

Vincent Manet

Méthode des éléments finis

*Vulgarisation des aspects mathématiques
et illustration de la méthode*

Vincent Manet — 2013 (Ceci est la version « livre » de ce document)

Ce document est sous licence Creative Commons 3.0 France :

- paternité ;
- pas d'utilisation commerciale ;
- partage des conditions initiales à l'identique ;

<http://creativecommons.org/licenses/by-nc-sa/3.0/deed.fr>



Introduction

Dans ce (de moins en moins court) document, plutôt à destination d'ingénieurs mécaniciens connaissant déjà la méthode des éléments finis, nous allons essayer de faire une présentation un peu plus théorique que ce qui leur est généralement proposé (et qui est quand même souvent de type « preuve par les mains », ce qui occulte trop de points).

Nous ne ferons appel qu'à des notions mathématiques de bases généralement déjà vues pour la plupart en taupe (ou en tout début de cycle d'ingé)... bien que des compléments que l'on peut qualifier d'élémentaires nous aient été demandés et aient été inclus.

Nous espérons, grâce à cette présentation théorique montrer toute la souplesse et la puissance de la méthode, afin de permettre au lecteur d'envisager d'autres simulations que celles qu'il a pu déjà réaliser par le passé.

But du document

Le but initial était de *présenter brièvement la théorie mathématique* derrière les éléments finis afin que les ingénieurs utilisant cette méthode puisse en envisager toutes les applications, ainsi que de *couvrir les aspects qui, selon nous, devraient être connus de tout ingénieur mécanicien impliqué ou intéressé par le calcul numérique.*

Toutefois, il s'envisage comme support de référence à plusieurs cours, cours qui ne portent pas sur tous les aspects traités dans ce document, et pendant lesquels les aspects pratiques sont plus développés (avec mise en situation sur machine).

Même si nous avons voulu rester le plus succinct possible, l'introduction de notions de proche en proche a conduit à un document fait aujourd'hui une certaine taille (par exemple, nous avons besoins des espaces de Sobolev, mais comment les introduire sans parler des espaces de Lebesgue, mais comment les introduire sans parler...).

Aussi le document a-t-il finalement été découpé en plusieurs parties : un survol des notions mathématiques, puis le traitement du problème continu constituent l'ossature théorique nécessaire à asseoir la MEF sur un socle solide. La discrétisation par éléments finis à proprement parler n'est abordé qu'ensuite, et d'ailleurs un seul chapitre suffirait à en faire le tour... sauf à entrer plus dans le détail concernant « ce qui fâche » : homogénéisation, non linéarité, dynamique, ce qui est fait dans des chapitres séparés.

Enfin, d'autres méthodes sont abordées car également très employées aujourd'hui. Aussi est-il indispensable selon nous d'en avoir entendu parlé et d'en connaître les principales notions (BEM, FEEC...).

En annexes, se trouve un petit fourre-tout comprenant des choses censées être maîtrisées depuis la taupe (mais qui parfois nous sont demandées) et les compléments qui alourdiraient encore les propos précédents.

Certaines notions (essentiellement de topologie) ne sont pas présentées dans ce document. Il nous a semblé que le lecteur devait avoir quelques souvenirs de ce qu'est un ouvert, un fermé, l'adhérence, la densité... Par ailleurs, leur nom peut être suffisamment évocateur pour se passer d'une définition formelle dans le contexte de ce document.

Attention, ce document n'est pas un document de mathématiques, il ne contient d'ailleurs aucune preuve. C'est, dans ces deux premières parties, un document de vulgarisation de notions mathématiques nécessaires à une bonne compréhension de la méthode des éléments finis.

Nous avons voulu réaliser un survol des notions importantes, mais malgré tout, afin de ne pas être parfois trop laconique, nous avons un peu débordé.

En fin de document, un petit index des noms propres permettra au lecteur de replacer les divers développements mentionnés dans l'histoire... Il se peut qu'il subsistent quelques erreurs, notamment au niveau des nationalités mentionnées, car il n'est pas toujours aisé de déterminer rapidement cette information (et nous ne connaissons pas toutes les biographies des personnes citées).

Ce document a été réalisé très rapidement, et de manière extrêmement hachée. Il comporte forcément encore beaucoup de fautes : merci de m'en faire part.

Démarche de l'ingénieur numéricien

En préambule à ce document, nous tenions à synthétiser la démarche complète de l'ingénieur numéricien :

- Modélisation / mise en équations – Construction du problème continu (système d'EDP).
- Analyse mathématique du problème posé – Existence, unicité, propriétés des solutions.
- Conception d'une méthode numérique – Construction d'un problème discrétisé.
- Analyse numérique – Questions de stabilité, convergence, précision.
- Algorithmique – Choix de méthodes de résolution en dimension finie.
- Mise en œuvre sur ordinateur – Programmation.
- Pre et Post Traitement (maillages / visualisation) – Interpolation, extrapolation, outils de la CAO.

Tous ces points ne seront évidemment pas abordés dans ce document !

Remerciements :

Nous n'avions pas prévu de réaliser une deuxième version aussi rapidement. Celle-ci existe suite aux sollicitations de Mathias Legrand. C'est lui qui a développé les macros nécessaires à l'amélioration très très nette de la qualité typographique (environnements pour les notes historiques, les théorèmes, lemmes...).

C'est également pourquoi coexistent aujourd'hui deux versions (mais issues du même code source) : l'un que nous appelons « version cours » (plus en accord avec ce que nous proposons en cours), et l'autre « version livre », plus proche d'un ouvrage.

Table des matières

Introduction	3
But du document	3
Démarche de l'ingénieur numéricien	4
Table des matières	5

I	ÉLÉMENTS FINIS
1	Calcul efficient : qualité des résultats et efforts de calcul 9
1.1	Amélioration d'un modèle : méthodes r, h et p 9
1.2	Post-traitement 9
1.3	Exemple d'implémentation d'un post-traitement dans ANSYS 10
1.3.1	Macro dans ANSYS 10
1.3.2	Poutre en U 13
1.3.3	Résultats 15
1.4	Sous-structuration et simulation multi-échelles 16
1.4.1	Condition de périodicité – méthodes multi-niveaux 20
1.4.2	Couplage des cellules micro – méthodes de décomposition de domaine 22
1.5	Super-éléments 24
1.5.1	Condensation statique 25
1.5.2	Remonter aux ddl internes 26
1.6	Pseudo-inversion et réanalyse 26
1.6.1	Modification du chargement uniquement 27
1.6.2	Modification de la matrice 27
1.6.3	Modification des conditions cinématiques 27
1.7	Dérivées d'ordre supérieur 29
1.7.1	Dérivées par rapport à la géométrie 29
1.7.2	Calcul des dérivées 30

Chapitre 1

Calcul efficient : qualité des résultats et efforts de calcul

1.1 Amélioration d'un modèle : méthodes r , h et p

Les indicateurs d'erreur ont pour but de nous indiquer si « nous sommes loin de la solution », afin de pouvoir modifier le modèle si besoin. Ainsi, par itérations successives, on pourra tendre vers une solution de plus en plus proche de la solution exacte.

Les indicateurs locaux nous permettent notamment de faire des modifications uniquement locales du modèle, i.e. uniquement là où il y en a besoin, par exemple en raffinant le maillage.

Globalement, il existe trois stratégies pour améliorer la précision de la solution obtenue :

Méthode r Pour un maillage et un type d'élément donné, il s'agit de déplacer les nœuds, en fonction des indicateurs d'erreur, et donc sans impacter le nombre de ddl du système. Ainsi, la taille des éléments peut augmenter (maillage plus grossier) dans les zones les moins sollicitées et diminuer (maillage plus fin) dans les zones les plus sollicitées. On voit que la restriction de la méthode est qu'elle ne joue pas sur le nombre de nœuds, et que par conséquent, sans violer les contraintes de distorsion, elle est limitée.

Méthode h En conservant le même type d'élément, on les subdivise dans les zones les plus sollicitées selon le ou les indicateurs d'erreur choisis. Dans cette méthode, on augmente le nombre de nœuds afin de contrôler les erreurs et la précision du modèle. Il est nécessaire de se fixer une limite dans la précision recherchée afin de ne pas trop raffiner le maillage.

Méthode p À nombre d'éléments constant, dans les zones les plus sollicitées, on va modifier les éléments en introduisant des fonctions de formes polynomiales d'ordre plus élevé, dites hiérarchiques. La complexité du système est cette fois encore accrue.

Évidemment, ces méthodes peuvent être combinées. La méthode h - p propose de modifier à la fois le maillage et les fonctions d'interpolation. Elle semble aujourd'hui être la méthode optimale en terme d'efficacité et de vitesse de convergence.

Notons que les méthodes r - et h - dépendent des capacités du mailleur automatique implémenté. De nombreux travaux existent sur le maillage automatique, nous n'entrerons pas dans ce détail.

1.2 Post-traitement

Une manière d'améliorer les résultats est de recourir à des méthodes dites de post-traitement, i.e. des méthodes qui, à partir des données issues de la résolution du système matriciel correspondant au problème, fournissent des données complémentaires ou améliorent toute ou partie des données déjà disponibles.

En fonction des problèmes (donc des données disponibles et des données souhaitées) de nombreuses méthodes existent. Elles sont généralement dédiées à un problème donné.

Un exemple déjà abordé dans ce document est celui où, à partir des déplacements nodaux obtenus par un calcul EF d'une structure mécanique composée de deux matériaux différents dans une discrétisation classique « en déplacements », on souhaite remonter aux contraintes à l'interface entre lesdits matériaux.

Une méthode déjà mentionnée est la méthode dite de « Reissner local » qui consiste à intégrer les équations d'équilibre sous la forme mixte de Reissner, mais uniquement sur des paires d'éléments ayant des propriétés matérielles différentes et possédant une face commune.

On obtient alors un champ de contrainte complètement continu dont on ne retient que les composantes correspondant à la trace des contraintes, les autres étant calculées comme dans la méthode classique. On montre que l'on améliore grandement la qualité de l'approximation des contraintes aux interfaces, mêmes avec un maillage grossier.

Un autre exemple serait, disposant d'un point de pression constant par élément, de calculer la pression en un point quelconque comme interpolation (linéaire ou non) à partir des points disponibles.

Un troisième exemple serait à partir des données nodales dans un modèle 1D (poutre ou barre selon le cas), de remonter à la répartition des contraintes en un point quelconque de la structure (donc via les hypothèses faites dans le modèle sur la répartition des contraintes dans une section + via une interpolation lorsque l'on se trouve dans une section ne passant pas par un nœud)...

1.3 Exemple d'implémentation d'un post-traitement dans ANSYS

Dans ce chapitre, nous présentons une manière d'implémenter la méthode de Reiss local mentionnée aux paragraphes 1.2 et ??.

Il s'agit de calculer les contraintes à l'interface entre deux matériaux en utilisant la fonctionnelle mixte d'Hellinger-Reissner donnée à l'équation (??).

Le but est de montrer que même sur un maillage grossier, il est possible de correctement estimer les contraintes aux interfaces, pour peu que l'on dispose d'une méthode appropriée. Nous espérons qu'au passage, nous démontrerons également qu'il est assez simple d'implémenter des fonctions dans un code existant.

1.3.1 Macro dans ANSYS

Si un code ne dispose pas de la méthode que l'on souhaite utiliser, il est toujours possible de retraiter les résultats... toutefois, selon les codes, il est plus ou moins aisé d'implémenter des fonctions dans ledit code.

ANSYS est souvent qualifié de « code industriel », ce qui pourrait laisser supposer qu'il n'a pas la même capacité à résoudre les problèmes que des codes dits « de recherche » ou « spécialisés ». Or il n'en est rien (même si en toute rigueur, il y a une quinzaine d'années il était plus faible que d'autres sur les non linéarités, ce qui n'est plus vrai depuis longtemps).

C'est vrai que disposant d'une interface utilisateur remarquable (comparée à celle de nombreux autres codes), il est possible de ne réaliser les calculs que via cette interface, i.e. sans passer par un fichier batch... ce que ne font de toutes façons pas les gens sérieux (donc cet argument ne tient pas).

Le qualificatif d'industriel peut également laisser penser que le logiciel est plus fermé que d'autres, mais ce n'est que partiellement le cas, en tous les cas on peut y remédier facilement, et c'est pourquoi nous allons présenter l'implémentation d'une macro dans ce code.

Pour implémenter une nouvelle fonction dans ANSYS, plusieurs méthodes s'offrent à nous. Nous citerons :

- écrire le bout de programme et recompiler le noyau : c'est faisable en théorie, mais personnellement je n'ai jamais réussi...

- partir des fichiers de résultats binaires pour les retravailler : c'est également théoriquement faisable, mais il faut décortiquer les formats d'écriture desdits fichiers, les relire, réécrire dans le même format... et c'est donc beaucoup de travail (purements informatique) ;
- programmer la fonction directement sous ANSYS (ANSYS possède un langage assez riche et puissant) ;
- ou alors, et c'est la voie que nous allons montrer, utiliser ANSYS en « coopération » avec un petit programme extérieur.

La structure de la macro ANSYS que nous proposons (et qui peut être également incluse dans les menus d'ANSYS, mais nous n'entrons pas dans ce niveau de détail d'interfaçage) est très simple :

- on récupère les données dont nous avons besoin et on les stocke dans des fichiers externes (simples) ;
- on lance un programme externe (dont la structure sera exposée plus bas) ;
- on réintègre les résultats dans ANSYS.

On peut alors se servir de toutes les fonctions de visualisation disponibles dans ANSYS avec les résultats modifiés...

```

1 /nop
2 ! Vincent MANET
3 ! ENSM.SE, 1998
4 !
5 nall
6 !
7 ! Parametres
8 *get,Nnoeuds,node,,count
9 *get,Nmater,mat,,count
10 *get,Nelem,elem,,count
11 *cfcopen,temp,par
12 *vwrite,Nnoeuds
13 (E13.7,' ')
14 *vwrite,Nelem
15 (E13.7,' ')
16 *vwrite,Nmater
17 (E13.7,' ')
18 *cfcclose
19 !
20 ! Materiaux
21 *cfcopen,temp,mat
22 *dim,Mater,,Nmater,3
23 *do,I,1,Nmater
24     *get,Mater(I,1),Ex,I
25     *get,Mater(I,2),Ey,I
26     *get,Mater(I,3),Nuxy,I
27 *enddo
28 *vwrite,Mater(1,1),Mater(1,2),Mater(1,3)
29 (3(E13.7,' '))
30 *cfcclose
31 !
32 ! Noeuds
33 *dim,CoordX,,Nnoeuds
34 *dim,CoordY,,Nnoeuds
35 *dim,CoordZ,,Nnoeuds
36 *vget,CoordX(1),node,,loc,x
37 *vget,CoordY(1),node,,loc,y
38 *cfcopen,temp,coo
39 *vwrite,CoordX(1),CoordY(1)
40 (2(E13.7,' '))
41 *cfcclose
42 !
43 ! Elements + numero du materiau
44 *dim,E1,Nelem,9
45 *vget,E1(1,1),elem,1,node,1
46 *vget,E1(1,2),elem,1,node,2
47 *vget,E1(1,3),elem,1,node,3
48 *vget,E1(1,4),elem,1,node,4
49 *vget,E1(1,5),elem,1,node,5
50 *vget,E1(1,6),elem,1,node,6
51 *vget,E1(1,7),elem,1,node,7
52 *vget,E1(1,8),elem,1,node,8
53 *vget,E1(1,9),elem,1,attr,mat
54 *cfcopen,temp,elt
55 *vwrite,E1(1,1),E1(1,2),E1(1,3),E1(1,4),E1(1,5),E1(1,6),
56     E1(1,7),E1(1,8),E1(1,9)
57 (9(E13.7,' '))
58 *cfcclose
59 !
60 ! Resultats
61 *vget,CoordX(1),node,1,U,x
62 *vget,CoordY(1),node,1,U,y
63 *cfcopen,temp,res
64 *vwrite,CoordX(1),CoordY(1)
65 (2(E13.7,' '))
66 *cfcclose
67 !
68 ! Contraintes
69 *vget,CoordX(1),node,1,S,x
70 *vget,CoordY(1),node,1,S,y
71 *vget,CoordZ(1),node,1,S,xy
72 *cfcopen,temp,str
73 *vwrite,CoordX(1),CoordY(1),CoordZ(1)
74 (3(E13.7,' '))
75 *cfcclose
76 !
77 ! run locreiss.exe
78 /sys,'locreiss.exe'
79 !
80 ! Effacer les fichiers
81 !   generes par ANSYS :
82 !       - temp.par : parametres
83 !       - temp.coo : coordonnees nodales
84 !       - temp.elt : incidences
85 !       - temp.mat : proprietes materielles
86 !       - temp.res : déplacements nodaux
87 !       - temp.str : contraintes nodales
88 !   generes par locreiss.exe :
89 !       - temp.dum
90 !       - temp.tmp
91 ! Il reste les fichiers
92 !       - temp.sij : contraintes nodales modifiees
93 !
94 !/sys,'rm ./temp.par'
95 !/sys,'rm ./temp.coo'
96 !/sys,'rm temp.elt'
97 !/sys,'rm temp.mat'
98 !/sys,'rm temp.res'
99 !/sys,'rm temp.str'
100 !/sys,'rm temp.dum'

```

```

101 !/sys,'rm temp.tmp'
102 !
103 ! lire les valeurs dans temp.sij
104 ! ces fichiers contiennent: sxx, syy, sxy
105 ! ce sont toutes les composantes continues.
106 ! Elles n'ont pas toutes un sens:
107 ! Sxx discontinu, mais dans le repere
108 ! local, donc on la laisse pour qu'ANSYS
109 ! puisse faire la ! rotation de repere
110 ! si necessaire.
111 *vread,CoordX(1),temp,sxx,,
112 (E13.7)
113 *vread,CoordY(1),temp,syy,,
114 (E13.7)
115 *vread,CoordZ(1),temp,sxy,,
116 (E13.7)
117 *vput,CoordX(1),node,1,S,x
118 *vput,CoordY(1),node,1,S,y
119 *vput,CoordZ(1),node,1,S,xy
120 !
121 ! Cleanup
122 CoordX(1)=
123 CoordY(1)=
124 CoordZ(1)=
125 Mater(1,3)=
126 E1(1,9)=
127 Nnoeuds=
128 Nmater=
129 Nelem=
130 !
131 /gop

```

Le programme extérieur locreiss.exe, est structuré comme suit :

```

1      PROGRAM VM2
2 C
3      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
4      IMPLICIT INTEGER (I-N)
5 C
6 C-----
7 C
8 C  But:
9 C    1. on relit les fichiers d'ANSYS generes avec la macro INTERF
10 C    2. on determine les noeuds de post-traitement
11 C    3. on effectue un Reissner local aux interfaces
12 C
13 C  WARNING:
14 C    On ne traite que le cas de PLANE 82 avec une geometrie definie
15 C    dans le plan (X,Y)
16 C
17 C  MAIN of : VM
18 C
19 C  Written by : Vincent MANET
20 C              EMSE, 98
21 C
22 C-----
23 C
24      PARAMETER(NODMAX=1000,NELTMAX=300,MATMAX=4)
25 C
26      COMMON /VMCTS/ COMPI,ZERO,ONE,IINP,IOUT,ICON,MONI
27      COMMON /NPB/ NNOD,NELT,NMAT,NCAL
28 C
29 C
30      DIMENSION COORD(NODMAX,3),DISP(NODMAX,2)
31      DIMENSION XMAT(MATMAX,3),NLT(NELTMAX,9)
32      DIMENSION NCALC(NODMAX,5),NADJ(NODMAX,8),NDT(NODMAX,8)
33      DIMENSION D1(3,3),D2(3,3)
34 C
35 C---- Presettings
36      MONI=0          ! monitoring level (debug)
37      ICON=6          ! standard output = screen
38      IINP=10         ! input file
39      IOUT=20         ! output file
40      ZERO=0.0D0      ! zero
41      ONE=1.0D0       ! one
42      COMPI=DACOS(-ONE) ! pi
43 C
44 C
45 C---- Read input files
46      CALL READANS(NODMAX,NELTMAX,MATMAX,COORD,XMAT,NLT,DISP)
47 C
48 C---- Find Adjacent elements
49      CALL CHELT(NODMAX,NELTMAX,NLT,NCALC,NADJ,NDT)
50 C
51 C---- Material properties
52      CALL MATER(MATMAX,XMAT,D1,D2,D3,D4,NODMAX,NADJ,NELTMAX,NLT)
53 C

```

```

54 C---- Computation: Local Reissner
55     CALL LOCREISS(NODMAX,NELTMAX,D1,D2,D3,D4,
56     x             COORD,NLT,NADJ,DISP)
57 C
58 C---- Average multi-computed nodes
59     CALL AVERAGE(NODMAX,MULTI)
60 C
61 C---- Last line of VM2
62     END

```

La routine READANS permet de lire les données. Celles-ci sont stockées dans les matrices COORD(NODMAX,3), DISP(NODMAX,2), XMAT(MATMAX,3) et NLT(NELTMAX,9).

la routine CHELT détecte les interfaces, i.e. les faces dont les nœuds appartiennent à des éléments dont les propriétés matérielles sont différentes. Le vecteur NADJ(NODMAX,8) contient les numéros des deux éléments adjacents (positions 1 et 2) ainsi que les numéros des nœuds de l'interface (position 3, 4 et 5).

La routine MATER construit, pour chacune des faces de l'interface, la matrice de rigidité (aussi bien pour le cas isotrope que pour le cas orthotrope).

La routine LOCREISS calcule les contraintes nodales le long de chaque face.

Les contraintes ayant été calculées pour les nœuds de chacune des faces de l'interface, on moyenne les résultats pour un nœud appartenant à plusieurs faces. C'est ce que fait la routine AVERAGE. À ce stade, nous disposons donc des contraintes nodales, calculées par la fonctionnelle mixte d'Helliner-Reissner, le long de toutes les interfaces présentes dans le modèle. Il ne reste plus qu'à écrire ces résultats pour qu'ils soient relus par la macros ANSYS.

1.3.2 Poutre en U

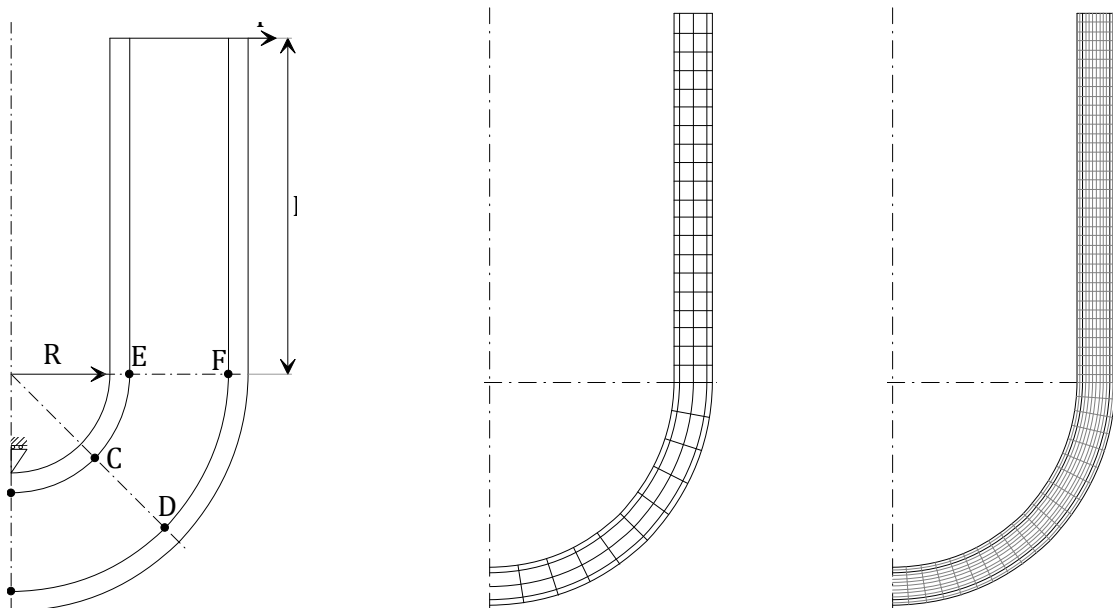


FIGURE 1.1 – a) Géométrie de la poutre en U, b) le maillage utilisé pour le calcul (test de la méthode Reissner local, c) maillage permettant d'obtenir la « solution de référence » puisqu'il n'y a pas de solution analytique

Disposant de la macro précédente, nous allons l'appliquer au calcul d'une poutre en U.

Il s'agit d'une poutre sandwich dont les peaux sont en aluminium et l'âme en résine époxy. Par symétrie, seule une demi-poutre est modélisée. Elle est soumise à une force unitaire sur chaque montant.

Les points pour lesquels nous nous intéresserons aux contraintes sont les points indiqués sur la figure 1.1, à savoir les points A, B, C, D, E et F.

Le listing ANSYS est le suivant :

```

1 !* geom: partie droite
2 long=20
3 H1=0.2
4 H2=1.8
5 H3=2
6 !*
7 !* geom coude
8 Rint=10
9 Rh1=Rint+H1
10 Rh2=Rint+H2
11 Rh3=Rint+H3
12 !*
13 !* maillage
14 NCUTS=40
15 NRCUTS=20
16 NCORE=16
17 NSKIN=2
18 Forc=1
19 !*
20 /PREP7
21 !*
22 ET,1,PLANE82
23 !*
24 UIMP,1,EX, , ,70000,
25 UIMP,1,NUXY, , ,0.3,
26 UIMP,1,EMIS, , ,1,
27 !*
28 !*
29 UIMP,2,EX, , ,3400,
30 UIMP,2,NUXY, , ,0.34,
31 UIMP,2,EMIS, , ,1,
32 !*
33 k,1,0,-Rh3
34 k,2,Rh3,0
35 k,3,Rh2,0
36 k,4,0,-Rh2
37 k,5,0,-Rh1
38 k,6,Rh1,0
39 k,7,Rint,0
40 k,8,0,-Rint
41 k,9,Rh3,long
42 k,10,Rh2,long
43 k,11,Rh1,long
44 k,12,Rint,long
45 k,13,0,0
46 !*
47 larc,1,2,13,Rh3,NRCUTS
48 l,2,3,NSKIN
49 larc,4,3,13,Rh2,NRCUTS
50 l,1,4,NSKIN
51 l,4,5,NCORE
52 l,3,6,NCORE
53 larc,5,6,13,Rh1,NRCUTS
54 l,5,8,NSKIN
55 larc,8,7,13,Rint,NRCUTS
56 l,6,7,NSKIN
57 !*
58 l,2,9,NCUTS
59 l,9,10,NSKIN
60 l,10,3,NCUTS
61 l,10,11,NCORE
62 l,11,6,NCUTS
63 l,11,12,NSKIN
64 l,12,7,NCUTS
65 !*
66 mat,1
67 a,1,2,3,4
68 a,5,6,7,8
69 a,2,9,10,3
70 a,6,11,12,7
71 mat,2
72 a,4,3,6,5
73 a,3,10,11,6
74 mat,1
75 amesh,1
76 amesh,2
77 amesh,3
78 amesh,4
79 mat,2
80 amesh,5
81 amesh,6
82 !*
83 DL,4,,symm
84 DL,5,,symm
85 DL,8,,symm
86 !*
87 dk,8,uy,0
88 !*
89 fk,9,fx,Forc
90 !*
91 FINISH
92 /SOLU
93 /STAT,SOLU
94 SOLVE
95 FINISH
96 /POST1
97 pldisp,1
98 *USE,INTERF
99 ...

```

On utilise la macro en tapant *USE, INTERF, et les contraintes sont directement modifiées dans ANSYS. On peut alors utiliser les mêmes fonctions qu’habituellement pour visualiser les différentes composantes des contraintes.

Le listing Cast3M est le suivant :

```

1 * geom: partie droite
2 long=20.0;
3 H1=0.2;
4 H2=1.8;
5 H3=2.0;
6 *
7 * geom coude
8 Rint=10.0;
9 Rh1=Rint+H1;
10 Rh2=Rint+H2;
11 Rh3=Rint+H3;
12 MRint = -1. * Rint;
13 MRh1 = -1. * Rh1;
14 MRh2 = -1. * Rh2;
15 MRh3 = -1. * Rh3;
16 *
17 * maillage
18 NCUTS=40;
19 NRCUTS=20;
20 NCORE=16;
21 NSKIN=2;
22 Forc=1.0;
23 *
24 OPTION DIMENSION 2 ELEMENT QUA4;

```

```

25 *
26 k1 = 0. MRh3;
27 k2 = Rh3 0.;
28 k3 = Rh2 0.;
29 k4 = 0. MRh2;
30 k5 = 0. MRh1;
31 k6 = Rh1 0.;
32 k7 = Rint 0.;
33 k8 = 0. MRint;
34 k9 = Rh3 long;
35 k10 = Rh2 long;
36 k11 = Rh1 long;
37 k12 = Rint long;
38 k13 = 0. 0.;
39 *
40 L1 = CERCLE NRCUTS k1 k13 k2 ;
41 L2 = DROITE k2 k3 NSKIN;
42 L3 = CERCLE NRCUTS k4 k13 k3 ;
43 L4 = DROITE k1 k4 NSKIN;
44 L5 = DROITE k4 k5 NCORE;
45 L6 = DROITE k3 k6 NCORE;
46 L7 = CERCLE NRCUTS k5 k13 k6 ;
47 L8 = DROITE k5 k8 NSKIN;
48 L9 = CERCLE NRCUTS k8 k13 k7 ;
49 L10 = DROITE k6 k7 NSKIN;
50 *
51 L11 = DROITE k2 k9 NCUTS;
52 L12 = DROITE k9 k10 NSKIN;
53 L13 = DROITE k10 k3 NCUTS;
54 L14 = DROITE k10 k11 NCORE;
55 L15 = DROITE k11 k6 NCUTS;
56 L16 = DROITE k11 k12 NSKIN;
57 L17 = DROITE k12 k7 NCUTS;
58 *
59 SURF1 = DALLER L1 L2 L3 L4;
60 SURF2 = DALLER L2 L11 L12 L13;
61 PEAUEXT = SURF1 ET SURF2;
62 * ELIMINE PEAUEXT;
63 SURF1 = DALLER L3 L6 L7 L5;
64 SURF2 = DALLER L6 L13 L14 L15;

65 AME = SURF1 ET SURF2;
66 * ELIMINE AME;
67 SURF1 = DALLER L8 L7 L10 L9;
68 SURF2 = DALLER L10 L15 L16 L17;
69 PEAUINT = SURF1 ET SURF2;
70 * ELIMINE PEAUINT;
71 PoutU = PEAUEXT ET AME ET PEAUINT;
72 *
73 Model1 = MODL peauext MECANIQUE ELASTIQUE ISOTROPE QUA4;
74 Model2 = MODL ame MECANIQUE ELASTIQUE ISOTROPE QUA4;
75 Model3 = MODL peauint MECANIQUE ELASTIQUE ISOTROPE QUA4;
76 ModelTot = Model1 ET Model2 ET Model3;
77 *
78 Mater1 = MATERIAU Model1 YOUNG 70000.0 NU 0.3 RHO 2700.0;
79 Mater2 = MATERIAU Model2 YOUNG 3400.0 NU 0.34 RHO 1000.0;
80 Mater3 = MATERIAU Model3 YOUNG 70000.0 NU 0.3 RHO 2700.0;
81 *
82 MR1 = RIGIDITE Model1 Mater1;
83 MR2 = RIGIDITE Model2 Mater2;
84 MR3 = RIGIDITE Model3 Mater3;
85 Mrigid = MR1 ET MR2 ET MR3;
86 *
87 CondL2 = BLOQUER UX (L4 ET L5 ET L8);
88 CondL1 = BLOQUER UY k8;
89 CondLtot = CondL1 ET CondL2;
90 FOR1 = FORCE(Forc 0.) k9;
91 Mrigid = Mrigid ET CondLtot;
92 *
93 Depl1 = RESOUD Mrigid For1 ;
94 *
95 UX1 = EXCO 'UX' depl1;
96 UY1 = EXCO 'UY' depl1;
97 Trace UX1 PoutU;
98 Trace UY1 PoutU;
99 *
100 def0 = DEFORMEE poutU Depl1 0.0 BLEU;
101 def1 = DEFORMEE poutU Depl1 ROUGE;
102 trace (def0 ET def1);
103 * ...

```

Il reste maintenant à avoir les résultats dans les repères locaux élémentaires, et voir ce qui se passe aux interfaces...

1.3.3 Résultats

Dans les listings ci-dessus, nous n'avons pas exposé comme atteindre les contraintes, car cela sera fait en TP.

Les résultats directement obtenus sont donnés à la figure 1.2 pour la déformée et les déplacements selon x et y , et à la figure 1.3 pour les déformations.

Néanmoins nous présentons quelques résultats issus de l'analyse des contraintes aux interfaces avec ou sans utilisation de la méthode de Reissner local. Nous nous concentrerons sur les résultats numériques et graphiques permettant d'apprécier la précision de la méthode de Reissner local par rapport aux résultats d'ANSYS aux points A , B , E et F ainsi que d'étudier l'influence de la valeur du rayon R .

Une première remarque s'impose : plus la valeur du rayon est faible, plus les résultats sont mauvais, quelle que soit la méthode employée (et vous devez savoir pourquoi à ce niveau du document).

Au point A , l'influence des conditions aux limites est encore sensible : en plus de la condition de symétrie de la structure, le blocage du déplacement vertical induit une légère détérioration des résultats numériques. L'influence de cette condition aux limites n'est plus visible au point B .

En A et B , la composante discontinue est σ_{xx} , les composantes σ_{yy} et σ_{xy} sont continues. Par contre, aux points E et F , c'est σ_{yy} la composante discontinue et σ_{xx} et σ_{xy} les composantes

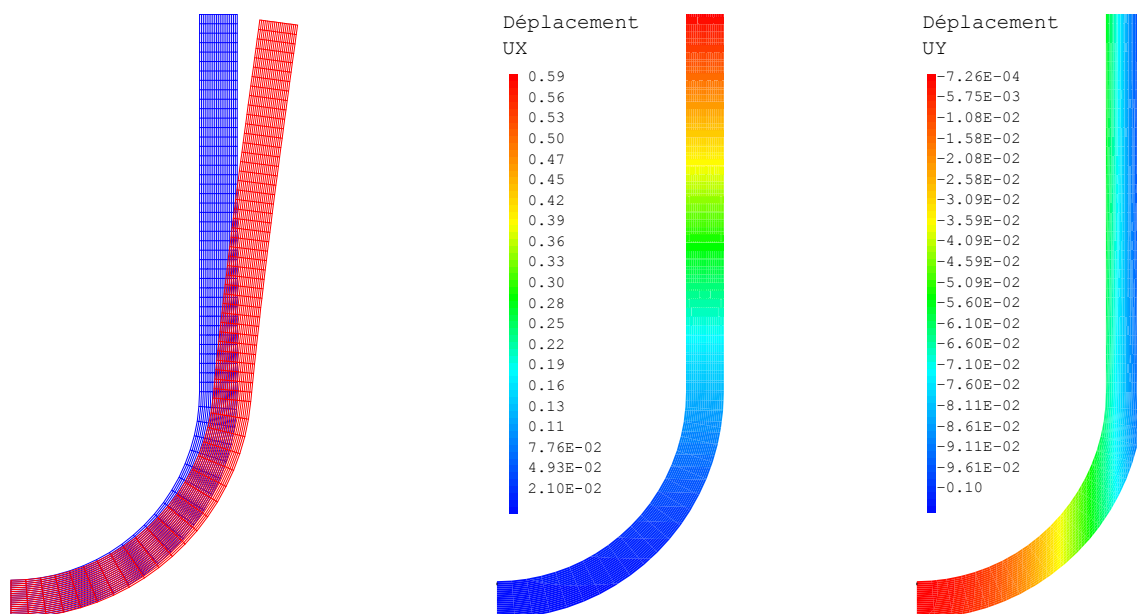


FIGURE 1.2 – Déformée et déplacements

continues.

D'après les résultats présentés ci-après, il est visible que la méthode de Reissner local permet d'améliorer les résultats par rapport à ANSYS.

Méthode	R (mm)	σ_{xx} peau (MPa)	σ_{xx} âme (MPa)	σ_{yy} (MPa)	σ_{xy} (MPa)
Ref	10	72,085	4,0121	1,5883	-0,0023339
Reiss	10	71,957	3,9670	1,5987	-0,0155720
ANSYS	10	71,957	3,9670	1,5116	0,0086231
Ref	8	66,500	3,8400	1,8556	-0,0022273
Reiss	8	66,665	3,7863	1,8888	-0,0118667
ANSYS	8	66,665	3,7863	1,7943	0,0111910
Ref	5	57,289	3,6598	2,6317	-0,0018770
Reiss	5	57,727	3,5771	2,6979	-0,0071024
ANSYS	5	57,727	3,5771	2,6066	0,0128210
Ref	3	49,551	3,6827	3,9304	-0,0014368
Reiss	3	49,454	3,5552	4,0110	-0,0046055
ANSYS	3	49,454	3,5552	3,9621	0,0106410
Ref	2	42,699	3,8289	5,4227	-0,0009727
Reiss	2	42,362	3,6593	5,4785	-0,0037443
ANSYS	2	42,362	3,6593	5,5264	0,0059470
Ref	1	27,171	4,3142	9,2850	-0,0004173
Reiss	1	25,876	4,0707	9,1122	-0,0021565
ANSYS	1	25,876	4,0707	9,6345	0,0005231

TABLE 1.1 – Résultats au point A

1.4 Sous-structuration et simulation multi-échelles

Le fonctionnement des produits industriels met en jeu des phénomènes physiques à des échelles très différentes. La prise en compte de tous ces phénomènes dans les simulations numériques demanderait d'utiliser des modèles extrêmement détaillés, entraînant des coûts d'identification et/ou de calcul prohibitifs. La simulation multi-échelles est une réponse à cette problématique.

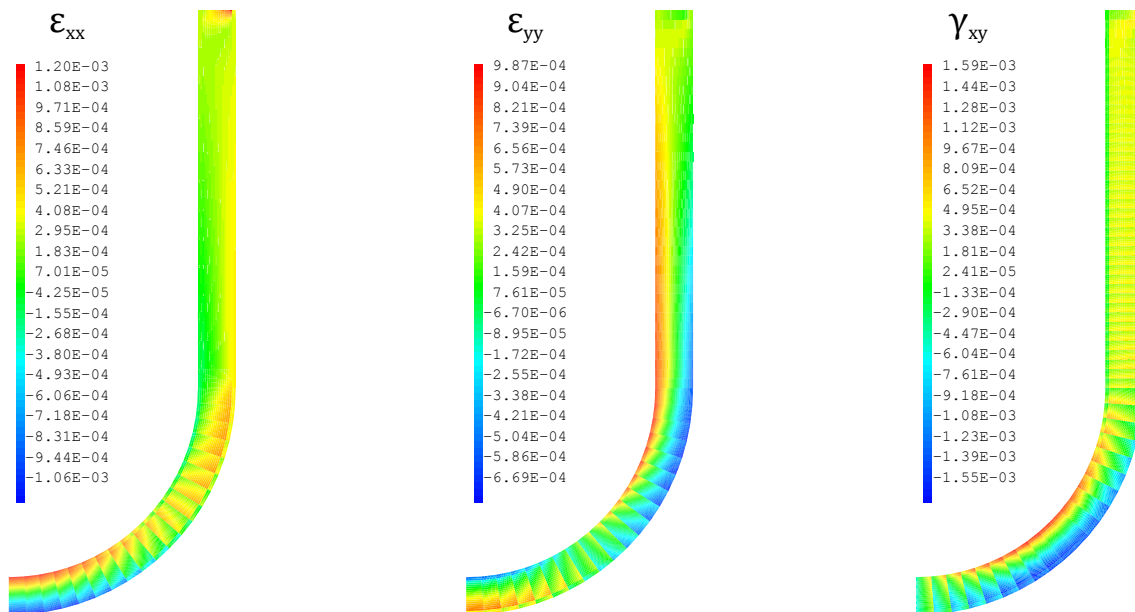


FIGURE 1.3 – Déformations

Elle consiste à simuler chaque phénomène à l'échelle la plus pertinente, i.e. en utilisant plusieurs modèles de tailles et de finesses différentes ; cela permet, grâce à des solveurs adaptés, de réaliser des simulations qui seraient inaccessibles par des approches plus directes.

Imaginons que nous souhaitions calculer un avion, mais que, pour des raisons évidentes de sécurité nous avons besoin d'obtenir une précision « boulon par boulon ». On se doute bien que l'on ne peut mailler tout l'avion avec un tel niveau de détail... On recourt alors à une cascade de modèles de dimensions et de finesses différentes, allant de l'avion tout entier (mais modélisé relativement grossièrement) à des détails structuraux modélisés très finement (mais limités à de petites zones), comme montré sur la figure 1.12.

La simulation est donc décomposée en différents niveaux, chacun représentant une échelle différente. Pour coordonner ces niveaux entre eux, on utilise généralement une approche dite descendante : on commence par simuler le comportement global de l'avion, puis les résultats sont utilisés pour déterminer les conditions aux limites appliquées sur les niveaux inférieurs. Toutefois, il est parfois également nécessaire de combiner ces approches descendantes à des approches ascendantes : les résultats des simulations fines sont utilisés pour construire des modèles de comportements plus grossiers.

De très nombreuses méthodologies multi-échelles existent et reposent toutes sur le fait de simuler chaque phénomène à l'échelle la plus pertinente. Pour cela, il est nécessaire :

1. de distinguer différentes échelles dans la modélisation et dans la simulation ;
2. de modéliser les relations existant entre ces différentes échelles.

Considérons un problème comportant deux échelles. Le point 1 évoqué ci-dessus se traduit par le fait de disposer de deux modèles distincts :

- Un modèle macroscopique

Il représente le produit et son environnement extérieur et est constitué d'une géométrie et d'un modèle de comportement relativement grossiers (puisque n'ayant pas vocation à représenter les phénomènes microscopiques) ;

- Un modèle microscopique

Il possède une description géométrique et un maillage suffisamment fins ainsi qu'un modèle de comportement détaillé. Il ne comprend qu'une ou quelques « cellules types » de petites dimensions.

Il est souvent possible, pour les matériaux notamment, de disposer d'hypothèses simplificatrices (telles que l'élasticité linéaire) permettant de se restreindre à une seule cellule type.

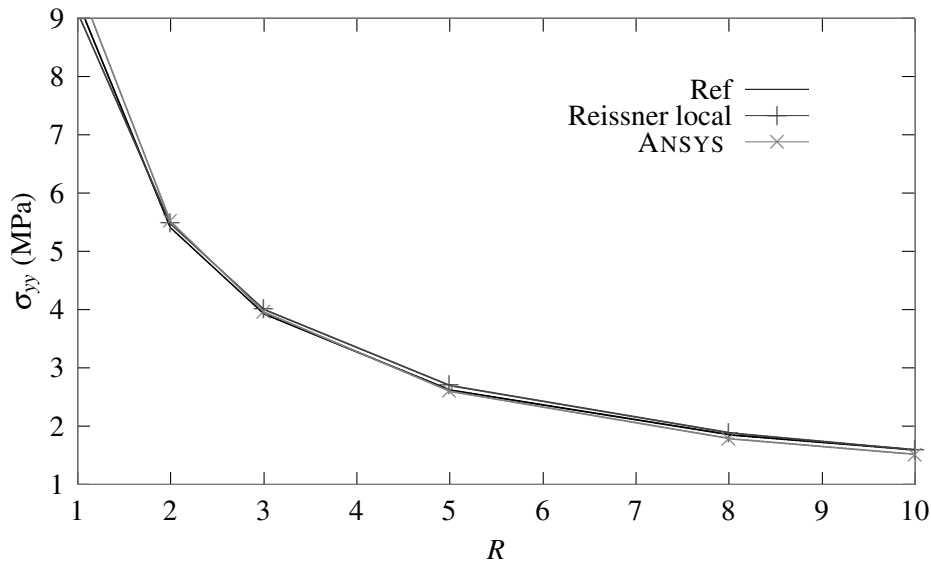


FIGURE 1.4 – σ_{yy} en A

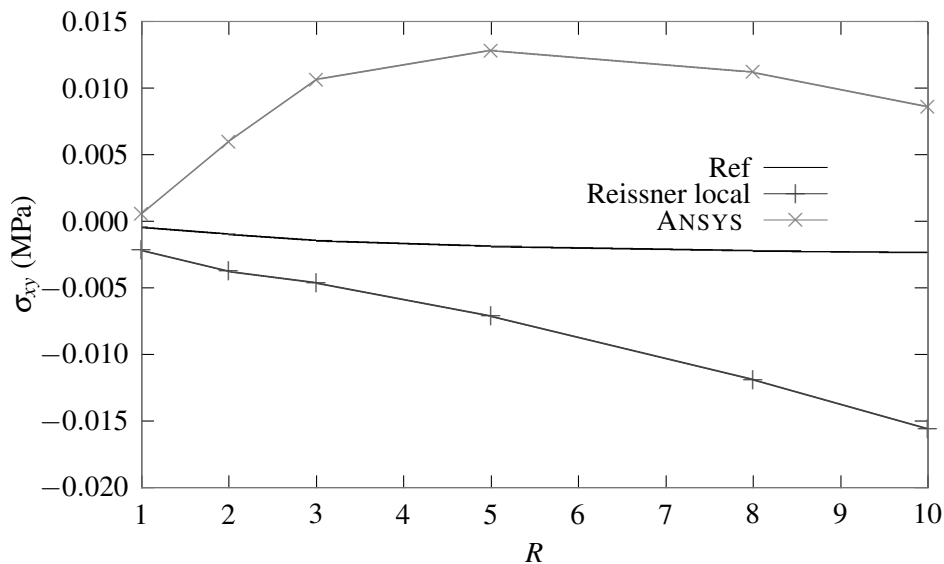


FIGURE 1.5 – σ_{xy} en A

Pour le cas des matériaux anisotropes constitués de plusieurs autres matériaux tels que les matériaux composites et fibreux, on se reportera au chapitre suivant sur l'homogénéisation. Toutefois, certaines fois on ne dispose pas de telles hypothèses et il faut multiplier le nombre de cellules (jusqu'à parfois couvrir tout le modèle macro !)

Le point 2 consiste donc à relier les deux échelles de modélisation.

En effet, les modèles macro et micro ne sont pas indépendants puisqu'ils modélisent la même physique à des échelles différentes. Il est donc nécessaire qu'ils soient cohérents l'un vis-à-vis de l'autre, en tout point et à chaque instant de la simulation. Pour cette raison, les modélisations multi-échelles comportent des couplages, i.e. des modèles d'interactions, entre les échelles.

Dans le cadre de la mécanique des solides déformables, on procède généralement comme suit :

- Le modèle de comportement macroscopique : qui modélise des phénomènes se produisant « au cœur du matériau », doit correspondre à la relation contraintes/déformations observée sur la cellule micro ;
- Le modèle de comportement microscopique : qui traduisent la façon dont la cellule est

Méthode	R (mm)	σ_{xx} peau (MPa)	σ_{xx} âme (MPa)	σ_{yy} (MPa)	σ_{xy} (MPa)
Ref	10	-72,248	-3,0668	1,4080	-0,0023570
Reiss	10	-72,542	-3,1193	1,2573	-0,0138800
ANSYS	10	-72,542	-3,1193	1,5464	-0,0198700
Ref	8	-67,723	-2,7905	1,5799	-0,0026011
Reiss	8	-68,001	-2,8457	1,4570	-0,0107700
ANSYS	8	-68,001	-2,8457	1,7111	-0,0197300
Ref	5	-61,079	-2,3199	2,0320	-0,0024099
Reiss	5	-61,335	-2,3813	1,9460	-0,0067001
ANSYS	5	-61,335	-2,3813	2,1535	-0,0186210
Ref	3	-56,823	-1,9080	2,6611	-0,0022154
Reiss	3	-57,072	-1,9756	2,5888	-0,0045050
ANSYS	3	-57,072	-1,9756	2,7738	-0,0176540
Ref	2	-54,772	-1,6200	3,2396	-0,0021234
Reiss	2	-55,026	-1,6902	3,1612	-0,0036124
ANSYS	2	-55,026	-1,6902	3,3438	-0,0172760
Ref	1	-52,491	-1,1687	4,2871	-0,0020797
Reiss	1	-52,777	-1,2391	4,1625	-0,0031088
ANSYS	1	-52,777	-1,2391	4,3710	-0,0174200

TABLE 1.2 – Résultats au point B

sollicitée par son environnement extérieur, doivent correspondre à l'état de contraintes ou de déformations macro.

Le modèle macro ayant une résolution beaucoup plus grossière, la cohérence des deux modèles ne peut donc pas se traduire par une correspondance exacte, point par point, des conditions aux limites ou du comportement. Pour cette raison, la plupart des approches multi-échelles postulent que les quantités macroscopiques doivent correspondre à des « moyennes » des quantités micro correspondantes, la définition mathématique exacte de cette moyenne variant fortement d'une approche à l'autre.

On peut donc dire que :

- *Le modèle de comportement macroscopique d'un élément de volume est choisi égal au comportement moyen de la cellule micro correspondante, calculé en utilisant une technique d'homogénéisation ;*
- *Les conditions aux limites micro sont appliquées en moyenne, car les champs de contraintes ou de déplacements macro sont beaucoup trop grossiers.*

Le problème est donc d'échanger des données pertinentes entre les différentes échelles, compte tenu des différentes résolutions des modèles.

Dans le sens micro vers macro, on utilisera les techniques d'homogénéisation dont il sera l'objet au chapitre suivant.

Dans le sens macro vers micro, il s'agit donc de spécifier des conditions aux limites à appliquer sur le bord des cellules micro, à partir d'une solution macro.

Les méthodes les plus simples se contentent d'imposer directement le champ de déplacements (ou de contraintes) macro comme condition aux limites. Le modèle macro ayant par définition une résolution beaucoup plus grossière que le modèle micro, il est incapable de capturer l'allure microscopique des déplacements ou des contraintes. Ces CL sont donc souvent trop imprécises par rapport aux finalités du modèle micro. Cela peut dégrader fortement la qualité des résultats et donc restreindre le domaine de validité de ces approches.

Pour palier ce problème, on écrit les CL micro de manière plus subtile en les décomposant en la somme d'un terme moyen et d'un terme de moyenne nulle. Ainsi, le champ micro u^m sera écrit $u^m = u^M + v^m$ où u^M est le champ macro (i.e. la moyenne du champ micro), et v^m est le reste (à

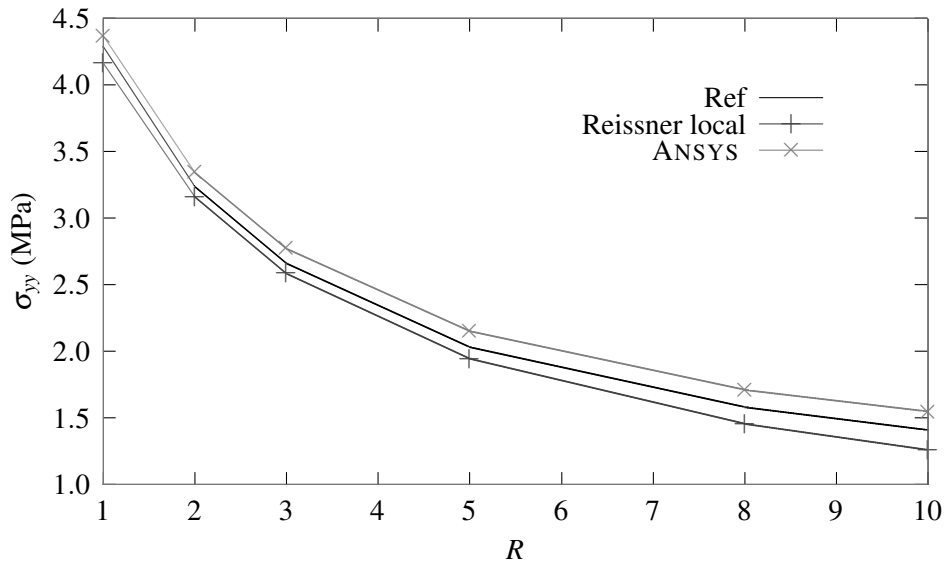


FIGURE 1.6 – σ_{yy} en B

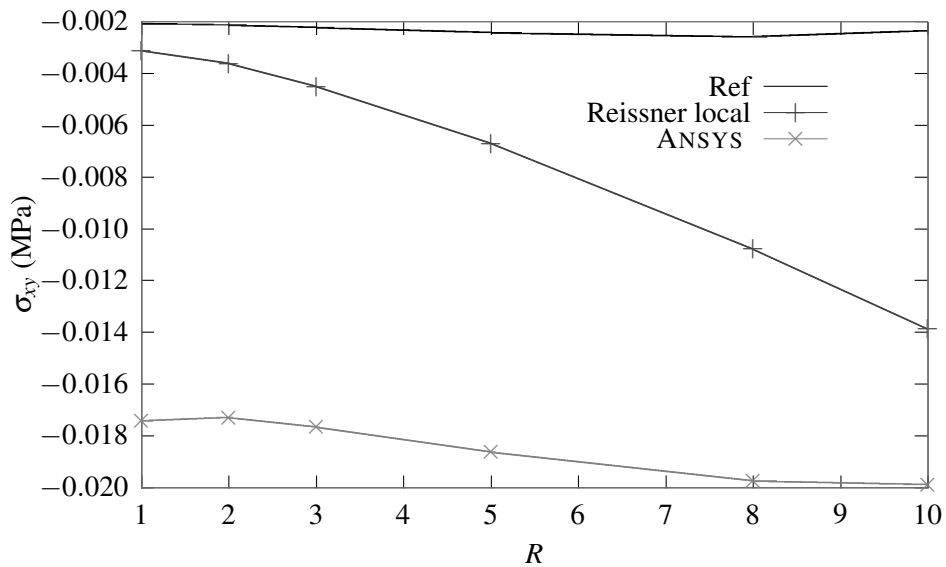


FIGURE 1.7 – σ_{xy} en B

moyenne nulle). Notre problème devient donc de déterminer le reste v^m , et c'est là que les méthodes divergent le plus. On distingue néanmoins deux grosses familles :

- Condition de périodicité

On postule que l'on connaît l'allure du reste. On peut alors enchaîner un calcul macro avec un solveur spécifique puis un calcul micro avec un solveur classique ;

- Couplage des cellules micro

Lorsqu'il n'est pas si simple de « séparer » les échelles, alors il faut résoudre en même temps les deux problèmes micro et macro avec échange de données.

Ces deux approches vont être un peu plus détaillées maintenant.

1.4.1 Condition de périodicité – méthodes multi-niveaux

Souvent, les méthodes multi-niveaux sont basées sur des conditions de périodicité inspirées de l'homogénéisation périodique. Ces conditions supposent que la partie micro du champ de déplacement, v^m , est périodique, i.e. est égale sur chaque paire de faces opposées de la cellule type considérée. La

Méthode	R (mm)	σ_{yy} peau (MPa)	σ_{yy} âme (MPa)	σ_{xx} (MPa)	σ_{xy} (MPa)
ref	10	46,0340	2,410200	0,57548	0,77518
Reiss	10	46,0265	2,397450	0,48986	0,71342
ANSYS	10	46,0265	2,397450	0,62529	0,80572
ref	8	45,8625	2,443000	0,68465	0,83563
Reiss	8	45,8505	2,421450	0,62992	0,76781
ANSYS	8	45,8505	2,421450	0,71910	0,90760
ref	5	45,4490	2,539450	1,00620	1,00390
Reiss	5	45,5025	2,499550	1,00810	0,91559
ANSYS	5	45,5025	2,499550	0,94099	1,19800
ref	3	44,5760	2,694400	1,56820	1,25790
Reiss	3	45,0015	2,644550	1,62260	1,10690
ANSYS	3	45,0015	2,644550	1,30760	1,64980
ref	2	43,1510	2,856600	2,23920	1,50420
Reiss	2	44,1375	2,805000	2,32600	1,25870
ANSYS	2	44,1375	2,805000	1,77080	2,12250
ref	1	37,6330	3,184100	4,02110	1,90380
Reiss	1	40,1735	3,143850	4,04390	1,41220
ANSYS	1	40,1735	3,143850	3,04360	3,15350

TABLE 1.3 – Résultats au point *E*

même hypothèse est formulée sur les contraintes. On obtient ainsi un ensemble de CL permettant de déterminer entièrement la solution micro à partir d'une déformation (ou d'une contrainte) macro imposée.

Le résultat micro ainsi obtenu est pertinent à deux conditions : 1) il faut que la microstructure soit effectivement périodique, et 2) que le principe de Saint-Venant s'applique, i.e. que l'on se trouve suffisamment loin de la surface de la pièce (y compris des détails géométriques tels que des trous, des fissures...). Dans le cas contraire, des effets de bord peuvent affecter l'allure de la solution, qui n'est alors plus périodique : il faut donc recourir à d'autres modélisations.

La simulation multi-niveaux fait appel à deux types de modèles, chacun équipé de son solveur :

- un modèle macro ne possédant pas de relation de comportement du matériau prédéfinie, équipé d'un solveur EF modifié ;
- un ensemble de modèles micro ne possédant pas de CL prédéfinies, équipés de solveurs EF classiques.

En fait, le solveur EF modifié est un solveur EF classique avec une petite différence : à chaque fois que le solveur macro a besoin du comportement d'un élément de volume quelconque, il envoie l'état de déformation macro de cet élément au solveur micro. Ce dernier a alors toutes les données nécessaires pour simuler numériquement son comportement à l'échelle microscopique. Il renvoie alors l'état de contraintes macro, selon la décomposition mentionnée ci-dessus. Les cellules micro étant pour ainsi dire « indépendantes », il est possible de recourir à des calculateurs parallèles.

Le domaine de validité de ces méthodes est bon dès que l'on sait faire des hypothèses réalistes sur l'allure de la solution micro. Ce n'est pas toujours le cas : par exemple, la fissuration, lorsqu'elle sort d'un cadre microscopique pour atteindre un cadre macroscopique, se prête très mal à cet exercice. Le cas de la fissuration sera traité dans un chapitre ultérieur de ce document.

En résumé :

Les méthodes multi-niveaux abordent la simulation à l'échelle macro et se « nourrissent » du comportement simulé à l'échelle microscopique.

Dans les méthodes multi-niveaux, l'emploi d'hypothèses de périodicité se traduit par des « sauts » de contraintes et de déplacements d'une cellule à l'autre ; si les échelles sont mal séparées, ces sauts sont non négligeables. Ne correspondant a priori pas à la physique, ils traduisent un

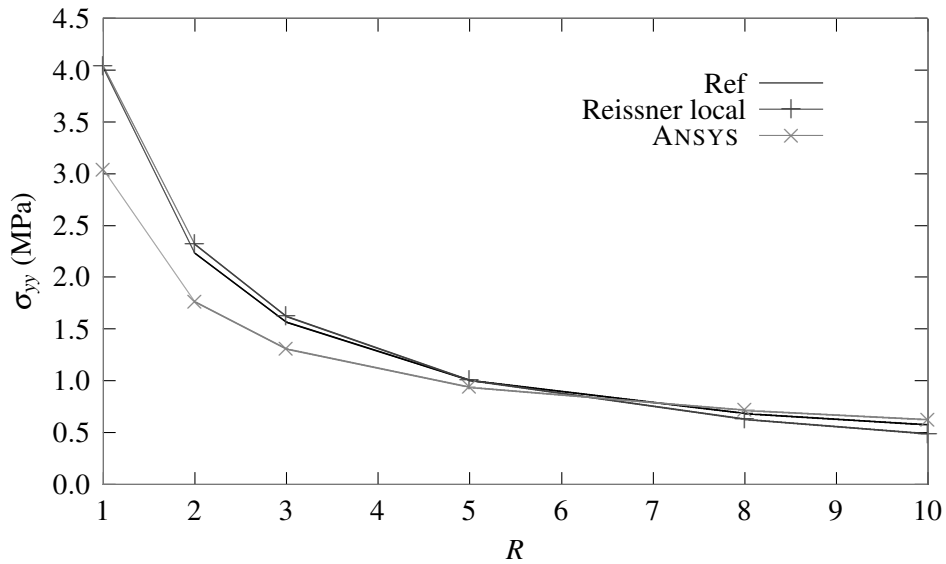


FIGURE 1.8 – σ_{xx} en E

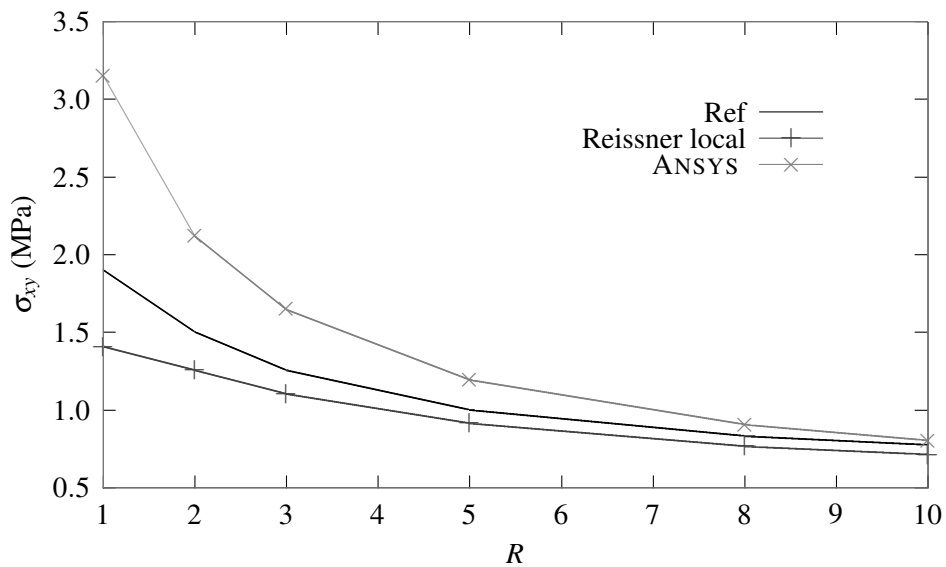


FIGURE 1.9 – σ_{xy} en E

écart avec la réalité.

1.4.2 Couplage des cellules micro – méthodes de décomposition de domaine

Les méthodes multi-niveaux présentées au paragraphe précédent sont mises à mal lorsque le comportement micro « déborde » un peu sur le comportement macro, i.e. lorsque les échelles ne sont pas suffisamment bien séparées. On recourt alors aux méthodes de décomposition de domaines, dont la validité est plus large, mais qui sont plus complexes.

Puisque nous ne pouvons plus supposer une allure de la solution micro, nous n'allons utiliser que les seules connaissances disponibles a priori : la continuité du champ de déplacements ainsi que la vérification par les contraintes du principe d'action-réaction. Il « suffit » donc d'écrire qu'à l'interface entre deux cellules micro, il y a égalité des déplacements et nullité de la somme des traces des contraintes (on retrouve ce que nous avons déjà plusieurs fois évoqué avec l'état des contraintes à l'interface entre deux matériaux différents).

La présence de couplages entre les cellules micro (qui en quelque sorte correspond à une

Méthode	R (mm)	σ_{yy} peau (MPa)	σ_{yy} âme (MPa)	σ_{xx} (MPa)	σ_{xy} (MPa)
ref	10	-47,438	-2,1425	0,54431	0,30073
Reiss	10	-47,510	-2,1533	0,39416	0,38936
ANSYS	10	-47,510	-2,1533	0,65814	0,30000
ref	8	-47,4555	-2,11415	0,62480	0,24743
Reiss	8	-47,5605	-2,12825	0,50447	0,33138
ANSYS	8	-47,5605	-2,12825	0,72921	0,21663
ref	5	-47,415	-2,03645	0,83167	0,10558
Reiss	5	-47,5335	-2,05940	0,76054	0,18360
ANSYS	5	-47,5335	-2,05940	0,87666	0,00251
ref	3	-47,223	-1,92305	1,11940	-0,10042
Reiss	3	-47,1885	-1,95170	1,07760	-0,01314
ANSYS	3	-47,1885	-1,95170	1,05090	-0,28747
ref	2	-46,940	-1,81445	1,38480	-0,29901
Reiss	2	-46,6760	-1,84130	1,35230	-0,18934
ANSYS	2	-46,6760	-1,84130	1,21420	-0,55238
ref	1	-46,1935	-1,61030	1,85860	-0,67358
Reiss	1	-45,4620	-1,62185	1,81270	-0,50497
ANSYS	1	-45,4620	-1,62185	1,52280	-1,03400

TABLE 1.4 – Résultats au point F

généralisation de la condition de périodicité du paragraphe précédent) change complètement le déroulement de la simulation par rapport aux méthodes multi-niveaux vues au paragraphe précédent. En effet, la prise en compte de ces couplages implique d'échanger directement des données entre les différents solveurs micro, qui ne sont plus « indépendants ». En contrepartie, cela permet de propager une information fine sur l'ensemble de la pièce et, ainsi, de se passer de l'hypothèse de séparation des échelles : il n'est plus nécessaire de modéliser séparément les phénomènes micro et macro.

Concrètement, les méthodes de décomposition de domaine sont des solveurs, qui ont généralement un fonctionnement multi-échelles. Elles partent d'un modèle micro du produit décomposé en sous-structures, et consistent à coupler les sous-structures en échangeant des contraintes et des déplacements sur les interfaces. Des versions plus ou moins simples existent. Notons bien qu'il n'y a pas de modèle macro dans une telle approche.

En résumé :

Dans la décomposition de domaine, la simulation est abordée à l'échelle la plus fine : la sous-structuration et le problème grossier ne sont utilisés que pour améliorer l'efficacité de la résolution.

Une simulation par décomposition de domaine conduit toujours à un champ de déplacement micro continu sur toute la structure, et un champ de contraintes micro équilibré (au sens des EF) sur toute la structure.

Actuellement, la simulation multi-échelles est encore relativement peu répandue dans le monde de l'ingénierie. Elle représente en effet un changement considérable par rapport aux pratiques usuelles de simulation ; de plus, la plupart des logiciels de calcul multi-échelles sont des outils développés par des chercheurs, qui n'ont pas encore l'ergonomie et la robustesse des solveurs utilisés dans l'industrie. Les industriels attendent donc l'apparition d'outils mieux adaptés à leurs problématiques avant d'envisager une utilisation plus fréquente de ces méthodes.

Cependant, à plus long terme, la simulation multi-échelles suscite un intérêt considérable dans l'industrie : en rendant accessibles à la simulation des phénomènes qui ne peuvent actuellement être étudiés qu'expérimentalement, elle constitue un pas important en direction du « virtual testing ». Pour cette raison, elle fait toujours l'objet de nombreux projets de recherche. Ceux-ci concernent

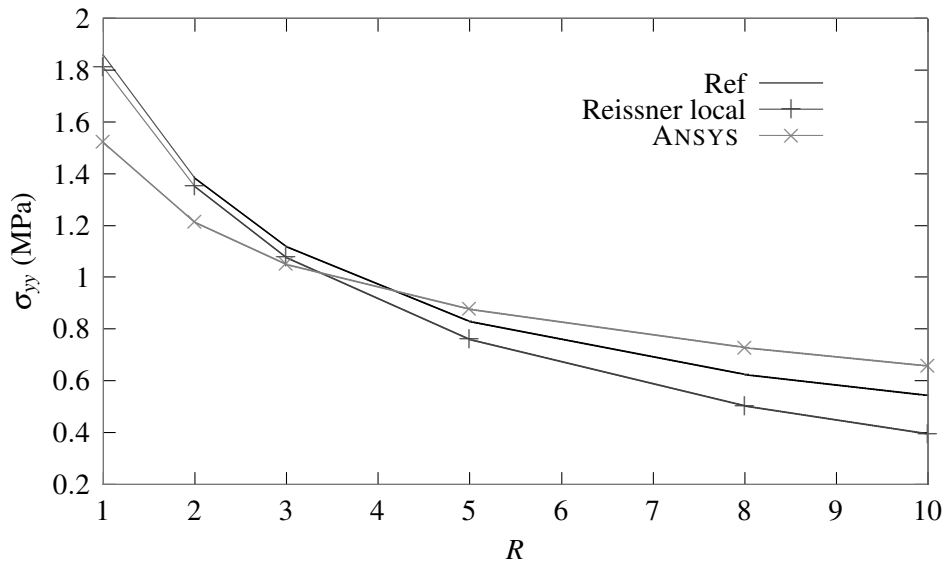


FIGURE 1.10 – σ_{xx} en F

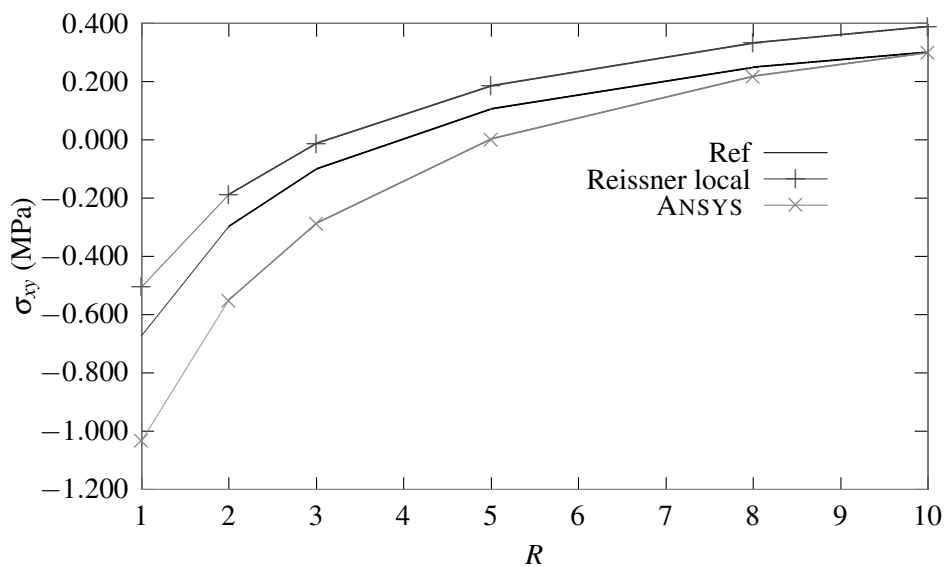


FIGURE 1.11 – σ_{xy} en F

aussi bien la modélisation, avec notamment la mise au point de « matériaux virtuels » (qui ne sont rien d'autre que des modèles multi-échelles, notamment pour les matériaux composites), que les solveurs qui sont en constante évolution.

1.5 Super-éléments

Dans ce paragraphe, nous allons parler du concept de super-élément, qui n'est qu'une application de ce qui a été présenté au paragraphe précédent.

Un super-élément est un groupement d'éléments qui, après assemblage, peuvent être vus comme un élément individuel du point de vue du calcul. Cet assemblage peut être requis pour des raisons de modélisation ou de calcul.

Pour constituer un super-éléments, les éléments groupés ne peuvent être pris au hasard. Ils doivent au moins constituer une « structure » en eux-même, mais d'autres conditions sont nécessaires qui seront détaillées plus loin.

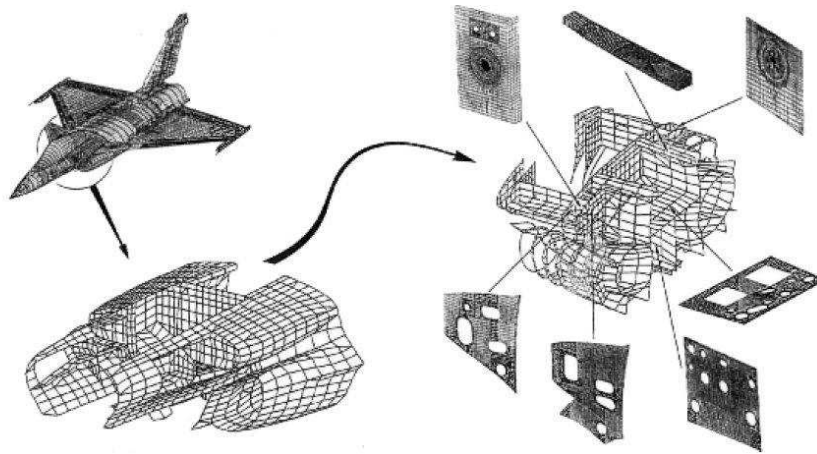


FIGURE 1.12 – Sous-structuration en aéronautique

Comme nous l'avons dit au paragraphe précédent, il y a deux voies duales pour considérer ce processus.

L'approche descendante consiste à considérer un super-élément comme constitué d'un ensemble d'éléments. On parle alors de macro-élément. L'approche ascendante consiste à considérer un super-élément comme un sous-ensemble d'une structure complète. On parle alors de sous-structure.

Finalement, quand parle-t-on de sous-structure ou de macro-élément ? En fait, il n'y a pas de règle, et le terme générique de super-élément couvre tout le spectre depuis l'élément individuel jusqu'à la structure complète.

Histoire

Originellement introduit dans l'aéronautique dans les années 60 (d'où le schéma au paragraphe précédent), le concept de sous-structuration répondait à trois motivations :

- faciliter la division du travail : des sous-structures avec des fonctions différentes (fuselage, ailes, train d'atterrissage...) pouvaient être traitées par des groupes d'experts différents. Chaque groupe pouvait à loisir améliorer, raffiner... sa partie tant que l'interface avec les autres parties restait inchangée.
- profiter de la répétition : en remarquant qu'une même structure peut contenir plusieurs sous-structures identiques, il est possible de diminuer le temps d'étude (par exemple symétrie des ailes...)
- contourner les limitations des ordinateurs : les ordinateurs de l'époque atteignaient vite leurs limites (par exemple en terme de taille mémoire). Diviser une structure complexe, que l'on était incapable de calculer en une seule fois, permettait de sauvegarder des résultats sous-structure par sous-structure puis d'effectuer « l'assemblage » des résultats.

Si les deux premiers points sont toujours d'actualité, le troisième l'est moins, surtout depuis le recourt aux algorithmes parallèles.

C'est d'ailleurs le développement de procédures pour le calcul parallèle qui a conduit les mathématiciens appliqués au concept de sous-domaines, alors qu'ils devaient grouper des éléments pour des raisons de calcul.

1.5.1 Condensation statique

En tant qu'assemblage de plusieurs éléments, un super-élément possède :

- des degrés de liberté internes : qui ne sont pas connectés à des éléments n'appartenant pas au super-élément considéré. Les nœuds ayant des ddl internes sont dits nœuds internes.
- des degrés de liberté aux frontières : qui sont connectés à au moins une entité (élément, super-élément) n'appartenant pas au super-élément considéré.

L'opération consistant à éliminer tous les ddl internes est appelée condensation statique ou condensation. Regardons ce qui se passe d'un point de vue matriciel. Pour cela, considérons que

nous ayons à résoudre le système :

$$\mathbf{K}\mathbf{q} = \mathbf{F} \quad (1.1)$$

Le vecteur \mathbf{q} se compose des composantes internes \mathbf{q}_i et des composantes de frontière \mathbf{q}_b , de sorte que le système, une fois réordonné s'écrit :

$$\begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix} \begin{pmatrix} \mathbf{q}_b \\ \mathbf{q}_i \end{pmatrix} = \begin{pmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{pmatrix} \quad (1.2)$$

Si la matrice \mathbf{K}_{ii} n'est pas singulière, alors la seconde équation peut être résolue en terme de variables internes en :

$$\mathbf{q}_i = \mathbf{K}_{ii}^{-1} (\mathbf{f}_i - \mathbf{K}_{ib}\mathbf{q}_b) \quad (1.3)$$

En reportant cela dans la première équation, on obtient le système avec matrice de rigidité condensée :

$$\tilde{\mathbf{K}}_{bb}\mathbf{q}_b = \tilde{\mathbf{f}}_b \quad (1.4)$$

avec :

$$\tilde{\mathbf{K}}_{bb} = \mathbf{K}_{bb} - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib} \quad \text{et} \quad \tilde{\mathbf{f}}_b = \mathbf{f}_b - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{f}_i \quad (1.5)$$

Après condensation, on peut donc bien considérer le super-élément, d'un point de vue calculatoire, comme un élément individuel.

Notons que la matrice \mathbf{K}_{ii} n'est pas singulière si elle possède la condition de rang suffisant, i.e. si elle ne contient que des modes à énergie nulle correspondant aux modes rigides (cette condition a déjà évoquée à propos de la validation des éléments). Si cela n'est pas le cas, le super-élément est dit flottant, et peut quand même être traité (en utilisant les projecteurs et inverses généralisés, mais c'est un peu plus compliqué).

La condensation statique est une opération matricielle appelée inversion partielle ou élimination partielle ou pseudo-inversion.

1.5.2 Remonter aux ddl internes

Une fois le système condensé résolu, on obtient les valeurs aux nœuds internes en réutilisant la formule :

$$\mathbf{q}_i = \mathbf{K}_{ii}^{-1} (\mathbf{f}_i - \mathbf{K}_{ib}\mathbf{q}_b) \quad (1.6)$$

1.6 Pseudo-inversion et réanalyse

On peut être amené, notamment lors de phases de conceptions, à devoir considérer plusieurs problèmes « relativement proches » les uns des autres (i.e. tester plusieurs configurations). On est alors tenté d'utiliser tout ou partie de la première modélisation afin de réaliser les suivantes.

L'idée de la méthode de réanalyse est d'analyser le comportement d'une structure élastique par EF sans particulariser le système d'équations final par la prise en compte de conditions cinématiques.

On obtient alors une solution générale de ce système faisant intervenir une matrice de rigidité régularisée. La nature de cette matrice (somme d'une matrice bande et d'une matrice pleine) ne permet pas d'utiliser les méthodes les plus classiques et optimales de résolution, mais il est possible d'en développer d'autres permettant d'accéder à la quasi-inverse d'une matrice singulière semi-définie positive (tout en profitant de son caractère bande).

Il est alors possible, tout en modifiant les conditions cinématiques et les chargements appliqués, de procéder à des réanalyses qui consistent alors simplement à résoudre des systèmes dits secondaires de tailles très inférieures au système global (mais un surcoût a été « payé » initialement pour calculer la pseudo-inverse).

Des problèmes de contact avec ou sans frottement entre solides élastiques peuvent bénéficier de cette méthode, ainsi que la modélisation du comportement élastique incompressible.

Dans ce paragraphe sur la réanalyse, nous n'utiliserons pas la notation avec les crochets et les accolades pour les matrices et les vecteurs afin d'alléger l'écriture.

1.6.1 Modification du chargement uniquement

Restons sur le problème structurel correspondant au système matriciel :

$$\mathbf{K}\mathbf{q} = \mathbf{F} \quad (1.7)$$

où \mathbf{K} est de dimension $n \times n$.

Si l'on souhaite considérer plusieurs cas de chargement $\mathbf{F}_1, \dots, \mathbf{F}_k$, on peut soit résoudre k fois le système précédent, soit résoudre le système :

$$\mathbf{K}\bar{\mathbf{q}} = \bar{\mathbf{F}} \quad (1.8)$$

où $\bar{\mathbf{F}}$ est la matrice $n \times k$ des k vecteurs de chargement, et $\bar{\mathbf{q}}$ est la matrice $n \times k$ des k vecteurs solutions correspondants.

Cette méthode, la plus simple des méthodes de réanalyse, permet malgré tout d'économiser des opérations.

1.6.2 Modification de la matrice

Considérons maintenant le cas où ce n'est plus le chargement \mathbf{F} qui peut varier d'une analyse à l'autre, mais la matrice \mathbf{K} .

On est amené à chercher une solution du système :

$$(\mathbf{K} + \Delta\mathbf{K})\mathbf{q} = \mathbf{F} \quad (1.9)$$

en fonction de la solution du système non perturbé, i.e. à calculer $(\mathbf{K} + \Delta\mathbf{K})^{-1}$ en fonction des autres matrices. On rappelle que $\Delta\mathbf{K}$ est appelé perturbation de la matrice \mathbf{K} .

Certaines formules existent pour des modification mineures de la matrice \mathbf{K} , mais nous ne les présenterons pas.

1.6.3 Modification des conditions cinématiques

Nous ne considérons ici que des modifications des conditions cinématiques. Généralement, les conditions cinématiques sont prises en compte en supprimant ou en modifiant les équations de l'équilibre avant résolution, ce qui rend le système régulier. Nous avons vu que le système à résoudre (incluant les conditions cinématiques) revient à chercher le minimum de la forme quadratique :

$$M = \frac{1}{2} \mathbf{q}^T \mathbf{K} \mathbf{q} - \mathbf{q}^T \mathbf{F} \quad (1.10)$$

Notons que l'on peut écrire les p conditions cinématiques sous la forme : $\mathbf{C}\mathbf{q} = \boldsymbol{\delta}$ avec \mathbf{C} une matrice $p \times n$, et \mathbf{q} et $\boldsymbol{\delta}$ des vecteurs.

Nous avons également déjà vu que ces conditions peuvent être prises en compte par l'intermédiaire de p multiplicateurs de Lagrange $\boldsymbol{\lambda}$ dans la fonctionnelle précédente qui devient alors :

$$M^* = \frac{1}{2} \mathbf{q}^T \mathbf{K} \mathbf{q} - \mathbf{q}^T \mathbf{F} - \boldsymbol{\lambda}^T (\mathbf{C}\mathbf{q} - \boldsymbol{\delta}) \quad (1.11)$$

Le système à résoudre est symétrique, régulier, mais non défini-positif.

Pour pallier la lenteur des algorithmes disponibles pour la résolution numérique d'un tel système (méthode de Gauss ou de décomposition avec pivotage), il est préférable de régulariser le système.

Pour cela, on s'appuie sur la connaissance de la matrice \mathbf{K} : celle-ci est singulière d'ordre r , où r correspond aux mouvements de corps rigide, ou modes rigides. Cela veut également dire qu'elle possède $(n - r)$ valeurs propres strictement positives (et r valeurs propres nulles).

Il est évident qu'il faut au moins disposer de $p \geq n$ conditions aux limites cinématiques pour que le système puisse admettre une solution. C'est évidemment l'hypothèse que nous ferons (sinon le problème est mal posé).

Nous considérons la matrice \mathbf{R} de dimension $n \times r$ des r vecteurs propres de \mathbf{K} correspondant à la valeur propre nulle. Quitte à construire ces vecteurs (qui sont orthogonaux), nous les choisirons normés. On a alors :

$$\begin{cases} \mathbf{K}\mathbf{R} = \mathbf{0} \\ {}^T\mathbf{R}\mathbf{R} = \mathbf{I}_r \end{cases} \quad (1.12)$$

On introduit la matrice :

$$\mathbf{K}_\alpha = \mathbf{K} + \alpha \mathbf{R}^T \mathbf{R} \quad (1.13)$$

qui, pour $\alpha > 0$ est régulière car symétrique et définie-positive. Les colonnes de \mathbf{R} sont vecteurs propres de \mathbf{K}_α pour la valeur propre α .

Dans le système $\mathbf{K}\mathbf{q} = \mathbf{F}$, on remplace \mathbf{K} par $\mathbf{K}_\alpha - \mathbf{R}^T \mathbf{R}$, et en introduisant le fait que $\mathbf{K}_\alpha \mathbf{R}^T \mathbf{R} = \alpha \mathbf{R}^T \mathbf{R}$, on obtient finalement :

$$\mathbf{K}_\alpha (\mathbf{I}_n - \mathbf{R}^T \mathbf{R}) \mathbf{q} = \mathbf{F} \quad (1.14)$$

Si l'on change de variable en posant $\mathbf{v} = (\mathbf{I}_n - \mathbf{R}^T \mathbf{R}) \mathbf{q}$ (\mathbf{v} est la projection de \mathbf{q} sur l'orthogonal du noyau de \mathbf{K}), \mathbf{v} devient l'inconnue du système régularisé :

$$\mathbf{K}_\alpha \mathbf{v} = \mathbf{F} \quad (1.15)$$

et l'on retrouvera la solution du système initial \mathbf{q} par :

$$\mathbf{q} = \mathbf{v} + \mathbf{R}^T \mathbf{R} \mathbf{q} = \mathbf{K}_\alpha^{-1} \mathbf{F} + \mathbf{R}^T \mathbf{R} \mathbf{q} \quad (1.16)$$

On peut remarquer que \mathbf{K}_α^{-1} , que l'on note \mathbf{S}_α possède les mêmes valeurs propres que \mathbf{K}_α (et donc les mêmes que \mathbf{K} et $\mathbf{R}^T \mathbf{R}$). On est donc naturellement amené à décomposer \mathbf{S}_α comme :

$$\mathbf{S}_\alpha = \mathbf{S} + \frac{1}{\alpha} \mathbf{R}^T \mathbf{R} \quad (1.17)$$

avec $\mathbf{S}\mathbf{R} = \mathbf{0}$ et ${}^T\mathbf{R}\mathbf{S} = \mathbf{0}$.

C'est cette matrice \mathbf{S} que l'on appelle quasi-inverse de \mathbf{K} , car :

$$\mathbf{S}\mathbf{K} = \mathbf{K}\mathbf{S} = \mathbf{I} - \mathbf{R}^T \mathbf{R} \quad (1.18)$$

Elle est de dimension $n \times n$, symétrique, semi-définie positive (ses valeurs propres sont de même signe), admet les mêmes valeurs propres que \mathbf{K} et en particulier ceux de la valeur propre nulle dont \mathbf{R} est une base, mais elle ne possède pas de caractère bande.

Voici brossé, en quelques lignes, les idées principales de la méthode. Nous n'irons pas plus loin dans sa présentation.

On rappelle que l'intérêt de la méthode est qu'une fois une première étape consistant à intégrer les données relatives à la structure (géométrie, discrétisation, matériaux) réalisée, on peut alors effectuer autant de fois que nécessaire la seconde étape qui porte sur la prise en compte des chargements et conditions aux limites.

Notons qu'il est possible de modifier un peu la forme du système secondaire afin de pouvoir prendre en compte, lors de la seconde étape, des conditions plus complexes, comme des conditions mixtes par exemple...

1.7 Dérivées d'ordre supérieur

Au paragraphe précédent, nous avons vu comment, sur une structure donnée, il était possible de prendre en compte plusieurs chargements et conditions aux limites cinématiques sans avoir à refaire tout le calcul.

Dans le même état d'esprit, nous allons voir maintenant comment optimiser la forme d'une structure, sans refaire tous les calculs.

Considérons le cas d'une conception ayant pour but de déterminer la forme la plus adaptée selon certains critères (rigidité, déformée, contraintes, énergie transmise...). Chaque évaluation d'une fonction coût conduit à une analyse par EF. L'utilisation de dérivées par rapport à la géométrie, ou plus généralement par rapport à la fonction coût, permet de réduire ce nombre d'analyse.

En fait, peut-être contrairement à l'intuition, le calcul de ces dérivées est relativement peu coûteux ; il n'est pas difficile et peut être fait automatiquement. Les dérivées d'ordre supérieur d'une fonction coût peuvent en fait être calculées avec autant de précision que la fonction elle-même. Ainsi, en un seul calcul, il est possible d'obtenir un développement de Taylor de la fonction coût, et donc d'éviter de nombreuses analyses.

1.7.1 Dérivées par rapport à la géométrie

En plus de considérer un domaine borné Ω de \mathbb{R}^n , nous allons considérer une perturbation V de Ω , et nous noterons :

$$\Omega + V = (I + V)(\Omega); \quad V \in W^{1,\infty}(\Omega; \mathbb{R}^n) \quad (1.19)$$

Nous considérons la fonction coût $\mathbf{J}(\Omega, u_\Omega)$ choisie pour décrire le problème d'optimisation. Cette fonction dépend donc naturellement du domaine Ω et de la solution du problème sur ce domaine u_Ω . Nous allons donner un sens à la dérivée de la fonction coût par rapport aux variations du domaine.

Si l'on considère le cas simple $\mathbf{J}(\Omega, u) = \int_\Omega u$, alors il vient :

$$\frac{d\mathbf{J}}{d\Omega}(\Omega, u_v, V) = \int_\Gamma u_\Omega V \cdot n + \int_\Omega u'_{\Omega;v} \quad (1.20)$$

avec une fois encore n la normale extérieure de $\Gamma = \partial\Omega$, et :

$$u'_{\Omega;v} = \lim_{t \rightarrow 0} \frac{u_{\Omega+tV}(x) - u_\Omega(x)}{t}, \quad \forall x \in \Omega \quad (1.21)$$

La méthode de dérivation par transport conduit à :

$$\frac{d\mathbf{J}}{d\Omega}(\Omega, u_v, V) = \int_\Gamma u_\Omega \operatorname{div} V + \int_\Omega \dot{u}_{\Omega;v} \quad \text{avec } \dot{u}_{\Omega;v} = \lim_{t \rightarrow 0} \frac{u_{\Omega+tV} \circ (I + tV)(x) - u_\Omega(x)}{t}, \quad \forall x \in \Omega \quad (1.22)$$

On définit alors la dérivée de la fonction coût par rapport aux variations du domaine par :

$$\frac{\partial \mathbf{J}}{\partial \Omega}(\Omega, u, V) = \int_\Gamma u V \cdot n \quad (1.23)$$

La difficulté tient à ce que l'ensemble des domaines Ω ne constitue pas un espace vectoriel (les perturbations ne s'ajoutent pas, ou au moins ne sont pas associatives : $(\Omega + V) + W \neq (\Omega + W) + V$).

Toutefois, si l'on utilise un paramétrage du domaine, il est alors possible d'utiliser les outils classiques du calcul différentiel dans les espaces normés. On introduit alors le paramètre F par :

$$\mathbf{J}(F, u) = \mathbf{J}(F(\Omega), u \circ F^{-1}), \quad F \in W^{1,\infty}(\Omega; \mathbb{R}^n) \quad (1.24)$$

et l'on a alors $(F + V) + W = (F + W) + V$.

La dérivée partielle de J par rapport au paramètre F est définie sans ambiguïté. Dans notre exemple, cette dérivée, prise au point $F = I$ est :

$$\frac{\partial J}{\partial F}(I, u) \cdot V = \int_{\Omega} u \operatorname{div} V \quad (1.25)$$

On pourra alors nous faire remarquer que :

$$\frac{\partial \mathbf{J}}{\partial \Omega}(\Omega, u, V) \neq \frac{\partial J}{\partial F}(I, u) \cdot V \quad (1.26)$$

ce à quoi nous répondrons que c'est le prix à payer pour se ramener à un espace vectoriel. Cette démarche peut alors être généralisée aux ordres supérieurs.

1.7.2 Calcul des dérivées

Nous n'entrons pas dans le détail, mais en cours de calcul se pose la question intéressante : obtient-on le même résultat si l'on discrétise d'abord et dérive ensuite ? En fait, la réponse est oui, sous certaines conditions de régularité que nous ne mentionnerons pas dans le cadre de ce document.

Encore une fois, nous n'avons fait qu'effleurer le problème, juste pour « faire connaître » la méthode. Il nous semblait intéressant de présenter la notion de dérivée par rapport à la géométrie.

De telles méthodes sont d'ores et déjà implémentées dans certains codes de calculs. Leur intérêt devrait apparaître clairement au lecteur (nous l'espérons).