Manuel d'Utilisation
Fascicule U2.08 : Fonctions avancées et contrôle des calculs
Document : U2.08.04

Notice de calcul au flambage

Résumé :

L'objectif de cette documentation est de présenter un guide méthodologique pour une analyse de flambage non linéaire d'une structure. On y aborde principalement deux fonctionnalités du *Code_Aster* :

- l'analyse de flambement linéaire, dite d'Euler, au travers de MODE_ITER_SIMULT, (option TYPE_RESU : 'MODE_FLAMB'),
- le calcul de l'évolution quasi-statique (opérateur STAT_NON_LINE) de la structure qui présente des non linéarités géométriques et comportementales, dont on cherche un point limite, voire la réponse post-critique.

La première étape est, généralement, un calcul de flambage d'Euler, qui permettra de connaître les modes de flambement et les charges critiques correspondantes. Du point de vue du concepteur, la connaissance du premier mode et de sa charge critique est souvent suffisante, afin de se définir une marge de fonctionnement par rapport au chargement imposé : le coefficient multiplicateur entre le chargement imposé et la charge critique la plus faible donne la marge de sécurité.

Remarques

- *La connaissance du premier mode de flambement peut aussi servir d'indication pour optimiser la gestion du calcul incrémental non linéaire mené par la suite. En effet, à l'approche de la charge critique, on peut alors décider de modifier le pilotage ou de réduire le pas de temps, voire d'augmenter le nombre d'itérations de vérification de l'équilibre dans la méthode de résidu, à chaque pas de charge.*
- *L'allure du mode de flambement d'Euler peut aussi servir pour imposer un défaut géométrique initial sur la structure, afin de s'assurer, entre autre, que le calcul non linéaire incrémental bifurquera bien sur ce mode.*

L'analyse d'Euler étant par définition linéaire, elle ne permet pas de prendre en compte des relations de comportement inélastiques ou du contact. Il est alors nécessaire de faire un calcul non linéaire, qui en quasi-statique s'appuiera sur la commande STAT_NON_LINE du *Code_Aster*. C'est la méthode classique incrémentale par résidu en équilibre. Les points particuliers de son utilisation seront abordés par la suite.

1 Analyse de flambement d'Euler

Le calcul des modes de flambement au sens d'Euler [bib5] peut se faire par l'opérateur de résolution de problèmes aux valeurs propres MODE_ITER_SIMULT (ou bien MODE_ITER_INV). Dans le cadre du flambage, on a la syntaxe typique suivante :

```
MODP1 = MODE_ITER_SIMULT ( MATR_A = RAMEP1 ,
                           MATR_B = RAGEP1 ,
                           TYPE_RESU = 'MODE_FLAMB' ,
                           CALC_FREQ = _F( OPTION = 'BANDE' ,
                                           CHAR_CRIT = ( -2.4 ,-2.2 , ) ,
                                           DIM_SOUS_ESPACE = 80 ,
                                           NMAX_ITER_SOREN = 80 , ) , )
```

L'argument du mot clé MATR_A doit être la matrice de rigidité dite matérielle, alors que le mot clé MATR_B attend la matrice de rigidité géométrique. Si on avait employé l'opérateur MODE_ITER_INV, les arguments des mots clés MATR_A et MATR_B seraient les mêmes.

Pour rappel, les modes de flambement sont les modes propres du problème aux valeurs propres suivant :

$$\left(\mathbf{K}+\mu\mathbf{K}_g\right)\mathbf{x}=0 \Leftrightarrow \mathbf{K}\mathbf{x}=\lambda\mathbf{K}_g\mathbf{x}$$

Avec $\begin{cases}\mathbf{K} : \text{matrice de rigidité matérielle}\\ \mathbf{K}_g : \text{matrice de rigidité géométrique}\\ \lambda : \text{valeur propre } (\lambda = -\mu \text{ avec } \mu : \text{coefficient multiplicateur du chargement})\end{cases}$

La rigidité matérielle (ou élastique) se calcule avec l'option 'RIGI_MECA' de CALC_MATR_ELEM.
La rigidité géométrique se calcule à partir du champ de contrainte solution du problème linéaire (option 'RIGI_GEOM' de CALC_MATR_ELEM). Il faut donc avoir effectué un calcul linéaire statique préalablement à l'utilisation de MODE_ITER_SIMULT pour le flambement.

Si le chargement est composé d'une partie fixe (non pilotée) et d'une partie variable, le coefficient multiplicateur du chargement ne doit, bien sûr, porter que sur la partie variable. La contribution de l'autre partie du chargement se retrouve dans le premier membre. Notons f_c le chargement fixe et f_v le chargement piloté (proportionnel à μ). Le problème aux valeurs propres devient :

$$\left(\mathbf{K}+\mathbf{K}_g\left(f_c+\mu f_v\right)\right)\mathbf{x}=0 \Leftrightarrow \left(\mathbf{K}+\mathbf{K}_g\left(f_c\right)\right)\mathbf{x}=\lambda\mathbf{K}_g\left(f_v\right)\mathbf{x}$$

Avec $\begin{cases}\mathbf{K} : \text{matrice de rigidité matérielle}\\ \mathbf{K}_g\left(f_c\right) : \text{matrice de rigidité géométrique pour le chargement non piloté}\\ \mathbf{K}_g\left(f_v\right) : \text{matrice de rigidité géométrique pour le chargement variable}\\ \lambda : \text{valeur propre } (\lambda=-\mu)\end{cases}$

Dans ce cas, il faut donc résoudre deux problèmes élastiques linéaires préalables, pour pouvoir calculer les deux matrices de rigidité géométriques différentes.
Afin d'être exhaustif, la présentation portera sur une structure soumise à des déplacements imposés ainsi que des efforts, qui seront la combinaison d'un chargement fixe et d'un chargement variable que l'on pilotera avec un coefficient croissant pouvant conduire au flambage.

1.1 Etape 1 : Calcul(s) linéaire(s) préalable(s)

On va se servir de MECA_STATIQUE. La structure, maillée en éléments de type coque (éléments de type coques volumiques [bib3]), est soumise à des conditions aux limites de Dirichlet (CONDLIM) et de Neumann. Ces dernières se décomposent en :

- PESA : champ de pesanteur,
- PRESPH : champ de pression imposé non piloté,
- PRESPS1 : champ de pression imposé variable.

Pour l'analyse de flambage, il faut séparer les efforts constants de ceux qui sont variables (pilotés par un coefficient). On va donc faire deux calculs statiques linéaires. Le premier sera le cas de la structure soumise aux déplacements imposés et aux efforts constants, le second verra la structure soumise aux déplacements imposés et aux efforts variables.

Chargement piloté :

```
RESC11P1 = MECA_STATIQUE ( MODELE = MODELE ,
                           CHAM_MATER = CHMAT ,
                           CARA_ELEM = CARAELEM ,
                           EXCIT = ( _F( CHARGE = CONDLIM , ) ,
                                     _F( CHARGE = PRESPS1 , ) , ) ,
                           OPTION = 'SIEF_ELGA_DEPL' ,
                           PLAN = 'MOY' , )
```

Chargement non piloté :

```
RESC12P1 = MECA_STATIQUE ( MODELE = MODELE ,
                           CHAM_MATER = CHMAT ,
                           CARA_ELEM = CARAELEM,
                           EXCIT = ( _F( CHARGE=CONDLIM , ) ,
                                     _F( CHARGE = PESA , ) ,
                                     _F( CHARGE = PRESPH , ) , ) ,
                           OPTION = 'SIEF_ELGA_DEPL' ,
                           PLAN = 'MOY' , )
```

On va utiliser le champ de contrainte pour calculer les matrices de rigidité géométrique associées, pour les deux chargements :

```
SIGC11P1 = CREA_CHAMP ( TYPE_CHAM = 'ELGA_SIEF_R' ,
                        OPERATION = 'EXTR' ,
                        RESULTAT = RESC11P1 ,
                        NOM_CHAM = 'SIEF_ELGA_DEPL' ,
                        TYPE_MAXI = 'MINI' ,
                        TYPE_RESU = 'VALE' , )
#
REGC11P1 = CALC_MATR_ELEM ( OPTION = 'RIGI_GEOM' ,
                            MODELE = MODELE ,
                            CARA_ELEM = CARAELEM ,
                            SIEF_ELGA = SIGC11P1 , )
```

REGC11P1 est donc la matrice de raideur géométrique associée au cas de chargement variable (PRESPS1).

On calcule, de même, la matrice de raideur géométrique pour le chargement constant (PESA et PRESPH), à partir de RESC12P1 :

```
SIGC12P1 = CREA_CHAMP ( TYPE_CHAM = 'ELGA_SIEF_R' ,
                        OPERATION = 'EXTR' ,
                        RESULTAT = RESC12P1 ,
                        NOM_CHAM = 'SIEF_ELGA_DEPL' ,
                        TYPE_MAXI = 'MINI' ,
                        TYPE_RESU = 'VALE' , )
#
REGC12P1 = CALC_MATR_ELEM ( OPTION = 'RIGI_GEOM' ,
                            MODELE = MODELE ,
                            CARA_ELEM = CARAELEM ,
                            SIEF_ELGA = SIGC12P1 , )
```

Il reste à calculer la matrice de rigidité matérielle pour le chargement total :

```
REMEP1 = CALC_MATR_ELEM ( OPTION = 'RIGI_MECA' ,
                          MODELE = MODELE ,
                          CHAM_MATER = CHMAT ,
                          CARA_ELEM = CARAELEM ,
                          CHARGE = ( CONDLIM , PESA ,
                                     PRESPH , PRESPS1 , ) , )
```

Toutes les matrices élémentaires sont calculées, l'étape suivante est donc leur assemblage :

```
NUP1 = NUME_DDL ( MATR_RIGI = REMEP1 , )
#
RAMC1P1 = ASSE_MATRICE ( MATR_ELEM = REMEP1 ,
                         NUME_DDL = NUP1 , )
#
RAGEP1 = ASSE_MATRICE ( MATR_ELEM = REGC11P1 ,
                        NUME_DDL = NUP1 , )
#
RAGC12P1 = ASSE_MATRICE ( MATR_ELEM = REGC12P1 ,
                          NUME_DDL = NUP1 , )
```

On somme ensuite les matrices de rigidité matérielle (RAMC1P1) et géométrique (RAGC12P1) correspondant au cas de chargement constant :

```
RAMEP1 = COMB_MATR_ASSE ( COMB_R = ( _F( MATR_ASSE = RAMC1P1 ,
                                        COEF_R = 1.0 , ) ,
                                     _F( MATR_ASSE = RAGC12P1 ,
                                        COEF_R = 1.0 , ) , ) , )
```

Les deux matrices nécessaires au calcul des modes de flambement sont donc construites.

1.2 Etape 2 : Calcul des modes d'Euler

Il peut être utile de faire des tests de STURM (opérateur IMPR_STURM) sur l'intervalle de recherche sur lequel on veut trouver les cas de flambement. Ainsi, cela permettra d'optimiser la taille de l'intervalle et de contrôler le bon déroulement du calcul modal ultérieur puisqu'on connaîtra d'avance le nombre de modes existants. La syntaxe est :

```
IMPR_STURM  ( MATR_A = RAMEP1 ,
              MATR_B = RAGEP1 ,
              TYPE_RESU = 'MODE_FLAMB' ,
              CHAR_CRIT_MIN = -2.4 ,
              CHAR_CRIT_MAX = -2.2 , )
```

Une fois l'intervalle de recherche de charge critique de flambage choisi, on peut alors mettre en œuvre MODE_ITER_SIMULT comme suit :

```
MODP1 = MODE_ITER_SIMULT ( MATR_A = RAMEP1 ,
                           MATR_B = RAGEP1 ,
                           TYPE_RESU = 'MODE_FLAMB' ,
                           CALC_FREQ = _F( OPTION = 'BANDE' ,
                                           CHAR_CRIT = ( -2.4 ,-2.2 , ) ,
                                           DIM_SOUS_ESPACE = 80 ,
                                           NMAX_ITER_SOREN = 80 , ) , )
```

Remarque

Si l'algorithme ne converge pas ou si le nombre de modes n'est pas celui prédit par IMPR_STURM, il peut être utile d'augmenter les valeurs de DIM_SOUS_ESPACE et NMAX_ITER_SOREN.

On norme les modes [bib6], uniquement en se servant des degrés de liberté de translation :

```
MODP1 = NORM_MODE ( reuse = MODP1
                    MODE = MODP1 ,
                    NORME = 'TRAN' , )
```

Les modes peuvent ensuite être post-traités.

Remarques

- *Il est indispensable de vérifier que la raideur géométrique du modèle choisi est bien une option disponible dans Code_Aster (par exemple, ce n'est pas le cas des DKT).*
- *Une discrétisation plus fine conduit normalement à une baisse des charges critiques.*
- *La discrétisation doit être apte à capter les modes de flambement, sachant que ces modes peuvent engendrer des déformations localisées (plis). Le calcul préalable des modes dynamiques peut constituer une première indication sur la qualité du maillage, bien que ces modes puissent être très différents des modes de flambement.*
- *Les charges critiques des différents modes sont proportionnelles au module d'Young E.*

2 Etude non linéaire quasistatique de la structure

Cette étape se justifie si la structure présente de fortes non linéarités, dont l'analyse d'Euler ne peut tenir compte. L'opérateur de résolution des problèmes non linéaires en quasi-statique se nomme STAT_NON_LINE [bib7].
Ces non linéarités peuvent être liées au matériau qui peut avoir un comportement élastoplastique [bib8], comme dans l'exemple qui va suivre. La prise en compte du contact, voire du frottement, est une autre source de non linéarités. On peut aussi citer le cas des chargements suiveurs, comme la pression ([bib1] et [bib2] pour les éléments de type coques volumiques), qui nécessitent une approche non linéaire.

Pour l'étude d'une structure potentiellement instable ou susceptible de connaître un point limite, qui risque donc de rencontrer une bifurcation en solution au cours de l'évolution du chargement, il est souvent utile de pouvoir choisir une branche de solution particulière (souvent la solution physique quand elle est définie *a priori* sans ambiguïtés). Pour cela, l'utilisateur peut avoir à introduire un défaut initial qui va « forcer » la structure à bifurquer sur la branche de solution particulière.
Plusieurs méthodes existent pour définir ce défaut.

- L'une des plus adaptée est de prédéformer légèrement la structure suivant l'allure du mode d'Euler de flambement correspondant à la branche que l'on veut suivre. L'amplitude de cette prédéformation doit être faible, par exemple moins de 1/10ème de l'épaisseur pour une structure mince. L'idéal étant de trouver le défaut minimal qui est compatible avec une performance satisfaisante de l'algorithme de résidu en équilibre. En effet, un défaut trop faible peut entraîner une difficulté de convergence du résidu, principalement dans le cas d'un pilotage en effort.
- Le défaut géométrique peut aussi être défini par mesures expérimentales de la pièce réelle dont la géométrie ne saurait être parfaite.
- Le défaut peut aussi prendre la forme d'une perturbation du chargement (défaut d'alignement, rajout d'un chargement localisé, ...) ou des caractéristiques mécaniques du matériau (affaiblissement local du module d'Young, par exemple). Il peut néanmoins être alors plus difficile d'adapter le défaut au mode de flambage désiré, surtout si la structure présente des modes relativement voisins.

Remarque

Dans certains cas, même sur le problème non perturbé, le chargement est tel qu'il provoque la bifurcation désirée.

Un des autres points particuliers, liés à l'instabilité, est le choix de la technique de pilotage de l'algorithme STAT_NON_LINE. En effet, le pilotage classique en effort n'est plus adapté car il ne peut capter une branche instable de solution. De même, à l'approche d'un point limite, la convergence avec le pilotage en effort deviendra de plus en plus difficile, la matrice de rigidité tangente devenant singulière. Il est alors nécessaire de réduire l'incrément de charge et d'augmenter le nombre maximal d'itération pour continuer le calcul.
Il existe des techniques de pilotage [bib9] permettant de contourner ces difficultés numériques. Parmi les méthodes proposées par le *Code_Aster*, celle dite par longueur d'arc [bib12] (option TYPE='LONG_ARC' du mot clé PILOTAGE dans STAT_NON_LINE), qui est la plus adaptée pour les instabilités de type flambage, dans le cas de snap-backs éventuels « doux » [bib13]. Pour les cas de snap-backs plus brutaux, Crisfield propose une variante [bib13], non disponible dans la version 6 du *Code_Aster*.
D'autres méthodes existent, comme celle de Riks [bib14] (non disponible non plus), qui traite aussi le cas dynamique.
Si l'on ne veut qu'obtenir le point limite, y compris avec une bonne précision, un pilotage en chargement peut suffire, à condition de bien gérer les paramètres de pas d'incrément de charge (SUBD_PAS et SUBD_PAS_MINI du mot clé INCREMENT) et de nombre d'itérations maximal autorisé (ITER_GLOB_MAXI de CONVERGENCE). Il peut aussi être utile, à l'approche du point limite, de ne plus employer la matrice tangente réactualisée pour le solveur, puisqu'elle est quasi-singulière. On peut alors se contenter de ne pas réactualiser cette matrice à chaque calcul (paramètres REAC_INCR et REAC_ITER) ou, dans le pire des cas, adopter la matrice élastique de base (PREDICTION='ELASTIQUE' et MATRICE='ELASTIQUE' du mot clé NEWTON).

Voici un exemple d'utilisation de STAT_NON_LINE pour un calcul élastoplastique en grands déplacements ([bib4] pour les éléments employés, qui sont de type coques volumiques), avec pilotage en efforts :

```
RESU = STAT_NON_LINE ( MODELE = MODELE ,
                       CHAM_MATER = CHMAT ,
                       CARA_ELEM = CARAELEM ,
                       EXCIT = ( _F( CHARGE = CONDLIM ,
                                     TYPE_CHARGE = 'FIXE_CSTE' , ) ,
                                 _F( CHARGE = PESA ,
                                     TYPE_CHARGE = 'FIXE_CSTE' , ) ,
                                 _F( CHARGE = PRESPH ,
                                     FONC_MULT = FONCMUL2 ,
                                     TYPE_CHARGE = 'SUIV' , ) ,
                                 _F( CHARGE = PRESPS1 ,
                                     FONC_MULT = FONCMUL ,
                                     TYPE_CHARGE = 'SUIV' , ) , ) ,
                       COMP_INCR = ( _F( RELATION = 'VMIS_ISOT_TRAC' ,
                                         COQUE_NCOU = 1 ,
                                         DEFORMATION = 'GREEN_GR' ,
                                         GROUP_MA = ( 'VIROLE' , 'TOIT' ,
                                                      'ANNEAUX' , 'SGOU') ,
                                                    ) , ) ,
                       COMP_ELAS = _F( RELATION = 'ELAS' ,
                                       COQUE_NCOU = 1 ,
                                       DEFORMATION = 'GREEN_GR' ,
                                       GROUP_MA = 'LTIGE' , ) ,
                       INCREMENT = _F( LIST_INST = L_INST1 ,
                                       NUME_INST_FIN = 14 ,
                                       SUBD_PAS = 4 ,
                                       SUBD_PAS_MINI = 1.E-9 , ) ,
                       NEWTON = _F( REAC_INCR = 1 ,
                                    PREDICTION = 'TANGENTE' ,
                                    MATRICE = 'TANGENTE' ,
                                    REAC_ITER = 1 , ) ,
                       CONVERGENCE = _F( RESI_GLOB_RELA = 1.E-06 ,
                                         ITER_GLOB_MAXI = 40 ,
                                         ARRET = 'OUI' , ) ,
                       SOLVEUR = _F( METHODE = 'MULT_FRONT' ,
                                     RENUM = 'METIS' , ) , )
```

Remarques

- *On utilise la matrice tangente réactualisée à chaque calcul, en autorisant le sous-découpage du pas de charge.*
- *Les pressions imposées sont des efforts suiveurs (TYPE_CHARGE='SUIV').*
- *Dans le cas d'une modélisation en éléments massifs, le tenseur de déformation recommandé en grands déplacements est 'SIMO_MIEHE'.*

Si on veut remplacer le pilotage en effort par une méthode par longueur d'arc, il suffit d'écrire :

```
RESU = STAT_NON_LINE ( MODELE = MODELE ,
                       CHAM_MATER = CHMAT ,
                       CARA_ELEM = CARAELEM ,
                       EXCIT = ( _F( CHARGE = CONDLIM ,
                                     TYPE_CHARGE = 'FIXE_CSTE' , ) ,
                                 _F( CHARGE = PESA ,
                                     TYPE_CHARGE = 'FIXE_CSTE' , ) ,
                                 _F( CHARGE = PRESPH ,
                                     FONC_MULT = FONCMUL2 ,
                                     TYPE_CHARGE = 'SUIV' , ) ,
                                 _F( CHARGE = PRESPS1 ,
                                     TYPE_CHARGE = 'FIXE_PILO' , ) , ) ,
                       COMP_INCR = ( _F( RELATION = 'VMIS_ISOT_TRAC' ,
                                         COQUE_NCOU = 1 ,
                                         DEFORMATION = 'GREEN_GR' ,
                                         GROUP_MA = ( 'VIROLE' , 'TOIT' ,
                                                      'ANNEAUX' , 'SGOU') ,
                                                    ) , ) ,
                       COMP_ELAS = _F( RELATION = 'ELAS' ,
                                       COQUE_NCOU = 1 ,
                                       DEFORMATION = 'GREEN_GR' ,
                                       GROUP_MA = 'LTIGE' , ) ,
                       INCREMENT = _F( LIST_INST = L_INST1 ,
                                       NUME_INST_FIN = 14 ,
                                       SUBD_PAS = 4 ,
                                       SUBD_PAS_MINI = 1.E-9 , ) ,
                       NEWTON = _F( REAC_INCR = 1 ,
                                    PREDICTION = 'TANGENTE' ,
                                    MATRICE = 'TANGENTE' ,
                                    REAC_ITER = 1 , ) ,
                       CONVERGENCE = _F( RESI_GLOB_RELA = 1.E-06 ,
                                         ITER_GLOB_MAXI = 40 ,
                                         ARRET = 'OUI' , ) ,
                       PILOTAGE = _F( GROUP_NO = 'G' ,
                                      TYPE = 'LONG_ARC' ,
                                      NOM_CMP = ( 'DY' , ) ,
                                      COEF_MULT = 7. ) , )
```

Remarques

- *Dans la version 6 du Code_Aster, on ne peut pas piloter de forces suiveuses.*
- *Pour le pilotage par longueur d'arc, il est, en général, recommandé que GROUP_NO contienne toute la structure.*

Pour finir, citons deux articles de Crisfield qui donnent une bonne vision générale des problèmes et méthodes liés aux calculs non linéaires pouvant présenter divers types d'instabilités ([bib15] et [bib11]).

Quelques cas-tests du *Code_Aster* traitant du flambage :

Modes d'Euler :

- sdls504
- sdls505
- ssll103
- ssll105
- ssll403
- ssll404
- ssls110

Modes d'Euler et calcul non linéaire :

- ssnl123

Calcul non linéaire :

- ssnl502
- ssnp305 : calcul jusqu'à un snap-through

3 Bibliographie

[1] E. LORENTZ : *Code_Aster*, documentation de référence, [R3.03.04], 1996

[2] P. MASSIN, M. AL MIKDAD : *Code_Aster*, documentation de référence, [R3.03.07], 2000

[3] P. MASSIN, A. LAULUSA : *Code_Aster*, documentation de référence, [R3.07.04], 2000

[4] P. MASSIN, M. AL MIKDAD : *Code_Aster*, documentation de référence, [R3.07.05], 2000

[5] O. BOITEAU : *Code_Aster*, documentation de référence, [R5.01.01], 2001

[6] B. QUINNEZ, J.R. LEVESQUE : *Code_Aster*, documentation de référence, [R5.01.03], 1997

[7] N. TARDIEU, I. VAUTIER : *Code_Aster*, documentation de référence, [R5.03.01], 2001

[8] J.M. PROIX, E. LORENTZ : *Code_Aster*, documentation de référence, [R5.03.02], 2001

[9] E. LORENTZ : *Code_Aster*, documentation de référence, [R5.03.80], 2001

[10] C. ROSE : *Code_Aster*, documentation de référence, [R6.02.02], 2001

[11] M.A. CRISFIELD, G. JELENIC, Y. MI, H.-G. ZHONG & Z. FAN : Some aspects of the non-linear finite element method, Finite Elements in Analysis and Design, Vol. 27, 19-40, 1997

[12] M.A. CRISFIELD : A fast incremental iterative solution procedure that handles snap through, Computers & Structures, Vol. 13, 55-62, 1981

[13] H.-B. HELLWEG & M.A. CRISFIELD : A new arc-length method for handling sharp snap-backs, Computers & Structures, Vol. 66, 705-709, 1998

[14] E. RIKS, C.C. RANKIN & F.A. BROGAN : On the solution of mode jumping phenomena in thin-walled shell structures, Comp. Meth. In Applied Mech. And Engrg., Vol. 1367, 59-92, 1996

[15] J. SHI & M.A. CRISFIELD : Combining arc-length and line searches in path-following, Comm. Numer. Meth. Engrg, Vol. 11, 793-803, 1995