



Le meilleur adaptatif bidimensionnel BL2D : manuel d'utilisation et documentation

Houman Borouchaki, Patrick Laug

► To cite this version:

Houman Borouchaki, Patrick Laug. Le meilleur adaptatif bidimensionnel BL2D : manuel d'utilisation et documentation. [Rapport de recherche] RT-0185, INRIA. 1995, pp.65. inria-00069986

HAL Id: inria-00069986

<https://hal.inria.fr/inria-00069986>

Submitted on 19 May 2006

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

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

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

***Le mailleur adaptatif bidimensionnel BL2D :
manuel d'utilisation et documentation***

Houman BOROUCHEKI et Patrick LAUG

N° 0185

Décembre 1995

PROGRAMME 6



*rapport
technique*



Le mailleur adaptatif bidimensionnel BL2D : manuel d'utilisation et documentation

Houman BOROUCHEKI et Patrick LAUG*

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

Rapport technique n° 0185 — Décembre 1995 — 65 pages

Résumé : Nous avons réalisé un mailleur adaptatif bidimensionnel appelé BL2D. Ce rapport contient à la fois le manuel d'utilisation du mailleur et la description des principales caractéristiques du logiciel.

Mots-clé : maillage adaptatif, maillage anisotrope, métrique riemannienne, structure de données.

(Abstract: pto)

Ce travail a été effectué en partie dans le cadre du programme de recherche "GÉNIE" (DASSAULT AVIATION - INRIA). Nous remercions P.L. George, F. Hecht et E. Saltel pour leur amicale coopération.

*. E-mail: Patrick.Laug@inria.fr

The BL2D Bidimensional Adaptive Mesh Generator: User's Manual and Documentation

Abstract: We have made a bidimensional adaptive mesh generator named BL2D. This report includes both the user's manual of the mesh generator and the description of the software main features.

Key-words: adaptive mesh, anisotropic mesh, riemannian metric, data structure.

Table des matières

1	Présentation générale	6
1.1	Principales étapes d'utilisation	6
1.2	Exemple complet d'utilisation	8
1.3	Gestion des fichiers	17
2	Programmes	18
2.1	Programme blambda	18
2.2	Programme bldraw	19
2.3	Programme blemc2	20
2.4	Programme blestim	20
2.5	Programme blinterpol	21
2.6	Programme blnc	21
2.7	Programme blms	21
2.8	Programme blprems	22
2.9	Programme blprepro	22
2.10	Programme blsmooth	22
2.11	Programme blverif	23
3	Formats	24
3.1	Format auxiliaire AMDBA	27
3.2	Format de sortie C	29
3.3	Format d'entrée G	32
3.4	Format auxiliaire GOV	41
3.5	Format d'entrée H	42
3.6	Format auxiliaire IS	45

3.7	Format auxiliaire MC	47
3.8	Format de sortie MS	49
3.9	Format auxiliaire P	51
3.10	Format auxiliaire SMOOTH	54
4	Structures de données	56
4.1	Structure g	57
4.2	Structure c	57
4.3	Structure s	58
4.4	Structure h	58
5	Procédures	59
5.1	Lecture et écriture	59
5.2	Références physiques	60
5.3	Splines	62

Préface

Le logiciel BL2D réalise des maillages en deux dimensions avec ou sans adaptation, isotropes ou anisotropes. Il comprend deux composants principaux :

- Un mailleur de segments courbes dans l'espace \mathbb{R}^2 . Les méthodes employées pour ce mailleur sont décrites dans [1].
- Un mailleur de domaines dans l'espace \mathbb{R}^2 . Ce mailleur est de type Delaunay généralisé [2, 3].

Ce rapport s'adresse à la fois aux utilisateurs du logiciel et aux développeurs (ces derniers pouvant y ajouter de nouvelles fonctionnalités ou modifier ses caractéristiques). Il est divisé en cinq chapitres :

1. une présentation générale de l'utilisation du logiciel,
2. une description détaillée des programmes que les utilisateurs peuvent activer,
3. une description des formats de fichiers d'entrée, de sortie ou auxiliaires,
4. une description des principales structures de données qui sont accessibles aux développeurs de logiciel,
5. une description de quelques procédures qui sont également accessibles aux développeurs.

Chapitre 1

Présentation générale

1.1 Principales étapes d'utilisation

Le logiciel BL2D réalise des maillages en deux dimensions avec ou sans adaptation, isotropes ou anisotropes. L'objectif de ce chapitre est d'apprendre comment utiliser les différentes fonctionnalités du logiciel. Nous présentons dans ce chapitre un exemple typique d'utilisation : la simulation d'un problème physique qui est modélisé par une formulation mathématique, supposée connue, dans un domaine du plan.

Nous devons tout d'abord définir les données géométriques et physiques du domaine. La frontière du domaine est définie par des segments droits ou courbes, que nous représentons par des *splines*. Nous devons ensuite mailler les splines et le domaine, et effectuer les calculs par éléments finis. Enfin, selon les résultats obtenus, nous devons adapter le maillage et recommencer les calculs, ou bien arrêter le processus.

En pratique, ceci revient à activer plusieurs programmes exécutables. Les programmes du logiciel BL2D sont essentiellement `blsmooth` (calcul des splines), `blmc` (mailleur de courbes) et `blms` (mailleur de domaines). Les principales

étapes à effectuer sont présentées rapidement ci-dessous (et sont précisées dans la section 1.2) :

1. Définir les données géométriques et physiques, ainsi que les données qui gouvernent le premier maillage. Pour cela, deux fichiers initiaux doivent être créés :
 - le fichier “G”, qui contient les données géométriques et physiques d’un domaine du plan,
 - le fichier “HG”, qui permet de gouverner les mailleurs de courbes et de domaines.

En option, le fichier supplémentaire “GOV” permet de gouverner plus finement le mailleur de domaines.

Le moyen de créer ces fichiers est au choix de l’utilisateur. Le moyen le plus commode est d’utiliser une interface avec un système interactif graphique, par exemple l’éditeur de maillages et de contours bidimensionnels EMC² [8]. Il est également possible d’utiliser le programme `blprepro` qui fait partie du logiciel BL2D, mais qui n’est pas interactif. Enfin, le moyen le plus rudimentaire, mais qui ne convient que dans certains cas, consiste simplement à utiliser un éditeur de textes. Les formats de tous ces fichiers sont décrits au chapitre 3.

2. Activer le programme de calcul des “splines” (`blsmooth`). Une spline est un segment droit ou courbe défini par des points de contrôle. Le programme `blsmooth` sert à définir, de manière aussi exacte que possible, la frontière du domaine et éventuellement les segments imposés.
3. Activer le mailleur de courbes (`blmc`) qui subdivise chaque spline en un ensemble d’arêtes, qui sont ensuite transmises au mailleur de domaines.
4. Activer le mailleur de domaines (`blprems` et `blms`) qui subdivise chaque sous-domaine en un ensemble de triangles, et crée ainsi le maillage demandé.
5. Activer le programme de calcul par éléments finis. Si le maillage doit être adapté, créer un nouveau fichier pour gouverner le mailleur et retourner à l’étape 3. Sinon, sortir de la boucle.

1.2 Exemple complet d'utilisation

Dans cette section, nous présentons en détail la succession d'étapes précédente. Après quelques définitions préalables, nous montrons comment réaliser un maillage initial isotrope, puis des maillages adaptés anisotropes.

Définitions et notations

Nous supposons que l'utilisateur travaille sous système UNIX (d'autres systèmes sont utilisables, moyennant quelques adaptations). Les fichiers sont organisés de manière hiérarchique en *répertoires*.

Shell

Un *shell* est un interpréteur de commandes. Pour inviter l'utilisateur à taper une commande, il imprime un caractère d'appel (*prompt*) noté ici `%`.

En outre, le *shell* est programmable : un *shell-script* est un programme écrit dans le langage du *shell*.

Répertoire de distribution

Le répertoire de distribution contient l'ensemble des programmes et des données du logiciel BL2D. Il est noté `~b12d` et contient en fait plusieurs sous-répertoires :

- `~b12d/s` : programmes sources,
- `~b12d/machine` (*machine* = `alpha`, `hp700`, `ibm`, `sun`, ...) : programmes exécutables pour une machine cible particulière (respectivement DEC Alpha, HP série 700, IBM RS6000, Sun 4, ...),
- `~b12d/data` : données fournies à titre d'exemple.

Il est recommandé d'ajouter le sous-répertoire `~b12d/machine` à la variable d'environnement `path` (qui indique au *shell* où trouver les commandes).

Répertoire de travail

Le répertoire de travail est un espace disque quelconque où l'utilisateur peut créer des fichiers. Il est noté ici `/tmp`, mais il est évidemment possible d'utiliser un répertoire personnel créé par la commande `mkdir`.

Création d'un maillage initial isotrope

Rappelons que les données nécessaires au logiciel BL2D peuvent être fournies par un système graphique interactif, par exemple EMC². Ici, les données initiales sont spécifiées par un fichier de texte créé par l'utilisateur (étapes 1 et 2 ci-dessous). À partir de ces données, un maillage isotrope est obtenu (étapes 5 à 9 ci-dessous). Ces différentes étapes sont détaillées ci-dessous :

1. Aller dans le répertoire de travail `/tmp`, et y créer le fichier `x.0.p` qui est imprimé à la fin de la section 3.9. Ce fichier est normalement distribué avec le logiciel BL2D, et il suffit alors de le copier (voir les commandes UNIX ci-dessous). Sinon, il est assez rapide de le taper à nouveau en sachant qu'il est en format libre [5] : le nombre exact d'espaces entre les données n'est pas important, et tout texte qui commence par les deux caractères `--` est un commentaire ignoré par le logiciel.

```
% cd /tmp
% cp ~/bl2d/data/quart/x.0.p .
% emacs x.0.p &
```

2. Activer le programme `blprepro` (cf. section 2.9). Les données sont `x` et `0` (voir le dialogue ci-dessous). Le programme lit le fichier `x.0.p` et écrit les deux fichiers initiaux `x.0.g` et `x.0.hg` (FIG. 1.1) :

- Le fichier `x.0.g` (données géométriques et physiques) décrit un domaine dont la frontière est représentée par 5 splines. Les splines (1) et (5) sont des segments de droites de longueur 1. Les splines (2) et (4) sont des segments de droites de longueur 0.5. La spline (3) est définie par les points de contrôle 5, 6, 7, 8, 9 dans le but d'approcher un quart de cercle. Le point 4 est un point imposé à l'intérieur du

domaine (ce point sera obligatoirement un sommet de triangle dans tous les maillages de domaines).

- Le fichier `x.0.hg` (informations pour gouverner le mailleur) demande une taille $h = 0.1$ aux points 4 (imposé), 1, 2, 6, 8, 3 (extrémités) et 7 (point de contrôle interne à une spline).

```
% blprepro
  Prefixe ?
x
  Iteration ?
0
...
```

3. Facultatif: activer le programme de vérification `blverif` (cf. section 2.11). Les données sont `x` et `0`. Normalement, le programme ne doit afficher aucun message d'erreur.
4. Facultatif: activer le programme de tracé `bldraw` (cf. section 2.2). Les données sont dirigées par des menus. Le premier menu est le choix du terminal de sortie graphique. En donnant une ligne vide, on obtient un terminal par défaut. En tapant ensuite les données ci-dessous (qui sont expliquées dans la section 2.2), on obtient un tracé à partir du fichier initial `x.0.g` (FIG. 1.1 en haut).

```
% bldraw
...
Fichiers : g | hg | smooth | c (mc) | hc | s (ms) | hs | h
Autres : b base | t test | q quitter bldraw
g
  Prefixe ?
x
  Iteration ?
0
  p points | s splines (polygone de controle) | q quitter draw_g
s
  d dessin | n numeros | t tangentes | q quitter draw_gs
d
  d dessin | n numeros | t tangentes | q quitter draw_gs
```

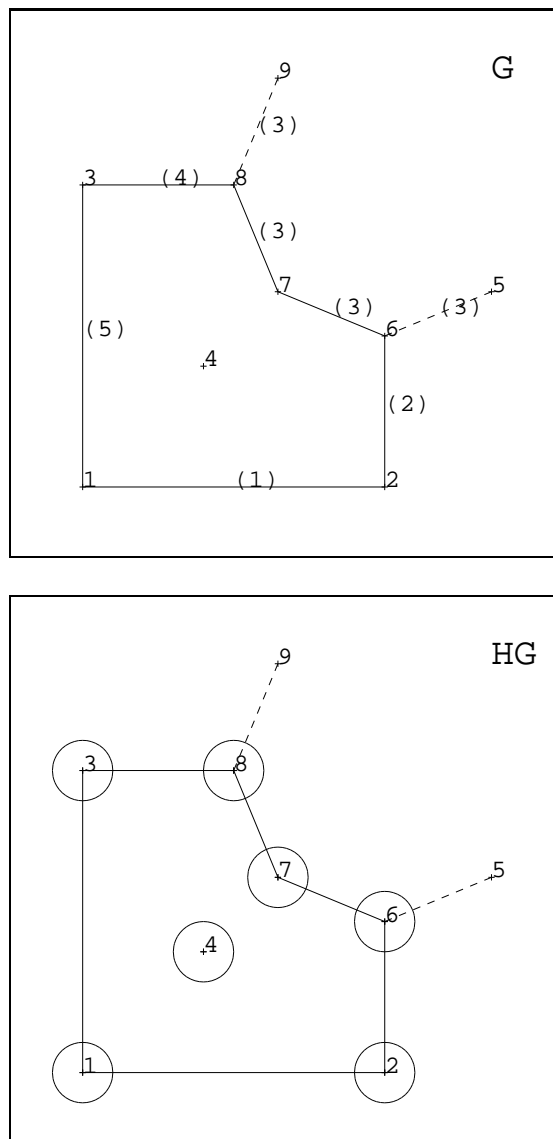


FIG. 1.1 - Tracés à partir des fichiers initiaux x.0.g et x.0.hg.

```

q
  p points | s splines (polygone de controle) | q quitter draw_g
q
  Fichiers : g | hg | smooth | c (mc) | hc | s (ms) | hs | h
  Autres : b base | t test | q quitter bldraw
q
  TAPER c POUR CONFIRMER
c

```

5. Activer le programme de calcul des splines `blsmooth` (cf. section 2.10). Les données sont `x` et `0`.
6. Activer le mailleur de courbes `blmc` (cf. section 2.6). Les données sont `x` et `0`. Le programme affiche, pour chaque spline, le nombre d'arêtes créées et la longueur d'une arête dans la métrique souhaitée (cette longueur doit être voisine de 1).

```

% blmc
  Prefixe ?
x
  Iteration ?
0
  spline 1 nb 10 m/nb  1.0000000000000000
  spline 2 nb  5 m/nb  1.0000000000000000
  spline 3 nb  8 m/nb  0.9773936540696194
  spline 4 nb  5 m/nb  1.0000000000000000
  spline 5 nb 10 m/nb  1.0000000000000000

```

7. Activer le programme `blprems` (cf. section 2.8), qui crée les données du mailleur de domaines. Les données sont `x` et `0`.
8. Activer le mailleur de domaines `blms` (cf. section 2.7). Les données sont `x` et `0`. Le programme affiche de nombreuses statistiques sur les qualités des éléments et les temps écoulés.

```

% blms
...
prefix file --> x
adaption (0,i) ----> 0
...

```

9. Le maillage initial isotrope du domaine est obtenu. En activant à nouveau le programme `bldraw` (cf. étape 4) avec le dialogue ci-dessous, on peut visualiser ce maillage initial (FIG. 1.2, $i = 0$).

```
% bldraw
...
Fichiers : g | hg | smooth | c (mc) | hc | s (ms) | hs | h
Autres : b base | t test | q quitter bldraw
s
  Prefixe ?
x
  Iteration ?
0
  p points | t triangles | q quitter draw_s
t
...
```

Au cours des étapes précédentes, les programmes du logiciel BL2D créent des fichiers dans différents formats (cf. chapitres 2 et 3). Dans la mesure du possible, les programmes ne dupliquent pas les informations déjà existantes, et ainsi chaque fichier créé ne contient que des informations nouvelles. Les principaux fichiers créés sont le maillage de courbes (format C) et le maillage de domaines (format MS). On peut par exemple extraire une partie du fichier initial G et des nouveaux fichiers C et MS pour obtenir un fichier au format AMDBA. Ce dernier regroupe un certain nombre de données nécessaires à un programme de calcul par éléments finis (cf. programme `blambda` section 2.1 et format AMDBA section 3.1).

Création de maillages adaptés anisotropes

Après le premier calcul par éléments finis, il est possible d'adapter le maillage et d'effectuer un nouveau calcul, par exemple pour améliorer la précision des résultats. Dans ce cas, les résultats sont habituellement analysés par un programme appelé *estimateur*. Ce dernier crée un fichier, par exemple une carte de tailles ou de métriques, qui gouverne le mailleur dans une nouvelle itération. Ici, nous avons simulé cette étape à l'aide du programme `blestim`

(cf. section 2.4). Ce programme contient plusieurs cas tests prédéfinis, dont le cas numéro 5 qui est défini de la façon suivante :

```
case(5)
  theta = atan2(y-1, x-1)
  h1 = 0.4*abs((x-1)**2 + (y-1)**2 - (0.75)**2) + 0.003
  h2 = 0.1
```

Les trois variables `theta`, `h1` et `h2` définissent une métrique en tout point (x, y) . L'ensemble des points situés à une distance 1 du point (x, y) , dans cette métrique, est une ellipse d'inclinaison `theta` et de tailles `h1` et `h2` selon les deux directions principales (cf. section 3.5). Les expressions de `theta`, `h1` et `h2` sont telles que, sur le cercle de centre $(1,1)$ et de rayon 0.75, on obtient des triangles étirés tangents à ce cercle, dans un rapport de $0.003/0.1 = 3\%$.

Le programme `blestim` est activé de la façon suivante :

```
% blestim
  Prefixe ?
x
  Iteration ?
0
  Numero du test ?
5
```

Nous pouvons à présent commencer les itérations de la boucle d'adaptation. Il existe plusieurs façons de mettre en œuvre cette boucle :

- Une première possibilité est d'activer manuellement, une nouvelle fois, les programmes `blmc`, `blprems` et `blms` (étapes 6, 7 et 8 de la section précédente), en remplaçant le numéro d'itération 0 par 1. Il en résulte un premier maillage adapté, et la boucle peut se poursuivre (avec le numéro d'itération 2, 3, ...) tant que le maillage n'est pas jugé satisfaisant.
- Une deuxième possibilité, plus commode, est d'utiliser un *shell-script* comportant des structures de contrôle. Par exemple, une boucle d'adaptation avec trois itérations est réalisée en créant le fichier `boucle.sh` suivant :

```
#!/bin/sh
for i in 1 2 3
do (echo x ; echo $i) | blmc
    (echo x ; echo $i) | blprems
    (echo x ; echo $i) | blms
    (echo x ; echo $i ; echo 5) | blestim
done
```

Après avoir créé le fichier une première fois, il est généralement nécessaire de le rendre exécutable :

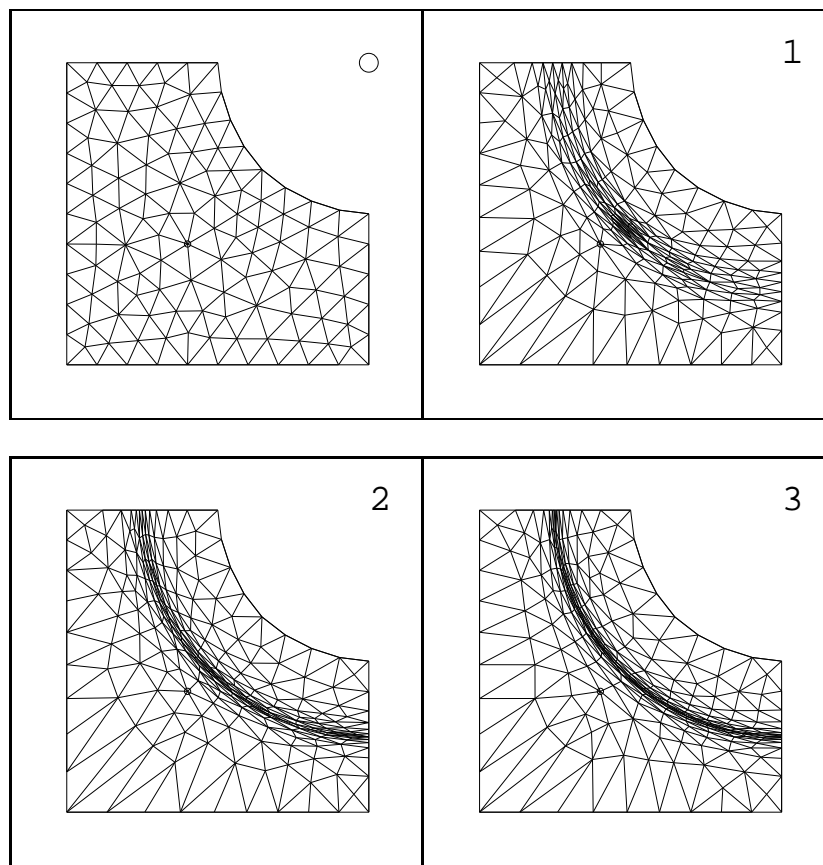
```
% chmod +x boucle.sh
```

La boucle est alors activée en tapant simplement le nom du *shell-script* :

```
% ./boucle.sh
```

- Une troisième possibilité, plus longue à mettre en œuvre, serait d'utiliser un environnement parallèle de type PVM ou MPI (dans le cas de plusieurs sous-domaines).

Quelle que soit la possibilité choisie, on obtient trois nouveaux maillages (FIG. 1.2, $i = 1$ à 3). Les maillages sont de plus en plus conformes à la métrique souhaitée. Remarquons que tous les maillages respectent exactement le point imposé de coordonnées (0.4, 0.4). Remarquons aussi, dans les coins du domaine, que le mailleur subdivise automatiquement les arêtes internes dont les deux extrémités sont sur la frontière (procédure propre aux éléments finis).



o : emplacement du point imposé.

FIG. 1.2 - *Maillage initial ($i = 0$) et maillages adaptés ($i = 1$ à 3).*

1.3 Gestion des fichiers

Au cours des étapes précédentes, les différents programmes du logiciel BL2D se transmettent des données par l'intermédiaire de fichiers. La gestion de ces fichiers est pratiquement transparente pour l'utilisateur grâce à une convention simple sur leurs noms, qui sont toujours de la forme :

préfixe.itération.suffixe

- *préfixe* est un nom choisi par l'utilisateur.
- *itération* est le compteur de la boucle d'adaptation, qui vaut 0 lors de la création du maillage initial (non adapté) et est incrémenté de 1 à chaque itération.
- *suffixe* indique le format du fichier (cf. chapitre 3).

Chapitre 2

Programmes

La description détaillée des programmes que les utilisateurs peuvent activer est donnée ci-dessous par ordre alphabétique. Pour chaque programme, les rubriques suivantes peuvent être mentionnées :

- Données (lues sur l'entrée standard qui est par défaut le terminal).
- Fichiers d'entrée.
- Fichiers de sortie.

La plupart des programmes demandent seulement à l'utilisateur un *préfixe* x et un numéro d'*itération* i (cf. section 1.3). En outre, le nom de chaque fichier se termine par un *suffixe* qui indique son format (cf. chapitre 3).

2.1 Programme `blambda`

Données : *préfixe* x *itération* i
Fichiers d'entrée : `x.0.g` `x.i.c` `x.i.ms`
Fichiers de sortie : `x.i.amdba`

Le programme `blambda` lit trois fichiers : `x.0.g` (données géométriques et physiques), `x.i.c` (maillage de courbes) et `x.i.ms` (maillage de domaines). Il extrait une partie de ces données pour créer un fichier au format AMDBA, qui est utilisé par plusieurs codes de calcul par éléments finis (cf. section 3.1).

2.2 Programme `bldraw`

Données : dirigées par des menus
Fichiers d'entrée : selon le tracé à effectuer
Fichiers de sortie : éventuellement PostScript

Le programme `bldraw` réalise des tracés à partir de différents fichiers. Il utilise la bibliothèque graphique Fortran 3D, qui permet un choix dynamique du terminal de sortie graphique [6, partie 3]. L'utilisateur est constamment guidé par des menus. Le menu principal est le suivant :

```
Fichiers : g | hg | smooth | c (mc) | hc | s (ms) | hs | h
Autres : b base | t test | q quitter bldraw
```

Le menu `Fichiers` permet de choisir un ou plusieurs fichiers à tracer (cf. formats définis au chapitre 3):

- `g` : données géométriques et physiques initiales (fichier G),
- `hg` : carte H initiale (fichier HG),
- `smooth` : splines créées par le programme `blsmooth`,
- `c` ou `mc` : maillage de courbes créé par le programme `blmc`,
- `hc` : carte H créée par le programme `blmc`,
- `s` ou `ms` : maillage de domaines créé par le programme `blms`,
- `hs` : carte H créée par le programme `blms`,
- `h` : carte H créée par le programme `blestim` ou par un *estimateur*.

Le menu `Autres` permet d'accéder à d'autres fonctionnalités :

- `b` : fonctions de base (effacer l'écran, changer la hauteur des caractères ou l'épaisseur des traits, afficher un texte, faire un zoom, ...),
- `t` : tests divers,
- `q` : quitter le programme `bldraw`.

Par exemple, en tapant `g` et en répondant aux menus qui apparaissent successivement, on obtient un tracé à partir d'un fichier G (FIG. 1.1 en haut).

2.3 Programme `blemc2`

Données : *préfixe* x
 Fichiers d'entrée : x.emc2.bd
 Fichiers de sortie : x.0.g x.0.hg

Le programme `blemc2` permet de créer les fichiers initiaux G et HG (cf. section 1.1). Il utilise l'éditeur de maillages et de contours bidimensionnel EMC² [8]. Un autre moyen de créer ces fichiers initiaux est d'activer le programme `blprepro` (cf. section 2.9).

2.4 Programme `blestim`

Données : *préfixe* x *itération* i *numéro du cas*
 Fichiers d'entrée : x.i.c x.i.ms
 Fichiers de sortie : x.i.h

Le programme `blestim` simule un *estimateur*. Habituellement, un *estimateur* analyse le résultat d'un calcul par éléments finis et en déduit une carte de tailles ou de métriques. Ici, la carte est définie analytiquement par l'utilisateur, sous forme d'instructions Fortran.

Par exemple (cf. section 1.2), le cas numéro 5 du programme `blestim` est reproduit ci-dessous. Sur le cercle de centre (1,1) et de rayon 0.75, il définit des triangles étirés tangents à ce cercle, dans un rapport de $0.003/0.1 = 3\%$.

```
case(5)
  theta = atan2(y-1, x-1)
  h1 = 0.4*abs((x-1)**2 + (y-1)**2 - (0.75)**2) + 0.003
  h2 = 0.1
```

Il est facile d'ajouter de nouveaux cas dans le programme source `blestim.f90` présent dans le répertoire de distribution (cf. section 1.2).

2.5 Programme blinterpol

Données : *préfixe* x *itération* i
 Fichiers d'entrée : x.i.bb x.i+1.is
 Fichiers de sortie : x.i.bbi

Le programme `blinterpol` utilise deux fichiers d'entrée qui contiennent respectivement :

- la solution d'un calcul par éléments finis sur un maillage i ,
- les coordonnées barycentriques des points du maillage $i + 1$ par rapport au maillage i (cf. section 3.6).

Il en déduit une solution interpolée qui permet d'initialiser le calcul suivant (la version fournie réalise une interpolation de type P1 mais peut être modifiée).

2.6 Programme blmc

Données : *préfixe* x *itération* i
 Fichiers d'entrée : si $i = 0$: x.0.g x.0.smooth x.0.hg
 si $i \geq 1$: x.0.g x.0.smooth x.i-1.c x.i-1.h
 Fichiers de sortie : si $i = 0$: x.0.c x.0.hc
 si $i \geq 1$: x.i.c x.i.hc x.i.ic

Le programme `blmc` est un mailleur de courbes : il discrétise chaque segment courbe en arêtes. Les tailles de ces arêtes sont déterminées par le fichier d'entrée `x.0.hg` si $i = 0$, et `x.i-1.h` si $i \geq 1$.

2.7 Programme blms

Données : *préfixe* x *itération* i
 Fichiers d'entrée : si $i = 0$: x.0.mc x.0.hc
 si $i \geq 1$: x.i.mc x.i.hc x.i-1.ms x.i-1.h
 Fichiers de sortie : si $i = 0$: x.0.ms x.0.hs
 si $i \geq 1$: x.i.ms x.i.hs x.i.is

Le programme `blms` est un mailleur de domaines : il discrétise chaque sous-domaine en triangles. Les tailles et les formes de ces triangles sont déterminées par le fichier d'entrée `x.i.hc` créé par le mailleur de courbes.

En option, le fichier supplémentaire `x.0.gov` permet de gouverner plus finement le mailleur de domaines (cf. chapitre 3).

2.8 Programme `blprems`

Données : *préfixe* `x` *itération* `i`
Fichiers d'entrée : `x.0.g` `x.0.smooth` `x.i.c`
Fichiers de sortie : `x.i.mc`

Le programme `blprems` crée les données du mailleur de domaines `blms`. Ces données sont au format MC (cf. section 3.7).

2.9 Programme `blprepro`

Données : *préfixe* `x` *itération* `i=0`
Fichiers d'entrée : `x.0.p`
Fichiers de sortie : `x.0.g` `x.0.hg`

Le programme `blprepro` permet de créer les fichiers initiaux `G` et `HG` (cf. section 1.1). Il utilise un fichier de données créé par l'utilisateur (cf. section 3.9). Un autre moyen de créer ces fichiers initiaux est d'activer le programme `blmc2` (cf. section 2.3).

2.10 Programme `blsmooth`

Données : *préfixe* `x` *itération* `i=0`
Fichiers d'entrée : `x.0.g`
Fichiers de sortie : `x.0.smooth`

Le programme `blsmooth` crée un fichier `SMOOTH` (cf. section 3.10).

2.11 Programme blverif

Données : *préfixe* x *itération* i=0
Fichiers d'entrée : x.0.g x.0.hg
Fichiers de sortie : *aucun*

Le programme `blverif` vérifie les fichiers initiaux G et HG : les coordonnées des points doivent être toutes différentes, les numéros des points imposés et des extrémités doivent être tous différents, les tailles ou les métriques souhaitées doivent être fixées aux points imposés et aux extrémités, ...

Chapitre 3

Formats

Cahier des charges

Un code de calcul scientifique est généralement formé de plusieurs composants logiciels qui communiquent entre eux via des fichiers. Ces fichiers doivent respecter une certaine organisation des données ou “format”. L’analyse de l’existant nous a conduits à spécifier une nouvelle organisation des informations [7]. Plus précisément, les possibilités requises sont recensées ci-dessous :

- Définition de structures simples, tant lors de leur création que de leur manipulation.
- Présence de toutes les informations utiles dans les problèmes, quelle que soit la nature de ceux-ci.
- Unicité (et non duplication) des informations.
- Évolutivité de l’organisation des informations.
- Efficacité en temps et en place mémoire.
- Espace 2D ou 3D.
- P2 droit ou courbe (position des nœuds autres que sommets).
- Possibilité de traiter tous les types d’éléments connus, y compris ceux qui font l’objet d’études récentes (élément *mortier*) et ménager la possibilité de traiter des types inconnus à ce jour.
- Fissures.
- Tangentes - normales.
- Adaptation, nécessitant en particulier le remaillage des contours.
- Indépendance vis à vis des systèmes de CAO (divers et mouvants).

- Possibilité de traiter simplement le cas d'un seul type d'élément.
- Structures particulières (grille par exemple).
- Recollement de maillages non compatibles.
- Maillages à frontières libres.
- Problèmes multi-physiques.

Présentation des formats

Dans le cadre du mailleur bidimensionnel BL2D, nous avons défini les formats ci-après, classifiés en trois catégories :

1. Les formats d'entrée du logiciel BL2D :
 - G : données géométriques et physiques,
 - H : carte de tailles ou de métriques.
2. Les formats de sortie du logiciel BL2D (en fait, dans le cas d'une boucle d'adaptation de maillage, ces formats de sortie deviennent des formats d'entrée à l'itération suivante) :
 - C : maillage de courbes,
 - MS : maillage de domaines.
3. Les formats auxiliaires. Il s'agit de formats d'interface avec d'autres logiciels (AMDBA et P), de formats internes au logiciel BL2D (IS, MC, SMOOTH) ou de données que l'utilisateur peut fournir en option (GOV) :
 - AMDBA : maillage de domaines et physique,
 - GOV : données optionnelles pour gouverner le mailleur de domaines,
 - IS : coordonnées barycentriques, en vue d'une interpolation,
 - MC : maillage de courbes utilisé en entrée du mailleur de domaines,
 - P : données du préprocesseur `blprepro`,
 - SMOOTH : représentation des splines.

L'organisation en plusieurs formats que nous avons définie tente de répondre le mieux possible au cahier des charges précédent. Notamment, la distinction entre les données géométriques et physiques, les maillages de courbes et les maillages de domaines évite la duplication d'informations dans des fichiers séparés. Ceci entraîne des gains en simplicité, en espace disque et en temps d'entrée/sortie.

Remarques communes à tous les formats

1. Ces formats définissent, au choix, des fichiers de texte (*ascii*) ou des fichiers binaires. Les fichiers de texte facilitent la communication entre des programmes écrits dans des langages différents ou compilés sur des machines différentes. Les fichiers binaires évitent les temps de codage et de décodage.
2. Tout nombre flottant doit pouvoir être lu dans une variable en double précision dans le langage Fortran 90 ou C. Le nombre de chiffres significatifs dépend de la machine et des programmes utilisés.
 - Exemple de lecture en Fortran 90 :

```
double precision d
read (file,*) d
```
 - Exemple de lecture en C :

```
double d;
scanf(file, "%lf", d);
```
3. Les formats commencent généralement par une ligne de la forme :
$$format \quad v1$$
 - *format* est le nom du format, ce qui permet une vérification à la lecture.
 - *v1* est le numéro de version, ce qui devrait faciliter les évolutions futures du logiciel.
4. Pour des raisons de commodité, ces formats sont présentés ci-dessous par ordre alphabétique.

3.1 Format auxiliaire AMDBA

Objectif

L'objectif du format AMDBA est de décrire un maillage de domaines et ses références physiques. Il est utilisé par plusieurs codes de calcul par éléments finis. Cependant, il est moins général que les formats G, C et MS réunis. Notamment, il ne donne pas les références physiques des arêtes et ne permet pas de calculer facilement les coordonnées des nœuds P2.

Ces données sont générées par le programme `blambda`.

Définition du format AMDBA

NP NT

Pour tous les points $i = 1..NP$

i x_i y_i φ_i

Pour tous les triangles $i = 1..NT$

i p_{i1} p_{i2} p_{i3} φ_i

Notations

NP Nombre de points.

NT Nombre de triangles.

Pour un point $i = 1..NP$:

x_i y_i Coordonnées du point.

φ_i Référence physique du point.

Pour un triangle $i = 1..NT$:

p_{i1} p_{i2} p_{i3} Numéros des trois sommets donnés dans le sens direct.

φ_i Référence physique du triangle.

Exemple de fichier au format AMDBA

Le fichier AMDBA ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
121  202
1  0.4000000  0.4000000  10
2  0.0000000  0.0000000  20
3  1.0000000  0.0000000  30
...
1  69  70  98  200
2  96  92  93  200
3  35  58  57  200
...
```

3.2 Format de sortie C

Objectif

Rappelons que, dans le format G, la description d'un domaine comporte des splines (droites ou courbes). Chaque spline peut ensuite être maillée, c'est-à-dire subdivisée en arêtes. L'objectif du format C est de décrire cet ensemble d'arêtes.

Ces données sont générées par le programme `blmc`.

Définitions préalables

Pour un maillage de courbes donné, de nombreux types d'interpolations sont possibles. Si le maillage est de type P1, les nœuds sont confondus avec les extrémités des arêtes. Si le maillage est de type P2, chaque arête possède un nœud supplémentaire entre ses extrémités. Dans notre exemple (FIG. 3.1), le maillage contient 3 arêtes (a_1, a_2, a_3), 4 extrémités (n_1, n_3, n_5, n_7) et 7 nœuds (n_1, n_2, \dots, n_7). Afin d'autoriser tout type d'interpolation, le format C ne contient que des nœuds P1 mais associe à chaque nœud son abscisse curviligne s sur la spline considérée ($0 \leq s \leq$ longueur de la spline). Il est donc possible de calculer ensuite les coordonnées d'autres nœuds.

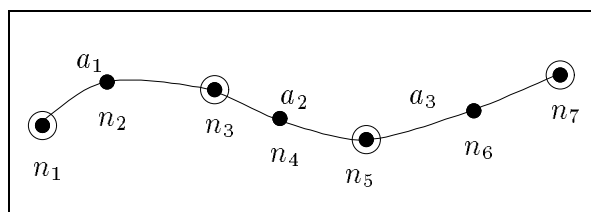


FIG. 3.1 - Maillage P2 d'une courbe.

Définition du format C

C v1

NP NPI NPE NPC NS

Pour tous les points $i = 1..NP$

$x_i y_i$

Pour toutes les splines $i = 1..NS$

$np_i (p_{ij} s_{ij})_{j=1..np_i}$

Notations

NP Nombre total de points: $NP = NPI + NPE + NPC$.

NPI Nombre de points imposés.

NPE Nombre d'extrémités.

NPC Nombre de points restants (ni imposés ni extrémités).

NS Nombre de splines.

Pour un point $i = 1..NP$:

$x_i y_i$ Coordonnées du point.

Pour une spline $i = 1..NS$:

np_i Nombre de points sur la spline, extrémités comprises ($np_i \geq 2$).

p_{ij} Numéros des points sur la spline ($1 \leq p_{ij} \leq NP$), donnés dans l'ordre suivant :

$p_{i,1}$ Numéro de l'extrémité 1.

$p_{i,2..np_i-1}$ Numéros des points internes à la spline, en allant de l'extrémité 1 vers l'extrémité 2.

p_{i,np_i} Numéro de l'extrémité 2.

s_{ij} Abscisse curviligne du point p_{ij} . Pour j variant de 1 à np_i , s croît de 0 à la longueur de la spline.

Remarques

Les points sont numérotés de 1 à NP en considérant successivement :

- les NPI points imposés définis dans G,
- les NPE extrémités définies dans G,
- les NPC points créés par le mailleur de courbes.

Les références physiques des points et des arêtes, qui sont déjà décrites dans le format G, ne sont pas dupliquées dans le format C. Elles peuvent être obtenues par des fonctions prévues à cet effet (cf. section 5.2).

Exemple de fichier au format C

Le fichier C ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```

C v1
39 1 5 33 5
0.4000000000000000 0.4000000000000000
0.0000000000000000 0.0000000000000000
1.0000000000000000 0.0000000000000000
...
0.9032169485291625 0.5120700124670046
0.8102791308792039 0.5419733047033631
0.7234173490701071 0.5865454450721955
0.6464466094067263 0.6464466094067263
0.5865454450721955 0.7234173490701072
0.5419733047033631 0.8102791308792039
0.5120700124670046 0.9032169485291625
...
11
2 0.0 7 0.1 8 0.2 9 0.3 10 0.4 11 0.5
12 0.6 13 0.7 14 0.8 15 0.9 3 1.0
6
3 0.0 16 0.1 17 0.2 18 0.3 19 0.4 5 0.5
9
5 0.0000000000000000
20 9.7739365406961959E-02
21 0.1954787308139239
22 0.2932180962208858
23 0.3909574616278478
24 0.4886968270348097
25 0.5864361924417716
26 0.6841755578487336
6 0.7819149232556956
6
6 0.0 27 0.1 28 0.2 29 0.3 30 0.4 4 0.5
11
4 0.0 31 0.1 32 0.2 33 0.3 34 0.4 35 0.5
36 0.6 37 0.7 38 0.8 39 0.9 2 1.0

```

3.3 Format d'entrée G

Objectif

L'objectif du format G est de décrire les données géométriques et physiques d'un domaine du plan. Ces données sont normalement générées par un système interactif graphique ou par un préprocesseur (cf. chapitre 1).

Définitions préalables

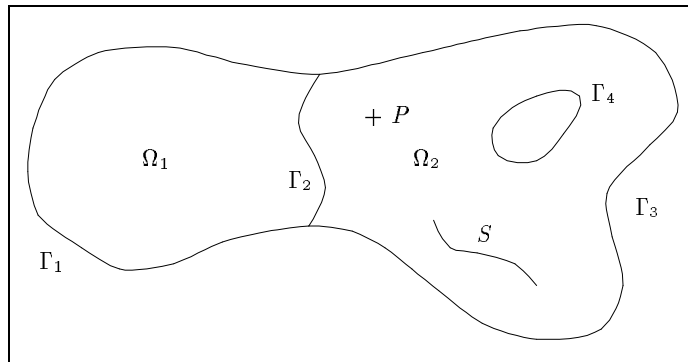
Rappels

Un domaine peut être composé d'un ou de plusieurs sous-domaine(s). Chaque sous-domaine est délimité par sa frontière.

Pour effectuer des calculs par éléments finis, il est nécessaire de mailler le domaine. Plusieurs maillages peuvent être réalisés sur un même domaine (par exemple pour adapter le maillage, ou encore pour résoudre un problème multi-physique). Tout maillage doit respecter les frontières des sous-domaines. On peut également imposer que le maillage s'appuie sur des segments ou des points internes à un sous-domaine (on parle alors de segment imposé ou de point imposé). Un segment interne a au plus une extrémité commune avec un segment de frontière (s'il avait deux extrémités communes, il serait lui-même un segment de frontière).

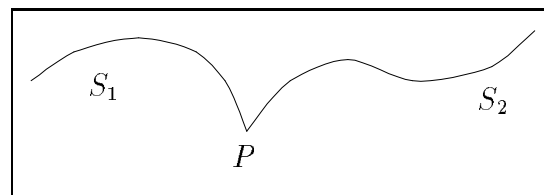
Par exemple (FIG. 3.2), le domaine Ω comprend deux sous-domaines Ω_1 et Ω_2 . Le sous-domaine Ω_1 a pour frontière la composante connexe $\Gamma_1 \cup \Gamma_2$. Le sous-domaine Ω_2 a pour frontière les deux composantes connexes $\Gamma_2 \cup \Gamma_3$ et Γ_4 (il comporte un "trou"). Il contient le segment imposé S et le point imposé P .

Des propriétés physiques peuvent être associées aux entités que nous venons de rappeler (sous-domaines, frontières, segments imposés et points imposés). Par exemple, dans le cas de la simulation d'un problème thermique, on peut associer à un sous-domaine sa conductivité et sa source de chaleur, à une frontière son coefficient de transfert et sa température extérieure, et à un point imposé sa température.

FIG. 3.2 - Exemple de domaine $\Omega = \Omega_1 \cup \Omega_2$.

Les splines

Il est commode de représenter chaque segment de frontière ou segment imposé par une ou plusieurs spline(s). Une spline seule ne contient pas de point anguleux. En revanche, le point de jonction entre deux splines consécutives peut être anguleux (FIG. 3.3).

FIG. 3.3 - Segment courbe formé de deux splines S_1 et S_2 .

Parmi les nombreuses méthodes utilisées pour définir des splines, nous utilisons celle de Catmull-Rom, où chaque spline est une courbe cubique par morceaux [4].

Par exemple (FIG. 3.4), la spline $P_2P_3P_4P_5$ est définie par le polygone de contrôle $P_1P_2P_3P_4P_5P_6$. Le morceau de spline P_2P_3 est un morceau cubique

défini par ses extrémités P_2 et P_3 , par la tangente en P_2 égale à $\overrightarrow{P_1P_3}/2$, et par la tangente en P_3 égale à $\overrightarrow{P_2P_4}/2$. Les morceaux cubiques P_3P_4 et P_4P_5 sont définis de manière analogue.

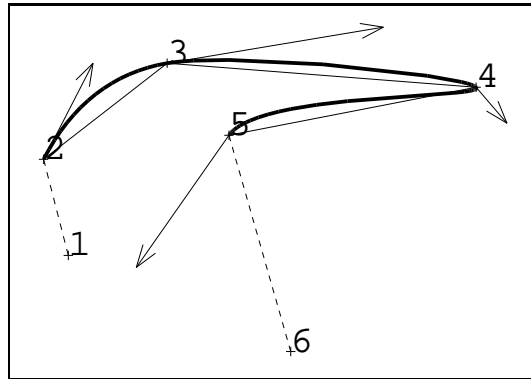


FIG. 3.4 - Une spline et son polygone de contrôle.

De façon plus générale, une spline est décrite par un polygone de contrôle $P_1 \dots P_n$ ($n \geq 4$, P_1 et P_n facultatifs). Parmi les sommets de ce polygone, on peut distinguer :

- Les extrémités de la spline : P_2 et P_{n-1} .
- Les points de contrôle internes à la spline : $P_3 \dots P_{n-2}$. Si $n = 4$, ces points sont absents. Sinon ($n > 4$), la spline passe par tous ces points P_i . En chaque point, la tangente aux morceaux $P_{i-1}P_i$ et P_iP_{i+1} est continue et égale au vecteur $\overrightarrow{P_{i-1}P_{i+1}}/2$.
- Les deux points de contrôle externes à la spline : P_1 et P_n . Si le point P_1 (resp. P_n) est présent, il sert à fixer la tangente à l'extrémité P_2 (resp. P_{n-1}) de la spline, en utilisant la méthode précédente. Si un point est absent, la tangente à l'extrémité associée est considérée comme libre.

Ainsi, chaque morceau de spline $P_2P_3, \dots, P_{n-2}P_{n-1}$ est défini par ses deux extrémités et éventuellement les tangentes en ses deux extrémités. Si

les tangentes sont toutes deux fixées, le morceau est cubique (degré ≤ 3). Si une seule tangente est fixée, le morceau est parabolique (degré ≤ 2). Enfin, si aucune tangente n'est fixée, le morceau est linéaire (degré ≤ 1). Il en résulte que, si seules les deux extrémités de la spline sont définies (c'est-à-dire qu'il n'existe aucun point de contrôle, ni interne ni externe), la spline se réduit à un segment de droite.

Les cercles

Le cercle est une figure géométrique fréquemment rencontrée, par exemple dans la représentation de nombreuses pièces mécaniques. Lorsqu'un cercle est approché par une spline, il faut veiller à ce que le nombre n de sommets du polygone de contrôle soit suffisant pour atteindre la précision souhaitée (dans le formalisme précédent, le polygone de contrôle serait noté $P_n P_1 P_2 P_3 \dots P_n P_1 P_2$, les 3 premiers et les 3 derniers sommets étant confondus). A l'œil nu, le nombre $n = 4$ est insuffisant mais $n = 8$ est déjà acceptable, à condition que les sommets soient équidistants (FIG. 3.5).

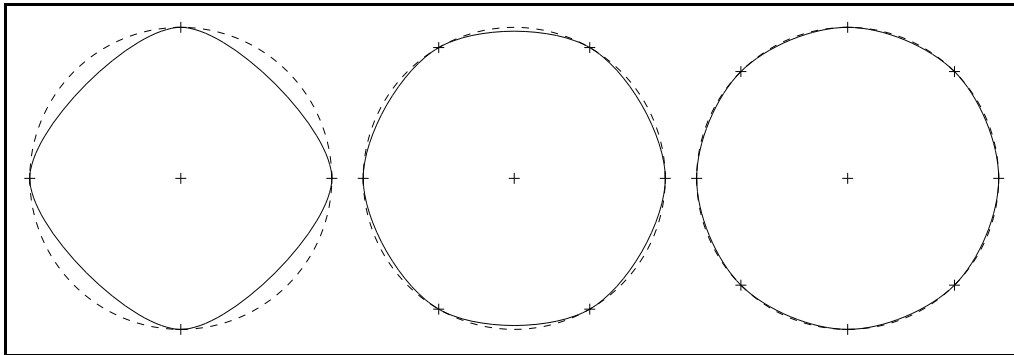


FIG. 3.5 - Cercle défini par $n = 4, 6$ ou 8 points.

Plus précisément, le tableau ci-dessous donne, pour certaines valeurs du nombre n , la distance maximale Δ entre un cercle de rayon 1 et la spline cubique approchante (d'où la courbe FIG. 3.6).

n	Δ	n	Δ	n	Δ	n	Δ
4	116.1e-3	20	226.4e-6	40	14.24e-6	60	2.817e-6
6	25.72e-3	24	109.4e-6	44	9.731e-6	64	2.177e-6
8	8.470e-3	28	59.18e-6	48	6.873e-6	68	1.708e-6
12	1.721e-3	32	34.72e-6	52	4.991e-6	72	1.359e-6
16	0.550e-3	36	21.69e-6	56	3.712e-6	76	1.095e-6

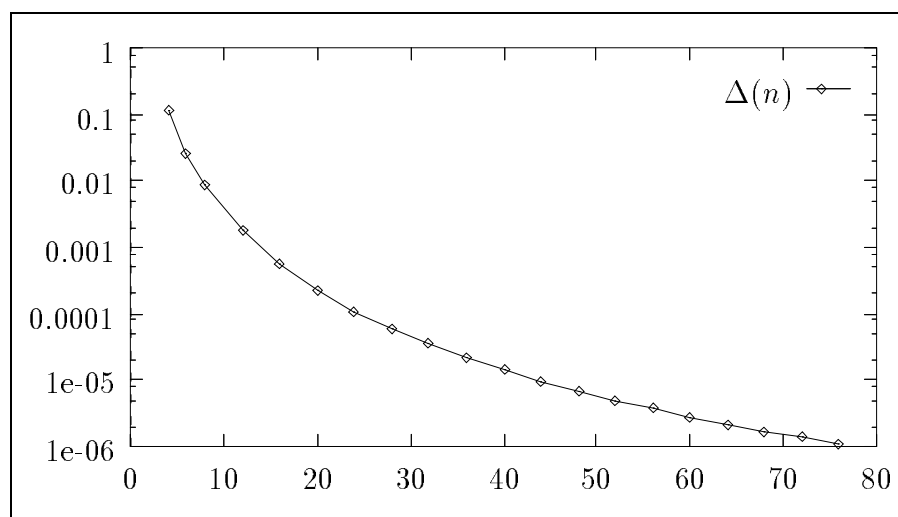


FIG. 3.6 - Approximation d'un cercle par une spline : courbe montrant l'erreur Δ (échelle logarithmique) en fonction du nombre de points de contrôle n .

Définition du format G

G v1

NP NPI NPE NPG NS ND

Pour tous les points $i = 1..NP$

x_i y_i

Pour tous les points imposés $i = 1..NPI$

p_i φ_i

Pour toutes les extrémités $i = 1..NPE$

p_i φ_i

Pour toutes les splines $i = 1..NS$

np_i $(p_{ij})_{j=1..np_i}$ φ_i

Pour tous les sous-domaines $i = 1..ND$

s_i o_i φ_i

Notations

NP Nombre total de points: $NP = NPI + NPE + NPG$.

NPI Nombre de points imposés.

NPE Nombre d'extrémités.

NPG Nombre de points restants (ni imposés ni extrémités).

NS Nombre de splines.

ND Nombre de sous-domaines.

Pour un point $i = 1..NP$:

x_i y_i Coordonnées du point.

Pour un point imposé, $i = 1..NPI$:

p_i Numéro du point.

φ_i Référence physique du point.

Pour une extrémité, $i = 1..NPE$:

p_i Numéro du point.

φ_i Référence physique du point.

Pour une spline $i = 1..NS$:

np_i Nombre de points servant à définir la spline ($np_i \geq 4$).

p_{ij} Numéros des points servant à définir la spline, donnés dans l'ordre suivant:

$p_{i,1}$	Numéro du point de contrôle externe 1. Si ce point est absent, $p_{i,1} = 0$.
$p_{i,2}$	Numéro de l'extrémité 1.
$p_{i,3..np_{i-2}}$	Numéros des points de contrôle internes à la spline, en allant de l'extrémité 1 vers l'extrémité 2.
$p_{i,np_{i-1}}$	Numéro de l'extrémité 2.
p_{i,np_i}	Numéro du point de contrôle externe 2. Si ce point est absent, $p_{i,np_i} = 0$.
φ_i	Référence physique de la spline.
Pour un sous-domaine $i = 1..ND$:	
s_i	Numéro d'une spline appartenant à la frontière.
o_i	Orientation telle que le sous-domaine soit à gauche de la spline s_i . Si $o_i = 1$, la spline est parcourue de l'extrémité 1 vers l'extrémité 2. Si $o_i = -1$, la spline est parcourue dans l'autre sens.
φ_i	Référence physique du sous-domaine.

Remarques

La première boucle $i = 1..NP$ définit les coordonnées de tous les points utilisés dans le format G. Ces coordonnées doivent être toutes différentes.

Par définition, un point est dit :

- imposé s'il apparaît dans la boucle $i = 1..NPI$. Les numéros p_i des points doivent être tous différents.
- extrémité s'il apparaît dans la boucle $i = 1..NPE$. Les numéros p_i des points doivent aussi être tous différents. En outre, cette boucle doit contenir tous les points qui apparaissent en tant qu'extrémités dans la description des splines (points $p_{i,2}$ et $p_{i,np_{i-1}}$).
- point de contrôle externe s'il apparaît en tant que tel dans la description des splines (points $p_{i,1}$ et p_{i,np_i}).
- point de contrôle interne s'il apparaît en tant que tel dans la description des splines (points $p_{i,3..np_{i-2}}$).
- avec métrique si le fichier HG définit en ce point une taille ou une métrique fixée ($h_i \neq 0$ ou $a_i \neq 0$, cf. section 3.5).

Les différentes combinaisons possibles sont précisées dans le tableau symétrique ci-dessous (avec l'abréviation p.c. = point de contrôle) :

—	imposé	extrémité	p.c. externe	p.c. interne	avec métrique
imposé	—	interdit	possible	possible	obligatoire
extrémité		—	possible	interdit	obligatoire
p.c. externe			—	possible	inutile
p.c. interne				—	possible
avec métrique					—

La combinaison “imposé - extrémité” est interdite afin de numérotter de façon systématique les points des maillages (cf. sections 3.2 et 3.8).

La combinaison “imposé - p.c. externe” est utile dans la définition de certaines splines.

La combinaison “imposé - p.c. interne” permet de d'imposer un point au mailleur de courbes sans diviser artificiellement une spline.

La combinaison “extrémité - p.c. externe” est utile dans le cas des courbes presque fermées.

La combinaison “extrémité - p.c. interne” est interdite car deux splines ne peuvent être adjacentes que si elles ont des extrémités communes.

La combinaison “p.c. externe - p.c. interne” est utile dans le cas des courbes fermées.

La combinaison “p.c. interne - avec métrique” permet de contrôler plus finement le mailleur de courbes.

Dans la description d'un point, d'une spline ou d'un sous-domaine, on appelle référence physique φ_i un nombre entier qui est utilisé comme index vers les propriétés physiques de cette entité. Par convention, $\varphi_i = 0$ indique qu'aucune propriété physique ne doit être associée à l'entité.

Exemple de fichier au format G

Le fichier G ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
G v1
9 1 5 3 5 1
0.0000000000000000 0.0000000000000000
1.0000000000000000 0.0000000000000000
0.0000000000000000 1.0000000000000000
0.4000000000000000 0.4000000000000000
1.3535533905932737 0.6464466094067263
1.0000000000000000 0.5000000000000000
0.6464466094067263 0.6464466094067263
0.5000000000000000 1.0000000000000000
0.6464466094067263 1.3535533905932737

4 10

1 20
2 30
3 40
6 50
8 60

4
0 1 2 0 110
4
0 2 6 0 120
5
5 6 7 8 9 130
4
0 8 3 0 140
4
0 3 1 0 150

1 1 200
```

3.4 Format auxiliaire GOV

Objectif

L'objectif du format GOV est de gouverner le mailleur de domaines lors de la création du maillage initial (itération $i = 0$). Pour cela, il fixe en certains points la taille souhaitée des éléments à créer au voisinage du point (cas isotrope) ou la métrique au voisinage du point (cas anisotrope). Les points ne sont pas imposés, ce qui permet par exemple de définir manuellement de grandes tailles d'éléments sur une grille fine.

Définition du format GOV

NP

isotrope ou anisotrope

pour chaque point $i = 1..NP$

si isotrope: $x_i \ y_i \ h_i$

si anisotrope: $x_i \ y_i \ a_i \ b_i \ c_i$

Notations

NP Nombre de points où sont définies les tailles ou les métriques.

Pour un point $i = 1..NP$:

$x_i \ y_i$ Coordonnées du point.

h_i Taille au point i (cas isotrope).

$a_i \ b_i \ c_i$ Métrique au point i (cas anisotrope).

3.5 Format d'entrée H

Objectif

Un mailleur doit créer des éléments de courbes ou de domaines. L'objectif du format H est de gouverner ce mailleur en donnant, en certains points d'un domaine, la taille souhaitée des éléments à créer au voisinage du point (cas isotrope) ou la métrique au voisinage du point (cas anisotrope). Les points du domaine sont eux-mêmes définis dans le format G (données géométriques et physiques), C (maillage de courbes) ou MS (maillage de domaines).

Ces données sont normalement générées par un préprocesseur (pour contrôler le maillage initial) ou par un *estimateur* (pour adapter le maillage).

Définitions préalables

La carte de tailles ou de métriques est définie sur un ensemble de points, qui sont eux-mêmes donnés dans le format G, C ou MS.

Si le maillage à créer est isotrope, il suffit de donner en chaque point la taille souhaitée des éléments à générer au voisinage de ce point.

Si le maillage à créer est anisotrope, la méthode retenue consiste à fournir en chaque point une *métrique* dans laquelle la taille souhaitée est égale à l'unité [1, 9]. Une métrique est représentée par une matrice symétrique définie positive à trois coefficients (a, b, c) . En prenant pour origine le point où la métrique est donnée, tout point (x, y) situé à une distance 1 de l'origine, dans la métrique (a, b, c) , vérifie l'équation :

$$\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 1 \quad \iff \quad a x^2 + 2 b x y + c y^2 = 1$$

Il s'agit de l'équation d'une ellipse centrée à l'origine. Par une rotation d'angle θ , de manière à rendre le repère parallèle à l'un des deux axes de l'ellipse (FIG. 3.7), l'équation se met sous la forme simplifiée :

$$\frac{X^2}{h_1^2} + \frac{Y^2}{h_2^2} = 1$$

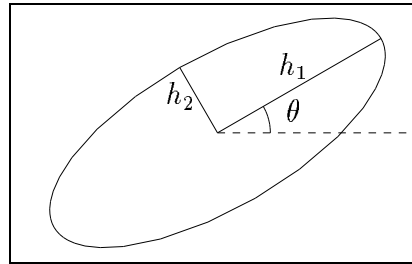


FIG. 3.7 - Ellipse définie par θ , h_1 et h_2 .

h_1 et h_2 représentent les tailles souhaitées selon deux directions orthogonales, dans la métrique usuelle égale à l'identité. Inversement, si l'on connaît θ , h_1 et h_2 , il est facile d'obtenir la métrique (a, b, c) grâce à la relation :

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} = P \begin{pmatrix} \frac{1}{h_1^2} & 0 \\ 0 & \frac{1}{h_2^2} \end{pmatrix} P^{-1}$$

où P est la "matrice de passage" :

$$P = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad \text{et} \quad P^{-1} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$$

Définition du format H

H v1

NP

isotrope ou anisotrope

pour chaque point $i = 1..NP$

si isotrope: h_i

si anisotrope: $a_i \quad b_i \quad c_i$

Notations

NP Nombre de points où sont définies les tailles ou les métriques.

Pour un point $i = 1..NP$:

h_i Taille au point i (cas isotrope).

a_i b_i c_i Métrique au point i (cas anisotrope).

Remarques

Il est parfois utile de laisser libre la taille ou la métrique en certains points. Par convention, la taille ou la métrique est considérée comme libre en tout point p_i tel que $h_i = 0$ (cas isotrope) ou $a_i = 0$ (cas anisotrope).

Exemple de fichier au format H

Le fichier H ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
H v1
9
isotrope
0.100000000000000000
0.100000000000000000
0.100000000000000000
0.100000000000000000
0.000000000000000000
0.100000000000000000
0.100000000000000000
0.100000000000000000
0.000000000000000000
```

3.6 Format auxiliaire IS

Objectif

Au cours d'une boucle d'adaptation, plusieurs maillages d'un même domaine sont générés. L'objectif du format IS est de donner les coordonnées barycentriques des points d'un "nouveau maillage" (*foreground mesh*) par rapport à ceux d'un "ancien maillage" (*background mesh*). Ces données sont créées en vue de l'interpolation de la solution d'un calcul par éléments finis.

Définition du format IS

IS v1

NP

Pour tous les points du nouveau maillage $i = 1..NP$

p_{i1} p_{i2} p_{i3} λ_{i1} λ_{i2}

Notations

NP Nombre de points du nouveau maillage.

Pour un point du nouveau maillage $i = 1..NP$:

p_{i1} p_{i2} p_{i3} Numéros de trois points de l'ancien maillage, ou 0 (cf. remarques).

λ_{i1} λ_{i2} Coordonnées barycentriques (cf. remarques).

Remarques

Soit P_i le point numéro i (dans le nouveau maillage).

En général, (p_{i1}, p_{i2}, p_{i3}) sont les numéros des sommets (P_{i1}, P_{i2}, P_{i3}) du triangle de l'ancien maillage qui englobe le point P_i . Alors P_i est défini vectoriellement par :

$$P_i = \lambda_{i1} P_{i1} + \lambda_{i2} P_{i2} + (1 - \lambda_{i1} - \lambda_{i2}) P_{i3}$$

Cependant, il est possible qu'il n'existe pas de triangle englobant (par exemple si une frontière courbe est remaillée, ou si un segment courbe est défini extérieurement au domaine). Dans ce cas, seuls les deux points englobants dans le maillage de courbes sont considérés. Par convention, le numéro

du troisième point est nul. Les coordonnées barycentriques sont alors calculées en fonction des abscisses curvilignes. Par exemple, si $p_{i3} = 0$, on aura :

$$\lambda_{i1} = \lambda \quad \lambda_{i2} = 1 - \lambda \quad (1 - \lambda_{i1} - \lambda_{i2}) = 0$$

Exemple de fichier au format IS

Le fichier IS ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
IS v1
204
49   1   65   0.000000   1.000000
  2  39   0   1.000000   0.000000
  3  16   0   1.000000   0.000000
  4  31   0   1.000000   0.000000
  5  20 101   1.000000   0.000000
  6 105  26   1.000000   0.000000
  7   8   0   0.672134   0.327866
  8   9  61   0.422772   0.577228
  9  10   0   0.242179   0.757821
...
30   4   71   0.051587   0.172009
  3  16  48   0.087147   0.073293
  5 101  19   0.457181   0.475098
```

3.7 Format auxiliaire MC

Objectif

L'objectif du format MC est de décrire toutes les données géométriques nécessaires au mailleur de domaines `blms`. Il décrit donc essentiellement un maillage de courbes. Ces données sont générées par le programme `blprems` à partir des formats G, SMOOTH et C.

Définition du format C

NP NA ND
 bb_1 bb_2 bb_3 bb_4
 Pour tous les points $i = 1..NP$
 x_i y_i
 Pour toutes les arêtes $i = 1..NA$
 p_{i1} p_{i2}
 Pour tous les sous-domaines $i = 1..ND$
 q_{i1} q_{i2}

Notations

NP Nombre de points.
 NA Nombre d'arêtes.
 ND Nombre de sous-domaines.
 $bb_1..bb_4$ Bounding box :
 (bb_1, bb_2) sont les coordonnées du coin inférieur gauche,
 (bb_3, bb_4) sont les coordonnées du coin supérieur droit.
 Pour un point $i = 1..NP$:
 x_i y_i Coordonnées du point.
 Pour une arête $i = 1..NA$:
 p_{i1} p_{i2} Numéros des deux extrémités de l'arête.
 Pour un sous-domaine $i = 1..ND$:
 q_{i1} q_{i2} Numéros des deux extrémités d'une arête frontière ou interne
 au sous-domaine. L'ordre de ces extrémités est tel que le sous-
 domaine soit à gauche en allant de q_{i1} vers q_{i2} .

Exemple de fichier au format MC

Le fichier MC ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
39 38 1
0.0000000000000000 0.0000000000000000
1.0000000000000000 1.0000000000000000

0.4000000000000000 0.4000000000000000
0.0000000000000000 0.0000000000000000
1.0000000000000000 0.0000000000000000
...
2 7
7 8
8 9
...
2 7
```

3.8 Format de sortie MS

Objectif

Dans un maillage de domaines, chaque sous-domaine est subdivisé ici en triangles. L'objectif du format MS est de décrire cet ensemble de triangles.

Ces données sont générées par le programme `blms`.

Définition du format MS

NP NT
 Pour tous les points $i = 1..NP$
 x_i y_i
 Pour tous les triangles $i = 1..NT$
 p_{i1} p_{i2} p_{i3} v_{i1} v_{i2} v_{i3} d_i

Notations

NP Nombre de points.

NT Nombre de triangles.

Pour un point $i = 1..NP$:

x_i y_i Coordonnées du point.

Pour un triangle $i = 1..NT$:

p_{i1} p_{i2} p_{i3} Numéros des 3 sommets donnés dans le sens direct.

v_{i1} v_{i2} v_{i3} Numéros des 3 triangles voisins.

d_i Numéro du sous-domaine :

$d_i = 0$ pour le sous-domaine compris entre l'objet et la *bounding box*,

$1 \leq d_i \leq ND$ pour les sous-domaines définis dans G,

$d_i \geq ND + 1$ pour les autres sous-domaines ("trous").

Remarques

Les points sont numérotés de 1 à NP en considérant successivement :

- les NPI points imposés définis dans G,
- les NPE extrémités définies dans G,

- les NPC points créés par le mailleur de courbes (cf. section 3.2),
- les points créés par le mailleur de domaines,
- les 4 points de la *bounding box*.

Les références physiques des points, des arêtes et des triangles, qui sont déjà décrites dans le format G, ne sont pas dupliquées dans le format MS. Elles peuvent être obtenues par des fonctions prévues à cet effet (cf. section 5.2).

Exemple de fichier au format MS

Le fichier MS ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```

125 244
 3.99999985098829058e-01  3.99999985098829058e-01
-1.49011709638166786e-08 -1.49011709638166786e-08
 9.99999985098829036e-01 -1.49011709638166786e-08
-1.49011709638166786e-08  9.99999985098829036e-01
 9.99999985098829036e-01  4.99999985098829036e-01
...
 69   70   98   57  128  193   1
 96   92   93  109  235   95   1
125   10    9  160   37   20   0
125   35  124   46    0   42   0
122    5  123   30    0   64   0
...

```

3.9 Format auxiliaire P

Objectif

Rappelons que le programme `blprepro` permet de créer les fichiers initiaux G (données géométriques et physiques) et HG (carte de tailles ou de métriques) (cf. section 1.1). L'objectif du format P est de permettre à l'utilisateur de décrire ces données de façon aussi simple que possible.

Principes généraux

Le format P ressemble aux formats G et H mis bout à bout, mais offre les possibilités suivantes :

- Les objets géométriques (points, splines, sous-domaines, ...) peuvent être désignés par des noms (identificateurs) et non par des numéros.
- Les nombres d'objets sont calculés automatiquement.
- Les lectures sont en format libre [5]. Il est donc possible d'y inclure des commentaires et des expressions arithmétiques.
- Les métriques anisotropes sont données sous la forme θ , h_1 , h_2 et non pas a , b , c . L'angle θ est donné en degrés. La valeur h_1 (resp. h_2) est la taille souhaitée le long de l'axe d'angle θ (resp. $\theta + 90^\circ$).

Définition du format P

```
P v1
NP  NS  ND
Pour tous les points i :
    id_i  x_i  y_i
;
Pour tous les points imposés :
    id_i
;
Pour toutes les splines :
```

```

     $id_i$  de la spline
     $id.$  du point de contrôle 1 (NULL s'il est absent)
     $id.$  de l'extrémité 1
     $id.$  des sommets intermédiaires
     $id.$  de l'extrémité 2
     $id.$  du point de contrôle 2 (NULL s'il est absent)
    ;
;
Pour tous les points imposés ou extrémités t.q.  $\varphi_i \neq 0$ :
     $id_i$   $\varphi_i$ 
;
Pour toutes les splines t.q.  $\varphi_i \neq 0$ :
     $id_i$   $\varphi_i$ 
;
Pour tous les sous-domaines t.q.  $\varphi_i \neq 0$ :
     $id_i$  d'une spline  $o_i$  (+1 ou -1)  $\varphi_i$ 
;
isotrope ou anisotrope
Pour tous les points imposés ou extrémités :
     $id_i$ 
    si isotrope:  $h$  au point  $i$ 
    si anisotrope:  $\theta$   $h_1$   $h_2$  au point  $i$ 
;

```

Notations

NP	Nombre <u>maximal</u> de points.
NS	Nombre <u>maximal</u> de splines.
ND	Nombre <u>maximal</u> de sous-domaines.
id_i	Identificateur d'un objet i (point, spline, sous-domaine, ...).
x_i y_i	Coordonnées du point i .
φ_i	Référence physique d'un point, d'une spline ou d'un sous-domaine.

Exemple de fichier au format P

Le fichier P ci-dessous a été utilisé dans notre exemple (cf. section 1.2).

```
P v1
100 -- nombre maximal de points
100 -- nombre maximal de segments
100 -- nombre maximal de sous-domaines

-- points
!X=SQRT(2)/4
A 0 0      B 1 0      C 0 1      P 0.4 0.4      D 1+X 1-X
E 1 0.5    F 1-X 1-X  G 0.5 1    H 1-X 1+X      ;

-- points imposes
P ;

-- segments
AB NULL A B NULL ;
BE NULL B E NULL ;
EG D E F G H ;
GC NULL G C NULL ;
CA NULL C A NULL ;
;

-- references physiques de certains points imposes ou extremités
P 10  A 20  B 30  C 40  E 50  G 60  ;

-- references physiques de certains segments
AB 110  BE 120  EG 130  GC 140  CA 150  ;

-- sous-domaines
AB +1 200 ;

-- carte de h ou de metriques
isotrope
A 0.1  B 0.1  C 0.1  P 0.1  E 0.1  F 0.1  G 0.1  ;
```


3.10 Format auxiliaire SMOOTH

Objectif

L'objectif du format SMOOTH est de représenter des splines, de manière à calculer rapidement les coordonnées des points appartenant à ces splines.

Définition du format SMOOTH

Pour représenter une spline, il existe essentiellement deux méthodes :

- réaliser un maillage de courbes d'autant plus fin que la courbure de la spline est forte,
- réaliser un maillage de courbes plus grossier, mais y adjoindre certaines informations (coefficients de polynômes cubiques par exemple).

La première méthode a été implémentée dans le logiciel BL2D. En outre, le maillage fin est précédé de sa *bounding box*.

Exemple de fichier au format SMOOTH

Le fichier SMOOTH ci-dessous a été utilisé dans notre exemple. (cf. section 1.2). Le quart de cercle est approché par deux splines cubiques (FIG. 3.8).

```
SMOOTH v1
  0.0000000000000000  0.0000000000000000
  1.0000000000000000  1.0000000000000000
C v1
37 1 5 31 5
  0.4000000000000000  0.4000000000000000
  0.0000000000000000  0.0000000000000000
  1.0000000000000000  0.0000000000000000
  ...
```

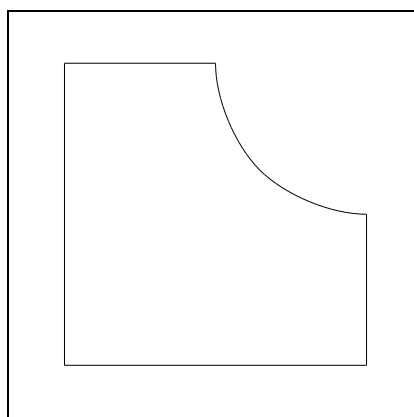


FIG. 3.8 - *Tracé à partir du fichier quart.0.smooth.*

Chapitre 4

Structures de données

Le logiciel BL2D est écrit en deux langages : Fortran 90 et en C. La programmation de ce logiciel est basée sur le concept de *classe*, qui regroupe en une même entité des *structures de données* et des *procédures*. L'implémentation en Fortran 90 est réalisée selon le tableau de correspondance suivant :

Terminologie "objet"	Terminologie Fortran 90
classe	module
– structure	– type dérivé
– procédure	– procédure (subroutine ou fonction)

Ce chapitre décrit les principales structures de données implémentées en Fortran 90. Les procédures, quant à elles, sont décrites au chapitre suivant.

Les principaux formats définis précédemment peuvent être représentés en mémoire par des structures de données. Ainsi, aux formats G, C, MS et H correspondent respectivement les structures *g*, *c*, *s* et *h*. L'organisation des structures étant très voisine de celle des formats, et les noms des composants des structures étant relativement explicites, seules les déclarations en Fortran 90 sont reproduites ci-dessous.

4.1 Structure *g*

```
type g_point_
  integer :: ip, physique
end type g_point_

type g_spline_
  integer, dimension(:), pointer :: points
  integer :: physique
end type g_spline_

type g_domaine_
  integer :: is, orientation, physique
end type g_domaine_

type g_
  integer :: np, npi, npe, npg, ns, nd
  double precision, dimension(:,:), pointer :: coor ! (2,:)
  type(g_point_), dimension(:), pointer :: imposes
  type(g_point_), dimension(:), pointer :: extremités
  type(g_spline_), dimension(:), pointer :: splines
  type(g_domaine_), dimension(:), pointer :: domaines
end type g_
```

4.2 Structure *c*

```
type c_spline_
  integer, dimension(:), pointer :: points
  double precision, dimension(:), pointer :: abscisses
end type c_spline_

type c_
  integer :: np, npi, npe, npc, ns
  double precision, dimension(:,:), pointer :: coor ! (2,:)
  type(c_spline_), dimension(:), pointer :: splines
end type c_
```

4.3 Structure *s*

```
type s_  
  integer :: np, nt  
  double precision, dimension(:,:), pointer :: coor    ! (2,:)   
  integer, dimension(:,:), pointer :: triangles      ! (3,:)   
  integer, dimension(:,:), pointer :: voisins       ! (3,:)   
  integer, dimension(:), pointer :: domaines        ! (:)  
end type s_
```

4.4 Structure *h*

```
type h_  
  integer :: np  
  character(len=10) :: type_met  
  double precision, dimension(:,:), pointer :: mets    ! (3,:)   
end type h_
```

Chapitre 5

Procédures

5.1 Lecture et écriture

Les procédures suivantes lisent ou écrivent des structures de types g , c , s ou h définis précédemment :

```
subroutine g_read(g)          subroutine g_write(g)
subroutine c_read(c)          subroutine c_write(c)
subroutine s_read(s)          subroutine s_write(s)
subroutine h_read(h)          subroutine h_write(h)
```

Les extraits qui concernent les allocations de tableaux, à l'intérieur des procédures de lecture, sont reproduits ci-dessous :

```
! structure g_
allocate(g%coor (1:2, g%np))
allocate(g%imposes (g%mpi))
allocate(g%extremities(g%npe))
allocate(g%splines (g%ns))
allocate(g%domaines (g%nd))
do i = 1, c%ns
  allocate(g%splines(i)%points(n))
end do

! structure c_
allocate(c%splines(c%ns))
allocate(c%coor(1:2, c%np))
```

```

allocate(c%splines(c%ns))
do i = 1, c%ns
  allocate(c%splines(i)%points(n))
  allocate(c%splines(i)%abscisses(n))
end do

! structure s_
allocate(s%coor(1:2, s%np))
allocate(x_s%triangles(3, x_s%nt))
allocate(x_s%voisins (3, x_s%nt))
allocate(x_s%domaines ( x_s%nt))

! structure h_
allocate(h%nets(3, h%np))

```

5.2 Références physiques

La référence physique d'un point p , d'une arête (p, q) ou d'un triangle t est obtenue respectivement par les fonctions `physique_p`, `physique_a` et `physique_d`:

```

function physique_p(p, x_g, x_c) ! p = numero dans le maillage
  integer, intent(in) :: p
  type(g_), intent(in) :: x_g
  type(c_), intent(in) :: x_c
  integer :: physique_p ! intent(out)
end function physique_p

function physique_a(p, q, x_g, x_c)
  integer, intent(in) :: p, q
  type(g_), intent(in) :: x_g
  type(c_), intent(in) :: x_c
  integer :: physique_a ! intent(out)
end function physique_a

function physique_d(t, x_g, x_s)
  integer, intent(in) :: t
  type(g_), intent(in) :: x_g
  type(s_), intent(in) :: x_s
  integer :: physique_d ! intent(out)
end function physique_d

```

Les trois fonctions précédentes ont été appelées pour tracer les figures ci-dessous (FIG. 5.1, 5.2 et 5.3).

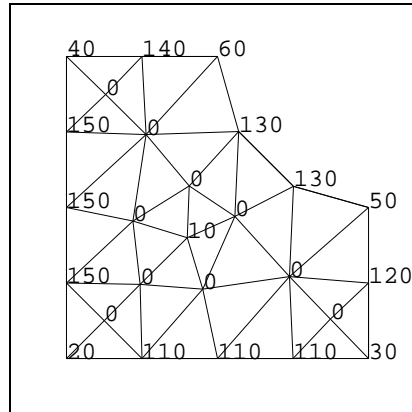


FIG. 5.1 - *Références physiques des points.*

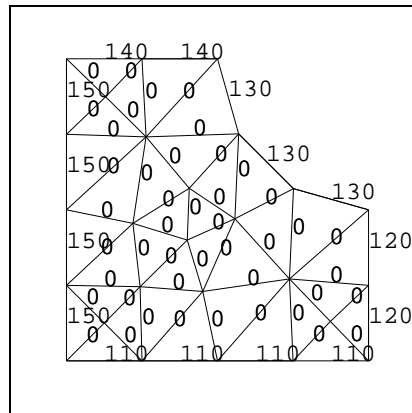
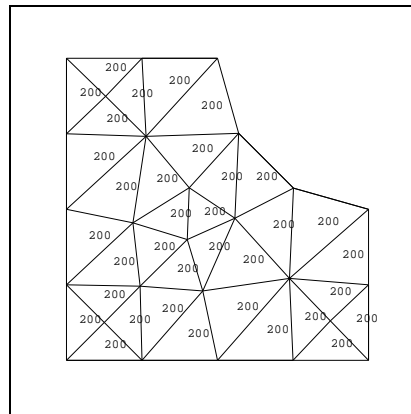


FIG. 5.2 - *Références physiques des arêtes.*

FIG. 5.3 - *Références physiques des sous-domaines.*

5.3 Splines

La fonction `eval_s` retourne les coordonnées du point situé sur la spline `is` et d'abscisse curviligne `s` :

```
function eval_s(x_c, is, abscisse)
  type(c_), intent(in) :: x_c
  integer, intent(in) :: is
  double precision, intent(in) :: abscisse
  double precision :: eval_s(2) ! intent(out)
end function eval_s
```

Étant donnée une arête (p, q) , la subroutine suivante cherche si cette arête appartient à une spline. Si oui, elle retourne le numéro de la spline et les indices des points p et q . Sinon, elle retourne 0.

```
subroutine donner_spline(p, q, x_c, is, ip, iq)
  integer, intent(in) :: p, q
  type(c_), intent(in) :: x_c
  integer, intent(out) :: is, ip, iq
end subroutine donner_spline
```

En utilisant les deux procédures précédentes, on peut par exemple calculer les coordonnées du nœud P2 situé au milieu d'une arête (p, q) :

```
call donner_spline(p, q, x_c, is, ip, iq)
if (is == 0) then ! arete droite
  coor = (x_s%coor(1:2,p) + x_s%coor(1:2,q)) / 2
else ! arete courbe
  abscisse = (x_c%splines(is)%abscisses(ip) + &
             x_c%splines(is)%abscisses(iq)) / 2
  coor = eval_s(x_smooth%c, is, abscisse)
end if
```

Cette méthode a été appliquée à notre exemple précédent. Les nœuds sont situés correctement au voisinage du quart de cercle (FIG. 5.4).

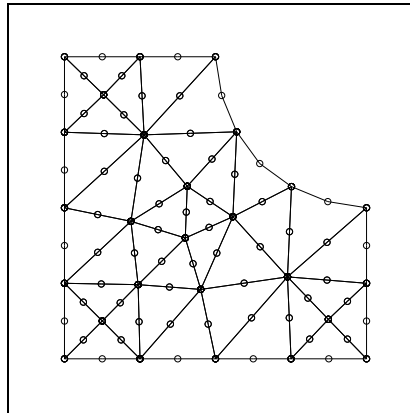


FIG. 5.4 - *Triangles P2.*

Bibliographie

- [1] H. BOROUCAKI et P. LAUG, *Maillage de courbes gouverné par une carte de métriques*, Rapport de Recherche INRIA à paraître.
- [2] H. BOROUCAKI, P.L. GEORGE, F. HECHT, P. LAUG et E. SALTEL, *Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: algorithmes*, Rapport de Recherche INRIA n° 2741, décembre 1995.
- [3] H. BOROUCAKI, P.L. GEORGE, F. HECHT, P. LAUG, B. MOHAMMADI et E. SALTEL, *Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie II: applications*, Rapport de Recherche INRIA à paraître.
- [4] E. CATMULL, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, Univ. Utah Comp. Sci. Dept. UTEC-CSC-74-133, 1974.
- [5] P.L. GEORGE et P. LAUG, *Normes d'utilisation et de programmation*, Guide Modulef n° 2, INRIA, 1992.
- [6] P.L. GEORGE et E. SALTEL, *Post-traitements et graphiques*, Guide Modulef n° 6, INRIA, 1992.
- [7] P. LAUG, *Rapport intermédiaire "GÉNIE" - Tâches T-2.4.3 à T-2.4.6*, DASSAULT AVIATION - INRIA, 1995.
- [8] E. SALTEL et F. HECHT, *EMC² : un logiciel d'édition de maillages et de contours bidimensionnels*, Rapport Technique INRIA n° 118, octobre 1995.

- [9] M.G. VALLET, *Génération de maillages éléments finis anisotropes et adaptatifs*, Thèse Université Paris 6, 1992.



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