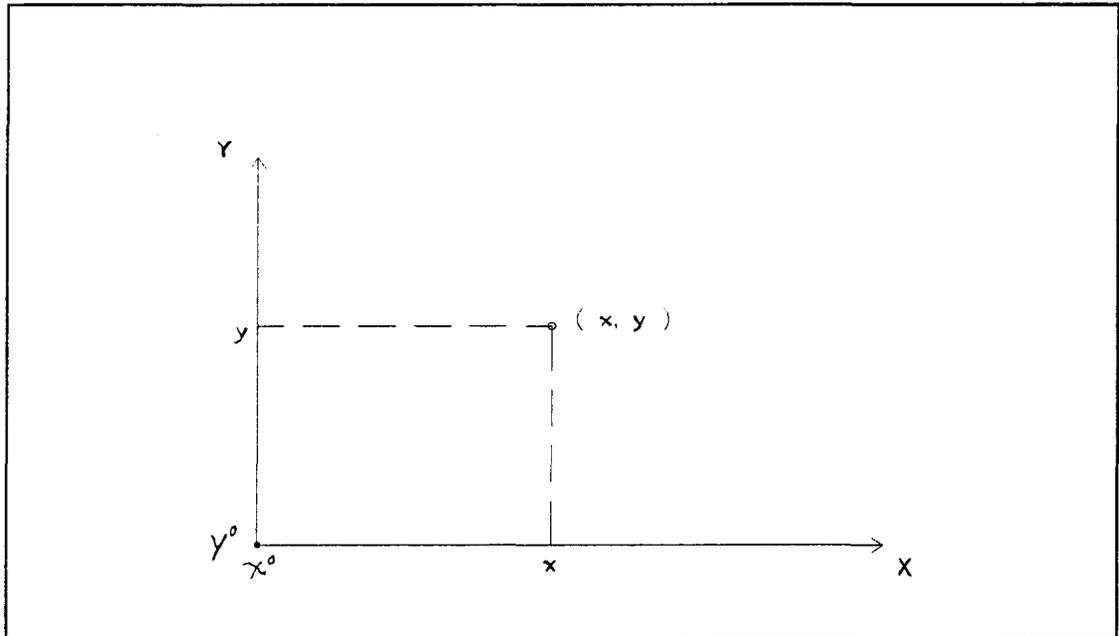


Figure 2.2: Système de coordonnées cartésien

est la représentation de frontière pour des objets "bien définis" bi-dimensionnels.

L'entité topologique de base est le point ou noeud. Celui-ci est déterminé par ses deux coordonnées. Chaque coordonnée correspond à un degré de liberté. Pour un point n , on aura: (x_n, y_n) .

La méthode choisie pour définir le contour est la division en éléments de base. Comme nous avons besoin d'éléments pour définir un contour dans un espace à deux dimensions, nous utiliserons des éléments à une dimension.

Chaque élément est identifié par un nom précisant sa forme. De plus, nous avons le nombre de noeuds géométriques nécessaires pour définir l'élément. La figure 2.3 illustre trois types possibles d'éléments finis pouvant servir à modéliser un contour.

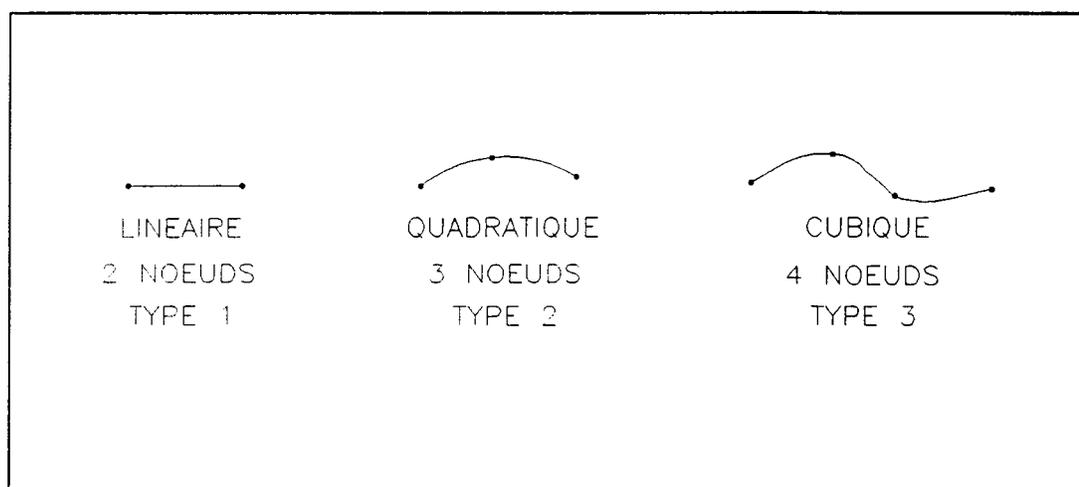


Figure 2.3: Trois types d'éléments à une dimension

Requicha [5.38] fait appel à une représentation de frontière pour son modèle: un objet tridimensionnel est représenté complètement par l'assemblage des éléments bi-dimensionnels qui le composent. Dans notre cas, le même type de représentation est utilisé mais avec une dimension en moins: un objet bi-dimensionnel est représenté complètement par l'assemblage des éléments unidimensionnels qui le composent.

Un des avantages d'utiliser cette méthode est de pouvoir représenter n'importe quel contour complexe à partir d'éléments de base simples. Le

Le système est évolutif car on peut créer de nouveaux éléments de base selon les besoins. La mathématique nécessaire à la manipulation de ces éléments nous est fournie par la méthode des éléments finis.

2.4.1.1 TABLES DE DÉFINITION DES NOEUDS ET ÉLÉMENTS

Les noeuds géométriques sont numérotés séquentiellement de 1 à n. Chaque noeud est défini par ses coordonnées x et y. On représente les noeuds dans un vecteur:

$$\mathbf{x} = \langle x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_n, y_n \rangle \quad (2.1)$$

Les éléments sont numérotés séquentiellement de 1 à e. Chaque élément est défini par la liste des numéros de ses noeuds géométriques. On représente les noeuds formant les éléments dans un vecteur:

$$\mathbf{n} = \langle \begin{array}{cccccc} i_{1,1}, & i_{1,2}, & i_{1,3}, & \dots & i_{1,n}^e, \\ i_{2,1}, & i_{2,2}, & i_{2,3}, & \dots & i_{2,n}^e, \\ i_{3,1}, & i_{3,2}, & i_{3,3}, & \dots & i_{3,n}^e, \\ \dots & \dots & \dots & \dots & \dots \\ i_{e,1}, & i_{e,2}, & i_{e,3}, & \dots & i_{e,n}^e, \end{array} \rangle \quad (2.2)$$

où: i est le numéro de noeud
 e est le nombre maximum d'éléments
 n^e est le nombre maximum de noeuds pour l'élément e

Un dernier vecteur est défini pour avoir le type d'élément utilisé:

$$\mathbf{t} = \langle t_1, t_2, t_3, \dots, t_e \rangle \quad (2.3)$$

2.4.1.2 SENS DE LA NUMÉROTATION

Il est important d'adopter un sens de numérotation, notamment pour le calcul vectoriel. Le sens adopté est le sens trigonométrique (sens inverse des aiguilles d'une montre). Ainsi, les noeuds du contour de la forme seront numérotés dans ce sens à partir d'un noeud de départ quelconque.

Si le modèle de l'objet comporte des "trous", c'est à dire des contours intérieurs, ils seront numérotés en sens contraire du sens trigonométrique.

2.4.2 CONTRAINTES

La méthode de la géométrie variationnelle est utilisée pour l'expression des contraintes. Le concept unique de contrainte remplace les différentes méthodes de cotation (grandeur, position, forme, ...) telles qu'utilisées en dessin technique. Toutes les contraintes doivent être décrites d'une façon explicite. Ceci ne signifie pas que le concepteur doive toutes les spécifier car un bon système de CAO peut prendre en charge les contraintes implicites. Nous prenons comme point de départ l'ensemble complet des contraintes telles qu'elles pourraient être générées d'une façon explicite par un système de CAO.

Les contraintes sont des équations qui lient les degrés de liberté entre eux. On doit avoir autant de contraintes qu'il y a de degrés de liberté pour avoir un schème admissible. Les paramètres des équations de contrainte sont définis par:

$$C_1, C_2, C_3, \dots C_n. \quad (2.4)$$

2.4.2.1 ORDRE DES CONTRAINTES

En théorie, l'ordre d'application des contraintes à la topologie n'a pas d'importance. Le système d'équation peut être résolu dans n'importe quel ordre. En pratique toutefois, l'ordre peut avoir une certaine importance. Selon la méthode matricielle de solution choisie, un ordre peut être nécessaire afin d'arriver à une solution (McComber [4.1]).

L'ordre suivant a été choisi: les premières contraintes sont celles qui s'appuient directement sur la référence. Les autres contraintes s'appuieront les unes sur les autres à partir de ces premières contraintes. Cet ordonnancement permet de bâtir une structure arborescente pour les contraintes. Comme nous le verrons, cet ordre est très important quand nous introduirons les tolérances, puisqu'à chaque contrainte, une tolérance est associée.

2.4.2.2 CONTRAINTES QUI DÉFINISSENT LA RÉFÉRENCE

Les notions de référence ("datum") et de système de référence sont bien décrits dans les standards courants (ANSI 1973). Les références sont utilisées pour construire des systèmes de coordonnées partiels ou complets qui, par la suite, sont utilisés comme références de positionnement pour les entités géométriques des objets.

Dans notre cas, nous utiliserons la théorie proposée par Requicha [5.38]. Celui-ci spécifie qu'il ne doit y avoir qu'un système de référence principal bien défini et complet. Toutes les entités géométriques sont localisées par rapport à ce système de référence. Si on utilise d'autres systèmes de référence, ceux-ci doivent être définis par rapport au système de référence principal.

Dans le cas d'une géométrie dans un domaine en deux dimensions, on doit avoir une translation nulle de la géométrie et une rotation nulle de la géométrie. Cette translation nulle doit être faite selon deux axes, ce qui implique deux contraintes:

- translation nulle de la géométrie selon x

$$\text{- fixer un degré de liberté en x: } x_1 - x_1^0 - C_1 = 0 \quad (2.5)$$

- translation nulle de la géométrie selon y

$$\text{- fixer un degré de liberté en y: } y_1 - y_1^0 - C_2 = 0 \quad (2.6)$$

Afin d'avoir une rotation nulle de la géométrie, on doit fixer un degré de liberté soit en x ou en y ce qui implique une contrainte:

- rotation nulle de la géométrie selon x ou y

- fixer un degré de liberté en x ou en y:

$$x_2 - x_2^0 - C_3 = 0 \text{ ou } y_2 - y_2^0 - C_3 = 0 \quad (2.7)$$

2.4.2.3 AUTRES CONTRAINTES PERMETTANT DE FIXER COMPLÈTEMENT LA GÉOMÉTRIE

Nous développerons ici cinq types de contraintes. D'autres types sont possibles. Elles doivent cependant toutes pouvoir s'exprimer par une équation algébrique égale à une constante. Cette constante est aussi appelée paramètre.

- Distance horizontale entre deux noeuds

Soit C_4 la distance entre les premiers degrés de liberté des deux points (x_1, y_1) et (x_2, y_2) , on aura:

$$\sqrt{(x_2 - x_1)^2} - C_4 = 0$$

(2.8)

- Distance verticale entre deux noeuds

Soit C_5 la distance entre les deuxièmes degrés de liberté des deux points (x_1, y_1) et (x_2, y_2) , on aura:

$$\sqrt{(y_2 - y_1)^2} - C_5 = 0 \quad (2.9)$$

- Distance linéaire entre deux noeuds

Soit C_6 la distance entre deux points (x_1, y_1) et (x_2, y_2) , on aura:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} - C_6 = 0 \quad (2.10)$$

- Angle entre deux éléments linéaires

Soit C_7 l'angle entre les deux éléments, cette contrainte sert aussi à exprimer deux éléments perpendiculaires: $C_7 = 90^\circ$ et deux éléments parallèles: $C_7 = 0^\circ$. L'angle C_7 est trouvé en exprimant les éléments sous forme de vecteur. Par le produit scalaire on aura:

$$\cos^{-1}\left(\frac{xy}{|x||y|}\right) - C_7 = 0 \quad (2.11)$$

où le produit scalaire de deux vecteurs est défini par:

$$x \cdot y = x_1 y_1 + x_2 y_2$$

(2.12)

et la magnitude ou grandeur de deux vecteurs est définie par:

$$|x| = \sqrt{(x_1^2 + x_2^2 + \dots + x_n^2)}$$

(2.13)

Par le produit vectoriel on aura:

$$\sin^{-1}\left(\frac{|x \times y|}{|x| |y|}\right) - C_7 = 0$$

(2.14)

où le produit vectoriel de deux vecteurs est défini par:

$$x \times y = (x_2 y_3 - x_3 y_2, x_3 y_1 - x_1 y_3)$$

(2.15)

Par le produit scalaire et vectoriel, on aura:

$$\tan^{-1}\left(\frac{|x \times y|}{(x \cdot y)}\right) - C_7 = 0$$

(2.16)

- Distance perpendiculaire entre un noeud et un élément linéaire

Soit C_8 la distance perpendiculaire entre un point (x_3, y_3) et un élément linéaire défini par (x_1, y_1) et (x_2, y_2) , on aura:

$$\sqrt{(x_3-x_1)^2+(y_3-y_1)^2}\sin(\tan^{-1}(\frac{|x \times y|}{x \cdot y})) - C_8 = 0$$

(2.17)

2.4.3 TOLÉRANCES

L'information donnée par les tolérances est essentielle pour planifier la fabrication de pièces et les opérations d'assemblage, ainsi que pour d'autres opérations de conception et de production dont l'inspection des pièces. Nous avons besoin ici d'information sur les tolérances pour l'inspection visuelle de pièces ou objets.

Requicha [5.38] fait état d'un manque de facilités pour inclure les informations sur les tolérances dans les systèmes de CAO existants. Dans la pratique courante, les standards existants n'offrent pas de définition formelle et sont définis principalement pour les situations spéciales. Le besoin de définir plus formellement la mathématique des tolérances est évident si on veut les incorporer aux systèmes de CAO.

Requicha [5.38] propose une théorie mathématique pour l'utilisation des tolérances. Sa théorie est d'abord basée sur la modélisation pour la représentation non ambiguë d'un solide. Ce solide est défini par une représentation de surface: le solide est décomposé en éléments de surface bi-dimensionnels. Les tolérances définissent mathématiquement une classe d'objets qui sont (1) interchangeables dans les opérations d'assemblage et (2) fonctionnellement équivalentes. De telles classes d'objets sont appelées "classes variationnelles".

Lin [3.1] n'aborde pas le sujet des tolérances.

Le présent exposé propose une théorie formelle utilisable pour inclure les informations de tolérances au modèle mathématique développé au cours de ce chapitre. La théorie découle de celle proposée par Requicha [5.38] mais appliquée au modèle variationnel développé par Lin [3.1]. Ainsi, nous ne définirons aucun classement particulier des tolérances telle que tolérance de grandeur, forme, position, etc. Nous relierons plutôt les tolérances aux équations de contraintes telles que définies dans notre modèle variationnel. Un classement pourrait être fait selon le type d'équation utilisé.

2.4.3.1 THÉORIE DES TOLÉRANCES

La tolérance est définie comme étant la variation permise sur un degré de liberté. Mathématiquement, les degrés de liberté sont liés entre eux par les équations de contrainte. On a autant d'équations de contrainte qu'il y a de degrés de liberté. La tolérance sera la variation appliquée sur un paramètre de chaque équation de contrainte.

Soit le paramètre C_n et T_n la tolérance permise sur ce paramètre:

$$C_n - T_{ni} \leq C_N \leq C_n + T_{no}$$

(2.18)

Les tolérances permises sur les équations de contrainte peuvent être aisément ramenées en variation sur les degrés de liberté; pour chaque équation, on doit déterminer les degrés de liberté de référence. Les tolérances peuvent être décomposées selon les axes d'application et appliquées aux degrés de liberté restants. Si les degrés de liberté de référence ont déjà une tolérance, celle-ci est aussi ajoutée aux degrés de liberté restants. On voit ici que l'ordre des contraintes a une importance: lorsque la tolérance sur un paramètre d'une contrainte est appliquée aux degrés de liberté, à mesure qu'on s'éloigne de la référence, les tolérances s'additionnent.

Nous n'irons pas plus avant dans cette discussion puisque nous prenons

comme point de départ un schème déterminé lors de l'étape de conception. Les tolérances doivent aussi être définies lors de cette étape.

La définition de la tolérance que l'on vient de donner n'est pas complète en soi. Elle ne s'applique qu'aux noeuds où sont définis les degrés de liberté.

Si on utilise ce modèle pour vérifier les dimensions d'un objet réel, rien ne permet de vérifier si un point faisant partie d'un élément entre deux noeuds respecte les limites de tolérance.

Pour avoir une définition complète, nous devons ajouter: tous les points composants un élément doivent satisfaire les tolérances interpolées entre les deux noeuds adjacents.

Comme la tolérance n'est pas définie formellement pour chaque point d'un élément, on pose que la tolérance d'un point particulier est déterminée par interpolation linéaire entre les valeurs de tolérance des deux noeuds adjacents.

Cette définition semble mener aux mêmes conclusions que Requicha [5.38]: celui-ci définit des "solides décalés" et il base des règles sur notions de conditions de matériel maximum (MMC) et condition de matériel minimum (LMC).

Le modèle des tolérances qui vient d'être défini est complet: nous devons avoir une variation permise sur chacun des degrés de liberté. Cette tolérance est spécifiée sur un paramètre de chacune des équations de contraintes lors de l'étape de conception. Le modèle des tolérances est complété par l'interpolation linéaire des tolérances à l'intérieur des éléments.

Il faut mentionner aussi que l'application des tolérances sur la géométrie pourrait amener un changement de la topologie de l'objet. Par exemple, une pièce avec deux trous rapprochés: si les tolérances sur la grandeur des trous permettent à ceux-ci de se rejoindre, la topologie n'est plus la même. Nous posons l'hypothèse que la grandeur des tolérances acceptées ne permet pas un tel changement.

2.4.3.2 TOLÉRANCES SUR LES CONTRAINTES QUI DÉFINISSENT LA RÉFÉRENCE

Si on permet une variation sur la contrainte C_1 et C_2 , on aura:

Tolérance minimum sur C_1 :

$$x_1 - x_1^0 - C_1 = \delta_i C_1$$

(2.19)

Tolérance maximum sur C_1 :

$$x_1 - x_1^0 - C_1 = \delta_o C_1 \quad (2.20)$$

Tolérance minimum sur C_2 :

$$y_1 - y_1^0 - C_2 = \delta_i C_2 \quad (2.21)$$

Tolérance maximum sur C_2 :

$$y_1 - y_1^0 - C_2 = \delta_o C_2 \quad (2.22)$$

Définies d'une façon informelle, les tolérances appliquées sur les contraintes de translation nulle sont les variations permises sur le positionnement de l'objet.

Si on permet une variation sur la contrainte C_3 , on aura:

Tolérance minimum sur C_3 :

$$x_2 - x_2^0 - C_3 = \delta_i C_3 \quad (2.23)$$

ou

$$y_2 - y_2^0 - C_3 = \delta_i C_3 \quad (2.24)$$

Tolérance maximum sur C_3 :

$$x_2 - x_2^0 - C_3 = \delta_o C_3 \quad (2.25)$$

ou

$$y_2 - y_2^0 - C_3 = \delta_o C_3 \quad (2.26)$$

2.4.3.3 TOLÉRANCES SUR LES AUTRES CONTRAINTES

Les tolérances minimum et maximums s'appliquent de la même façon que pour les contraintes qui définissent la référence. Pour chacune des contraintes C_4 , C_5 , C_6 , C_7 et C_8 , on a une tolérance minimum et une tolérance maximum.

2.5 RÉSUMÉ DU MODÈLE VARIATIONNEL

Le modèle qui a été développé au cours de ce chapitre est basé sur la géométrie variationnelle. La théorie des tolérances a été ajoutée au modèle afin de le compléter. Le modèle est basé sur la représentation de frontière et permet de représenter des objets bi-dimensionnels. La définition de l'objet par la division en éléments s'inspire de la méthode des éléments finis.

La théorie développée au cours de ce chapitre a sûrement besoin de raffinements, peut nécessiter des modifications substantielles et pourrait aussi se révéler totalement inadéquate. La discussion a été limitée à quelques éléments simples et quelques équations de contraintes. Le modèle obtenu est cependant suffisant pour une utilisation en vue de la vérification dimensionnelle à l'aide de la vision par ordinateur. Le modèle permet de définir formellement un objet en terme de topologie, de contraintes et de tolérances sur les contraintes.

Le développement de la théorie amène à se poser certaines questions:

- Est-ce que la représentation par éléments permet de représenter n'importe quel type d'objet ? Est-ce que quelques éléments permettront de représenter à peu près n'importe quel type d'objet ou de nouveaux

types d'éléments devront être constamment utilisés ?

- Sept types de contraintes ont été développées. Est-ce que tous les types de contraintes sont définissables facilement en équations ?

- La théorie a été développée en deux dimensions. Est-ce que l'extension en trois dimensions est réalisable pratiquement ?

- Est-ce que la théorie satisfait adéquatement le besoin réel au niveau de la conception ? du domaine de la vérification dimensionnelle (Par exemple dans l'industrie) ? Est-ce que la théorie peut être prise en charge par ordinateur ?

CHAPITRE III

VISION PAR ORDINATEUR

3.1 INTRODUCTION

Ballard et Brown [2.1] définissent la vision par ordinateur comme étant la construction d'une description explicite et significative d'un objet à partir d'une image.

La figure 3.1 présente les différentes étapes du processus de vision appliqué à l'ordinateur. Tout au long de ce chapitre, la théorie qui s'applique au processus de vision nécessaire au présent mémoire sera présentée. Ceci permettra de mieux connaître les possibilités et limites actuelles. Dans le cadre de ce travail, une description explicite et significative doit permettre d'évaluer les dimensions d'un objet. La présentation permettra de mieux saisir à quel(s) endroit(s) il est possible de faire le lien entre ce processus de vision par ordinateur et un modèle paramétrique afin d'en arriver à la réalisation de l'inspection dimensionnelle.

Le processus de vision décrit à la figure 3.1 suppose l'utilisation interne

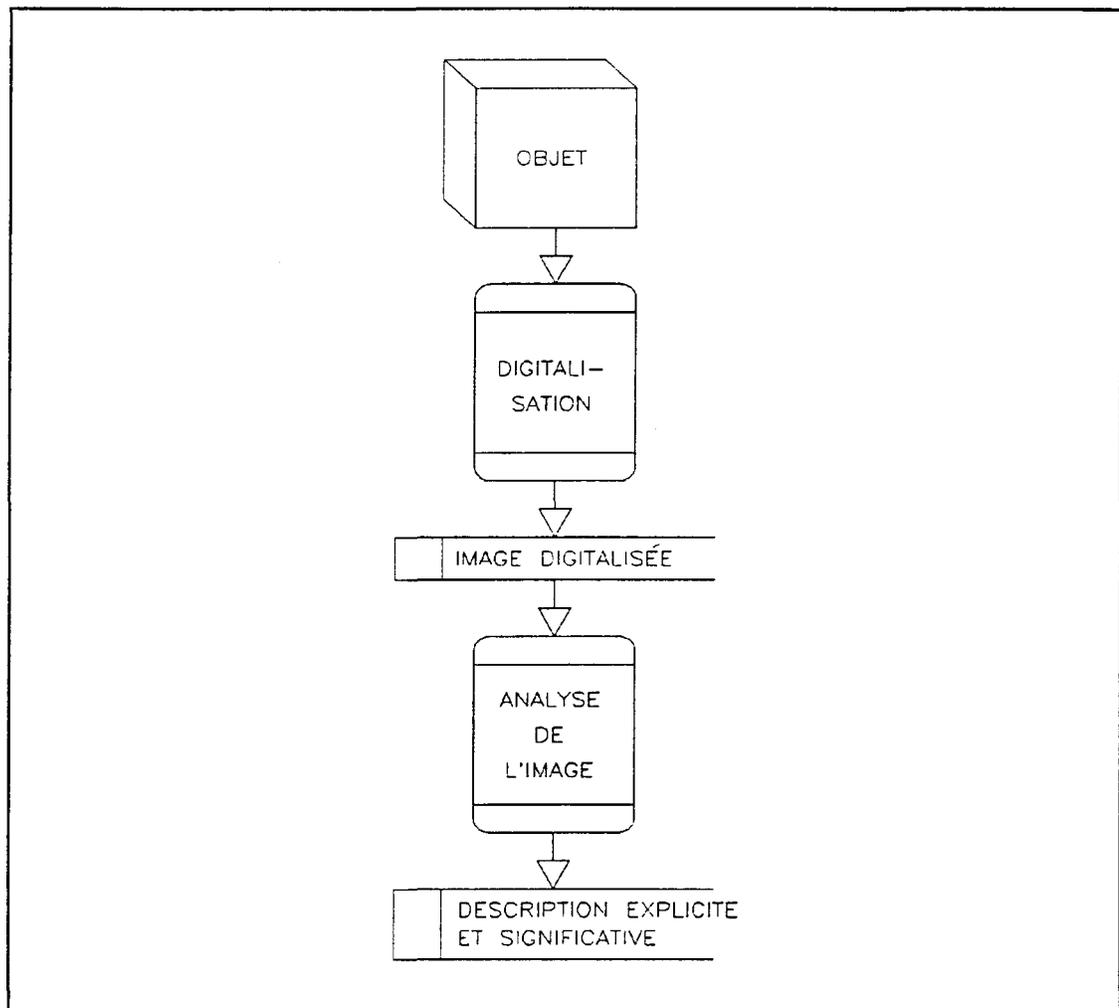


Figure 3.1: Processus de vision par ordinateur

d'un modèle lors de l'analyse pour arriver à une description explicite et significative d'une image.

L'analyse de l'image est divisée en plusieurs étapes intermédiaires. A chacune de ces étapes correspond une représentation. Le processus d'analyse implique la construction de ces représentations intermédiaires et la réalisation des algorithmes pour les construire et les relier entre elles. Nous verrons comment l'utilisation d'un modèle est présente à chacune des ces étapes.

La présentation du chapitre suit sensiblement l'ordre du processus décrit à la figure 3.1. D'abord, le processus de digitalisation de l'objet menant à l'obtention d'une image digitalisée est analysé. L'analyse de l'image est divisée en deux parties. L'analyse primaire part de l'image digitalisée jusqu'à l'obtention d'une représentation secondaire: la description de la forme ou contour de l'objet. L'analyse secondaire part de cette représentation et fait le lien avec le modèle paramétrique pour l'obtention des résultats. Deux méthodes seront présentées pour l'analyse secondaire: la première méthode fait le lien direct entre la représentation du contour et le modèle. La deuxième méthode fait d'abord la transformation de la représentation du contour en une représentation selon le modèle paramétrique. Le lien entre cette dernière représentation et le modèle est alors fait. Une discussion sur les avantages et désavantages de chacune des deux méthodes est présentée. Finalement, une discussion sur l'utilisation du modèle tout au long du processus de vision est faite séparément.

3.2 ACQUISITION D'IMAGES

3.2.1 INTRODUCTION

Il sera ici question du processus de vision à partir de l'objet jusqu'à l'obtention d'une représentation de l'objet sous un format traitable par ordinateur (image digitalisée). Le processus de digitalisation sera expliqué en se basant sur le théorème d'échantillonnage. Le calcul des dimensions de l'objet étant réalisé avec l'image digitalisée, le but de la digitalisation est donc d'obtenir une image qui soit la représentation la plus fidèle possible de l'objet. En effet, le processus de digitalisation est imparfait: la technologie utilisée peut créer de la distorsion ou faire apparaître des défauts qui n'existent pas sur l'objet et ce, indépendamment de la perte d'information due à l'échantillonnage. Mentionnons que la méthode utilisée ici doit être adaptée au type d'objet que l'on digitalise et au traitement subséquent que l'on désire réaliser sur l'image digitalisée.

3.2.2 DIGITALISATION DE L'IMAGE

Le processus de digitalisation est présenté à la figure 3.2. Ce processus peut être scindé en plusieurs étapes qui seront expliquées en détail.

La formation de l'image survient quand un capteur (ici de type CCD)

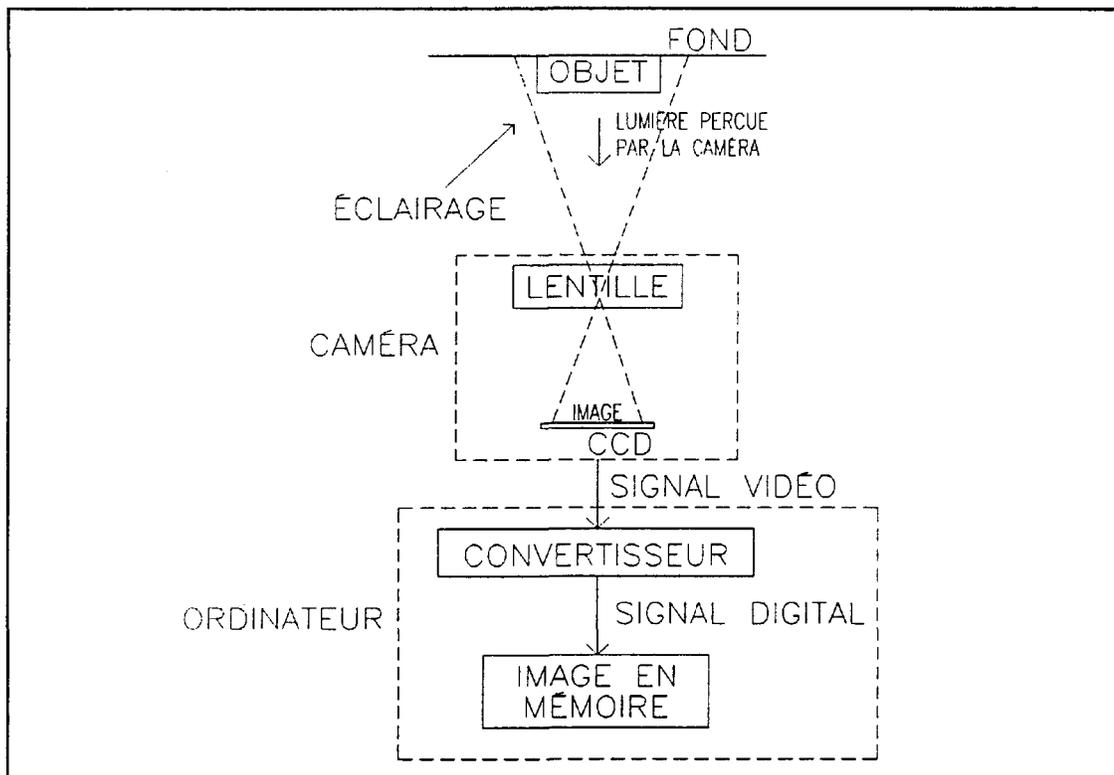


Figure 3.2: Processus de digitalisation

enregistre une radiation (ici la lumière visible) qui a interagi avec un objet physique. L'image est formée par la différence d'intensité de la lumière réfléchiée par l'objet et le fond sur lequel celui-ci est posé. Cette lumière réfléchiée peut provenir de sources primaires (lampes d'éclairage) ou encore de lumière réfléchiée par d'autres surfaces. La lumière passe au travers d'une lentille et est focalisée sur une plaque CCD tel que présenté à la figure 3.2. Cette plaque CCD est une matrice de $m \times n$ éléments qui convertissent la lumière recue en courant électrique. Les CCD (charge-coupled devices) ressemblent aux MOSFETs (metal-oxyde semiconductor field-effect transistor) en ce qu'ils sont un type de transistor semblable. Pour les besoins de la vision, les CCD peuvent être considérés comme une matrice rectangulaire

monolithique de condensateurs MOS distribués régulièrement pour former un registre à décalage. Le signal de chaque élément de la matrice est recueilli à tour de rôle. Un balayage complet de la matrice est fait dans une période de temps T . Ce courant est modulé par la caméra en un signal vidéo. La fréquence de balayage complet de la matrice CCD est de 30 fois par seconde. Le signal vidéo est ensuite envoyé à un convertisseur qui numérise le signal et enregistre le résultat en mémoire. La valeur numérique résultante pour chaque élément dépend de l'intensité du signal électrique.

Il ne sera pas discuté ici des fréquences d'acquisition puisque le présent travail ne couvre que la mesure d'objets statiques.

Si on analyse ensuite le processus au niveau théorique, il s'agit d'un échantillonnage spatial fini combiné à un échantillonnage d'intensité fini. Le théorème d'échantillonnage de Nyquist décrit par Negin [5.27] dit:

"Si un signal à bande limitée est échantillonné ponctuellement à au moins deux fois la valeur de la plus haute fréquence présente dans le signal, alors le signal peut être complètement reconstruit à partir des données d'échantillonnage."

Bien que le théorème de Nyquist original fut conçu pour utiliser avec un signal à une dimension, son application et son extension à des signaux bi-dimensionnels (images) va de soi. Cependant, si une longueur de mot finie est utilisée, alors théoriquement, le signal ne peut être complètement reconstruit.

Mathématiquement, une image digitalisée peut être définie par une matrice rectangulaire:

$$I_{m \times n} \quad \text{où} \quad I(x,y) = g_{xy} \quad (3.1)$$

- où:
- g = niveau de gris = 1, 2, ... 2^{nb}
 - nb = nombre de bits pour enregistrer le niveau de gris
 - m = nombre de colonnes de l'image
 - n = nombre de lignes de l'image
 - x = colonne = 1, 2, ... m
 - y = ligne = 1, 2, ... n

Chaque élément de la matrice est appelé pixel pour "picture element". L'image digitale est donc formée d'une mosaïque de pixels dont chacun correspond à un élément de la matrice. La discussion concernant la précision possible avec ce type d'image est faite en 3.4.3.4.

3.2.3 PROBLÈMES D'INTENSITÉ ET LEUR CORRECTION

En plus de l'éclairage qui peut causer des erreurs d'intensité, d'autres causes sont possibles: le bruit présent dans le système de vision, des défauts dans la matrice CCD, les conditions de prise d'image, la lentille de la caméra, la précision du convertisseur vidéo ... et cette liste n'est pas complète.

Toutes ces causes sont difficilement contrôlables. Deux solutions sont possibles pour minimiser les problèmes d'intensité: réduire les causes au minimum en utilisant le meilleur matériel possible et en contrôlant l'environnement ou encore utiliser un procédé de calibrage.

Le but visé est d'obtenir une image digitalisée dont les différents niveaux d'intensité (aussi appelés niveaux de gris) correspondent aux couleurs de l'objet et du fond sur lequel cet objet est posé. Afin de simplifier le problème, on prend comme hypothèse que l'objet est d'une intensité uniforme ainsi que le fond sur lequel il est posé, mais que les deux intensités sont différentes. La couleur est parfaitement uniforme si on a une surface mate parfaite ainsi que le démontre Horn [5.10]. Afin d'avoir le résultat voulu, on doit donc éclairer la scène (objet et fond) d'une façon uniforme.

La figure 3.3 présente la fonction de transfert d'une caméra idéale. Si

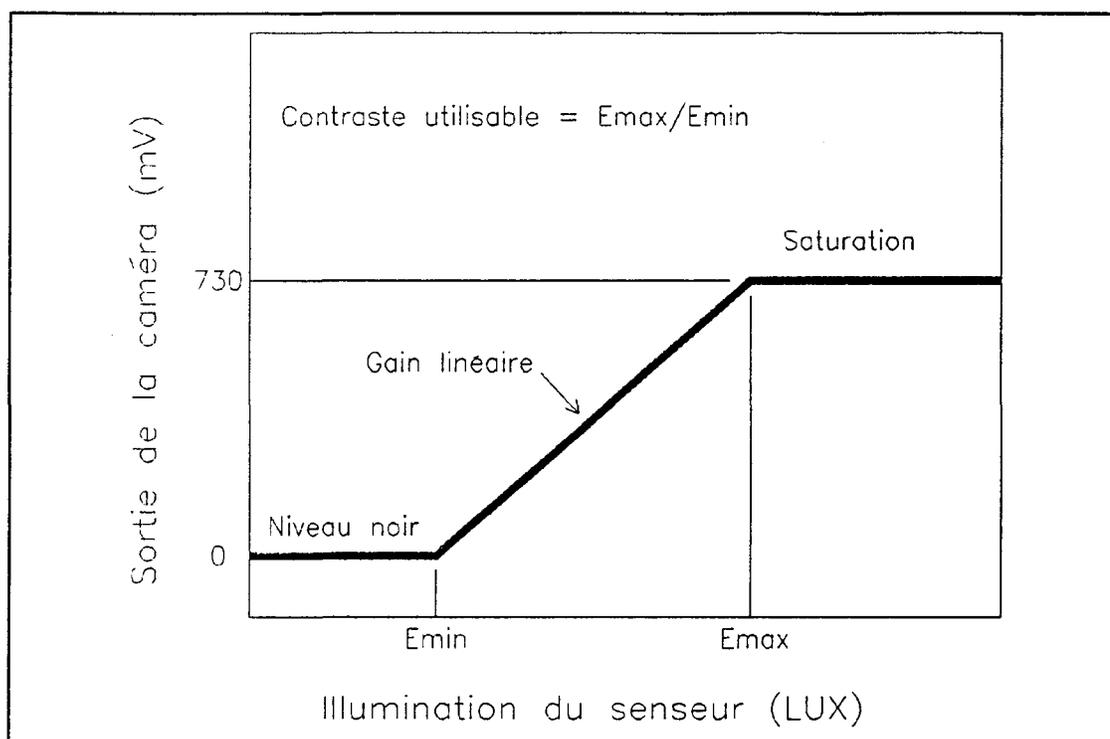


Figure 3.3: Fonction de transfert d'une caméra idéale

on désire utiliser toute la plage d'utilisation de la caméra, les parties les plus claires d'une scène devraient causer une illumination du capteur qui est juste au-dessous du niveau de saturation de la caméra et les parties les plus sombres une illumination qui est au-dessus du niveau de bruit du système de vision. On ne doit pas dépasser le niveau de saturation car à ce niveau, l'illumination peut causer un déplacement du signal à des pixels adjacents (bloom smear). Aussi, plusieurs caméras souffrent d'un "niveau de noir" (dark level) qui peut varier en fonction de l'illumination moyenne du capteur. La figure 3.4 présente la partie (plage) utilisable de la fonction de transfert réelle d'une caméra. Il est important de noter que la fonction est non-linéaire. Le lecteur trouvera à l'annexe A les fonctions de transfert de quelques caméras à technologie CCD

disponibles commercialement.

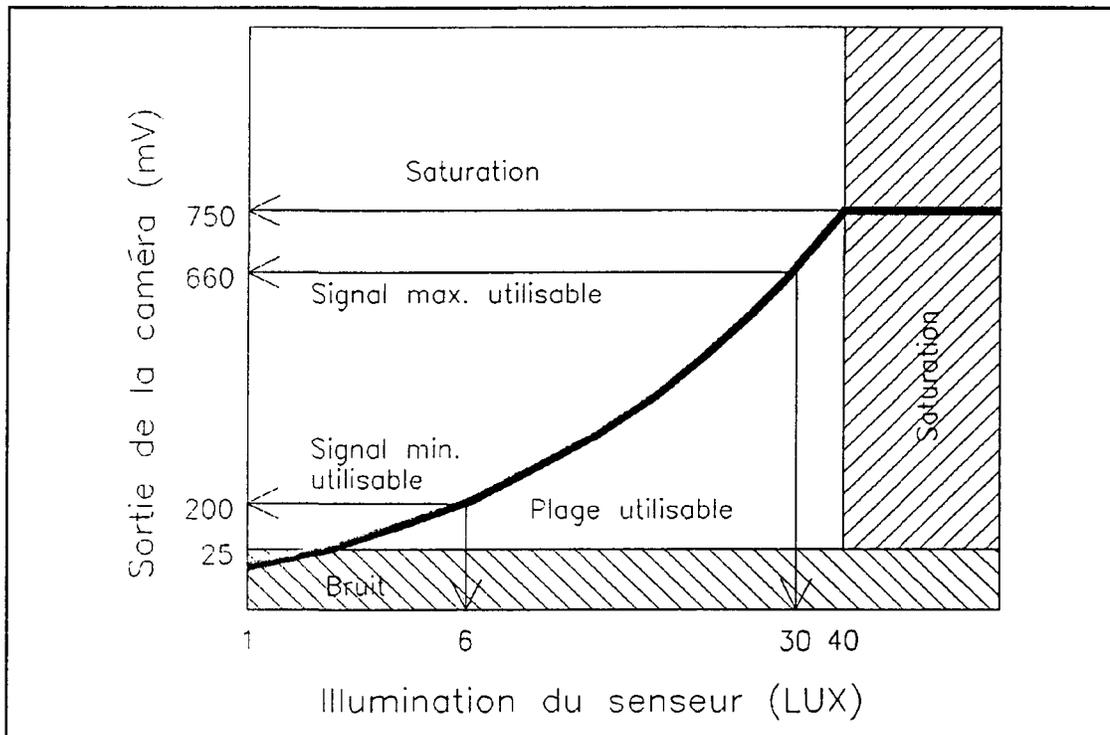


Figure 3.4: Plage utilisable de la fonction de transfert réelle

La méthode de calibrage suivante est proposée:

- Faire l'acquisition de l'image d'une surface d'intensité moyenne uniforme.
- Déterminer un modèle de caméra pour transformer les intensités trouvées en une intensité uniforme pour toute l'image. Par exemple, il suffit de soustraire l'image de référence plus une constante pour calibrer les images qui seront digitalisées.

Cette méthode a une précision limitée car la fonction de transfert réelle n'est pas linéaire.

La méthode proposée pour réduire le bruit est de prendre plusieurs images et de faire une moyenne par pixel sur la somme des images. En théorie, si on fait la moyenne sur un nombre infini d'images, le bruit devrait être complètement éliminé. Malheureusement, un compromis doit être trouvé car un temps d'acquisition et de traitement trop long est inacceptable.

3.2.4 DISTORSION ET CORRECTION DE LA DISTORSION

Un peu comme pour les problèmes d'intensité, de nombreux facteurs entrent en ligne de compte et peuvent causer de la distorsion dans l'image. Les sources de distorsion peuvent provenir de la lentille, de la longueur focale de la lentille, des imperfections et de la précision possible avec une matrice de type CCD. Une autre source est la position et l'alignement de la caméra avec le sujet. Pour une bonne analyse des sources de distorsion, se référer à White [5.16] et Butler et Pierson [5.17].

Deux solutions sont possibles pour réduire la distorsion: minimiser ce qui cause la distorsion ou utiliser une méthode de calibrage. On peut donc utiliser une lentille de qualité avec une grande longueur focale couplée à une caméra

de qualité.

La méthode proposée pour le calibrage est l'utilisation d'une grille. Cette méthode est développée et a une efficacité prouvée expérimentalement par Butler et Pierson [5.17]. La méthode se résume à:

- Faire l'acquisition de l'image d'une cible de calibrage. Cette cible doit comporter des points dont la position est connue dans un système de coordonnées de références.
- Déterminer la position des points dans le système de coordonnées des pixels $I(x,y)$.
- Déterminer un modèle de caméra pour transformer les coordonnées d'un point d'un système à l'autre (coordonnées de références à coordonnées pixels et vice-versa).
- Calculer les constantes du modèle de caméra (ex: échelle, rotation, etc.)

Un exemple complet de modèle avec les fonctions de transformation résultantes est donné par Butler et Pierson [5.17].

3.3 ANALYSE PRIMAIRE

3.3.1 INTRODUCTION

L'image digitalisée est l'image de l'objet transformée de façon à avoir un format analysable par ordinateur. L'analyse sera limitée aux images de type matricielle deux dimensions soit ce qui sera utilisé lors de l'expérimentation. La méthode d'analyse développée ici est présentée dans la littérature. Cependant, ce domaine de recherche est vaste; en général, plus on veut avoir des résultats précis, plus on doit utiliser des techniques exactes et complexes.

La première étape de la méthode est la technique du niveau de séparation global ("global tresholding technique") qui permet de séparer l'objet du fond sur lequel il est posé. Par la suite, un suivi de contour permet de représenter la forme à analyser sous un format intermédiaire. Chacune de ces étapes sera expliquée en détail. Bien que l'utilisation d'un modèle soit présente à chacune des étapes, ce point sera discuté séparément. Il en est de même de la précision de la méthode.

3.3.2 NIVEAU DE SÉPARATION

Cette technique du niveau de séparation ("tresholding") assume que l'image est bi-modale, c'est à dire que l'image contient un objet brillant sur un fond sombre ou vice-versa. L'objectif est de choisir un niveau de séparation "T" qui divise les pixels de l'image soit en objet ou en fond:

Si $I(x,y) > T$ alors

$$I(x,y) = \text{OBJET} \quad (3.2)$$

autrement

$$I(x,y) = \text{FOND} \quad (3.3)$$

où : $T =$ Niveau de séparation global

OBJET et **FOND** sont deux niveaux d'intensité prédéterminés

Une façon de trouver le niveau T dans un cas bi-modal est d'analyser l'histogramme de l'image afin de trouver un minimum entre le maximum représentant l'objet et le maximum représentant le fond. La figure 3.5 illustre la méthode. C'est suffisant dans le cas qui nous intéresse. Dans le cas où l'objet et le fond sont moins différenciés ou encore que la qualité de l'image laisse à désirer, des méthodes complexes sont développées par Chang [5.18] et Ballard et Brown [2.1].

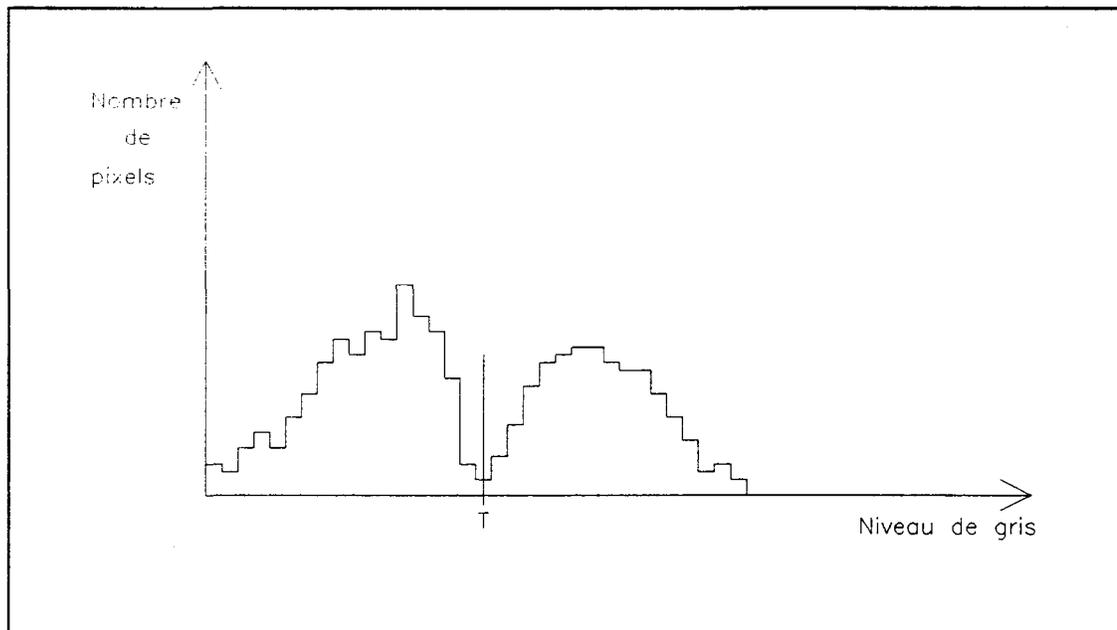


Figure 3.5: Détermination du niveau de séparation T à partir de l'histogramme

3.3.3 SUIVI DE CONTOUR ET REPRÉSENTATION DE LA FORME

Si rien n'est connu sur la forme du contour mais qu'un ou des objets sont présents sur l'image, le contour peut être trouvé par une opération simple de suivi de contour. Avec une image binaire, le but est de trouver le contour de toutes les régions distinctes dans l'image. Ceci peut être réalisé au moyen d'un algorithme simple:

- Balayer l'image jusqu'à ce qu'un pixel **OBJET** soit trouvé
- Si c'est un pixel **OBJET**, tourner à gauche et avancer; sinon tourner à droite et avancer
- Terminer quand on revient au pixel de départ

La figure 3.6 illustre cette procédure. Cet algorithme peut être élaboré

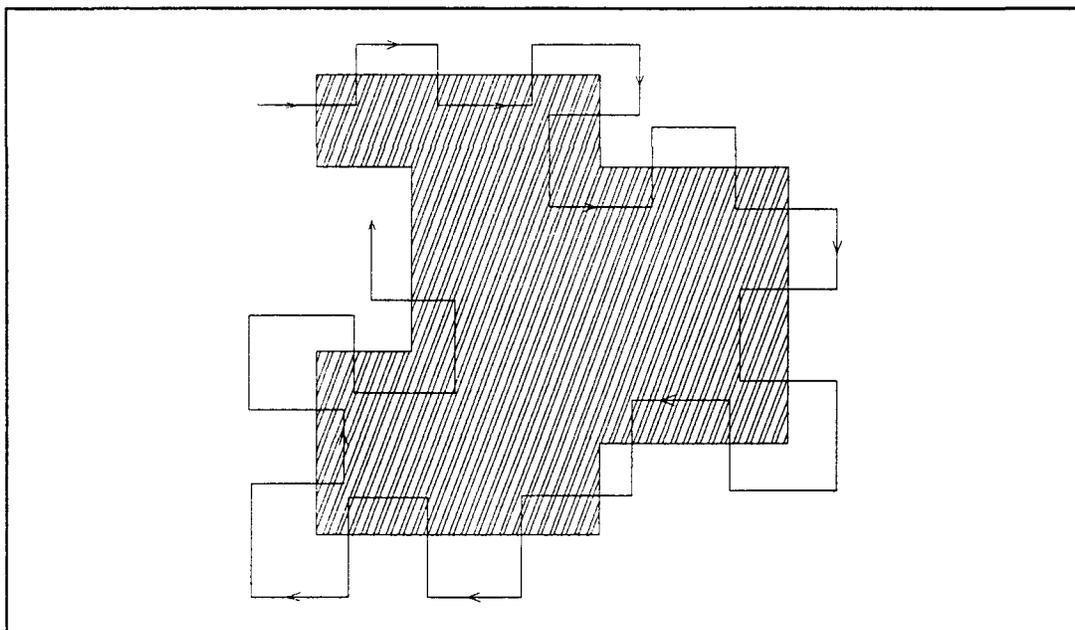


Figure 3.6: Suivi de contour d'une image binaire

afin de s'appliquer aux pixels qui se touchent seulement par un "coin". Il peut aussi être complété afin de trouver tous les contours et aussi trouver s'il y a des trous, c'est à dire des contours à l'intérieur des objets.

Le contour suivi peut être enregistré de différentes manières selon l'objectif poursuivi. C'est ce qu'on nomme la représentation de la forme. Par exemple, si on désire minimiser les données à enregistrer, on peut utiliser le codage en chaîne ("chain coding"). Dans notre cas, la méthode utilisée sera d'enregistrer les coordonnées x et y de chaque point du contour. Cette dernière méthode permet d'utiliser des algorithmes de traitement simples pour le traitement subséquent. Le contour peut être exprimé par un vecteur c :

$$c = \langle x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_n, y_n \rangle \quad (3.4)$$

où: n = nombre de points du contour

Pour en arriver à la description du contour, certaines hypothèses ont été posées. Nous savons que l'objet est posé sur un fond et qu'en déterminant un niveau de séparation T de façon à rendre l'image bi-modale, nous obtenons une image où chaque pixel faisant partie de l'objet est facilement identifiable. Le suivi de contour nous permet par la suite d'obtenir le contour de l'objet exprimé sous forme de coordonnées.

Si nous désirons aller plus avant dans la description de l'objet, nous devons poser d'autres hypothèses. Nous savons que nous désirons pouvoir exprimer l'objet, obtenu expérimentalement par l'image, sous une forme mathématique qui permettra une comparaison avec le modèle mathématique.

La première étape consiste à discriminer sur l'image ce qui fait partie de l'objet ou non. Ce qui ne fait pas partie de l'objet fait partie du fond. Cette étape fait appel à un modèle intrinsèque "objet" et "fond". On ne fait pas appel au modèle externe. Un algorithme de suivi de contour est appliqué à l'image résultante et permet d'obtenir une représentation intermédiaire. Cette représentation n'est que l'obtention d'un format qui permet de réaliser la deuxième étape.

3.4 ANALYSE SECONDAIRE

3.4.1 PRÉSENTATION DES DEUX MÉTHODES

L'analyse secondaire est ici définie comme l'étape qui part de la représentation de contour et qui fait le lien avec le modèle paramétrique pour l'obtention des résultats.

Deux méthodes d'analyse sont présentées. La première méthode fait le lien entre la représentation de contour et le modèle. La deuxième méthode fait d'abord la transformation de la représentation du contour en une représentation selon le modèle paramétrique avant de faire le lien avec le modèle. On a donc une représentation intermédiaire de plus.

Deux méthodes sont présentées car chacune a des avantages et des désavantages. Après une présentation détaillée de chacune des deux méthodes, une discussion permettra de comparer les avantages et désavantages.

3.4.2 PREMIÈRE MÉTHODE: CONDITIONS DE MATÉRIEL

3.4.2.1 DESCRIPTION DES ÉTAPES

La première méthode fait le lien entre la représentation du contour et le modèle. La représentation du contour a été obtenue lors de l'analyse primaire de l'objet.

Cette méthode est basée sur le modèle développé par Requicha [5.38]. Celui-ci définit deux zones de décalage autour d'un objet à partir des tolérances. La première zone est définie par le décalage de la frontière de l'objet de façon à obtenir une condition de matériel maximum ("Maximum Material Condition" ou MMC). La deuxième zone est obtenue d'une façon similaire par un décalage de la frontière de l'objet mais de façon à obtenir une condition de matériel minimum ("Least Material Condition" ou LMC).

La méthode proposée est la suivante: Si le contour de l'objet respecte les conditions de MMC et de LMC, alors toutes les contraintes définies par le modèle sont respectées par l'objet. La figure 3.7 illustre la méthode.

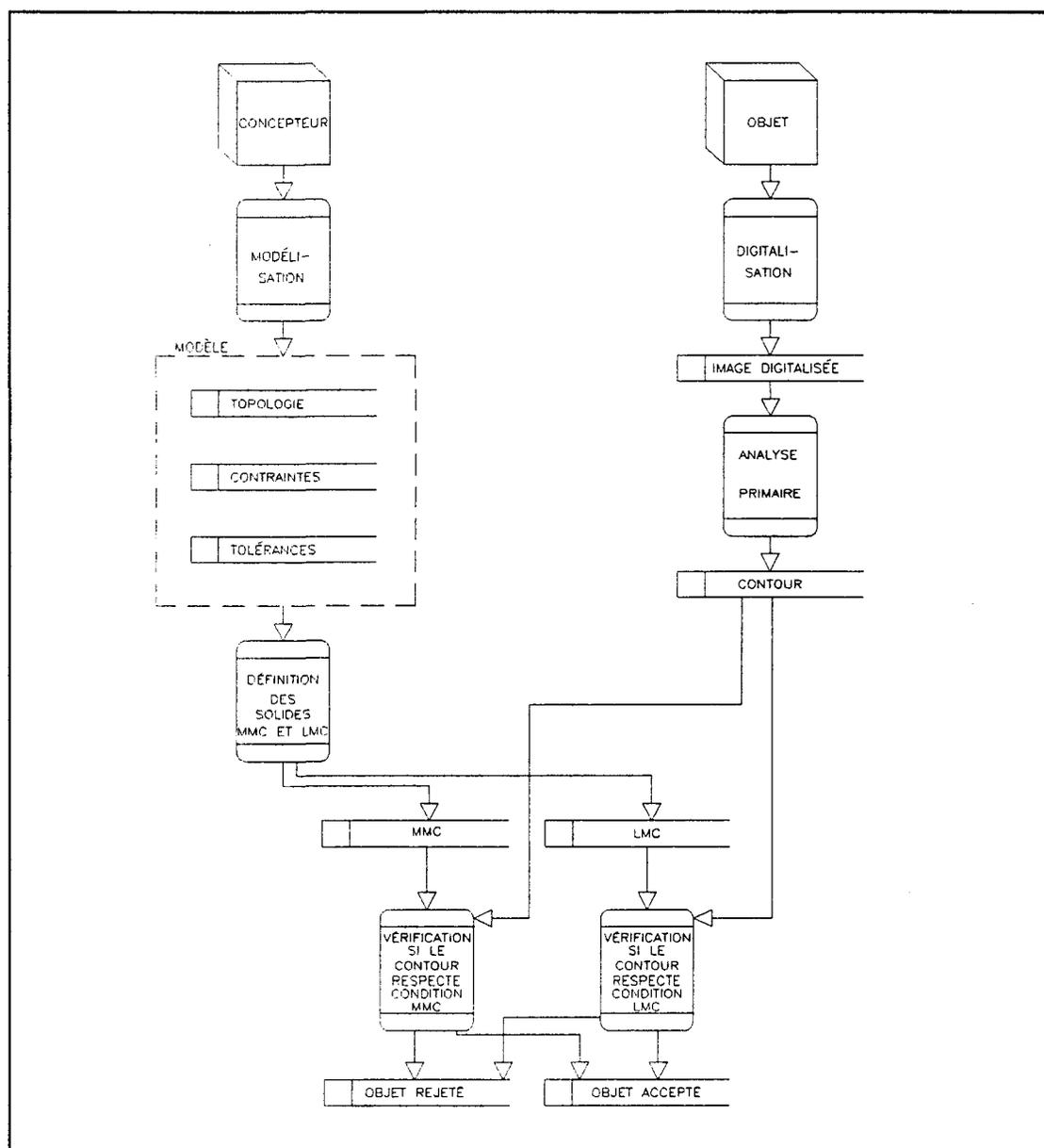


Figure 3.7: Diagramme de la méthode "conditions de matériel"

3.4.2.2 ÉLABORATION DES ALGORITHMES

La méthode présentée à la figure 3.7 demande l'élaboration de quelques algorithmes qui sont simples à expliquer mais qui peuvent être difficiles à réaliser en pratique. Les deux premiers algorithmes nécessaires sont ceux servant à définir les solides MMC et LMC à partir du modèle géométrique variationnel.

L'algorithme proposé pour déterminer le MMC est le suivant:

- 1 - Appliquer la tolérance maximum sur un degré de liberté.
- 2 - Calculer l'aire de l'objet.
- 3 - Appliquer la tolérance minimum sur un degré de liberté.
- 4 - Calculer l'aire de l'objet
- 5 - Conserver la tolérance qui maximise l'aire de l'objet calculée à l'étape 2 ou 4.
- 6 - Si tous les degrés de liberté ont été examinés, arrêter sinon revenir à l'étape 1.

L'algorithme pour déterminer le LMC est semblable sauf que l'on doit minimiser l'aire de l'objet à l'étape 5.

Une fois que les solides MMC et LMC ont été déterminés, on doit vérifier si le contour provenant de l'analyse primaire respecte ces conditions. L'algorithme à développer doit permettre de déterminer si un contour peut être complètement inclus dans un autre. Les points réels ne peuvent être associés à des points particuliers du modèle: il n'y a pas de "rigidité" entre le réel et le modèle. Cependant, il y a "rigidité" entre tous les points du réel.

3.4.2.3 AVANTAGES ET DÉSAVANTAGES

Le principal avantage de la méthode est de savoir avec certitude si l'objet respecte toutes les contraintes ou non. La méthode est simple à comprendre et, à première vue, est simple à mettre en oeuvre; cependant, les algorithmes décrits pourraient être difficiles à réaliser pratiquement.

Le principal désavantage est que la réponse est bi-modale: on sait oui ou non si les contraintes sont respectées globalement. On n'a pas la mesure d'aucun paramètre en particulier et donc on ne sait pas la variation sur les paramètres. On ne peut savoir quelle contrainte n'est pas satisfaite.

Finalement, le temps de calcul pour arriver à une réponse pourrait s'avérer trop long pour être pratique.

3.4.3 DEUXIÈME MÉTHODE: MODÉLISATION DES DONNÉES RÉELLES

3.4.3.1 DESCRIPTION DES ÉTAPES

La deuxième méthode fait la transformation des données réelles sous un format semblable à celui du modèle. Le modèle peut servir de références aux différentes étapes. Les données réelles sont donc "modélisées" avant de pouvoir les comparer au modèle.

La transformation du contour sous un format semblable à celui du modèle entraîne une perte d'information. Cette perte d'information fait que le modèle obtenu n'est qu'une approximation du réel. Le modèle est approché par des méthodes statistiques et ainsi, le résultat ne peut être exact.

Cette méthode peut être divisée en quatre étapes. Lors d'une première étape, le contour obtenu lors de l'analyse primaire est traité afin de l'exprimer par une topologie qui l'approche d'une façon acceptable. Cette étape peut se référer à la topologie exacte du modèle ou encore se référer à un modèle topologique général.

La deuxième étape consiste à introduire les valeurs des degrés de liberté obtenus lors de la première étape dans les équations de contrainte du modèle.

Les équations de contrainte permettent alors de calculer la valeur des paramètres réels.

La troisième étape est de comparer les paramètres réels obtenus lors de la deuxième étape aux paramètres du modèle. Pour chaque paramètre, on peut calculer la différence obtenue en soustrayant la valeur réelle à celle du modèle. Le résultat est la variation réelle sur les degrés de liberté. La quatrième et dernière étape est un simple test qui permet de vérifier si ces variations restent à l'intérieur des limites de tolérance du modèle. Si une ou des variations dépassent les tolérances, l'objet réel ne respecte pas les contraintes. La figure 3.8 illustre la méthode que l'on vient de décrire.

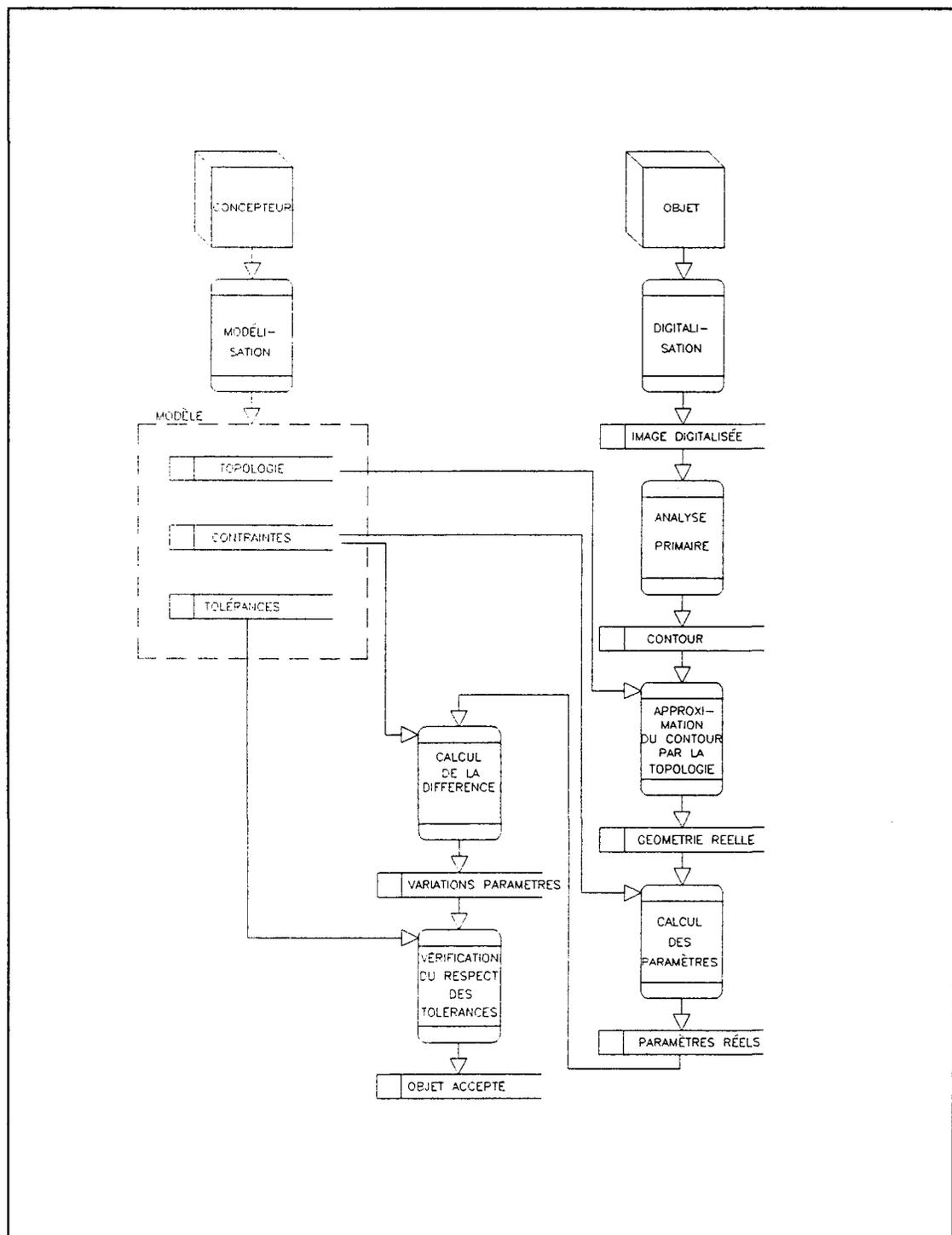


Figure 3.8: Diagramme de la méthode "modélisation des données réelles"

3.4.3.2 ÉLABORATION DES ALGORITHMES

La première étape de la méthode demande l'élaboration d'algorithmes complexes. Différents problèmes sont rencontrés lors de cette étape. La littérature explore assez bien le sujet.

Les principes à la base de cette méthode sont les mêmes que ceux utilisés pour le lissage de courbe ("curve fitting"). On a une série de points qu'on désire représenter par une droite ou un type d'équation déterminé par le type d'élément qu'on utilise comme modèle. Par exemple, si on a un élément linéaire, on peut utiliser la méthode des moindres carrés pour trouver l'équation de la droite qui correspond le mieux aux points.

Le premier point à souligner est qu'il faut adapter les méthodes utilisées dans un espace à deux dimensions. Dans le cas où un seul élément peut représenter le contour, il n'y a pas de problème particulier. Cependant, dans pratiquement tous les cas, le contour à représenter est formé de plus d'un élément. Le problème qui surgit alors est de savoir où se trouve la séparation entre deux éléments: à partir de quel point on doit cesser d'utiliser le points pour modéliser un élément et les utiliser pour modéliser l'élément suivant. La figure 3.9 illustre ce problème qu'on nomme point de séparation ("breakpoint").

Dans un algorithme de fusion ("merging"), les points le long d'un élément

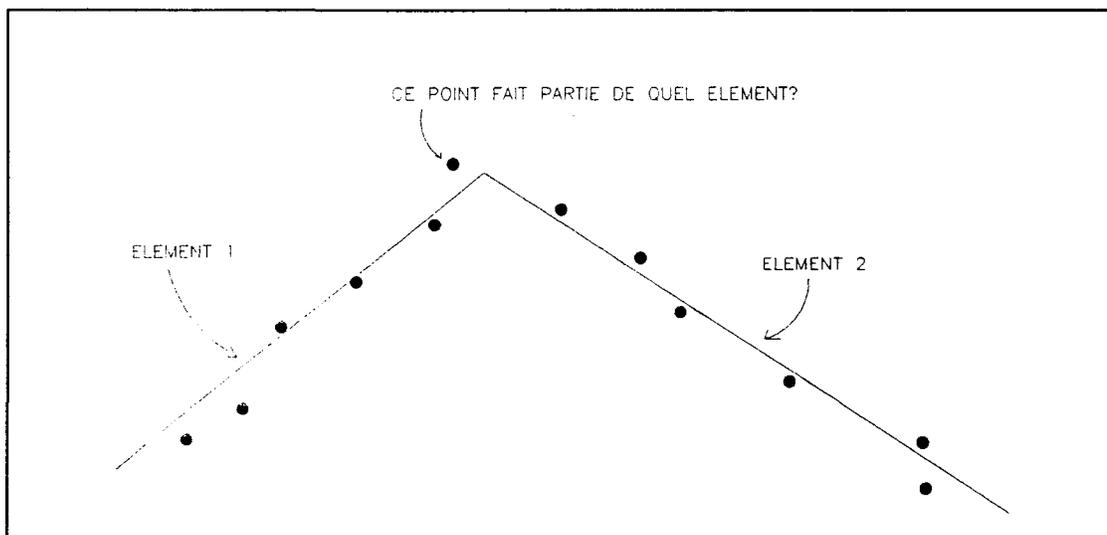


Figure 3.9: Problème du point de séparation entre deux éléments

sont considérés en ordre et acceptés comme faisant partie d'un élément aussi longtemps qu'ils s'en rapprochent suffisamment. Le problème du point de séparation vient du fait que celui-ci est déterminé seulement après que le lissage s'est détérioré, indiquant qu'habituellement, un point de séparation plus rapproché aurait été préférable. Des algorithmes à plusieurs itérations permettent d'obtenir une plus grande précision. Les algorithmes doivent être adaptées aux différents types d'éléments utilisés.

Des algorithmes explicites sont disponibles dans la littérature dont Pavlidis [2.8]. Ici, on fait appel à un modèle extrinsèque à l'image; on utilise le modèle topologique externe obtenu par la modélisation. On pourrait utiliser lors de cette étape un modèle topologique général par opposition à un modèle topologique particulier. Le modèle topologique particulier est la topologie de l'objet telle que définie dans le modèle. Un modèle topologique général est la

description des éléments servant à définir la topologie. L'utilisation d'un modèle topologique général permettrait de trouver une topologie à partir de l'image. Cette topologie ne serait pas nécessairement la même que celle du modèle. Une topologie différente amènerait à un rejet de l'objet. Les tableaux 3.1 et 3.2 décrivent les différences entre l'utilisation d'un modèle topologique général par opposition à un modèle particulier (utilisation de la topologie du modèle). A et B sont deux topologies différentes.

Topologie du modèle	Topologie de l'objet	Topologie trouvée par l'utilisation de la topologie du modèle	Conséquences
A	A	A	Bon modèle: Légère variation sur les paramètres
A	B	A	Mauvais modèle: Très grande variation sur les paramètres

Tableau 3.1: Utilisation de la topologie du modèle

Topologie du modèle	Topologie de l'objet	Topologie trouvée par l'utilisation d'un modèle topologique général	Conséquences
A	A	A	Topologie identique: Légère variation sur les paramètres
A	B	B	Topologie différente: Impossible de faire le lien entre les paramètres

Tableau 3.2: Utilisation d'un modèle topologique général

Les deux façons de faire arrivent au même résultat. Cependant, l'utilisation du modèle topologique de l'objet amène un rejet plus tardif de celui-ci si la topologie de l'objet ne correspond pas à celle du modèle. L'objet ne sera pas rejeté à l'étape de la reconnaissance de la topologie mais dans les

étapes subséquentes quand on vérifiera la variation sur les paramètres. Dans le présent travail, nous posons l'hypothèse que la topologie de l'objet est la même que celle du modèle. L'acceptation ou non de l'objet se fera à partir des tolérances sur les paramètres des contraintes et non à partir de la topologie.

La deuxième étape de la méthode consiste à insérer dans les équations de contrainte du modèle, les valeurs des degrés de liberté trouvées, afin de calculer les paramètres réels. Premièrement, on doit trouver quel degré de liberté réel est relié à quel degré de liberté des équations de contrainte. On pose le paramètre comme variable et il suffit de remplacer les degrés de liberté par leur valeur réelle afin de calculer la valeur réelle de chaque paramètre.

La troisième étape consiste à calculer la différence entre les paramètres réels (calculés) et ceux du modèle. Il suffit simplement de savoir quel paramètre réel est associé à quel paramètre du modèle, ce qui a été déterminé à l'étape précédente.

La dernière étape permet de vérifier si oui ou non, l'objet examiné respecte les tolérances. Chaque variation sur les paramètres est vérifiée en la comparant à la tolérance associée du modèle. Si une ou des variations dépassent la tolérance, l'objet est jugé inacceptable. Si toutes les variations sont inférieures aux tolérances, l'objet est jugé acceptable.

3.4.3.3 AVANTAGES ET DÉSAVANTAGES

Le principal avantage de la méthode de modélisation des données réelles est de pouvoir "mesurer" les différents paramètres de la pièce. Chaque paramètre obtient une valeur numérique. Ainsi, on peut faire la comparaison item par item avec le modèle et savoir quels paramètres sont respectés. Les variations sur les paramètres sont facilement calculables et le lien avec les tolérances est direct.

Le principal désavantage est que cette méthode n'est pas exacte. Les mesures des paramètres sont faites sur les éléments qui modélisent les points réels. Cette modélisation n'est pas parfaite; elle n'est qu'une sorte de représentation statistique. Donc, la réponse n'est pas sûre à 100%. Il est toujours possible qu'un ou des points réels excèdent les contraintes.

En raison du problème de détermination des points de séparation, la mise en oeuvre de la méthode est complexe. On doit faire appel à des méthodes itératives si on veut avoir des résultats précis. Ainsi, plus on veut être précis et plus la méthode demandera des temps de calcul considérables.

3.4.3.4 PRÉCISION DES MESURES POUR LA DEUXIÈME MÉTHODE

L'utilisation de caméras CCD produisant des images de type matrices de pixels en deux dimensions permet d'atteindre un certain niveau de précision. La discussion suivante porte sur la précision qu'il est possible d'atteindre.

Plusieurs articles de la bibliographie analysent le fait qu'il est possible d'atteindre une résolution meilleure que le pixel appelée résolution "subpixel". On peut définir simplement la résolution subpixel comme étant toute technique qui résulte en mesure ayant une résolution plus petite qu'un pixel.

Le trait commun de toutes ces techniques est qu'elles font toutes appel à la modélisation d'image. Il y a toujours un modèle conceptuel derrière toutes ces techniques.

Sommairement, les considérations générales qui devraient être faites pour établir ou évaluer une technique de mesure ayant une résolution subpixel sont les suivantes:

- Modélisation d'image
- Technique de subpixellation valide mathématiquement
- Algorithme d'application valide (ou approximation acceptable)

- Accord entre le modèle utilisé et l'échantillon du monde réel

Un exemple simple permet de comprendre le problème: Considérons un objet foncé posé sur un fond clair. Comme nous l'avons vu précédemment, une caméra CCD a une performance en réponse finie. Un exemple de résultats semblables à ce qu'on peut obtenir expérimentalement est donné au tableau 3.3.

Coordonnée x du pixel	Niveau de gris
100	2
101	2
102	2
103	3
104	7
105	9
106	10
107	10
108	10
109	10

Tableau 3.3: Exemple de mesures de niveau de gris en fonction de x

Les données sont pour une ligne de pixels avec x variant de 100 à 109. Le niveau de gris a une valeur comprise entre 0 et 15. La question est de savoir à quel endroit se situe le rebord. Il y a de nombreuses façons de calculer la localisation du rebord dont:

- Calculer la valeur de x à mi-chemin entre les valeurs minimum et maximum de niveau de gris obtenu.
- Calculer la valeur de x où le taux de changement est le plus élevé en utilisant un calcul simple de première différence.
- Faire passer un polynôme par les points en utilisant la méthode des moindres carrés. x serait le point où la dérivée est maximum (pente maximum).

Toutes ces méthodes peuvent donner un résultat où x n'est pas un entier. Ce sont donc toutes des méthodes pouvant donner une mesure ayant une précision subpixel. Il n'y a pas de réponse absolue à savoir quelle méthode est la meilleure ou la plus correcte. Chacune devrait être testée avec des objets de grandeur connue et sa performance évaluée.

Physiquement, la technique de subpixellation est basée sur le théorème d'échantillonnage de Nyquist que l'on a vu précédemment. Comme l'échantillonnage est fait avec une longueur de mot finie, alors théoriquement,

le signal ne peut être totalement reconstruit. En fait, on peut montrer que seulement les statistiques du signal peuvent être retrouvées. Pour reconstruire complètement le signal, il faudrait une longueur de mot infinie. La démonstration mathématique de ce principe ne sera pas faite ici. Notons simplement que plus on utilise de bits pour digitaliser le niveau de gris d'un pixel, plus le processus sera précis.

Dans la littérature, les valeurs expérimentales obtenues varient entre 0.1 et 0.02 pixels. Comme il a été dit au début de cette discussion (3.4.3.4, 3e paragraphe), dans tous les cas, cette précision est obtenue par l'utilisation d'un modèle d'image. Notons qu'il est possible d'obtenir autant de décimales que l'on veut après le point, mais que ces chiffres ne sont pas nécessairement significatifs.

On peut résumer la discussion par les deux règles suivantes:

- Une méthode de subpixelation sera précise si le modèle est une représentation valide de la situation réelle.
- La précision d'une mesure calculée sera approximativement proportionnelle à la racine carrée du nombre de mesures échantillon utilisées.

Plus le modèle choisi correspondra à la réalité et plus les mesures seront précises. Moins le modèle correspondra à la réalité et moins les mesures seront précises. Plus on aura de mesures échantillons (pixels) sur un objet, meilleure sera la mesure calculée. Donc, plus un objet sera petit sur une image, moins on pourra être précis dans sa mesure.

La fonction suivante sera utilisée lors de l'expérimentation:

$$\frac{1}{\sqrt{n}}$$

(3.5)

où n est le nombre de pixels utilisés. Cette fonction est très imparfaite car selon Mitchell [5.28], la précision est fonction de l'orientation angulaire, de la translation et de la différence de niveau de gris entre les deux surfaces.

3.4.4 DISCUSSION SUR LES MÉTHODES

Les deux méthodes proposées ne sont pas concurrentes mais plutôt complémentaires car le type de réponse obtenu n'est pas le même pour chacune d'elle:

La méthode de conditions de matériel est exacte mais ne donne qu'une réponse bi-modale; on sait si oui ou non l'objet respecte le modèle.

La méthode de modélisation des données réelles est seulement approximative car elle fait appel à des concepts humains de mesure qui sont utilisables pour le modèle mais ne s'appliquent pas d'une façon exacte au réel. Cependant, cette méthode donne une réponse qui permet de juger un objet selon des critères internes plus détaillés. Cette dernière méthode applique le concept de géométrie variationnelle et permet de l'utiliser pour l'inspection dimensionnelle.

La méthode de modélisation des données réelles est celle retenue pour l'expérimentation. C'est cette méthode qui est développée au chapitre IV.

CHAPITRE IV

RÉALISATION

4.1 INTRODUCTION

Un système a été élaboré afin de vérifier l'application de la méthode présentée aux chapitres deux et trois. Le système a nécessité l'utilisation de matériel appliqué à la vision par ordinateur. Un ensemble logiciel a été développé pour traiter les images et appliquer les concepts de géométrie variationnelle pour la vérification dimensionnelle. Des exemples d'objets ont été utilisés afin de vérifier le système développé. Les résultats obtenus pour différents objets sont présentés. Chaque objet a une topologie et des dimensions particulières que le système développé retrouve et compare à un modèle.

4.2 RÉALISATION MATÉRIELLE

La description de la réalisation matérielle est divisée en trois parties: une description du matériel utilisé, le montage réalisé pour la digitalisation des images et la procédure suivie pour cette digitalisation.

Le matériel utilisé pour l'acquisition et le traitement des images consiste en:

- Une caméra de type CCD.
- Du matériel servant à contrôler les conditions d'acquisition des images.
- Un système informatique pour l'enregistrement des images.
- Un système informatique pour le traitement des images.

Les objets examinés sont des formes planes découpées dans un matériel sombre. Le fond sur lequel étaient déposées ces formes est une surface plane et claire. Deux objets utilisés comme exemples sont présentés à l'annexe C.

La caméra utilisée pour "percevoir" les objets est le modèle KP-131 de Hitachi. Le contrôle de gain automatique est à "OFF" car l'éclairage est suffisant et le bruit doit être réduit le plus possible. Le système de balayage de l'écran est en position entrelacée ("interlace") 2:1. La résolution de la caméra est de 450 lignes TV dans le sens horizontal et de 350 lignes TV dans le sens vertical (au centre de l'écran). Ceci définit la résolution maximale du système. Une partie importante de la caméra est la lentille. Celle utilisée est une lentille TV 16mm 1:1.6 infini à .25m. Il est important d'avoir une longueur focale assez grande pour avoir le minimum de distorsion de l'image.

La caméra est soutenue par un support vertical au-dessus d'une surface plane. Un système d'éclairage, constitué de deux lampes orientables ainsi que d'un environnement d'où on a éliminé toute lumière parasite, permet de contrôler l'illumination de la scène. La scène est ici définie par l'objet sombre placé sur un fond clair.

Le circuit électronique qui sert à convertir le signal vidéo analogique en un signal digital utilisable par ordinateur est contenu dans deux cartes graphiques. Un bref résumé du fonctionnement est présenté.

Les deux cartes graphiques sont reliées au Q-BUS d'un Micro-PDP 11/23. Ce "bus" sert à échanger des données ou instructions entre les cartes et l'ordinateur. La carte Matrox QRGB-GRAPH est une carte graphique de base comprenant une mémoire d'écran de 512 X 512 X 4 bits. Cette mémoire d'écran permet d'enregistrer une image de 512 X 512 pixels avec quatre bits par pixel pour l'intensité, soit $2^4 = 16$ intensités possible par pixel. On peut brancher un moniteur RGB 10 MHz à cette carte pour visualiser la mémoire d'écran. La carte QVAF-512 est un processeur multi-fonction qui permet d'étendre les possibilités de la carte QRGB-GRAPH. Elle permet entre autre de convertir un signal analogique provenant d'une caméra vidéo en un signal digital et de l'enregistrer dans la mémoire d'écran de la carte QRGB-GRAPH. Les deux cartes graphiques sont programmables, c'est-à-dire que leurs différentes

fonctions sont contrôlables par programmation du Micro-PDP 11/23. La description complète des différents registres des cartes graphiques est donnée dans les manuels d'utilisation.

Différents programmes ont été utilisés sur l'ordinateur Micro-PDP 11/23 afin d'obtenir les images digitalisées sous un format matriciel en mémoire de l'ordinateur. Une liste des programmes utilisés est donnée à l'annexe B. Les programmes utilisés ne font que l'acquisition des images digitalisées avec la caméra et l'enregistrement dans des fichiers de données. Le traitement des images n'est pas réalisé sur le Micro-PDP 11/23 dû à ses possibilités limitées au plan de la mémoire vive (RAM), de la capacité de disque, de sa vitesse limitée et du manque de souplesse pour la programmation de ce type d'ordinateur.

Le traitement a été réalisé sur un ordinateur de type Micro-VAX II et tous les programmes ont été écrits en FORTRAN 77 standard. Un terminal graphique couleur de type VT340 (résolution 500 X 800) a été utilisé pour la visualisation des images et des résultats.

La figure 4.1 décrit le montage réalisé pour l'acquisition des images. La caméra est positionnée au-dessus d'une surface plane. Cette surface est claire et l'objet foncé est déposé dessus. La distance caméra-objet et la longueur

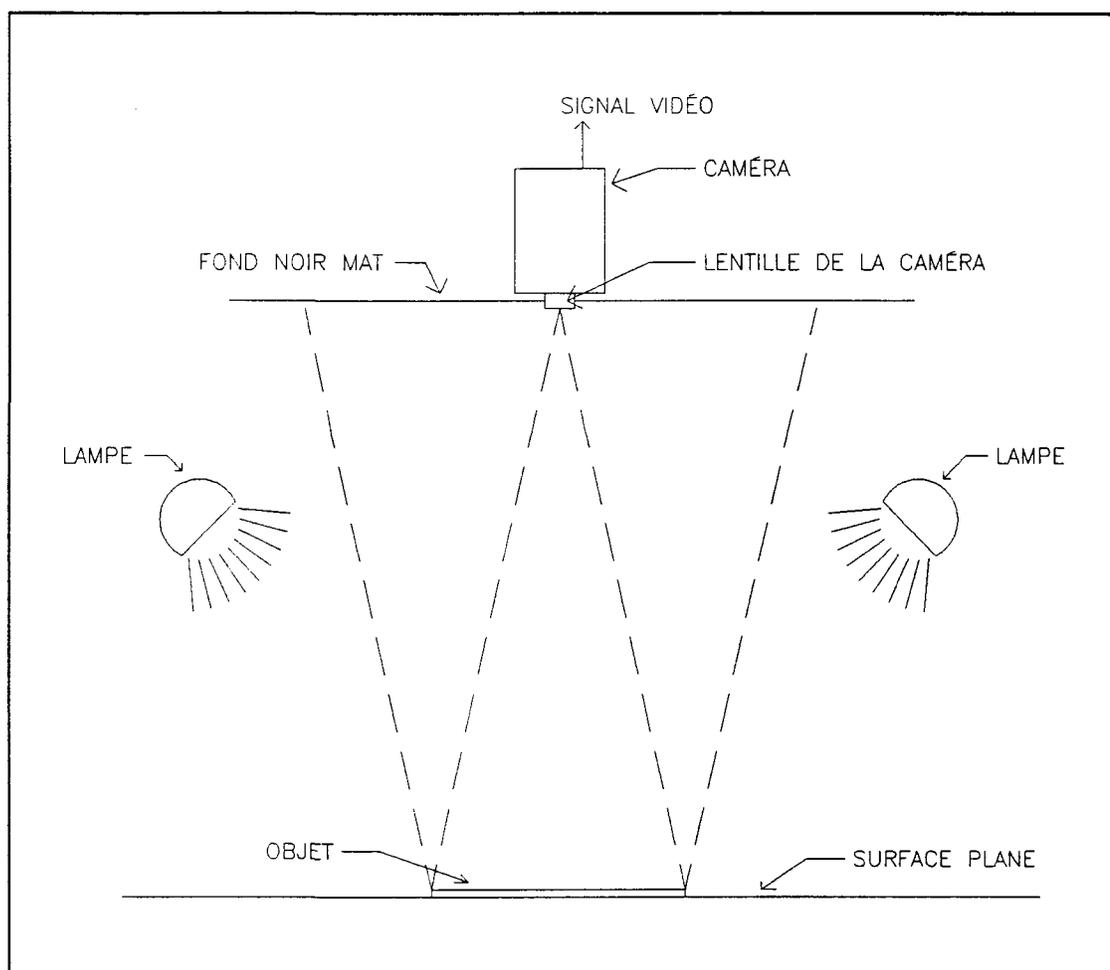


Figure 4.1: Montage pour l'acquisition des images

focale de l'objectif sont déterminées de façon à ce que l'objet soit inclus complètement dans le champ de vision de la caméra. Deux lampes inclinées à 45 degrés de chaque côté de la scène permettent un éclairage uniforme. Un fond noir mat autour de la caméra permet d'éviter les réflexions parasites directes de l'objet ou du fond.

La procédure expérimentale suivie peut être divisée en trois étapes:

- Définition et enregistrement du modèle de l'objet.

- Acquisition de l'image de l'objet.
- Traitement logiciel pour l'obtention des résultats.

Les première et troisième étapes seront discutées lors de la réalisation
logicielle. L'acquisition de l'image de l'objet consiste à:

- Calibrer la caméra.
- Poser un objet sur le fond.
- Faire l'acquisition de l'image de l'objet avec les programmes du
Micro-PDP 11/23.
- Transférer l'image sur le Micro-VAX II pour le traitement
subséquent.

4.3 RÉALISATION LOGICIELLE

4.3.1 INTRODUCTION

Un ensemble logiciel qui applique les concepts de la géométrie variationnelle à la vérification dimensionnelle a été développé. La méthode suivie est celle présentée à la figure 3.8 (méthode de modélisation des données réelles). Une base de données, permettant d'exprimer un modèle variationnel, a été construite. La structure de cette base de données sera présentée mais l'étape de la modélisation ne sera pas abordée. Différents programmes et sous-programmes implémentent les différentes étapes de la méthode. Comme l'étape de la digitalisation a été présentée au point 4.2, la discussion portera sur les cinq autres étapes. Pour chaque étape, les programmes réalisés sont analysés.

Tous les programmes ont été écrits par l'auteur sauf deux exceptions: le sous-programme FIT2 qui est une adaptation du livre NUMERICAL RECIPIES IN FORTRAN (page 508) et le sous-programme RESOL qui a été développé à l'origine par Pierre McComber.

4.3.2 STRUCTURE DE LA BASE DE DONNÉES DU MODÈLE

La base de données du modèle doit contenir toute l'information sur celui-ci. Cette information est exprimée d'une façon telle qu'elle pourrait être générée par un logiciel de CAO (modélisation) utilisant les principes de la géométrie variationnelle.

La base de données du modèle doit donc contenir l'information sur la topologie, les contraintes et les tolérances. Par commodité, la base de données est divisée en quatre fichiers. La description du contenu de chaque fichier est donnée ci-dessous. Le format FORTRAN est donné entre parenthèses.

- 1 - Fichier de la topologie ".SHP"
 - Nombre total de noeuds (I5)
 - Nombre de surfaces: 1 car on est en 2D (I5)
 - Nombre d'éléments pour la surface (I5)
 - Type de chaque élément (I5)
 - Numéros des noeuds formant chaque élément (I5)

- 2 - Fichier des contraintes ".SCH"**
 - Nombre de contraintes (I5)
 - Type de chaque contrainte (I5)
 - Numéros des degrés de liberté pour chaque contrainte (I5)

- 3 - Fichier des paramètres ".PAR"**
 - Nombre de paramètres (I5)
 - Valeur de chacun des paramètres (F10.4)

- 4 - Fichier des tolérances ".TOL"**
 - Nombre de tolérances (I5)
 - Valeur maximum et valeur minimum pour chaque tolérance (2F10.4)

La division en quatre fichiers séparés est pratique car: pour chaque fichier de topologie, on peut avoir plusieurs fichiers de contraintes; pour chaque fichier de contraintes, on peut avoir plusieurs fichiers de paramètres; pour chaque fichier de paramètres, on peut avoir plusieurs fichiers de tolérances. Les liens entre les fichiers sont les suivants:

- Le nombre de noeuds est égal à la moitié du nombre de degrés de liberté contraints. On est en 2D et chaque noeud a deux coordonnées ou degrés de liberté d'associés.

- Le nombre de contraintes est égal au nombre de paramètres. Un paramètre est associé à chaque contrainte.
- Le nombre de tolérances est égal au nombre de paramètres. Pour chaque tolérance, on a deux valeurs: une maximum et une minimum.

Les types d'éléments sont présentés à la figure 2.3 et les types de contraintes le sont au tableau 4.1.

Type #	Description	Exemple	Sous-programme (Paramètre)	Nombre de DL contraints
1	Noeud fixe en x	C_1 ou C_3	FIXE(XF)	1
2	Noeud fixe en y	C_2 ou C_3	FIXE(YF)	1
3	Distance horizontale (selon x)	C_4	DIST(DH)	2
4	Distance verticale (selon y)	C_5	DIST(DV)	2
5	Distance linéaire (entre deux noeuds)	C_6	DISTL(DLIN)	4
6	Angle entre 2 éléments de type 1	C_7	ANGLE(A)	8
7	Distance noeud- élément de type 1 (perpendiculaire)	C_8	DISTPE(DP)	6

Tableau 4.1: Types de contraintes

4.3.3 ANALYSE DES PROGRAMMES

Chaque programme est analysé. Les programmes sont classés en programmes principaux et en programmes utilitaires. Pour chacun des programmes (et sous-programmes) principaux, on a :

- Description détaillée des fonctions
- Organigramme
- Difficultés principales rencontrées
- Limites et améliorations possibles
- Liste du programme à l'annexe E

Pour chacun des programmes et des sous-programmes utilitaires, on a une description sommaire des fonctions.

4.3.3.1 HISTO

Ce programme réalise la fonction "Analyse primaire" de la figure 3.8. Il utilise la méthode de l'histogramme pour extraire le contour de l'objet présent dans l'image digitalisée. L'organigramme du programme HISTO est présenté à la figure 4.2. Chacun des sous-programmes est décrit ci-après.

Comme les images traitées sont d'une très bonne qualité, HISTO répond

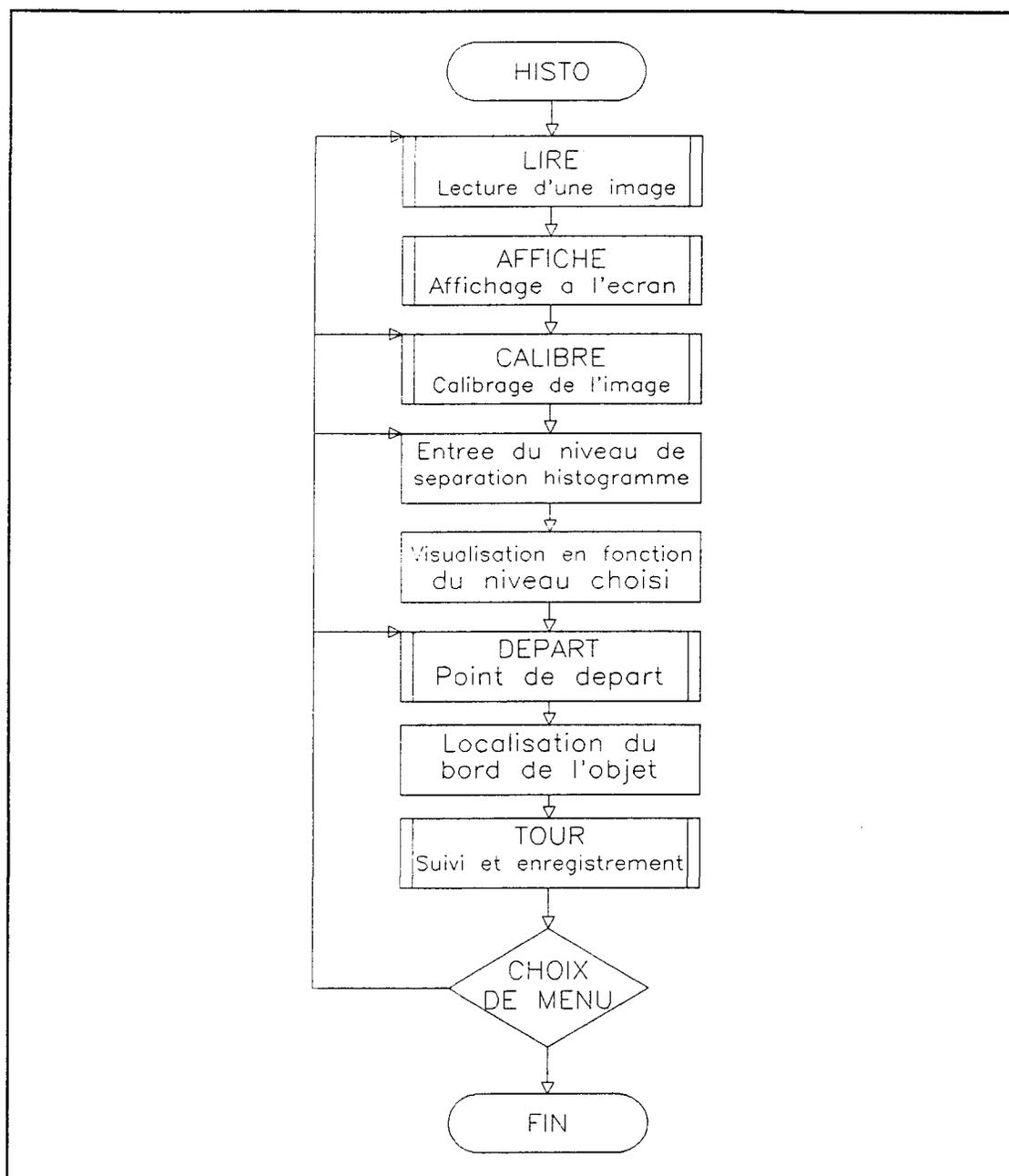


Figure 4.2: Organigramme du programme HISTO

très bien aux besoins. Les améliorations possibles viseraient surtout à pouvoir traiter des images d'une qualité moindre, ou encore, à l'automatisation et à l'amélioration de la rapidité de traitement. La liste du programme HISTO et des sous-programmes LIRE, AFFICHE, CALIBRE, DEPART et TOUR est

donnée à l'annexe D.

LIRE

Ce sous-programme fait la lecture d'une image digitalisée sous forme pixel et l'enregistre dans un tableau bi-dimensionnel.

AFFICHE

Ce sous-programme permet d'afficher une image à l'écran d'un terminal graphique.

GKS_*

Librairie de sous-programmes réalisant diverses fonctions graphiques selon le standard GKS ("Graphics Kernel System").

CALIBRE

Ce sous-programme sert au calibrage d'une image. A chaque image est associée une image de référence. La référence permet de déterminer l'échelle de l'image ainsi que le modèle de distorsion. Un modèle de transformation des

coordonnées est passé à **TOUR** pour que ce dernier enregistre les coordonnées réelles des points du contour.

DEPART

Sous-programme qui détermine un point de départ pour le contour de la surface. Ce point de départ servira au sous-programme **TOUR**.

TOUR

Sous-programme de suivi de contour de l'objet et enregistrant le contour détecté dans un fichier de données. Il pourrait être amélioré pour traiter des images où plusieurs objets sont présents.

4.3.3.2 POLY

Ce programme réalise la fonction "Approximation du contour par la topologie" de la figure 3.8. A partir du fichier de contour de l'objet, il extrait la topologie et la valeur de chacun des degrés de liberté. La topologie des éléments est retrouvée par méthode statistique de lissage de courbes. Les valeurs des degrés de liberté sont calculées par l'intersection des équations d'approximation des éléments. L'organigramme du programme **POLY** est

présenté à la figure 4.3. Chacun des sous-programmes est décrit ci-après.

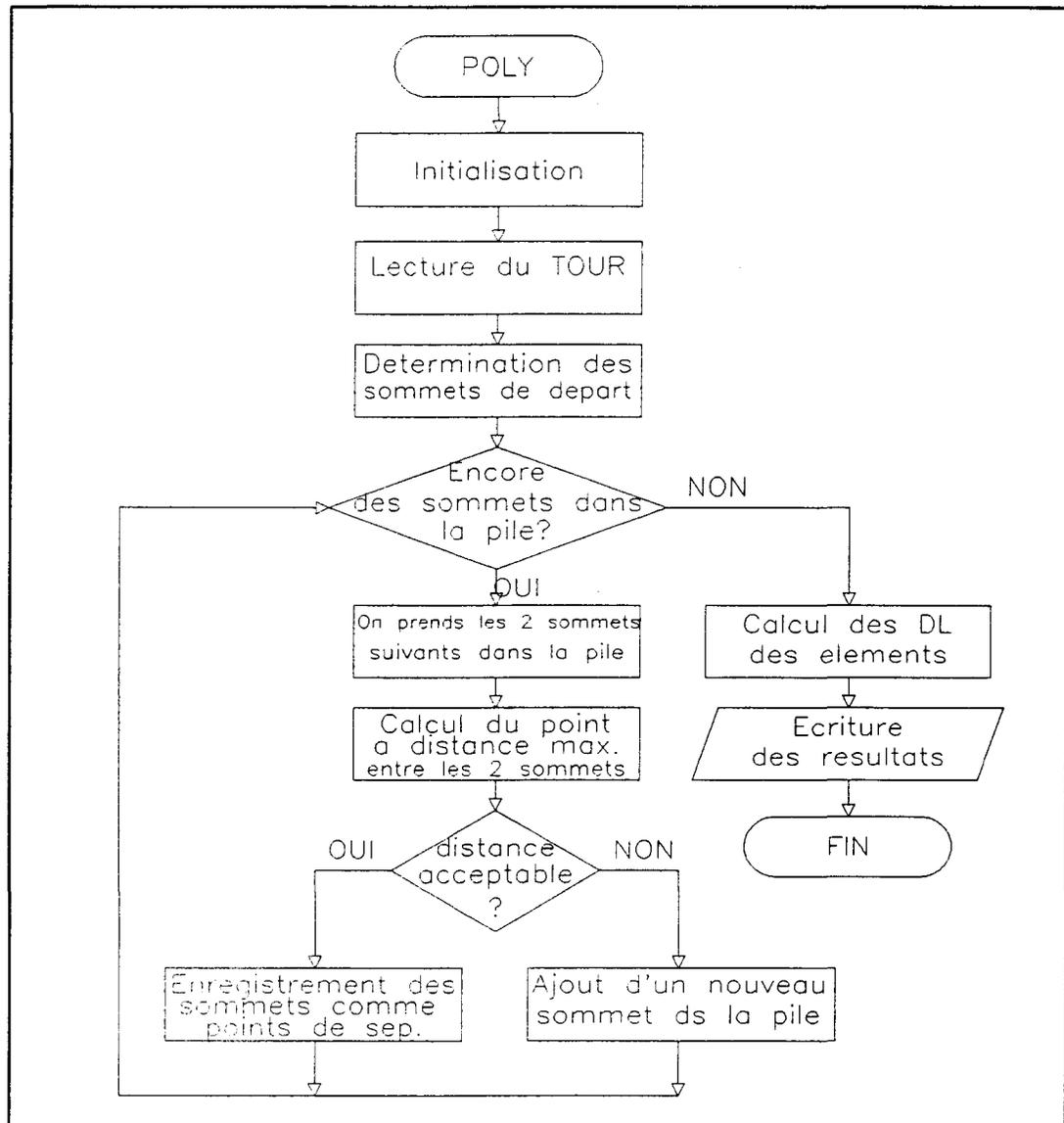


Figure 4.3: Organigramme du programme POLY

Le programme POLY ne fonctionne que pour des objets de type "polygone à n côtés". Le seul type d'élément qu'il reconnaît est le type 1: linéaire deux noeuds. Le programme ne fait pas appel au modèle topologique. L'algorithme utilisé est une méthode de division "splitting" semblable à l'algorithme 8.1 de Ballard & Brown [2.1]. L'ajout de la reconnaissance de

nouveaux types d'éléments nécessiterait de faire appel au modèle. La liste du programme POLY et des sous-programmes FIT2, DIST et INTER est donnée à l'annexe D.

FIT2

Ce sous-programme trouve l'équation de la droite passant par n points en utilisant la méthode des moindres carrés.

DIST

Ce sous-programme calcule la distance entre un point défini par ses coordonnées (x,y) et une droite définie par sa pente et une constante.

INTER

Ce sous-programme calcule l'intersection entre deux droites définies chacune par sa pente et une constante.

4.3.3.3 GEO

Ce programme est divisé en cinq sous-programmes de lecture des données et en six autres qui réalisent les fonctions "Calcul des paramètres", "Calcul de la différence" et "Vérification des tolérances" de la figure 3.8.

L'organigramme du programme GEO est présenté à la figure 4.4.

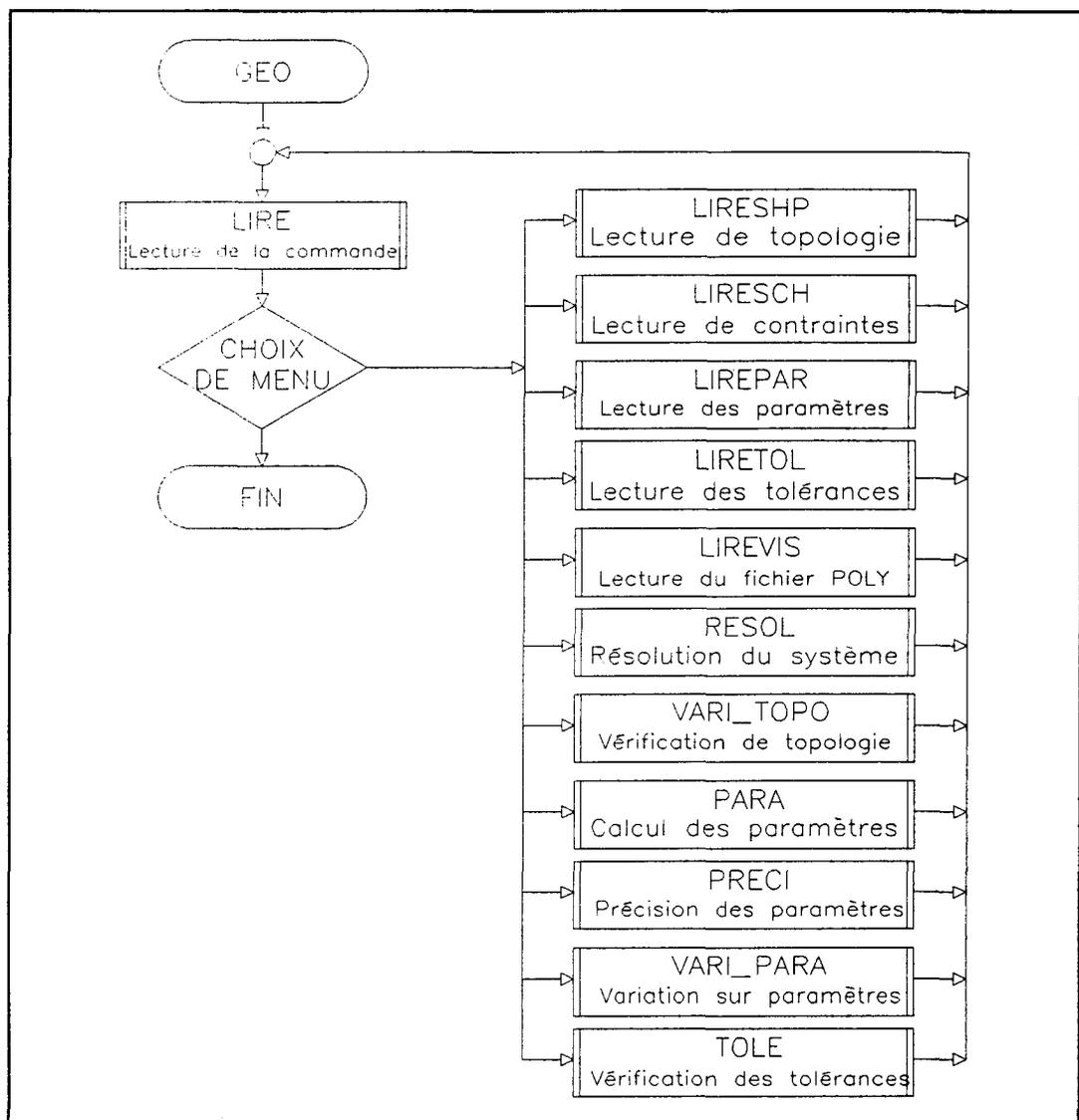


Figure 4.4: Organigramme du programme GEO

La description détaillée de chacune des fonctions (sous-programmes) est donnée ci-après.

Le programme GEO n'est en fait qu'un menu regroupant différentes fonctions. Chaque fonction est réalisée par un sous-programme distinct. Cette structure a été choisie en raison de sa souplesse. Selon les données du modèle disponible et de la vision, il est possible d'effectuer certaines fonctions. La figure 4.5 représente les préalables pour les différentes fonctions. La signification des préalables est donnée lors de la description individuelle des fonctions.

LIRESHIP, LIRESCH, LIREPAR, LIRETOL et LIREVIS

Sous-programmes de lecture des données respectivement de: topologie du modèle, contraintes du modèle, paramètres des contraintes du modèle, tolérances sur les paramètres du modèle et finalement, données obtenues de la vision (degrés de liberté et topologie). La signification des préalables de ces sous-programmes est donnée en 4.3.2 (description de la base de données du modèle).

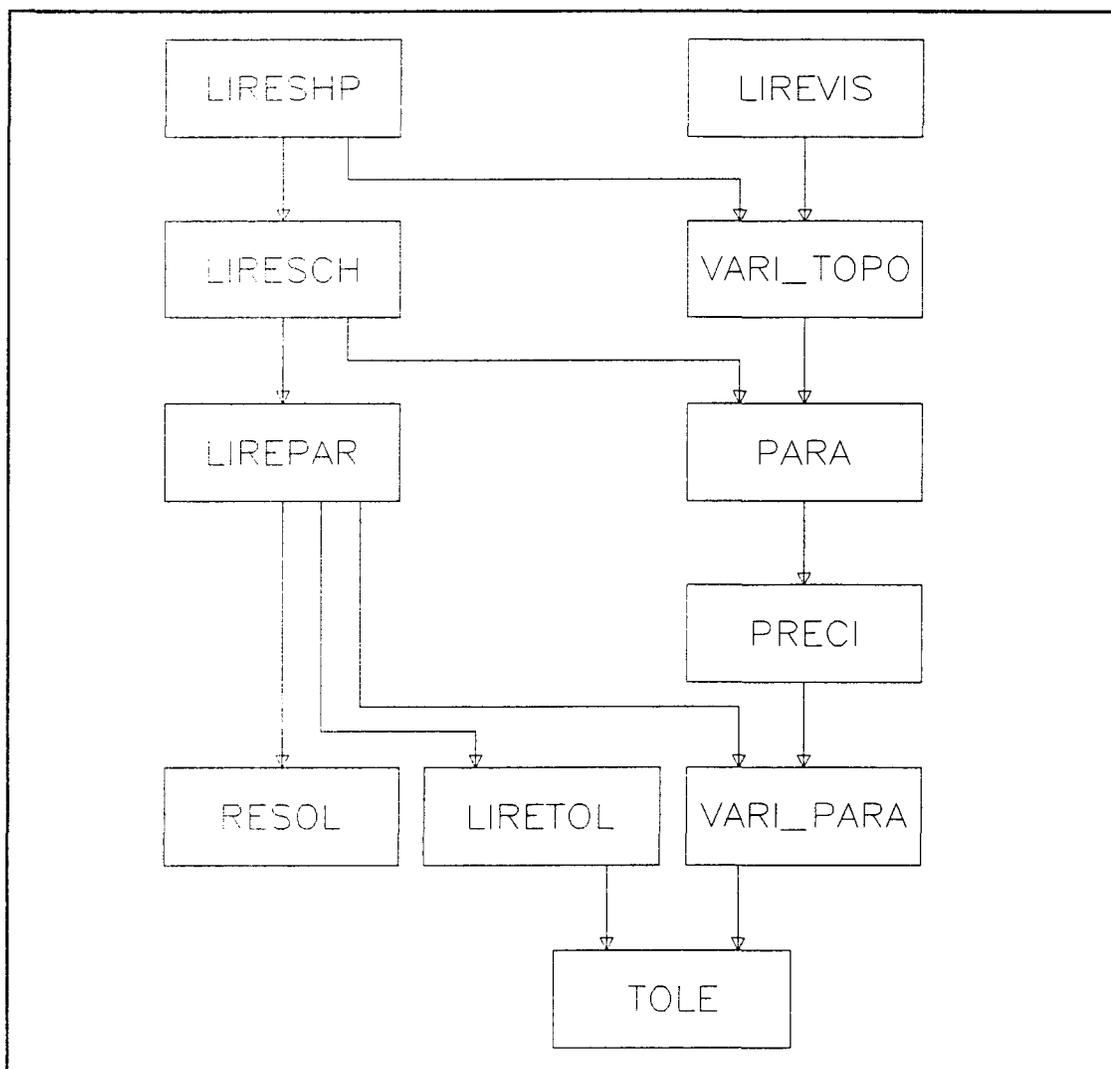


Figure 4.5: Arbre des préalables des fonctions de GEO

RESOL

Ce sous-programme résout un système d'équations non symétriques par la méthode de Gauss. Il permet de trouver la valeur des degrés de liberté à partir des équations de contraintes et des paramètres. Le préalable est d'avoir

un modèle complet. Ce sous-programme n'est inclus qu'à titre utilitaire; normalement, il ferait partie du logiciel de CAO (modélisation) paramétrique.

VARI_TOPO

Ce sous-programme fait la correspondance entre la topologie obtenue par la vision et celle du modèle. La correspondance permet de détecter si des variations existent entre les deux topologies. Les préalables sont d'avoir évidemment la topologie du modèle et celle obtenue par la vision.

PARA

Ce sous-programme réalise la fonction "Calcul des paramètres" de la figure 3.8. Le premier préalable est d'avoir la même topologie entre le modèle et la vision. Les degrés de liberté obtenus par la vision seront insérés dans les équations de contrainte du modèle pour calculer les paramètres, ce qui demande comme deuxième préalable d'avoir lu les contraintes du modèle.

PRECI

Ce sous-programme rapporte la précision sur les degrés de liberté obtenus par la vision, aux paramètres calculés par le sous-programme **PARA**.

VARI_PARA

Ce sous-programme réalise la fonction "Calcul de la différence" de la figure 3.8. On doit connaître les paramètres du modèle et ceux calculés par **PARA**.

TOLE

Ce sous-programme réalise la fonction "Vérification du respect des tolérances" de la figure 3.8. On doit avoir lu les tolérances du modèle (**LIRETOL**) et calculé les variations réelles sur les paramètres (**VARI_PARA**).

4.4 EXEMPLES

Quatre exemples sont présentés. Pour chaque exemple on a un modèle, des données de la vision (simulées pour les exemples 1 et 2, réelles pour les exemples 3 et 4) et des résultats de traitement.

Les quatre modèles utilisés comme exemples sont:

- 1 - Figure 4.6: Un élément de type 1
- 2 - Figure 4.8: Un rectangle
- 3 - Figure 4.10: Un modèle du rectangle présenté à l'annexe C
- 4 - Figure 4.12: Un modèle du polygone présenté à l'annexe C

La liste des fichiers de données pour les quatre modèles d'objets est présentée à l'annexe D. Pour chacun des objets, on a les fichiers de topologie, des contraintes, des paramètres et des tolérances.

Les données de la vision sont présentées pour les quatre exemples aux figures 4.7, 4.9, 4.11 et 4.13. Les fichiers pour les données simulées (exemples 1 et 2) sont données à l'annexe F (ET1.VIS et CARRE.VIS). Les fichiers pour les données réelles (exemples 3 et 4) sont données à l'annexe F (FOR047.RES et FOR033.RES).

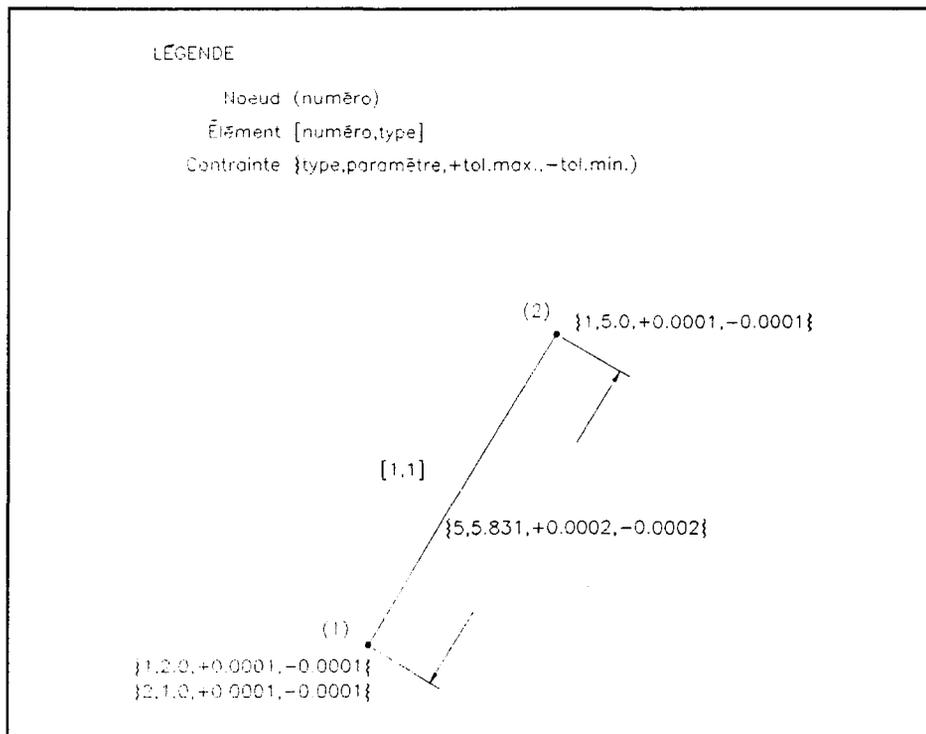


Figure 4.6: Modèle de l'exemple 1: Élément de type 1

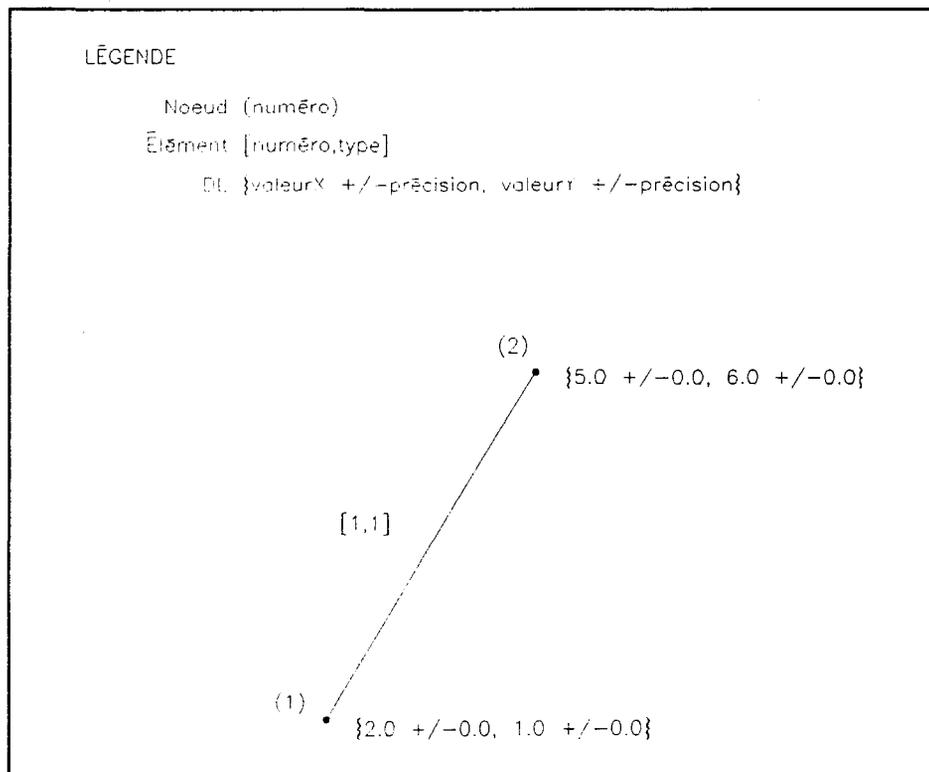


Figure 4.7: Données de la vision simulées pour l'exemple 1

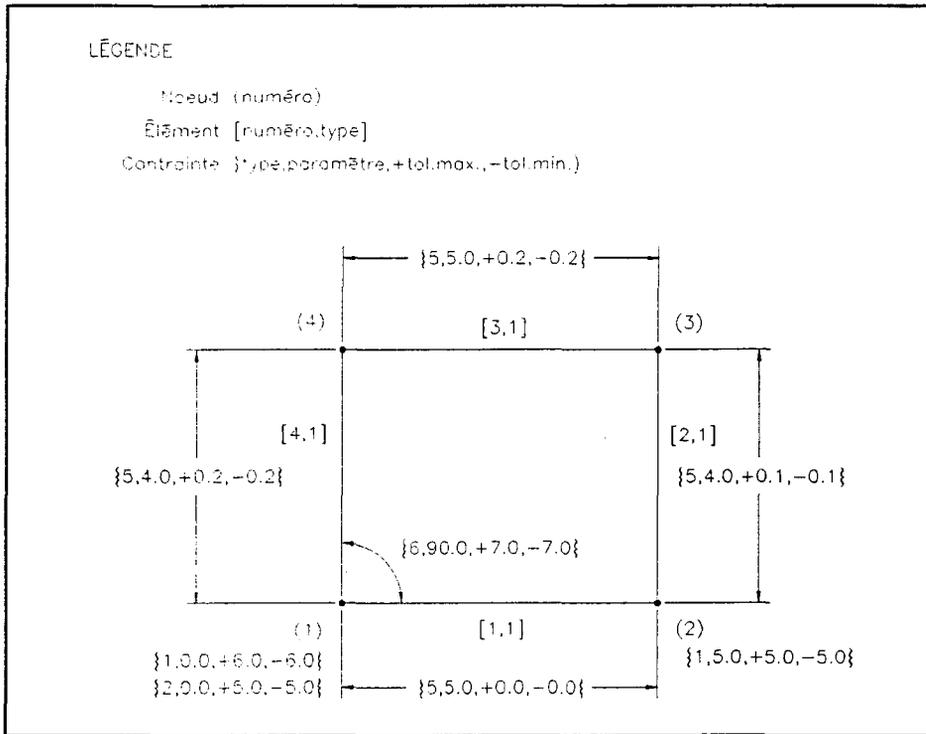


Figure 4.8: Modèle de l'exemple 2: Rectangle

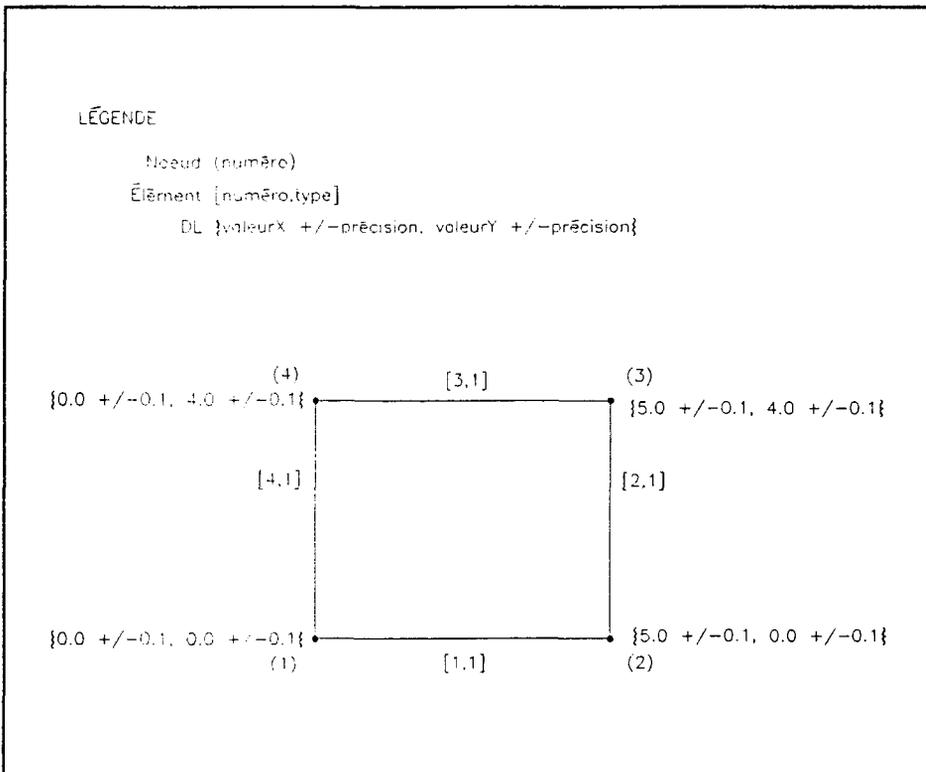


Figure 4.9: Données de la vision simulées pour l'exemple 2

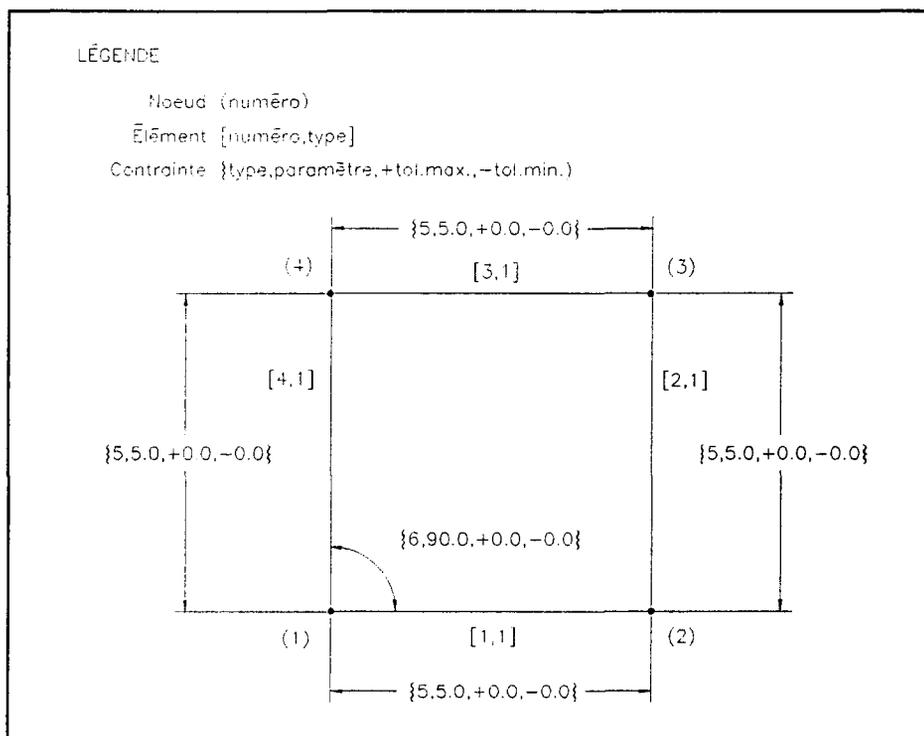


Figure 4.10: Modèle de l'exemple 3: Rectangle

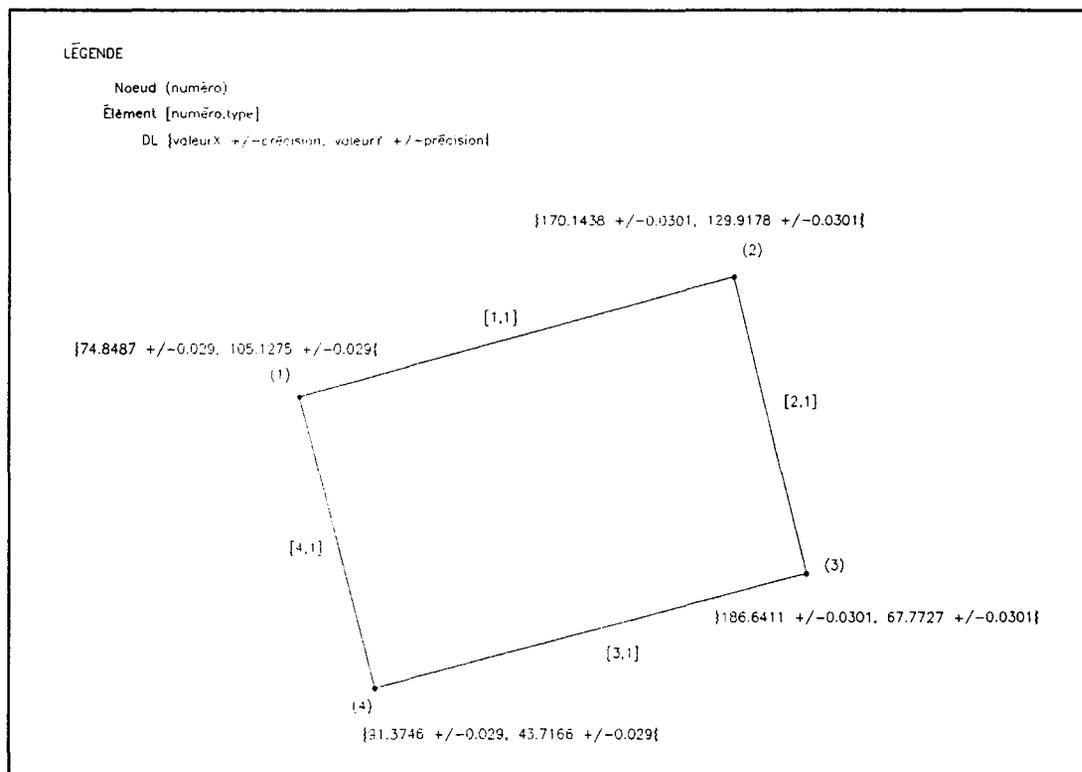


Figure 4.11: Données obtenues de la vision pour l'exemple 3

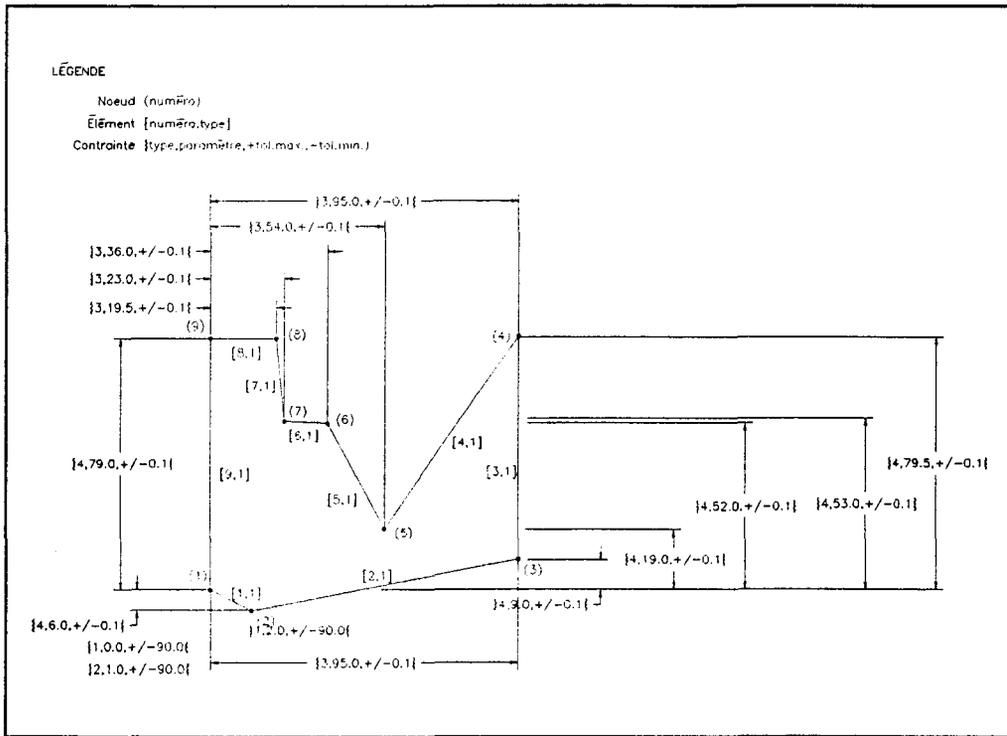


Figure 4.12: Modèle de l'exemple 4: Polygone à 9 côtés

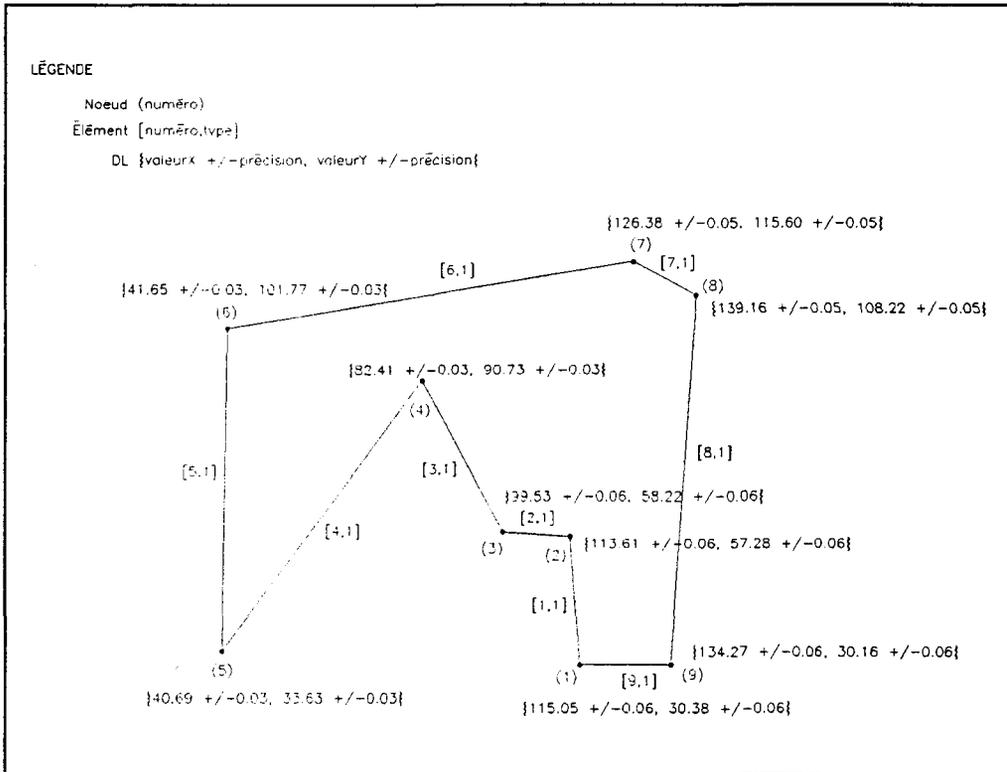


Figure 4.13: Données obtenues de la vision pour l'exemple 4

4.5 RÉSULTATS OBTENUS AVEC LES EXEMPLES

Les contours obtenus avec HISTO pour le rectangle et le polygone sont donnés partiellement à l'annexe F (FOR047.RES et FOR033.RES). Vu le grand nombre de points, il est impossible de donner le fichier au complet. Des calculs de propriétés mécaniques de surface ont été obtenus afin de vérifier les résultats du contour. Les résultats de ces calculs sont présentés dans les fichiers FOR094.RES et FOR093.RES à l'annexe F.

Les résultats obtenus avec POLY à partir des fichiers de contour générés par HISTO sont donnés pour le rectangle et le polygone dans les fichiers FOR072.DAT et FOR063.DAT à l'annexe F.

Des résultats obtenus avec GEO sont donnés à l'annexe F pour:

- L'élément de type 1 avec des données de la vision simulées (ET1.RES).
- Le modèle de rectangle avec des données de la vision simulées (CARRE.RES).
- Le rectangle avec les données de la vision calculées par POLY (FOR072.RES).
- Le polygone avec les données de la vision calculées par POLY (FOR063.RES).

Les résultats de traitement complets générés par le programme GEO sont présentés à l'annexe F. Le tableau 4.2 constitue un bref résumé.

Ex.#	Nombre de degrés de lib.	Nombre de correspondances entre les topologies	Nombre maximum de tolérances respectées	L'objet satisfait toutes les contraintes du modèle
1	4	1	4	OUI
2	8	4	4	NON
3	8	4	0	NON
4	18	9	3	NON

Tableau 4.2: Résumé des résultats pour les exemples

On retrouve au tableau 4.2 les éléments suivants dans chacune des colonnes:

- 1 - Le numéro de l'exemple
- 2 - Le nombre de degrés de liberté. Comme on est en deux dimensions, on a deux degrés de liberté par noeud. Pour l'exemple 2, on a 4 noeuds, alors $4 \times 2 = 8$ degrés de liberté.
- 3 - Le nombre de correspondances possibles entre la topologie du modèle et celle de la vision.
- 4 - Pour chacune des correspondances trouvées, on a vérifié le nombre de

tolérances respectées. Le nombre maximum est donné au tableau 4.2.

- 5 - Si le nombre de tolérances respectées correspond au nombre de degrés de liberté, alors l'objet satisfait toutes les contraintes du modèle.

Pour chacun des quatre modèles, les résultats sont calculés pour chacune des topologies correspondantes. Ceci est fait car un objet peut correspondre au modèle selon diverses orientations.

Les contraintes de translation et de rotation nulle peuvent être utilisées de deux façons: pour vérifier la position de l'objet (translation en x, en y et rotation) ou encore pour assurer une indépendance de la position (en x, en y et angulaire). La première façon a été utilisée pour l'obtention des résultats. La deuxième façon consisterait à faire un modèle de correction sur les données réelles afin de les ramener selon la position en x, en y et angulaire du modèle. Ceci permettrait de vérifier un objet quelle que soit sa position en x, en y ou angulaire par rapport au modèle.

La démarche complète du traitement est donnée dans chacun des fichiers. Les étapes sont:

- Lecture des fichiers de données (menus 1, 2, 3, 4 et 5)

- Comparaison des topologies (menu 7). Le détail de l'algorithme est donné un peu plus loin dans le texte.
- Calcul des paramètres réels: les valeurs des degrés de liberté obtenus par la vision sont mis dans le modèle des contraintes. Les résultats sont les paramètres réels et sont donnés pour chacune des topologies (menu 8).
- Calcul de la précision sur les paramètres réels (menu 9).
- Calcul de la variation sur les paramètres (menu 10).
- Vérification si la variation dépasse les tolérances permises (menu 11).

La vérification de la topologie de l'objet par rapport au modèle est faite comme suit:

- On vérifie d'abord si le nombre d'éléments correspond.
- Ensuite on fait la correspondance entre chacun des éléments du modèle et ceux de l'objet un à un. Le nombre de correspondance à vérifier est donc égal au nombre d'éléments.
- Pour chaque correspondance, on vérifie si chaque type d'élément du modèle correspond au type d'élément de l'objet.
- On conserve chacune des correspondances trouvées entre le modèle et l'objet pour analyse subséquente.

Ceci permet de vérifier complètement l'objet car on ne sait pas quel degré de liberté du modèle correspond à quel degré de liberté de l'objet. L'objet peut par conséquent être acceptable selon plusieurs correspondances.

4.6 DISCUSSION DES RÉSULTATS OBTENUS AVEC LES EXEMPLES

Les résultats obtenus permettent de saisir l'importance des tolérances du modèle et de la précision de la vision pour juger de l'acceptation ou non d'un objet.

Dans l'exemple 1, les données de la vision ont été choisies de telle sorte qu'elles correspondent exactement au modèle. Toutes les contraintes sont donc satisfaites (annexe F, fichier ET1.RES, page 2). Pour l'exemple 2, les données ont été choisies égales au modèle en ce qui concerne les valeurs mais la précision n'est pas assez grande pour assurer avec certitude que toutes les contraintes sont respectées (voir annexe F, fichier CARRE.RES, pages 3 et 4). Ceci illustre un résultat incertain dû à un manque de précision. Une meilleure précision ou encore des tolérances plus grandes auraient permis la satisfaction de toutes les contraintes. De plus, pour cet exemple, on constate que deux correspondances sont valables entre les topologies. Pour les deux autres exemples (3 et 4), les données de la vision étant trop différentes du modèle, les contraintes ne sont pas satisfaites (voir annexe F, fichiers FOR072.RES pages

3-4 et FOR063.RES pages 10 à 13). Ceci permet d'illustrer la fiabilité de la méthode développée.

Au niveau des temps de traitement, le traitement des images est de loin le plus exigeant. Le temps de traitement du programme faisant la correspondance entre le modèle et les données obtenues par la vision (GEO) est négligeable par rapport au traitement des images.

CHAPITRE V

CONCLUSIONS ET RECOMMANDATIONS

5.1 CONCLUSIONS

Un modèle variationnel a été élaboré. Sept types d'équations de contraintes permettent la modélisation d'objets en deux dimensions à l'aide du modèle. D'autres types de contraintes sont possibles en autant que celles-ci puissent se définir sous forme d'équations. Trois types d'éléments sont proposés et un a été utilisé. Le modèle inclut également l'information sur les tolérances pour chacune des contraintes. Ces éléments ont permis la modélisation d'objets simples. L'utilisation d'éléments plus complexes permettrait la modélisation d'objets plus complexes.

Une méthode de traitement d'image, permettant de retrouver le modèle de l'objet, a été adaptée. Celle-ci a été utilisée sur des images obtenues à l'aide d'une caméra CCD. Une fonction estimant la précision sur les résultats a été utilisée. Un ensemble logiciel, faisant le lien entre le modèle et les données obtenues par la vision, a été développé. Des résultats ont été obtenus pour différents exemples.

Le modèle variationnel est utilisable pour la vérification dimensionnelle à l'aide de la vision par ordinateur. Le processus de vision demande l'utilisation d'un modèle et le modèle variationnel a l'avantage d'être déjà utilisé pour la conception. Le même modèle peut donc être utilisé dans les deux cas.

Un des avantages d'obtenir les valeurs réelles des paramètres est de pouvoir les comparer directement au modèle. Le choix des paramètres est fait lors de la conception.

Le modèle variationnel s'adapte pour l'utilisation en vision par ordinateur dans la reconnaissance de formes. Cependant, le domaine de la vision est encore en développement et c'est ce qui limite les possibilités d'utilisation.

La théorie et l'expérimentation ont été faites en deux dimensions. Les bases théoriques de la méthode permettent l'extension en trois dimensions. Cette extension nécessiterait le développement d'éléments permettant la modélisation tridimensionnelle et de contraintes applicables à ces éléments. Les résultats de la vision devraient être alors exprimés dans un système de référence tridimensionnel.

5.2 RECOMMANDATIONS

Le développement d'une meilleure fonction de précision sur les données obtenues par la vision est nécessaire pour une utilisation réelle. Étant donné la technologie utilisée (CCD), les paramètres de la fonction de précision pourraient être déterminés par calibrage à partir d'objets dont les dimensions sont connues.

Le processus de vision utilisé ne fait pas appel au modèle particulier à identifier. Ceci signifie que l'image est analysée en ayant, à priori, aucune information sur l'objet à identifier: nombre et types d'éléments qui le définissent. Le processus se limite à l'utilisation d'un modèle général. Il serait intéressant d'utiliser le modèle particulier directement dans le processus de vision.

Les méthodes utilisées ont été volontairement simplifiées. Le développement devrait se poursuivre notamment pour:

- Le développement d'autres types de contraintes.
- La reconnaissance de type d'éléments plus complexes dans le processus de vision.
- L'extension des méthodes en trois dimensions.

L'utilisation dans un environnement réel, en milieu industriel par exemple, permettrait de juger si les méthodes sont avantageuses par rapport à d'autres.

BIBLIOGRAPHIE

1. Bibliographies

- 1.1 Azriel Rosenfeld, Image Analysis and Computer Vision: 1987, Computer Vision, Graphics, and Image Processing **42**, 234-293 (1988).
- 1.2 Roland T. Chin, Automated Visual Inspection techniques and Applications: A Bibliography, Pattern Recognition Vol. 15, No. 4, pp 343-357, (1982).
- 1.3 Roland T. Chin, Automated Visual Inspection: 1981 to 1987, Computer Vision, Graphics, and Image Processing **41**, 346-381 (1988).

2. Livres

- 2.1 Dana H. Ballard and Christopher M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- 2.2 King Sun Fu, Robotics: Control, Sensing, Vision, and Intelligence., McGraw-Hill series in CAD-CAM robotics and computer vision, U.S.A., (1987).
- 2.3 Gonzalez and Wintz, Digital Image Processing, Addison-Wesley publishing company, Advanced Book Program/World Science Division Reading, Massachussets.
- 2.4 William K. Pratt, Digital Image Processing, John Wiley & Sons Inc. (a Wiley-Interscience Publication), 1978.
- 2.5 Michael Brady (Editor), Computer Vision, AI Laboratory, MIT, Cambridge, MA, USA, North-Holland publishing company, Amsterdam, Reprinted from the journal Artificial Intelligence, Vol. 17, August 1981.
- 2.6 B. G. Batchelor, D. A. Hill and D. C. Hodgson, Automated Visual Inspection, IFS (Publications) Ltd, UK, North-Holland, 1985.
- 2.7 Gouri Dhatt et Gilbert Touzot, Une présentation de la méthode des éléments finis, Maloine S.A. Editeur Paris et Les Presses de l'Université Laval, Québec, 1981.

3. Thèses

- 3.1 V. C. Lin, Three Dimensional Variational Geometry in Computer-Aided Design, M.S. Thesis, M.I.T., May 1981.

BIBLIOGRAPHIE (suite)

4. Rapports de recherche

- 4.1 Pierre McComber, La Géométrie Variationnelle: Un exemple à deux dimensions et perspective d'utilisation dans la conception assistée par ordinateur, Rapport scientifique, Département de sciences appliquées, Université du Québec à Chicoutimi, Mars 1985.
- 4.2 Réjean Fournier, La Géométrie Variationnelle: Synthèse de la Théorie, Équipe de recherche en inspection dimensionnelle automatisée de pièces mécaniques, Département de mécanique, École de Technologie Supérieure, Février 1989.

5. Articles

- 5.1 John F. Wagner, Sensors for Dimensional Measurement, SME VISION '87 Conf. Detroit, MI, June 1987.
- 5.2 Christian Fortunel, Building a Vision Laboratory, SME VISION '87 Conf. Detroit, MI, June 1987.
- 5.3 Steve Mersch, Overview of Machine Vision Lighting Techniques, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.4 Norman Wittels, Lighting Design for Industrial Machine Vision, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.5 Amir Novini, Fundamentals of Machine Vision Lighting, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.6 Gordon T. Uber, Illumination Methods for Machine Vision, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.7 Alain Durand, La pratique de la Macro-Photo, Publications Photo-Cinéma Paul Montel, 189 rue St-Jacques, Paris 5e, (1981).
- 5.8 Philip C. Flora, International Computer Vision Directory, Technical DataBase corp., (1985).
- 5.9 Ed Winfield, Image Processing Products for the Q-Bus Meet Industry Need for Precision Inspection, Hardcopy, Vol. 6, No. 3, July 1986.

BIBLIOGRAPHIE (suite)

- 5.10 Berthold K. P. Horn, Understanding Image Intensities, *Artificial Intelligence* **8**, 201-203, (1977).
- 5.11 J. L. Mundy, *Automatic Visual Inspection*, 1978.
- 5.12 Dragutin Petkovic and Eric B. Hinkle, A Rule-Based System for Verifying Engineering Specifications in Industrial Visual Inspection Applications, *IEEE transactions on pattern analysis and machine intelligence*, Vol. PAMI-9, No. 2, 306-311, March 1987.
- 5.13 Larry S. Davis, Shape Matching Using Relaxation Techniques, *IEEE transactions on pattern analysis and machine intelligence*, Vol. PAMI-1, No. 1, January 1979.
- 5.14 A. Baer, C. Eastman and M. Henrion, Geometric modelling: a survey, *Computer aided design*, Vol. 11, No. 5, September 1979.
- 5.15 CIME Staff Report (Prepared with the assistance of Cliff Brown, Robert Light and Lawrence W. Rosenfeld), *CAD Follows New Scripts*, *Mechanical Engineering*, pp. 62-67, November 1989.
- 5.16 Timothy P. White Sr., Optics for Precision Gaging with Machine Vision, *SME VISION '87 Conf.* Detroit, MI, June 1987.
- 5.17 David A. Butler and Patricia K. Pierson, Correcting Distorsion in Digital Images, *SME VISION '87 Conf.* Detroit, MI, June 1987.
- 5.18 Heng H. Chang, Image Tresholding for Blueprint Image Extraction, *SME VISION '87 Conf.* Detroit, MI, June 1987.
- 5.19 G. B. Porter, An application of grey level image processing to an industrial dimensional inspection problem, *Pattern Recognition*, Vol. 14, Nos. 1-6, pp. 405-410, 1981.
- 5.20 David W. Paglieroni and Anil K. Jain, Contour control point algorithms for shape measurement and inspection, *SPIE Vol. 730 Automated Inspection and Measurement* (1986).
- 5.21 Larry S. Davis, Two-Dimensional Shape Representation, *Handbook of pattern recognition and image processing*, Chapter 10, Edited by Tzay Y. Young & King-Sun Fu, Academic Press Inc., 1986.

BIBLIOGRAPHIE (suite)

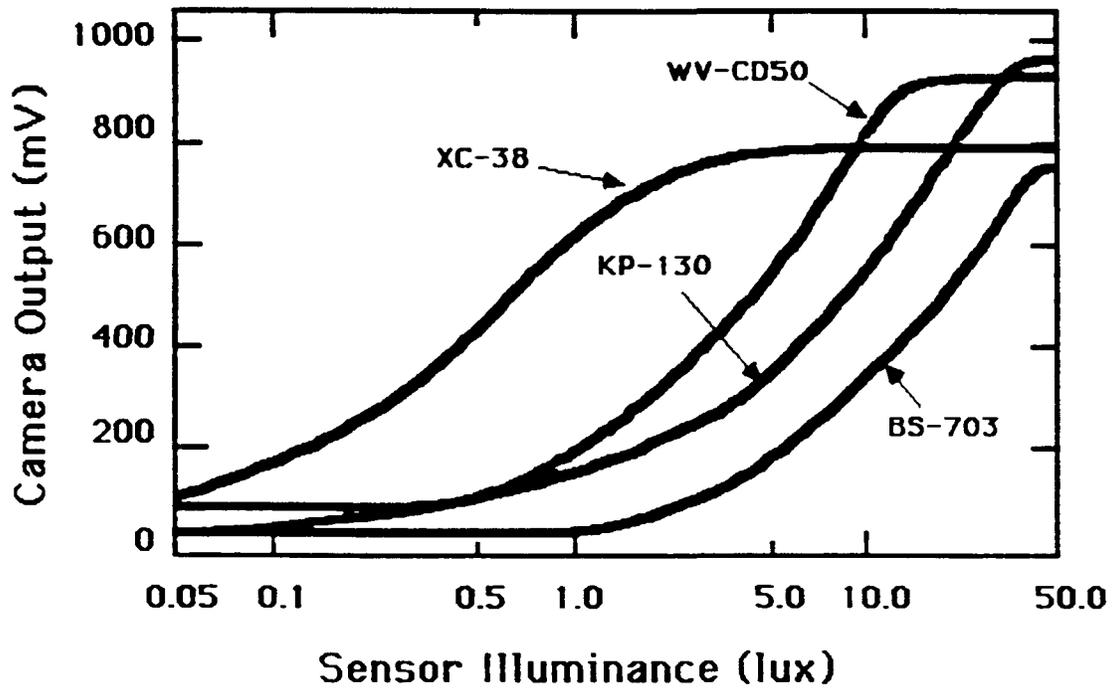
- 5.22 Yoshiaki Shirai, Analyzing intensity arrays using knowledge about scenes, Psychology of computer vision, Chapter 3, McGraw-Hill Computer Sciences Series, 1975.
- 5.23 S. Olympieff, J. C. Pineda and P. Horaud, The use of vision for industrial inspection, Proceeding of the 6th international conference on Automated Inspection and Product Control, Birmingham, U.K. 27-29 April 1982.
- 5.24 Ian Overington and Philip Greenway, Theoretical and practical super-resolution performance in simple processing of local form, motion and stereo disparity, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.25 Stanley H. Zisk and Norman Wittels, How an Edge Looks to a camera, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.26 Richard A. Young, Locating Industrial Parts with Subpixel Accuracies, SPIE Vol. 728 Optics, Illumination, and Image Sensing for Machine Vision, (1986).
- 5.27 Michael Negin, Subpixel Resolution Methodology Limitations, SME VISION '87 Conf. Detroit, MI, June 1987.
- 5.28 O. Robert Mitchell, Edward P. Lyvers, Kirk A. Dunkelberger and Mark L. Akey, Recent Results in Precision Measurements of Edges, Angles, Areas and Perimeters, SPIE Vol. 730 Automated Inspection and Measurement, (1986).
- 5.29 Arnold C. Englander, Expanding Machine Vision Gauging with Sub-Pixel Techniques, Sensors, Vol. 4, No. 6, 9-18, June 1987.
- 5.30 William Brogan & George Nagy, Lacunae in the theory of digital image registration, Pattern Recognition in Practice, E. S. Gelsema and L. N. Kanal (eds.), North-Holland Publishing Company, 1980.
- 5.31 Juha Roning and E. L. Hall, SPIE Vol. 730 Automated Inspection and Measurement, (1986).
- 5.32 Dov Dori and Amir Pnueli, The Grammar of Dimensions in Machine Drawings, Computer vision, graphics, and image processing 42, 1-18, (1988).
- 5.33 P. W. Woods, C. J. Taylor, D. H. Cooper and R. N. Dixon, The use of geometric and grey-level models for industrial inspection, Pattern Recognition Letters 5, No. 1, 11-17, 1987.

BIBLIOGRAPHIE (suite)

- 5.34 Faryar Etesami and John J. Uicker Jr., Automatic Dimensional Inspection of Machine Part Cross-Sections Using Fourier Analysis, Computer vision, graphics, and image processing 29, 216-247 (1985).
- 5.35 R. M. Spencer, Tactile Sensing in Flexible Manufacturing Cells, Mechanical Engineering Transactions, 1985
- 5.36 David B. Arnold, Gillian Hall and Graham J. Reynolds, GKS programming in a PHIGS Environment, Computer Graphics Forum 4, 349-358, 1985.
- 5.37 Peter R. Bono, Report of the Timberline Meeting of ISO TC97/SC21/WG2 "Computer Graphics", Computer Graphics Forum 4, 349-358, 1985.
- 5.38 Aristides A. G. Requicha, Toward a Theory of Geometric Tolerancing, The International Journal of Robotics Research, Vol. 2, No. 4, Winter 1983.

ANNEXE A

Fonctions de transfert de quelques caméras à technologie CCD disponibles commercialement.



Transfer functions for some Machine Vision Cameras

ANNEXE B

**Description des programmes du Micro-PDP 11/23
utilisés pour l'acquisition d'images.**

GRAPHV.CMD

Ce programme permet de fixer le "driver" GRAPH dans la mémoire de l'ordinateur. Il définit des adresses de mémoire pour l'accès aux différents registres des cartes graphiques

INITVAF.MAC

Ce programme initialise les différents paramètres des cartes graphiques. Pour l'explication des différentes valeurs, se référer à la partie "DISPLAY INITIALISATION" du manuel des cartes graphiques.

SYNCINT.MAC

Permet la synchronisation de l'image sur le signal interne de la carte graphique.

SYNCEXT.MAC

Permet la synchronisation de l'image sur le signal externe soit le signal de la caméra vidéo.

MOY.FTN

Ce programme permet de sauver une image, c'est à dire d'enregistrer la mémoire d'écran dans un fichier de données. On a besoin de quatre bits pour enregistrer le niveau de gris de chaque pixel ($2^4 = 16$ valeurs possibles). Ce programme permet aussi de faire la moyenne de plusieurs images afin d'éliminer le bruit de type "gaussien".

RESTO.FTN

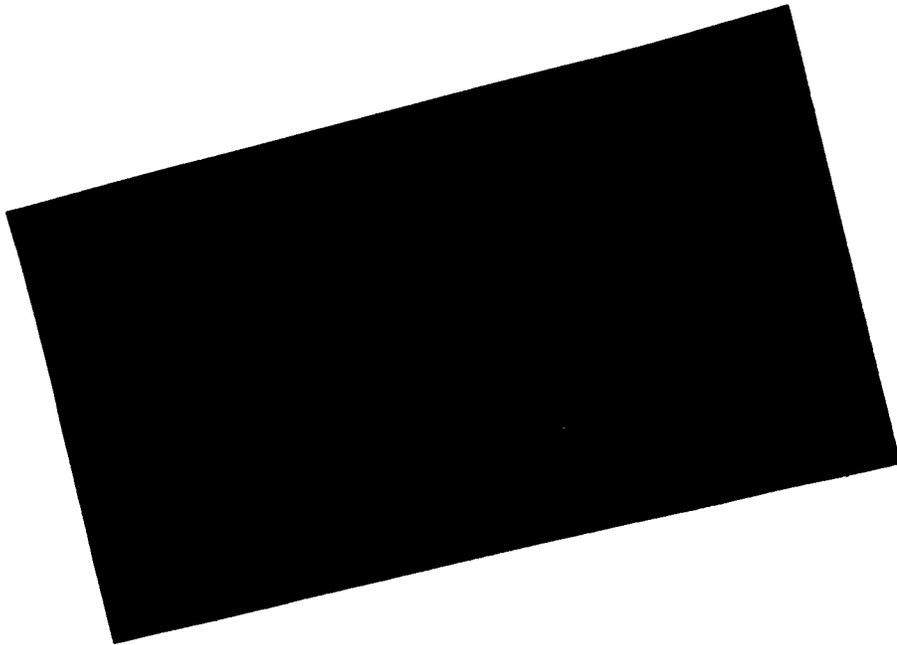
Permet de ramener en mémoire d'écran un fichier préalablement enregistré.

ANNEXE C

Objets utilisés pour l'obtention des résultats.

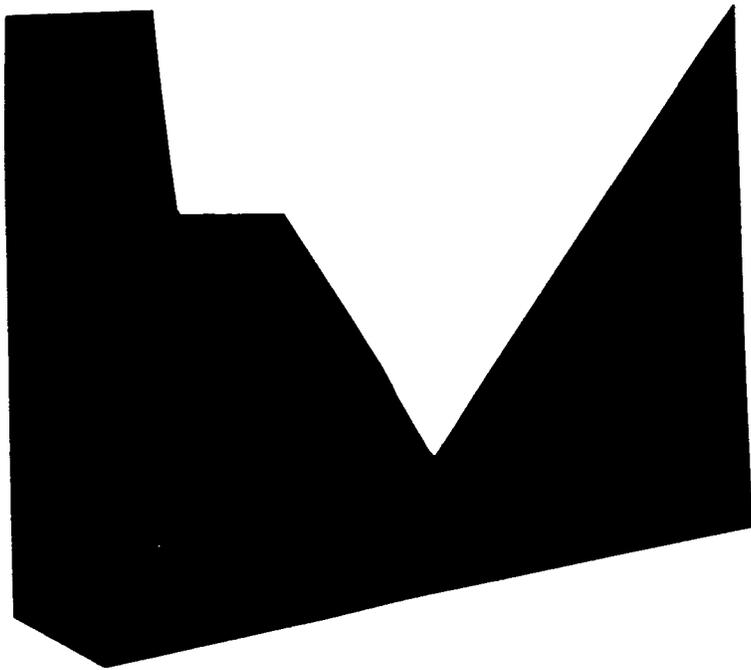
Rectangle

Échelle 1:1



Polygone

Échelle 1:1



ANNEXE D

Fichiers de données du modèle.

<u>Nom du fichier</u>	<u>Description</u>
ET1	Élément linéaire de type 1
CARRE	Carré de côté 5.0 formé de quatre éléments de type 1
FOR072	Modèle du rectangle présenté à l'annexe C
FOR063	Modèle du polygone présenté à l'annexe C

<u>Type du fichier</u>	<u>Description</u>
.SHP	Topologie
.SCH	Contraintes
.PAR	Paramètres
.TOL	Tolérances
.VIS	Simulation des données de vision pour ET1 et CARRE

2
1
1
1
1
2

4
1
1
2
3
1
4
5
1
3
4
6

4
2.0000
1.0000
5.0000
5.8310

4

0.0001	0.0001
0.0001	0.0001
0.0001	0.0001
0.0002	0.0002

2						
2.0000	0.0000	0.0000				
0.0000	0.0000	0.0000				
1.0000	0.0000	0.0000				
5.0000	0.0000	0.0000				
0.0000	0.0000	0.0000				
6.0000	0.0000	0.0000				
1						
1						
1						
1						
2						
2	0.0000	0.0000	0.0000	0.0000	2.0000	0.0000

4
1
4
1
1
2
1
2
3
1
3
4
1
4
1

8
1
1
2
3
1
4
5
1
3
4
6
5
4
6
7
9
5
7
9
10
12
5
1
3
10
12
6
1
3
4
6
1
3
10
12

8
0.0000
0.0000
5.0000
5.0000
4.0000
5.0000
4.0000
90.0000

8	
6.0000	6.0000
5.0000	5.0000
5.0000	5.0000
0.0000	0.0000
0.1000	0.1000
0.2000	0.2000
0.2000	0.2000
7.0000	7.0000

4		
0.0000	0.1000	0.1000
0.0000	0.1000	0.1000
0.0000	0.1000	0.1000
5.0000	0.1000	0.1000
0.0000	0.1000	0.1000
0.0000	0.1000	0.1000
5.0000	0.1000	0.1000
0.0000	0.1000	0.1000
4.0000	0.1000	0.1000
0.0000	0.1000	0.1000
0.0000	0.1000	0.1000
4.0000	0.1000	0.1000

1						
4						
1						
1						
2						
1						
2						
3						
1						
3						
4						
1						
4						
1						
2	0.0000	0.0000	0.0000	0.0000	2.0000	0.0000

4
1
4
1
1
2
1
2
3
1
3
4
1
4
1

8
1
1
2
3
1
4
5
1
3
4
6
5
4
6
7
9
5
7
9
10
12
5
1
3
10
12
6
1
3
4
6
1
3
10
12

8
0.0000
0.0000
5.0000
5.0000
5.0000
5.0000
5.0000
5.0000
90.0000

9
1
9
1
1
2
1
2
3
1
3
4
1
4
5
1
5
5
1
5
7
1
7
8
1
8
9
1
9
1

18
1
1
2
2
1
3
4
2
4
3
1
5
4
2
6
3
1
7
4
2
8
3
1
9
4
2
10
3
1
11
4
2
12
3
1
13
4
2
14
3
1
15
4
2
16
3
1
17
4
2
18

18
0.0000
1.0000
2.0000
6.0000
95.0000
9.0000
95.0000
79.5000
54.0000
19.0000
36.0000
52.0000
23.0000
53.0000
19.5000
79.0000
0.0000
79.0000

ANNEXE E

Liste des programmes

<u>Programme</u>	<u>Sous-programme</u>
HISTO	DEPART TOUR CALIBRE
POLY	FIT2 DIST INTER
GEO	LIRESHP LIRESCH LIREPAR LIRETOL LIREVIS VARI_TOPO PARA PRECI VARI_PARA TOLE

```

C=====
C HISTO.FOR
C
C PROGRAMME DE TRAITEMENT D'IMAGE PAR HISTOGRAMME
C
C AUTEUR : REJEAN FOURNIER
C         20 OCTOBRE 1987
C         MODIFIE LE 07 MARS 1992
C

```

```

C-----
C
C SOUS-PROGRAMMES APPELES : CE277.GKS_REGISJLIRE.FOR
C                           CE277.GKS_REGISJAFFICHE.FOR
C                           CE277.GKS_REGISJGKS_OPEN_WKS
C                           CE277.GKS_REGISJGKS_CLOSE_WKS
C                           CALIBRE.FOR
C                           DEPART.FOR
C                           TOUR.FOR
C

```

```

C-----
C
C VARIABLES UTILISEES :
C
C   ECRAN  : MATRICE 512 X 512 CONTENANT LA COULEUR DE CHAQUE
C           PIXEL D'UNE IMAGE
C   SEP    : NUMERO DE SEPARATION DES COULEURS
C   PIX1   : VALEUR D'UN PIXEL
C   PIX2   : VALEUR D'UN PIXEL A DROITE DE PIX1
C   FAC    : FACTEUR DE DISTORSION VERTICALE DE L'IMAGE
C           (CAUSE LORS DE LA DIGITALISATION PAR LA CARTE MATROX)
C   ECH    : FACTEUR D'ECHELLE DETERMINE PAR L'USAGER
C

```

```

C=====
C
C   COMMON /ECHELLE/FAC,ECH
C   BYTE  ECRAN(512,512),PIX1,PIX2
C   CHARACTER*35 NOM
C   REAL  SEP,SEP2
C

```

```

C---LECTURE DE L'IMAGE DANS LE FICHER DESIRE
100  CALL LIRE(ECRAN,NOM)
C
C---AFFICHAGE DE L'IMAGE ORIGINALE
    CALL AFFICHE(ECRAN,512,100,0,0)
200  CONTINUE
C
C---CHANGE LA MAP DE COULEUR
300  WRITE(6,2040)
2040  FORMAT('+_',60X,$)
    WRITE(6,1000)SEP2
1000  FORMAT('+_NUMERO DE SEPARATION (0.5 A 14.5 0.0=FIN): ',F5.1,$)
    READ(5,1010)SEP
1010  FORMAT(F5.2)
C
C---CHANGEMENT DE LA MAP DE COULEUR EN FONCTION DU NUMERO DE SEPARATION
C
    IF (SEP.NE.0.0) THEN
        CALL GKS_OPEN_WKS(1,A,A)
C-MOIR 0 A NSEP
        SEP2=SEP

```

```

      DD 20 I=0,NSEP
        WRITE(6,1020)I
1020   FORMAT('+SCM',I2,'(AD))',5)
20     CONTINUE
C--BLANC NSEP+1 A 15
      DD 30 I=NSEP+1,15
        WRITE(6,1030)I
1030   FORMAT('+SCM',I2,'(AW))',5)
30     CONTINUE
C
      CALL GKS_CLOSE_WKS(1)
      GOTO 300
ELSE
      NSEP=INT(SEP2)
ENDIF
C
C--DETERMINATION DU POINT DE DEPART A GAUCHE DE LA SURFACE VOULUE
400   CALL DEPART(JJ,II,NOM)
C--LOCALISATION DU REBORD DE LA SURFACE A PARTIR DU POINT DE DEPART
C ET EN ALLANT VERS LA DROITE
      DD 40 I=II,480
      DD 50 J=JJ,512
        PIX1=ECRAN(J-1,I)
        PIX2=ECRAN(J,I)
        IF((PIX1.LE.NSEP).AND.(PIX2.GT.NSEP)) THEN
          WRITE(6,1050)J,I
1050   FORMAT('/','+ _BORD DE LA SURFACE LOCALISE A: ',2I4,' ',/)
C--ENREGISTRE LA SURFACE DANS UN FICHIER
          R=1.0
          IXA=J
          IYA=I
          IXB=J-1
          IYB=I
          NCOL=512
          NLLIG=512
          CALL TOUR( ECRAN,NCOL,NLLIG,IXA,IYA,IXB,IYB,R,SEP2,NOM)
          GOTO 60
        ENDIF
50     CONTINUE
40     CONTINUE
C
60     CONTINUE
C
      WRITE(6,1060)
1060   FORMAT('+ _DONNER: 1=IMAGE 2=ECHELLE 3=NIVEAU 4=SURFACE 5=FIN >
1     ',5)
      READ(5,*)ICHOIX
      GOTO (100,200,300,400,500),ICHOIX
C--FIN
500   CONTINUE
C
      END

```

SUBROUTINE DEPART(COLONNE,LIGNE,NOM)

```

C=====
C SOUS-PROGRAMME DONNANT UN POINT DE DEPART A GAUCHE DE LA
C SURFACE VOULUE
C
C AUTEUR : REJEAN FOURNIER
C         07 MARS 1992
C         MODIFIE LE
C
C-----
C SOUS-PROGRAMMES APPELES :
C-----
C
C VARIABLES UTILISEES :
C
C     COLONNE: COLONNE DE DEPART (1 A 512)
C     LIGNE   : LIGNE DE DEPART (1 A 512)
C     NOM     : NOM DE L'IMAGE
C=====
C
C     INTEGER*2 COLONNE,LIGNE
C     CHARACTER*5 NOM
C
C---ENTREE DE LA POSITION DE DEPART AVEC LA SOURIS
C400  WRITE(6,1040)
C1040  FORMAT('+ _INDIQUER UN POINT A GAUCHE DE LA SURFACE DESIREE',%)
C     CALL MINPUT(JJ,II)
C
C----DETERMINATION DU POINT DE DEPART AUTOMATIQUE
C     IF (NOM(14:20) .EQ. '11M4187') THEN
C         COLONNE A GAUCHE
C         COLONNE = 50
C         LIGNE DU MILIEU
C         LIGNE = 256
C     ELSE IF (NOM(14:19) .EQ. '2DEC87') THEN
C         COLONNE A GAUCHE
C         COLONNE = 50
C         LIGNE PRES DU MILIEU
C         LIGNE = 220
C     ELSE IF (NOM(14:20) .EQ. '14DEC87') THEN
C         COLONNE A GAUCHE
C         COLONNE = 50
C         LIGNE DU MILIEU
C         LIGNE = 256
C     ELSE IF (NOM(14:20) .EQ. '10MAR88') THEN
C         COLONNE A GAUCHE
C         COLONNE = 50
C         LIGNE PRES DU MILIEU
C         LIGNE = 250
C     ELSE
C         WRITE(6,*) '+ POINT DE DEPART CENTRAL POUR ',NOM
C SI ON N'A PAS D'IMAGE, ON PREND UN POINT DE DEPART CENTRAL
C     COLONNE DU MILIEU
C     COLONNE = 256
C     LIGNE DU MILIEU
C     LIGNE = 256
C ENDIF

```

C-FIN
1000 RETURN
C
END

SUBROUTINE TOUR(ECRAN, NCOL, NLI, IXA, IYA, IXB, IYB, R, SEP, NOM)

SOUS-PROGRAMME ENREGISTRANT DANS UN FICHIER LE CONTOUR D'UNE SURFACE
FORMEE DE PIXELS A PARTIR D'UN RAYON DE MESURE R

AUTEUR : REJEAN FOURNIER

DATE : 10 NOVEMBRE 1987

MODIFIE EN MARS 88

LE CONTOUR DESIRE DOIT PASSER ENTRE LES DEUX PIXELS
A (IXA, IYA) ET B (IXB, IYB). CES DEUX PIXELS DOIVENT
AVOIR UN COTE COMMUN ET NE PAS ETRE DE LA MEME COULEUR
ON DOIT AVOIR ECRAN(IXA, IYA) > SEP > ECRAN(IXB, IYB)

L'ORIGINE DES AXES SE SITUE DANS LE COIN SUPERIEUR GAUCHE
L'AXE X EST HORIZONTAL ET L'AXE Y EST VERTICAL

LES PIXELS N'AYANT QU'UN COIN COMMUN SONT CONSIDERES COMME
FAISANT PARTIE DE LA MEME SURFACE.

---DESCRIPTION DES ARGUMENTS :

ECRAN : GENRE : BYTE ECRAN(NCOL*NLI)
CONTIENT LA COULEUR DES PIXELS (0 A 255) A PARTIR
DU COIN SUPERIEUR GAUCHE JUSQU'AU COIN INFERIEUR
DROIT (COMME POUR LA LECTURE D'UNE PAGE).

NCOL : GENRE : INTEGER
CONTIENT LE NOMBRE DE COLONNES DE L'ECRAN
NCOL >= 1

NLI : GENRE : INTEGER
CONTIENT LE NOMBRE DE LIGNES DE L'ECRAN
NLI >= 1

IXA : GENRE : INTEGER
POSITION EN X DU PIXEL DU CONTOUR A ENREGISTRER

IYA : GENRE : INTEGER
POSITION EN Y DU PIXEL DU CONTOUR A ENREGISTRER

IXB : GENRE : INTEGER
POSITION EN X DU PIXEL ADJACENT AU PIXEL A ET A
L'EXTERIEUR DU CONTOUR A MESURER

IYB : GENRE : INTEGER
POSITION EN Y DU PIXEL ADJACENT AU PIXEL A ET A
L'EXTERIEUR DU CONTOUR A MESURER
(LE CONTOUR A MESURER PASSE ENTRE IX1, IY2 ET IX2, IY2
ON DOIT AVOIR ECRAN(IX1, IY1) <> ECRAN(IX2, IY2)).

R : GENRE : REAL
RAYON DE MESURE DESIRE. CE RAYON EST EXPRIME EN
UNITES DE PIXEL. UNE UNITE DE PIXEL EST EGALE A LA
LONGUEUR DU COTE D'UN PIXEL.
ON DOIT AVOIR R >= 1.0
AU RETOUR DU SOUS-PROGRAMME, R EST LE RAYON MOYEN
CALCULE.

SEP : VALEUR DE LA COULEUR DE SEPARATION POUR LE CONTOUR
EXEMPLE: SI SEP=5.5, LA SEPARATION DU CONTOUR A
ENREGISTRER SERA ENTRE 5 ET 6 (ENTRE LES COULEURS 5 ET 6)

```

C
C XMOVE : DISTANCE EN X DU DERNIER POINT FIXE
C YMOVE : DISTANCE EN Y ...
C
C NXA : POSITION EN X DU PIXEL INTERNE AU CONTOUR
C NYA : POSITION EN Y ...
C NXB : POSITION EN X DU PIXEL EXTERNE AU CONTOUR
C NYB : POSITION EN Y ...
C
C N1X : POSITION EN X DU POINT 1 (POINT ADJACENT AU POINT A
C ET DANS LA DIRECTION TETA
C N1Y : POSITION EN Y DU POINT 1 (POINT ADJACENT AU POINT A
C ET DANS LA DIRECTION TETA
C N2X : POSITION EN X DU POINT 2 (POINT ADJACENT AU POINT B
C ET DANS LA DIRECTION TETA
C N2Y : POSITION EN Y DU POINT 2 (POINT ADJACENT AU POINT B
C ET DANS LA DIRECTION TETA
C
C TETA : ANGLE DE LA DIRECTION DU CONTOUR
C (MESURE EN RADIANS, TETA=0.0 QUAND IL POINTE PAR EN
C HAUT ET CROISSANT DANS LE SENS HORAIRE)
C X : POSITION EN X D'UN POINT POUR L'ENREGISTREMENT DU CONTOUR
C Y : POSITION EN Y ...
C
C FAC : FACTEUR DE DISTORSION VERTICALE DU A LA DIGITALISATION
C PAR LA CARTE MATROX
C ECH : ECHELLE A LAQUELLE ON VEUT ENREGISTRER LE CONTOUR
C (LES COORDONNEES DE CHAQUE POINT DU CONTOUR SONT LES
C VRAIES COORDONNEES AVEC L'ORIGINE 0,0 DANS LE COIN
C SUPERIEUR GAUCHE DE L'IMAGE)

```

=====

C----INITIALISATION

```

COMMON /ECHELLE/FAC,ECH
BYTE ECRAN(NCCL,NLIG)
DIMENSION X(5000),Y(5000)
CHARACTER*35 NOM
DATA MP/6/,MR/5/,PI/3.14159265359/

```

C----CHOIX DU FICHIER D'ENREGISTREMENT DES DONNEES

```

WRITE(MP,1000)
READ(MR,1010)ND
1000 FORMAT(/,' DONNER LE NO DU FICHIER D'ENREGISTREMENT
1 DU CONTOUR: ',*)
1010 FORMAT(I3)

```

C----VALEURS INITIALES

```

N=0
RM=0.0
NXA=IXA
NYA=IYA

```

C-LES COORDONNEES DES POINTS DU CONTOUR VARIENT DE 0.5 A 512.5

C (L'ECHELLE EST AJOUTEE A LA FIN)

```

X(1)=REAL(IXA)+0.5
Y(1)=REAL(IYA)-0.5
NXB=IXB
NYB=IYB
NSEP=INT(SEP)
XMOVE=0.0

```

```

C
C----ANGLE INITIAL
      IF (NXA.GT.NXB) THEN
          TETA=0.0
      ENDIF
      IF (NYA.LT.NYB) THEN
          TETA=PI/2.0
      ENDIF
      IF (NXA.LT.NXB) THEN
          TETA=PI
      ENDIF
      IF (NYA.GT.NYB) THEN
          TETA=3.0*PI/2.0
      ENDIF

C
C----AVANCEMENT LE LONG DU CONTOUR
10    CONTINUE

C
C--POSITION DU POINT 1 (DEVANT LE POINT B)
      N1X=NXB+NINT(SIN(TETA))
      N1Y=NYB+NINT(COS(TETA))

C
C--POSITION DU POINT 2 (DEVANT LE POINT A)
      N2X=NXA+NINT(SIN(TETA))
      N2Y=NYA+NINT(COS(TETA))

C
C--TEST SI LE POINT 1 DEPASSE LES LIMITES DE L'ECRAN
      IF ((N1X.LT.1).OR.(N1X.GT.NCOL).OR.
1      (N1Y.LT.1).OR.(N1Y.GT.NLIG)) GOTO 20

C
C--TEST SI LE POINT 1 EST DE LA COULEUR DU CONTOUR A ENREGISTRER
      IF (ECRAN(N1X,N1Y).GT.NSEP) THEN
          NXA=N1X
          NYA=N1Y
          XMOVE=XMOVE+SIN(TETA)
          YMOVE=YMOVE+COS(TETA)
          TETA=TETA-PI/2.0
          GOTO 40
      ENDIF

C
C--TEST SI LE POINT 2 DEPASSE LES LIMITES DE L'ECRAN
20    CONTINUE
      IF ((N2X.LT.1).OR.(N2X.GT.NCOL).OR.
1      (N2Y.LT.1).OR.(N2Y.GT.NLIG)) GOTO 30

C
C--TEST SI LE POINT 2 EST DE LA COULEUR DU CONTOUR A ENREGISTRER
      IF (ECRAN(N2X,N2Y).GT.NSEP) THEN
          NXA=N2X
          NYA=N2Y
          NXB=N1X
          NYB=N1Y
          XMOVE=XMOVE+SIN(TETA)
          YMOVE=YMOVE+COS(TETA)
          GOTO 40
      ENDIF

C
C--ROTATION DE +90 DEGRES (+PI/2 RAD)
30    CONTINUE
      NXB=N2X
      NYB=N2Y

```

```
YMOVE=YMOVE+COS(TETA)
TETA=TETA+PI/2.0
```

```
C
C--CALCUL DU RAYON
40  CONTINUE
    RC=SQRT(XMOVE*XMOVE+YMOVE*YMOVE)
C
C----VERIFICATION SI RETOUR ENTRE LES 2 PIXELS DE DEPART
    IF ((NXA.EQ.IXA).AND.(NYA.EQ.IYA).AND.
1    (NXB.EQ.IXB).AND.(NYB.EQ.IYB)) GOTO 90
C
C----TEST SI LE RAYON CALCULE EST <=R
C
    IF (RC.LT.R) GOTO 10
C
C----NOUVEAU POINT POUR LE CALCUL DU RAYON
C
    N=N+1
    RM=RM+RC
    X(N+1)=X(N)+XMOVE
    Y(N+1)=Y(N)+YMOVE
    XMOVE=0.0
    YMOVE=0.0
    GOTO 10
C
C----FIN DU CONTOUR
C
90  IF (RC.GE.R) THEN
    N=N+1
    RM=RM+RC
    ENDIF
C
C--RAYON MOYEN
    IF (N.GT.0) THEN
        R=RM/FLOAT(N)
    ENDIF
C
    SEP=REAL(N)
C
C----CALCUL DE L'ECHELLE DE CALIBRATION
C
    CALL CALIBRE(SEP,NOM,N,X,Y)
C
C--ENREGISTREMENT DU CONTOUR DANS LE FICHIER FOROMD.DAT
C-NOMBRE DE POINTS POUR LE PROGRAMME GREEN.FOR
    WRITE(MD,1020)N
1020  FORMAT(I5)
C-POSITION X ET Y DE CHAQUE POINT (SENS ANTIHORAIRE)
C MISE A L'ECHELLE EN TENANT COMPTE DE LA DISTORSION VERTICALE
    DO 100 I=N,1,-1
    WRITE(MD,1030)X(I)*ECH,Y(I)*ECH/FAC
1030  FORMAT(2F10.5)
100  CONTINUE
C
    RETURN
END
```

SUBROUTINE CALIBRE(CSEP,NOM,N,X,Y)

```

C=====
C SOUS-PROGRAMME DE CALIBRAGE D'UNE IMAGE
C
C AUTEUR : REJEAN FOURNIER
C         07 MARS 1992
C
C MODIFIE LE
C-----
C SOUS-PROGRAMMES APPELES :
C-----
C VARIABLES UTILISEES :
C
C     SEP   : NUMERO DE SEPARATION DES COULEURS (0.5 A 14.5)
C     NOM   : NOM DE L'IMAGE A CALIBRER
C     N     : NOMBRE DE POINTS DU CONTOUR A CALIBRER
C     X,Y   : COORDONNEES DU CONTOUR
C
C     FAC   : FACTEUR DE DISTORSION VERTICALE DE L'IMAGE
C            (CAUSE LORS DE LA DIGITALISATION PAR LA CARTE MATROX)
C     ECH   : FACTEUR D'ECHELLE (PIXELS EN UNITES DE MESURE)
C-----
C
C     COMMON /ECHELLE/FAC,ECH
C     CHARACTER*35 NOM
C     INTEGER N
C     REAL X(1),Y(1)
C
C---LECTURE DE L'IMAGE DE CALIBRAGE DANS LE FICHIER DESIRE
C100  CALL LIRE(SCRAN)
C
C---AFFICHAGE DE L'IMAGE DE CALIBRAGE
C     CALL AFFICHE(SCRAN,512,300,0,0)
C
C---ENTREE DE L'ECHELLE
C200  CONTINUE
C
C----ENTREE DE L'ECHELLE PAR LA SOURIS
C     WRITE(6,2000)
C2000 FORMAT('+ _INDIQUER LE PREMIER POINT POUR L''ECHELLE')
C     CALL MINPUT(IX1,IY1)
C     X1=REAL(IX1)
C     Y1=REAL(IY1)
C
C     WRITE(6,2010)
C2010 FORMAT('+ _INDIQUER LE DEUXIEME POINT POUR L''ECHELLE')
C     CALL MINPUT(IX2,IY2)
C     X2=REAL(IX2)
C     Y2=REAL(IY2)
C
C     WRITE(6,2020)
C2020 FORMAT('+ _DONNER LA DISTANCE ENTRE LES DEUX POINTS (REAL): ',)
C     READ(5,2030)DIST
C2030 FORMAT(F10.5)

```

```
2040  FORMAT('+ _',60X,$)
```

```
C  
C----ENTREE DE L'ECHELLE MANUELLEMENT
```

```
C      WRITE(6,2000)
```

```
C2000  FORMAT('+ _INDIQUER LE PREMIER POINT POUR L'ECHELLE X1,Y1: ', $)
```

```
C      READ(5,2001)X1,Y1
```

```
C2001  FORMAT(2F10.5)
```

```
C  
C      WRITE(6,2010)
```

```
C2010  FORMAT('+ _INDIQUER LE DEUXIEME POINT POUR L'ECHELLE X2,Y2: ', $)
```

```
C      READ(5,2011)X2,Y2
```

```
C2011  FORMAT(2F10.5)
```

```
C  
C      WRITE(6,2020)
```

```
C2020  FORMAT('+ _DONNER LA DISTANCE ENTRE LES DEUX POINTS (REAL): ', $)
```

```
C      READ(5,2030)DIST
```

```
C2030  FORMAT(F10.5)
```

```
C  
C----ENTREE DE L'ECHELLE AUTOMATIQUE
```

```
      IF (NOM(14:20) .EQ. '11MAI87') THEN
```

```
        WRITE(6,*)NOM
```

```
        UN PIXEL=UNE UNITE
```

```
        X1=0.0
```

```
        Y1=0.0
```

```
        X2=3.0
```

```
        Y2=4.9948
```

```
        DIST=5.0
```

```
      ELSE IF (NOM(14:19) .EQ. '20DEC87') THEN
```

```
        WRITE(6,*)NOM
```

```
        ON TROUVE LES COORDONNEES DU POINT YMAX(Y1) ET YMIN(Y2)
```

```
        Y1=Y(1)
```

```
        Y2=Y(1)
```

```
        DO 100 I=2,N
```

```
          IF (Y1.LT.Y(I)) THEN
```

```
            Y1=Y(I)
```

```
            X1=X(I)
```

```
          ENDIF
```

```
          IF (Y2.GT.Y(I)) THEN
```

```
            Y2=Y(I)
```

```
            X2=X(I)
```

```
          ENDIF
```

```
100    CONTINUE
```

```
        DISTANCE EN MM
```

```
        DIST=117.0
```

```
      ELSE IF (NOM(14:20) .EQ. '14DEC87') THEN
```

```
        WRITE(6,*)NOM
```

```
        Y1=Y(1)
```

```
        Y2=Y(1)
```

```
        DO 200 I=2,N
```

```
          IF (Y1.LT.Y(I)) THEN
```

```
            Y1=Y(I)
```

```
            X1=X(I)
```

```
          ENDIF
```

```
          IF (Y2.GT.Y(I)) THEN
```

```
            Y2=Y(I)
```

```
            X2=X(I)
```

```
          ENDIF
```

```
200    CONTINUE
```

```
        DISTANCE EN MM
```

```
        DIST=100.0
```

```
WRITE(6,*)NOM
```

```
Y1=Y(1)
```

```
Y2=Y(1)
```

```
DO 300 I=2,N
```

```
  IF (Y1.LT.Y(I)) THEN
```

```
    Y1=Y(I)
```

```
    X1=X(I)
```

```
  ENDIF
```

```
  IF (Y2.GT.Y(I)) THEN
```

```
    Y2=Y(I)
```

```
    X2=X(I)
```

```
  ENDIF
```

```
300 CONTINUE
```

```
C DISTANCE EN MM
```

```
DIST=36.0
```

```
ELSE
```

```
WRITE(6,*)'+ FACTEUR D'ECHELLE INDETERMINE POUR ',NOM
```

```
C SI ON N'A PAS D'IMAGE,ON RESTE EN PIXELS (UN PIXEL = UNE UNITE)
```

```
X1=0.0
```

```
Y1=0.0
```

```
X2=3.0
```

```
Y2=4.9948
```

```
DIST=3.0
```

```
ENDIF
```

```
C  
C-CALCULS RELATIFS A L'ECHELLE
```

```
FAC=1.2487
```

```
ECH=DIST/SQRT(((Y2-Y1)/FAC)**2.0+(X2-X1)**2.0)
```

```
C  
C-FIN
```

```
500 RETURN
```

```
C  
C ENL
```

```

=====
C
C RECONNAISSANCE DE POLYGONES QUELCONQUES DANS UNE IMAGE
C A PARTIR D'UN FICHER CONTENANT LES COORDONNEES DE TOUS
C LES POINTS DEFINISSANT LE CONTOUR DE LA SURFACE
C (CE FICHER DOIT DABORD ETRE GENERE PAR LE PROGRAMME
C HISTO.FOR QUI UTILISE LE SOUS-PROGRAMME TOUR.FOR)

```

```

C AUTEUR : REJEAN FOURNIER

```

```

C DATE : 8 MARS 1988

```

```

=====
C --PROPRIETES EVALUEES PAR LE PROGRAMME:

```

```

C     TOUT LES SOMMETS DU POLYGONE (SENS ANTI-HORAIRE)
C     EQUATIONS DE TOUTES LES DROITES FORMANT LE POLYGONE

```

```

C --METHODE UTILISEE (ALGORITHME):

```

```

C -ON TROUVE DABORD LES EXTREMUMS DU TOUR DE LA SURFACE
C   (MINIMUMS ET MAXIMUMS SELON X ET Y)
C -A PARTIR DES EXTREMUMS ON TROUVE 2, 3 OU 4 SOMMETS
C -ON CALCULE L'EQUATION DES DROITES ENTRE LES SOMMETS
C   EN SE SERVANT DES POINTS ENTRE CHAQUE PAIRE DE SOMMETS
C -ON CALCULE LA DISTANCE DE CHAQUE POINT DE LA DROITE
C   ENTRE 2 SOMMETS
C -ON FAIT LE TEST SUIVANT POUR LE POINT SE TROUVANT A LA
C   DISTANCE MAXIMUM DE LA DROITE: SI CETTE DISTANCE EST
C   PLUS GRANDE QUE LA VALEUR ACCEPTABLE, ALORS CE POINT
C   DEFINIT UN NOUVEAU SOMMET
C -ON REPETE LE PROCESSUS JUSQU'A CE QU'ON NE TROUVE PLUS
C   DE NOUVEAUX SOMMETS ET DONC QUE LE TOUR DE LA SURFACE
C   SOIT DEFINI COMME UNE SUITE DE DROITES FORMANT UN
C   POLYGONE

```

```

C N.B.: CET ALGORITHME FONCTIONNE POUR TOUT POLYGONE QU'IL
C       SOIT CONVEXE OU CONCAVE

```

```

=====
C --VARIABLES UTILISEES :

```

```

C N : NOMBRE DE POINTS DEFINISSANT LE TOUR DE LA SURFACE
C X(I) : POSITION EN X DU POINT I
C Y(I) : POSITION EN Y DU POINT I
C NXMIN : NUMERO DU X MINIMUM
C NXMAX : NUMERO DU X MAXIMUM
C NYMIN : NUMERO DU Y MINIMUM
C NYMAX : NUMERO DU Y MAXIMUM
C NSOMP : NOMBRE DE SOMMETS DANS LA PILE (TEMPORAIRE)
C PENTE(J) : PENTE DE LA DROITE J      Y=PENTE*X+CTE
C CTE(J) : CONSTANTE DE LA DROITE J   Y=PENTE*X+CTE
C NPILE(K) : PILE DES NUMEROS DE SOMMETS
C DACC : DISTANCE ACCEPTABLE POUR QU'UN POINT FASSE PARTIE
C       D'UNE DROITE (EN UNITES)
C DACCPIX : DISTANCE ACCEPTABLE POUR QU'UN POINT FASSE PARTIE
C          D'UNE DROITE (EN PIXELS)
C NSOM1 : 1ER SOMMET DE LA DROITE

```

```

C NSOM2 : NOMBRE DU POINT AYANT LA DISTANCE MAXIMUM D'UNE
C DROITE (CENTRE NSOM1 ET NSOM2)
C GMAX : DISTANCE DU POINT NSOM3
C NORDITE : NOMBRE DE DROITES FORMANT LE POLYgone
C XX,YY : POINT D'INTERSECTION DE DEUX DROITES
=====

```

```

C --Sous-PROGRAMMES APPELES :
C
C SITE.FOR (POUR TROUVER L'EQUATION DES DROITES A PARTIR
C D'UNE SERIE DE POINTS)
C DIST.FOR (DISTANCE D'UN POINT A UNE DROITE)
=====

```

```

C ----INITIALISATION :
C
C DIMENSION X(5000),Y(5000),PENTE(100),CTE(100),NFILE(100)
C DATA MP/6/,ME/5/,DACCP/11.0/
C NORDITE=0

```

```

C ----LECTURE DES DONNEES N ET X,Y

```

```

WRITE(MP,1000)
READ(ME,1070)MR
WRITE(MP,1060)MR
READ(MR,1000,ERR=100N)
C ORDRE DE LECTURE INVERSE SI LA SURFACE EST NEGATIVE:I=N,1,-1
C DO 5 I=N,1,-1
C READ(MR,1010,ERR=100)X(CI),Y(CI)
C CONTINUE
C FORMATS DE LECTURE
1000 FORMAT(I5)
1010 FORMAT(2F10.5)
1060 FORMAT(/, NOMBRE DU FICHIER DE LECTURE : ',I)
1070 FORMAT(I2)
1080 FORMAT(' FORO',I2,',',.PAT')

```

```

C ----CALCUL DE LA DISTANCE POUR UN PIXEL

```

```

IF (X(1)*N*X(2)) THEN
  PPIX=ABS(X(1)-X(2))
ELSE
  PPIX=ABS(Y(1)-Y(2))
ENDIF
DACCP=DACCP+PPIX
C ----DETERMINATION DES EXTREMUMS
NXXMIN=1
NXXMAX=1
NYMIN=1
NYMAX=1
DO 10 I=2,N
  IF (X(I).LT.X(NXXMIN)) NXXMIN=I
  IF (X(I).GT.X(NXXMAX)) NXXMAX=I
  IF (Y(I).LT.Y(NYMIN)) NYMIN=I
  IF (Y(I).GT.Y(NYMAX)) NYMAX=I
CONTINUE

```

```

I=N
DO WHILE (X(I).EQ.X(NXMIN))
  NXMIN=I
  I=I-1
END DO
I=N
DO WHILE (X(I).EQ.X(NXMAX))
  NXMAX=I
  I=I-1
END DO
I=N
DO WHILE (X(I).EQ.X(NYMIN))
  NYMIN=I
  I=I-1
END DO
I=N
DO WHILE (X(I).EQ.X(NYMAX))
  NYMAX=I
  I=I-1
END DO

```

C
C ---- DETERMINATION DE SOMMETS A PARTIR DES EXTREMUMS
C (3, 4 OU 5 SOMMETS POSSIBLES SONT MIS DANS LA PILE)
C (LE DERNIER SOMMET ETANT AUSSI LE PREMIER)

```

NSOMP=1
NPILE(NSOMP)=NYMIN
IF (NYMIN.NE.NXMAX) THEN
  NSOMP=NSOMP+1
  NPILE(NSOMP)=NXMAX
ENDIF
IF (NXMAX.NE.NYMAX) THEN
  NSOMP=NSOMP+1
  NPILE(NSOMP)=NYMAX
ENDIF
IF (NYMAX.NE.NXMIN) THEN
  NSOMP=NSOMP+1
  NPILE(NSOMP)=NXMIN
ENDIF
IF (NXMIN.NE.NYMIN) THEN
  NSOMP=NSOMP+1
  NPILE(NSOMP)=NYMIN
ENDIF

```

```

C  

C WRITE(6,102)N
C WRITE(6,*)NXMIN,NXMAX,NYMIN,NYMAX
C DO 15 I=1,NSOMP
C WRITE(6,*)I,NPILE(I),X(NPILE(I)),Y(NPILE(I))
C15 CONTINUE

```

C ---- 1ER SOMMET EXTRAIT DU HAUT DE LA PILE

```

NSOM1=NPILE(NSOMP)
NSOMP=NSOMP-1

```

C ---- BUCLE PRINCIPALE
C --- SOMMET SUIVANT EXTRAIT DU HAUT DE LA PILE

```

20 CONTINUE
NSOM2=NPILE(NSOMP)
NSOMP=NSOMP-1

```

```

X1(1)=X(NSOM1)
Y1(1)=Y(NSOM1)
X1(2)=(X(NSOM1)+X(NSOM2))/2.0
Y1(2)=(Y(NSOM1)+Y(NSOM2))/2.0
X1(3)=X(NSOM2)
Y1(3)=Y(NSOM2)
NDATA=3
CALL FIT2(X1,Y1,NDATA,A,B,SIGA,SIGB,CHI2,Q)
C ---DETERMINATION DU NOMBRE DE POINTS ENTRE LES DEUX SOMMETS
C (LES DEUX SOMMETS SONT COMPRIS)
C ---LES POINTS FAISANT PARTIE DE LA DROITE SONT MIS
C DANS UNE VARIABLE TEMPORAIRE X1(),Y1()
  IF (NSOM2.GT.NSOM1) THEN
    DO 30 I=NSOM1,NSOM2
      X1(I-NSOM1+1)=X(I)
      Y1(I-NSOM1+1)=Y(I)
30    CONTINUE
    NDATA=NSOM2-NSOM1+1
  ELSE
    DO 40 I=NSOM1,N
      X1(I-NSOM1+1)=X(I)
      Y1(I-NSOM1+1)=Y(I)
40    CONTINUE
    DO 50 I=1,NSOM2
      X1(I+N-NSOM1+1)=X(I)
      Y1(I+N-NSOM1+1)=Y(I)
50    CONTINUE
    NDATA=N-NSOM1+NSOM2+1
  ENDIF
C ---CALCUL DE LA DISTANCE DE TOUS LES POINTS A LA
C DROITE CALCULEE ENTRE LES DEUX SOMMETS
C CONSERVATION DE LA DISTANCE MAXIMUM
  DMAX=0.0
  NSOM3=0
  DO 60 I=2,NDATA-1
    CALL DIST(B,A,X1(I),Y1(I),D)
    WRITE(5,*)I,D
    IF (D.GT.DMAX) THEN
      DMAX=D
      NSOM3=I
    ENDIF
60  CONTINUE
  NSOM3=NSOM3+NSOM1-1
  IF (NSOM3.GT.N) NSOM3=NSOM3-N
C ---TEST SI LA DISTANCE MAXIMUM EST ACCEPTABLE
C SI OUI: ENREGISTREMENT DES VALEURS
C SI NON: CREATION D'UN NOUVEAU SOMMET
  WRITE(6,*)DMAX,NSOM3
  WRITE(6,*)NSOM1,NSOM2,NDATA
  IF (DMAX.LE.DACC) THEN
C ---CALCUL DE LA DROITE AVEC TOUS LES POINTS ENTRE
C LES DEUX SOMMETS ACTUELS (2 SOMMETS INCLUS)
    CALL FIT2(X1,Y1,NDATA,A,B,SIGA,SIGB,CHI2,Q)
    NDRDITE=NDRDITE+1
    PENTE(NDRDITE)=B
    CTE(NDRDITE)=A
    NDATA(NDRDITE)=NDATA
    NSOM1=NSOM2
  ELSE
    NSOMP=NSOMP+1

```

```

NSDMP=NSDMP+1
NPILE(NSDMP)=NSDM3

```

```

ENDIF

```

```

C ---VERIFICATION SI ON EST RENDU A LA FIN DE LA PILE
C   SI OUI: FIN
C   SI NON: RETOUR AU DEBUT POUR LA PROCHAINE DROITE
C   IF (NSDMP.GT.0) GOTO 20

```

```

C ----CALCUL DES POINTS D'INTERSECTION DES DROITES

```

```

PENTE(NDRDITE+1)=PENTE(1)
CTE(NDRDITE+1)=CTE(1)

```

```

C ----SORTIE DES RESULTATS DANS UN FICHIER POUR QUE
C   LE PROGRAMME DISPLAY.FOR PUISSE LE LIRE

```

```

WRITE(MP,2000)
READ(ME,2010)MR
WRITE(MP,2020)MR

```

```

2000  FORMAT(/ ' NUMERO DU FICHIER D'ECRITURE DE LA SURFACE: ',*)

```

```

2010  FORMAT(I2)

```

```

2020  FORMAT(' FORO ',I2,'.DAT')

```

```

C-NOMBRE DE NOEUDS

```

```

NT=NDRDITE
WRITE(MR,2030)NT

```

```

C-X,Y,Z DE CHAQUE NOEUD ET PRECISION SUR CHAQUE NOEUD

```

```

ZZ=0.0
NDATAS(NDRDITE+1)=NDATAS(1)
DO 25 I=1,NDRDITE

```

```

    CALL INTER(PENTE(I),CTE(I),PENTE(I+1),CTE(I+1),XX,YY)

```

```

    WRITE(6,1100)I,XX,YY,PENTE(I),CTE(I)

```

```

C LE CALCUL DE LA PRECISION SUR CHAQUE NOEUD N'EST QU'UNE APPROXIMATION

```

```

    IF (NDATAS(I).GT.NDATAS(I+1)) THEN
        NDATA=NDATAS(I+1)
    ELSE

```

```

        NDATA=NDATAS(I)
    ENDIF

```

```

RNDATA=REAL(NDATA)
DELTA=DPIX/SQRT(RNDATA)
DELTAXMAX=DELTA
DELTAXMIN=DELTA
DELTAYMAX=DELTA
DELTAYMIN=DELTA
DELTAZMAX=DELTA
DELTAZMIN=DELTA

```

```

WRITE(MR,2040)XX,DELTAXMAX,DELTAXMIN

```

```

WRITE(MR,2040)ZZ,DELTAZMAX,DELTAZMIN

```

```

WRITE(MR,2040)YY,DELTAYMAX,DELTAYMIN

```

```

25  CONTINUE

```

```

1100  FORMAT(I4,4F10.4)

```

```

C-NOMBRE DE PLANS

```

```

NP=1
WRITE(MR,2030)NP

```

```

C-CONNECTIVITE

```

```

C-NOMBRE D'ELEMENTS POUR LE PLAN

```

```

WRITE(MR,2050)NT

```

```

C-TYPE ET NOEUDS DE L'ELEMENT

```

```

NTYPE=1

```

WRITE(MR,2050)NTYPE

WRITE(MR,2050)I

WRITE(MR,2050)I+1

27 CONTINUE

I=1

WRITE(MR,2050)NTYPE

WRITE(MR,2050)NT

WRITE(MR,2050)I

C-VECTEUR DE VISION

IVIEW=2

B1=0.0

B2=0.0

B3=0.0

A1=0.0

A2=2.0

A3=0.0

WRITE(MR,2060)IVIEW,B1,B2,B3,A1,A2,A3

C FORMATS D'ECRITURE

2030 FORMAT(2I5)

2040 FORMAT(3F10.4)

2050 FORMAT(I5)

2060 FORMAT(I5,6F10.4)

STOP

C ----ERREUR LORS DE LA LECTURE DU FICHIER DE DONNEES

100 WRITE(MP,1050)

C FORMAT D'ERREUR

1050 FORMAT(/," ERREUR LORS DE LA LECTURE DES DONNEES",

//," LE FORMAT DOIT ETRE LE SUIVANT :",

1 /," N (I5)",

2 /," X Y (2F10.5)",

3 /,"",

4 /," XN YN (2F10.5)",//)

STOP

END

SUBROUTINE FIT2(X,Y,NDATA,A,B,SIGA,SIGB,CHI2,Q)

```

C =====
C Etant donne NDATA points X(I),Y(I)
C ce sous-programme trouve l'equation de la droite
C Y=BX+1 en minimisant les carres.
C
C REFERENCE : NUMERICALS RECIPES P. 508
C             ADAPTE PAR REJEAN FOURNIER 4/3/88
C
C MODIFIE LE 12/03/92 POUR ELIMINER LES DIVISIONS PAR 0
C QUAND: 1- PENTE INFINIE, PENTE=9999999.99 ET A=X
C         2- NDATA=2 ALORS LES SIGMA NE SONT PAS VALIDES
C =====

```

DIMENSION X(NDATA),Y(NDATA)

SX=0.

SY=0.

ST2=0.

B=0.

DO 12 I=1,NDATA

 SX=SX+X(I)

 SY=SY+Y(I)

CONTINUE

SS=FLOAT(NDATA)

SXOSS=SX/SS

DO 14 I=1,NDATA

 T=X(I)-SXOSS

 ST2=ST2+T*T

 B=B+T*Y(I)

CONTINUE

WRITE(6,*)ST2,SS

IF (ST2.NE.0.0) THEN

 B=B/ST2

 A=(SY-SX*B)/SS

 SIGA=SQRT((1.+SX*SX/(SS*ST2))/SS)

 SIGB=SQRT(1./ST2)

ELSE

 B=9999999.9

 A=X(1)

 SIGA=9999999.99

 SIGB=9999999.99

ENDIF

CHI2=0.

DO 15 I=1,NDATA

 CHI2=CHI2+(Y(I)-A-B*X(I))**2

CONTINUE

Q=1.

IF (NDATA.GT.2) THEN

 SIGDAT=SQRT(CHI2/(NDATA-2))

ELSE

 SIGDAT=9999999.99

ENDIF

SIGA=SIGA*SIGDAT

SIGB=SIGB*SIGDAT

END

```
SUBROUTINE DIST(PENTE,CTE,X,Y,D)
```

```
=====
```

```
C  
C SOUS-PROGRAMME CALCULANT LA DISTANCE ENTRE UN POINT  
C X,Y ET UNE DROITE DEFINIE PAR SA PENTE ET UNE  
C CONSTANTE CTE.
```

```
C  
C AUTEUR : REJEAN FOURNIER  
C 8 MARS 1988  
C MODIFIE LE 12/03/92 POUR PENTE=0.0 ET PENTE=INFINIE (>9999999.0)
```

```
=====
```

```
C VARIABLES
```

```
C  
C PENTE : PENTE DE LA DROITE  
C CTE : CONSTANTE DE LA DROITE  
C X,Y : POINT QUELCONQUE  
C D : DISTANCE DU POINT A LA DROITE
```

```
=====
```

```
C  
C IF (PENTE.EQ.0.0) THEN  
C   D=ABS(Y-CTE)  
C ELSE IF (PENTE.GT.9999999.0) THEN  
C   D=ABS(X-CTE)  
C ELSE  
C   YA=PENTE*X+CTE  
C   XB=(Y-CTE)/PENTE  
C   CA=XB-X  
C   HYP2=YA-Y  
C   IF ((CA.EQ.0.0).OR.(HYP2.EQ.0.0)) THEN  
C     D=0.0  
C   ELSE  
C     D=ABS((CA*HYP2)/SQRT(CA**2.0+HYP2**2.0))  
C   ENDIF  
C ENDIF  
C  
C RETURN  
C ENF
```

```
      SUBROUTINE INTER(PENTE1,CTE1,PENTE2,CTE2,X,Y)
```

```
=====
```

```
C SOUS-PROGRAMME CALCULANT L'INTERSECTION DE DEUX  
C DROITES DEFINIES PAR UNE PENTE ET UNE  
C CONSTANCE CTE.
```

```
C AUTEUR :REJEAN FOURNIER  
C        25 AVRIL 1988
```

```
=====
```

```
C VARIABLES
```

```
C PENTE1 : PENTE DE LA DROITE 1  
C CTE1   : CONSTANCE DE LA DROITE 1  
C PENTE2 : PENTE DE LA DROITE 2  
C CTE2   : CONSTANCE DE LA DROITE 2  
C X,Y    : POINT D'INTERSECTION ENTRE LES DEUX DROITES
```

```
=====
```

```
      IF (PENTE1.EQ.PENTE2) THEN  
        RETURN  
      ENDIF
```

```
      X=(CTE2-CTE1)/(PENTE1-PENTE2)  
      Y=(CTE1*PENTE2-CTE2*PENTE1)/(PENTE2-PENTE1)
```

```
      RETURN  
      END
```

PROGRAM GEO

PROGRAMME PRINCIPAL DE GEOMETRIE VARIATIONNELLE

CE PROGRAMME UTILISE UN MODELE VARIATIONNEL POUR L'INSPECTION DES DIMENSIONS A L'AIDE DE LA VISION PAR ORDINATEUR. VOIR LA DESCRIPTION DES SOUS-PROGRAMMES POUR L'EXPLICATION DES DIFFERENTES FONCTIONS DU PROGRAMME.

AUTEUR : REJEAN FOURNIER

DATE : MARS 1992

MODIFIE LE:

RAISON :

SOUS-PROGRAMMES APPELES :

- 1 LIFESHP : LECTURE D'UN FICHIER MODELE DE TOPOLOGIE ".SHP"
- 2 LIRESCH : LECTURE D'UN FICHIER MODELE DE CONTRAINTES ".SCH"
- 3 LIREPAR : LECTURE D'UN FICHIER MODELE DES PARAMETRES ".PAR"
- 4 LIRETOL : LECTURE D'UN FICHIER MODELE DES TOLERANCES ".TOL"
- 5 LIREVIS : LECTURE D'UN FICHIER DE VISION ".VIS" (TOPOLOGIE, DL ET PRECISION). CES DONNEES PROVIENNENT DES IMAGES TRAITES PAR HISTO.FOR ET POLY.FOR.
- 6 RESOL : SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DES DL). C'EST LA RESOLUTION DU SYSTEME D'EQUATIONS DU MODELE.
- 7 VARI_TOPO : VARIATIONS SUR LA TOPOLOGIE (COMPARAISON MODELE-REEL). C'EST LA VERIFICATION SI LA TOPOLOGIE DU MODELE CORRESPOND BIEN A LA TOPOLOGIE REELLE.
- 8 PARA : CALCUL DES PARAMETRES REELS. LES PARAMETRES REELS SONT CALCULES EN ENTRANT LE DL REELS DANS LES EQUATIONS DE CONTRAINTE DU MODELE.
- 9 PRECI : CALCUL DE LA PRECISION SUR LES PARAMETRES REELS.
- 10 VARI_PARA : CALCUL DES VARIATIONS SUR LES PARAMETRES. IL S'AGIT DU CALCUL DE LA DIFFERENCE ENTRE LES PARAMETRES DU MODELE ET LES PARAMETRES REELS.
- 11 TOL : VERIFICATION DU RESPECT DES TOLERANCES DU MODELE PAR LES PARAMETRES REELS.

ON ENTEND ICI PAR DONNEES REELLES, LES DONNEES PROVENANT DE LA VISION (PAR OPPOSITION AUX DONNEES DU MODELE).

FICHIERS UTILISES:

- *.SHP : FICHIER DECRIVANT LE MODELE TOPOLOGIQUE DE L'OBJET ("SHAPE")
- *.SCH : FICHIER DECRIVANT L'ENSEMBLE DES CONTRAINTES SUR LA TOPOLOGIE ("SCHEME"). IL PEUT Y AVOIR PLUSIEURS SCHEMES POSSIBLES POUR LA MEME TOPOLOGIE.
- *.PAR : FICHIER DES PARAMETRES (IL Y A UN PARAMETRE PAR CONTRAINTE)
- *.TOL : FICHIER DES TOLERANCES (IL Y A UNE TOLERANCE MAX ET UNE TOLERANCE MIN PAR PARAMETRE)

LES QUATRES FICHIERS PRECEDANTS CONSTITUENT LE MODELE VARIATIONNEL

*.VIS : FICHIER DE LA TOPOLOGIE, DES DL ET DE LA PRECISION SUR LES
DL (DONNEES REELLES PROVENANT DE LA VISION).
CE FICHIER A ETE GENERE PAR POLY.

=====

VARIABLES UTILISEES PAR GED:

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
CORRESPONDANT A ETE EXECUTE (VRAI) OU NON (FAUX)
(EX: SI COND()=.TRUE. LE 2eme SOUS PROGRAMME A ETE
EXECUTE)
ICOM : NUMERO DE LA COMMANDE
NCOM : NOMBRE DE COMMANDES
MR : NUMERO DU L'UNITE DE LECTURE
MP : NUMERO DU L'UNITE D'ECRITURE

VARIABLES LUES PAR LIRESHP:

NN1 : NOMBRE DE NOEUDS DU MODELE
NN1 DOIT ETRE EGAL LA MOITIE DU NOMBRE DE DL EN 2D
NE1 : NOMBRE D'ELEMENTS DU MODELE
TE1() : TYPE D'ELEMENT
N1() : NUMEROS DES NOEUDS DE LA TOPOLOGIE

VARIABLES LUES PAR LIRESCH:

TCC() : TYPE DE CONTRAINTE
DLC() : DL CONTRAINTS

VARIABLES LUES PAR LIREPAR:

P1() : PARAMETRES DE CHAQUE CONTRAINTE

VARIABLES LUES PAR LIRETOL:

TOL() : TOLERANCES MAX ET MIN SUR CHAQUE PARAMETRE

VARIABLES LUES PAR LIREVIS:

NN2 : NOMBRE DE NOEUDS DONNE PAR LA VISION
NE2 : NOMBRE D'ELEMENTS DONNE PAR LA VISION
TE2() : TYPE D'ELEMENT ET POINTEUR DE N2()
N2() : NUMEROS DES NOEUDS DE LA TOPOLOGIE
X2() : COORDONNEES (DL) ET PRECISION DE CHAQUE
NOEUD DANS L'ORDRE SUIVANT: X, DELTAXMAX, DELTAXMIN,
Y, DELTAYMAX, DELTAYMIN, Z, DELTAZMAX, DELTAZMIN

VARIABLES CALCULEES PAR RESCL:

X1() : COORDONNEES (DL) DE CHAQUE NOEUD DU MODELE

VARIABLES CALCULEES PAR VARI_TOPD:

NTCJR : NOMBRE DE TOPOLOGIES CORRESPONDANTES
EODR(): LISTE DES ELEMENTS POUR CHACUNE DES TOPOLOGIES
CORRESPONDANTES

```

C
C      P2() : PARAMETRES REELS
C
C      VARIABLES CALCULEES PAR PRECI:
C
C      PRE() : PRECISION MAX ET MIN SUR LES PARAMETRES REELS
C
C      VARIABLES CALCULEES PAR VARI_PARA:
C
C      VP() : VARIATIONS SUR LES PARAMETRES
C
C      VARIABLES CALCULEES PAR TOLE:
C
C      AUCUNE
C
C=====
C      IMPLICIT NONE
C      LOGICAL COND(20)
C      INTEGER ICDM,NCDM,MR,MP
C      INTEGER NN1,NE1,TE1(500),N1(500),TC(500),DL(500)
C      REAL P1(500),TOL(500),X2(500)
C      INTEGER NN2,NE2,TE2(500),N2(500)
C      INTEGER NTCGR,ECGR(500)
C      REAL P2(500),PRE(500),VP(500)
C
C      DATA NCDM/11/,MR/5/,MP/6/
C
C-----INITIALISATION
C
C      DO 5 ICDM=1,NCDM
C         COND(ICDM) =.FALSE.
C      5 CONTINUE
C
C-----LECTURE DE LA COMMANDE DESIREE (CHOIX DU MENU)
C
C      10 WRITE(MP,1000)
C         WRITE(MP,1010)
C         READ(MR,2000,BRR=10)ICDM
C
C-----VERIFICATION SI LA COMMANDE EST POSSIBLE
C
C      IF ((ICDM.LT.1).OR.(ICDM.GT.NCDM+1)) GOTO 10
C
C-----APPEL DU SOUS-PROGRAMME CORRESPONDANT AU CHOIX
C
C      GOTO (101,102,103,104,105,106,107,108,109,110,111,500),ICDM
C
C---LECTURE D'UN FICHIER MODELE DE TOPOLOGIE ".SHP"
C
C      101 CALL LIRESHP(COND,NN1,NE1,TE1,N1)
C         GOTO 10
C
C---LECTURE D'UN FICHIER MODELE DE CONTRAINTES ".SCH"
C
C      102 CALL LIRESCH(COND,NN1,TC,DL)
C         GOTO 10
C
C---LECTURE D'UN FICHIER MODELE DES PARAMETRES DES CONTRAINTES ".PAR"
C
C      103 CALL LIREPAR(COND,NN1,P1)

```

```

C
C---LECTURE D'UN FICHIER MODELE DES TOLERANCES SUR LES PARAMETRES ".TGL"
C
  104 CALL LIRETOL(COND,NN1,TOL)
      GOTD 10
C
C---LECTURE D'UN FICHIER DE VISION ".VIS" (TOPOLOGIE,
C   DL ET PRECISION). CES DONNEES PROVIENNENT DES
C   IMAGES TRAITES PAR HISTO.FOR ET POLY.FOR.
C
  105 CALL LIRFVIS(COND,NN2,NE2,TE2,N2,X2)
      GOTD 10
C
C---SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DFS DL).
C   C'EST LA RESOLUTION DU SYSTEME D'EQUATIONS DU MODELE.
C
  106 CALL RESDL(COND,NN1,NE1,TE1,N1,TC,DL,P1)
      GOTD 10
C
C---VARIATIONS SUR LA TOPOLOGIE (COMPARAISON MODELE-REEL).
C   C'EST LA VERIFICATION SI LA TOPOLOGIE DU MODELE
C   CORRESPOND BIEN A LA TOPOLOGIE REELLE.
C
  107 CALL VARI_TOPO(COND,NN1,NE1,TE1,N1,NN2,NE2,TE2,N2,NTCOR,ECOR)
      GOTD 10
C
C---CALCUL DES PARAMETRES REELS.
C   LES PARAMETRES REELS SONT CALCULES EN ENTRANT LE DL
C   REELS DANS LES EQUATIONS DE CONTRAINTE DU MODELE.
C
  108 CALL PARA(COND,NN1,TC,DL,X2,P2,NTCOR,ECOR)
      GOTD 10
C
C---CALCUL DE LA PRECISION SUR LES PARAMETRES REELS.
C
  109 CALL PPECI(COND,NN1,TC,DL,X2,PRE,NTCOR,ECOR,P2)
      GOTD 10
C
C---CALCUL DES VARIATIONS SUR LES PARAMETRES. IL S'AGIT
C   DU CALCUL DE LA DIFFERENCE ENTRE LES PARAMETRES DU
C   MODELE ET LES PARAMETRES REELS.
C
  110 CALL VARI_PARA(COND,NN1,P1,P2,VP,NTCOR,ECOR)
      GOTD 10
C
C---VERIFICATION DU RESPECT DES TOLERANCES DU MODELE
C   PAR LES PARAMETRES REELS.
C
  111 CALL TOLE(COND,NN1,P1,VP,TOL,PRE,NTCOR,ECOR)
      GOTD 10
C
C---FIN DU PROGRAMME
C
  500 STOP
C
C-----FORMATS D'ENTREE/SORTIE.
C
  1000 FORMAT(/,' 1 : LIRE UN FICHIER MODELE DE TOPOLOGIE ".SHP" ',/,
    1 ' 2 : LIRE UN FICHIER MODELE DE CONTRAINTES ".SCH" ',/,
    1 ' 3 : LIRE UN FICHIER MODELE DE PARAMETRES ".PAR" ',/,
    . . .

```

```
1 * 5 : LIRE UN FICHIER DE VISION .VIS (TOPOLOGIE, DL *  
2 * ET PRECISION)*,/,  
1 * 6 : SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DES DL*)  
1010 FORMAT( 7 : VARIATIONS SUR LA TOPOLOGIE (COMPARAISON *  
2 * MODELE-REEL)*,/,  
1 * 8 : CALCUL DES PARAMETRES REELS (DL REELS DANS MODELE *  
2 * DE CONTRAINTES)*,/,  
1 * 9 : CALCUL DE LA PRECISION SUR LES PARAMETRES REELS*,/,  
1 * 10: VARIATIONS SUR LES PARAMETRES (COMPARAISON MODELE *  
2 * -REEL)*,/,  
1 * 11: VERIFICATION DU RESPECT DES TOLERANCES SUR LES *  
2 * PARAMETRES*,/,  
1 * 12: FIN*,//.  
1 * DONNER LE NUMERO DE COMMANDE CHOISIE : *,s)  
2000 FORMAT(I2)  
  
END
```

SUBROUTINE LIRESHP(COND,NN1,NE1,TE1,N1)

=====

SOUS-PROGRAMME DE LECTURE D'UN FICHIER "SHAP", FICHIER
CONTENANT LE MODELE TOPOLOGIQUE DE L'OBJET

=====

AUTEUR : REJEAN FOURNIER
DATE : MARS 1992

MODIFIE LE:
RAISON :

=====

SOUS-PROGRAMMES APPELES:

=====

FICHIERS UTILISES:

*.SHP : FICHIER DECRIVANT LE MODELE TOPOLOGIQUE DE L'OBJET
("SHAP")

=====

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)

NN1 : NOMBRE DE NOEUDS DU MODELE

NE1 : NOMBRE D'ELEMENTS DU MODELE

TE1(): TYPE D'ELEMENT ET POINTEUR DE N1() DU MODELE

N1() : NUMEROS DES NOEUDS DE LA TOPOLOGIE DU MODELE

MR : NUMERO DU L'UNITE DE LECTURE

MP : NUMERO DE L'UNITE D'ECRITURE

NOM : NOM DU FICHIER DE DONNEES

INOM : LONGUEUR DE NOM

NOM2 : NOM DU FICHIER DE DONNEES ENTRE PAR L'USAGER

INOM2 : LONGUEUR DE NOM2

NP : NOMBRE DE PLANS (1 CAR ON EST EN 2D)

IP : NUMERO DU PLAN

IN : NUMERO DU NOEUD

IE : NUMERO DE L'ELEMENT

=====

IMPLICIT NONE

LOGICAL COND(1)

INTEGER NN1,NE1,TE1(1),N1(1)

CHARACTER*20 NOM,NOM2

INTEGER INOM,INOM2

INTEGER NP,IP,IN,IE

INTEGER MR,MP

COMMON /NOM/NOM,INOM

```

C
C-----INITIALISATION :
C
C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION
C
C IL N'Y A PAS DE CONDITIONS PREALABLES POUR LIRESHP
C
C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0
      CONDC(2) =.FALSE.
      CONDC(3) =.FALSE.
      CONDC(4) =.FALSE.
      CONDC(6) =.FALSE.
      CONDC(7) =.FALSE.
      CONDC(8) =.FALSE.
      CONDC(9) =.FALSE.
      CONDC(10)=.FALSE.
      CONDC(11)=.FALSE.
C
C-----LECTURE DU NOM DU FICHIER ET OUVERTURE DE CELUI-CI
C
      NOM(INOM-2:INOM)='SHP'
      WRITE(MP,1000)NOM(1:INOM)
      READ(MR,1010)INOM2,NOM2
C-TEST SI UN NOM EST ENTRE PAR L'USAGER
      IF(INOM2.NE.0)THEN
        NOM=NOM2
        INOM=INOM2
C-TEST SI UN TYPE EST DONNE PAR L'USAGER
      IF (INDEX(NOM,'.').EQ.0)THEN
C-PAR DEFAUT, LE TYPE EST .SHP
        NOM(INOM+1:INOM+4)='.SHP'
        INOM=INOM+4
      ENDIF
      ENDIF
      OPEN(UNIT=1,FILE=NOM(1:INOM),STATUS='OLD',ERR=90)
C
C-----LECTURE DU NOMBRE DE NOEUDS
C
      READ(1,1020,ERR=90,END=90)NN1
      WRITE(MP,*)NN1
C
C-----LECTURE DU NOMBRE DE SURFACES OU PLANS (1 SI ON EST EN 2D)
C
      READ(1,1020,ERR=90,END=90)NP
      WRITE(MP,*)NP
      DO 10 IP=1,NP
C-----LECTURE DU NOMBRE D'ELEMENTS POUR LA SURFACE
      READ(1,1020,ERR=90,END=90)NE1
      WRITE(MP,*)NE1
      INITIALISATION DU POINTEUR DES NOEUDS
      IN=1
      DO 20 IS=1,NE1
C---LECTURE DU TYPE D'ELEMENT
        READ(1,1020,ERR=90,END=90)TE1(IE)
      WRITE(MP,*)TE1(IE)
C--BRANCHEMENT SELON LE TYPE D'ELEMENT
        IF (TE1(IE).EQ.1) THEN
C
          TYPE #1 : LINEAIRE 2 NOEUDS

```

```

READ(1,1020,ERR=90,END=90)N(IN+1)
WRITE(MP,*(N(IN)
WRITE(MP,*(N(IN+1)
MPITG(MP,*(N(IN+1)
IN=IN+2
ELSE IF(TE1(IE).EQ.2) THEN
TYPE #2 : 3 NOBUDS
READ(1,1020,ERR=90,END=90)N(IN)
READ(1,1020,ERR=90,END=90)N(IN+1)
READ(1,1020,ERR=90,END=90)N(IN+2)
IN=IN+3
ELSE IF(TE1(IE).EQ.3) THEN
TYPE #3 : 4 NOBUDS
READ(1,1020,ERR=90,END=90)N(IN)
READ(1,1020,ERR=90,END=90)N(IN+1)
READ(1,1020,ERR=90,END=90)N(IN+2)
READ(1,1020,ERR=90,END=90)N(IN+3)
IN=IN+4
ELSE
TYPE INCONNU
GOTO 90
ENDIF
20 CONTINUE
10 CONTINUE
C-----FIN NORMALE DE LA LECTURE DU FICHIER
COND(1)=.TRUE.
CLOSE(UNIT=1)
RETURN
C-----ERREUR LORS DE LA LECTURE DU FICHIER DE DONNEES
90 WRITE(MP,2000)
C-LE SOUS-PROGRAMME NE S'EST PAS EXECUTE CORRECTEMENT
COND(1) =.FALSE.
CLOSE(UNIT=1)
RETURN
C-----FORMATS
1000 FORMAT(/, ' DONNER LE NOM DU FICHIER A LIRE < ,A, > ,',)
1010 FORMAT(Q,M)
1020 FORMAT(5)
2000 FORMAT(' ERREUR LORS DE LA LECTURE DU FICHIER')
END

```

SUBROUTINE LIRESCH(COND,NN1,TC,DL)

=====

SOUS-PROGRAMME DE LECTURE D'UN FICHIER "SCHEME", FICHIER
CONTENANT L'ENSEMBLE DES CONTRAINTES GEOMETRIQUES DE LA PIECE

=====

AUTEUR : REJEAN FOURNIER
DATE : MARS 1992

MODIFIE LE:
RAISON :

=====

SOUS-PROGRAMMES APPELES:

=====

FICHIERS UTILISES:

SCHEM.SCH : FICHIER DECRIVANT L'ENSEMBLE DES CONTRAINTES SUR LA
TOPOLOGIE ("SCHEME").

=====

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)

NN1 : NOMBRE DE NOEUDS DU MODELE

TC() : TYPE DE CONTRAINTE ET POINTEUR DE DL()

DL() : DL CONTRAINTS

MR : NUMERO DE L'UNITE DE LECTURE

MP : NUMERO DE L'UNITE D'ECRITURE

NOM : NOM DU FICHIER DE DONNEES

INOM : LONGUEUR DE NOM

NOM2 : NOM DU FICHIER DE DONNEES ENTRE PAR L'USAGER

INOM2 : LONGUEUR DE NOM2

NC : NOMBRE DE CONTRAINTES DU MODELE

IC : NUMERO DE LA CONTRAINTE

IDL : NUMERO DU DL

NTYPE1: NOMBRE DE CONTRAINTES DE TYPE 1

NTYPE2: NOMBRE DE CONTRAINTES DE TYPE 2

=====

IMPLICIT NONE

LOGICAL COND(1)

INTEGER NN1,TC(1),DL(1)

INTEGER INOM,INOM2

INTEGER NC,IC,IDL

INTEGER MR,MP

INTEGER NTYPE1,NTYPE2

CHARACTER#20 NOM,NOM2

```

C
C      DATA MR/5/,MP/6/
C
C-----INITIALISATION :
C
C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION
C
C      IF (.NOT.COND(1)) THEN
C-      LA LECTURE DE LA TOPOLOGIE DU MODELE N'EST PAS FAITE
C      WRITE(MP,3000)
C      RETURN
C      ENDIF
C
C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0
C
C      COND(2) =.FALSE.
C      COND(4) =.FALSE.
C      COND(6) =.FALSE.
C      COND(8) =.FALSE.
C      COND(9) =.FALSE.
C      COND(10)=.FALSE.
C      COND(11)=.FALSE.
C
C-----LECTURE DU NOM DU FICHIER ET OUVERTURE DE CELUI-CI
C
C      NOM(INOM-2:INOM)='SCH'
C      WRITE(MP,1000)NOM(1:INOM)
C      READ(MR,1010)INOM2,NOM2
C-TEST SI UN NOM EST ENTRE PAR L'USAGER
C      IF(INOM2.NE.0)THEN
C      NOM=NOM2
C      INOM=INOM2
C-TEST SI UN TYPE EST DONNE PAR L'USAGER
C      IF (INDEX(NOM,'.')EQ.0)THEN
C- PAR DEFAUT, LE TYPE EST .SCH
C      NOM(INOM+1:INOM+4)='.SCH'
C      INOM=INOM+4
C      ENDIF
C      ENDIF
C      OPEN(UNIT=1,FILE=NOM(1:INOM),STATUS='OLD',ERR=90)
C
C-----LECTURE DU NOMBRE DE CONTRAINTES
C
C      READ(1,1020,ERR=90,END=90)NC
C
C-----VERIFICATION SI LE NOMBRE DE CONTRAINTES EST EGAL
C      AU NOMBRE DE OL (2 FOIS LE NOMBRE DE NOEUDS)
C      IF (NC.NE.(2*NN1)) THEN
C      WRITE(MP,4000)
C      GOTO 90
C      ENDIF
C
C-----LECTURE DU TYPE DE CONTRAINTE
C
C      IOL=1
C      NTYPE1=0
C      NTYPE2=0
C      DO 10 IC=1,NC
C      READ(1,1020,ERR=90,END=90)TC(IC)
C- BRANCHEMENT SELON LE TYPE DE CONTRAINTE

```

```
C
C-----TYPE #1 : POINT FIXE EN X
C
      READ(1,1020)DL(IDL)
      IDL=IDL+1
      NTYPE1=NTYPE1+1
      ELSE IF (TC(IC).EQ.2) THEN
C
C-----TYPE #2 : POINT FIXE EN Y
C
      READ(1,1020)DL(IDL)
      IDL=IDL+1
      NTYPE2=NTYPE2+1
      ELSE IF (TC(IC).EQ.3) THEN
C
C-----TYPE #3 : DISTANCE HORIZONTALE
C
      READ(1,1020)DL(IDL)
      READ(1,1020)DL(IDL+1)
      IDL=IDL+2
      ELSE IF (TC(IC).EQ.4) THEN
C
C-----TYPE #4 : DISTANCE VERTICALE
C
      READ(1,1020)DL(IDL)
      READ(1,1020)DL(IDL+1)
      IDL=IDL+2
      ELSE IF (TC(IC).EQ.5) THEN
C
C-----TYPE #5 : DISTANCE LINEAIRE
C
      READ(1,1020)DL(IDL)
      READ(1,1020)DL(IDL+1)
      READ(1,1020)DL(IDL+2)
      READ(1,1020)DL(IDL+3)
      IDL=IDL+4
      ELSE IF (TC(IC).EQ.6) THEN
C
C-----TYPE #6 : ANGLE ENTRE DEUX ELEMENTS DE TYPE 1
C
      READ(1,1020)DL(IDL)
      READ(1,1020)DL(IDL+1)
      READ(1,1020)DL(IDL+2)
      READ(1,1020)DL(IDL+3)
      READ(1,1020)DL(IDL+4)
      READ(1,1020)DL(IDL+5)
      READ(1,1020)DL(IDL+6)
      READ(1,1020)DL(IDL+7)
      IDL=IDL+8
      ELSE IF (TC(IC).EQ.7) THEN
C
C-----TYPE #7 : DISTANCE POINT-ELEMENT TYPE 1
C
      READ(1,1020)DL(IDL)
      READ(1,1020)DL(IDL+1)
      READ(1,1020)DL(IDL+2)
      READ(1,1020)DL(IDL+3)
      READ(1,1020)DL(IDL+4)
      READ(1,1020)DL(IDL+5)
      IDL=IDL+6
```

```

C
C-----TYPE INCONNU:
C
        WRITE(MP,5000)
        GOTO 90
    ENDIF
10 CONTINUE
C
C-----VERIFICATION DES TYPES DE CONTRAINTES
C
C-----LA SOMME NOMBRE DE CONTRAINTES DE TYPE 1 ET 2 DOIT
C     ETRE AU MOINS EGALE A 3 (TRANSLATION ET ROTATION NULLF)
C
    IF ((NTYPE1+NTYPE2).LT.3) THEN
        WRITE(MP,5000)
        GOTO 90
    ENDIF
C
C-----CHAQUE DL DOIT APPARAITRE DANS AU MOINS UNE CONTRAINTE
C
C-----FIN NORMALE DE LA LECTURE DU FICHIER
C
    COND(2)=.TRUE.
    CLOSE(UNIT=1)
    WRITE(6,*)TC(1),TC(2),TC(3),TC(4)
    WRITE(6,*)DL(1),DL(2),DL(3),DL(4),DL(5),DL(6),DL(7)
    RETURN
C
C-----ERREUR LORS DE LA LECTURE DU FICHIER DE DONNEES
C
    90 WRITE(6,2000)
C-LE SOUS-PROGRAMME NE S'EST PAS EXECUTE CORRECTEMENT
    COND(2)=.FALSE.
    CLOSE(UNIT=1)
    RETURN
C
C-----FORMATS
C
1000 FORMAT(/,' DONNER LE NOM DU FICHIER A LIRE (<',A,'> ',3)
1010 FORMAT(Q,A)
1020 FORMAT(I5)
2000 FORMAT(' ERREUR LORS DE LA LECTURE DU FICHIER')
3000 FORMAT(' UN FICHIER MODELE DE TOPOLOGIE DOIT DABORD ETRE LU')
4000 FORMAT(' LE NOMBRE DE CONTRAINTES N'EST PAS EGAL AU NOMBRE',
1 ' DE DL')
5000 FORMAT(' LE TYPE DE CONTRAINTE LU N'EST PAS CONNU')
6000 FORMAT(' LA SOMME DES CONTRAINTES DE TYPE 1 ET 2 DOIT ETRE > 2')
C
    END

```

SUBROUTINE LIREPAR(COND,NN1,P)

SOUS-PROGRAMME DE LECTURE DES PARAMETRES DU MODELE

AUTEUR : REJEAN FOURNIER

DATE : 13 MARS 1992

MODIFIE LE:

RAISON :

SOUS-PROGRAMMES APPELES:

FICHIERS UTILISES:

*.PAR : FICHIER DES PARAMETRES. IL DOIT Y AVOIR UN PARAMETRE PAR CONTRAINTE.

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)

NN1 : NOMBRE DE NOEUDS DU MODELE

P() : PARAMETRE DE CHAQUE CONTRAINTE

MR : NUMERO DE L'UNITE DE LECTURE

MP : NUMERO DE L'UNITE D'ECRITURE

NOM : NOM DU FICHIER DE DONNEES

INOM : LONGUEUR DE NOM

NOM2 : NOM DU FICHIER DE DONNEES ENTRE PAR L'USAGER

INOM2 : LONGUEUR DE NOM2

NP : NOMBRE DE PARAMETRES

IP : NUMERO DU PARAMETRE

IMPLICIT NONE

LOGICAL COND(1)

REAL P(1)

INTEGER NN1

INTEGER INOM,INOM2

INTEGER NP,IP

INTEGER MR,MP

CHARACTER*20 NOM,NOM2

COMMON /NOM/NOM,INOM

DATA MR/5/,MP/6/

-----INITIALISATION :

```
0
C IF (.NOT.COND(2)) THEN
C- LA LECTURE DU FICHIER DES CONTRAINTES DOIT DABORD ETRE FAIT
  WRITE(MP,3000)
  RETURN
C- ENDIF

C
C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0
C
C COND(4) =.FALSE.
C COND(6) =.FALSE.
C COND(10)=.FALSE.
C COND(11)=.FALSE.

C
C-----LECTURE DU NOM DU FICHIER ET OUVERTURE DE CELUI-CI
C
C NOM(INOM-2:INOM)=".PAR"
C WRITE(MP,1000)NOM(1:INOM)
C READ(MR,1010)INOM2,NOM2
C-TEST SI UN NOM EST ENTRE PAR L'USAGER
C IF(INOM2.NE.0)THEN
  NOM=NOM2
  INOM=INOM2
C-TEST SI UN TYPE EST DONNE PAR L'USAGER
C IF (INDEX(NOM, ".") .EQ. 0) THEN
C- PAR DEFAULT, LE TYPE EST .PAR
  NOM(INOM+1:INOM+4)=".PAR"
  INOM=INOM+4
C- ENDIF
C- ENDIF
C OPEN(UNIT=1, FILE=NOM(1:INOM), STATUS='OLD', ERR=90)

C
C---LECTURE DU NOMBRE DE PARAMETRES
C
C READ(1,1020, ERR=90, END=90)NP
C WRITE(MP,*)NP

C
C---VERIFICATION SI LE NOMBRE DE PARAMETRES EST EGAL AU NOMBRE
C DE CONTRAINTES
C
C IF (MP.NE.(2*NN1)) THEN
  WRITE(MP,2500)
  GO TO 90
C- ENDIF

C
C---LECTURE DES PARAMETRES
C
C DO 10 IP=1,NP
  READ(1,1040, ERP=90, END=90)P(IP)
C WRITE(MP,*)P(IP)
C 10 CONTINUE

C
C-----FIN NORMALE DE LA LECTURE DU FICHIER
C
C COND(3)=.TRUE.
C CLOSE(UNIT=1)
C RETURN

C
C-----ERREUR LORS DE LA LECTURE DU FICHIER DE DONNEES
C
```

C-LE SOUS-PROGRAMME NE S'EST PAS EXECUTE CORRECTEMENT

COND(3) =.FALSE.

CLOSE(UNIT=1)

RETURN

C

C-----FORMATS

C

1000 FORMAT(/, ' DONNER LE NOM DU FICHIER A LIRE < ', A, ' > ', \$)

1010 FORMAT(Q, A)

1020 FORMAT(I5)

1040 FORMAT(F10.4)

2000 FORMAT(' ERREUR LORS DE LA LECTURE DU FICHIER')

2500 FORMAT(' LE NOMBRE DE PARAMETRES DOIT ETRE EGAL',

1 ' AU NOMBRE DE CONTRAINTES')

3000 FORMAT(' UN FICHIER MODELE DE CONTRAINTES DOIT DABORD ETRE LU')

C

END

SUBROUTINE LIRETOL(COND, NN1, TCL)

SOUS-PROGRAMME DE LECTURE DES TOLERANCES

AUTEUR : REJEAN FOURNIER
 DATE : 13 MARS 1992

MODIFIE LE:
 RAISON :

SOUS-PROGRAMMES APPELES:

FICHIERS UTILISES:

*.TOL : FICHIER DES TOLERANCES. IL DOIT Y AVOIR UNE TOLERANCE
 MAXIMUM ET UNE TOLERANCE MINIMUM PAR PARAMETRE

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
 S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)

NN1 : NOMBRE DE NOEUDS DU MODELE
 TCL() : TOLERANCE MAX ET MIN SUR CHAQUE CONTRAINTE

NOM : NOM DU FICHIER DE DONNEES
 INOM : LONGUEUR DE NOM
 NOM2 : NOM DU FICHIER DE DONNEES ENTRE PAR L'USAGER
 INOM2 : LONGUEUR DE NOM2

IMPLICIT NONE
 LOGICAL COND(1)
 REAL TCL(1)
 INTEGER NN1
 INTEGER NTCL, ITCL
 INTEGER INOM, INOM2
 INTEGER MR, MP, I, II
 CHARACTER*20 NOM, NOM2

COMMON /NOM/NOM, INOM

DATA MR/5/, MP/5/

-----INITIALISATION :

-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION

IF (.NOT.COND(3)) THEN
 LA LECTURE DU FICHIER DES PARAMETRES DOIT DABORD ETRE FAITE
 WRITE(MP, 3000)
 RETURN

```

      ENDIF
C
C-TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0
      COND(1)=.FALSE.
C
C---LECTURE DU NOM DU FICHIER ET OUVERTURE DE CELUI-CI
      NOM(INOM-2:INOM)='TOL'
      WRITE(MP,1000)NOM(1:INOM)
      READ(MR,1010)INOM2,NOM2
C-TEST SI UN NOM EST ENTRE PAR L'USAGER
      IF(INOM2.NE.0)THEN
          NOM=NOM2
          INOM=INOM2
C-TEST SI UN TYPE EST DONNE PAR L'USAGER
          IF (INDEX(NOM,'.').EQ.0)THEN
C-PAR DEFAUT, LE TYPE EST .TOL
          NOM(INOM+1:INOM+4)='.TOL'
          INOM=INOM+4
          ENDIF
          ENDIF
          OPEN(UNIT=1,FILE=NOM(1:INOM),STATUS='OLD',ERR=90)
C---LECTURE DU NOMBRE DE TOLERANCES
          READ(1,1020,ERR=90,END=90)NTOL
C          WRITE(MP,*)NTOL
C---VERIFICATION SI LE NOMBRE DE TOLERANCES EST EGAL AU NOMBRE
C DE CONTRAINTES
          IF (NTOL.NE.(2*NN1)) THEN
              WRITE(MP,2500)
              GOTO 90
          ENDIF
C---LECTURE DES TOLERANCES
          DO 10 ITOL=1,NTOL
              READ(1,1040,ERR=90,END=90)TOL(ITOL*2-1),TOL(ITOL*2-0)
C              WRITE(MP,*)TOL(ITOL*2-1),TOL(ITOL*2-0)
          10 CONTINUE
C
C-----FIN NORMALE DE LA LECTURE DU FICHIER
C
          COND(4)=.TRUE.
          CLOSE(UNIT=1)
          RETURN
C
C-----ERREUR LORS DE LA LECTURE DU FICHIER DE DONNEES
C
          90 WRITE(MP,2000)
C-LE SOUS-PROGRAMME NE S'EST PAS EXECUTE CORRECTEMENT
          COND(4) =.FALSE.
          CLOSE(UNIT=1)
          RETURN
C
C-----FORMATS
C
          1000 FORMAT(/,' DONNER LE NOM DU FICHIER A LIRE <','A,'> ','$)
          1010 FORMAT(Q,A)
          1020 FORMAT(I5)
          1040 FORMAT(2F10.4)
          2000 FORMAT(' ERREUR LORS DE LA LECTURE DU FICHIER')
          2500 FORMAT(' LE NOMBRE DE TOLERANCES DOIT ETRE EGAL',
          1 ' AU NOMBRE DE PARAMETRES')
          3000 FORMAT(' UN FICHIER MODELE DE PARAMETRES DOIT DABORD ETRE LU')

```

END

SUBROUTINE LIREVIS(COND, NN2, NE2, TE2, N2, X2)

=====
C
C SOUS-PROGRAMME DE LECTURE D'UN FICHIER DE VISION ".VIS" (TOPOLOGIE,
C DL ET PRECISION). CES DONNEES PROVIENNENT DES
C IMAGES TRAITÉES PAR HISTO.FOR ET POLY.FOR.

=====
C
C AUTEUR : REJEAN FOURNIER
C DATE : MARS 1992
C
C MODIFIE LE:
C RAISON :

=====
C
C SOUS-PROGRAMMES APPELES:

=====
C
C FICHIERS UTILISES:
C
C *.VIS : FICHIER DE LA TOPOLOGIE, DES DL ET DE LA PRECISION SUR LES
C DL. CE FICHIER CONTIENT LES DONNEES "REELLES" PROVENANT
C DE LA VISION (IMAGE TRAITÉE PAR HISTO ET POLY).

=====
C
C VARIABLES UTILISEES :
C
C COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
C S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)
C
C NN2 : NOMBRE DE NOEUDS DONNE PAR LA VISION
C
C NE2 : NOMBRE D'ELEMENTS DONNE PAR LA VISION
C
C TE2(): TYPE D'ELEMENT ET POINTEUR DE N2()
C
C N2(): NUMEROS DES NOEUDS DE LA TOPOLOGIE
C
C X2(): COORDONNEES (DL) ET PRECISION DE CHAQUE
C NOEUD DANS L'ORDRE SUIVANT: X, DELTAXMAX, DELTAXMIN,
C Y, DELTAYMAX, DELTAYMIN, Z, DELTAZMAX, DELTAZMIN
C
C
C MR : NUMERO DE L'UNITE DE LECTURE
C
C MP : NUMERO DE L'UNITE D'ECRITURE
C
C
C
C NOM : NOM DU FICHIER DE DONNEES
C
C INOM : LONGUEUR DE NOM
C
C NOM2 : NOM DU FICHIER DE DONNEES ENTRE PAR L'USAGER
C
C INOM2 : LONGUEUR DE NOM2
C
C
C
C
C IN : NUMERO DU NOEUD
C
C NP : NOMBRE DE PLANS
C
C IP : NUMERO DU PLAN
C
C TE : NUMERO DE L'ELEMENT

=====
IMPLICIT NONE
LOGICAL COND(1)
REAL X2(1),A1,A2,A3,B1,B2,B3
INTEGER NN2,NE2,TE2(500),N2(500)
INTEGER INOM,INOM2,IVIEW

```

INTEGER MR,MP
CHARACTER*20 NOM,NOM2

```

```

COMMON /NOM/NOM,INOM

```

```

DATA MR/5/,MP/6/

```

```

C-----INITIALISATION :

```

```

C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION

```

```

C-IL N'Y A PAS DE CONDITIONS PREALABLES POUR LIREVIS

```

```

C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0

```

```

COND(7) =.FALSE.
COND(8) =.FALSE.
COND(9) =.FALSE.
COND(10)=.FALSE.
COND(11)=.FALSE.

```

```

C-----LECTURE DU NOM DU FICHIER ET OUVERTURE DE CELUI-CI

```

```

NOM(INOM-2:INOM)='VIS'
WRITE(MP,1000)NOM(1:INOM)
READ(MR,1010)INOM2,NOM2

```

```

C-TEST SI UN NOM EST ENTRE PAR L'USAGER

```

```

IF(INOM2.NE.0)THEN
  NOM=NOM2
  INOM=INOM2

```

```

C-TEST SI UN TYPE EST DONNE PAR L'USAGER

```

```

IF (INDEX(NOM,'.').EQ.0)THEN

```

```

C-PAR DEFAULT, LE TYPE EST .VIS

```

```

  NOM(INOM+1:INOM+4)='.VIS'
  INOM=INOM+4

```

```

ENDIF

```

```

ENDIF

```

```

OPEN(JUNIT=1,FILE=NOM(1:INOM),STATUS='OLD',ERR=90)

```

```

C---LECTURE DU NOMBRE DE NOEUDS

```

```

READ(1,1020,ERR=90,END=90)NN2
WRITE(MP,#)NN2

```

```

C---LECTURE DES DL DE CHAQUE NOEUDS DANS L'ORDRE:

```

```

X, DELTAXMAX, DELTAXMIN, Y, DELTAYMAX, DELTAYMIN, Z, DELTAZMAX, DELTAZMIN

```

```

DO 5 IN=1,NN2

```

```

  READ(1,1040,ERR=90,END=90)X2(IN*9-8),X2(IN*9-7),X2(IN*9-6)

```

```

  READ(1,1040,ERR=90,END=90)X2(IN*9-5),X2(IN*9-4),X2(IN*9-3)

```

```

  READ(1,1040,ERR=90,END=90)X2(IN*9-2),X2(IN*9-1),X2(IN*9-0)

```

```

  WRITE(MP,#)X2(IN*9-8),X2(IN*9-7),X2(IN*9-6)

```

```

  WRITE(MP,#)X2(IN*9-5),X2(IN*9-4),X2(IN*9-3)

```

```

  WRITE(MP,#)X2(IN*9-2),X2(IN*9-1),X2(IN*9-0)

```

```

5 CONTINUE

```

```

C-----LECTURE DU NOMBRE DE SURFACES DU PLANS (1 SI ON EST EN 2D)

```

```

READ(1,1020,ERR=90,END=90)NP

```

```

WRITE(MP,#)NP

```

```

C---LECTURE DU NOMBRE D'ELEMENTS POUR LA SURFACE
      READ(1,1020,ERR=90,END=90)NE2
C      WRITE(MP,*)NE2
C      INITIALISATION DU POINTEUR DES NOEUDS
      IN=1
      DO 10 IE=1,NE2
C---LECTURE DU TYPE D'ELEMENT
      READ(1,1020,ERR=90,END=90)TE2(IE)
C      WRITE(MP,*)TE2(IE)
C-BRANCHEMENT SELON LE TYPE D'ELEMENT
      IF (TE2(IE).EQ.1) THEN
-
C          TYPE #1 : 2 NOEUDS
C
          READ(1,1020,ERR=90,END=90)N2(IN)
          READ(1,1020,ERR=90,END=90)N2(IN+1)
C      WRITE(MP,*)N2(IN)
C      WRITE(MP,*)N2(IN+1)
          IN=IN+2
      ELSE IF(TE2(IE).EQ.2) THEN
C
          TYPE #2: 3 NOEUDS
C
          READ(1,1020,ERR=90,END=90)N2(IN)
          READ(1,1020,ERR=90,END=90)N2(IN+1)
          READ(1,1020,ERR=90,END=90)N2(IN+2)
          IN=IN+3
      ELSE IF(TE2(IE).EQ.3) THEN
C
          TYPE #3 : 4 NOEUDS
C
          READ(1,1020,ERR=90,END=90)N2(IN)
          READ(1,1020,ERR=90,END=90)N2(IN+1)
          READ(1,1020,ERR=90,END=90)N2(IN+2)
          READ(1,1020,ERR=90,END=90)N2(IN+3)
          IN=IN+4
      ELSE
C
          TYPE D'ELEMENT INCONNU
C
          WRITE(MP,3000)
          GOTO 90
      ENDIF
      10 CONTINUE
      10 CONTINUE
C-----READ THE KIND OF PROJECTION AND THE VIEWING DIRECTION
C
      READ(1,5000)IVIEW,B1,B2,B3,A1,A2,A3
      WRITE(MP,*)IVIEW,B1,B2,B3,A1,A2,A3
C-----FIN NORMALE DE LA LECTURE DU FICHIER
C
      COND(5)=.TRUE.
      CLOSE(UNIT=1)
      RETURN
C-----ERREUR LORS DE LA LECTURE DU FICHIER DE DONNEES
      90 WRITE(MP,2000)

```

```
COND(5) =.FALSE.  
CLOSE(UNIT=1)  
RETURN
```

```
C  
C-----FORMATS
```

```
C  
1000 FORMAT(/, ' DONNER LE NOM DU FICHIER A LIRE <',A,'> ',%)  
1010 FORMAT(Q,A)  
1020 FORMAT(I5)  
1040 FORMAT(3F10.4)  
2000 FORMAT(' ERREUR LORS DE LA LECTURE DU FICHIER')  
3000 FORMAT(' TYPE D''ELEMENT INCONNU')  
3000 FORMAT(I5,6F10.4)
```

```
C  
END
```

SUBROUTINE VARI_TOPO(COND,NN1,NE1,TE1,N1,NN2,NE2,TE2,N2,
NTCOR,ECOR)

=====

SOUS-PROGRAMME DE CALCUL DE VARIATION SUR LA TOPOLOGIE
(COMPARAISON MODELE-REEL).
C'EST LA VERIFICATION SI LA TOPOLOGIE DU MODELE
CORRESPOND BIEN A LA TOPOLOGIE REELLE.
ON ENTEND ICI PAR DONNEES REELLES, LES DONNEES PROVENANT DE LA
VISION (PAR OPPOSITION AUX DONNEES DU MODELE).

=====

AUTEUR : REJEAN FOURNIER
DATE : 13 MARS 1992
MODIFIE LE:
RAISON :

=====

SOUS-PROGRAMMES APPELES:

=====

FICHIERS UTILISES:

=====

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)
NN1 : NOMBRE DE NOEUDS DU MODELE
NN1 DOIT ETRE EGAL LA MOITIE DU NOMBRE DE DL EN 2D
NE1 : NOMBRE D'ELEMENTS DU MODELE
TE1() : TYPE D'ELEMENT ET POINTEUR DE N1()
N1() : NUMEROS DES NOEUDS DE LA TOPOLOGIE
NN2 : NOMBRE DE NOEUDS DONNE PAR LA VISION
NE2 : NOMBRE D'ELEMENTS DONNE PAR LA VISION
TE2() : TYPE D'ELEMENT ET POINTEUR DE N2()
N2() : NUMEROS DES NOEUDS DE LA TOPOLOGIE
NTCOR : NOMBRE DE TOPOLOGIES CORRESPONDANTES
NECOR : NOMBRE D'ELEMENTS CORRESPONDANTS POUR UNE COMPARAISON

MR : NUMERO DU FICHIER DE LECTURE
MP : NUMERO DU FICHIER D'ECRITURE

NE : NOMBRE D'ELEMENTS (=NE1=NE2)
IE1 : NUMERO DU PREMIER ELEMENT DE CORRESPONDANCE DU MODELE
IE2 : NUMERO DE L'ELEMENT DE CORRESPONDANCE DE LA VISION
IE3 : NUMERO DE L'ELEMENT DE CORRESPONDANCE DU MODELE
ECOR(): NUMERO DE L'ELEMENT DU MODELE CORRESPONDANT AU PREMIER
ELEMENT DE LA VISION POUR CHAQUE TOPOLOGIE CORRESPONDANTE

=====

IMPLICIT NONE
LOGICAL COND(1)

```

INTEGER NN2,NE2,TE2(1),N2(1)
INTEGER MR,MP
INTEGER NE,IE1,IF2,IF3,NTCOR,NECOR,ECOR(1)

```

```

C
DATA MR/5/,MP/6/

```

```

C
C-----INITIALISATION :

```

```

C
C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION

```

```

C
IF (.NOT.CONDC(1).AND..NOT.CONDC(5)) THEN
C- LA LECTURE DU FICHIER DE VISION ET DE TOPOLOGIE DU MODELE
C DOIT DABORD ETRE FAITE
WRITE(MP,3000)
RETURN

```

```

ELSE IF (.NOT.CONDC(1)) THEN
C- LA LECTURE DU FICHIER DE TOPOLOGIE DU MODELE
C DOIT DABORD ETRE FAITE
WRITE(MP,3010)
RETURN

```

```

ELSE IF (.NOT.CONDC(5)) THEN
C- LA LECTURE DU FICHIER DE VISION
C DOIT DABORD ETRE FAITE
WRITE(MP,3020)
RETURN

```

```

ENDIF

```

```

C
C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0

```

```

COND(8) =.FALSE.
COND(9) =.FALSE.
COND(10)=.FALSE.
COND(11)=.FALSE.

```

```

C
C-----VERIFICATION

```

```

C
C-----VERIFICATION DU NOMBRE DE NOEUDS

```

```

C
IF (NN1.NE.NN2) THEN
WRITE(MP,4000)NN1,NN2
GOTO 90
ENDIF

```

```

C
C-----VERIFICATION DU NOMBRE D'ELEMENTS

```

```

C
IF (NE1.NE.NE2) THEN
WRITE(MP,5000)NE1,NE2
GOTO 90
ENDIF

```

```

C- NE1=NE2=NE
NE=NE1

```

```

C
C-----VERIFICATION ET CORRESPONDANCE DES ELEMENTS

```

```

C
NTCOR=0
DO 10 IE1=1,NE
NECOR=0
DO 20 IE2=1,NE
IE3=IE1+IE2-1

```

```

      IE3=IE3-NE
      ENDIF
      IF (TE1(IE3).EQ.TE2(IE2)) THEN
        NECCR=NECCR+1
      ENDIF
20    CONTINUE
      IF (NECCR.EQ.NE) THEN
        NTCOR=NTCOR+1
        ECOR(NTCOR)=IE1
      ENDIF
10    CONTINUE
      IF (NTCOR.EQ.0) THEN
C-      AUCUNE CORRESPONDANCE TROUVEE
        WRITE(CMP,5000)NTCOR
        GO TO 90
      ENDIF
C-      UNE OU PLUSIEURS CORRESPONDANCES TROUVEES
        WRITE(CMP,5000)NTCOR
C
C-----FIN NORMALE
C
      COND(7)=.TRUE.
      RETURN
C-----FIN ANORMALE
C
      90 WRITE(CMP,5000)
        COND(7)=.FALSE.
        RETURN
C
C-----FORMATS
C
3000 FORMAT(' UN FICHIER MODELE DE TOPOLOGIE ET ',
1         /,' UN FICHIER DE VISION DOIVENT DABORD ETRE LUS')
3010 FORMAT(' UN FICHIER MODELE DE TOPOLOGIE DOIT DABORD ETRE LU')
3020 FORMAT(' UN FICHIER DE VISION DOIT DABORD ETRE LU')
4000 FORMAT(' LE NOMBRE DE NOEUDS DU MODELE (' ,I5,') NE CORRESPOND ',
1         /,' PAS AU NOMBRE DE NOEUDS DONNE PAR LA VISION(' ,I5,')')
5000 FORMAT(' LE NOMBRE D'ELEMENTS DU MODELE (' ,I5,') NE CORRESPOND ',
1         /,' PAS AU NOMBRE D'ELEMENTS DONNE PAR LA VISION(' ,I5,')')
6000 FORMAT(I5,' CORRESPONDANCE(S) TROUVEE(S) ENTRE LES 2 TOPOLOGIES')
9000 FORMAT(' RELIRE UN FICHIER MODELE DE TOPOLOGIE OU ',
1         /,' RELIRE UN FICHIER DE VISION')
C
      END

```

SUBROUTINE PARACOND,NN1,TC,DL,X2,P2,NTCOR,ECCR)

SOUS-PROGRAMME VERIFIANT LES CONTRAINTES DU MODELE AVEC LES DL
DONNES PAR LA VISION

AUTEUR : REJEAN FOURNIER
DATE : 13 MARS 1992

MODIFIE LE:
RAISON :

SOUS-PROGRAMMES APPELES:

FIXE
DIST
DISTL
ANGLE
DISTRE

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)

NN1 : NOMBRE DE NOEUDS DU MODELE
NN1 DOIT ETRE EGAL LA MOITIE DU NOMBRE DE DL EN 2D

TC() : TYPE DE CONTRAINTE ET POINTEUR DE EC()

DL() : DL CONTRAINTS

X2() : COORDONNEES (DL) ET PRECISION DE CHAQUE
NOEUD DANS L'ORDRE SUIVANT: X, DELTAXMAX, DELTAXMIN,
Y, DELTAYMAX, DELTAYMIN, Z, DELTAZMAX, DELTAZMIN

P2() : PARAMETRES CALCULE POUR CHAQUE CONTRAINTE

NTCOR : NOMBRE DE TOPOLOGIES CORRESPONDANTES

ECCR(): NUMERO DE L'ELEMENT DU MODELE CORRESPONDANT AU PREMIER
ELEMENT DE LA VISION POUR CHAQUE TOPOLOGIE CORRESPONDANTE

X3() : X2() AVEC ROTATION POUR CHAQUE TOPOLOGIE

ITCOR : NUMERO DE LA TOPOLOGIE CORRESPONDANTE

MR : NUMERO DU FICHIER DE LECTURE

MP : NUMERO DU FICHIER D'ECRITURE

NC : NOMBRE DE CONTRAINTES

TDL : NUMERO DU DL

IC : NUMERO DE LA CONTRAINTE

XF : POINT FIXE EN X

YF : POINT FIXE EN Y

DH : DISTANCE HORIZONTALE

DV : DISTANCE VERTICALE

DL : DISTANCE LINEAIRE

A : ANGLE

DP : DISTANCE NOEUD-ELEMENT

```

IMPLICIT NONE
LOGICAL COND(1)
INTEGER NN1,TC(1),DL(1),NTCOR,ECOR(1)
REAL X2(1),X3(500)
INTEGER IX,IXCOR,ITCOR
INTEGER MP
INTEGER NC,IDL,IC
REAL P2(1)
REAL XF,YF,DH,DV,DLIN,A,DP
REAL FIXE,DIST,DISTL,ANGLE,DISTPE

```

```

C
DATA MP/5/

```

```

C-----INITIALISATION :

```

```

C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION

```

```

C
IF (.NOT.COND(2).AND..NOT.COND(7)) THEN
C- LA LECTURE DU FICHIER DE CONTRAINTES DU MODELE
C DOIT DABORD ETRE FAITE
C ET LA VERIFICATION DE LA TOPOLOGIE DOIT ETRE FAITE
WRITE(MP,3000)
RETURN
ELSE IF (.NOT.COND(2)) THEN
C- LA LECTURE DU FICHIER DE CONTRAINTES DU MODELE
C DOIT DABORD ETRE FAITE
WRITE(MP,3010)
RETURN
ELSE IF (.NOT.COND(7)) THEN
C- LA VERIFICATION DE LA TOPOLOGIE DOIT ETRE FAITE
WRITE(MP,3020)
RETURN
ENDIF

```

```

C-----TOUS LES SOUS-PROGRAMMES DECULANT SONT REMIS A 0

```

```

COND(9) = .FALSE.
COND(10) = .FALSE.
COND(11) = .FALSE.

```

```

C---VERIFICATION

```

```

C-----BOUCLE POUR TOUTES LES TOPOLOGIES ET TOUTES LES CONTRAINTES

```

```

C
NC=NN1*2
DO 5 ITCOR=1,NTCOR
C ROTATION DE LA TOPOLOGIE (NE FONCTIONNE QUE POUR DES
C ELEMENTS DE TYPE 1 ET QUI SE SUIVENT)
DO 7 IX=1,NN1
IXCOR=IX+ECOR(ITCOR)-1
IF (IXCOR.GT.NN1) THEN
IXCOR=IXCOR-NN1
ENDIF
X3(IX*9-8)=X2(IXCOR*9-8)
X3(IX*9-7)=X2(IXCOR*9-7)
X3(IX*9-6)=X2(IXCOR*9-6)
X3(IX*9-5)=X2(IXCOR*9-5)
X3(IX*9-4)=X2(IXCOR*9-4)
X3(IX*9-3)=X2(IXCOR*9-3)
X3(IX*9-2)=X2(IXCOR*9-2)

```

X3(IX*9-0)=X2(IXCOR*9-0)

7 CONTINUE

WRITE(MP,1000)ITCOR

IDL=1

DO 10 IC=1,NC

C-BRANCHEMENT SELON LE TYPE DE CONTRAINTE

C
C---TYPE #1: POINT FIXE EN X

C
C
IF (TC(IC).EQ.1) THEN
CALL FIXE(X3(DL(IDL)*3-2),XF)
P2(IC+ITCOR*NC-NC)=XF
IDL=IDL+1

C
C---TYPE #2: POINT FIXE EN Y

C
C
ELSE IF (TC(IC).EQ.2) THEN
CALL FIXE(X3(DL(IDL)*3-2),YF)
P2(IC+ITCOR*NC-NC)=YF
IDL=IDL+1

C
C---TYPE #3: DISTANCE HORIZONTALE

C
C
ELSE IF (TC(IC).EQ.3) THEN
CALL DIST(X3(DL(IDL)*3-2),X3(DL(IDL+1)*3-2),DH)
P2(IC+ITCOR*NC-NC)=DH
IDL=IDL+2

C
C---TYPE #4: DISTANCE VERTICALE

C
C
ELSE IF (TC(IC).EQ.4) THEN
CALL DIST(X3(DL(IDL)*3-2),X3(DL(IDL+1)*3-2),DV)
P2(IC+ITCOR*NC-NC)=DV
IDL=IDL+2

C
C---TYPE #5: DISTANCE LINEAIRE ENTRE DEUX NOEUDS

C
C
ELSE IF (TC(IC).EQ.5) THEN
CALL DISTL(X3(DL(IDL+0)*3-2),X3(DL(IDL+1)*3-2),
1 X3(DL(IDL+2)*3-2),X3(DL(IDL+3)*3-2),DLIN)
1 P2(IC+ITCOR*NC-NC)=DLIN
1 IDL=IDL+4

C
C---TYPE #6: ANGLE ENTRE DEUX ELEMENTS LINEAIRES

C
C
ELSE IF (TC(IC).EQ.6) THEN
CALL ANGLE(X3(DL(IDL+0)*3-2),X3(DL(IDL+1)*3-2),
1 X3(DL(IDL+2)*3-2),X3(DL(IDL+3)*3-2),
1 X3(DL(IDL+4)*3-2),X3(DL(IDL+5)*3-2),
1 X3(DL(IDL+6)*3-2),X3(DL(IDL+7)*3-2),A)
P2(IC+ITCOR*NC-NC)=A
IDL=IDL+8

C
C---TYPE #7: DISTANCE NOEUD-ELEMENT LINEAIRE

C
C
ELSE IF (TC(IC).EQ.7) THEN
CALL DISTPE(X3(DL(IDL+0)*3-2),X3(DL(IDL+1)*3-2),
1 X3(DL(IDL+2)*3-2),X3(DL(IDL+3)*3-2),
1 X3(DL(IDL+4)*3-2),X3(DL(IDL+5)*3-2),DP)
P2(IC+ITCOR*NC-NC)=DP

```

      ENDIF
      WRITE(MP,1030)P2(IC+ITCOR*NC-NC)

```

```

10  CONTINUE
5   CONTINUE

```

```

C
C-----FIN NORMALE
C

```

```

      CONO(8)=.TRUE.
      RETURN

```

```

C-----FORMATS UTILISES
C

```

```

1000 FORMAT(' TOPOLOGIE #',I3)
1030 FORMAT(F10.4)
3000 FORMAT(' UN FICHIER MODELE DE CONTRAINTES DOIT ETRE LU ET ',
1      /, ' LA VARIATION SUR LA TOPOLOGIE DOIT ETRE VERIFIEE ')
3010 FORMAT(' UN FICHIER MODELE DE CONTRAINTES DOIT ETRE LU ')
3020 FORMAT(' LA VARIATION SUR LA TOPOLOGIE DOIT ETRE VERIFIEE ')

```

```

      END
      SUBROUTINE FIXE(X,F)

```

```

C=====
C
C SOUS-PROGRAMME DE POINT FIXE EN X OU Y
C

```

```

      IMPLICIT NONE
      REAL X,F

```

```

      F=X

```

```

      RETURN
      END

```

```

      SUBROUTINE DIST(X1,X2,D)

```

```

C=====
C
C SOUS-PROGRAMME DE CALCUL D'UNE DISTANCE UNE DIMENSION
C
C LA DISTANCE D EST LA VALEUR ABSOLUE DE LA DIFFERENCE X2-X1
C

```

```

      IMPLICIT NONE
      REAL X1,X2,D

```

```

      D=ABS(X2-X1)

```

```

      RETURN
      END

```

```

      SUBROUTINE DISTL(X1,Y1,X2,Y2,D)

```

```

C=====
C
C SOUS-PROGRAMME DE CALCUL D'UNE DISTANCE LINEAIRE (DEUX DIMENSIONS)
C
C 1er POINT : (X1,Y1)
C 2e POINT : (X2,Y2)
C LA DISTANCE D OBTENUE EST TOUJOURS POSITIVE
C

```

```

      IMPLICIT NONE
      REAL X1,Y1,X2,Y2,D

```

```
D=SQRT((X2-X1)**2.0+(Y2-Y1)**2.0)
```

```
RETURN
END
```

```
SUBROUTINE ANGLE(X1,Y1,X2,Y2,X3,Y3,X4,Y4,A)
```

```
=====
```

```
C
C SOUS-PROGRAMME DE CALCUL D'UN ANGLE ENTRE DEUX DROITES (EN DEGRES)
```

```
C 1ere DROITE : (X1,Y1)-(X2,Y2)
```

```
C 2eme DROITE : (X3,Y3)-(X4,Y4)
```

```
C L'ANGLE A OBTENU EST ENTRE LES DEUX VECTEURS DANS LE MEME SENS ET
C EST TOUJOURS POSITIF
```

```
=====
```

```
IMPLICIT NONE
```

```
REAL X1,Y1,X2,Y2,X3,Y3,X4,Y4,A,PS,PV
```

```
C---PRODUIT SCALAIRE PS
```

```
PS=(X2-X1)*(X4-X3)+(Y2-Y1)*(Y4-Y3)
```

```
C---PRODUIT VECTORIEL PV
```

```
PV=(X2-X1)*(Y4-Y3)-(Y2-Y1)*(X4-X3)
```

```
C---ANGLE EN DEGRES (VALEUR ABSOLUE)
```

```
IF ((PS.EQ.0.0).AND.(PV.EQ.0.0)) THEN
```

```
  A=0.0
```

```
ELSE
```

```
  A=ABS(ATAN2D(PV,PS))
```

```
ENDIF
```

```
RETURN
```

```
END
```

```
SUBROUTINE DISTPE(X1,Y1,X2,Y2,X3,Y3,D)
```

```
=====
```

```
C
C SOUS-PROGRAMME DE CALCUL D'UNE DISTANCE ENTRE UN POINT ET UNE LIGNE
```

```
C POINT: (X3,Y3)
```

```
C LIGNE: (X1,Y1)-(X2,Y2)
```

```
C LA DISTANCE D OBTENUE EST TOUJOURS POSITIVE
```

```
=====
```

```
IMPLICIT NONE
```

```
REAL X1,Y1,X2,Y2,X3,Y3,D,A
```

```
CALL ANGLE(X2,Y2,X1,Y1,X2,Y2,X3,Y3,A)
```

```
CALL DISTL(X2,Y2,X3,Y3,D)
```

```
D=D*SIND(A)
```

```
RETURN
```

```
END
```

SUBROUTINE PRECI(COND,NN1,TC,DL,X2,PRE,NTCOR,ECOR,P2)

SOUS-PROGRAMME DE CALCUL DE LA PRECISION SUR LES PARAMETRES REELS

AUTEUR : REJEAN FOURNIER
 DATE : 13 MARS 1992

MODIFIE LE:
 RAISON :

SOUS-PROGRAMMES APPELES:

- FIXE : CALCULE UN DL FIXE EN X OU Y
- DIST : CALCULE UNE DISTANCE HORIZ. OU VERT.
- DISTL : CALCULE UNE DISTANCE LINEAIRE
- ANGLE : CALCULE UN ANGLE ENTRE DEUX ELEMENTS TYPE 1
- DISTPE: CALCULE UNE DISTANCE NOEUD-ELEMENT TYPE 1

FICHIERS UTILISES:

VARIABLES UTILISEES :

- COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)
- NN1 : NOMBRE DE NOEUDS DU MODELE
 NN1 DOIT ETRE EGAL LA MOITIE DU NOMBRE DE DL EN 2D
- TC() : TYPE DE CONTRAINTE ET POINTEUR DE EC()
- DL() : DL CONTRAINTS
- X2() : COORDONNEES (DL) ET PRECISION DE CHAQUE NOEUD DANS L'ORDRE SUIVANT: X, DELTAXMAX, DELTAXMIN, Y, DELTAYMAX, DELTAYMIN, Z, DELTAZMAX, DELTAZMIN
- PRE() : PRECISION CALCULEE POUR CHAQUE PARAMETRE
- NTCOR : NOMBRE DE TOPOLOGIES CORRESPONDANTES
- ECOR(): NUMERO DE L'ELEMENT DU MODELE CORRESPONDANT AU PREMIER ELEMENT DE LA VISION POUR CHAQUE TOPOLOGIE CORRESPONDANTE
- P2() : PARAMETRES CALCULES POUR CHAQUE CONTRAINTE
- MR : NUMERO DU FICHIER DE LECTURE
- MP : NUMERO DU FICHIER D'ECRITURE
- X3() : X2() AVEC ROTATION POUR CHAQUE TOPOLOGIE
- ITCOR : NUMERO DE LA TOPOLOGIE CORRESPONDANTE
- NC : NOMBRE DE CONTRAINTES
- IDL : NUMERO DU DL
- IC : NUMERO DE LA CONTRAINTE

POUR LES VARIABLES SUIVANTES, ON A UN MAX ET UN MIN:

- XF : POINT FIXE EN X
- YF : POINT FIXE EN Y

```

C      DV      : DISTANCE VERTICALE
C      DLIN    : DISTANCE LINFAIRE
C      A       : ANGLE
C      DP      : DISTANCE NOEUD-ELEMENT
C
C      IXn,IYn : COMPTEURS

```

```

C=====

```

```

IMPLICIT NONE
LOGICAL COND(1)
INTEGER NN1,TC(1),DL(1),NTCOR,ECOR(1)
REAL X2(1),X3(500),P2(1)
INTEGER IX,IXCOR,ITCOR
INTEGER MP
INTEGER NC,IDL,IC
REAL PRE(1)
REAL XFMAX,YFMAX,DHMAX,DVMAX,DLINMAX,AMAX,DPMAX
REAL XFMIN,YFMIN,DHMIN,DVMIN,DLINMIN,AMIN,DPMIN
REAL FIXE,DIST,DISTL,ANGLE,DISTPE
INTEGER IX1,IY1,IX2,IY2
INTEGER IX3,IY3,IX4,IY4
REAL A,D

```

```

DATA MP/6/

```

```

C-----INITIALISATION :

```

```

C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION

```

```

IF (.NOT.COND(8)) THEN
  LE CALCUL DES PARAMETRES DOIT DABORD ETRE FAIT
  WRITE(MP,3000)
  RETURN
ENDIF

```

```

C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0

```

```

COND(10)=.FALSE.
COND(11)=.FALSE.

```

```

C---VERIFICATION

```

```

C-----BOUCLE POUR TOUTES LES CONTRAINTES

```

```

NC=NN1*2
DO 5 ITCOR=1,NTCOR
  ROTATION DE LA TOPOLOGIE
  DO 7 IX=1,NN1
    IXCOR=IX+ECOR(ITCOR)-1
    IF (IXCOR.GT.NN1) THEN
      IXCOR=IXCOR-NN1
    ENDIF
    X3(IX*9-8)=X2(IXCOR*9-8)
    X3(IX*9-7)=X2(IXCOR*9-7)
    X3(IX*9-6)=X2(IXCOR*9-6)
    X3(IX*9-5)=X2(IXCOR*9-5)
    X3(IX*9-4)=X2(IXCOR*9-4)
    X3(IX*9-3)=X2(IXCOR*9-3)
    X3(IX*9-2)=X2(IXCOR*9-2)
    X3(IX*9-1)=X2(IXCOR*9-1)
  
```

```

7  CONTINUE
  WRITE(MP,1000)ITCOR
  IDL=1
  DO 10 IC=1,NC

```

```

C-BRANCHEMENT SELON LE TYPE DE CONTRAINTE
C

```

```

C---TYPE #1: POINT FIXE EN X
C

```

```

  IF (TC(IC).EQ.1) THEN
    CALL FIXE(X3(DL(IDL)*3-1),XFMAX)
    CALL FIXE(X3(DL(IDL)*3-0),XFMIN)
    PRE((IC+ITCOR*NC-NC)*2-1)=XFMAX
    PRE((IC+ITCOR*NC-NC)*2-0)=XFMIN
    IDL=IDL+1

```

```

C---TYPE #2: POINT FIXE EN Y
C

```

```

  ELSE IF (TC(IC).EQ.2) THEN
    CALL FIXE(X3(DL(IDL)*3-1),YFMAX)
    CALL FIXE(X3(DL(IDL)*3-0),YFMIN)
    PRE((IC+ITCOR*NC-NC)*2-1)=YFMAX
    PRE((IC+ITCOR*NC-NC)*2-0)=YFMIN
    IDL=IDL+1

```

```

C---TYPE #3: DISTANCE HORIZONTALE
C

```

```

  ELSE IF (TC(IC).EQ.3) THEN
    DHMAX=0.0
    DHMIN=999999.0
    DO 20 IX1=-1,1,2
      CALL DIST(X3(DL(IDL+0)*3-2)+IX1*X3(DL(IDL+0)*3-(IX1+1)/2),
                X3(DL(IDL+1)*3-2)-IX1*X3(DL(IDL+1)*3-(-1*IX1+1)/2),D)
      IF (D.GT.DHMAX) THEN
        DHMAX=D
      ENDIF
      IF (D.LT.DHMIN) THEN
        DHMIN=D
      ENDIF
    20  CONTINUE
    PRE((IC+ITCOR*NC-NC)*2-1)=DHMAX
    PRE((IC+ITCOR*NC-NC)*2-0)=DHMIN
    IDL=IDL+2

```

```

C---TYPE #4: DISTANCE VERTICALE
C

```

```

  ELSE IF (TC(IC).EQ.4) THEN
    DVMAX=0.0
    DVMIN=999999.0
    DO 30 IY1=-1,1,2
      CALL DIST(X3(DL(IDL+0)*3-2)+IY1*X3(DL(IDL+0)*3-(IY1+1)/2),
                X3(DL(IDL+1)*3-2)-IY1*X3(DL(IDL+1)*3-(-1*IY1+1)/2),D)
      IF (D.GT.DVMAX) THEN
        DVMAX=D
      ENDIF
      IF (D.LT.DVMIN) THEN
        DVMIN=D
      ENDIF
    30  CONTINUE
    PRE((IC+ITCOR*NC-NC)*2-1)=DVMAX
    PRE((IC+ITCOR*NC-NC)*2-0)=DVMIN

```

C
C
C

C---TYPE #5: DISTANCE LINEAIRE ENTRE DEUX NOEUDS

ELSE IF (TC(IC).EQ.5) THEN

DLINMAX=0.0

DLINMIN=9999999.0

DO 40 IX1=-1,1,2

DO 40 IY1=-1,1,2

DO 40 IX2=-1,1,2

DO 40 IY2=-1,1,2

CALL DISTL(X3(DL(IDL+0)*3-2)+IX1*X3(DL(IDL+0)*3-(IX1+1)/2),

1 X3(DL(IDL+1)*3-2)+IY1*X3(DL(IDL+1)*3-(IY1+1)/2),

2 X3(DL(IDL+2)*3-2)-IX2*X3(DL(IDL+2)*3-(-1*IX2+1)/2),

3 X3(DL(IDL+3)*3-2)-IY2*X3(DL(IDL+3)*3-(-1*IY2+1)/2),D)

IF (D.LT.DLINMIN) THEN

DLINMIN=D

ENDIF

IF (D.GT.DLINMAX) THEN

DLINMAX=D

ENDIF

40 CONTINUE

PRE((IC+ITCOR*NC-NC)*2-1)=DLINMAX-P2(IC+ITCOR*NC-NC)

PRE((IC+ITCOR*NC-NC)*2-0)=P2(IC+ITCOR*NC-NC)-DLINMIN

IDL=IDL+4

C
C
C

C---TYPE #6: ANGLE ENTRE DEUX ELEMENTS LINEAIRES

ELSE IF (TC(IC).EQ.6) THEN

AMAX=0.0

AMIN=360.0

DO 50 IX1=-1,1,2

DO 50 IY1=-1,1,2

DO 50 IX2=-1,1,2

DO 50 IY2=-1,1,2

DO 50 IX3=-1,1,2

DO 50 IY3=-1,1,2

DO 50 IX4=-1,1,2

DO 50 IY4=-1,1,2

CALL ANGLE(X3(DL(IDL+0)*3-2)+IX1*X3(DL(IDL+0)*3-(IX1+1)/2),

1 X3(DL(IDL+1)*3-2)+IY1*X3(DL(IDL+1)*3-(IY1+1)/2),

2 X3(DL(IDL+2)*3-2)-IX2*X3(DL(IDL+2)*3-(-1*IX2+1)/2),

3 X3(DL(IDL+3)*3-2)-IY2*X3(DL(IDL+3)*3-(-1*IY2+1)/2),

4 X3(DL(IDL+4)*3-2)+IX1*X3(DL(IDL+4)*3-(IX1+1)/2),

5 X3(DL(IDL+5)*3-2)+IY1*X3(DL(IDL+5)*3-(IY1+1)/2),

6 X3(DL(IDL+6)*3-2)-IX4*X3(DL(IDL+6)*3-(-1*IX4+1)/2),

7 X3(DL(IDL+7)*3-2)-IY4*X3(DL(IDL+7)*3-(-1*IY4+1)/2),A)

WRITE(6,*)A

IF (A.LT.AMIN) THEN

AMIN=A

ENDIF

IF (A.GT.AMAX) THEN

AMAX=A

ENDIF

50 CONTINUE

PRE((IC+ITCOR*NC-NC)*2-1)=AMAX-P2(IC+ITCOR*NC-NC)

PRE((IC+ITCOR*NC-NC)*2-0)=P2(IC+ITCOR*NC-NC)-AMIN

IDL=IDL+8

C
C
C

C---TYPE #7: DISTANCE NOEUD-ELEMENT LINEAIRE

```
C          CALL DISTPE(X3(DL(IDL+0)*3-1),X3(DL(IDL+1)*3-1),
C          1          X3(DL(IDL+2)*3-1),X3(DL(IDL+3)*3-1),
C          1          X3(DL(IDL+4)*3-1),X3(DL(IDL+5)*3-1),DPMAX)
C          CALL DISTPE(X3(DL(IDL+0)*3-0),X3(DL(IDL+1)*3-0),
C          1          X3(DL(IDL+2)*3-0),X3(DL(IDL+3)*3-0),
C          1          X3(DL(IDL+4)*3-0),X3(DL(IDL+5)*3-0),DPMIN)
C          IL NE S'AGIT QUE D'UNE APPROXIMATION
C          DPMAX=X3(DL(IDL+4)*3-1)+X3(DL(IDL+4)*3-0)
C          DPMIN=X3(DL(IDL+5)*3-1)+X3(DL(IDL+5)*3-0)
C          PRE((IC+ITCOR*NC-NC)*2-1)=DPMAX
C          PRE((IC+ITCOR*NC-NC)*2-0)=DPMIN
C          IDL=IDL+6
C          ENDIF
C          WRITE(MP,1030)PRE(IC*2-1),PRE(IC*2-0)
C          10  CONTINUE
C             5  CONTINUE
C
C-----FIN NORMALE
C
C          COND(9)=.TRUE.
C          RETURN
C
C-----FORMATS UTILISES
C
C          1000 FORMAT(' TOPOLOGIE #',I3)
C          1030 FORMAT(2F10.4)
C          3000 FORMAT(' LE CALCUL DES PARAMETRES REELS DOIT DABORD ETRE FAIT')
C
C          END
```

SUBROUTINE VARI_PARA(COND,NN1,P1,P2,VP,NTCOR,ECOR)

=====

C
C
C SOUS-PROGRAMME CALCULANT LA VARIATION SUR LES PARAMETRES
C IL S'AGIT DU CALCUL DE LA DIFFERENCE ENTRE LES PARAMETRES
C REELS ET CEUX DU MODELE.
C
C

=====

C
C AUTEUR : REJEAN FOURNIER
C DATE : 13 MARS 1992
C
C

C
C MODIFIE LE:
C RAISON :
C
C

=====

C
C SOUS-PROGRAMMES APPELES:
C
C

=====

C
C FICHIERS UTILISES:
C
C

=====

C
C VARIABLES UTILISEES :
C
C

C
C COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
C S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)
C NN1 : NOMBRE DE NOEUDS DU MODELE
C P1() : PARAMETRES DU MODELE
C P2() : PARAMETRES REELS
C VP() : VARIATION SUR LES PARAMETRES
C NTCOR : NOMBRE DE TOPOLOGIES CORRESPONDANTES
C ECOR(): NUMERO DE L'ELEMENT DU MODELE CORRESPONDANT AU PREMIER
C ELEMENT DE LA VISION POUR CHAQUE TOPOLOGIE CORRESPONDANTE
C
C MR : NUMERO DU L'UNITE DE LECTURE
C MP : NUMERO DU L'UNITE D'ECRITURE
C
C ITCOR : NUMERO DE LA TOPOLOGIE CORRESPONDANTE
C NP : NOMBRE DE PARAMETRES
C IP : NUMERO DU PARAMETRE
C
C

=====

C
C IMPLICIT NONE
C LOGICAL COND(1)
C INTEGER NN1,NTCOR,ECOR(1)
C INTEGER ITCOR
C INTEGER MR,MP
C REAL P1(1),P2(1),VP(1)
C INTEGER IP,NP
C
C

C
C DATA MR/5/,MP/6/
C
C

C-----INITIALISATION :

C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION

C
C IF (.NOT.COND(3).AND..NOT.COND(9)) THEN

```
C      LE CALCUL DE LA PRECISION SUR LES PARAMETRES REELS
C      DOIVENT DABORD ETRE FAITS
      WRITE(MP,3000)
      RETURN
      ELSE IF (.NOT.COND(3)) THEN
C-     LA LECTURE DU FICHIER DE PARAMETRES DU MODELE
C     DOIT DABORD ETRE FAITE
      WRITE(MP,3010)
      RETURN
      ELSE IF (.NOT.COND(9)) THEN
C-     LE CALCUL DE LA PRECISION SUR LES PARAMETRES REELS
C     DOIT DABORD ETRE FAIT
      WRITE(MP,3020)
      RETURN
      ENDIF

C
C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0
C
      COND(11)=.FALSE.

C
C CALCUL DE LA VARIATION SUR LES PARAMETRES
C
C LE NOMBRE DE PARAMETRES EST EGAL AU NOMBRE DE DL
      NP=NN1*2
      DO 5 ITCOR=1,NTCOR
        WRITE(MP,1000)ITCOR
        DO 10 IP=1,NP
          VP(IP+ITCOR*NP-NP)=P2(IP+ITCOR*NP-NP)-P1(IP)
          WRITE(MP,*)P1(IP),P2(IP+ITCOR*NP-NP),VP(IP+ITCOR*NP-NP)
        10 CONTINUE
      5 CONTINUE

C
C-----FIN NORMALE
C
      COND(10)=.TRUE.
      RETURN

C
C-----FORMATS UTILISES
C
1000 FORMAT(' TOPOLOGIE #',I3)
3000 FORMAT(' UN FICHIER MODELE DE PARAMETRES DOIT ETRE LU ET',
1      '/', ' LE CALCUL DE LA PRECISION SUR LES PARAMETRES FAIT')
3010 FORMAT(' UN FICHIER MODELE DE PARAMETRES DOIT DABORD ETRE LU')
3020 FORMAT(' LE CALCUL DE LA PRECISION SUR LES PARAMETRES',
1      ' DOIT ETRE FAIT')

C
      END
```

SUBROUTINE TOLECOND,NN1,P1,VP,TOL,PRE,NTCOR,ECOR)

=====

SOUS-PROGRAMME DE VERIFICATION DU RESPECT DES TOLERANCES

=====

AUTEUR : REJEAN FOURNIER

DATE : 13 MARS 1992

MODIFIE LE:

RAISON :

=====

SOUS-PROGRAMMES APPELES:

=====

FICHIERS UTILISES:

=====

VARIABLES UTILISEES :

COND(): CETTE VARIABLE INDIQUE SI LE SOUS-PROGRAMME
S'EST EXECUTE CORRECTEMENT (VRAI) OU NON (FAUX)

NN1 : NOMBRE DE NOEUDS DU MODELE

P1() : PARAMETRES DU MODELE

VP() : VARIATION SUR LES PARAMETRES

TOL() : TOLERANCES DU MODELE

PRE() : PRECISION DES PARAMETRES REELS

NTCOR : NOMBRE DE TOPOLOGIES CORRESPONDANTES

ECOR(): NUMERO DE L'ELEMENT DU MODELE CORRESPONDANT AU PREMIER
ELEMENT DE LA VISION POUR CHAQUE TOPOLOGIE CORRESPONDANTE

MR : NUMERO DE L'UNITE DE LECTURE

MP : NUMERO DE L'UNITE D'ECRITURE

ITCOR : NUMERO DE LA TOPOLOGIE CORRESPONDANTE

NP : NOMBRE DE PARAMETRES

IP : NUMERO DU PARAMETRE

NTCLR : NOMBRE DE TOLERANCES RESPECTEES

NTCLN : NOMBRE DE TOLERANCES NON RESPECTEES

NTOLI : NOMBRE DE TOLERANCES INCERTAINES (LA PRECISION NE
PERMET PAS DE JUGER SI LA TOLERANCE EST RESPECTEE)

P1MAX : VALEUR MAXIMUM DU MODELE (VALEUR + TOLERANCE MAX)

P1MIN : VALEUR MINIMUM DU MODELE (VALEUR - TOLERANCE MIN)

P2MAX : VALEUR MAXIMUM DU REEL (VALEUR + PRECISION MAX)

P2MIN : VALEUR MINIMUM DU REEL (VALEUR - PRECISION MIN)

=====

IMPLICIT NONE

LOGICAL COND(1)

INTEGER NN1

REAL P1(1),VP(1),TOL(1),PRE(1)

INTEGER NTCOR,ECOR(1)

INTEGER ITCOR

INTEGER IP,NP

INTEGER NTCLR,NTCLN,NTOLI

```
REAL P1MAX,P1MIN,P2MAX,P2MIN
INTEGER MP
```

```
C
DATA MP/5/
```

```
C-----INITIALISATION :
```

```
C-----VERIFICATION DES CONDITIONS PREALABLES A L'EXECUTION
```

```
C
IF (.NOT.COND(4).AND..NOT.COND(10)) THEN
```

```
C- LA LECTURE DU FICHIER DE TOLERANCES ET LE CALCUL DE LA
C VARIATION SUR LES PARAMETRES DOIVENT DABORD ETRE FAITS
```

```
WRITE(MP,3000)
```

```
RETURN
```

```
ELSE IF (.NOT.COND(4)) THEN
```

```
C- LA LECTURE DU FICHIER DE TOLERANCES
C DOIT DABORD ETRE FAITE
```

```
WRITE(MP,3010)
```

```
RETURN
```

```
ELSE IF (.NOT.COND(10)) THEN
```

```
C- LE CALCUL DE LA
C VARIATION SUR LES PARAMETRES DOIT DABORD ETRE FAITE
```

```
WRITE(MP,3020)
```

```
RETURN
```

```
ENDIF
```

```
C-----TOUS LES SOUS-PROGRAMMES DECOULANT SONT REMIS A 0
```

```
C AUCUN PROGRAMME NE DECOULE DE TOLE
```

```
C---VERIFICATION DES TOLERANCES
```

```
C-- LE NOMBRE DE PARAMETRES EST EGAL AU NOMBRE DE DL
```

```
NP=NN1*2
```

```
DO 5 ITCOR=1,NTCOR
```

```
WRITE(MP,1015)ITCOR
```

```
NTOLR=0
```

```
NTOLN=0
```

```
NTOLI=0
```

```
WRITE(MP,1005)
```

```
DO 10 IP=1,NP
```

```
P1MAX=P1(IP)+TOL(IP*2-1)
```

```
P1MIN=P1(IP)-TOL(IP*2-0)
```

```
P2MAX=P1(IP)+VP(IP+ITCOR*NP-NP)+PREC(IP*2-1+ITCOR*NP-NP)
```

```
P2MIN=P1(IP)+VP(IP+ITCOR*NP-NP)-PREC(IP*2-0+ITCOR*NP-NP)
```

```
IF ((P2MAX.LE.P1MAX).AND.
```

```
(P2MIN.GE.P1MIN)) THEN
```

```
C- TOLERANCE RESPECTEE: LA VALEUR REELLE ET SA PRECISION EST
C COMPLETEMENT INCLUSE DANS LA VALEUR MODELE ET SA TOLERANCE
```

```
NTOLR=NTOLR+1
```

```
WRITE(MP,1000)IP,P1MAX,P1MIN,P2MAX,P2MIN
```

```
ELSE IF ((P2MAX.LT.P1MIN).OR.
```

```
(P2MIN.GT.P1MAX)) THEN
```

```
C- TOLERANCE NON-RESPECTEE: LA VALEUR REELLE ET SA PRECISION
C EST COMPLETEMENT EN-DEHORS DE LA VALEUR MODELE ET SA TOL.
```

```
NTOLN=NTOLN+1
```

```
WRITE(MP,1010)IP,P1MAX,P1MIN,P2MAX,P2MIN
```

```
ELSE
```

```
C- CAS INCERTAIN: LA PRECISION NE PERMET PAS DE JUGER D'UNE
C FACON CERTAINE SI LES TOLERANCES SONT RESPECTEES
```

```

        WRITE(MP,1020)IP,P1MAX,P1MIN,P2MAX,P2MIN
        ENDIF
10    CONTINUE
        WRITE(MP,1030)NTOLR,NTOLN,NTOLI
        IF (NTOLR.EQ.NP) THEN
C--      OBJET ACCEPTE: TOUTES LES TOLERANCES RESPECTEES
            WRITE(MP,1040)
        ELSE IF (NTOLN.GT.0) THEN
C--      OBJET REFUSE: AU MOINS UNE TOLERANCE NON RESPECTEE
            WRITE(MP,1050)
        ELSE
C--      LA PRECISION NE PERMET PAS DE JUGER
            WRITE(MP,1060)
        ENDIF
        S CONTINUE
C
C-----FIN NORMALE
C
        COND(11)=.TRUE.
        RETURN
C
C-----FORMATS
C
1015  FORMAT(' TOPOLOGIE #',I3)
1005  FORMAT(' TOLERANCE RESPECTEE - PARAMETRE -',
1      ' MODELE MAX-MODELE MIN- REEL MAX - REEL MIN')
1000  FORMAT('          OUI                ',I5,4X,4F10.4)
1010  FORMAT('          NON                ',I5,4X,4F10.4)
1020  FORMAT('          INCERTAIN          ',I5,4X,4F10.4)
1030  FORMAT(I2,' TOL. RESPECTEES ',
1          I2,' TOL. NON RESPECTEES ',
2          I2,' TOL. INCERTAINES')
1040  FORMAT(' --->OBJET ACCEPTE')
1050  FORMAT(' --->OBJET REFUSE')
1060  FORMAT(' --->LA PRECISION NE PERMET PAS ACCEPTATION OU REFUS')
3000  FORMAT(' LA LECTURE DU FICHIER DE TOLERANCES ET LE CALCUL DE LA',
1      ' /, VARIATION SUR LES PARAMETRES DOIVENT DABORD ETRE FAITS')
3010  FORMAT(' UN FICHIER MODELE DE TOLERANCES DOIT DABORD ETRE LU')
3020  FORMAT(' LE CALCUL DE LA VARIATION SUR LES PARAMETRES DOIT',$,
1      ' ETRE FAIT')
C
        ENO

```

ANNEXE F

Résultats

<u>Fichier</u>	<u>Description</u>
FOR047.RES	Contour du rectangle. C'est une version abrégée où on n'a que le début et la fin du fichier complet. Résultats de HISTO après l'analyse de l'image du rectangle.
FOR033.RES	Début et fin du contour du polygone. C'est une version abrégée où on n'a que le début et la fin du fichier complet. Résultats de HISTO après l'analyse de l'image du polygone.
FOR094.RES	Propriétés mécaniques calculées à partir de FOR042.RES en version non abrégée.
FOR093.RES	Propriétés mécaniques calculées à partir de FOR033.RES en version non abrégée.
FOR072.DAT	Données de la vision générées par POLY pour le rectangle. POLY a lu ses données dans FOR033.RES dans sa version non abrégée.
FOR063.DAT	Données de la vision générées par POLY pour le polygone. POLY a lu ses données dans FOR033.RES dans sa version non abrégée.
ET1.RES	Résultats de GEO pour l'élément de type 1.
CARRE.RES	Résultats de GEO pour le carré.
FOR072.RES	Résultats de GEO pour le rectangle avec les données de FOR063.DAT.
FOR063.RES	Résultats de GEO pour le polygone avec les données de FOR072.DAT.

874
180.81383 89.65857
180.81383 90.06703
180.81383 90.47550
180.81383 90.88397
180.30377 90.88397
180.30377 91.29243
180.30377 91.70090
180.30377 92.10938
180.30377 92.51784
179.79372 92.51784

••
181.83392 86.39083
181.83392 86.79929
181.32388 86.79929
181.32388 87.20776
181.32388 87.61623
181.32388 88.02470
181.32388 88.43317
180.81383 88.43317
180.81383 88.84163
180.81383 89.25010

1396

74.50803	79.82459
74.50803	80.14453
74.90754	80.14453
74.90754	80.46447
74.90754	80.78441
75.30705	80.78441
75.30705	81.10435
75.30705	81.42429
75.70656	81.42429
75.70656	81.74422

••

72.91001	77.90496
72.91001	78.22490
73.30952	78.22490
73.30952	78.54484
73.30952	78.86478
73.70902	78.86478
74.10853	78.86478
74.10853	79.18472
74.10853	79.50465
74.50803	79.50465

NUMERO DU FICHIER DE LECTURE :
FOR047.DAT

NOMBRE DE POINTS	N	=	874	
AIRE (UNITES^2)	A	=		6305.61768
CENTRE DE GRAVITE	XC	=		130.76941
CENTRE DE GRAVITE	YC	=		86.64348
MOMENTS D'INERTIE	IXX	=		49662668.00000
	IYY	=		112753864.00000
	IXY	=		72133992.00000
	IZZ	=		162416544.00000

FORTRAN STOP

NUMERO DU FICHER DE LECTURE :
FOR033.DAT

NOMBRE DE POINTS	N	=	1396	
AIRE (UNITES^2)	A	=	5038.56299	
CENTRE DE GRAVITE	XC	=	94.93962	
CENTRE DE GRAVITE	YC	=	79.88405	
MOMENTS D'INERTIE	IXX	=	34174360.00000	
	IYY	=	50074764.00000	
	IXY	=	38029132.00000	
	IZZ	=	84249240.00000	

FORTRAN STOP

4		
74.8487	0.0290	0.0290
0.0000	0.0290	0.0290
105.1275	0.0290	0.0290
170.1439	0.0301	0.0301
0.0000	0.0301	0.0301
129.9178	0.0301	0.0301
186.6411	0.0301	0.0301
0.0000	0.0301	0.0301
67.7727	0.0301	0.0301
91.3746	0.0290	0.0290
0.0000	0.0290	0.0290
43.7166	0.0290	0.0290

1
4
1
1
2
1
2
3
1
3
4
1
4
1
2

0.0000	0.0000	0.0000	0.0000	2.0000	0.0000
--------	--------	--------	--------	--------	--------

9		
115.0464	0.0559	0.0559
0.0000	0.0559	0.0559
30.3822	0.0559	0.0559
113.6095	0.0609	0.0609
0.0000	0.0609	0.0609
57.2778	0.0609	0.0609
99.5321	0.0609	0.0609
0.0000	0.0609	0.0609
58.2210	0.0609	0.0609
82.4114	0.0335	0.0335
0.0000	0.0335	0.0335
90.7319	0.0335	0.0335
40.6912	0.0269	0.0269
0.0000	0.0269	0.0269
33.6278	0.0269	0.0269
41.6547	0.0269	0.0269
0.0000	0.0269	0.0269
101.7719	0.0269	0.0269
126.3840	0.0520	0.0520
0.0000	0.0520	0.0520
115.6022	0.0520	0.0520
139.1606	0.0520	0.0520
0.0000	0.0520	0.0520
106.2167	0.0520	0.0520
134.2729	0.0559	0.0559
0.0000	0.0559	0.0559
30.1611	0.0559	0.0559

1						
4						
1						
1						
2						
1						
2						
3						
1						
3						
4						
1						
4						
5						
1						
5						
6						
1						
6						
7						
1						
7						
8						
1						
8						
9						
1						
1						
2	0.0000	0.0000	0.0000	0.0000	2.0000	0.0000

- 1 : LIRE UN FICHER MODELE DE TOPOLOGIE ".SHP"
- 2 : LIRE UN FICHER MODELE DE CONTRAINTES ".SCH"
- 3 : LIRE UN FICHER MODELE DE PARAMETRES ".PAR"
- 4 : LIRE UN FICHER MODELE DE TOLERANCES ".TOL"
- 5 : LIRE UN FICHER DE VISION .VIS (TOPOLOGIE, DL ET PRECISION)
- 6 : SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DES DL)
- 7 : VARIATIONS SUR LA TOPOLOGIE (COMPARAISON MODELE-REEL)
- 8 : CALCUL DES PARAMETRES REELS (DL REELS DANS MODELE DE CONTRAINTES)
- 9 : CALCUL DE LA PRECISION SUR LES PARAMETRES REELS
- 10: VARIATIONS SUR LES PARAMETRES (COMPARAISON MODELE-REEL)
- 11: VERIFICATION DU RESPECT DES TOLERANCES SUR LES PARAMETRES
- 12: FIN

DONNER LE NUMERO DE COMMANDE CHOISIE : 1

DONNER LE NOM DU FICHER A LIRE <> ET1

DONNER LE NUMERO DE COMMANDE CHOISIE : 2

DONNER LE NOM DU FICHER A LIRE <ET1.SCH>

DONNER LE NUMERO DE COMMANDE CHOISIE : 3

DONNER LE NOM DU FICHER A LIRE <ET1.PAR>

DONNER LE NUMERO DE COMMANDE CHOISIE : 4

DONNER LE NOM DU FICHER A LIRE <ET1.TOL>

DONNER LE NUMERO DE COMMANDE CHOISIE : 5

DONNER LE NOM DU FICHER A LIRE <ET1.VIS>

DONNER LE NUMERO DE COMMANDE CHOISIE : 7

1 CORRESPONDANCE(S) TROUVEE(S) ENTRE LES 2 TOPOLOGIES

DONNER LE NUMERO DE COMMANDE CHOISIE : 8

TOPOLOGIE # 1

2.0000

1.0000

5.0000

5.3310

DONNER LE NUMERO DE COMMANDE CHOISIE : 9

TOPOLOGIE # 1

0.0000 0.0000

0.0000 0.0000

0.0000 0.0000

0.0000 0.0000

DONNER LE NUMERO DE COMMANDE CHOISIE : 10

TOPOLOGIE # 1

2.000000	2.000000	0.0000000E+00
1.000000	1.000000	0.0000000E+00
5.000000	5.000000	0.0000000E+00
5.831000	5.830952	-4.8160553E-05

DONNER LE NUMERO DE COMMANDE CHOISIE : 11

TOPOLOGIE # 1

TOLERANCE	RESPECTEE	-	PARAMETRE	-	MODELE MAX-	MODELE MIN-	REEL MAX	-	REEL MIN
OUI			1		2.0001	1.9999	2.0000		2.0000
OUI			2		1.0001	0.9999	1.0000		1.0000
OUI			3		5.0001	4.9999	5.0000		5.0000
OUI			4		5.8312	5.8308	5.8310		5.8310

4 TOL. RESPECTEES 0 TOL. NON RESPECTEES 0 TOL. INCERTAINES

--->OBJET ACCEPTE

DONNER LE NUMERO DE COMMANDE CHOISIE : 12

FORTAN STOP

- 1 : LIRE UN FICHER MODELE DE TOPOLOGIE ".SHP"
- 2 : LIRE UN FICHER MODELE DE CONTRAINTES ".SCH"
- 3 : LIRE UN FICHER MODELE DE PARAMETRES ".PAR"
- 4 : LIRE UN FICHER MODELE DE TOLERANCES ".TOL"
- 5 : LIRE UN FICHER DE VISION .VIS (TOPOLOGIE, DL ET PRECISION)
- 6 : SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DES DL)
- 7 : VARIATIONS SUR LA TOPOLOGIE (COMPARAISON MODELE-REEL)
- 8 : CALCUL DES PARAMETRES REELS (DL REELS DANS MODELE DE CONTRAINTES)
- 9 : CALCUL DE LA PRECISION SUR LES PARAMETRES REELS
- 10: VARIATIONS SUR LES PARAMETRES (COMPARAISON MODELE-REEL)
- 11: VERIFICATION DU RESPECT DES TOLERANCES SUR LES PARAMETRES
- 12: FIN

DONNER LE NUMERO DE COMMANDE CHOISIE : 1

DONNER LE NOM DU FICHER A LIRE <> CARRE

DONNER LE NUMERO DE COMMANDE CHOISIE : 2

DONNER LE NOM DU FICHER A LIRE <CARRE.SCH>

DONNER LE NUMERO DE COMMANDE CHOISIE : 3

DONNER LE NOM DU FICHER A LIRE <CARRE.PAR>

DONNER LE NUMERO DE COMMANDE CHOISIE : 4

DONNER LE NOM DU FICHER A LIRE <CARRE.TOL>

DONNER LE NUMERO DE COMMANDE CHOISIE : 5

DONNER LE NOM DU FICHER A LIRE <CARRE.VIS>

DONNER LE NUMERO DE COMMANDE CHOISIE : 7

4 CORRESPONDANCE(S) TROUVEE(S) ENTRE LES 2 TOPOLOGIES

DONNER LE NUMERO DE COMMANDE CHOISIE : 8

TOPOLOGIE # 1

- 0.0000
- 0.0000
- 5.0000
- 5.0000
- 4.0000
- 5.0000
- 4.0000
- 4.0000
- 90.0000

TOPOLOGIE # 2

- 5.0000
- 0.0000
- 5.0000
- 4.0000
- 5.0000
- 4.0000
- 5.0000

90.0000
 TOPOLOGIE # 3
 5.0000
 4.0000
 0.0000
 5.0000
 4.0000
 5.0000
 4.0000
 90.0000
 TOPOLOGIE # 4
 0.0000
 4.0000
 0.0000
 4.0000
 5.0000
 4.0000
 5.0000
 90.0000

DONNER LE NUMERO DE COMMANDE CHOISIE : 9

TOPOLOGIE # 1
 0.1000 0.1000
 0.1000 0.1000
 0.1000 0.1000
 0.2038 0.2000
 0.2048 0.2000
 0.2038 0.2000
 0.2048 0.2000
 5.3987 5.1530

TOPOLOGIE # 2
 0.1000 0.1000
 0.1000 0.1000
 0.1000 0.1000
 0.2038 0.2000
 0.2048 0.2000
 0.2038 0.2000
 0.2048 0.2000
 5.3987 5.1530

TOPOLOGIE # 3
 0.1000 0.1000
 0.1000 0.1000
 0.1000 0.1000
 0.2038 0.2000
 0.2048 0.2000
 0.2038 0.2000
 0.2048 0.2000
 5.3987 5.1530

TOPOLOGIE # 4
 0.1000 0.1000
 0.1000 0.1000
 0.1000 0.1000
 0.2038 0.2000
 0.2048 0.2000
 0.2038 0.2000
 0.2048 0.2000
 5.3987 5.1530

DONNER LE NUMERO DE COMMANDE CHOISIE : 10

TOPOLOGIE # 1
 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 0.0000000E+00 0.0000000E+00
 5.000000 5.000000 0.0000000E+00
 5.000000 5.000000 0.0000000E+00
 4.000000 4.000000 0.0000000E+00
 5.000000 5.000000 0.0000000E+00
 4.000000 4.000000 0.0000000E+00
 90.00000 90.00000 0.0000000E+00

TOPOLOGIE # 2
 0.0000000E+00 5.000000 5.000000
 0.0000000E+00 0.0000000E+00 0.0000000E+00
 5.000000 5.000000 0.0000000E+00
 5.000000 4.000000 -1.000000
 4.000000 5.000000 1.000000
 5.000000 4.000000 -1.000000
 4.000000 5.000000 1.000000
 90.00000 90.00000 0.0000000E+00

TOPOLOGIE # 3
 0.0000000E+00 5.000000 5.000000
 0.0000000E+00 4.000000 4.000000
 5.000000 0.0000000E+00 -5.000000
 5.000000 5.000000 0.0000000E+00
 4.000000 4.000000 0.0000000E+00
 5.000000 5.000000 0.0000000E+00
 4.000000 4.000000 0.0000000E+00
 90.00000 90.00000 0.0000000E+00

TOPOLOGIE # 4
 0.0000000E+00 0.0000000E+00 0.0000000E+00
 0.0000000E+00 4.000000 4.000000
 5.000000 0.0000000E+00 -5.000000
 5.000000 4.000000 -1.000000
 4.000000 5.000000 1.000000
 5.000000 4.000000 -1.000000
 4.000000 5.000000 1.000000
 90.00000 90.00000 0.0000000E+00

DONNER LE NUMERO DE COMMANDE CHOISIE : 11

TOPOLOGIE # 1

TOLERANCE RESPECTEE	PARAMETRE	-MODELE MAX	-MODELE MIN	REEL MAX	REEL MIN
OUI	1	5.0000	-6.0000	0.1000	-0.1000
OUI	2	5.0000	-5.0000	0.1000	-0.1000
OUI	3	10.0000	0.0000	5.1000	4.9000
INCERTAIN	4	5.0000	5.0000	5.2038	4.8000
INCERTAIN	5	4.1000	3.9000	4.2048	3.8000
INCERTAIN	6	5.2000	4.8000	5.2038	4.8000
INCERTAIN	7	4.2000	3.8000	4.2048	3.8000
OUI	8	97.0000	83.0000	95.3987	84.8470

4 TOL. RESPECTEES 0 TOL. NON RESPECTEES 4 TOL. INCERTAINES

--->LA PRECISION NE PERMET PAS ACCEPTATION OU REFUS

TOPOLOGIE # 2

TOLERANCE RESPECTEE	PARAMETRE	-MODELE MAX	-MODELE MIN	REEL MAX	REEL MIN
OUI	1	6.0000	-5.0000	5.2048	4.3000
OUI	2	5.0000	-5.0000	0.2038	-0.2000
OUI	3	10.0000	0.0000	5.2048	4.8000
INCERTAIN	4	5.0000	5.0000	9.3987	-1.1530
NON	5	4.1000	3.9000	5.1000	4.9000
NON	6	5.2000	4.8000	4.1000	3.9000
NON	7	4.2000	3.8000	5.1000	4.9000
OUI	8	97.0000	83.0000	95.3987	84.8470

4 TOL. RESPECTEES 3 TOL. NON RESPECTEES 1 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 3

TOLERANCE	RESPECTEE	- PARAMETRE	-MODELE MAX	-MODELE MIN	- REEL MAX	- REEL MIN
OUI		1	6.0000	-6.0000	5.1000	4.9000
OUI		2	5.0000	-5.0000	4.1000	3.9000
INCERTAIN		3	10.0000	0.0000	0.1000	-0.1000
INCERTAIN		4	5.0000	5.0000	5.2048	4.8000
INCERTAIN		5	4.1000	3.9000	4.2038	3.8000
INCERTAIN		6	5.2000	4.8000	5.2048	4.8000
INCERTAIN		7	4.2000	3.8000	4.2038	3.8000
OUI		8	97.0000	83.0000	95.3987	84.8470

3 TOL. RESPECTEES 0 TOL. NON RESPECTEES 5 TOL. INCERTAINES

--->LA PRECISION NE PERMET PAS ACCEPTATION DU REFUS

TOPOLOGIE # 4

TOLERANCE	RESPECTEE	- PARAMETRE	-MODELE MAX	-MODELE MIN	- REEL MAX	- REEL MIN
OUI		1	6.0000	-6.0000	0.2038	-0.2000
OUI		2	5.0000	-5.0000	4.2048	3.8000
INCERTAIN		3	10.0000	0.0000	0.2038	-0.2000
INCERTAIN		4	5.0000	5.0000	9.3987	-1.1530
NON		5	4.1000	3.9000	5.1000	4.9000
NON		6	5.2000	4.8000	4.1000	3.9000
NON		7	4.2000	3.8000	5.1000	4.9000
OUI		8	97.0000	83.0000	90.2038	89.8000

3 TOL. RESPECTEES 3 TOL. NON RESPECTEES 2 TOL. INCERTAINES

--->OBJET REFUSE

DONNER LE NUMERO DE COMMANDE CHOISIE : 12
FORTRAN STOP

- 1 : LIRE UN FICHER MODELE DE TOPOLOGIE ".SHP"
- 2 : LIRE UN FICHER MODELE DE CONTRAINTES ".SCH"
- 3 : LIRE UN FICHER MODELE DE PARAMETRES ".PAR"
- 4 : LIRE UN FICHER MODELE DE TOLERANCES ".TOL"
- 5 : LIRE UN FICHER DE VISION .VIS (TOPOLOGIE, DL ET PRECISION)
- 6 : SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DES DL)
- 7 : VARIATIONS SUR LA TOPOLOGIE (COMPARAISON MODELE-REEL)
- 8 : CALCUL DES PARAMETRES REELS (DL REELS DANS MODELE DE CONTRAINTES)
- 9 : CALCUL DE LA PRECISION SUR LES PARAMETRES REELS
- 10: VARIATIONS SUR LES PARAMETRES (COMPARAISON MODELE-REEL)
- 11: VERIFICATION DU RESPECT DES TOLERANCES SUR LES PARAMETRES
- 12: FIN

DONNER LE NUMERO DE COMMANDE CHOISIE : 1

DONNER LE NOM DU FICHER A LIRE <> FOR072

DONNER LE NUMERO DE COMMANDE CHOISIE : 2

DONNER LE NOM DU FICHER A LIRE <FOR072.SCH>

DONNER LE NUMERO DE COMMANDE CHOISIE : 3

DONNER LE NOM DU FICHER A LIRE <FOR072.PAR>

DONNER LE NUMERO DE COMMANDE CHOISIE : 4

DONNER LE NOM DU FICHER A LIRE <FOR072.TOL>

DONNER LE NUMERO DE COMMANDE CHOISIE : 5

DONNER LE NOM DU FICHER A LIRE <FOR072.VIS>

DONNER LE NUMERO DE COMMANDE CHOISIE : 7

4 CORRESPONDANCE(S) TROUVEE(S) ENTRE LES 2 TOPOLOGIES

DONNER LE NUMERO DE COMMANDE CHOISIE : 8

TOPOLOGIE # 1

- 74.8487
- 105.1275
- 170.1438
- 98.4668
- 64.2975
- 98.2568
- 63.5956
- 89.5202

TOPOLOGIE # 2

- 170.1438
- 129.9178
- 186.6411
- 64.2975
- 98.2568
- 63.5956

90.2852
 TOPOLOGIE # 3
 186.6411
 67.7727
 91.3746
 98.2568
 63.5956
 98.4668
 64.2975
 89.3046

TOPOLOGIE # 4
 91.3746
 43.7166
 74.8487
 63.5956
 98.4668
 64.2975
 98.2568
 90.8900

DONNER LE NUMERO DE COMMANDE CHOISIE : 9

TOPOLOGIE # 1
 0.0290 0.0290
 0.0290 0.0290
 0.0301 0.0301
 0.0721 0.0721
 0.0737 0.0736
 0.0718 0.0718
 0.0711 0.0711
 0.0924 0.0924

TOPOLOGIE # 2
 0.0290 0.0290
 0.0290 0.0290
 0.0301 0.0301
 0.0721 0.0721
 0.0737 0.0736
 0.0718 0.0718
 0.0711 0.0711
 0.0924 0.0924

TOPOLOGIE # 3
 0.0290 0.0290
 0.0290 0.0290
 0.0301 0.0301
 0.0721 0.0721
 0.0737 0.0736
 0.0718 0.0718
 0.0711 0.0711
 0.0924 0.0924

TOPOLOGIE # 4
 0.0290 0.0290
 0.0290 0.0290
 0.0301 0.0301
 0.0721 0.0721
 0.0737 0.0736
 0.0718 0.0718
 0.0711 0.0711
 0.0924 0.0924

DONNER LE NUMERO DE COMMANDE CHOISIE : 10

TOPOLOGIE # 1

0.0000000E+00	74.84870	74.84870
0.0000000E+00	105.1275	105.1275
5.000000	170.1438	165.1438
5.000000	98.46682	93.46682
5.000000	64.29755	59.29755
5.000000	98.25681	93.25681
5.000000	63.59563	58.59563
90.00000	89.52016	-0.4798431

TOPOLOGIE # 2

0.0000000E+00	170.1438	170.1438
0.0000000E+00	129.9178	129.9178
5.000000	186.6411	181.6411
5.000000	64.29755	59.29755
5.000000	98.25681	93.25681
5.000000	63.59563	58.59563
5.000000	98.46682	93.46682
90.00000	90.28522	0.2852173

TOPOLOGIE # 3

0.0000000E+00	186.6411	186.6411
0.0000000E+00	67.77270	67.77270
5.000000	91.37460	86.37460
5.000000	98.25681	93.25681
5.000000	63.59563	58.59563
5.000000	98.46682	93.46682
5.000000	64.29755	59.29755
90.00000	89.30464	-0.6953583

TOPOLOGIE # 4

0.0000000E+00	91.37460	91.37460
0.0000000E+00	43.71660	43.71660
5.000000	74.84870	69.84870
5.000000	63.59563	58.59563
5.000000	98.46682	93.46682
5.000000	64.29755	59.29755
5.000000	98.25681	93.25681
90.00000	90.88998	0.8899841

DONNER LE NUMERO DE COMMANDE CHOISIE : 11

TOPOLOGIE # 1

TOLERANCE RESPECTEE -	PARAMETRE	-MODELE MAX-	MODELE MIN-	REEL MAX	- REEL MIN
NON	1	0.0000	0.0000	74.8777	74.8197
NON	2	0.0000	0.0000	105.1565	105.0985
NON	3	5.0000	5.0000	170.1739	170.1137
NON	4	5.0000	5.0000	98.5389	98.3947
NON	5	5.0000	5.0000	64.3712	64.2239
NON	6	5.0000	5.0000	98.3286	98.1851
NON	7	5.0000	5.0000	63.6667	63.5246
NON	8	90.0000	90.0000	89.6126	89.4278

0 TOL. RESPECTEES 3 TOL. NON RESPECTEES 0 TOL. INCERTAINES

-->OBJET REFUSE

TOPOLOGIE # 2

TOLERANCE RESPECTEE -	PARAMETRE	-MODELE MAX-	MODELE MIN-	REEL MAX	- REEL MIN
NON	1	0.0000	0.0000	170.2175	170.0702
NON	2	0.0000	0.0000	129.9896	129.8460
NON	3	5.0000	5.0000	186.7122	186.5700
NON	4	5.0000	5.0000	64.3900	64.2052
NON	5	5.0000	5.0000	98.2869	98.2267
NON	6	5.0000	5.0000	63.6257	63.5655
NON	7	5.0000	5.0000	98.4969	98.4357
NON	8	90.0000	90.0000	90.2588	90.2116

0 TOL. RESPECTEES 3 TOL. NON RESPECTEES 0 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 3

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX-	MODELE MIN-	REEL MAX	- REEL MIN
NON		1	0.0000	0.0000	186.6712	186.6110
NON		2	0.0000	0.0000	67.8028	67.7426
NON		3	5.0000	5.0000	91.4047	91.3445
NON		4	5.0000	5.0000	98.3305	98.1832
NON		5	5.0000	5.0000	63.6674	63.5239
NON		6	5.0000	5.0000	98.5379	98.3958
NON		7	5.0000	5.0000	64.3696	64.2255
NON		8	90.0000	90.0000	89.4034	89.2059

0 TOL. RESPECTEES 8 TOL. NON RESPECTEES 0 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 4

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX-	MODELE MIN-	REEL MAX	- REEL MIN
NON		1	0.0000	0.0000	91.4464	91.3028
NON		2	0.0000	0.0000	43.7877	43.6455
NON		3	5.0000	5.0000	74.9208	74.7766
NON		4	5.0000	5.0000	63.6944	63.4969
NON		5	5.0000	5.0000	98.4969	98.4367
NON		6	5.0000	5.0000	64.3275	64.2674
NON		7	5.0000	5.0000	98.2858	98.2278
NON		8	90.0000	90.0000	90.9618	90.8182

0 TOL. RESPECTEES 3 TOL. NON RESPECTEES 0 TOL. INCERTAINES

--->OBJET REFUSE

DONNER LE NUMERO DE COMMANDE CHOISIE : 12
FORTRAN STOP

- 1 : LIRE UN FICHIER MODELE DE TOPOLOGIE ".SHP"
- 2 : LIRE UN FICHIER MODELE DE CONTRAINTES ".SCH"
- 3 : LIRE UN FICHIER MODELE DE PARAMETRES ".PAR"
- 4 : LIRE UN FICHIER MODELE DE TOLERANCES ".TOL"
- 5 : LIRE UN FICHIER DE VISION .VIS (TOPOLOGIE, DL ET PRECISION)
- 6 : SOLUTION DE LA GEOMETRIE DU MODELE (CALCUL DES DL)
- 7 : VARIATIONS SUR LA TOPOLOGIE (COMPARAISON MODELE-REEL)
- 8 : CALCUL DES PARAMETRES REELS (DL REELS DANS MODELE DE CONTRAINTES)
- 9 : CALCUL DE LA PRECISION SUR LES PARAMETRES REELS
- 10: VARIATIONS SUR LES PARAMETRES (COMPARAISON MODELE-REEL)
- 11: VERIFICATION DU RESPECT DES TOLERANCES SUR LES PARAMETRES
- 12: FIN

DONNER LE NUMERO DE COMMANDE CHOISIE : 1

DONNER LE NOM DU FICHIER A LIRE <> FOR063

DONNER LE NUMERO DE COMMANDE CHOISIE : 2

DONNER LE NOM DU FICHIER A LIRE <FOR063.SCH>

DONNER LE NUMERO DE COMMANDE CHOISIE : 3

DONNER LE NOM DU FICHIER A LIRE <FOR063.PAR>

DONNER LE NUMERO DE COMMANDE CHOISIE : 4

DONNER LE NOM DU FICHIER A LIRE <FOR063.TOL>

DONNER LE NUMERO DE COMMANDE CHOISIE : 5

DONNER LE NOM DU FICHIER A LIRE <FOR063.VIS>

DONNER LE NUMERO DE COMMANDE CHOISIE : 7

9 CORRESPONDANCE(S) TROUVEE(S) ENTRE LES 2 TOPOLOGIES

DONNER LE NUMERO DE COMMANDE CHOISIE : 8

TOPOLOGIE # 1

- 115.0464
- 0.0000
- 30.3322
- 113.6095
- 115.0464
- 57.2778
- 15.5143
- 0.0000
- 56.8254
- 82.4114
- 115.0464
- 90.7319
- 74.3552
- 0.0000
- 81.4186

115.0464
 101.7719
 TOPOLOGIE # 2
 113.6095
 0.0000
 57.2778
 99.5321
 113.6095
 58.2210
 31.1981
 0.0000
 22.9776
 40.6912
 113.6095
 33.6278
 71.9548
 0.0000
 11.3376
 126.3840
 113.6095
 115.6022

TOPOLOGIE # 3
 99.5321
 0.0000
 58.2210
 82.4114
 99.5321
 90.7319
 58.8409
 0.0000
 65.5043
 41.6547
 99.5321
 101.7719
 26.8519
 0.0000
 16.0701
 139.1606
 99.5321
 108.2167

TOPOLOGIE # 4
 82.4114
 0.0000
 90.7319
 40.6912
 82.4114
 33.6278
 40.7567
 0.0000
 19.3605
 126.3840
 82.4114
 115.6022
 56.7492
 0.0000
 25.8053
 134.2729
 82.4114
 30.1611

TOPOLOGIE # 5
 101.7719

0.0000
 33.6278
 41.6547
 40.6912
 101.7719
 85.6928
 0.0000
 74.9110
 139.1606
 40.6912
 108.2167
 93.5817
 0.0000
 10.5301
 115.0464
 40.6912
 30.3822

TOPOLOGIE # 6

41.6547
 0.0000
 101.7719
 126.3840
 41.6547
 115.6022
 97.5059
 0.0000
 66.5620
 134.2729
 41.6547
 30.1611
 73.3917
 0.0000
 11.2725
 113.6095
 41.6547
 57.2778

TOPOLOGIE # 7

126.3840
 0.0000
 115.6022
 139.1606
 126.3840
 108.2167
 7.8889
 0.0000
 96.2229
 115.0464
 126.3840
 30.3822
 12.7745
 0.0000
 69.1062
 99.5321
 126.3840
 58.2210

TOPOLOGIE # 8

139.1606
 0.0000
 108.2167
 134.2729
 139.1606

30.1611
 24.1142
 0.0000
 108.7784
 113.6095
 139.1606
 57.2778
 39.6235
 0.0000
 80.9396
 82.4114
 139.1606
 90.7319

TOPOLOGIE # 9

134.2729
 0.0000
 30.1611
 115.0464
 134.2729
 30.3822
 20.6634
 0.0000
 76.9951
 99.5321
 134.2729
 58.2210
 51.8615
 0.0000
 43.5410
 40.6912
 134.2729
 33.6278

DONNER LE NUMERO DE COMMANDE CHOISIE : 9

TOPOLOGIE # 1

0.0559 0.0559
 0.0559 0.0559
 0.0559 0.0559
 113.7263 113.4927
 115.1632 114.9296
 57.3946 57.1610
 15.6311 15.3975
 0.1168 0.1168
 56.9422 56.7086
 82.5008 82.3220
 115.1358 114.9570
 90.8213 90.6425
 74.4380 74.2724
 0.0828 0.0828
 81.5014 81.3358
 41.7379 41.5719
 115.1292 114.9636
 101.8547 101.6891

TOPOLOGIE # 2

0.0559 0.0559
 0.0559 0.0559
 0.0559 0.0559
 113.7263 113.4927
 115.1632 114.9296
 57.3946 57.1610

15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

TOPOLOGIE # 3

0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

TOPOLOGIE # 4

0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

TOPOLOGIE # 5

0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220

115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

TOPOLOGIE # 6

0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

TOPOLOGIE # 7

0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

TOPOLOGIE # 8

0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5008	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724

81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891
TOPOLOGIE # 9	
0.0559	0.0559
0.0559	0.0559
0.0559	0.0559
113.7263	113.4927
115.1632	114.9296
57.3946	57.1610
15.6311	15.3975
0.1168	0.1168
56.9422	56.7086
82.5009	82.3220
115.1358	114.9570
90.8213	90.6425
74.4380	74.2724
0.0828	0.0828
81.5014	81.3358
41.7375	41.5719
115.1292	114.9636
101.8547	101.6891

DONNER LE NUMERO DE COMMANDE CHOISIE : 10

TOPOLOGIE # 1

0.000000E+00	115.0464	115.0464
1.000000	0.000000E+00	-1.000000
2.000000	30.38220	28.38220
6.000000	113.6095	107.6095
95.00000	115.0464	20.04640
9.000000	57.27780	48.27780
95.00000	15.51431	-79.48569
79.50000	0.000000E+00	-79.50000
54.00000	55.82540	2.825401
19.00000	32.41140	63.41140
36.00000	115.0464	79.04640
52.00000	90.73190	38.73190
23.00000	74.35520	51.35520
53.00000	0.000000E+00	-53.00000
19.50000	81.41860	61.91860
79.00000	41.65470	-37.34530
0.000000E+00	115.0464	115.0464
79.00000	101.7719	22.77190

TOPOLOGIE # 2

0.000000E+00	113.6095	113.6095
1.000000	0.000000E+00	-1.000000
2.000000	57.27780	55.27780
6.000000	99.53210	93.53210
95.00000	113.6095	18.60950
9.000000	58.22100	49.22100
95.00000	31.19810	-63.60190
79.50000	0.000000E+00	-79.50000
54.00000	22.87759	-31.12241
19.00000	40.69120	21.69120
36.00000	113.6095	77.60950
52.00000	33.62780	-18.37220
23.00000	71.95480	48.95480
53.00000	0.000000E+00	-53.00000

79.00000	126.3840	47.38400
0.0000000E+00	113.6095	113.6095
79.00000	115.6022	36.60220
TOPOLOGIE # 3		
0.0000000E+00	99.53210	99.53210
1.000000	0.0000000E+00	-1.000000
2.000000	58.22100	56.22100
6.000000	82.41140	76.41140
95.00000	99.53210	4.532097
9.000000	90.73190	81.73190
95.00000	58.84090	-36.15910
79.50000	0.0000000E+00	-79.50000
54.00000	65.90430	11.90430
19.00000	41.65470	22.65470
36.00000	99.53210	63.53210
52.00000	101.7719	49.77190
23.00000	26.85191	3.851906
53.00000	0.0000000E+00	-53.00000
19.50000	16.07011	-3.429893
79.00000	139.1606	60.16060
0.0000000E+00	99.53210	99.53210
79.00000	108.2167	29.21670

TOPOLOGIE # 4		
0.0000000E+00	82.41140	82.41140
1.000000	0.0000000E+00	-1.000000
2.000000	90.73190	88.73190
6.000000	40.69120	34.69120
95.00000	82.41140	-12.58860
9.000000	33.62780	24.62780
95.00000	40.75670	-54.24330
79.50000	0.0000000E+00	-79.50000
54.00000	19.36050	-34.63950
19.00000	126.3840	107.3840
36.00000	82.41140	46.41140
52.00000	115.6022	63.60220
23.00000	56.74920	33.74920
53.00000	0.0000000E+00	-53.00000
19.50000	25.80530	6.305298
79.00000	134.2729	55.27290
0.0000000E+00	82.41140	82.41140
79.00000	30.16110	-48.83890

TOPOLOGIE # 5		
0.0000000E+00	40.69120	40.69120
1.000000	0.0000000E+00	-1.000000
2.000000	33.62780	31.62780
6.000000	41.65470	35.65470
95.00000	40.69120	-54.30880
9.000000	101.7719	92.77190
95.00000	85.69280	-9.307198
79.50000	0.0000000E+00	-79.50000
54.00000	74.91100	20.91100
19.00000	139.1606	120.1606
36.00000	40.69120	4.691200
52.00000	108.2167	56.21670
23.00000	93.58170	70.58170
53.00000	0.0000000E+00	-53.00000
19.50000	10.53010	-8.969900
79.00000	115.0464	36.04640
0.0000000E+00	40.69120	40.69120
79.00000	30.38220	-48.61780

TOPOLOGIE # 6

0.000000E+00	41.65470	41.65470
1.000000	0.000000E+00	-1.000000
2.000000	101.7719	99.77190
6.000000	126.3840	120.3840
95.00000	41.65470	-53.34530
9.000000	115.6022	106.6022
95.00000	97.50590	2.505898
79.50000	0.000000E+00	-79.50000
54.00000	66.56200	12.56200
19.00000	134.2729	115.2729
36.00000	41.65470	5.654701
52.00000	30.16110	-21.83890
23.00000	73.39170	50.39170
53.00000	0.000000E+00	-53.00000
19.50000	11.27250	-8.227499
79.00000	113.6095	34.60950
0.000000E+00	41.65470	41.65470
79.00000	57.27780	-21.72220

TOPOLOGIE # 7

0.000000E+00	126.3840	126.3840
1.000000	0.000000E+00	-1.000000
2.000000	115.6022	113.6022
6.000000	139.1606	133.1606
95.00000	126.3840	31.38400
9.000000	108.2167	99.21670
95.00000	7.888901	-87.11110
79.50000	0.000000E+00	-79.50000
54.00000	96.22290	42.22290
19.00000	115.0464	96.04640
36.00000	126.3840	90.38400
52.00000	30.38220	-21.61780
23.00000	12.77451	-10.22549
53.00000	0.000000E+00	-53.00000
19.50000	69.10620	49.60620
79.00000	99.53210	20.53210
0.000000E+00	126.3840	126.3840
79.00000	58.22100	-20.77900

TOPOLOGIE # 8

0.000000E+00	139.1606	139.1606
1.000000	0.000000E+00	-1.000000
2.000000	108.2167	106.2167
6.000000	134.2729	128.2729
95.00000	139.1606	44.16060
9.000000	30.16110	21.16110
95.00000	24.11420	-70.88530
79.50000	0.000000E+00	-79.50000
54.00000	108.7784	54.77840
19.00000	113.6095	94.60950
36.00000	139.1606	103.1606
52.00000	57.27780	5.277802
23.00000	39.62850	16.62850
53.00000	0.000000E+00	-53.00000
19.50000	80.93960	61.43960
79.00000	82.41140	3.411400
0.000000E+00	139.1606	139.1606
79.00000	90.73190	11.73190

TOPOLOGIE # 9

0.000000E+00	134.2729	134.2729
1.000000	0.000000E+00	-1.000000
2.000000	30.16110	28.16110

95.00000	134.2729	39.27290
9.000000	30.38220	21.38220
95.00000	20.66341	-74.33659
79.50000	0.0000000E+00	-79.50000
54.00000	76.99510	22.99510
19.00000	99.53210	80.53210
36.00000	134.2729	98.27290
52.00000	59.22100	6.221001
23.00000	51.86150	28.86150
53.00000	0.0000000E+00	-53.00000
19.50000	43.54100	24.04100
79.00000	40.69120	-38.30880
0.0000000E+00	134.2729	134.2729
79.00000	33.62780	-45.37220

DONNER LE NUMERO DE COMMANDE CHOISIE : 11

TOPOLOGIE # 1

TOLERANCE	RESPECTEE	- PARAMETRE	-MODELE MAX-	MODELE MIN-	REEL MAX	- REEL MIN
NON		1	90.0000	-90.0000	115.1023	114.9905
OUI		2	91.0000	-89.0000	0.0559	-0.0559
OUI		3	92.0000	-88.0000	30.4381	30.3263
INCERTAIN		4	6.1000	5.9000	227.3358	0.1168
INCERTAIN		5	95.1000	94.9000	230.2096	0.1168
INCERTAIN		6	9.1000	8.9000	114.6724	0.1168
NON		7	95.1000	94.9000	31.1454	0.1168
NON		8	79.6000	79.4000	0.1168	-0.1168
INCERTAIN		9	54.1000	53.9000	113.7676	0.1168
INCERTAIN		10	19.1000	18.9000	164.9122	0.0894
INCERTAIN		11	36.1000	35.9000	230.1822	0.0894
INCERTAIN		12	52.1000	51.9000	181.5532	0.0894
INCERTAIN		13	23.1000	22.9000	148.7932	0.0828
NON		14	53.1000	52.9000	0.0828	-0.0828
INCERTAIN		15	19.6000	19.4000	162.9200	0.0828
INCERTAIN		16	79.1000	78.9000	83.3922	0.0828
INCERTAIN		17	0.1000	-0.1000	230.1756	0.0828
INCERTAIN		18	79.1000	78.9000	203.6266	0.0828

2 TOL. RESPECTEES 4 TOL. NON RESPECTEES 12 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 2

TOLERANCE	RESPECTEE	- PARAMETRE	-MODELE MAX-	MODELE MIN-	REEL MAX	- REEL MIN
INCERTAIN		1	90.0000	-90.0000	196.1103	31.2875
INCERTAIN		2	91.0000	-89.0000	115.1358	-114.9570
INCERTAIN		3	92.0000	-88.0000	148.0991	-33.3647
NON		4	6.1000	5.9000	173.9701	25.2597
NON		5	95.1000	94.9000	113.6923	113.5267
INCERTAIN		6	9.1000	8.9000	139.7224	-23.1148
NON		7	95.1000	94.9000	72.9356	-10.3738
INCERTAIN		8	79.6000	79.4000	115.1292	-114.9636
INCERTAIN		9	54.1000	53.9000	124.7323	-78.8115
NON		10	19.1000	18.9000	40.7521	40.6303
NON		11	36.1000	35.9000	113.6704	113.5486
NON		12	52.1000	51.9000	33.6887	33.5669
INCERTAIN		13	23.1000	22.9000	171.6087	-27.4555
INCERTAIN		14	53.1000	52.9000	113.7313	-113.4877
INCERTAIN		15	19.6000	19.4000	70.1804	-46.2616
NON		16	79.1000	78.9000	157.6765	95.2803
NON		17	0.1000	-0.1000	113.7039	113.5151
NON		18	79.1000	78.9000	138.5742	92.8190

0 TOL. RESPECTEES 9 TOL. NON RESPECTEES 9 TOL. INCERTAINES

TOPOLOGIE # 3

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX	-MODELE MIN	REEL MAX	REEL MIN
NON	1		90.0000	-90.0000	99.5930	99.4712
OUI	2		91.0000	-89.0000	0.0609	-0.0609
OUI	3		92.0000	-88.0000	58.2819	58.1601
INCERTAIN	4		6.1000	5.9000	182.0653	-16.9989
INCERTAIN	5		95.1000	94.9000	213.2634	-13.9556
NON	6		9.1000	8.9000	149.0747	32.6327
NON	7		95.1000	94.9000	90.1334	27.7372
NON	8		79.6000	79.4000	0.0944	-0.0944
INCERTAIN	9		54.1000	53.9000	88.8763	43.1211
INCERTAIN	10		19.1000	18.9000	82.4337	1.0513
INCERTAIN	11		36.1000	35.9000	213.2294	-13.9896
NON	12		52.1000	51.9000	135.4875	68.2319
INCERTAIN	13		23.1000	22.9000	98.8945	-45.0151
NON	14		53.1000	52.9000	0.0878	-0.0878
INCERTAIN	15		19.6000	19.4000	27.9955	4.3203
INCERTAIN	16		79.1000	78.9000	265.6575	12.8895
INCERTAIN	17		0.1000	-0.1000	213.2545	-13.9645
INCERTAIN	18		79.1000	78.9000	223.9318	-7.2726

2 TOL. RESPECTEES 6 TOL. NON RESPECTEES 10 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 4

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX	-MODELE MIN	REEL MAX	REEL MIN
INCERTAIN	1		90.0000	-90.0000	123.1904	41.8080
INCERTAIN	2		91.0000	-89.0000	113.6973	-113.5217
INCERTAIN	3		92.0000	-88.0000	124.4475	57.1919
INCERTAIN	4		6.1000	5.9000	112.7338	-31.1758
NON	5		95.1000	94.9000	82.4992	82.3236
NON	6		9.1000	8.9000	45.5532	21.8780
INCERTAIN	7		95.1000	94.9000	167.2536	-85.5144
INCERTAIN	8		79.6000	79.4000	113.7224	-113.4966
INCERTAIN	9		54.1000	53.9000	135.0756	-96.1288
NON	10		19.1000	18.9000	126.4449	126.3231
NON	11		36.1000	35.9000	82.4723	82.3505
NON	12		52.1000	51.9000	115.6631	115.5413
INCERTAIN	13		23.1000	22.9000	139.2550	-25.5678
INCERTAIN	14		53.1000	52.9000	99.6265	-99.4377
INCERTAIN	15		19.6000	19.4000	116.6316	-64.8322
INCERTAIN	16		79.1000	78.9000	193.2016	75.5198
NON	17		0.1000	-0.1000	82.4992	82.3236
INCERTAIN	18		79.1000	78.9000	96.1532	-35.6554

0 TOL. RESPECTEES 6 TOL. NON RESPECTEES 12 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 5

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX	-MODELE MIN	REEL MAX	REEL MIN
OUI	1		90.0000	-90.0000	40.7521	40.6303
OUI	2		91.0000	-89.0000	0.0609	-0.0609
OUI	3		92.0000	-88.0000	33.6887	33.5669
INCERTAIN	4		6.1000	5.9000	124.1605	-40.6623
INCERTAIN	5		95.1000	94.9000	140.3177	-58.7465
NON	6		9.1000	8.9000	192.5982	11.1344
INCERTAIN	7		95.1000	94.9000	144.6215	26.9397
NON	8		79.6000	79.4000	0.0878	-0.0878
INCERTAIN	9		54.1000	53.9000	140.9031	9.0945
NON	10		19.1000	18.9000	180.9031	97.5937
INCERTAIN	11		36.1000	35.9000	140.3111	-58.7531
INCERTAIN	12		52.1000	51.9000	210.0764	6.5326
NON	13		23.1000	22.9000	120.5465	66.8427
NON	14		53.1000	52.9000	0.1129	-0.1129

INCERTAIN	16	79.1000	78.9000	254.3199	-24.0013
INCERTAIN	17	0.1000	-0.1000	140.3362	-58.7280
INCERTAIN	18	79.1000	78.9000	138.7118	-77.7216

3 TOL. RESPECTEES 5 TOL. NON RESPECTEES 10 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 5

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX-	-MODELE MIN-	REEL MAX	- REEL MIN
OUI	1		90.0000	-90.0000	83.3972	0.0878
INCERTAIN	2		91.0000	-89.0000	99.6199	-99.4443
INCERTAIN	3		92.0000	-88.0000	203.6316	0.0878
NON	4		6.1000	5.9000	153.3488	99.6450
NON	5		95.1000	94.9000	41.7676	41.5418
NON	6		9.1000	3.9000	131.7852	99.6450
INCERTAIN	7		95.1000	94.9000	236.7794	-41.5418
INCERTAIN	8		79.6000	79.4000	99.6450	-99.4192
INCERTAIN	9		54.1000	53.9000	174.8916	-41.5418
NON	10		19.1000	18.9000	134.3064	134.2394
NON	11		36.1000	35.9000	41.6882	41.6212
NON	12		52.1000	51.9000	30.1946	30.1276
NON	13		23.1000	22.9000	114.1433	32.7609
INCERTAIN	14		53.1000	52.9000	82.4718	-82.3510
INCERTAIN	15		19.6000	19.4000	44.9607	-22.2949
INCERTAIN	16		79.1000	78.9000	154.4266	72.9132
NON	17		0.1000	-0.1000	41.7151	41.5943
NON	18		79.1000	78.9000	76.6987	37.9777

1 TOL. RESPECTEES 9 TOL. NON RESPECTEES 8 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 7

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX-	-MODELE MIN-	REEL MAX	- REEL MIN
NON	1		90.0000	-90.0000	126.4175	126.3505
OUI	2		91.0000	-89.0000	0.0335	-0.0335
NON	3		92.0000	-88.0000	115.6357	115.5687
NON	4		6.1000	5.9000	179.9122	98.5298
INCERTAIN	5		95.1000	94.9000	208.8558	44.0330
NON	6		9.1000	8.9000	141.9049	74.6493
NON	7		95.1000	94.9000	48.7060	-32.8074
NON	8		79.6000	79.4000	0.0604	-0.0604
NON	9		54.1000	53.9000	115.6438	76.9228
INCERTAIN	10		19.1000	18.9000	241.5159	-11.2521
NON	11		36.1000	35.9000	208.8809	44.0581
INCERTAIN	12		52.1000	51.9000	146.0699	-85.1345
INCERTAIN	13		23.1000	22.9000	69.6092	-43.8892
NON	14		53.1000	52.9000	0.0855	-0.0855
NON	15		19.6000	19.4000	94.9970	43.3864
INCERTAIN	16		79.1000	78.9000	233.8944	-34.6514
NON	17		0.1000	-0.1000	208.8848	44.0620
INCERTAIN	18		79.1000	78.9000	88.4715	28.1493

1 TOL. RESPECTEES 11 TOL. NON RESPECTEES 6 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 8

TOLERANCE	RESPECTEE	PARAMETRE	-MODELE MAX-	-MODELE MIN-	REEL MAX	- REEL MIN
INCERTAIN	1		90.0000	-90.0000	265.6301	12.8621
OUI	2		91.0000	-89.0000	82.4969	-82.3259
INCERTAIN	3		92.0000	-88.0000	223.9044	-7.3000
NON	4		6.1000	5.9000	191.1076	77.6092
NON	5		95.1000	94.9000	139.2461	139.0751
INCERTAIN	6		9.1000	8.9000	56.0519	4.4413
INCERTAIN	7		95.1000	94.9000	158.4765	-110.0693
INCERTAIN	8		79.6000	79.4000	82.5008	-82.3220
NON	9		54.1000	53.9000	139.0289	78.7067

NON	11	36.1000	35.9000	139.1875	139.1337
NON	12	52.1000	51.9000	57.3047	57.2509
INCERTAIN	13	23.1000	22.9000	81.3370	-1.9724
NON	14	53.1000	52.9000	40.7450	-40.6374
INCERTAIN	15	19.6000	19.4000	182.7653	-20.7785
INCERTAIN	16	79.1000	78.9000	168.1831	-3.2025
NON	17	0.1000	-0.1000	139.2395	139.0817
INCERTAIN	18	79.1000	78.9000	165.7218	15.8998

1 TOL. RESPECTEES 8 TOL. NON RESPECTEES 9 TOL. INCERTAINES

--->OBJET REFUSE

TOPOLOGIE # 9

TOLERANCE RESPECTEE	PARAMETRE	MODELE MAX	MODELE MIN	REEL MAX	REEL MIN
NON	1	90.0000	-90.0000	134.2998	134.2460
OUI	2	91.0000	-39.0000	0.0269	-0.0269
OUI	3	92.0000	-38.0000	30.1880	30.1342
NON	4	6.1000	5.9000	156.7549	73.4455
INCERTAIN	5	95.1000	94.9000	175.0179	93.6355
INCERTAIN	6	9.1000	8.9000	132.2079	-71.3359
INCERTAIN	7	95.1000	94.9000	106.4351	-64.9505
NON	8	79.6000	79.4000	0.0789	-0.0789
INCERTAIN	9	54.1000	53.9000	151.9850	2.1630
INCERTAIN	10	19.1000	18.9000	238.7716	-39.5496
NON	11	36.1000	35.9000	175.0430	93.6606
INCERTAIN	12	52.1000	51.9000	166.5166	-49.9168
INCERTAIN	13	23.1000	22.9000	145.5260	-41.6374
NON	14	53.1000	52.9000	0.0828	-0.0828
NON	15	19.6000	19.4000	54.1539	33.0937
INCERTAIN	16	79.1000	78.9000	155.8204	-74.2724
NON	17	0.1000	-0.1000	175.0469	93.6645
NON	18	79.1000	78.9000	64.0928	3.3284

2 TOL. RESPECTEES 8 TOL. NON RESPECTEES 8 TOL. INCERTAINES

--->OBJET REFUSE

DONNER LE NUMERO DE COMMANDE CHOISIE : 12

FORTRAN STOP