INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

EMC^2 Un logiciel d'édition de maillages et de contours bidimensionnels

Eric Saltel, Frédéric Hecht

N° 118

Octobre 1995

PROGRAMME 6

Rapport technique

ISSN 0249-6399

EMC^2 Un logiciel d'édition de maillages et de contours bidimensionnels

Eric Saltel, Frédéric Hecht

Programme 6 — Calcul scientifique, modélisation et logiciel numérique
Projet MENUSIN

Rapport technique n° 118 — Octobre 1995 — 66 pages

Résumé : Emc^2 est un logiciel portable, graphique et intéractif d'édition de maillage et contours en 2 dimensions. Il permet de générer intéractivement des maillages bidimensionnels pour la méthode des éléments finis en définissant la géometrie (D.A.O), la discrétisation des contours, les sous-domaines et les numéros de reférence (afin d'introduire un lien avec la physique: conditions aux limites, propriétés des matériaux). Ces maillages, formés de triangles ou de quadrangles, sont de type grille ou de type Delaunay-Voronoï. Il est possible d'éditer un maillage en ajoutant, supprimant, déplaçant des sommets,... et en lui appliquant des transformations affines: symetrie, rotation,...) etc.

Mots-clé : MAILLAGE, ELEMENTS FINI, D.A.O, GEOMETRY, TRIANGLE, QUADRANGLE

(Abstract: pto)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Wysiwyg 2D finite elements mesh generator

Abstract: Emc2 is a portable, interactive and graphic software Edition of two dimensional geometry and mesh. We can create and modify the geometry (CAD), define the discretization on the lines, define the subdomains, and define some reference numbers to take into account the boundary conditions and material properties. The elements of the mesh are traingles and quadrilaterals. We have two kind of meshes : grid mesh and Delaunay Voronoï(automatic mesh). We make the edition of the mesh by moving, removing, adding vertices, by regularization, or by transformations (symmetry, rotation,..), etc.

Key-words: MESH, GRID, FINITE ELEMENT, CAD, GEOMETRIE, TRIANGLE,QUADRANGLE

Table des matières

Table des figures

Chapitre 1

INTRODUCTION

Une des difficultés de la méthode des éléments finis est la définition du domaine de calcul Ω et la construction d'un maillage de Ω. Etant donné que l'on est amené à résoudre des problèmes de plus en proches de la réalité, la complexité des domaines de calcul croît parallèlement,et le temps humain passé à définir la géométrie et générer les maillages peut devenir exorbitant si l'on ne dispose pas d'outils adéquats.

Ces différentes considérations ainsi que le développement des stations de travail (graphiques) ont motivé le développement et la réalisation du logiciel Emc^2 (pour **E**dition de **m**aillages et de **c**ontours en **2** dimensions).

L'utilisation de ce logiciel portable, graphique, rapide, basé sur des menus (*qui orientent les actions pour la définition des géométries et le maillage de celles-ci*), diminuera de façon significative le temps nécessaire à la préparation des données dans la méthode des éléments finis.

Chapitre 2

Généralités

Le logiciel est divisé en trois applications principales (exclusives):

1. l'application **CONSTRUCTION** : édition et création de la géométrie (contours);
2. l'application **PREP_MESH** : édition et définition du maillage des contours, des références des lignes, points, et sous domaines;
3. l'application **EDIT_MESH** : création et édition d'un maillage triangulaire, ou quadrangulaire.

Dans chacune de ces applications on accède aussi aux menus suivants

- *GENERALITES* permettant entre autre de changer d'application (commuter), de sauver ou restaurer une base de données, de faire des copies d'écran, de ré-exécuter une session, de sortir de Emc^2 (finir)...
- *GESTION DE L'ECRAN GRAPHIQUE* servant à gérer l'affichage de la base de données,
- *CALCULETTE* permettant d'entrer une valeur numérique (pouvant être le résultat d'une expression) quand cela est nécessaire, ou de renvoyer certaines valeurs comme le rayon d'un cercle ou d'un arc de cercle, la longueur d'un arc ou d'un segment...
- *DESIGNATION* servant à définir le type de l'élément que l'on veut désigner, par exemple coord, point, droite, cercle, arc, segment, spline... Ce menu dépendra de l'application en cours car les ITEM à désigner peuvent être différents selon l'application.

Chaque application (principale) possède un menu qui lui est propre. Toutes les applications sont dirigées par des grammaires LL1.

L'écran (figure 2.1) est découpé en 9 zônes: 5 sont des menus ((1), (2), (4), (7), (6)), 3 sont des zônes d'affichage ((8), (9), (5)) , la zône centrale est graphique.

Les menus **GENERAL(ITES)**, **CALCULETTE**, **GESTION DE L'ECRAN** sont fixes et indépendants des applications. Les menus **APPLICATION** et **DESIGNATION** sont évidemment dépendants de l'application en cours.

Le logiciel est interactif, toutes les entrées se font à l'aide de la souris et du clavier. La souris permet de désigner les ITEM (cases) des menus ainsi que les éléments graphiques (après sélection dans le menu **DESIGNATION**). Le clavier permet d'entrer du texte (par exemple le nom d'un fichier) ainsi que des *raccourcis*, (les ITEM de certains menus ont été mis en *équivalence* avec des touches du clavier).

2.1 L'application CONSTRUCTION

Cette application permet de définir les contours géométriques du domaine à l'aide de points, segments, arcs de cercle, et de splines. On a de plus introduit deux autres entités de construction qui sont utiles dans certains cas: les droites et les cercles. Ces deux dernières entités ne servent d'ailleurs que dans cette application.

On peut construire toutes les entités, excepté les splines, à l'aide des théorèmes de géométrie élémentaire; quand les solutions sont multiples, les ambiguités sont levées en utilisant l'heuristique suivante: les points de désignation sont proches des points de tangence des éléments (point, droite, cercle, arc, segment).

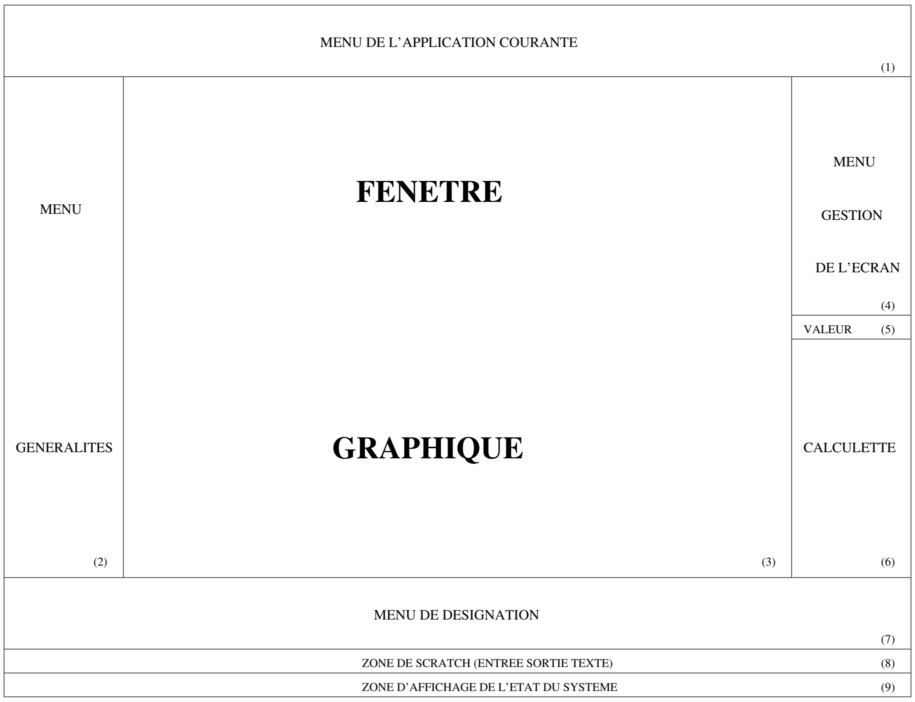

FIG. 2.1 - *Représentation de l'écran*

Une spline est définie comme une courbe C^1 passant par une liste de points. Elle sera fermée si le premier et le dernier point sont confondus.

On peut dupliquer toutes ces entités à l'aide des transformations affines suivantes: symétrie, rotation, homothétie, translation.

De plus on peut :

- arrondir les angles,
- couper les segments, arcs, splines par des points, segments, arcs, splines,
- détruire des entités à l'aide du menu **GENERALITES**

2.2 L'application PREP_MESH

A ce niveau les composantes connexes des bords des sous domaines de Ω sont connues. On appellera ces composantes connexes, des composantes.

Cette application permet de définir la discrétisation sur les entités définissant le bord du domaine et des sous domaines (frontière de matériaux), ainsi que des numéros de références sur les sous domaines, sur les lignes, sur les points; en vue d'entrer des données physiques différentes (exemple: plusieurs conditions aux limites, plusieurs matériaux).

Elle permet aussi de créer un fichier de données en vue d'interfacer le mailleur de MODULEF: APNOXX, en définissant les sous domaines comme une liste de composantes où la 1^{iere} est la composante extérieure (les autres composantes étant les composantes des trous), plus des lignes internes, et des points internes à forcer.

2.3 L'application EDIT_MESH

Au moment de la commutation on peut modifier la valeur par défaut de certains paramètres du mailleur. Le programme construit un maillage triangulaire ou quadrangulaire régle des sous domaines définis à l'étape précédente, ou bien de tous les sous domaines si aucun n'a été défini.

A ce niveau on peut éditer le maillage:

- ajouter des sommets internes aux sous domaines
- supprimer des sommets, des sous domaines
- retourner des arêtes internes d'un quadrilatère
- bouger des sommets
- régulariser ou rendre de *Delaunay* le maillage
- quadranguler le maillage
- transformer des sous domaines avec des symétries, rotations, homothéties, translations
- modifier les références des sommets, des arêtes, des sous domaines (régions);
- sauver ou restaurer un maillage (sous différentes formes: MESH structure interne à Emc^2, structure NOPO de modulef, structure simplifiée, etc)
- fissurer des lignes du maillage (les points de part et d'autre de la ligne sont dupliqués)
- renuméroter en vue de diminuer la taille du profil des matrices éléments finis P^1 ou Q^1.

2.4 Un exemple détaillé très simple

Pour changer les habitudes nous allons mailler un cercle unité plutôt qu'un carré. Il faut donc construire le cercle (un arc de 360°) , puis définir la discrétisation sur l'arc, puis construire le maillage. Ces 3 phases correspondent respectivement aux trois applications CONSTRUCTION, PREP_MESH, EDIT_MESH. La liste

des actions :

1. Exécuter emc2 puis entrer le numéro du périférique graphique (dépendant de la machine). Après l'initialisation on est implicitement dans l'application construction, et le mode de désignation est point tablette (PT_TABL). Pour construire l'arc:
   (a) on clique dans la case `ARC` du menu de construction (en haut),
   (b) on clique dans la case `CENTRE` du même menu pour définir le centre,
   (c) on clique dans la case `PT_XY` du menu désignation

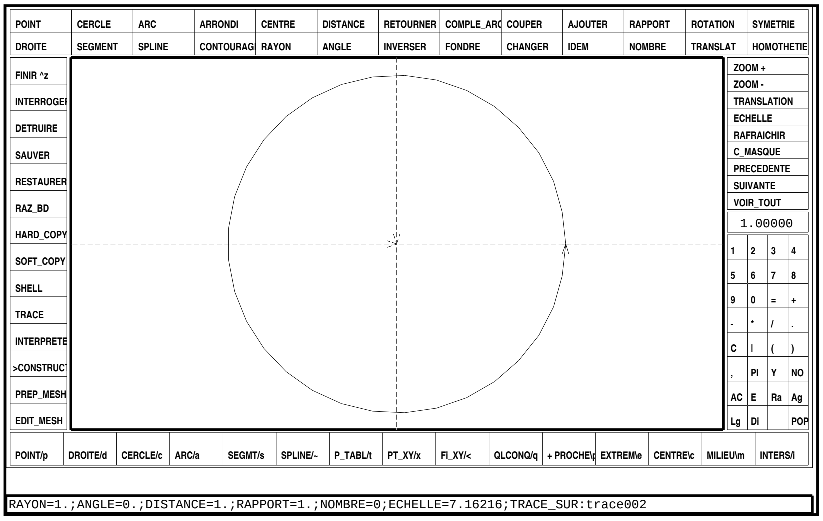

FIG. 2.2 - *Le cercle unité*

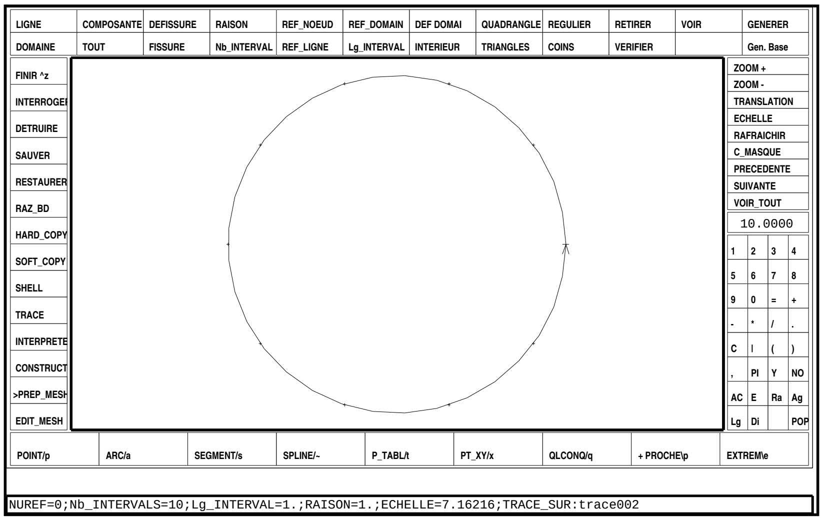

FIG. 2.3 - *Le cercle unité discrétisé*

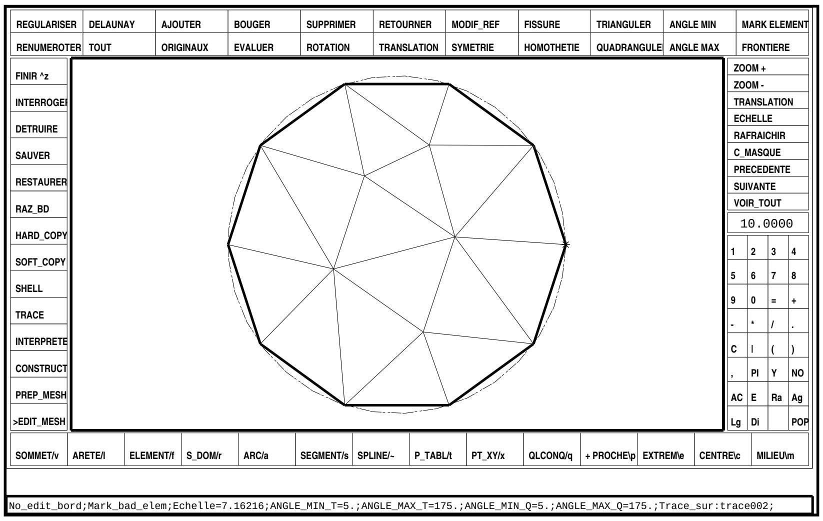

FIG. 2.4 - *Le cercle unité maillé*

(d) on tape au clavier: `0=0=` ce qui défini le centre au point (0,0)

(e) on clique dans la case `RAYON` du menu de désignation en haut

(f) on tape au clavier 1= ce qui défini le rayon à 1 (la variable d'état rayon est mis à 1, elle est affiche en bas de l'écran);

un cercle de arc de centre (0,0) de rayon 1 et d'angle 360° s'affiche. *Remarque: bien que la variable d'état "angle" soit égale 0, l'angle de l'arc est en fait égale à 360° ($0 \equiv 360\, mod\, 360$).*

2. Le cercle est petit (environ 2 cm) parce que l'echelle implicite est égale à 1, pour le voir plein écran (figure 2.2), il suffit de cliquer dans la case `VOIR_TOUT` du menu de gestion de l'écran (en haut à droite).

3. On a fini la construction du cercle. On passe au niveau de l'application PREP_MESH en cliquant dans cette case du menu général.

   Pour définir le nombre de point sur l'arc il suffit de:

   (a) cliquer dans la case `NB_NTERVAL`

   (b) taper le nombre d'intervalle `10=`

   (c) de désigner l'arc en cliquant dans la case `ARC` du menu de désignation (en bas)

   (d) de cliquer dans la fenêtre graphique près de l'arc

   Les points générés s'affichent sur l'écran (figure 2.3). On peut construire le maillage, en passant dans l'application EDIT_MESH, pour cela il suffit de cliquer dans la case `EDIT_MESH` du menu général, puis de taper quatre fois `(CR)`au clavier afin d'utiliser les options données par défaut aux 4 questions posées par le mailleur. Le maillage du cercle s'affiche sur l'écran (figure 2.4). On peut sauver le maillage avec la case `SAUVER` du menu général, en donnant :

   (a) le type du maillage à sauver par exemple en tapant `am_fmt(CR)`

   (b) le nom du préfixe du maillage par exemple en tapant `cercle(CR)`

   un fichier de nom *cercle.am_fmt* contenant le maillage est généré.

2.5 Un exemple moins trivial

Construction d'un maillage entre les 8 cercles tangents aux 3 cercles en pointillés (c.f. figure 2.5). On a coupé les arcs au voisinage des points de "presque tangence" en vue de raffiner le maillage. On montre les différentes étapes (figures 2.5,2.6 et 2.7), et des agrandissements du maillage figures 2.8 et 2.9. La figure 2.9 étant un très fort zoom (environ 360 fois) autour d'un point de presque tangence.

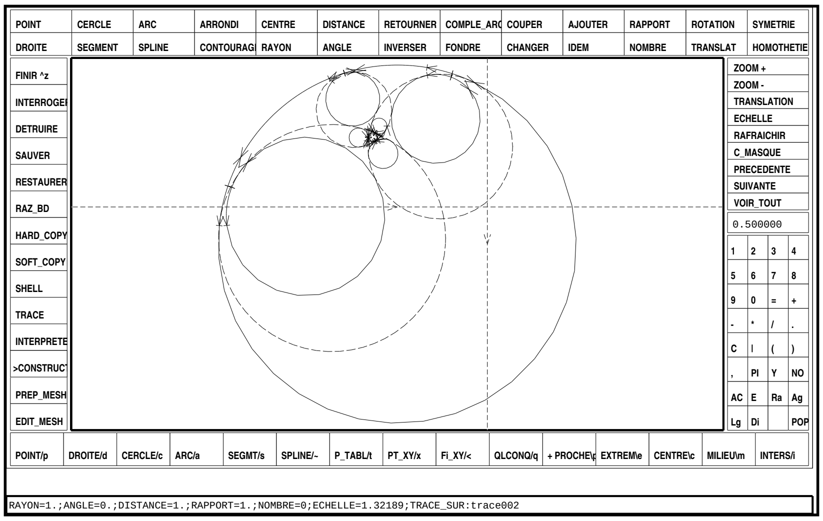

FIG. 2.5 - *La géométrie du domaine*

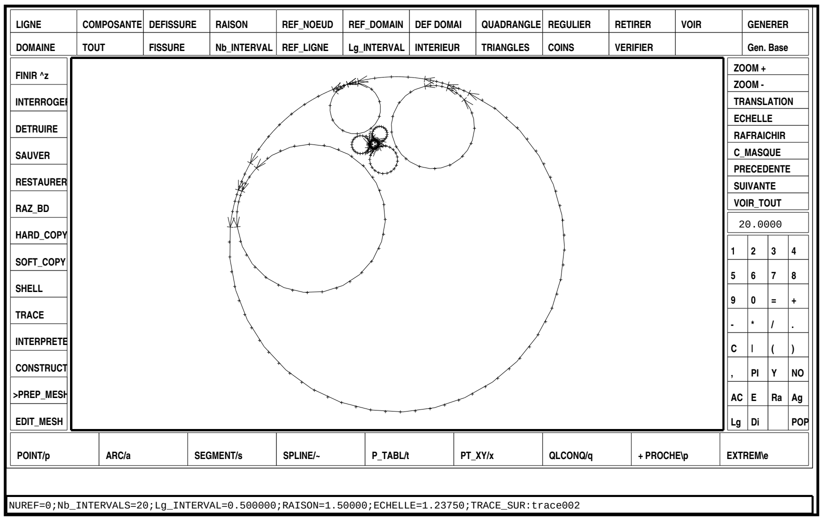

FIG. 2.6 - *La discrétisation des contours.*

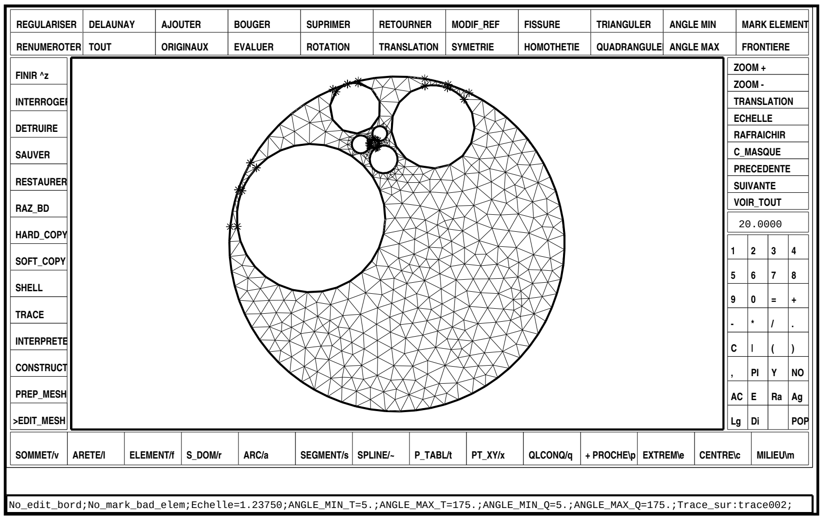

FIG. 2.7 - *Visualisation du maillage total.*

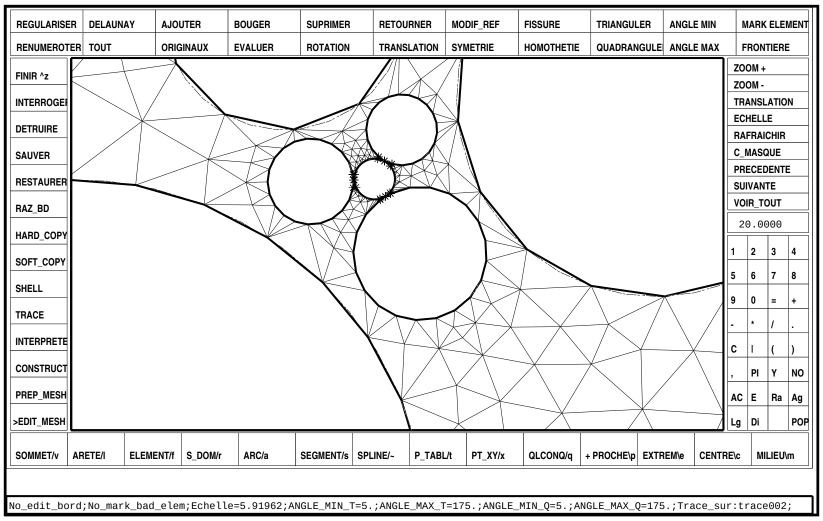

FIG. 2.8 - *Un premier zoom.*

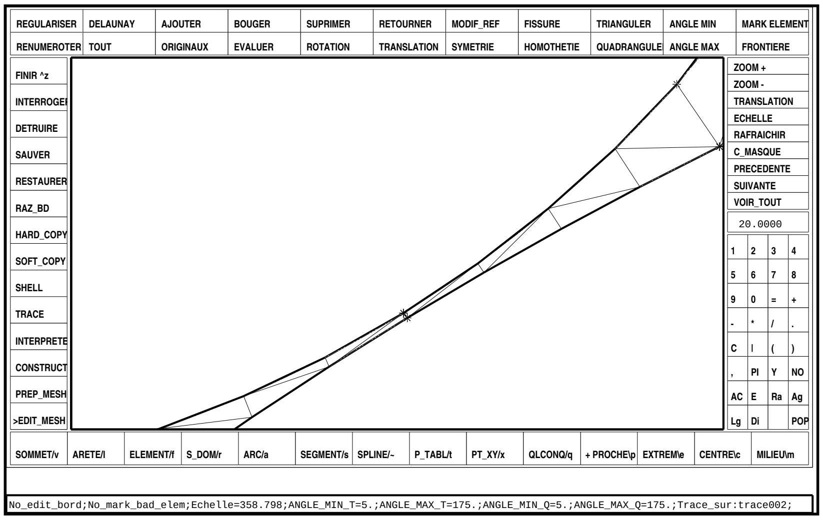

FIG. 2.9 - *Zoom autour l'un des points de presque tangence*

Chapitre 3

Les menus globaux

L'utilisateur avec les menus crée, édite, détruit, ... des objets qui sont stockés dans deux bases de données (BD), une pour la géométrie, l'autre pour les maillages.

3.1 Les menus de désignation

Ces menus (au bas de la fenêtre graphique) permettent de définir le *type de désignation graphique* (un point, droite, etc...). Ils dépendent de l'application courante (c.f. figures 3.1, 3.2, 3.3). Il est fondamental de bien comprendre les sorties de ce menu car tout le reste en dépend. Toutes ces sorties (exemple un cercle, un nombre, ou l'extrémite d'un arc) sont typées et les types définissent les entités syntaxiques de désignation connues par les grammaires qui dirigent les trois applications.

POINT	DROITE	CERCLE	ARC	SEGMENT	SPLINE	P_TABL	PT_XY	→ → →

← ← ←	FI_XY	QLCONQ	+_PROCHE	EXTREM	CENTRE	MILIEU	INTERS

FIG. 3.1 - *Le menu de désignation pour l'application CONSTRUCTION*

POINT	ARC	SEGMENT	SPLINE	P_TABL	PT_XY	QLCONQ	+_PROCHE	EXTREM

FIG. 3.2 - *Le menu de désignation pour l'application PREP_MESH*

SOMMET	ARETE	ELEMENT	S_DOM	ARC	SEGMENT	SPLINE	P_TABL	→ → →

← ← ←	PT_XY	QLCONQ	+_PROCHE	EXTREM	CENTRE	MILIEU

FIG. 3.3 - *Le menu de désignation pour l'application EDIT_MESH*

Les types générables sont :

- `<COORD>` point immédiat (point tablette, point XY ou extrémité, centre, milieu d'un élément de la BD)
- `<POINT>` point de la BD
- `<DROITE>` droite de la BD
- `<CERCLE>` cercle de la BD
- `<ARC>` arc de la BD
- `<SEGMENT>` segment de la BD
- `<SPLINE>` spline de la BD
- `<VALEUR>` valeur numérique (entrée avec la calculette)
- `<SOMMET>` sommet du maillage
- `<ARETE>` arête du maillage
- `<ELEMENT>` élément fini (triangle ou quadrangle) du maillage
- `<S_DOM>` sous domaine du maillage (composante connexe du maillage)

Ces menus dépendent de l'application en cours car ils servent à définir le type de l'élément que l'on veut désigner. Les ITEM des menus (leur "raccourci" clavier est entre parenthèses) peuvent se classer en plusieurs catégories:

1. Les item `POINT (p)`, `DROITE (d)`, `CERCLE (c)`, `ARC (a)`, `SEGMENT (s)`, `SPLINE (~)` permettent de désigner l'élément de la B.D. le plus proche du point cliqué, le type de désignation est défini par la nature de l'élément trouvé et par l'item des catégories (4), (5) (si il y a une incompatibilié, l'item de la catégorie (4) est forcé à `+P_PROCHE`.
2. Les item `P_TABL (t), PT_XY (x), FI_XY (<)` permettent d'entrer des points immédiats respectivement comme suit
   - ils sont définis en cliquant avec la souris dans la fenêtre graphique;
   - ils sont définis par une liste de couples de valeurs (coordonnées) entrée avec la calculette;
   - ils sont lus dans un fichier 'FORMATTED' contenant 2 coordonnées (real) par ligne jusqu'à la fin du fichier (EOF), le nom du fichier est demandé dans la zone scratch;

   Remarque: dans les 2 derniers cas le point généré peut-être hors de la fenêtre graphique;
3. l'item `QLCONQ (q)` n'oriente pas la désignation (on prendra l'élément de la BD le plus proche du point cliqué), le type de désignation est défini par la nature de l'élément trouvé et par l'item des catégories 4,5 utilisé;
4. l'item `+_PROCHE` permet de désigner effectivement l'élément de BD et le type de la désignation est celui de l'élément de BD,
5. les item `EXTREM(ITE) (esc e), CENTRE (esc c), MILIEU (esc m)` permettent de définir ce que l'on va désigner de l'élément de BD (c.f. (1) et (3)) : on désigne le point (de type de désignation: `<COORD>`) qui est respectivement l'extre'mité, le centre, le milieu de l'élément de BD désigné si cela a un sens sinon on force le mode de désignation principal à quelconque (cf 3.) tout en conservant l'item de la catégorie 5. ;
6. l'item INTERSECTION, permet de désigner le point immédiat qui est l'intersection de 2 éléments de BD désignés par la suite (le type de désignation généré est `<COORD>`);
7. les item `ELEMENT(f), ARETE(l), SOMMET(v), S_DOM(r)` permettent de désigner respectivement l'élément (triangle ou quadrangle) contenant le point cliqué (pour le type de désignation: `<ELEMENT>`), l'arête la plus proche du point cliqué qui doit être dans un élément (pour le type de désignation: `<ARETE>`), le sommet le plus proche du point cliqué qui doit être dans un él'ement (pour le type de désignation: `<SOMMET>`), le sous domaine contenant le point cliqué (pour le type de désignation: `<S_DOM>`).

Important: En fait la désignation n'est effective que lorsque l'on a cliqué sur un élément de la fenêtre graphique, en effet on peut cliquer autant de fois que l'on veut les cases du menu de façon à obtenir la combinaison désirée de case(s) marquée(s). Une désignation définit le type de désignation et elle conserve les coordonnées du curseur au moment du clic, souvent ces coordonnées sont utiles pour l'application en cours, car elles servent à lever des ambiguitées sémantiques, plusieurs solutions au problème traité, exemple:

- on veut construire un cercle tangent à 3 cercles, il y a généralement 8 solutions, on choisira la solution telle que les points de tangences soient les plus proches des points de désignation;
- l'intersection de deux cercles donne deux points, dans INTERSECTION, on lève l'ambiguité en ne retenant que le point le plus proche de celui ayant servi à désigner le deuxième cercle.

Si l'on essaie de désigner quelque chose qui n'existe pas, on obtient le message *on n'a rien désigné*, ce qui ne produit aucun effet.

Suivant les applications les désignations possibles ne sont pas les mêmes car on n'a pas besoin de tous les types de désignation (ex: les cercles et les droites ne sont utilisés que dans l'application construction, les éléments, arêtes, sommets et sous domaines n'existent que dans l'application EDIT_MESH)

3.1.1 Les raccourcis

Les "raccourcis" clavier correspondant aux ITEM des catégories 1,2,3,4,7 sont équivalents à 2 cliques : un premier dans la case de l'ITEM , et le suivant sur le point du curseur si celui ci est dans la fenêtre graphique (on a donc changé le mode de désignation et on a désigné l'élément le plus proche du curseur).

3.1.2 Utilisation pratique

La désignation d'un élément se fait en cliquant près de cet élément (on recherche le plus proche du clique). Quand on clique un menu, il est marqué par l'apparition du signe > devant le texte du menu. Dans les menus de désignation, il peut y avoir deux cases de marquées à la fois, les combinaisons sont:

```
QLCONQ      ( +_PROCHE, EXTREMITE, CENTRE, MILIEU )
POINT       ( +_PROCHE)
DROITE      ( +_PROCHE)
CERCLE      ( +_PROCHE, CENTRE )
ARC         ( +_PROCHE, EXTREMITE, CENTRE, MILIEU )
SEGMENT     ( +_PROCHE, EXTREMITE, MILIEU )
SPLINE      ( +_PROCHE, EXTREMITE )
PT_TAB      ( +_PROCHE)
PT_XY       ( +_PROCHE)
SOMMET      ( +_PROCHE)
ARETE       ( +_PROCHE)
ELEMENT     ( +_PROCHE)
S_DOM       ( +_PROCHE)
```

Les autres combinaisons ne sont pas valides. Une désignation n'est effective que lorsque l'on a cliqué un élément à l'écran : en effet on peut cliquer autant de fois que l'on veut les cases du menu de façon à obtenir la combinaison désirée de case(s) marquée(s).

3.2 Le menu calculette

Ce menu (c.f. figure 3.4) peut aussi être considéré comme un menu de désignation. Il permet de définir une valeur numérique qui est de type `<VALEUR>`.

On peut écrire n'importe quelle expression FORTRAN. Le calcul effectif de la valeur est effectué sur le clique de l'ITEM = (de raccourci =) ou en tapant (`CR`).

les ITEM, avec leur raccourci clavier entre parenthèses sont :

1	2	3	4
5	6	7	8
9	0	=	+
−	∗	/	.
C	|	(	)
,	*PI*	*Y*	*NO*
AC	*E*	*Ra*	*Ag*
Lg	*Di*		*POP*

On passe d'un menu à l'autre en cliquant la case POP

SIN	*COS*
TAN	*ATAN*
*ATAN*2	*EXP*
LOG	*LOG*10
SQRT	*MOD*
ABS	*SIGN*
INT	*NINT*
MIN	*MAX*
	POP

Fig. 3.4 - *Les deux menus calculettes*

1	(1)	les chiffres
2	(2)	
3	(3)	
4	(4)	
5	(5)	
6	(6)	
7	(7)	
8	(8)	
9	(9)	
.	(.)	point décimal
+	(+)	
-	(-)	
∗	(∗)	
/	(/)	
)	())	
(	(()	
,	(,)	virgule pour séparer, les arguments des fonctions
=	(=)	
|	(|)	Opérateur puissance.
PI		$= \pi$
Y		$= 0$
NO		$= 1$
E	(E)	Exposant des nombres en format exponentiel.
C	(Z)	Raz de l'accumulateur de la calculette.
AC	(R)	Raz de la calculette.
Ra		= Rayon d'un élément désigné (`<ARC>` ou `<CERCLE>`).
Ag		= Angle d'un élément désigné (`<ARC>`) ou angle entre deux `<DROITES>`.
Lg		= Longueur d'un élément désigné (`<ARC>` ou `<SEGMENT>` ou `<SPLINE>`).
Di		= distance entre un `<POINT>` et un `<POINT>`, une `<DROITE>`, un `<CERCLE>`, ou un `<ARC>`; ou distance entre deux `<DROITE>` parallèlles, ou distance entre deux `<CERCLE>`.
POP		permet l'affichage du menu calculette des fonctions FORTRAN SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2, EXP, LOG, LOG10, INT, NINT, MOD.

Exemples pour entrer les nombres:

1.234567E9 il suffit de taper au clavier `1.234567E9(CR)`

0.5 il suffit de taper `1/2=`

3.14159... il suffit de cliquer dans la case `PI(CR)`

2×rayon du cercle il suffit de taper `2*`, de cliquer dans la case `Ra`, puis de désigner le `<cercle>` (en cliquant de la case cercle du menu de désignation puis en cliquant près du cercle), pour finir il suffit de taper `(CR)` au clavier

3.3 Le menu généralités

Ce menu (c.f. figure 3.5) permet d'exécuter des actions générales: changer d'application CONSTRUCTION, PREP_MESH, EDIT_MESH, sauver, restaurer, copier l'écran, ou exécuter une commande système. Pour lancer une action, il suffit de cliquer dans l'item correspondant.

FINIR
INTERROGER
DETRUIRE
SAUVER
RESTAURER
RAZ_BD
HARD_COPY
SOFT_COPY
SHELL
TRACE
INTERPRETE
CONSTRUCTI
PREP_MESH
EDIT_MESH

FIG. 3.5 - *Le menu général*

Les ACTIONS du menu général sont :

3.3.1 Sortir du programme

`FINIR`

Termine l'exécution du programme.

3.3.2 Interroger la base de données

`INTERROGER`

Permet d'interroger les éléments que l'on désignera. Leurs caractéristiques seront affichées dans la zone d'entrées sorties ou "scratch".

3.3.3 Détruire des élément de la base de données

`DESTRUCTION`

Permet de détruire (uniquement dans l'application CONSTRUCTION) les éléments de la BD que l'on désignera. Attention: si l'on détruit un POINT, on vérifie si ce POINT est utilisé pour définir une SPLINE, si oui, on le retire aussi de la SPLINE.

3.3.4 Sauver la base de données

`SAUVER`

Permet de sauver les B.D. dans un fichier. Le nom du fichier est composé d'un préfixe et d'un suffixe. Le préfixe est demandé par le système dans la zone scratch (figure 2.1), le suffixe peut être, suivant les cas: `.emc2_bd, .mesh, .nopo, .am, .am_fmt, .amdba` et dépend de l'application courante.

- Dans l'application CONSTRUCTION, on sauve les éléments de construction avec leurs numéros de références (points, droites, cercles, arcs, segments, splines) ainsi que l'état du système; le suffixe est `.emc2_bd`.

- Dans l'application PREP_MESH, on sauve la même chose que dans CONSTRUCTION plus la définition des domaines; le suffixe est `.emc2_bd`.

- Dans l'application EDIT_MESH, on sauve différents types de maillage. Le type est entré dans la zone scratch avant d'entrer le préfixe du fichier. Les types connus implicitement sont :

  mesh (type par défaut) on sauve la BD_MESH et ses reférences à la BD définissant la géometrie (définissant les contours des sous domaines). Le suffixe est `.mesh`.

  nopo on extrait de la BD_MESH la structure de données NOPO (Modulef) que l'on écrit dans un fichier de suffixe `.nopo`. (on perd les courbes définissant les contours).

  am on extrait de la BD_MESH une structure simplifiée de type AM (c.f. appendice 9.4.1) que l'on écrit dans un fichier non formaté de suffixe `.am` (on perd les courbes définissant les contours ainsi que les références des arêtes).

  am_fmt idem am, mais le stockage est fait dans un fichier formaté de suffixe est `.am_fmt` (c.f. appendice 9.4.2).

  amdba on extrait de la BD_MESH une structure de type AMDBA (c.f. appendice 9.4.3) que l'on écrit dans un fichier formaté de suffixe `.amdba` (on perd les courbes définissant les contours et certaines références).

  Remarque: il est possible de sauver les maillage sous d'autres formes en modifiant la subroutine `ecrmsh`.

3.3.5 Restaurer la base de données

`RESTAURER`

Permet de restaurer la B.D. d'un fichier. Le nom du fichier est composé d'un préfixe et d'un suffixe. Le préfixe est demandé par le système dans la zone scratch (figure 2.1), le suffixe peut être: `.emc2_bd, .mesh, .nopo, .am, .am_fmt, .amdba` suivant les cas. L'information restaurée dépend de l'application courante.

- Dans l'application CONSTRUCTION, on restaure les éléments de construction avec leurs numéros de références (points, droites, cercles, arcs, segments) ainsi que l'état du système; le suffixe est `.emc2_bd`.

- Dans l'application PREP_MESH, on restaure la même chose que dans CONSTRUCTION plus la définition des domaines; le suffixe est `.emc2_bd`.

- Dans l'application EDIT_MESH, on restaure différents types de maillage. Le type qui est entré dans la zone scratch avant d'entrer le préfixe du fichier. Les types connus implicitement sont :

  mesh restaure un maillage et ses contours (les définitions géométiques de ceux-ci sont conservées); le suffixe est `.mesh`.

  nopo on construit la BD_MESH à partir d'une structure de type NOPO (Modulef) (on reconstruit les courbes définissant les contours avec des segments, on perd l'édition des fissures) ; le suffixe est `.nopo`.

  am on construit la BD_MESH à partir d'une structure de type AM (c.f. appendice 9.4.1). (on on reconstruit les courbes définissant les contours avec des segments; on a perdu l'édition des fissures et les références d'arêtes); le suffixe est `.am`.

  am_fmt idem am, mais le fichier est formaté; le suffixe est `.am_fmt` (c.f. appendice 9.4.2).

  amdba on construit la BD_MESH à partir d'une structure de type AMDBA (c.f. appendice 9.4.3). (on perd les courbes définissant les contours et les références d'arêtes) (on on reconstruit les courbes définissant les contours avec des segments et on a perdu l'édition des fissures); le suffixe est `.amdba` .

  Remarque: il est possible de restaurer des maillages écrits sous une autre forme en modifiant la subroutine `addmsh`.

  Attention: le maillage lu n'est pas recollé avec les maillages préexistants.

3.3.6 Mise à zéro

`RAZ_BD`

Permet de détruire tout ce qu'il y a en mémoire dans le 2 bases de données.

3.3.7 Imprimer l'écran

`HARD_COPY`

Hard copy de ce qu'il y a sur l'écran (avec la résolution de l'écran, imprime la bitmap de l'écran). *Dépend du système et de l'environnement (c.f. subroutine* `hardcp`*).*

3.3.8 Redessiner la fenêtre sur un autre périphérique graphique

`SOFT_COPY`

On réaffiche ce qu'il y a sur l'écran sur un autre périphérique de Fortran 3d. *Dépend du système et de l'environnement (c.f. subroutine* `softcp`*)*

3.3.9 Exécuter une commande système

`SHELL`

Permet d'exécuter une commande système (shell), le système demande d'entrer la commande ou les commandes dans une chaîne de caractères. *Dépend du système (c.f. subroutine* `exec`*).*

3.3.10 Sauvegarder une session

`TRACE`

Permet de sauver dans un fichier de nom composé d'un préfixe donné par utilisateur et d'un suffixe `.trace_emc2` toutes les actions de l'utilisateur. Si le préfixe est précédé par un +, on fait un 'append' au fichier de trace (on écrit à partir de la fin). Par défaut un fichier de trace est ouvert, il a pour préfixe `tracexxxx`, où xxxx est un numéro de version pour ne pas écraser la trace précédent).

3.3.11 Réexécuter une session

`INTERPRETE`

Interprète un fichier de trace, de préfixe donné par utilisateur.

3.3.12 Commuter dans l'application CONSTRUCTION

`CONSTRUCTIO`

Demande de commutation dans l'application de CONSTRUCTION. Après la commutation, on se trouve en désignation de P_TABL.

3.3.13 Commuter dans l'application PREP_MESH

`PREP_MESH`

Demande de commutation dans l'application PREPARATION DES DONNEES POUR MAILLAGE. Après la commutation, on se trouve en désignation de P_TABL.

3.3.14 Commuter dans l'application EDIT_MESH

`EDIT_MESH`

Demande de commutation dans l'application EDITION DE MAILLAGE (EDIT_MESH). Au moment de la commutation après avoir repondu aux 4 questions, on *"triangule"* tous les domaines définis (si aucun domaine n'est défini on maille tout). La triangulation dépend de 4 paramètres (par défaut on tape un `(CR)`au clavier par paramètre):

1. un paramètre *debug* (0 par defaut) qui défini deux paramètres :
   - impression =debug mod 10 : 0 => pas d'impression,à 9 => beaucoup d'impression
   - debugging graphique = debug / 10 : 0 => pas de debugging,à 9 => tout les phases de l'algorithmes de maillage sont affichées (*très long*).
2. un paramètre *puis* d'épaisseur de raffinements (plus *puis* est petit plus les zônes de raffinement se propageront loin, $puis = .25$ par défaut (les valeurs raisonables sont comprises entre .1 et 10.);
3. un paramètre *coef* de densité de triangles (le nombre de triangles générés est en $O(coef^2)$, $coef = .75$ par défaut).
4. de plus on demande si on active ou non une régularisation (par défaut elle est faite);

Après la commutation, on se trouve en désignation de `PT_TABL`.

3.4 Le menu gestion de l'écran

Il est représenté par la figure 3.6 et il est présent pour toutes les applications. Il sert à gérer l'affichage des 2 bases de données. Le contexte d'affichage consiste en la mémorisation du masque d'affichage (ce masque est un rectangle dans l'espace sujet et est calculé de manière à conserver l'égalité des échelles en X et Y). La pile des contextes d'affichage est en fait une pile circulaire à 8 niveaux. Ces contextes sont initialisés par : échelle 1, masque centré à l'origine.

ZOOM +
ZOOM -
TRANSLATION
ECHELLE
RAFRAICHIR
C_MASQUE
PRECEDENTE
SUIVANTE
VOIR_TOUT

FIG. 3.6 - *Le menu gestion de l'écran*

Les ACTIONS du menu gestion de l'écran sont:

3.4.1 Agrandir le dessin

`ZOOM + <COORD> <COORD>`

Le rectangle défini par deux `<COORD>` est affiché plein écran et on empile ce nouveau contexte dans la pile d'affichage. En général on clique 2 fois pour définir deux coins opposés du rectangle.

3.4.2 Réduire le dessin

`ZOOM - <COORD> <COORD>`

On affiche ce qui il y a actuellement plein écran dans le rectangle défini par deux `<COORD>` et on empile ce nouveau contexte dans la pile d'affichage.

3.4.3 Translater la fenêtre graphique

`TRANSLATION <COORD> <COORD>`

Translate la fenêtre de visualisation d'un vecteur allant du premier `<COORD>` au deuxième `<COORD>` et on empile ce nouveau contexte dans la pile d'affichage.

3.4.4 Définir une échelle

`ECHELLE`

Change l'échelle de visualisation et on empile ce nouveau contexte dans la pile d'affichage (le centre de la fenêtre est inchangé).

3.4.5 Redessiner la fenêtre graphique

`RAFRAICHIR`

Réaffiche le dessin courant. Cette action est très utile car bien souvent il reste des scories, qui sont effacées par cette action.

3.4.6 Centrer le dessin

`C_MASQUE <COORD>`

Centre la fenêtre sur le point désigné (`<COORD>`) et on empile ce nouveau contexte dans la pile d'affichage.

3.4.7 Redessiner la vue précédente

`PRECEDENTE`

Redescend de un dans la pile des contextes d'affichage et affiche (on retrouve le contexte d'affichage précédent). Remarque :ce contexte d'affichage existe toujours car les huit contextes sont initialisés par défaut.

3.4.8 Redessiner la vue Suivante

`SUIVANTE`

Remonte de un dans la pile des contextes d'affichage et affiche (on retrouve le contexte d'affichage suivant, voir remarque précédente).

3.4.9 Dessiner tout le dessin

`VOIR_TOUT`

Calcule le masque de façon à voir toute la base de données et affiche celle-ci plein écran.

Chapitre 4

L'application construction

4.1 Présentation

Cette application est dirigée par le menu de l'application (figure 4.1) en haut de l'écran et le menu de désignation (figure 3.1) en bas de l'écran. Elle permet de construire des objets 2D définis à partir des points, segments, arcs et splines; les cerles et les droites ne servant que d'artifices de construction. Elle permet de construire et manipuler ces objets:

- en utilisant la géométrie élémentaire pour définir les points, droites, cercles, segments, arcs (exemple: en utilisant la propriété de tangence, on peut définir un cercle tangent à 3 droites, ou bien une droite tangente à 2 cercles).
- en utilisant des splines (une suite de points) pour définir des courbes fermées ou non;
- en coupant les objets entre eux (excepté les droites et les cercles qui sont seulement des objets de construction). On peut par exemple, couper par une droite mais celle-ci ne sera pas affectée;
- en arrondissant des angles;
- en dupliquant n fois ces objets à l'aide des transformations affines classique: symétrie, rotation, homothétie;
- en faisant des *"contourages"* de figures;
- en inversant des arcs ou des segments;
- etc

POINT	CERCLE	ARC	ARRONDI	CENTRE	DISTANCE	RETOURNER	→ → →
DROITE	SEGMENT	SPLINE	CONTOURAGE	RAYON	ANGLE	INVERSER	→ → →

← ← ←	COMPLE_ARC	COUPER	AJOUTER	RAPPORT	ROTATION	SYMETRIE
← ← ←	FONDRE	CHANGER	IDEM	NOMBRE	TRANSLAT	HOMOTHETIE

FIG. 4.1 - *Le menu de l'application CONSTRUCTION*

4.2 Utilisation

Nous rappelons que grâce aux menus de désignation on peut désigner des points, droites, cercles, arcs, segments, splines, coordonnées (points immédiats, extrémités, etc...). Donc nous noterons par <POINT>, <DROITE>, <CERCLE>, <SEGMENT>, <ARC>, <SPLINE>, <COORD> le fait de désigner un de ces éléments ou une partie de ceux-ci, de plus on crée une <VALEUR> en utilisant la calculette ou ses raccourcis (voir les sections 3.1,3.2 pour plus de détails).

4.2.1 Méta syntaxe de description des actions

Nous utiliserons, dans la méta syntaxe de description du langage de construction, les symboles suivants:
Nous appellerons <xxx> une suite d'actions qui seront définies par <xxx> = ... ;

- A B pour définir la concaténation de A et B (et) "*on attend* A *puis* B".
- A|B pour définir une alternative (ou) "*on attend A ou B*" (action moins prioritaire que la concaténation).
- * pour la répétition de l'expression qui suit (0 ou plusieurs fois) .
- " pour rien (ce qui permet de faire quelque chose dans tous les cas)
- les parenthèses ( ) pour changer les priorités

Exemple: L'expression: <xxx>= (A | ") *(B|C) E ; nous permet d'entrer des suites de lettres (chaîne) commencant ou non par A formé de B ou de C en suite et ce terminant par E.
Exemple de chaînes admissibles:

ABBCCCCE	E	AE	CBCBCBCBCBCBCBE
ABBBBBBBE	BE	ABE	BCBCBCBCBCBCBCE
BBBBBBBBBE	CE	ACE	CCCCCCCCCCCCCCE

Nous appellerons

- <point_coor> la désignation de <POINT> ou <COORD>;
- <contrainte> la désignation de <POINT>, <COORD>, <DROITE>, <CERCLE>, <SEGMENT> ou <ARC>;
- <elements> la désignation de <POINT>, <COORD>, <DROITE>, <CERCLE>, <SEGMENT>, <ARC> ou <SPLINE>.
- etc...

ou encore, dans la méta syntaxe :

```
<point_coor> = <POINT>      | <COORD>;
<contrainte> = <point_coor> | <DROIT> | <CERCLE> | <ARC>
                            | &lt;SEGMENT&gt;;
<element>    = <contrainte> | <SPLINE>;
```

Dans la définition de <contrainte>, l'ARC est assimilé au cercle qui le supporte, de même que SEGMENT est assimilé à la droite qui le supporte.

Remarque: En base de données les éléments sont décrits par:

POINT	(a,b) avec: x=a; y=b
DROITE	(a,b,c) tels que ax+by+c=0
CERCLE	centre(a,b) et rayon r
SEGMENT	(P1,P2) avec P1=(x1,y1) et P2=(x2,y2)
ARC	centre en P1 passant par P2 et d'angle α, avec P1=(x1,y1) et P2=(x2,y2)
SPLINE	(nombre de points, tête de la liste des points)

Au niveau de l'application CONSTRUCTION tous les menus autres que le menu de CONSTRUCTION sont filtrés par les applications DESIGNATION, CALCULETTE, GENERALITE, GESTION DE L'ECRAN GRAPHIQUE. Donc le fait de cliquer dans ces menus n'a aucune action directe sur l'application (exepté le changement de type de désignation ou l'entrée d'une valeur). Les cases du menu de l'application et les désignations gouvernent l'application CONSTRUCTION. Par la même nous noterons `XXXX` : le fait de cliquer dans la case XXXX du menu de l'application courante (en haut). Il n'y aura aucune confusion avec des cases de même nom d'un autre menu.

exemple:

- le fait de désigner un cercle est noté `<CERCLE>`
- le fait de cliquer dans l'item CERCLE du menu CONSTRUCTION est noté `CERCLE`

il ne faut pas confondre ces deux "cercles", de même si l'on clique dans l'item CENTRE du menu désignation on désignera des centres d'arcs ou de cercles ce qui donnera des `<COOR>` par contre si l'on clique dans l'item CENTRE du menu CONSTRUCTION (en haut) on obtient `CENTRE`.

4.2.2 Modification de l'état du système

L' état du système est un ensemble de variables qui sont utilisées par l'application, ces variables globales à l'application sont :

- %DISTANCE permet définir une distance signée qui est utile pour des constructions de point, droite, segment, etc...
- %RAYON permet de définir le rayon d'un arc ou cercle, etc...
- %ANGLE permet de définir l'angle d'arcs ou de 2 droites, etc...
- %RAPPORT permet de définir le rapport d'une homothétie, etc...
- %NOMBRE permet de définir le nombre de duplications lors des transformations.

Ces variables sont modifiables presque à tout moment.

Nous appellerons `<modif etat>` une modification de l'état du système , c'est à dire une modification de la valeur affectée à l'une des variables

```
<modif etat>  = *( <distance> | <rayon> | <angle> | <rapport> | <nombre> )
```

- Modification de %DISTANCE: `<distance>= DISTANCE (<VALEUR> | IDEM)`
- Modification de %RAYON: `<rayon> = RAYON (<VALEUR> | IDEM)`
- Modification de %ANGLE: `<angle> = ANGLE (<VALEUR> | IDEM)`
- Modification de %RAPPORT: `<rapport> = RAPPORT (<VALEUR> | IDEM)`
- Modification de %NOMBRE: `<nombre> = NOMBRE (<VALEUR> | IDEM)`

IDEM Permet de garder la valeur précédente de la variable d'état.

4.2.3 Construction de points

```
POINT
      *(  <COORD>
        | <POINT>  (  <CERCLE> | <DROITE>
                    | <ARC>    | &lt;SEGMENT&gt; | <SPLINE> )
        | <DROITE>   <element>
        | <CERCLE>   <element>
        | <SPLINE>   <element>
       );
```

- Première alternative, `<COORD>`: Création du point de coordonnées immédiates par action des menus de désignation: `P_TABL`, `PT_XY`, `FI_XY`, `INTERS`, `EXTREM`, `MILIEU`, `CENTRE`.

- Autres alternatives: On construit un point après avoir désigné deux éléments.

  - Si l'un des élément est un `<point>` soit P le point projection de ce `<point>` sur l'autre élément
    - si cet autre élément est une `<droite>` alors on crée un point à une distance %DISTANCE de P sur la `<droite>` dans la direction de la `<droite>`
    - sinon on crée le point P.
  - Si aucun des deux éléments n'est un `<point>`, on crée le point d'intersection de ces deux éléments. Comme il y a en général deux intersections entre une `<droite>` et un `<cercle>` ou entre deux `<cercles>`, le choix du *bon point* d'intersection est fait en choisissant le plus proche des points ayant servis à désigner les éléments.

Exemple de construction de points
On commence par cliquer dans la case POINT du menu construction pour se mettre en mode de construction de points puis

- en cliquant dans la case `PT_TABL` du menu désignation, et puis pour chaque clique dans la fenêtre graphique on construit un point

- en cliquant dans la case `PT_XY` du menu désignation ont construit un point pour chaque couple de nombres entrés avec la calculette *(tapés au clavier et suivis de = ou* `(CR)`*)*

- en cliquant dans la case `FI_XY` et en entrant un nom de fichier, on construit un point par couple de nombres qu' il y a dans le fichier formaté.

- on construit des points intersection de deux éléments (droite, segment, cercle, arc, spline) en désignant un couple de ces éléments.

4.2.4 Construction de droites

```
DROITE  *(  <contrainte> <contrainte>
          | IDEM *&lt;SEGMENT&gt;
         );
```

Si on a comme contraintes (indépendamment de tout ordre) :

- `<CercleOuPoint>`:= `<cercle>` — `<point>`;

- `<CercleOuPoint>` `<CercleOuPoint>`: la droite créée est tangente aux deux `<cercle>` (aux points les plus proches des points ayant servi à désigner les deux `<cercle>`)

- `<droite>` `<droite>`:

  - si elles s'intersectent: la droite créée est bissectrice des autres `<droite>`.
  - si elles sont confondues: on crée une droite parallèle à une distance %DISTANCE du coté du point de désignation.
  - sinon: on crée la droite parallèle aux deux `<droite>`, à égale distance de celles-ci.

- `<droite>` `<cercle>`: la droite créée est parallèle à la `<droite>` et est tangente au `<cercle>` en un point le plus proche de celui ayant servi à désigner le `<cercle>`.

- `<droite>` `<Point>`: la droite créée fait un angle %ANGLE avec la `<droite>` et passe par le `<point>`.

Dans l'autre alternative (IDEM), on construit la droite s'appuyant sur les $*$`<segment>`.

4.2.5 Construction de cercles

```
CERCLE
  *(  <rayon> *( CENTRE <point_coor> | <contrainte> <contrainte>)
    | <contrainte> (  <rayon>  <contrainte>
                   | <contrainte> ( <rayon> | <contrainte> ))
    | CENTRE <point> *(<rayon> | <contrainte>)
    | IDEM *<ARC>
   );
```

Explication des différents cas (indépendamment de l'ordre)

- <rayon> CENTRE <point>: cercle de centre le point <point> et de rayon %RAYON.
- <rayon> <contrainte> <contrainte>: cercle de rayon donné par %RAYON et ayant deux contrainte de tangence par rapport aux deux éléments désignés.
- <contrainte> <contrainte> <contrainte>: cercle ayant trois contraintes de tangence par rapport aux trois éléments désignés.
- CENTRE <point> <contrainte>: cercle de centre donné tangent à la contrainte désignée.
- IDEM $*$ <arc>: cercle s'appuyant sur les $*$ <arc> (même centre, même rayon que les $*$<arc>).

4.2.6 Construction d'arcs

```
ARC *(  CENTRE <point> *(   <angle> <contrainte>
                          | <rayon>
                          | <contrainte> (<angle> | <contrainte>))
      | <rayon> *(  CENTRE <point>
                 | <contrainte> <contrainte> )
      | <contrainte> (   <rayon> <contrainte>
                       | <contrainte> (<rayon> | <contrainte>))
      | IDEM *<CERCLE>
    );
```

Explication des différents cas (indépendamment de l'ordre). Pour la construction des ARCS, on crée d'abord le cercle ayant les mêmes <contrainte> et on prend un arc sur ce cercle.

- CENTRE <point> <angle> <contrainte>: arc de centre donné commençant au point de tangence avec la <contrainte> et faisant un <angle> à partir de ce point.
- CENTRE <point> <rayon>: arc de centre donné de rayon donné %RAYON d'angle 2π
- CENTRE <point> <contrainte> <contrainte>: arc de centre donné commençant au point de tangence avec la première <contrainte> et finissant à l'aplomb du point de tangence d'avec la deuxième <contrainte> (en tournant dans le sens positif).
- <rayon> <contrainte> <contrainte>: arc de rayon donné %RAYON commençant au point de tangence avec la première <contrainte> et finissant au point de tangence d'avec la deuxième <contrainte>.
- <contrainte> <contrainte> <contrainte>: arc tangent aux trois <contrainte> reliant les trois points de tangence aux <contrainte> dans l'ordre des <contrainte>.
- IDEM $*$<cercle>: crée un arc d'angle $2\ \pi$ sur chaque <cercle> désigné.

4.2.7 Construction de segments

```
SEGMENT *( <contrainte> <contrainte> );
```

Explication des différents cas (indépendamment de l'ordre).

Pour la construction des SEGMENTS, on commence par créer la DROITE ayant les mêmes contraintes puis on prend un segment de cette droite reliant les deux points de tangence aux deux containtes si ils existent, sinon on ce sert de la variable d'état

Par exemple :

- `<cercle> <cercle>`: Le segment est tangent aux deux `<cercle>` (aux points les plus proches des points ayant servis à désigner les deux `<cercle>`).
- `<droite> <droite>`:
  - si elles s'intersectent: Le segment est sur la bissectrice des deux `<droite>` et est longueur %DISTANCE.
  - si les deux `<droite>` sont confondues: on crée un segment parallèle à une distance %DISTANCE du coté du point de désignation (Il est de longueur %DISTANCE ou de même longueur que `<droite>` selon que `<droite>` est une DROITE ou un SEGMENT.
  - sinon: Le segment est entre les deux `<droite>` parallèles, à égale distance (il est de longueur %DISTANCE).
- `<droite> <cercle>`: Le segment est parallèle à la `<droite>` désignée et est tangent au `<cercle>` en un point le plus proche de celui ayant servi à désigner le `<cercle>` (il est de longueur %DISTANCE et commence au point de tangence avec le `<cercle>`).

4.2.8 Construction de splines

```
SPLINE <point> *<point>;
```

Construit la spline (cubique par morceaux) passant par les `<points>`. On peut faire une spline fermée en définissant le dernier point identique au premier (en le désignant).

4.2.9 Transformations géométriques

ROTATION, TRANSLAT, HOMOTHETIE, SYMETRIE définissent des transformations que l'on va appliquer sur les éléments désignés.

```
ROTATION    <modif etat> ( CENTRE | " ) <point_coor> <modif etat>  *<element>
TRANSLAT    <point_coor> <point_coor>    *<element>
HOMOTHETIE  <modif etat> CENTRE (<point_coor> | " ) <modif etat>   *<element>
SYMETRIE   (<point_coor> | &lt;DROITE&gt;| &lt;SEGMENT&gt;)   *<element>
```

ROTATION on applique la rotation de centre `<point>` d'angle %ANGLE sur `<elements>`. Si %NOMBRE $\leq$ 1, on déplace `<elements>` sinon on crée %NOMBRE-1 nouvelles occurences de `<elements>`. (Donc il y a %NOMBRE fois `<elements>` sur l'écran après la transformation).

TRANSLAT on applique la translation d'un vecteur allant du premier `<point>` au deuxième `<point>` sur `<elements>`. Si %NOMBRE $\leq$ 1, on déplace `<elements>`, sinon on crée %NOMBRE-1 nouvelles occurences de `<elements>`. (Donc il y a %NOMBRE fois `<elements>` sur l'écran après la transformation).

HOMOTHETIE on applique l'homothétie de centre `<point>` de rapport %RAPPORT sur `<elements>`. Si %NOMBRE $\leq$ 1, on déplace `<elements>` sinon on crée %NOMBRE-1 nouvelles occurences de `<elements>`. (Donc le nombre d'occurences de `<elements>` sur l'écran après la transformation est %NOMBRE).

SYMETRIE on applique la symétrie par rapport à `<point>` ou à `<droite>` sur tous les `<elements>` que l'on désignera. Attention si %NOMBRE est plus grand que 2, il y aura des superpositions d'éléments.

4.2.10 Arrondir un angle

Cette commande permet de faire des arrondis entre SEGMENTS et ARCS.

```
ARRONDI  *( <contrainte> <contrainte> )
```

On crée un arrondi de rayon %RAYON entre les deux <contrainte>, *l'arrondi tourne toujours dans le sens positif.* L'arrondi est un arc dont le cercle qui le supporte est tangent aux deux <contrainte>, l'arc va du premier point de tangence au deuxième point de tangence en tournant dans le sens positif. Si une <contrainte> est un ARC ou un SEGMENT, on modifie cette <contrainte> de façon à ce que cet élément se termine au point de tangence avec l'arrondi.

4.2.11 Définir un Contour

```
CONTOURAGE   (<DROITE> | <CERCLE>)  (<DROITE> | <CERCLE>)
             (<DROITE> | <CERCLE>) (COMPLE_ARC | '')
           *( (<DROITE> | <CERCLE>) (COMPLE_ARC | '')      )
```

Permet de définir un contour formé de SEGMENTS et d'ARCS s'appuyant sur un ensemble de droites et de cercles. Le programme fonctionne en calculant l'intersection de la contrainte i avec la contrainte $i+1$ ce qui donne le point I puis la contrainte $i+1$ avec la $i+2$ ce qui donne le point I+1, I et I+1 forment un nouveau morceau du contour et ainsi de suite.

Par exemple avec des droites (figure 4.2) :

Supposons que nous avons les 5 droites D_1 D_2 D_3 D_4 D_5 s'intersectant en p_1 p_2 p_3 p_4 p_5 et que nous voulons définir le contour formé des segments (p_1, p_2), (p_2, p_3), (p_3, p_4), (p_4, p_5), (p_5, p_1). On désigne par +i le ième point de désignation. On désigne D_2 en +2 puis D_1 en +1 ce qui défini p_1

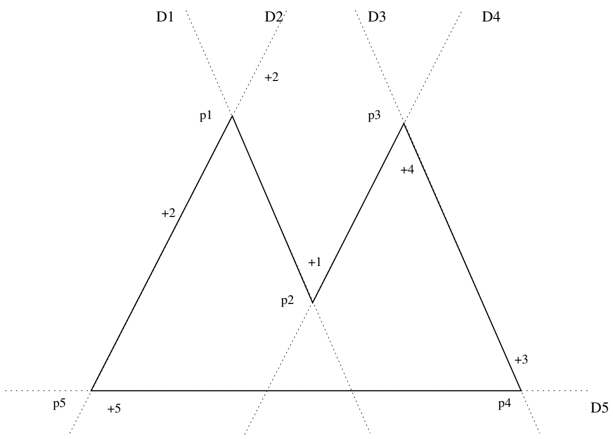

FIG. 4.2 - *Exemple de contourage*

Puis on désigne D_4 en +4 ce qui définit p_2 (ici le programme trace (p_1, p_2))
Puis on désigne D_3 en +3 ce qui définit p_3 (ici le programme trace (p_2, p_3))
Puis on désigne D_5 en +5 ce qui définit p_4 (ici le programme trace (p_3, p_4))
Puis on désigne D_2 en +2 ce qui définit p_5 (ici le programme trace (p_4, p_5))
Puis on désigne D_1 en +1 ce qui définit p_1 (ici le programme trace (p_5, p_1))

Dans le cas de cercles il faut faire attention car les points de désignation servent à lever les ambiguités d'intersections multiples (le point de désignation de la contrainte i sert à lever l'ambiguité de désignation entre les contraintes i et i+1).

4.2.12 Retourner des segments ou arcs

```
RETOURNE *( &lt;ARC&gt; | &lt;SEGMENT&gt; )
```

Retourne un SEGMENT ou un ARC

- ARC : $\alpha \rightarrow -\alpha$
- SEGMENT: $(P1, P2) \rightarrow (P1, 2 * P1 - P2)$ (Symétrie / P1)

4.2.13 Complémenter des arcs

```
COMPLE_ARC *ARC
```

Change un ARC en son complément si $(\alpha < 0)$ alors $\alpha = \alpha + 2\pi$ sinon $\alpha = \alpha - 2\pi$

4.2.14 Inverser des éléments

```
INVERSER *(<ARC> | &lt;SEGMENT&gt; | <SPLINE>)
```

Inverse un ARC, SEGMENT ou une SPLINE

- ARC : $(C, P, \alpha) \rightarrow (C, Rot(P, \alpha)/C, -\alpha)$
- SEGMENT: $(P_1, P_2) \rightarrow (P_2, P_1)$
- SPLINE : $(P_1, P_2, ..., P_n) \rightarrow (P_n, P_{n-1}, ..., P_2, P_1)$

4.2.15 Couper 2 à 2 des éléments

```
COUPER  * (  <point_coor> *(<ARC> | &lt;SEGMENT&gt; | <SPLINE>)
           | (<DROITE> | <CERCLE> | <ARC> | &lt;SEGMENT&gt; | <SPLINE>)
             (<DROITE> | <CERCLE> | <ARC> | &lt;SEGMENT&gt; | <SPLINE>)
          )
```

Si on désigne `<point>` en premier on coupe en deux un ARC ou un SEGMENT ou une SPLINE par la projection de `<point>` sur l'élément.
Pour la SPLINE on coupe au point de définition le plus proche du point `<point>`.
Sinon on coupe 2 à deux les éléments désignés si ceux-ci sont sécables (les éléments non sécables sont les DROITE et les CERCLE.

4.2.16 Changer une partie d'un élément

```
CHANGER *(  ANGLE <ARC> (<VALEUR> | IDEM)
          | RAYON <CERCLE> (<VALEUR> | IDEM)
          | <POINT> <point_coor>)
```

Permet donc de changer :

- l'angle d'un ARC désigné
- le rayon d'un CERCLE
- la position d'un `<point>` (on change le premier en le second, ce premier `<point>` peut être le centre d'un arc ou d'un cercle)
- l'extremité d'un SEGMENT ou d'un ARC.
- un point de définition d'une SPLINE, donc la forme d'une SPLINE.

4.2.17 Ajouter des éléments

```
AJOUTER *(  POINT   *( <ARC> | &lt;SEGMENT&gt; | <SPLINE> | <POINT> <POINT> )
          | SEGMENT *( &lt;SEGMENT&gt; &lt;SEGMENT&gt; )
          | SPLINE  *( <SPLINE> (   <SPLINE> | *<POINT>  ) )
          | ARC     *( <ARC> <ARC>  )
```

- Si POINT : on rajoute %NOMBRE points avec une raison %RAPPORT sur ARC ou SEGMENT ou SPLINE ou entre les 2 POINTS.
- Si SEGMENT : on rajoute %NOMBRE segments par interpolation linéaire de raison %RAPPORT entre les 2 SEGMENTS.
- Si SPLINE :
  - suivi de la désignation de 2 SPLINES alors : on rajoute %NOMBRE splines par interpolation linéaire de raison %RAPPORT entre les 2 SPLINES.
  - suivi de la désignation d'une spline et d'un point on ajoute à la SPLINE désignée de nouveaux POINTS de définition donnés par `*<point>`.
- Si ARC :on rajoute %NOMBRE arcs définis par 3 points (2 extremités, et le milieu) qui sont les interpolations linéaires de raison %RAPPORT entre les extrémités et les milieux des 2 arcs désignés

4.2.18 Fondre deux éléments en un

```
FONDRE *( &lt;segment&gt; &lt;segment&gt; | <arc> <arc> | <spline> <spline> )
```

Permet de fondre 2 SEGMENTS ou 2 ARCS ou 2 SPLINES ayant une extrémité commune en un seul élément de même type. L'élément résultant est dans le sens du premier désigné. Pour que cette instruction puisse s'effectuer, il faut:

- dans le cas de 2 segments, que les 2 segments soient colinéaires.
- dans le cas de 2 arcs, que les 2 arcs aient même centre et même rayon.

Chapitre 5

L'application PREP_MESH

5.1 Présentation

Cette application est dirigée par le menu de l'application (figure 5.1) en haut de l'écran et le menu de désignation (figure 3.2) en bas de l'écran. Elle permet de définir:

- la discrétisation sur les lignes des contours;
- les sous domaines à mailler;
- les références des lignes et de leurs extrémités en vue de la prise en compte des conditions aux limites (c.f. la méthode de éléments finis);
- le type de mailleur structuré *(carré déformé)* ou non structuré *(Voronoï)* de chaque sous domaine.

De plus à ce niveau nous pouvons interfacer le mailleur APNOPO en générant un fichier de données pour le préprocesseur APNOXX (voir [7]).

LIGNE	COMPOSANTE	DEFISSURE	RAISON	REF_NOEUD	REF_DOMAIN	→→→
DOMAINE	TOUT	FISSURE	Nb_INTERVAL	REF_LIGNE	Lg_INTERVAL	→→→

←←←	DEF_DOMAIN	QUADRANGLE	REGULIER	RETIRER	VOIR	GENERER
←←←	INTERIEUR	TRIANGLE	COINS	VERIFIER		Gen. Base

FIG. 5.1 - *Menu de l'application PREP_MESH*

5.2 Utilisation

A l'entrée dans cette application, le programme calcule automatiquement toutes les composantes connexes intérieures et extérieures du dessin (exemple figure 5.2). Les composantes connexes intérieures étant dans le sens positif, les extérieures dans le sens négatif.

On obtient automatiquement les composantes connexes suivantes:

```
P5,P3,P4      >0
P1,P3,P5      >0
```

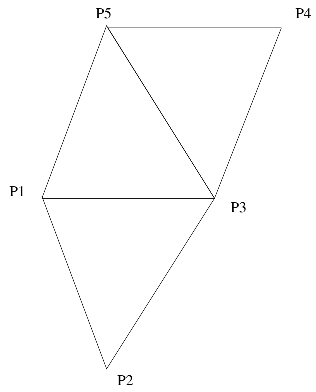

FIG. 5.2 - *Exemples de composantes*

```
P1,P2,P3       >0
P1,P5,P4,P3,P2 <0
```

- On appellera `<element>` un ARC ou un SEGMENT ou une SPLINE en tant que tel.
- On appellera `<composante>` la composante désignée par un de ses `<element>`.
- On appellera `<domaine>` le domaine désigné par un de ses éléments (`<composante>` ou `<element>` intérieur).
- On appellera `<ligne>` = LIGNE `<element1>` `<element2>` l'ensemble des éléments allant du premier élément `<element1>` au deuxième élément `<element2>` dans le sens de la composante à laquelle appartiennent les deux éléments.

5.2.1 Modification de l'état du système

L' état du système est un ensemble de variables qui sont utilisées par l'application, ces variables globales à l'application sont :

- %RAISON permet de définir la raison de la discrétisation d'une courbe (ligne, ou segment, ou arc, ou spline)
- %NBINTR permet de définir le nombre d'intervalles de la discrétisation d'une courbe
- %LGINTR permet de définir la longueur des intervalles de la discrétisation d'une courbe
- %NUREF permet de définir un numéro de référence à un point, courbe, sous domaine

et sont modifiables presque à tout moment.

Nous appellerons `<modif etat>` une modification de l'état du système, c'est à dire une modification de la valeur affectée à l'une des variables

```
<modif etat>  = *( <raison> | <nb interval> | <lg interval> | <numero de ref> )
```

- Modification de %RAISON: `<raison> = RAISON (<VALEUR> | IDEM)`
- Modification de %NBINTR: `<nb interval> = NBINTR (<VALEUR> | IDEM)`
- Modification de %LGINTR: `<lg interval> = LGINTR (<VALEUR> | IDEM)`
- Modification de %NUREF: `<numero de ref> = NUREF (<VALEUR> | IDEM)`

IDEM Permet de garder la valeur précédente de la variable d'état.

5.2.2 Définition des points intermédiaires

```
Nb_INTERVAL * <VALEUR>
              * (   LIGNE       *(( <VALEUR> | '')<element1> <element2>)
                  | COMPOSANTE  *(( <VALEUR> | '')<element3>)
                  | TOUT
                  |                ( <VALEUR> | '')<element4>  )
```

Permet de générer sur

- les éléments d'une `<ligne>` définie par `<element1> <element2>`
- les éléments d'une `<composante>` désignée par `<element3>`
- tous les éléments de la B.D.
- un élément seulement `<element4>`

des points intermédiaires de façon à obtenir %NB_INTERVAL intervalles, la `<VALEUR>` est entrée: %NB_INTERVAL prend cette valeur, et celle-ci est affichée dans la ligne en bas de l'écran.

```
Lg_INTERVAL  *<VALEUR>
              * (    LIGNE        *(( <VALEUR> | '')<element1> <element2>)
                  | COMPOSANTE    *(( <VALEUR> | '')<element3>)
                  | TOUT
                  |                  ( <VALEUR> | '')<element4>  )
```

Permet de générer sur

- les éléments d'une `<ligne>` définie par `<element1> <element2>`
- les éléments d'une `<composante>` désignée par `<element3>`
- tous les éléments de la B.D.
- un élément seulement `<element4>`

des points intermédiaires de façon à ce que la distance entre deux points consécutifs soit proche de %Lg_INTERVAL, si `<VALEUR>` est entré, %Lg_INTERVAL prend cette valeur, et celle-ci est affichée dans la ligne en bas de l'écran.

5.2.3 Définition de la raison

```
RAISON  * <VALEUR>
        * (   LIGNE      *((  <VALEUR> | '')<element1> <element2>)
            | COMPOSANTE *((  <VALEUR> | '')<element3>)
            | TOUT
            |               ( <VALEUR> | '')<element4>  )
```

Permet d'affecter %RAISON comme raison

- aux éléments d'une `<ligne>` définie par `<element1> <element2>`
- aux éléments d'une `<composante>` désignée par `<element3>`
- à tous les éléments de la B.D.
- à un élément seulement `<element4>`

, si `<VALEUR>` est entré, %RAISON prend cette valeur, et celle-ci est affichée dans la ligne en bas de l'écran.

5.2.4 Définition du numéro de référence d'extrémités

```
REF_NOEUDS * <VALEUR>
           * (   LIGNE      *(( <VALEUR>  | ")<element1> <element2>)
               | COMPOSANTE *(( <VALEUR>  | ")<element3>)
               | DOMAINE    *(( <VALEUR>  | ")<element4>)
               | TOUT
               |               ( <VALEUR>  | ")<element5>  )
```

Permet d'affecter %NUREF comme numéro de référence

- aux extrémités des éléments d'une <ligne> définie par <element1> <element2>
- aux extrémités des éléments d'une <composante> désignée par <element3>
- aux extrémités des éléments d'un <domaine> désigné par <element4>
- à toutes les extrémités des éléments de la B.D.
- aux extrémités d'un élément seulement <element5>

si <VALEUR> est entré, %NUREF prend cette valeur, et celle-ci est affichée dans la ligne en bas de l'écran.

Si les éléments désignés sont des fissures <VALEUR> sera affecté à l'extrémité et du côté où est cliqué l'élément, la LIGNE ou la COMPOSANTE et, par continuité sur les extrémités des éléments adjacents aux extrémités. En effet un élément fissure possède à sa gauche ou/et sa droite un numéro references de ligne et 2 numéros de reference aux extrémités. Ces numéros peuvent être identiques ou differents, s'ils sont differents l'élément est nécessairement une fissure.

5.2.5 Définition du numéro de référence de lignes

```
REF_LIGNE * <VALEUR>
            * (   LIGNE      *(( <VALEUR>  | ")<element1> <element2>)
                | COMPOSANTE *(( <VALEUR>  | ")<element3>)
                | DOMAINE    *(( <VALEUR>  | ")<element4>)
                | TOUT
                |               ( <VALEUR>  | ")<element5>)
```

Permet d'affecter %NUREF comme numéro de référence

- aux éléments d'une <ligne> définie par <element1> <element2>
- aux éléments d'une <composante> désignée par <element3>
- aux éléments d'un <domaine> désigné par <element4>
- à tous les éléments de la B.D.
- à un élément seulement <element5>

Si <VALEUR> est entré,%NUREF prend cette valeur, et celle-ci est affichée dans la ligne en bas de l'écran. Si les éléments désignés sont des fissures %NUREF sera affecté du côté ou est cliqué l'élément, la LIGNE ou la COMPOSANTE.

5.2.6 Fissurage des lignes

```
FISSURE   * (   LIGNE      *(<element1><element2>)
              | COMPOSANTE * <element3>
              | TOUT
              |               <element4>  )
```

Permet de fissurer :

- les éléments d'une LIGNE définie par <element1> <element2>

– les éléments d'une COMPOSANTE désignée par `<element3>`

– tous les éléments de la B.D.

– un élément seulement `<element4>`.

5.2.7 Défissurage des lignes

```
DEFISSURE * (   LIGNE      *(<element1><element2>)
              | COMPOSANTE * <element3>
              | TOUT
              |              <element4>  )
```

Permet de défissurer :

– les éléments d'une LIGNE définie par `<element1>` `<element2>`

– les éléments d'une COMPOSANTE désignée par `<element3>`

– tous les éléments de la B.D.

– un élément seulement `<element4>`.

C'est l'opération inverse du fissurage, on réduit le nombre de références à 1, en ne conservant que le numéro de gauche de la fissure.

5.2.8 Définition d'un numéro de référence de sous domaine

```
REF_DOMAIN *( <VALEUR>  | <domaine> | TOUT )
```

Permet d'affecter `<VALEUR>` comme numéro de sous domaine au domaine `<DOMAINE>` désigné ou à tous les domaines définis. Si `<DOMAINE>` n'existe pas, on le crée.

5.2.9 Définition d'un domaine

```
DEF_DOMAI <element1>
          * (<element2> | INTERIEUR *(<point> | <element3> | <ligne>) )
```

– Si `<element1>` n'appartient pas déjà à un domaine, alors on crée un domaine de numéro %NUREF ayant `<element1>` comme composante extérieure, et ses composantes intérieures sont désignées par les éléments `<element2>` suivants éventuels, ou bien ses éléments intérieurs sont définis par INTERIEUR suivi de `<point>` ou, les points intermédiaires de `<element3>` ou, les points intermédiaires des éléments de la `<ligne>`.

– Si `<element1>` appartient déjà à un domaine alors la composante désignée par `<element2>` est ajoutée aux composantes de ce domaine en position composante intérieure.

5.2.10 Ajout d'éléments intérieurs à un domaine

```
INTERIEUR *( <element1> *( <point> | <element2> | <ligne>) )
```

Permet d'ajouter des éléments intérieurs à un domaine désigné par l'élément `<element1>`. Les éléments intérieurs pouvant lui être rajoutés sont : `<point>` ou les points intermédiaires de l'`<element2>` ou les points intermédiaires des éléments de la `<ligne>`.

5.2.11 Vérification de la bonne définition d'un domaine

```
VERIFIER  *( TOUT | <element>)
```

– Permet de vérifier si un domaine désigné par `<element>` est correctement défini.

– Permet de vérifier si tous les domaines sont correctement définis (composantes intérieures de surfaces négatives, etc...)

5.2.12 Retrait d'éléments à un domaine

```
RETIRER  *(   INTERIEUR *( <point> | <element1>)
            | DOMAINE *<element2>
            | TOUT
            | <element3> )
```

Permet de retirer

- un élément `<element1>` ou, un `<point>` intérieur à un domaine désigné par cet `<element1>` ou ce `<point>`
- un domaine désigné par `<element2>`
- tous les domaines
- la composante désignée par `<element3>` d'un domaine. On ne peut pas le faire si cette composante est extérieure car la première composante trou deviendrait extérieure... (c.f. structure interne des données).

5.2.13 Mode de maillage d'un domaine

```
TRIANGLE   *<domaine>
```

Demande que le `<domaine>` soit découpé en triangles en utilisant un algorithme de Voronoï[7]

```
QUADRANGLE  *(<domaine>
              (  <ligne>
               | <element>
               | (  EXTREMITE<element1> EXTREMITE<element2>
                    EXTREMITE<element3> EXTREMITE<element4> )
               | " ) )
```

Demande que le `<domaine>` soit homéomorphe à un carré (quadrangle déformé) et on va faire un maillage structuré (de type différences finies NxM) dans ce sous domaine quadrangulaire. Celui-ci sera découpé en quadrangles sauf si on change le mode de découpage avec (5.2.14) Il y deux façons de décrire le domaine quadrangulaire (4 côtés ou 4 sommets):

- le premier côté du domaine est défini par la ligne `<ligne>` ou, par l'élément `<element>`,
- on énumère les 4 sommets du domaine en donnant les 4 extrémités d'éléments qui les définissent (on doit énumérer les 4 sommets en tournant dans le sens positif).

5.2.14 Mode de découpage d'un domaine quadrangulaire

```
REGULIER  *(<domaine> <VALEUR> )
```

Demande que dans le `<domaine>` la découpe des quadrangles en triangles (c.f. brochure APNOPO de MODULEF) se fasse de façon :

- régulière à droite (valeur=2)
- régulière à gauche (valeur=1)
- non régulière (valeur=0)

selon la valeur.

```
COINS  *(<domaine> <VALEUR> )
```

Demande que dans le `<domaine>` la découpe des quadrangles en triangles se fasse en traitant ou en ne traitant pas les coins selon la valeur (Y = 0 / NO $\neq$ 0).

5.2.15 Visualisation de domaines ou de composantes

`VOIR    DOMAINE *(<domaine> | <composante> | TOUT )`

Permet de visualiser en traits épaissis le `<domaine>` ou la `<composante>` désigné par un de ses éléments. Cette action est utile pour "debugger" un maillage (problème d'intersection, erreur d'arrondi).

5.2.16 Génération du fichier d'entrée de APNOXX

`GENERER`

Permet d'effectuer la génération du fichier pour APNOXX (Modulef).

- Ce fichier aura comme nom XXX donne par l'utilisateur suffixé par .DATA.
- Ce fichier demande à APNOXX de générer un NOPO de nom XXX.NOPO.

Le programme ne prend en compte que les domaines précédemment définis.

Chapitre 6

L'application EDIT_MESH

6.1 Présentation

Cette application est dirigée par le menu de l'application (figure 6.1) en haut de l'écran et le menu de désignation (figure 3.3) en bas de l'écran. Elle permet d'éditer un maillage 2D formé de triangles et quadrangles, obtenu soit en commutant de l'application **CONSTRUCTION** ou **PREP_MESH** à l'application **EDIT_MESH**, soit en restaurant un maillage (c.f. les menus généraux). Les possibilités sont brièvement : ajouter, supprimer, bouger des sommets du maillage; retourner l'arête coupant un quadrilatère, *quadranguler, trianguler*, régulariser (mettre les sommets au barycentre de leurs voisins), rendre le maillage de Delaunay, modifier les références du maillage (frontières et sous domaines), transformer géométriquement un sous domaine, renuméroter, fissurer.

REGULARISER	DELAUNAY	AJOUTER	BOUGER	SUPPRIMER	RETOURNER	→ → →
RENUMEROTER	TOUT	ORIGINAUX	EVALUER	ROTATION	TRANSLATION	→ → →

← ← ←	MODIF_REF	FISSURE	TRIANGULER	ANGLE MIN	MARK ELEMENT
← ← ←	SYMETRIE	HOMOTHETIE	QUADRANGULER	ANGLE MAX	FRONTIERE

FIG. 6.1 - *Le menu de l'application EDIT_MESH*

6.2 Généralités

Les nouveaux types de désignation de l'application sont :

- si le mode de désignation est ELEMENT, on désigne un élément de type `<ELEMENT>` qui est un "élément fini" (triangle ou quadrangle) (il suffit de cliquer dans l'élément fini);
- si le mode de désignation est ARETE, on désigne un élément de type `<ARETE>` qui est une arête d'un élément du maillage (il suffit de cliquer dans un élément contenant cette arête et près de celle-ci);
- si le mode de désignation est SOMMET on désigne un élément de type `<SOMMET>` qui est un sommet d'un élément du maillage (il suffit de cliquer dans un élément contenant ce sommet et près de celui-ci);

– si le mode de désignation est S_DOM, on désigne un élément de type `<S_DOM>` qui est la composante connexe bornée par les frontières internes ou externes, et contenant l'élément fini cliqué.

De plus on utilisera souvent la notation `<point>` pour désigner un point `<POINT>`, `<COORD>`ou `<SOMMET>`.

```
<point> = <POINT> | <COORD> | <SOMMET>;
```

6.2.1 Définition des variables d'état de l'application

– `MARK_ELEMENT` permet de visualiser les éléments dont les angles sont hors des bornes (angle minimal, angle maximal pour les triangles ou pour les quadrangles).

– `(TRIANGULER|QUADRANGULER) ANGLE_MIN *<VALEUR>` permet de changer la valeur minimale des angles en degrés des triangles ou des quadrangles.

– `(TRIANGULER|QUADRANGULER) ANGLE_MAX *<VALEUR>` permet de changer la valeur maximale des angles en degrés des quadrangles ou triangles.

– `FRONTIERE` permet de bouger ou non les points frontières non extrémités (flip flop), initialement on ne peut pas déplacer les points frontières avec les actions BOUGER, REGULARISER.

6.3 Edition du maillage

Les actions d'édition du maillage sont :

6.3.1 Bouger un sommet

```
BOUGER   *(<SOMMET>  <point>  |  FRONTIERE)
```

Cette action permet de bouger le sommet désigné (qui peut être frontière suivant la valeur de la variable d'état `FRONTIERE`) pour le déplacer au second point désigné) qui peut être soit un *point_tablette* (défini avec la souris), un *point_xy* (défini avec (x,y), 2 valeurs calculette), un *point* de BD, ou un *sommet* (dans ce cas dernier cas, on prend comme nouveau sommet la position du curseur et non le sommet désigné).
Remarque: si l'on bouge un point frontière on déplace le point sur celle-ci.

6.3.2 Ajouter un sommet au maillage

```
AJOUTER *(<SOMMET>  |  <ELEMENT>)
```

Cette action permet d'ajouter un sommet dans l'élément du maillage contenant le point désigné. Si l'élément désigné est un quadrangle on commence par le couper en 2 triangles. On ajoute un sommet au triangle désigné en le coupant en 3 triangles (les coordonnées du point ajouté sont celles du curseur), puis on rend le maillage *triangulaire* voisin de Delaunay.

6.3.3 Supprimer un sommet du maillage

```
SUPPRIMER  *(<SOMMET>  |  <S_DOM>)
```

Cette action permet de supprimer:

1. soit un sommet non frontière du maillage et de remailler le trou généré,
2. soit un sous domaine.

6.3.4 Retourner une arête du maillage

```
RETOURNER *( <ARETE>)
```

Cette action permet de retourner l'arête interne du quadrilatère défini avec les 2 triangles adjacents à cette arête à condition que les triangles générés soient corrects.

6.3.5 Maillage de Delaunay

```
DELAUNAY  *(<ELEMENT> |  <S_DOM>  |  TOUT )
```

Cette action permet de forcer le maillage à satisfaire le critère de Delaunay pour les triangles (les seuls sommets du maillage contenus dans le cercle circonscrit à un triangle sont les sommets du triangle).

1. soit dans le voisinage du triangle désigné,
2. soit dans tout le sous domaine désigné,
3. soit dans tout le maillage.

6.3.6 Régularisation du maillage

```
REGULARISER  *(    <SOMMET>  |  <ELEMENT>  |  <S_DOM>
                 |   TOUT |  ORIGINAUX |  FRONTIERE)
```

Cette action permet de régulariser le maillage ou une de ses parties. C'est-à-dire que l'on met tour à tour les sommets de la partie au barycentre de ses voisins. On déplace les points frontières si cela est autorisé (c.f. variable d'état).

Remarque: si l'on bouge un point frontière on déplace le point sur celle-ci.

6.3.7 Trianguler

```
TRIANGULER *(  <ELEMENT>  |<S_DOM>  |    TOUT |  ORIGINAUX
             | ANGLE_MIN  | ANGLE_MAX  )
```

Cette action permet de trianguler le maillage ou une partie et de modifier les variables d'état angle minimal ou maximal des triangles.

6.3.8 Quadranguler

```
QUADRANGULER    *(  <ELEMENT>   | <ARETE>  | <S_DOM>
                  |    TOUT     |  ORIGINAUX
                  | ANGLE_MIN   | ANGLE_MAX )
```

Cette action permet de *"quadranguler"* le maillage ou une partie de celui-ci.

1. si on désigne un *élément* et si l'élément désigné est un triangle et que ses 3 éléments adjacents sont aussi des triangles alors on génère 3 quadrangles si ceux-ci sont *valides* (convexes et avec les angles compris entre les angles min et max (c.f. variable d'état));
2. si l'on désigne une *arête* et si les 2 éléments adjacents sont des triangles et si l'arête n'est pas frontière de sous domaines alors on génère le quadrangle formé des 2 triangles si celui-ci est *valide*;
3. si l'on désigne un *sous domaine* alors on essaie de *"quadranguler"* tout le sous domaine (la méthode dépend de l'élément désigné pour définir le sous domaine), *attention on "re-quadrangule" tout le sous domaine et même les quadrangles déjà existants*;
4. si vous cliquez dans l'ITEM TOUT ou, dans l'item ORIGINAUX alors on essaie de *"quadranguler"* tout le maillage (il peut rester des triangles si le nombre d'arêtes frontières est impair ou si les angles sont mauvais).
5. si vous cliquez dans l'ITEM ANGLE_MIN (Resp. ANGLE_MAX), on modifie la variable d'état angle minimal (resp. maximal) des quadrangles.

6.3.9 Fissurer

```
FISSURER *( &lt;ARC&gt; | &lt;SEGMENT&gt; | &lt;SPLINE&gt; )
```

Cette action permet de générer ou de supprimer des fissures dans le maillage. Si l'élément de la BD (arc, segment ou spline) est fissuré alors on le défissure sinon on le fissure. Une fissure dans un maillage est une ligne ou un ensemble de lignes tel que les sommets qui sont de part et d'autre de la ligne soient différents. *Sur les dessins on écarte les sommets fissurés pour visualiser la fissure mais cela n'est qu'un artifice graphique.*

6.3.10 Modifier les références du maillage

```
MODIF_REF *( <VALEUR>
             *(  &lt;ARC&gt;    | &lt;SEGMENT&gt; |  &lt;SPLINE&gt;
               | <SOMMET> | <ARETE>   |  <S_DOM>)  )
```

Cette action permet de modifer les références des différentes parties d'un maillage.

Attention la référence d'un sommet est la référence de l'élément de BD le supportant (un point si c'est une extrémité), de même pour les arêtes. Si l'élément est fissuré alors il y a 2 références, une à gauche et une à droite.

6.3.11 Renuméroter les sommets du maillage

```
RENUMEROTER
```

Cette action permet de renuméroter les sommets du maillage dans le but de diminuer la largeur de bande ou le profil de la matrice élément fini correspondante (P_1 ou Q_1 lagrange)

6.4 Les transformations

Les transformations s'appliquent : soit à des sous domaines, soit à tout le domaine, soit seulement aux sous domaines originaux (c'est-à-dire à tous les sous domaines dessinés en trait continu).
Remarque: un sous domaine transformé (non original) n'est défini que par un pointeur sur le sous domaine original, et par la transformation affine (matrice 2×3), ce qui implique que toute modification (bouger, ajouter, supprimer,...) du sous domaine d'origine sera répercutée sur ses transformés. De plus les sous domaines originaux sont affichés en traits continus et les transformés sont affichés en traits pointillés. Pour générer effectivement tous les sous domaines transformés il faut cliquer dans l'ITEM EVALUER, ces sous domaines devenant alors des sous domaines originaux (on a évalué l'expression définissant le maillage).

Attention tant que les sous domaines transformés ne sont pas évalués, les éléments de BD frontières de ces sous domaines n'existent pas encore.

Un point `<point>` désignera soit un sommet, soit un point tablette, soit un point xy, soit un point immédiat. La syntaxe est :

```
<une_transformation> ( TOUT |  ORIGINAUX | <S_DOM>*)
```

C'est à dire que la transformation après être définie sera appliquée à tous les sous domaines, ou aux sous domaines originaux (en trait continu), ou bien aux sous domaines explicitement désignés.

La transformation est soit une translation, soit une symétrie, soit une rotation, ou soit une homothétie. C'est à dire syntaxiquement:

```
<une_transformation> =   <translation>  |  <symetrie>
                       | <rotation>     |  <homethetie>;
```

elles sont définies comme suit :

6.4.1 Translation

```
<translation> = TRANSLATION   <point>  <point>
```

Définit la translation allant du premier point au second.

6.4.2 Symétrie

```
<symetrie>  = SYMETRIE  (    <point_coor>  <point_coor>
                           | &lt;SEGMENT&gt;  | <ARETE>)
```

Définit la symétrie par rapport à la droite qui supporte le segment ou l'arête désigné, ou qui passe par les 2 points désignés.

6.4.3 Rotation

```
<rotation>  = ROTATION (<point>  <VALEUR>  |  <VALEUR>  <point>)
```

Définit la rotation autour du point, d'angle égal à la valeur définie avec la calculette.

6.4.4 Homothétie

```
<homothetie>  = HOMOTHETIE  (<point> <VALEUR> | <VALEUR> <point>)
```

Définit l'homothétie de centre le point et de rapport égal à la valeur.

6.4.5 Générer effectivement les transformations

```
EVALUER
```

Cette action permet de générer effectivement tous les sous-domaines transformés afin de pouvoir les éditer indépendamment de leur original. Cette action crée et transforme aussi tous les éléments (segments, arcs, splines) référencés par les sous domaines transformés.

6.4.6 Exemple de transformation

Pour symétriser des sous domaines, il suffit de cliquer dans l'item SYMETRIE puis soit de désigner 2 points, soit un segment ou soit une arête, puis de désigner soit les sous domaines à transformer, soit de cliquer dans l'item TOUT ou soit dans l'item ORIGINAUX puis avant de sauver le maillage sous une forme autre que "mesh" cliquer dans l'item EVALUER.

Chapitre 7

Exemple

7.1 Le carré unité troué

Dans cette section nous montrons comment mailler un carré unité troué. Il faut commencer par exécuter le programme emc2 et entrer le numéro du "device" graphique de F3D et de plus répondre à quelques questions supplémentaires *(dépend de l'implémentation de F3D).*

7.1.1 Construction du carré unité troué

1. Cliquer dans l'item `POINT` du menu construction
2. Cliquer dans l'item `PT_XY` du menu désignation
3. Taper au clavier `0=0=1=0=0=1=1=1=.5=.5=`, *5 points s'affichent au milieu de l'écran car l'échelle est égale à 1*
4. Cliquer dans l'item `VOIR_TOUT` pour bien voir les 5 points. Le $5^{\text{iéme}}$ point sera le centre du trou.
5. Cliquer dans l'item `SEGMENT` du menu construction (en haut, celui du bas est l'item `SEGMENT` du menu de désignation)
6. Cliquer dans l'item `POINT` du menu désignation (en bas)
7. Cliquer dans la fenêtre graphique sur les points extremités des 4 segments à créer après chaque couple de cliques, un segment est construit; donc il faut cliquer 8 fois par exemple près des 8 points suivant (0,0), (1,0), (1,0), (1,1), (1,1), (0,1), (0,1), (0,0) dans cet ordre.
8. Cliquer dans l'item `CERCLE` du menu construction (en haut) puis cliquer dans l'item CENTRE
9. Cliquer dans la fenêtre graphique près un point (.5,.5) (on est toujours en désignation de point)
10. Cliquer dans l'item `RAYON`
11. Taper au clavier `1/4=` (un cercle de centre (.5,.5) et de rayon 1/4 s'affiche), *attention le cercle comme la droite n'est qu'un support de construction donc il faut le transformer en arc*
12. Cliquer dans l'item `ARC` du menu pour construire des arcs
13. Cliquer dans l'item `IDEM`
14. Cliquer dans l'item `CERCLE` du menu désignation (pour désigner le cercle)
15. Cliquer dans la fenêtre graphique (comme il n'y a qu'un cercle on ne peut que désigner celui-ci). Le cercle ne servant plus, on le détruit.
16. Cliquer dans l'item `DETRUIRE` du menu général (à gauche)
17. Cliquer dans la fenêtre graphique (le cercle et l'arc disparaissent car il sont superposés). *Attention on est toujours en mode de destruction.*

18. Cliquer dans l'item **RAFRAICHIR** pour réafficher *(l'arc est toujours la)*

On a fini la construction de la géométrie (figure 7.1).
On peut passer à l'application PREP_MESH.

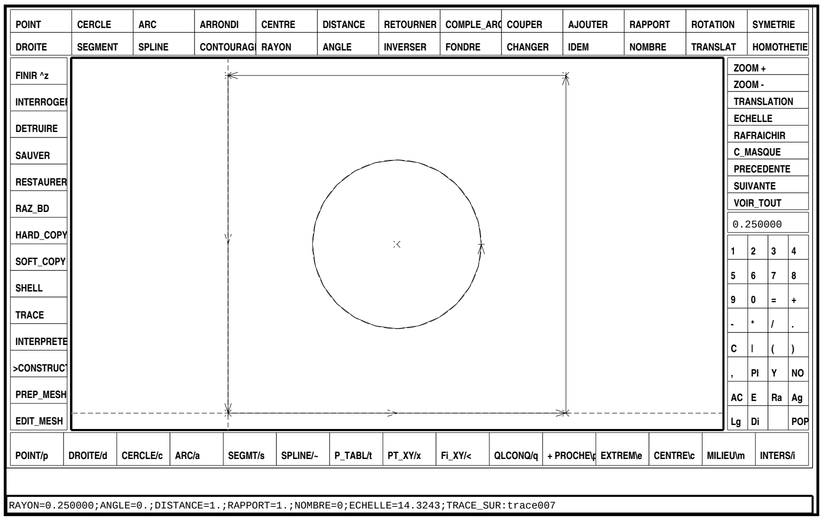

FIG. 7.1 - *La géométrie du domaine carré troué*

7.1.2 Définition de la discrétisation sur la frontière

1. Cliquer dans l'item **PREP_MESH** du menu général
2. Cliquer dans l'item **NB_INTERVAL** du menu de l'application (en haut)
3. Taper au clavier **4=**
4. TOUT du menu de l'application (en haut) (tous les éléments sont découpés en 4 intervalles)
5. Taper au clavier **12=**
6. Puis cliquer dans la fenêtre graphique près de l'arc qui se découpe en 12 intervalles). On n'a pas besoin de changer de mode de désignation.
7. Cliquer dans l'item **SAUVER** du menu général
8. Taper le nom de fichier **carre_troue(CR)**, on a créé le fichier *carre_troue.emc2_bd*

le fichier `carre_troue.emc2_bd`:

```
'-- TYPE N BD(1) BD(2) BD(3) BD(4) BD(5) NBNODE RAISON NUREFG NUREFD NUREF1G NUREF1D NUREF2G NUREF2D FIS'
'DROITE'      1 -1.000000 0.0000000 1.000000 0.0000000 0.0000000 0.0000000 0 0.0000000 0 0 0 0 0 0 F
'DROITE'      2 -1.000000 1.000000 0.0000000 0.0000000 0.0000000 0.0000000 0 0.0000000 0 0 0 0 0 0 F
'POINT'       3 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0 1.000000 0 0 0 0 0 0 F
'POINT'       4 0.0000000 1.000000 0.0000000 0.0000000 0.0000000 0.0000000 0 1.000000 0 0 0 0 0 0 F
'POINT'       5 0.0000000 0.0000000 1.000000 0.0000000 0.0000000 0.0000000 0 1.000000 0 0 0 0 0 0 F
'POINT'       6 0.0000000 1.000000 1.000000 0.0000000 0.0000000 0.0000000 0 1.000000 0 0 0 0 0 0 F
'POINT'       7 0.0000000 0.5000000 0.5000000 0.0000000 0.0000000 0.0000000 0 1.000000 0 0 0 0 0 0 F
'SEGMENT'     8 -3.000000 0.0000000 0.0000000 1.000000 0.0000000 0.0000000 5 1.000000 0 0 0 0 0 0 F
'SEGMENT'     9 -3.000000 1.000000 0.0000000 1.000000 1.000000 0.0000000 5 1.000000 0 0 0 0 0 0 F
'SEGMENT'     10 -3.000000 1.000000 1.000000 0.0000000 1.000000 0.0000000 5 1.000000 0 0 0 0 0 0 F
'SEGMENT'     11 -3.000000 0.0000000 1.000000 0.0000000 0.0000000 0.0000000 5 1.000000 0 0 0 0 0 0 F
'CERCLE'      12 0.2500000 0.5000000 0.5000000 0.0000000 0.0000000 0.0000000 2 1.000000 0 0 0 0 0 0 F
'ARC'         13 -2.000000 0.5000000 0.5000000 0.7500000 0.5000000 6.283185 13 1.000000 0 0 0 0 0 0 F
'MASQUE' 0 -0.4632312 1.465731 -5.0000010E-02 1.052500 0.0000000 0.0000000 0 0. 0 0 0  0 0 0  F
'RAYON' 0 0.2500000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'ANGLE' 0 0.0000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'DISTANCE' 0 1.000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'RAPPORT' 0 1.000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
```

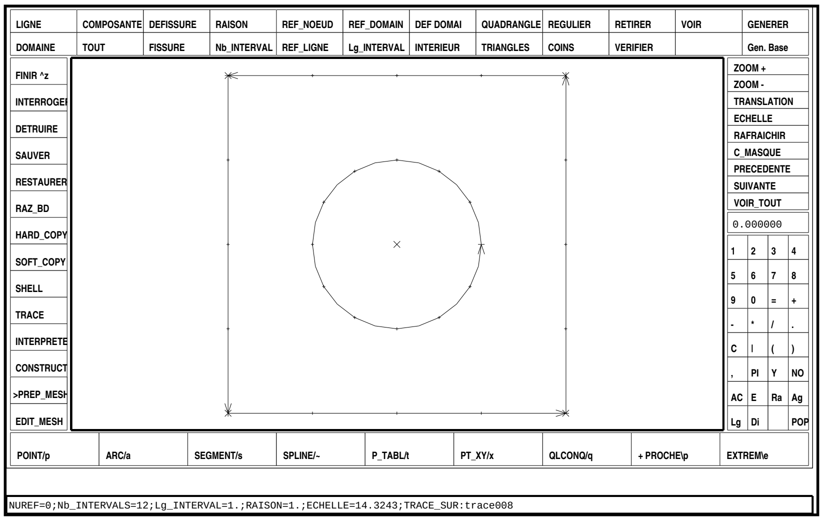

FIG. 7.2 - *La discrétisation des lignes du domaine carré troué*

```
'NOMBRE' 0 0.0000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'ECHELLE' 0 14.32432 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'NUREF' 0 0.0000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'NB_INTERVALS' 0 12.00000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'RAISON' 0 1.000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
'NUDSD' 0 1.000000 0. 0. 0. 0. 0. 0 0. 0 0 0  0 0 0  F
```

On a fini la préparation un maillage (figure 7.2)

7.1.3 Génération et écriture du maillage

1. Cliquer dans l'item **EDIT_MESH** du menu général pour activer l'application EDIT_MESH puis taper 4 (CR) au clavier pour prendre les options par défaut du mailleur. Le maillage s'affiche et le trou est maillé. On va supprimer le maillage du trou:

2. Cliquer dans l'item **S_DOM** du menu désignation.

3. Cliquer dans l'item **SUPPRIMER** du menu edit_mesh (en haut).

4. Cliquer dans le cercle maillé, le maillage du trou disparait.

5. Cliquer dans l'item **SAUVER** puis taper le type de sauvegarde par exemple **am_fmt(CR)** puis le nom du fichier **carre_troue(CR)**, un fichier carre_troue.am_fmt est généré.

6. Cliquer dans l'item **FINIR** pour sortir d'Emc^2

le fichier `carre_troue.am_fmt`:

```
 44 60  -- nbs,nbt
      1       2      29          25      24      29
     24      31      29           3      43      30
     15      16      31          22      34      23
     26      25      30          23      34      31
     32      27      43          28      27      32
     22      33      34          21      40      38
      4      32      43          34      15      31
     33      14      34          21      33      22
      5       6      39          20      42      40
     28      32      39          41      42      19
      7       8      36          18      36      37
     28      39      35          34      14      15
     12      13      38           6      35      39
     28      35      17          43      27      26
     38      13      14          23      31      24
      7      36      17          31      16      29
     11      12      40           3      30       2
     37       8      44          30      25      29
```

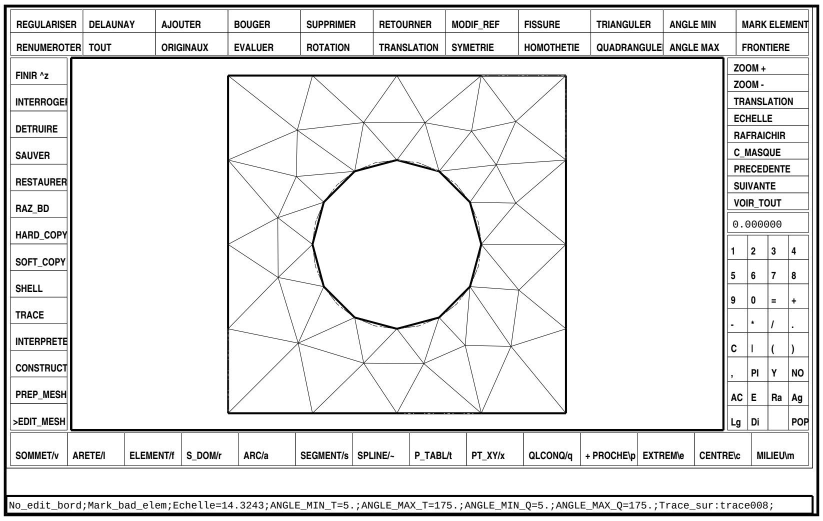

FIG. 7.3 - *Le maillage final du domaine carré troué*

```
    41      10      42          30      29       2
    44      10      41          29      16       1
    35       6       7          17      35       7
    36       8      37          17      36      18
    41      37      44          18      37      19
    38      14      33          21      38      33
    39      32       4           5      39       4
    40      12      38          20      40      21
    42      10      11          37      41      19
    42      11      40          19      42      20
    43      26      30           4      43       3
    44       8       9          10      44       9
 0.000000E+00   0.000000E+00       2.500000E-01   0.000000E+00
 5.000000E-01   0.000000E+00       7.500000E-01   0.000000E+00
 1.000000E+00   0.000000E+00       1.000000E+00   2.500000E-01
 1.000000E+00   5.000000E-01       1.000000E+00   7.500000E-01
 1.000000E+00   1.000000E+00       7.500000E-01   1.000000E+00
 5.000000E-01   1.000000E+00       2.500000E-01   1.000000E+00
 0.000000E+00   1.000000E+00       0.000000E+00   7.500000E-01
 0.000000E+00   5.000000E-01       0.000000E+00   2.500000E-01
 7.500000E-01   5.000000E-01       7.165064E-01   6.250000E-01
 6.250000E-01   7.165064E-01       5.000000E-01   7.500000E-01
 3.750000E-01   7.165064E-01       2.834937E-01   6.250000E-01
 2.500000E-01   5.000001E-01       2.834936E-01   3.750001E-01
 3.749999E-01   2.834937E-01       4.999999E-01   2.500000E-01
 6.249999E-01   2.834936E-01       7.165062E-01   3.749999E-01
 2.077567E-01   2.075703E-01       4.017089E-01   1.482612E-01
 1.483122E-01   4.021604E-01       7.024530E-01   2.009149E-01
 2.021900E-01   7.009841E-01       1.471843E-01   5.790281E-01
 8.609530E-01   3.646002E-01       8.519265E-01   6.222693E-01
 7.943417E-01   7.355210E-01       2.050139E-01   8.383228E-01
 8.379703E-01   1.980158E-01       4.015485E-01   8.609668E-01
 7.210978E-01   8.353896E-01       5.831956E-01   8.603261E-01
 5.796704E-01   1.470356E-01       8.526899E-01   8.640727E-01
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0
```

7.2 Une aile NACA0012

7.2.1 Introduction

La paramétrisation du naca0012 est pour $X \in [0, 1.008930411365]$

$$Y(X) = 5 \times 0.12(0.2969\sqrt{X} - 0.126X - 0.3516X^2 + 0.2843X^3 - 0.1015X^4)$$

Pour normaliser le naca0012, on utilise le changement de variables:

$$X = 1.008930411365x$$

$$Y = 1.008930411365y$$

Avec le petit programme fortran suivant:

```
      PROGRAM NACA12
      DOUBLE PRECISION X,Y,C,XX,YY
      PARAMETER (C=1.008930411365D0)
      INTEGER NB,I
C --------  ENTRER LE NOMBRE D'INTERVALLES -----------
      READ *,NB
      DO I=0,NB
C -----------------------------------------------
C       POUR QUE LA PROGRESSION DES POINTS GENERE  |
C         SOIT PLUS DENSE AU VOISINAGE DE 0        |
C                4                                 |
C       ON PREND X  PLUTOT QUE X                   |
C ------------------------------------------------
       XX= (DBLE(I)/DBLE(NB))**4
       X = C*XX
       Y = 5*.12*( 0.2969*SQRT(X) -0.126*X
     +            -0.3516*X**2    +0.2843*X**3 -0.1015*X**4)
       YY = Y / C
       PRINT*, XX,YY
      ENDDO
      END
```

on génére le fichier "naca12.21pts" en l'utilisant avec 20 intervalles.
le fichier "naca12.21pts" :

```
0.0000000000000000 0.0000000000000000
6.2500000000000000E-06 4.4290213310617220E-04
1.0000000000000000E-04 1.7659364374154260E-03
5.0625000000000000E-04 3.9520447457293480E-03
1.6000000000000000E-03 6.9724900930098020E-03
3.9062500000000000E-03 1.0785816129632120E-02
8.1000000000000000E-03 1.5335254409279880E-02
1.5006250000000000E-02 2.0543538719326360E-02
2.5600000000000000E-02 2.6304013971036800E-02
4.1006250000000000E-02 3.2467169756843240E-02
6.2500000000000000E-02 3.8822480697810590E-02
9.1506250000000000E-02 4.5076893778196040E-02
0.1296000000000000 5.0833559338836810E-02
0.1785062500000000 5.5577241169031020E-02
0.2401000000000000 5.8675384962268570E-02
0.3164062500000000 5.9403968668221290E-02
0.4096000000000000 5.7000809116986590E-02
0.5220062500000000 5.0728415111155510E-02
0.6561000000000000 3.9881091560527040E-02
0.8145062500000000 2.3576580331835940E-02
1.000000000000000 2.0060198232592660E-14
```

7.2.2 La construction de la géométrie

1. Cliquer dans l'item `SPLINE` du menu construction
2. Taper `<` au clavier ou cliquer dans l'item `Fi_XY` du menu désignation puis entrer le nom du fichier contenant les points du NACA0012 en tapant `naca12.21pts(CR)`, *la spline du naca s'affiche à l'écran*
3. Cliquer dans l'item `VOIR_TOUT` pour mieux voir la spline
4. On veut mailler l'extérieur du naca0012 pour cela il faut définir la frontière correspondant à l'infini que l'on placera à -5 en amont, +11 en aval et 8 orthogonalement. On construit les 3 points (-5,0), (0,8), (11,0)
   - Cliquer dans l'item `POINT` du menu construction (en haut)

- Entrer au clavier `x-5=0=0=8=11=0=`
- Cliquer dans l'item `VOIR_TOUT` (à droite)

On voit les 3 points. On va construire l'arc passant par ces 3 points:

- Cliquer dans l'item `ARC` du menu construction (en haut)
- Cliquer dans l'item `POINT` du menu désignation (en bas)
- Cliquer successivement près des points (11,0), (0,8), (-5,0)

5. Pour finir le domaine il suffit de construire les 2 segments allant de "l'infini" au naca.
   - Cliquer dans l'item `SEGMENT` du menu construction (en haut)
   - Cliquer près du point (-5,0)
   - Cliquer dans l'item `PRECEDENTE` du menu gestion de l'écran (à droite). On utilise le contexte d'affichage précédant qui met le naca plein écran.
   - Cliquer près du point (0,0) (en dessous à gauche pour être sûr de désigner le premier point).
   - Cliquer dans l'item `SUIVANTE` du menu gestion de l'écran (à droite), *on voit tous les objets.*
   - Cliquer près du point (11,0)

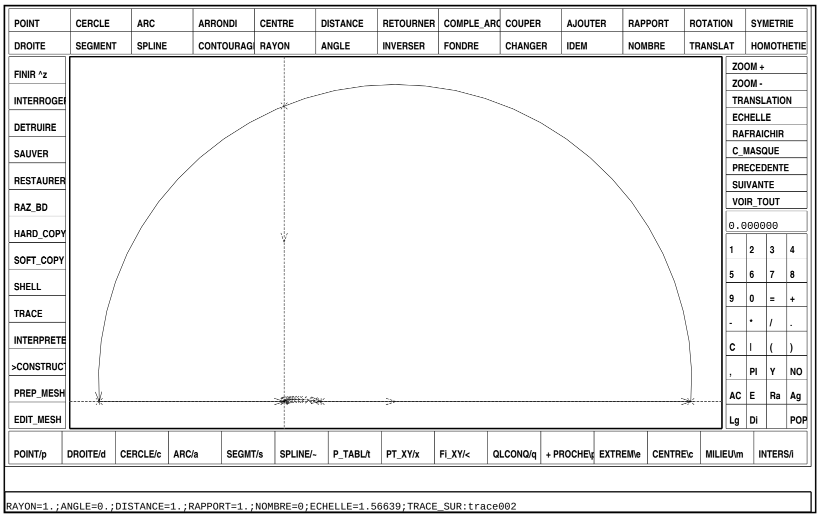

FIG. 7.4 - *Toute la géométrie du naca0012 et de l'infini*

On a fini théoriquement la construction du domaine géométrique (figures 7.4 et 7.5).

7.2.3 Discrétisation des contours

On passe dans l'application PREP_MESH pour définir la discrétisation des contours.

1. Cliquer dans l'item `PREP_MESH`
2. La discrétisation des lignes se fait comme suit :
   - Cliquer dans l'item `NB_INTERVAL` *En vue de définir le nombre d'intervalles*
   - Taper `26=`
   - Cliquer près de l'arc (attention, on est en mode de désignation QLCONQ donc il ne faut pas cliquer près du point (8,0)). L'arc est discrétisé en 26 intervalles.
   - Taper 39=

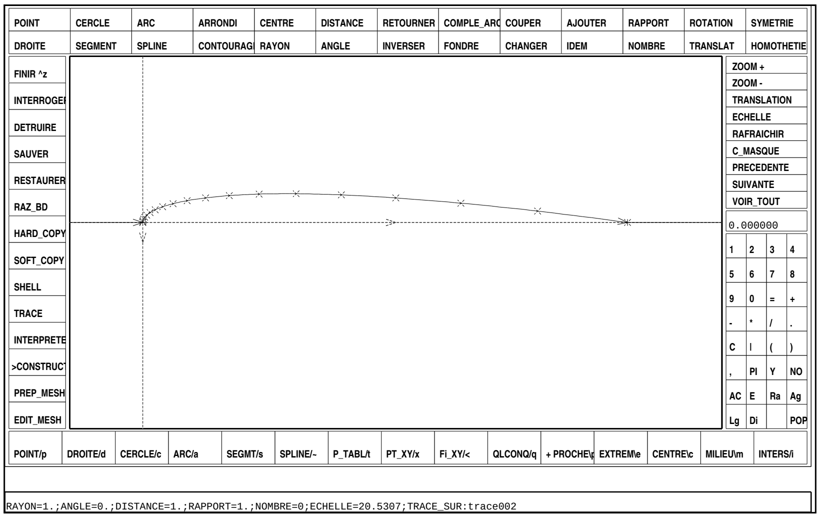

FIG. 7.5 - *Zoom autour du naca0012*

- Cliquer près du segment de gauche.
- Taper `30=`.
- Cliquer près du segment de droite.
- Taper `50=`.
- Cliquer dans l'item `SPLINE` du menu désignation.
- Cliquer dans la fenêtre graphique (il n'y a qu'une spline donc on la désignera).
- Cliquer dans l'item `RAISON` *pour changer la répartion de point sur les lignes*
- Taper `1.1=`
- Cliquer près de la spline.
- Cliquer dans l'item `SEGMENT` du menu désignation.
- Cliquer près du segment de droite
- Taper `1/1.2=`
- Cliquer près du segment de gauche

On a fini la définition de la dicrétisation des lignes (figures 7.6 et 7.7).

7.2.4 Génération et édition du maillage

On génére le maillage en passant dans l'application EDIT_MESH.

1. Cliquer dans l'item `EDIT_MESH` puis taper quatre fois `(CR)` pour répondre par défaut aux 4 questions. Un maillage s'affiche (figure 7.8).

   On définit des références en vue de la prise en compte des conditions aux limites : 1 sur l'infini Γ_∞, 2 sur le NACA, 3 au bord de fuite.

2. Cliquer dans l'item `MODIF_REF` du menu l'application

   - Taper `1=` *la reférence courante est 1*
   - Cliquer dans l'item `SOMMET` du menu désignation pour passer en mode de désignation de sommet
   - Cliquer dans la maillage près d'un sommet du maillage qui est sur l'arc et non une extrémité puis cliquer près des 2 extremités toujours dans le maillage. *Tous les sommets de l'arc ont pour référence 1.*

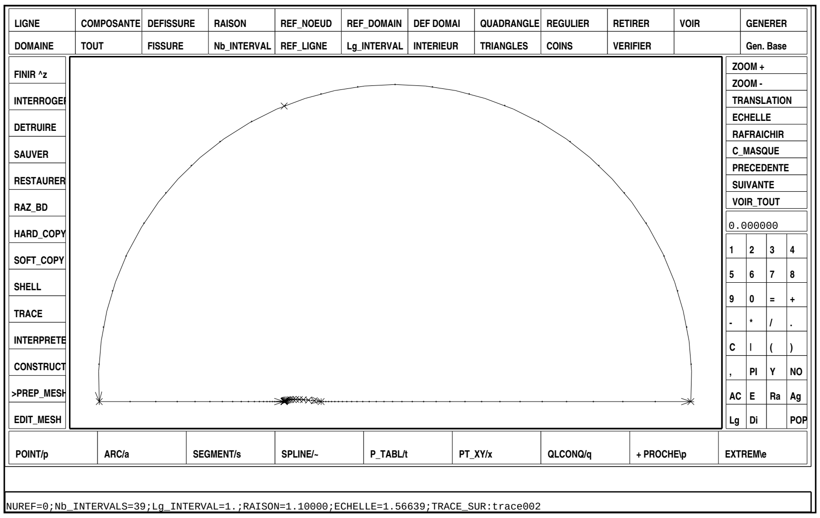

FIG. 7.6 - *Vu de la discrétisation de tous les contours*

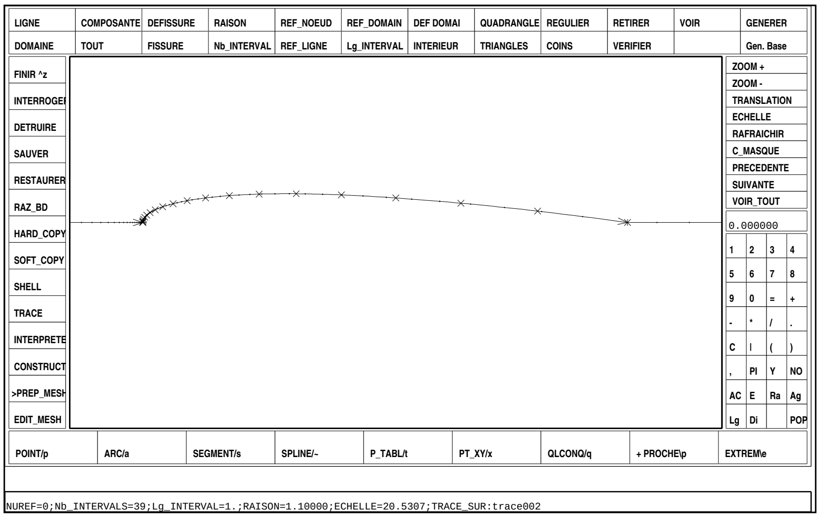

FIG. 7.7 - *Zoom de la discrétisation des contours autour du naca*

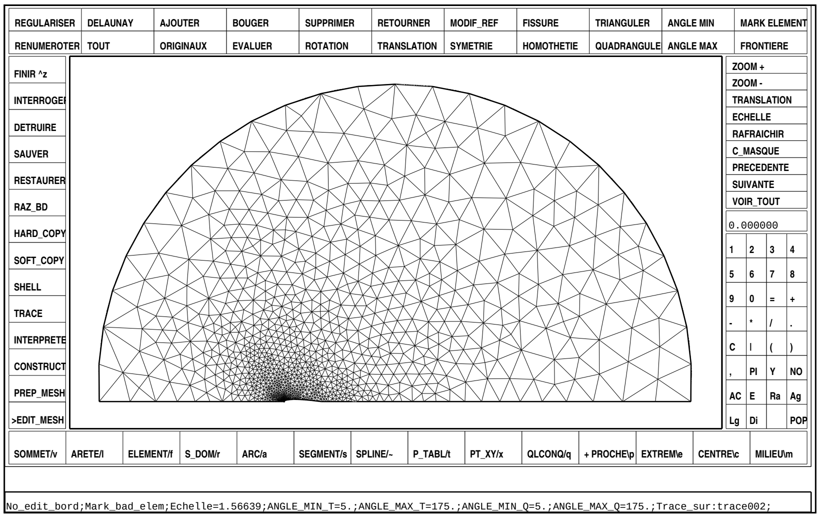

FIG. 7.8 - *Maillage du demi naca0012*

- Taper `2=` *la reférence courante est 2*
- On fait un zoom autour du naca
  - Cliquer dans l'item `ZOOM +` du menu gestion de l'écran
  - Cliquer dans l'item `P_TABL` du menu désignation pour passer en désignation de point curseur
  - Cliquer 2 fois en désignant les 2 coins opposés dans un rectangle fictif contenant le naca. Si le zoom n'est pas assez fort recliquer dans l'item `ZOOM +` puis entre 2 autres coins opposés Pour avoir un zoom autour de début du naca (coordonnées (0,0)):
    - Cliquer dans l'item `C_MASQUE` puis taper `x0=0=` pour centrer la fenêtre graphique autour du point (0,0) (Le x est le raccourci de POINT_XY)
    - Cliquer dans l'item `ECHELLE` puis taper `1000=` pour avoir une échelle de 1000

  (figure 7.9)

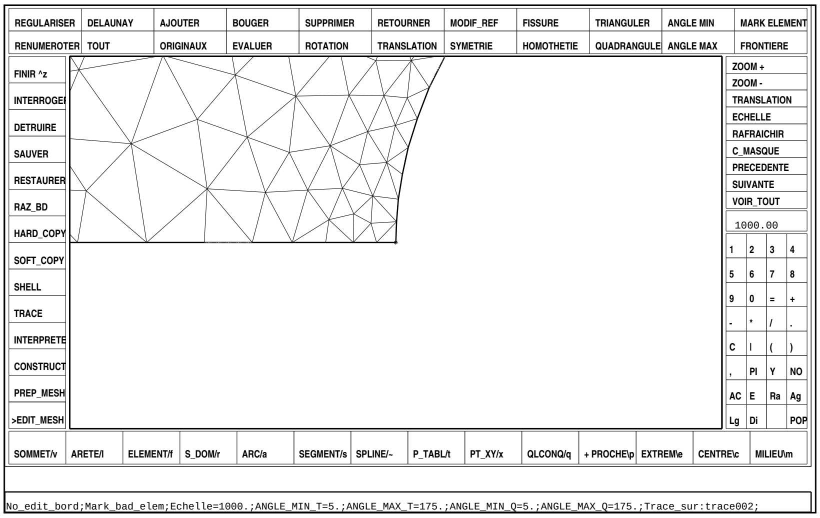

FIG. 7.9 - *Zoom de 1000 du Maillage du demi naca0012*

- Cliquer dans l'item `SOMMET` du menu désignation pour repasser mode de désignation de sommet
- En cas d'erreur (l'item MODIF_REF n'est plus marqué) pour continuer, il suffit de recliquer dans celui-ci et de taper `2=` pour redéfinir la référence courante.

- Cliquer dans le maillage près d'un sommet du maillage qui est sur le naca et non une extremité, puis cliquer dans le maillage près de l'extrémité aval (de coordonnées (0,0)).
- Cliquer dans l'item `C_MASQUE` puis taper `x0=1=` pour centrer la fenêtre graphique autour du point (0,1) qui est le bord de fuite.
- Taper `3=` *la reférence courante est 3*
- Cliquer dans le maillage près du bord de fuite.

On a fini l'entrée des références. On sauve le maillage du demi naca.

3. Cliquer dans l'item `SAUVER`, taper `mesh(CR)` puis `naca12_5(CR)`. Le fichier *naca12_5.mesh* est créé.

on symétrise le maillage pour obtenir un maillage complet, on commence par changer le point de vue pour voir tout le naca.

4. Cliquer dans l'item `C_MASQUE` puis taper `x1/2=0=` *pour centrer la fenêtre graphique autour du point* $(\frac{1}{2}, 0)$
5. Cliquer dans l'item `ECHELLE` puis taper `10=` pour avoir une échelle de 10
6. Cliquer dans l'item `SYMETRIE`
   - Cliquer dans l'item `SEGMENT` du menu désignation
   - Cliquer près d'un des 2 segments
   - Cliquer dans l'item `TOUT`, *tout le maillage est symétrisé, la partie symétrisée est affichée en traits pointillés (figure 7.10).*

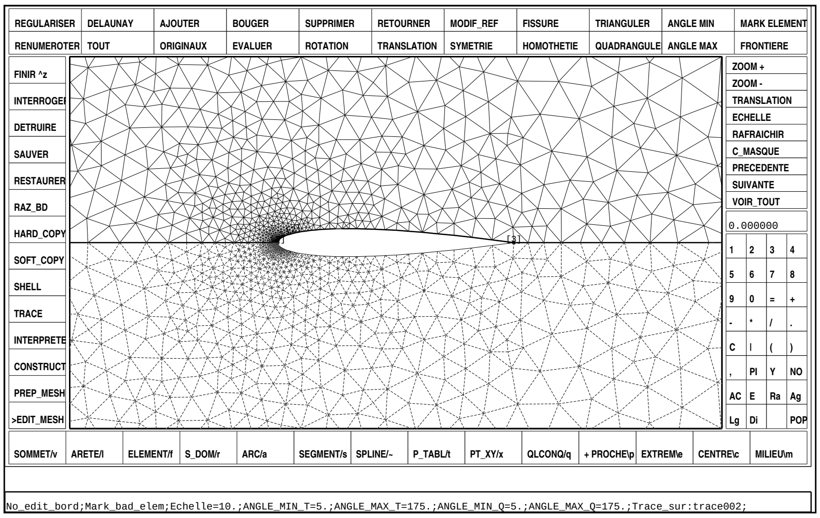

FIG. 7.10 - *Zoom de 10 du maillage du naca0012 symétrisé*

   - Cliquer dans l'item `EVALUER` pour générer effectivement le maillage total (figure 7.11).

On sauve le maillage du naca.

7. Cliquer dans l'item `SAUVER`, taper `mesh(CR)`, taper `naca12(CR)` le fichier *naca12.mesh* est créé

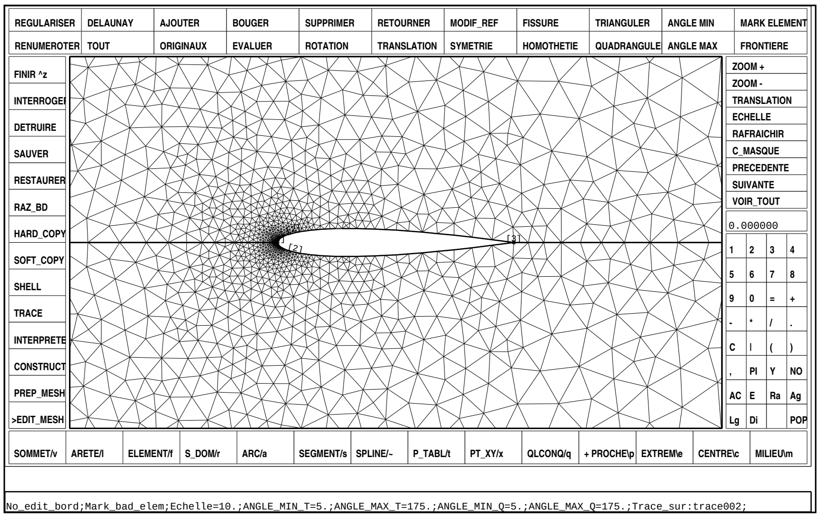

FIG. 7.11 - *Zoom de 10 du maillage du naca0012 complet*

Chapitre 8

Limitations et Bogues

- Fichiers pouvant être créés par le système
  Si xx est le nom donné par l'utilisateur, les noms des fichiers qui peuvent être créés à partir de xx sont :

traceNNN.emc2_trace	Trace des actions (noms générés automatiquement).
xx.emc2_trace	Trace des actions (nom spécifié par user).
xx.emc2_bd	Sauvetage en CONSTRUCTION ou en PREP MESH
xx_bak.emc2_bd	Fichier .bak du précédant.
xx.data	Fichier pour module APNOPO (il génèrera xx.nopo).
xx.nopo	Fichier généré par EDIT MESH.
xx.mesh	Fichier généré par EDIT MESH.
xx.am	Fichier généré par EDIT MESH.
xx.am_fmt	Fichier généré par EDIT MESH.
xx.ambda	Fichier généré par EDIT MESH.
xx.set	Pour liaison avec le logiciel VISIL.
xx.bas	Pour liaison avec le logiciel VISIL.

- Initialisation de variables globales
  Nombre maximal
  - d'éléments de BD MXBDX=5000
  - de points de discrétisation sur un élément de BD MXNOD=1000
  - de points définissant une spline MXPDEF=200
  - de sommets du maillage MBPMXX=15000
  - de triangles du maillage NBTMXX=2*NBPMXX-2=29998
  - d'arêtes du maillage définissant les frontières NBAMXX=5000
  - de sous domaines MBSDMXX=1000
  - d'éléments de listes MXLISTX=20000

  Toutes ces variables sont initialisées dans le programme `emc2.f` (répertoire `ppal`), exepté MXNOD et MXPDEF, qui sont elles initialisées dans le fichier d'include `emc2_commons_g.ins` (répertoire `include`).
- Il faut faire attention aux superpositions d'éléments de BD (exemple: une droite et un segment de même support) quand on désigne en mode quelconque car le resultat de la désignation est imprévisible.
- Lors de la commutation de l'application CONSTRUCTION à l'application PREP_MESH, si des arcs ou des segments ou des splines se superposent, on détruit l'un des deux.

- Si le mailleur, appelé au niveau de la commutation dans l'application EDIT_MESH, ne veut pas construire le maillage, c'est selon toute probabilité parce que :
  - **il faut couper explicitement tout les segments, arcs, splines** i.e. une extrémité de segment, arc, ou spline ne doit pas être sur une courbe non coupée.
  - lorsque l'on construit des arcs avec des propriétés de tangence, il peut y avoir des problèmes d'erreur d'arrondi. Pour éviter cela, on peut commencer par construire les points de tangence puis utiliser ceux-ci.
  - la précision du mailleur de 1/32000 relativement à la plus grande longueur ou largeur. Si le pas de discrétisation demandé est trop fin, le mailleur peut soit retourner des triangles ou soit refuser de mailler (il y a des points confondus)
  - une courbe n'est pas suffisament discrétisée (la frontière est croisée).

  le type de l'erreur est : il y a des points confondus, ou la frontière est croisée
- Le mailleur ne respecte généralement pas les symétries du domaine, si l'on veut respecter ces symétries : il faut mailler la partie minimale, puis générer le maillage total en utilisant les transformations de l'application EDIT_MESH.
- Attention: l'utilisation des transformations de l'application EDIT_MESH peut permettre la construction de sous domaines se chevauchant. Les résultats sont alors imprévisibles.
- Attention: les points d'un maillage lus au niveau de l'application EDIT_MESH ne sont pas recollés avec les points des maillages préexistants.
- Attention: Toutes les modifications d'un maillage sont perdues si l'on sort de l'application EDIT_MESH. On perd le maillage et on ne conserve que ses contours.
- Attention: en raison de problème fortran (INQUIRE), il est impossible d'exécuter emc2 dans un répertoire où l'on a pas d'accés en écriture (le programme essaie d'ouvrir 999 fichiers de "TRACE" (voir 3.3.10) ce qui prend du temps)
- Si vous perdez des informations dans la fenêtre graphique, le seul moyen pour rafaichir la fenêtre est de la retailler puis de cliquer à l'intérieur.

Chapitre 9

Structure interne des données

On a utilisé un système de listes à deux éléments que l'on appellera le CAR et le CDR. L'adresse vide est NIL=0 (parameter). Dans les schémas suivants le CAR sera toujours à gauche et le CDR à droite.

CAR	CDR

Les atomes libres sont dans une liste d'atomes libres, la tête de cette liste est FREE. Les atomes libres sont chaînés par leur CDR.

9.1 Description de la B.D. pour l'application CONSTRUCTION

TYPE	1	2	3	4	5	NUREF D & G	NUREF1 D & G	NUREF2 D & G	NBNODE	RAISON	FISSUR T/F
POINT=0	X	Y				nuref			>=2	>0	
RAYON>0	XC	YC							>=2	>0	
DROITE	A	B	C						>=2	>0	
SEGMENT	X1	Y1	X2	Y2		nuref	nuref1	nuref2	>=2	>0	
ARC	XC	YC	X1	Y1	ANGLE	nuref	nuref1	nuref2	>= 2|3|4	>0	
SPLINE	NB	PT *				nuref	nuref1	nuref2	>=3|4	>0	

```
                     |
 |----------<-----|   Liste des NB points de definition de la SPLINE
 V
 |------->|  *  |  *-------------->|  *  |  *---------->|  *  |  nil  |
             |                         |                    |
             V                         V                    V
       point en B.D.             point en B.D.        point en B.D.
```

FIG. 9.1 - *Description de la B.D. pour l'application CONSTRUCTION*

La figure 9.1 décrit la B.D. pour l'application CONSTRUCTION avec,

TYPE le type de l'élément de B.D.

- POINT = 0. POINT X,Y.
- RAYON > 0. CERCLE de centre XC,YC et de rayon RAYON.

- DROITE = -1. DROITE d'équation $A*x+B*y+C=0$
  Avec $A*A+B*B=1$
- SEGMENT = -3. SEGMENT allant du point X1,Y1 au point X2,Y2.
- ARC = -2. ARC de centre XC,YC passant par le point X1,Y1 et faisant un angle ANGLE à partir de ce point.
- SPLINE = -4. SPLINE.
- VIDE = -1000. élément vide.

et pour SPLINE,

- NB est le nombre de points de définition.
- PT est le pointeur sur la liste des POINTS de définition de la SPLINE.
- NUREF est le numéro de référence de l'élément (ARC ou SEGMENT). Par défaut NUREF=0.
- NUREF1 est le numéro de référence de l'extrémite 1 de l'élément (ARC ou SEGMENT). Par défaut NUREF1=0.
- NUREF2 est le numéro de référence de l'extremite 2 de l'élément (ARC ou SEGMENT). Par défaut NUREF2=0.
- NBNODE est le nombre de points intermédiaires de l'élément (extrémites comprises). Par défaut NBNODE=2.
- RAISON est la raison de la progression géométrique répartissant les NBNODE-2 points de l'extrémite 1 à l'extrémite 2. Par défaut RAISON=1.

9.2 Description de la B.D. pour l'application PREP MESH

La figure 9.2 décrit la B.D. pour l'application PREP MESH où

TYPE est le type de l'élément de B.D.

- POINT = 0. POINT X,Y.
- SEGMENT = -3. SEGMENT allant du point X1,Y1 au point X2,Y2.
- ARC = -2. ARC de centre XC,YC passant par le point XD,YD et faisant un angle ANGLE à partir de ce point.
- SPLINE = -4. SPLINE.
- VIDE = -1000. élément vide.

NUREF est le numéro de référence de l'élément (ARC ou SEGMENT).

ADP1 pointeur dans la B.D. sur le POINT extrémite 1 de l'élément (Si l'élément n'est pas fissuré, on a dans le NUREF de ce POINT le numéro de référence de l'extrémite 1 de l'élément).

ADP2 pointeur dans la B.D. sur le POINT extrémite 2 de l'élément ou le dernier POINT de la SPLINE. (Si l'élément n'est pas fissure, on a dans le NUREF de ce POINT le numéro de référence de l'extrémite 2 de l'élément).

NBNODE est le nombre de points intermédiaires de l'élément (extrémités comprises).

RAISON est la raison de la progression géométrique répartissant les NBNODE-2 points de l'extrémité 1 à l'extrémité 2.

CONX(2) pointeurs dans la B.D. Il permet le chaînage circulaire des éléments d'une même composante connexe. Pour un même élément, il y a deux et seulement deux composantes connexes qui contiennent cet élément : la composante qui est à gauche et celle qui est à droite (La droite et la gauche sont des notions relatives au sens de parcourt de l'élément). Il y a donc deux chaînages symbolisés par : CONXG et CONXD.

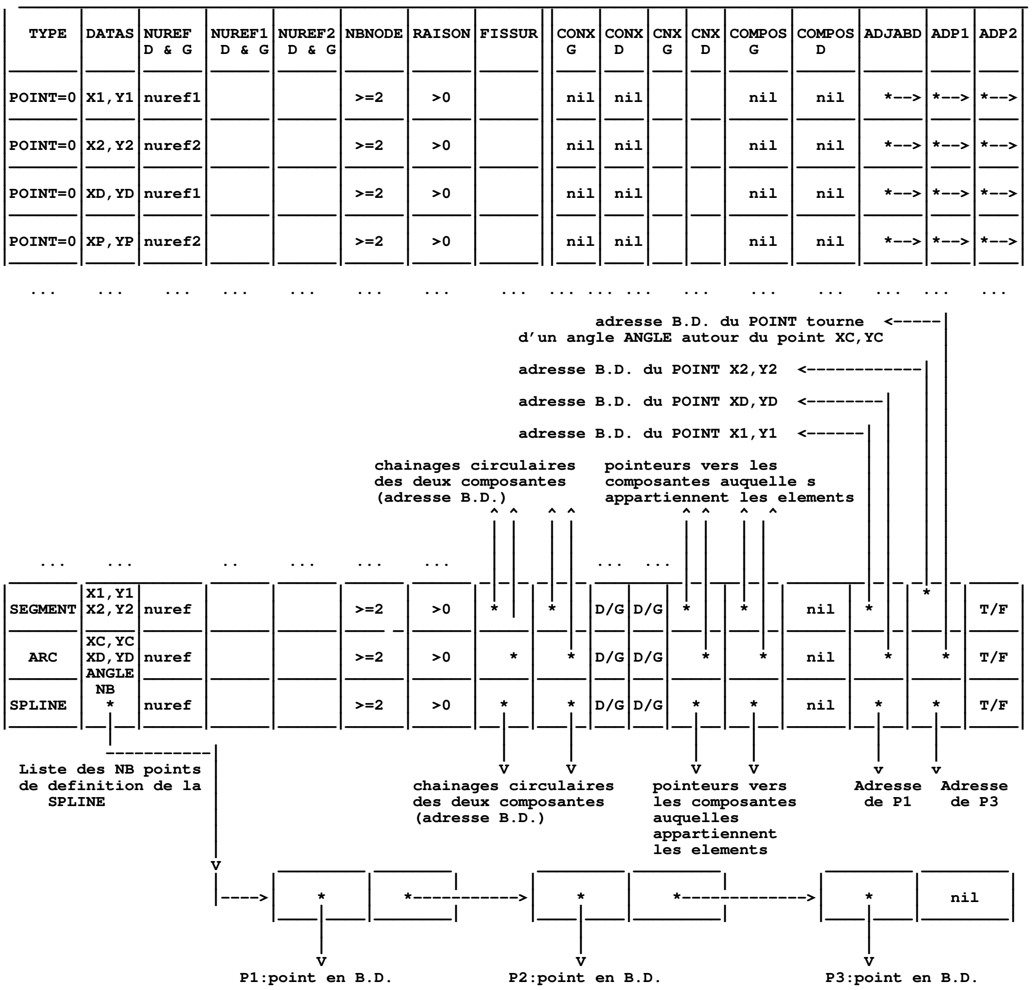

FIG. 9.2 - *Description de la B.D. pour l'application PREP MESH*

CNX(2) indique le CONX à utiliser pour l'élément suivant (CONXG ou CONXD) selon que CNX vaut GAUCHE/DROIT (GAUCHE=1, DROIT=2)(parameter).

COMPOS(2) pointeurs inverses vers les deux composantes auxquelles appartient l'élément. CNXG est relatif à CONXG et à COMPOSG (Ils n'existent que pour les ARCS et les SEGMENTS). CNXD est relatif à CONXD et à COMPOSD (Ils n'existent que pour les ARCS et les SEGMENTS).

ADJABD Tête de liste des éléments passant par le point (Il n'existe que pour les POINTS). (Voir description de cette liste plus loin).

9.2.1 Description de la liste des éléments passant par le point i

La tête de la liste des éléments passant par un point est ADJABD.
La figure 9.3 décrit la liste des éléments passant par le point i.

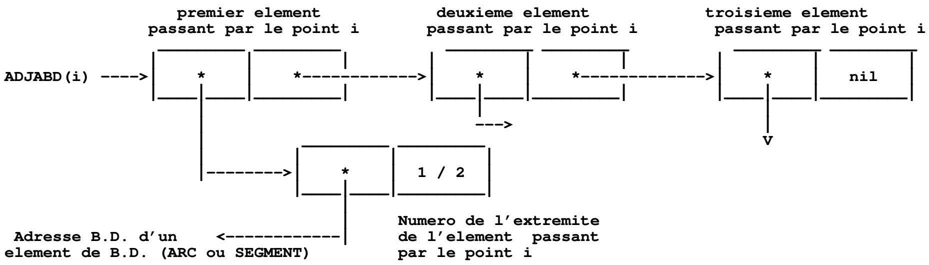

FIG. 9.3 - *Description de la liste des éléments passant par le point i*

9.2.2 Description de la liste des composantes

La tête de la liste des composantes est COMP.
La figure 9.4 décrit la liste des composantes.

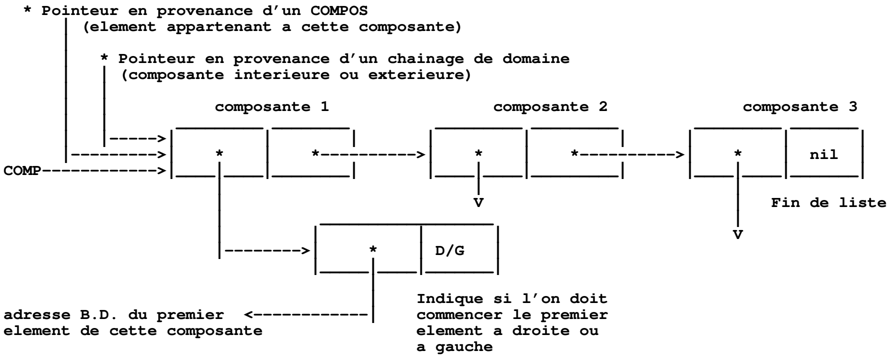

FIG. 9.4 - *Description de la liste des composantes*

9.2.3 Description de la liste des domaines

La tête de la liste des domaines est SDOMN.
La figure 9.5 décrit la liste des domaines

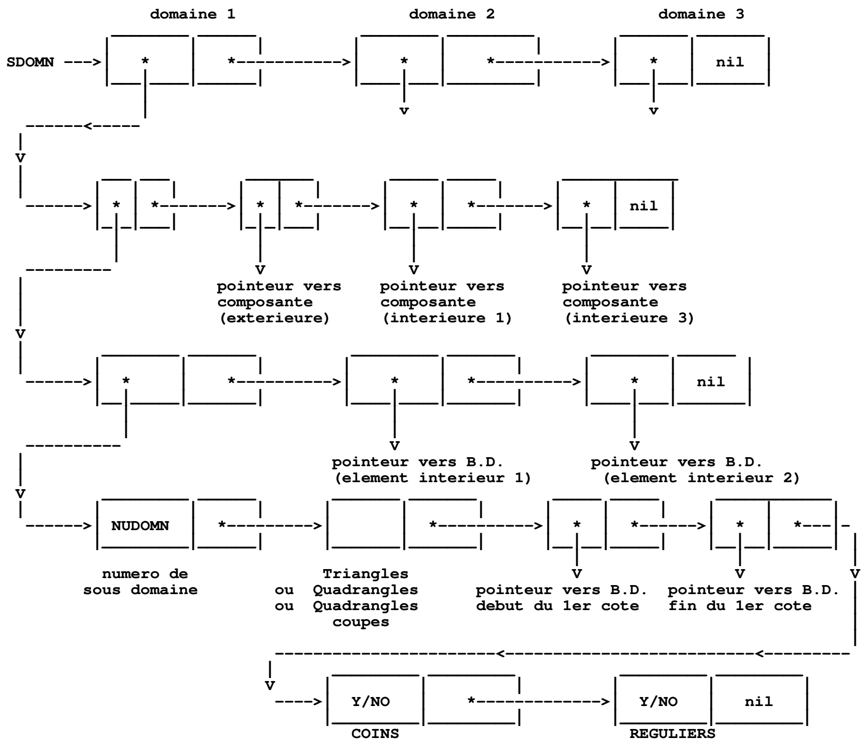

FIG. 9.5 - *Description de la liste des domaines*

9.3 Description des B.D. pour l'application EDIT_MESH

La base de données est la même que dans l'application PREP_MESH, et la structure de donnée SD_MESH est formée comme suit:

nbs est le numéro du dernier sommet

nbt est le numéro du dernier triangle

nba est le nombre d'arêtes qui sont sur des courbes (segment, arc, spline) que l'on appellera arête frontières

nbsd est le nombre de sous-domaines

nbsrft est le nombre de sommets référencés par les éléments

nbtria est le nombre de triangles

nbquad est le nombre de quadrangles

nbtrou est le nombre de trous dans le maillage

freetr est la tête de la liste des triangles détruits

finbd3 est le numéro du dernier élément de BD de PREP_MESH

Les Sommets sont définis avec les 4 tableaux en parallèle suivants:

cr(1:2,nbs) est le tableau des coordonnées des sommets.

nsorig(1:nbs) est le tableau donnant le numéro du sommet d'origine pour les sommets qui sont sur une fissure (les coordonnées réelles du sommet i sont $cr(1:2, nsorig(i))$, $cr(1:2,i)$ sont des coordonnées ne servant qu'à la visualisation de la fissure.

Remarque: si le sommet i n'est pas sur une fissure, on a $nsorig(i) = i$.

abcurv(1:nbs) est :

- 0 si le sommet n'est pas sur une courbe (segment, arc, spline), (le sommet est interne à un sous-domaine),
- abscisse du point sur la courbe si ce sommet n'est pas une extrémité de la courbe, 0 pour les extrémités de courbe. *remarque: les courbes sont toujours paramétrées de* 0 *à* 1.

refs(1:nbs) est le tableau donnant: le numéro dans BD de l'élément (point, segment, arc, spline) supportant le sommet si il existe , sinon 0.

Remarque: si le sommet est une extrémité de courbe alors l'élément choisi sera le point extrémité de courbe (c.f. ADP1 et ADP2 (Appendice 9.2))

Les éléments finis sont des triangles ou des quadrangles qui sont représentés de manière interne uniquement avec triangles auxquels on a ajouté la notion d'arête visible ou non afin de tracer les quadrangles. Les tableaux définissant les triangles sont:

nsea(1:6,nbt) définit pour chaque triangle les sommets et les arêtes ou les triangles adjacents si $nsea(1, ie) < 1$ alors le triangle ie n'existe pas et $reft(ie)$ donne le triangle détruit suivant, sinon le triangle ie existe et on a:

- $nsea(1:3, ie)$ donne pour chaque triangle les trois numéros de sommets tournant dans le sens trigonométrique.
- $d_i \overset{\text{def}}{=} nsea(i, ie), i = 4, 5, 6$, les 3 arêtes a_i du triangle numérotées de 4 à 6 ont pour sommets $nsorig(nsea(i-3, ie))$ et $(nsorig(nsea(mod(i, 3) + 1, ie))$.
  - si $d_i < 0$ alors l'arête est frontière, et est définie dans $aretdb$, et a pour numéro $-d_i$;
  - sinon l'arête est interne et $d_i = 8 * t_a + a_{t_a}$ où t_a est le numéro du triangle adjacent à l'arête et a_{t_a} est le numéro de l'arête dans t_a.

anovue(1:nbt) donne pour chaque triangle la visibilité de ses 3 arêtes, si $anovue(i)$ est égal à:

0 toutes les arêtes du triangle i sont vues,

1 seule l'arête 4 du triangle i n'est pas vue,
2 seule l'arête 5 du triangle i n'est pas vue,
3 seule l'arête 6 du triangle i n'est pas vue,

reft(1:nbt) est:

- soit le tableau des numéros du sous-domaine contenant le triangle;
- soit le tableau des chaînages des triangles d'un sous domaine, (reft(ie) donne le triangle suivant du sous-domaine, si $reft(ie) <> finsd = 2^{30}$,voir aussi le tableau $tetdt$ pour les têtse de liste des sous-domaines).

Les arêtes frontières du maillages sont définies comme suit:

aretbd(1:2,nba) donne pour chaque arête ses 2 numéro de sommets;

areadj(gauche:droite,nba) : $d_i \stackrel{\text{def}}{=} areadj(i,j), i = gauche, droite$ même définition que pour $nsea$;

refa(nba) donne pour chaque arête l'adresse dans la bd du support de celle-ci (le support existe toujours par définition).

Les sous-domaines sont définis avec les tableaux suivants:

refsd(nbsd) donne les numéros de réference des sous-domaines, si $refsd(i) = videsd = -2^{30}$ alors le sous-domaine n'existe pas.

trfsd(3,3,i) donne la transformation appliquée au sous domaine $ptorsd(i)$ pour obtenir ce sous domaine $i(i \in 1, .., , nbsd)$

strfsd(i) donne le signe de la transformation (exemple: -1 pour une symétrie par rapport à une droite)

ptorsd(i) donne le numéro sur le sous-domaine origine (si ptorsd(i)=i alors le sous-domaine i n'est pas transformé *(sous domaine original)*)

tetsd(nbsd) est un tableau donnant par sous-domaine le premier triangle de la liste des triangles de ce sous-domaine (c.f $reft$)

9.4 Description des maillages AM, AM_FMT, AMDBA

Le maillage n'est formé que de triangles et peux être défini avec les 2 entiers et les 4 tableaux suivants :

nbt est le nombre de triangles

nbs est le nombre de sommets

nu(1:3,1:nbt) est un tableau d'entiers donnant pour chaque triangle les trois numéros de sommets dans le sens trigonométrique.

c(1:2,nbs) est un tableau de réels donnant les 2 coordonnées pour chaque sommet.

refs(nbs) est un tableau d'entier donnant les numéros de reférence des sommets.

reft(nbs) est un tableau d'entier donnant les numéros de reférence des triangles.

Remarque: les sommets non référencés par des triangles seront éliminés, la numérotation des sommets dans ce cas sera compressée.

9.4.1 Le fichier AM

La lecture d'un fichier `xxx.am` est faite comme suit :

```
open(1,file='xxx.am',form='unformatted',status='old')
  read (1) nbs,nbt
  read (1)
+                 ((nu(i,j),i=1,3),j=1,nbt)
+                ,((c(i,j),i=1,2),j=1,nbs)
+                ,( reft(i),i=1,nbt)
+                ,( refs(i),i=1,nbs)

close(1)
```

9.4.2 Le fichier AM_FMT

La lecture d'un fichier `xxx.am_fmt` est faite comme suit :

```
open(1,file='xxx.am_fmt',form='formatted',status='old')
  read (1,*) nbs,nbt
  read (1,*)  ((nu(i,j),i=1,3),j=1,nbt)
  read (1,*)  ((c(i,j),i=1,2),j=1,nbs)
  read (1,*)  ( reft(i),i=1,nbt)
  read (1,*)  ( refs(i),i=1,nbs)
close(1)
```

9.4.3 Le fichier AMDBA

La lecture d'un fichier `xxx.amdba` est faite comme suit :

```
open(1,file='xxx.am_fmt',form='formatted',status='old')
  read (1,*) nbs,nbt
  read (1,*) (k,(c(i,k),i=1,2),refs(k),j=1,nbs)
  read (1,*) (k,(nu(i,k),i=1,3),reft(k),j=1,nbt)
close(1)
```

Chapitre 10

Index

Index

Bibliographie

[1] M.Bernadou and All, MODULEF: Une bibliothéque modulaire d'éléments finis, INRIA, 1988.

[2] P.G.Ciarlet, The finite element method for elliptic problem, North Holland 1978.

[3] J.L.Coulomb, Maillages 2D et 3D. Experimentation de la triangulation de Delaunay, Conférence on Automated mesh generation and adpatation, Grenoble 1987.

[4] P.L George, MODULEF: Génération automatique de maillage, Collection didactique n^o 2, INRIA, 1988.

[5] P.L George, MODULEF: Construction et modification de maillages, RT n^{o}104 INRIA 1988.

[6] P.L.George, F.Hermeline, Maillage de Delaunay d'un polyèdre connexe en dimension d. Extention à un polydédre quelconque, CEA-N-90,1989.

[7] F.Hermeline, Triangulation automatique d'un polyèdre en dimension N, RAIRO numerical analysis vol 16, n^o 3 1982.

[8] A.Jameson, T.J.Baker, N.P.Weatherill, Calculation of inviscid transonic flow over a complete aircraft. AIAA 24th Aerospace Sciences Meeting, Reno Nevada, USA, 1986.

[9] R.Lohner, P.Parikh, Generation of 3-D unstructured grids by the advancing front method, AIAA 26th Aerospace Sciences meeting, Reno Nevada, USA, 1988.

[10] A.Perronnet, Tétraèdrisation d'un objet multi-materiaux ou de l'exterieur d'un objet, R n^o 88005, LAN 189,Université Paris 6, 1988.

[11] J.Y.Talon, Algorithmes d'amélioration de maillages pour éléments finis en 2 et 3 dimensions, Conférence on Automated mesh generation and adaptation, Grenoble 1987.

[12] J.F.Thompson, A general three dimensional elliptic grid generation system on a composite block-structure, Computer Methods in Applied Mechanics and Enginneering, Vol 64, 1987.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399