

Calcul efficient : qualité des résultats et efforts de calcul

13.1 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 modèles, 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 fonctions des indicateurs d'erreur, et donc sans impacter le nombre de degrés de liberté 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 $-hp-$** 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.

13.2 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.

Nous nous proposons ici de donner quelques exemples :

- Nous avons déjà abordé dans ce document le cas où, à partir des déplacements nodaux obtenus par un calcul éléments finis 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 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.

- Disposant d'un point de pression constant par élément, on peut calculer la pression en un point quelconque comme interpolation (linéaire ou non) à partir des points disponibles.
- À partir des données nodales dans un modèle unidimensionnel (poutre ou barre selon le cas), il est possible de remonter à la répartition des contraintes en un point quelconque de la structure 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.

- Dans de nombreux cas industriels, une condition de la conception est que la contrainte reste en tout point en-deçà de la limite élastique du matériau. Un calcul linéaire est donc tout à fait licite, et l'on en est fort aise... Néanmoins, il se peut que dans certaines zones locales, ce critère ne soit pas respecté. Comment faire alors pour obtenir des contraintes « améliorées » dans cette zone, i.e. des contraintes plus proches de ce qu'aurait donné un calcul non-linéaire (mais coûteux) ?

Il existe plusieurs méthodes. L'une d'elle est de procéder à une **correction plastique**. Nous présentons ici sommairement la méthode de Neuber, initialement proposée en 1961 [107], et aujourd'hui couramment utilisée. D'autres méthodes existent, telle que celle de Molski-Glinka [108], ou encore d'autres plus ou moins raffinées [109].

Dans le cas d'un chargement uniaxial, Neuber postule que le produit de la contrainte par la déformation ne dépend pas de l'écoulement plastique. Cette solution ne permet pas de prendre en compte la contrainte moyenne. C'est pourquoi d'autres auteurs, comme par exemple Chaudonneret et Culié [110, 111], ont proposé une autre méthode basée sur un changement de repère à chaque extremum du chargement. Pour un état de contrainte multiaxial, Plusieurs généralisations des travaux de Neuber ont été faites. Les meilleurs résultats sont obtenus en considérant les invariants de von Mises des tenseurs. C'est la forme qui a été adoptée dans la majorité des postulats de type Neuber pour des états de contrainte multiaxiaux. On pourra se reporter à la thèse de Herbland [109] paragraphe III.2 pour plus de détails.

Sous l'hypothèse que la zone plastifiée reste confinée, le chemin d'effort global de la pièce considérée n'est pas perturbé. On écrit alors, comme annoncé plus haut, que le produit de la contrainte par la déformation ne dépend pas de l'écoulement plastique, i.e. qu'il est constant, ce qui s'écrit :

$$\sigma \cdot \varepsilon = K$$

où K est une constante calculée pour la contrainte linéaire max calculée.

Si l'on ajoute à l'équation précédente la loi de comportement de Hooke : $\sigma = E\varepsilon$, alors on en déduit que :

$$K = \frac{\sigma^2}{E}$$

En revenant à notre calcul par éléments finis, cela se traduit par l'équation de l'**hyperbole de Neuber** sous la forme :

$$\sigma \cdot \varepsilon = \frac{\sigma_{FEM}^2}{E_{FEM}}$$

Les valeurs corrigées de la contrainte et de la déformation sont obtenues de l'intersection de l'hyperbole de Neuber avec la courbe d'écrouissage du matériau. Elles sont donc solution du système :

$$\begin{cases} \sigma \cdot \varepsilon = \frac{\sigma_{FEM}^2}{E_{FEM}} \\ \sigma = R_e + \frac{(\varepsilon - \varepsilon_y)(R_m - R_e)}{\varepsilon_u - \varepsilon_y} \end{cases} \quad \text{soit} \quad \begin{cases} \sigma \cdot \varepsilon = \frac{\sigma_{FEM}^2}{E_{FEM}} \\ \sigma = R_e + \frac{(\varepsilon - \frac{R_e}{E_{FEM}})(R_m - R_e)}{\varepsilon_u - \varepsilon_y} \end{cases} \quad (13.1)$$

où ε_u est la déformation à rupture (u = ultimate) et ε_y la déformation élastique (y = yield). Cette méthode est illustrée à la figure 13.1.

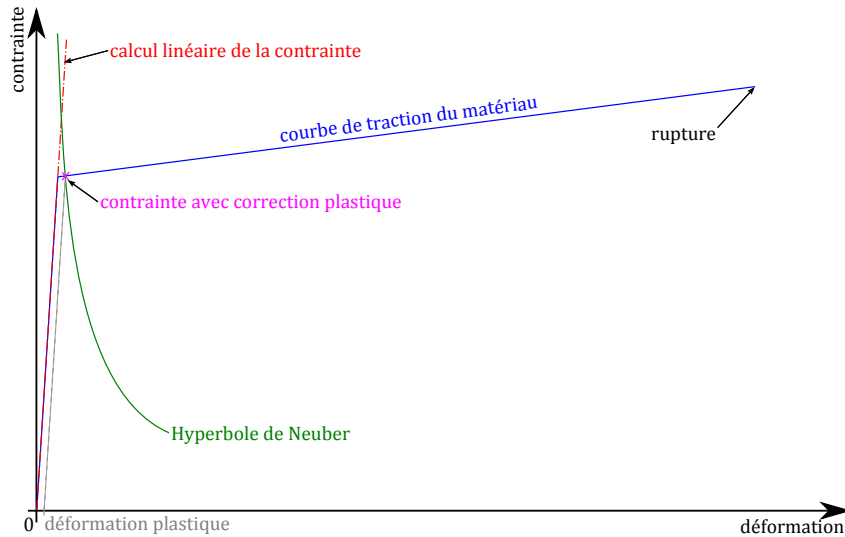


FIGURE 13.1: Méthode de Neuber

De plus, afin de se rassurer sur la validé de la méthode, on vérifie que la déformation plastique n'est pas excessive. En pratique, on demande généralement à ce qu'elle soit inférieure à 0.2%.

- Dans certains cas, on peut être amené à prendre en compte l'effet de la température de manière simpliste dans un calcul mécanique.

On remarque qu'un certain nombre de matériaux voient leurs propriétés élastiques varier relativement peu pour une plage raisonnable de température. Disposant d'un calcul « sans température » (i.e. avec les propriétés considérées à une température donnée, généralement l'ambiante), on peut ne faire varier que « l'admissible » du matériau pour la nouvelle température considérée. Les valeurs des contraintes sont donc comparées à ces nouveaux admissibles pour déterminer s'il y a un risque ou non.

Ces admissibles à la température T sont calculés par proportionnalité par rapport à la contrainte à rupture en traction à la température ambiante T_0 (aussi appelée contrainte ultime en traction, ou simplement R_m en français et F_{tu} en anglais) :

$$\text{Admissible}_T = \frac{R_{mT}}{R_{mT_0}} \cdot \text{Admissible}_{T_0} \quad (13.2)$$

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

Dans ce chapitre, nous présentons un manière d'implémenter la méthode « Reissner local » mentionnée aux paragraphes 13.2 et 12.3. 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 (8.52).

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 démontrons également qu'il est assez simple d'implémenter des fonctions dans un code existant.

13.3.1 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 nall
3 !
4 ! Parametres
5 *get,Nnoeuds,node,,count
6 *get,Nmater,mat,,count
7 *get,Nelem,elem,,count
8 *cfopen,temp,par
9 *vwrite,Nnoeuds
10 (E13.7,' ')
11 *vwrite,Nelem
12 (E13.7,' ')
13 *vwrite,Nmater
14 (E13.7,' ')
15 *cfclose
16 !
17 ! Matériaux
18 *cfopen,temp,mat
19 *dim,Mater,,Nmater,3
20 *do,I,1,Nmater
21   *get,Mater(I,1),Ex,I
22   *get,Mater(I,2),Ey,I
23   *get,Mater(I,3),Nuxy,I
24 *enddo
25 *vwrite,Mater(1,1),Mater(1,2),Mater(1,3)
26 (3(E13.7,' '))
27 *cfclose
28 !
29 ! Noeuds
30 *dim,CoordX,,Nnoeuds
31 *dim,CoordY,,Nnoeuds
32 *dim,CoordZ,,Nnoeuds
33 *vget,CoordX(1),node,,loc,x
34 *vget,CoordY(1),node,,loc,y
35 *cfopen,temp,coo
36 *vwrite,CoordX(1),CoordY(1)
37 (2(E13.7,' '))
38 *cfclose
39 !
40 ! Elements + numero du materiau
41 *dim,E1,,Nelem,9
42 *vget,E1(1,1),elem,1,node,1
43 *vget,E1(1,2),elem,1,node,2
44 *vget,E1(1,3),elem,1,node,3
45 *vget,E1(1,4),elem,1,node,4
46 *vget,E1(1,5),elem,1,node,5
47 *vget,E1(1,6),elem,1,node,6
48 *vget,E1(1,7),elem,1,node,7
49 *vget,E1(1,8),elem,1,node,8
50 *vget,E1(1,9),elem,1,attr,mat
51 *cfopen,temp,elt
52 *vwrite,E1(1,1),E1(1,2),E1(1,3),E1(1,4),E1
53   (1,5),E1(1,6),E1(1,7),E1(1,8),E1(1,9)
54 (9(E13.7,' '))
55 *cfclose
56 ! Resultats
57 *vget,CoordX(1),node,1,U,x
```

```

58 *vget,CoordY(1),node,1,U,y
59 *c fopen,temp,res
60 *vwrite,CoordX(1),CoordY(1)
61 (2(E13.7,' '))
62 *c fclose
63 !
64 ! Contraintes
65 *vget,CoordX(1),node,1,S,x
66 *vget,CoordY(1),node,1,S,y
67 *vget,CoordZ(1),node,1,S,xy
68 *c fopen,temp,str
69 *vwrite,CoordX(1),CoordY(1),CoordZ(1)
70 (3(E13.7,' '))
71 *c fclose
72 !
73 ! run locreiss.exe
74 /sys,'locreiss.exe'
75 !
76 ! Effacer les fichiers
77 ! generes par ANSYS:
78 ! - temp.par: parametres
79 ! - temp.coo: coordonnees nodales
80 ! - temp.elt: incidences
81 ! - temp.mat: proprietes materielles
82 ! - temp.res: déplacements nodaux
83 ! - temp.str: contraintes nodales
84 ! generes par locreiss.exe:
85 ! - temp.dum
86 ! - temp.tmp
87 ! Il reste les fichiers
88 ! - temp.sij: contraintes nodales modifiees
89 !
90 !/sys,'rm ./temp.par'
91 !/sys,'rm ./temp.coo'
92 !/sys,'rm temp.elt'

93 !/sys,'rm temp.mat'
94 !/sys,'rm temp.res'
95 !/sys,'rm temp.str'
96 !/sys,'rm temp.dum'
97 !/sys,'rm temp.tmp'
98 !
99 ! lire les valeurs dans temp.sij
100 ! ces fichiers contiennent: sxx, syy, sxy
101 ! ce sont toutes les composantes continues.
102 ! Elles n'ont pas toutes un sens:
103 ! Sxx discontinu, mais dans le repere
104 ! local, donc on la laisse pour qu'ANSYS
105 ! puisse faire la ! rotation de repere
106 ! si necessaire.
107 *vread,CoordX(1),temp,sxx,,
108 (E13.7)
109 *vread,CoordY(1),temp,syy,,
110 (E13.7)
111 *vread,CoordZ(1),temp,sxy,,
112 (E13.7)
113 *vput,CoordX(1),node,1,S,x
114 *vput,CoordY(1),node,1,S,y
115 *vput,CoordZ(1),node,1,S,xy
116 !
117 ! Cleanup
118 CoordX(1)=
119 CoordY(1)=
120 CoordZ(1)=
121 Mater(1,3)=
122 E1(1,9)=
123 Nnoeuds=
124 Nmater=
125 Nelem=
126 !
127 /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 But:
8 C 1. on relit les fichiers de ANSYS generes avec la macro INTERF
9 C 2. on determine les noeuds de post-traitement
10 C 3. on effectue un Reissner local aux interfaces
11 C
12 C WARNING:
13 C On ne traite que le cas de PLANE 82 avec une geometrie definie
14 C dans le plan (X,Y)
15 C
16 C MAIN of: VM2
17 C-----
18 C
19 PARAMETER(NODMAX=1000,NELTMAX=300,MATMAX=4)
20 C
21 COMMON /VMCTS/ COMPI,ZERO,ONE,IINP,IOUT,ICON,MONI
22 COMMON /NPB/ NNOD,NELT,NMAT,NCAL
23 C
24 C
25 DIMENSION COORD(NODMAX,3),DISP(NODMAX,2)
26 DIMENSION XMAT(MATMAX,3),NLT(NELTMAX,9)
27 DIMENSION NCALC(NODMAX,5),NADJ(NODMAX,8),NDT(NODMAX,8)
28 DIMENSION D1(3,3),D2(3,3)
29 C
30 C----- Presetings
31 MONI=0 ! monitoring level (debug)
32 ICON=6 ! standard output = screen
33 IINP=10 ! input file
34 IOUT=20 ! output file
35 ZERO=0.0D0 ! zero
36 ONE=1.0D0 ! one
37 COMPI=DACOS(-ONE) ! pi
38 C
39 C

```

```

40 C—— Read input files
41 CALL READANS (NODMAX , NELTMAX , MATMAX , COORD , XMAT , NLT , DISP)
42 C
43 C—— Find Adjacent elements
44 CALL CHELT (NODMAX , NELTMAX , NLT , NCALC , NADJ , NDT)
45 C
46 C—— Material properties
47 CALL MATER (MATMAX , XMAT , D1 , D2 , D3 , D4 , NODMAX , NADJ , NELTMAX , NLT)
48 C
49 C—— Computation: Local Reissner
50 CALL LOCREISS (NODMAX , NELTMAX , D1 , D2 , D3 , D4 ,
51 x          COORD , NLT , NADJ , DISP)
52 C
53 C—— Average multi-computed nodes
54 CALL AVERAGE (NODMAX , MULTI)
55 C
56 C—— Last line of VM2
57 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 (positions 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'Hellinger-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.

13.3.2 Poutre en U

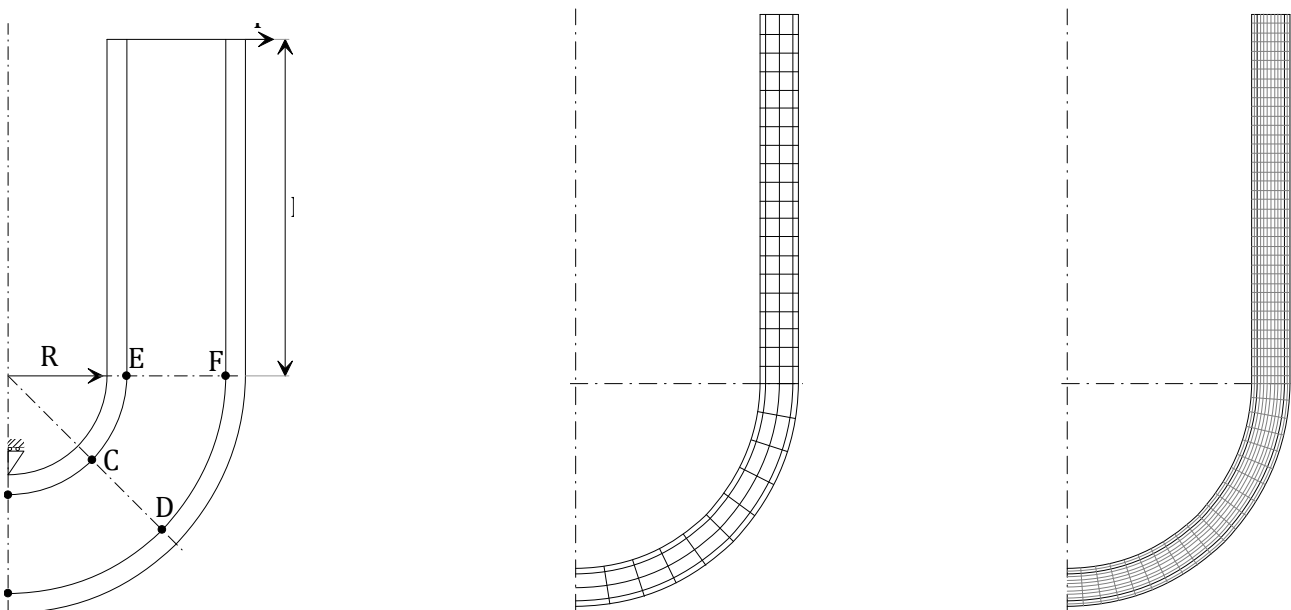


FIGURE 13.2: a) Géométrie de la poutre en U, b) le maillage utilisé pour le tester la méthode Reissner local, c) maillage permettant d'obtenir la « solution de référence » puisqu'il n'y a pas de solution analytique

Disposant de la macro précédente, nous allons l'appliquer au calcul de la poutre en U présentée à la figure 13.2a. 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 13.2a, 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 du même problème 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 ROUG;
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...

13.3.3 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 13.3 pour la déformée et les déplacements selon x et y , et à la figure 13.4 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 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.

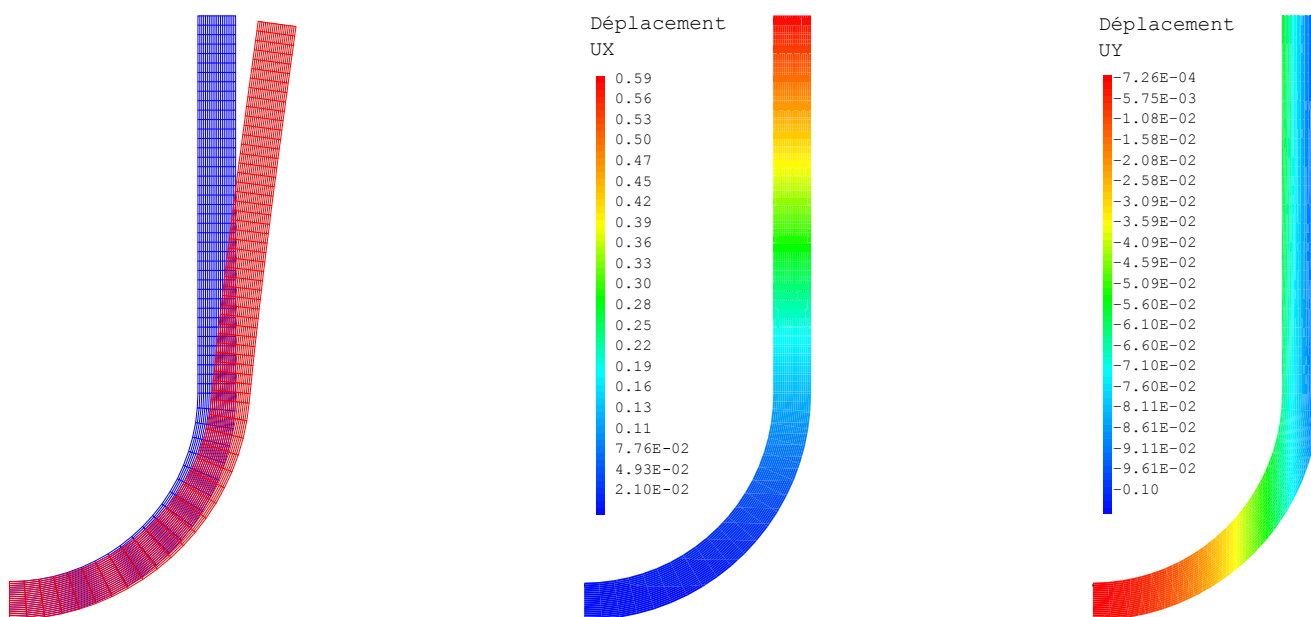


FIGURE 13.3: Déformée et déplacements

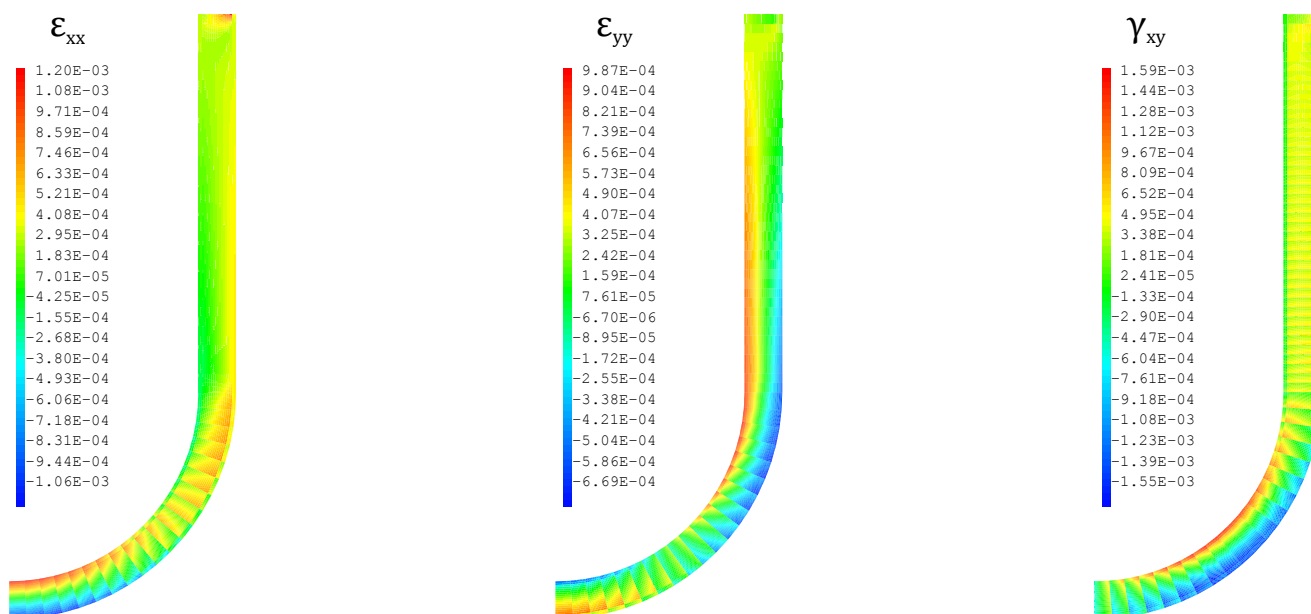


FIGURE 13.4: Déformations

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

Tableau 13.1: Résultats au point A

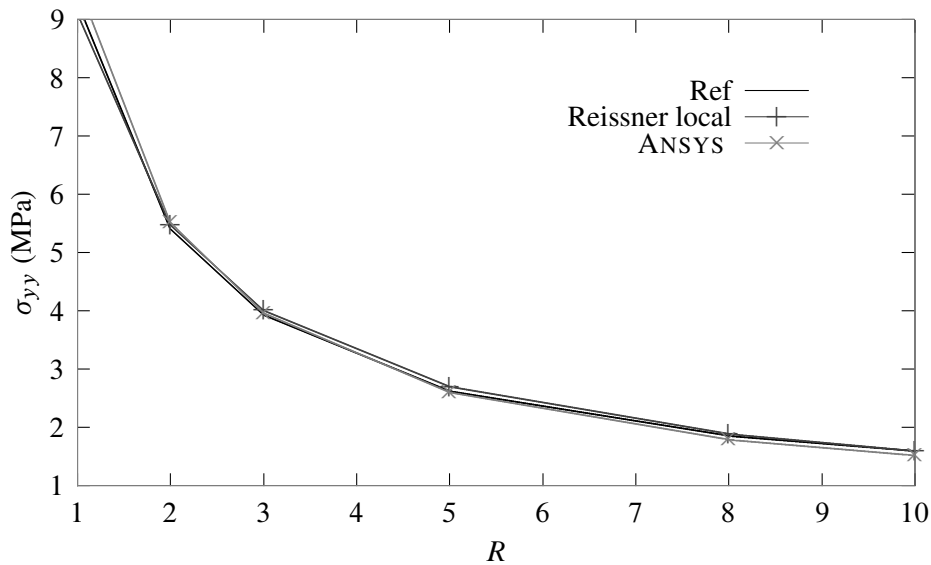


FIGURE 13.5: σ_{yy} en A

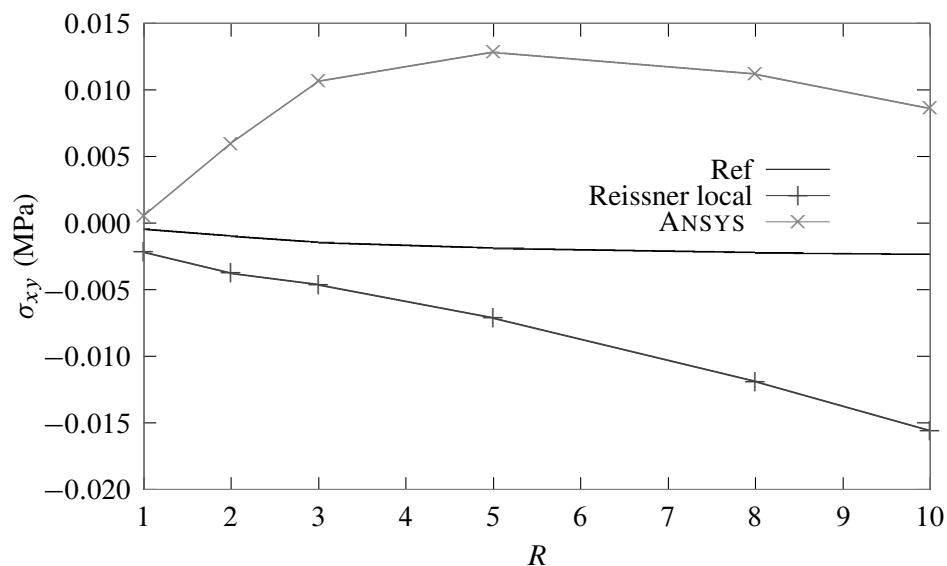


FIGURE 13.6: σ_{xy} en A

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

Tableau 13.2: Résultats au point B

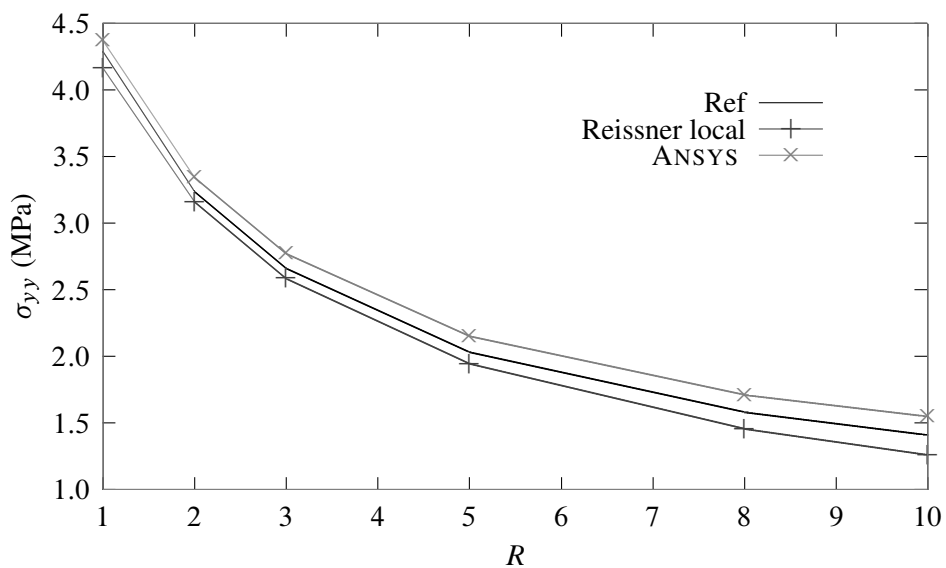


FIGURE 13.7: σ_{yy} en B

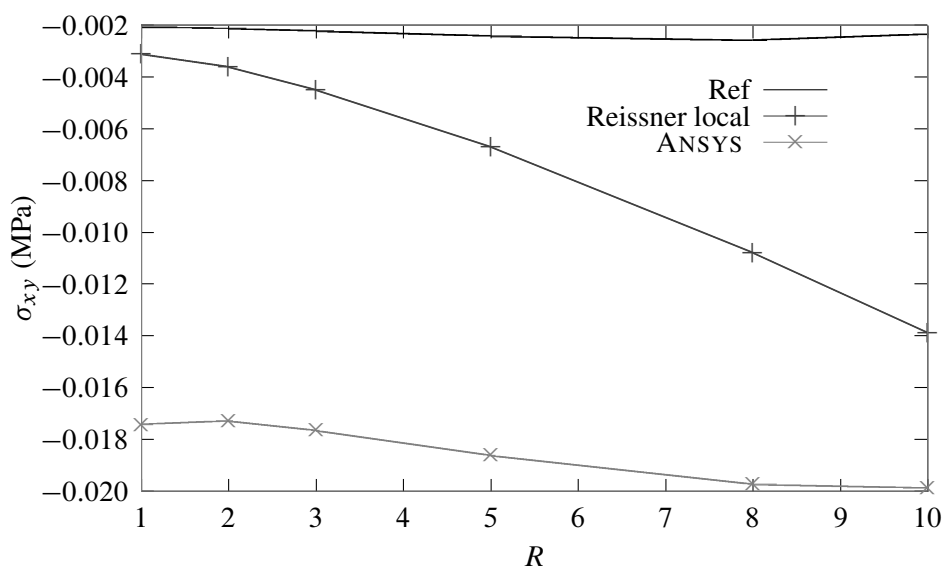


FIGURE 13.8: σ_{xy} en B

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

Tableau 13.3: Résultats au point E

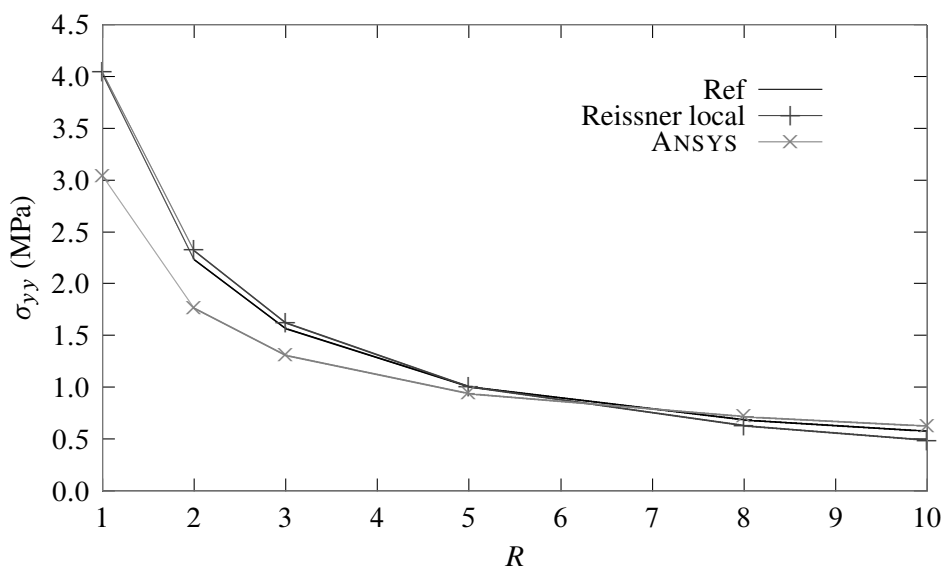


FIGURE 13.9: σ_{xx} en E

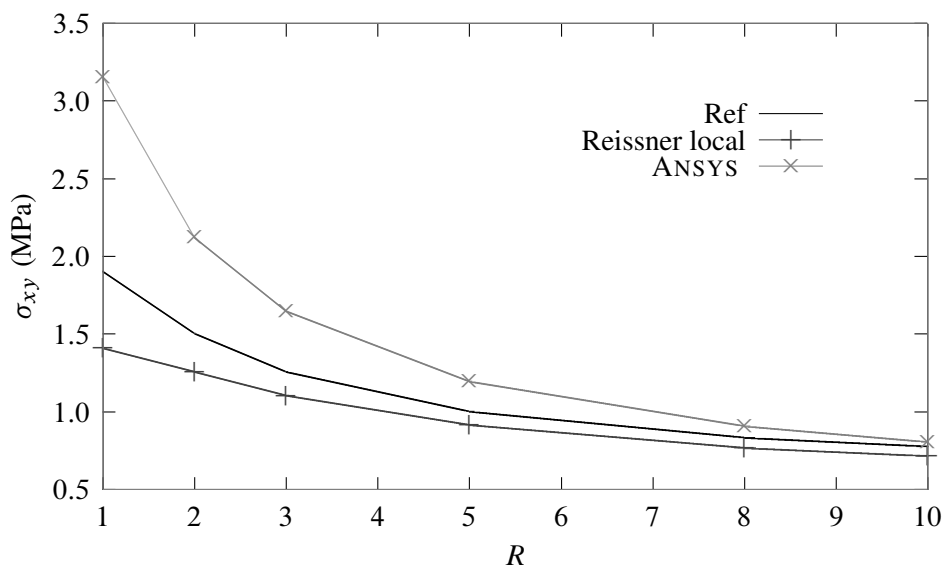


FIGURE 13.10: σ_{xy} en E

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

Tableau 13.4: Résultats au point F

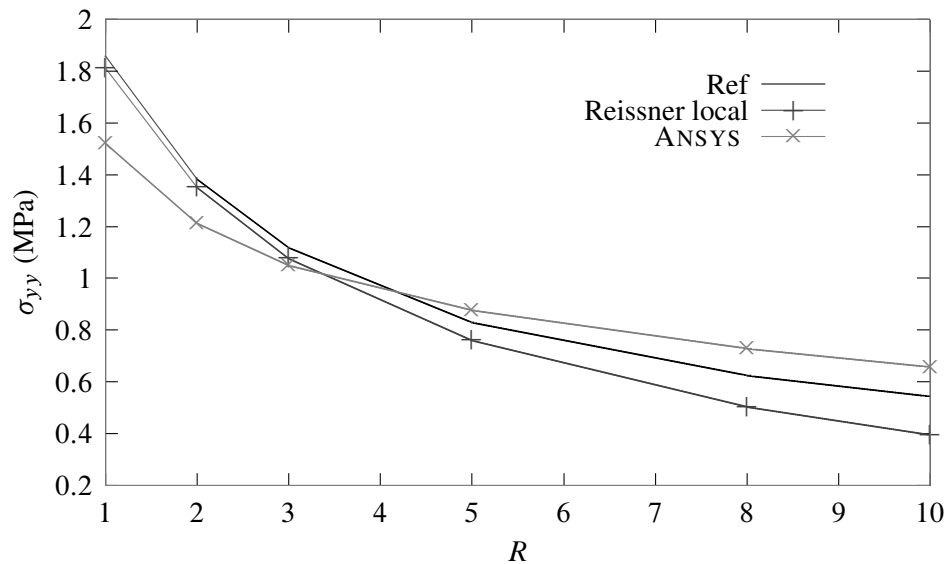


FIGURE 13.11: σ_{xx} en F

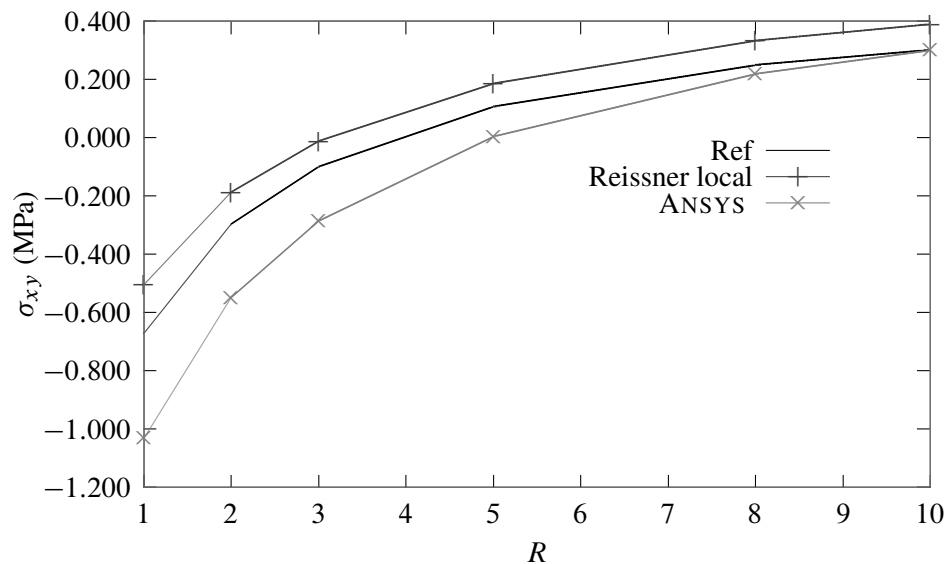


FIGURE 13.12: σ_{xy} en F

13.4 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. 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 ayons 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 13.13.

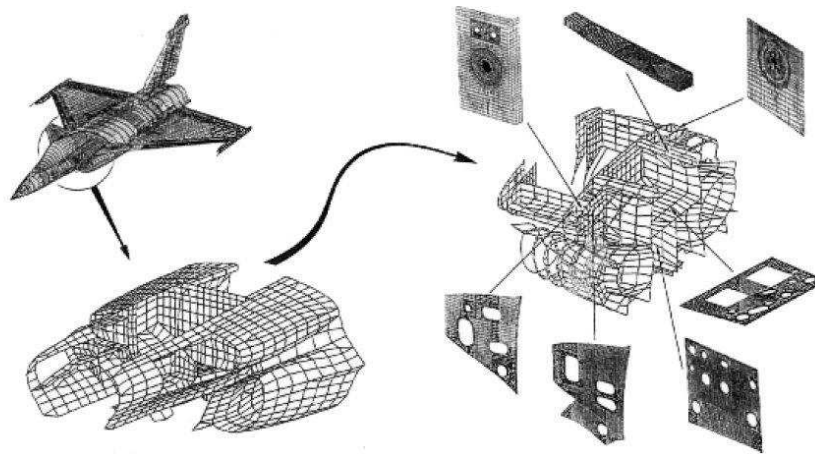


FIGURE 13.13: Sous-structuration en aéronautique

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 que soit modélisé le 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. Pour le cas des matériaux anisotropes constitués de plusieurs autres matériaux tels que les matériaux composites et fibreux, on se reportera au chapitre 15.

Toutefois, il peut arriver que l'on ne dispose pas de telles hypothèses et il soit alors nécessaire de 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 macroscopique et microscopique 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 microscopique ;
- Le modèle de comportement microscopique : qui traduit la façon dont la cellule est sollicitée par son environnement extérieur, doivent correspondre à l'état de contraintes ou de déformations macroscopique.

Le modèle macroscopique 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 microscopiques 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 microscopique correspondante, calculé en utilisant une technique d'homogénéisation ;
- Les conditions aux limites microscopiques sont appliquées en moyenne, car les champs de contraintes ou de déplacements macroscopiques 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 microscopique vers macroscopique, on utilisera les techniques d'homogénéisation dont il sera l'objet au chapitre 15.

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

Les méthodes les plus simples se contentent d'imposer directement le champ de déplacements (ou de contraintes) macroscopique comme condition aux limites. Le modèle macroscopique ayant par définition une résolution beaucoup plus grossière que le modèle microscopique, il est incapable de capturer l'allure microscopique des déplacements ou des contraintes. Ces conditions aux limites sont donc souvent trop imprécises par rapport aux finalités du modèle microscopique. 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 conditions aux limites microscopiques de manière plus subtile en les décomposant en la somme d'un terme moyen et d'un terme de moyenne nulle. Ainsi, le champ microscopique u^m sera écrit $u^m = u^M + v^m$ où u^M est le champ macroscopique (i.e. la moyenne du champ microscopique), et v^m est le reste (à 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 macroscopique avec un solveur spécifique puis un calcul microscopique avec un solveur classique ;

- **Couplage des cellules microscopiques**

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

Ces deux approches vont être un peu plus détaillées maintenant.

13.4.1 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 microscopique du champ de déplacement, v^m , est périodique, i.e. est égale sur chaque paire de faces opposées de la cellule type considérée. La même hypothèse est formulée sur les contraintes. On obtient ainsi un ensemble de conditions aux limites permettant de déterminer entièrement la solution microscopique à partir d'une déformation (ou d'une contrainte) macroscopique imposée.

Le résultat microscopique 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 macroscopique ne possédant pas de relation de comportement du matériau prédéfinie, équipé d'un solveur éléments fini modifié ;

- un ensemble de modèles microscopiques ne possédant pas de conditions aux limites prédéfinies, équipés de solveurs éléments finis classiques.

En fait, le solveur éléments finis modifié est un solveur éléments finis classique avec une petite différence : à chaque fois que le solveur macroscopique a besoin du comportement d'un élément de volume quelconque, il envoie l'état de déformation macroscopique de cet élément au solveur microscopique. Ce dernier dispose alors toutes les données nécessaires pour simuler numériquement son comportement à l'échelle microscopique. Il est donc capable de renvoyer l'état de contraintes macroscopiques, selon la décomposition mentionnée ci-dessus. Les cellules microscopiques 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 microscopique. 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 abordé au chapitre 21.

En résumé :

Les méthodes multi-niveaux abordent la simulation à l'échelle macroscopique 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 écart avec la réalité.

13.4.2 Couplage des cellules microscopiques – méthodes de décomposition de domaine

Les méthodes multi-niveaux présentées au paragraphe précédent sont mises à mal lorsque le comportement microscopique « déborde » un peu sur le comportement macroscopique, 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 microscopique, 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 microscopiques, 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 microscopiques (qui en quelque sorte correspond à une 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 microscopiques, 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 microscopiques et macroscopiques.

Concrètement, les méthodes de décomposition de domaine sont des solveurs qui ont généralement un fonctionnement multi-échelles. Elles partent d'un modèle microscopique 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 macroscopique 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 microscopique continu sur toute la structure, et un champ de contraintes microscopique équilibré (au sens des éléments finis) 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 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.

13.5 Super-éléments

Histoire

Originellement introduit dans l'aéronautique dans les années 60 (d'où le choix de la figure 13.13), 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.

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.

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.

13.5.1 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 degrés de liberté 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 degrés de liberté internes est appelée **condensation statique** ou simplement **condensation**.

Regardons ce qui se passe d'un point de vue matriciel. Pour cela, considérons que nous ayons à résoudre le système :

$$[K] \{q\} = \{F\} \quad (13.3)$$

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

$$\begin{bmatrix} [K_{bb}] & [K_{bi}] \\ [K_{ib}] & [K_{ii}] \end{bmatrix} \begin{Bmatrix} \{q_b\} \\ \{q_i\} \end{Bmatrix} = \begin{Bmatrix} \{f_b\} \\ \{f_i\} \end{Bmatrix} \quad (13.4)$$

Si la matrice $[K_{ii}]$ n'est pas singulière, alors la seconde équation peut être résolue en terme de variables internes en :

$$\{q_i\} = [K_{ii}]^{-1} (\{f_i\} - [K_{ib}] \{q_b\}) \quad (13.5)$$

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

$$[\tilde{K}_{bb}] \{q_b\} = \{\tilde{f}_b\} \quad (13.6)$$

avec :

$$[\tilde{K}_{bb}] = [K_{bb}] - [K_{bi}] [K_{ii}]^{-1} [K_{ib}] \quad \text{et} \quad \{\tilde{f}_b\} = \{f_b\} - [K_{bi}] [K_{ii}]^{-1} \{f_i\} \quad (13.7)$$

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

Notons que la matrice $[K_{ii}]$ 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**.

13.5.2 Remonter aux degrés de liberté internes

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

$$\{q_i\} = [K_{ii}]^{-1} (\{f_i\} - [K_{ib}] \{q_b\}) \quad (13.8)$$

13.6 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 éléments finis **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.

13.6.1 Modification du chargement uniquement

Restons sur le problème structurel correspondant au système matriciel :

$$[K] \{q\} = \{F\} \quad (13.9)$$

où $[K]$ est de dimension $n \times n$.

Si l'on souhaite considérer plusieurs cas de chargement $\{F_1\}, \dots, \{F_k\}$, on peut soit résoudre k fois le système précédent, soit résoudre le système :

$$[K] [\bar{q}] = [\bar{F}] \quad (13.10)$$

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

13.6.2 Modification de la matrice

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

On est amené à chercher une solution du système :

$$([K] + \Delta [K]) \{q\} = \{F\} \quad (13.11)$$

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

Certaines formules existent pour des modification mineures de la matrice $[K]$, mais nous ne les présenterons pas.

13.6.3 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 :

$$\mathcal{L} = \frac{1}{2} \langle q \rangle [K] \{q\} - \langle q \rangle \{F\} \quad (13.12)$$

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

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

$$\mathcal{L}^* = \frac{1}{2} \langle q \rangle [K] \{q\} - \langle q \rangle \{F\} - \langle \lambda \rangle ([C] \{q\} - \{\delta\}) \quad (13.13)$$

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

$$\begin{cases} [K][R] = [0] \\ [R]^T [R] = [I_r] \end{cases} \quad (13.14)$$

On introduit la matrice :

$$[K_\alpha] = [K] + \alpha [R][R]^T \quad (13.15)$$

qui, pour $\alpha > 0$ est régulière car symétrique et définie-positive. Les colonnes de $[R]$ sont vecteurs propres de $[K_\alpha]$ pour la valeur propre α .

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

$$[K_\alpha]([I_n] - [R][R]^T)\{q\} = \{F\} \quad (13.16)$$

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

$$[K_\alpha]\{v\} = \{F\} \quad (13.17)$$

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

$$\{q\} = \{v\} + [R][R]^T\{q\} = [K_\alpha]^{-1}\{F\} + [R][R]^T\{q\} \quad (13.18)$$

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

$$[S_\alpha] = [S] + \frac{1}{\alpha} [R][R]^T \quad (13.19)$$

avec $[S][R] = 0$ et $[R]^T[S] = 0$.

C'est cette matrice $[S]$ que l'on appelle **quasi-inverse** de $[K]$, car :

$$[S][K] = [K][S] = [I] - [R][R]^T \quad (13.20)$$

Elle est de dimension $n \times n$, symétrique, semi-définie positive (ses valeurs propres sont de même signe), admet les mêmes valeurs propres que $[K]$ et en particulier ceux de la valeur propre nulle dont 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...

13.6.4 Deux mots de statistiques

La méthode de réanalyse permet donc, pour un coût de calcul maîtrisé, d'obtenir les réponses à plusieurs problèmes proches. Il se peut donc tout à fait que l'on désire faire varier un paramètre afin d'en étudier l'influence. Notons X ce paramètre variable, et $Y = \mathcal{F}(X)$ la réponse du système correspondante.

Dans ce cas, si l'on connaît la répartition statistique de X , i.e. sa densité de probabilité (encore appelée fonction de masse), alors on peut réaliser un échantillonnage correct de la plage dans laquelle ce paramètre varie afin de pouvoir obtenir des statistiques sur la réponse Y . Comme on raisonne sur un nombre de points finis, on utilisera

donc une formulation discrète. Par exemple, une approximation de la moyenne de la réponse sera donnée par une formule de type :

$$E[Y] = \frac{\sum p_i \mathcal{F}(X_i)}{\sum p_i} \quad (13.21)$$

où p_i est le poids, ou la pondération, correspondant à la fréquence à laquelle la valeur X_i du paramètre X apparaît. On pourrait tout aussi bien, suite à l'analyse des réponses Y , approcher celle-ci par une fonction quelconque, i.e. donner une forme analytique à la fonction \mathcal{F} , et raisonner ensuite de manière continue.

Si c'est vraiment la caractérisation d'un système sous sollicitations aléatoires qui nous intéresse, alors on effectuera plutôt une analyse stochastique, et on se reportera au chapitre 22.

13.7 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 éléments finis. 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.

13.7.1 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 :

$$\Omega + V = (I + V)(\Omega); \quad V \in W^{1,\infty}(\Omega; \mathbb{R}^n) \quad (13.22)$$

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 :

$$\frac{d\mathbf{J}}{d\Omega}(\Omega, u_v, V) = \int_\Gamma u_\Omega V \cdot n + \int_\Omega u'_{\Omega;v} \quad (13.23)$$

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

$$u'_{\Omega;v} = \lim_{t \rightarrow 0} \frac{u_{\Omega+tV}(x) - u_\Omega(x)}{t}, \quad \forall x \in \Omega \quad (13.24)$$

La méthode de dérivation par transport conduit à :

$$\frac{d\mathbf{J}}{d\Omega}(\Omega, u_v, V) = \int_\Gamma u_\Omega \operatorname{div} V + \int_\Omega \dot{u}_{\Omega;v} \quad (13.25)$$

avec :

$$\dot{u}_{\Omega;v} = \lim_{t \rightarrow 0} \frac{u_{\Omega+tV} \circ (I + tV)(x) - u_\Omega(x)}{t}, \quad \forall x \in \Omega \quad (13.26)$$

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

$$\frac{\partial \mathbf{J}}{\partial \Omega}(\Omega, u, V) = \int_\Gamma u V \cdot n \quad (13.27)$$

La difficulté tient à ce que l'ensemble des domaines Ω ne constitue pas un espace vectoriel. En effet, 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 :

$$J(F, u) = \mathbf{J}(F(\Omega), u \circ F^{-1}), \quad F \in W^{1,\infty}(\Omega; \mathbb{R}^n) \quad (13.28)$$

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 :

$$\frac{\partial J}{\partial F}(I, u) \cdot V = \int_{\Omega} u \operatorname{div} V \quad (13.29)$$

On pourra alors nous faire remarquer que :

$$\frac{\partial \mathbf{J}}{\partial \Omega}(\Omega, u, V) \neq \frac{\partial J}{\partial F}(I, u) \cdot V \quad (13.30)$$

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.

Nous parlerons plus en détail de dérivées par rapport à la géométrie dans le chapitre 16 sur l'optimisation, où sera abordée l'optimisation de forme.

13.7.2 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).