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 de *Code_Aster* :

- l'analyse de flambement linéaire, dite d'Euler, au travers de CALC_MODES (avec le mot-clé TYPE_RESU='MODE_FLAMB'),
- le calcul de l'évolution quasistatique (opérateur STAT_NON_LINE) de la structure qui présente des non linéarités géométriques et comportementales, dont on cherche les instabilités (option CRIT_STAB), 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, du contact ou l'aspect suiveur des forces. Il est alors nécessaire de faire un calcul non linéaire, qui en quasi-statique s'appuiera sur la commande STAT_NON_LINE de *Code_Aster*. C'est la méthode classique incrémentale par résidu en équilibre.

Les points particuliers de son utilisation seront abordés par la suite, en particulier l'utilisation de l'analyse de stabilité non-linéaire avec le mot-clé CRIT_STAB (qui est aussi disponible dans DYNA_NON_LINE pour les études dynamiques).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Analyse de flambement d'Euler

Le calcul des modes de flambement au sens d'Euler [bib5] se fait par l'opérateur de résolution de problèmes aux valeurs propres CALC_MODES (avec le mot-clé TYPE_RESU = 'MODE_FLAMB'). Dans le cadre du flambage, on a la syntaxe typique suivante :

```
MODP1 = CALC_MODES ( MATR_RIGI = RAMEP1 ,
                     MATR_RIGI_GEOM = RAGEP1 ,
                     TYPE_RESU = 'MODE_FLAMB' ,
                     OPTION = 'BANDE' ,
                     CALC_CHAR_CRIT = _F( CHAR_CRIT = ( -2.4 ,-2.2 , ) , ),
                     SOLVEUR_MODAL=_F( METHODE= 'SORENSEN',
                                       DIM_SOUS_ESPACE = 80 ,
                                       NMAX_ITER_SOREN = 80 , ) , )
```

L'argument du mot clé MATR_RIGI doit être la matrice de rigidité dite matérielle, alors que le mot clé MATR_RIGI_GEOM attend la matrice de rigidité géométrique.

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

$$(\boldsymbol{K}+\mu\,\boldsymbol{K}_g)\boldsymbol{x}=0 \Leftrightarrow \boldsymbol{K}\,\boldsymbol{x}=\lambda\,\boldsymbol{K}_g\boldsymbol{x}$$

avec :

K : matrice de rigidité

K_g : matrice de rigidité géométrique

λ : valeur propre ($\lambda=-\mu$ avec μ coefficient multiplicateur du chargement)

Remarque :

C'est bien le coefficient de charge critique μ et non λ qui est stocké dans le concept résultat.

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 CALC_MODES 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 :

$$(\boldsymbol{K}+\boldsymbol{K}_g(f_c+\mu\,f_v))\boldsymbol{x}=0 \Leftrightarrow (\boldsymbol{K}+\boldsymbol{K}_g(f_c))\boldsymbol{x}=\lambda\,\boldsymbol{K}_g(f_v)\boldsymbol{x}$$

avec :

K : matrice de rigidité matérielle

$K_g(f_c)$: matrice de rigidité géométrique pour le chargement non piloté

$K_g(f_v)$: matrice de rigidité géométrique pour le chargement variable

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

λ : valeur propre $\lambda = -\mu$

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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 Étape 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' ,
                          )
```

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' ,
                          )
```

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' ,
                        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

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

(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' ,
                        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 Étape 2 : Calcul des modes d'Euler

Il peut être utile de faire des tests de STURM (opérateur INFO_MODE) 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 :

```
INFO_MODE  ( MATR_RIGI = RAMEP1 ,
             MATR_RIGI_GEOM = RAGEP1 ,
             TYPE_MODE = 'MODE_FLAMB' ,
             CHAR_CRIT = (-2.4 ,-2.2)  )
```

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

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
MODP1 = CALC_MODES ( MATR_RIGI = RAMEP1 ,
                     MATR_RIGI_GEOM = RAGEP1 ,
                     TYPE_RESU = 'MODE_FLAMB' ,
                     OPTION = 'BANDE' ,
                     CALC_ CHAR_CRIT =_F( CHAR_CRIT = ( -2.4 ,-2.2 , ) , ),
                     SOLVEUR_MODAL=_F( 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 INFO_MODE , 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.*
- *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 .*
- *Avec cette méthode, on ne peut tenir compte du caractère suiveur des forces. En effet, le calcul de raideur géométrique fait avec l'option RIGI_GEOM de CALC_MATR_ELEM se base sur l'hypothèse que toutes les forces imposées sur le problème mécanique sont de type force morte. Si l'on veut tenir compte du caractère suiveur de certaines sollicitations, il faut obligatoirement utiliser les opérateurs non-linéaires STAT_NON_LINE ou DYNA_NON_LINE, avec le mot-clé CRIT_STAB, comme on le verra dans la suite de ce document.*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Étude non linéaire quasi-statique 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 quasistatique 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 dans *Code_Aster*.

On peut distinguer deux types d'analyses de stabilité non-linéaires, pouvant se combiner.
D'une part la généralisation de l'analyse d'Euler présentée précédemment : on va faire du suivi des charges critiques et modes associés au cours du calcul non-linéaires incrémental. Cela se traduit par une analyse de type Euler sur les matrices de raideurs réactualisées. Ce type d'analyse se fait souvent sur une structure sans défaut initial.

D'autre part, on peut tenir compte de défauts introduits sur le modèle parfait, afin de « forcer » la bifurcation en solution et de faire du suivi de branche pour analyser la réponse post-critique.
Bien évidemment, ce suivi de solution post-critique peut être initié par l'analyse aux valeurs propres sur matrices de raideurs réactualisées, en particulier pour bien détecter la bifurcation et définir le défaut alors introduit en se basant sur le mode de flambement observé.

2.1 Analyse de stabilité sur matrices de raideurs réactualisées

Que ce soit en quasistatique (opérateur STAT_NON_LINE) ou en dynamique (opérateur DYNA_NON_LINE), *Code_Aster* permet de mener des analyses incrémentales de stabilité au sens du flambage sur les matrices de raideur courantes. Ces étapes de calcul sont gérées par le mot-clé facteur CRIT_STAB avec l'option TYPE = 'FLAMBEMENT' (*cf.* [U4.51.03] et [U4.53.01]) .

Remarques pour l'analyse transitoire

L'option CRIT_STAB agit comme pour le cas quasistatique : on mène toujours une analyse de flambement (donc ne se basant que sur l'étude des matrices de raideur), pas une analyse d'instabilité dynamique (amortissement devenant négatif).
Avec la modélisation fluide-structure couplée (u, p, Φ) *[R4.02.02], il faut modifier la matrice de raideur assemblée (ainsi que la raideur géométrique quand elle est utilisée). Pour cela, il faut renseigner les mot-clés suivants, sous* **CRIT_STAB** *:*
•**MODI_RIGI = 'OUI'** ,
•**DDL_EXCLUS=('PHI','PRES','DH',)** .
La liste des degrés de liberté exclus doit comporter tous les types de degrés de liberté liés au modèle fluide : dans l'exemple du cas-test FDNV100 on a donc le potentiel **PHI**, *la pression* **PRES** *et le déplacement vertical au niveau de la surface libre* **DH**. *Si l'on ne fait pas ce traitement, alors l'appel à* **CRIT_STAB** *va planter pour cause de matrice singulière et aucune stratégie de décalage ne saurait surmonter cela.*
En quasi-statique, ce problème ne se pose pas car la modélisation fluide-structure couplée n'a alors pas de sens.

Ce mot-clé permet de déclencher le calcul, à la fin de chaque incrément de temps, d'un critère de stabilité. Ce critère est utile pour déceler, au cours du chargement, le point à partir duquel on perd la stabilité (par flambage par exemple).

Ce critère est calculé de la façon suivante : à la fin d'un pas de temps, en petites perturbations, on résout $det\left(\boldsymbol{K}^{\boldsymbol{T}}+\mu.\boldsymbol{K}^{\boldsymbol{g}}\right)=0$. $\boldsymbol{K}^{\boldsymbol{T}}$ est la matrice tangente cohérente à cet instant. $\boldsymbol{K}^{\boldsymbol{g}}$ est la matrice

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

de rigidité géométrique, calculée à partir du champ de contraintes à cet instant et μ le coefficient de charge critique.

En pratique, le chargement est instable si $|\mu|<1$ (en fait $0<\mu<1$). On calcule les valeurs propres par la méthode de Sorensen (*cf.* CALC_MODES [U4.52.02]). Ceci peut être assez coûteux pour les problèmes de grande taille.

Le mot-clé CHAR_CRIT permet de gagner du temps en ne faisant qu'un test de Sturm dans la bande de fréquence fournie. Si on trouve au moins une fréquence, alors on calcule réellement les valeurs des charges critiques dans cet intervalle.

Pour les grands déplacements et les grandes déformations, on résout $det\left(\boldsymbol{K}^{\boldsymbol{T}}+\mu.\boldsymbol{I}\right)=0$ car $\boldsymbol{K}^{\boldsymbol{T}}$ contient alors $\boldsymbol{K}^{\boldsymbol{g}}$.
Le critère est alors un critère d'instabilité : quand μ change de signe (donc passe par 0) le chargement est instable.

Remarques concernant la matrice de raideur géométrique

L'analyse de stabilité sera plus précise si on dispose de la matrice de raideur géométrique : en effet, en son absence, certaines instabilités ne peuvent être détectées comme, par exemple, dans les cas purement élastiques. Entre des modèles suffisamment réalistes, il faut donc privilégier l'utilisation, pour des analyses de stabilité précises, de modèles permettant le calcul de cette raideur géométrique.

Le mot-clé NB_MODE désigne le nombre de charges critiques à calculer. Souvent la première suffit mais il peut y avoir des modes multiples.

On stocke le mode propre correspondant à la plus petite charge critique (en valeur absolue) dans l'objet résultat, sous le nom MODE_FLAMB. Ce mode propre peut être extrait et visualisé (comme un champ de déplacements ou un mode propre classique). Il est normalisé à 1 sur la plus grande composante de déplacement. Toutes les charges critiques calculées sont affichées dans le fichier .mess.

En pratique, afin de limiter le surcoût de calcul, on conseille d'optimiser les appels à CRIT_STAB. On peut pour cela utiliser les mots-clefs LIST_INST/INST/PAS_CALCUL dans le mot-clef facteur CRIT_STAB. On peut ainsi préciser à quels pas de temps on calculera les modes de flambement.
En complément, il est judicieux de n'utiliser CRIT_STAB que sur les intervalles de temps où l'on soupçonne la possibilité d'instabilités.

Enfin, si l'on veut une très bonne évaluation des charges critiques, il convient de bien raffiner le pas de temps à l'approche de cette zone. Ce conseil est d'autant plus pertinent en quasi-statique car l'utilisateur a alors souvent recours à des pas de temps plus grands qu'en dynamique.

Il est possible d'arrêter proprement (base réutilisable en poursuite) un calcul avec STAT_NON_LINE ou DYNA_NON_LINE lors de la détection d'une instabilité. Ce n'est pas le fonctionnement par défaut où le code va tenter de continuer à résoudre le problème, et si on a la convergence, alors cela signifie que l'on a réussi à suivre une des branches de solution.
Pour gérer cet arrêt (*cf.* cas-test ERREU10), il faut avoir au préalable utilisé DEFI_LIST_INST avec les arguments suivants :

```
ECHEC=_F(EVENEMENT='INSTABILITE',ACTION='ARRET',),
```

Ce qui signifie qu'en cas d'événement de type instabilité, l'action déclenchée sera l'arrêt.

Sous CRIT_STAB, les options facultatives suivantes permettent de piloter ce critère d'arrêt :

- PREC_INSTAB pour définir la précision (adimensionnelle) du critère d'arrêt,

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- SIGNE pour spécifier les valeurs critiques à considérer.

Le deuxième mot-clé ne sert que lorsque la matrice de raideur géométrique est utilisée. Par défaut on considère la solution comme étant instable si la charge critique devient comprise entre 1 et -1, mais on peut, au besoin, ne prendre en compte que la partie positive ou négative de cet intervalle.

Sans matrice de raideur géométrique, l'instabilité sera détectée lorsqu'une valeur propre de la matrice de raideur globale assemblée, soit :
- tendra vers 0 (avec une précision relative donnée par PREC_INSTAB),
- changera de signe.

Par exemple, dans le cas d'un réservoir rempli d'eau sous séisme, on peut commencer le calcul incrémental ou transitoire avec une charge critique valant 0,8 (analyse avec raideur géométrique) : ce qui signifie que le réservoir flamberait si on imposait une dépression valant 0,8 fois la pression hydrostatique imposée (le valeur positive de la charge critique correspond à une inversion de sens du chargement considéré). Donc si on ne spécifie rien, le calcul serait considéré instable et s'arrêterait. Comme, dans ce cas, on fait l'hypothèse qu'il n'y aura pas dépressurisation (par exemple par vidange), alors on dédouane la l'intervalle $[0,1]$ dans l'analyse de stabilité. Donc le problème deviendra instable si la charge critique atteint l'intervalle $[-1,0]$.

Pour les analyses en évolution monotone ce type de raisonnement se conçoit aisément, ce qui est bien le cas pour la partie statique du chargement, mais pour la partie dynamique, le chargement étant cyclique, et sauf à disposer d'informations spécifiques, il est plus sûr et plus conservatif de garder l'option par défaut et donc de considérer la structure instable si la charge critique devient inférieure à 1 en valeur absolue.

Lors d'un arrêt sur instabilité, le calcul va s'arrêter en fermant proprement la base : l'utilisateur pourra l'exploiter en poursuite.

2.2 Suivi de solution instable

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 pré-déformer légèrement la structure suivant l'allure du mode de flambement correspondant à la branche que l'on veut suivre. L'amplitude de cette pré-déformation 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. Ce défaut initial peut fort judicieusement être construit à partir du mode d'instabilité calculé avec l'option CRIT_STAB de STAT_NON_LINE / DYNA_NON_LINE (*cf.* paragraphe précédent). Ce mode tient alors compte de toutes les non-linéarités introduites dans le modèle complet. L'alternative plus économique est d'utiliser le mode de flambement d'Euler, mais qui correspond au cas linéaire.
- 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

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

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.

On peut aussi servir de la possibilité de s'arrêter proprement en cas d'instabilité (*cf.* paragraphe précédent) pour gérer en poursuite la bifurcation sur la branche de solution choisie en initiant la suite du calcul par une perturbation suivant ce mode d'instabilité.

Il existe des techniques de pilotage [bib9] permettant de contourner ces difficultés numériques. Parmi les méthodes proposées par *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 *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 (pensez à utiliser la commande DEFI_LIST_INST) 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 ( ...
                       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' , ) , ) ,
                       ... )
```

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 de rajouter :

```
RESU = STAT_NON_LINE ( ...
                       PILOTAGE = _F( GROUP_NO = 'G' ,
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                    TYPE = 'LONG_ARC' ,
                                    NOM_CMP = ( 'DY' , ) ,
                                    COEF_MULT = 7. ) ,

                          ...)
```

Remarques

- *Dans 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]).

La documentation [U2.06.11] montre un exemple d'utilisation de `CRIT_STAB` pour l'étude du comportement d'un réservoir métallique.

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

Modes d'Euler :

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

Modes d'Euler et calcul non linéaire :

- ssnl123

Modes non linéaires (`CRIT_STAB`) :

- sdnv106 (présente en plus `MODE_VIBR` pour le calcul des modes vibratoires sur raideur réactualisée)
- ssll105
- ssnl126
- ssnp306

Calcul non linéaire :

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

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Bibliographie

1) Efforts extérieurs de pression en grands déplacements [R3.03.04].

2) Pression suiveuse pour les éléments de coques volumiques [R3.03.07].

3) Éléments finis de coques volumiques [R3.07.04].

4) Éléments de coques volumiques en non linéaire géométrique [R3.07.05].

5) Algorithme de résolution pour le problème généralisé [R5.01.01].

6) Paramètres modaux et norme des vecteurs propres [R5.01.03].

7) Algorithme non linéaire quasi statique [R5.03.01].

8) Intégration des relations de comportement élasto-plastique de Von Mises [R5.03.02].

9) Méthodes de pilotage du chargement [R5.03.80].

10) Méthode multifrontale [R6.02.02].

11) M.A. Crisfield, G. Jelenic, Y. Mi, H.-G. ZhongG & Z. Fan : S*ome 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.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)