

Manuel d'Utilisation
Fascicule U4.5- : Méthodes de résolution
Document : U4.52.03

Opérateur *MODE_ITER_SIMULT*

1 But

Calculer des valeurs et vecteurs propres par des méthodes de type sous-espace. Pour le problème classique de dynamique (sans amortissement) ou le problème de flambement d'Euler, trois algorithmes sont disponibles : Sorensen, Lanczos, Bathe et Wilson. Pour le problème de dynamique avec amortissement, seules les méthodes de Sorensen et de Lanczos sont utilisables. Produit un concept *mode_meca_** (cas dynamique) ou *mode_flamb* (cas flambement d'Euler).

2 Syntaxe

```

mode_  [*] = MODE_ITER_SIMULT

# DONNEES DU PROBLEME MODAL
(  ♦  MATR_A =  A                                /  [matr_asse_DEPL_R]
                                           /  [matr_asse_DEPL_C]
                                           /  [matr_asse_PRES_R]
                                           /  [matr_asse_GENE_R]
      ♦  MATR_B =  B                                /  [matr_asse_DEPL_R]
                                           /  [matr_asse_PRES_R]
                                           /  [matr_asse_GENE_R]
      ♦  MATR_C =  C                                /  [matr_asse_DEPL_R]

# TYPE DE PROBLEME
  ♦  TYPE_RESU =      /  'DYNAMIQUE'                [DEFAULT]
                    /  'MODE_FLAMB'

# CHOIX DE LA METHODE
  ♦  METHODE =      /  'SORENSEN'                [DEFAULT]
                    /  'TRI_DIAG'
                    /  'JACOBI'

# Si METHODE = 'TRI_DIAG'
  ♦  OPTION =      /  'SANS'                [DEFAULT]
                    /  'MODE_RIGIDE'

# TYPE DE CALCUL MODAL
  ♦  CALC_FREQ =_F (  ♦  OPTION = /  'CENTRE'
                    /  'BANDE'
                    /  'PLUS_PETITE'                [DEFAULT]

# CARACTERISTIQUES DU CALCUL
#  Si TYPE_RESU = 'DYNAMIQUE'
      ♦  APPROCHE = /  'REEL'                [DEFAULT]
                    /  'IMAG'
                    /  'COMPLEXE'
#  Si OPTION = 'PLUS_PETITE'
      ♦  NMAX_FREQ = /  10                [DEFAULT]
                    /  nf                [I]
#  Si OPTION = 'CENTRE'
      ♦  FREQ =      1_f                [1_R]
      ♦  AMOR_REDUIT = 1_a                [1_R]
      ♦  NMAX_FREQ = /  10                [DEFAULT]
                    /  nf                [I]
#  Si OPTION = 'BANDE'
      ♦  FREQ =      1_f                [1_R]
#  Si TYPE_RESU = 'MODE_FLAMB'
      ♦  APPROCHE = /  'REEL'                [DEFAULT]
                    /  'IMAG'
#  Si OPTION = 'PLUS_PETITE'
      ♦  NMAX_FREQ = /  10                [DEFAULT]
                    /  nf                [I]
#  Si OPTION = 'CENTRE'
      ♦  CHAR_CRIT = 1_c                [1_R]
      ♦  NMAX_FREQ = /  10                [DEFAULT]
                    /  nf                [I]
#  Si OPTION = 'BANDE'
      ♦  CHAR_CRIT = 1_c                [1_R]

```

Titre : *Opérateur MODE_ITER_SIMULT*
Auteur(s) : *E. BOYERE, O. BOITEAU*

Clé : *U4.52.03-G* Date : *31/01/05*
Page : *3/16*

```
# CARACTERISTIQUES DE L'ESPACE DE PROJECTION
      ◇ DIM_SOUS_ESPACE =    dse            [I]
      ◇ COEF_DIM_ESPACE =    mse            [I]
      EXCLUS('DIM_SOUS_ESPACE', 'COEF_DIM_ESPACE')

# POUR PRE- ET POST-TRAITEMENTS
      ◇ PREC_SHIFT =        / 0.05            [DEFAULT]
                             / ps            [R]
      ◇ NMAX_ITER_SHIFT =    / 5            [DEFAULT]
                             / ns            [I]
      ◇ NPREC_SOLVEUR =    / 8            [DEFAULT]
                             / ndeci        [R]
      ◇ SEUIL_FREQ =        / 1.E-2        [DEFAULT]
                             / sf            [R]

# PARAMETRAGE INTERNE DES METHODES
#        Si METHODE = 'SORENSEN'
          ◇ PREC_SOREN =     / 0            [DEFAULT]
                             / pso           [R]
          ◇ NMAX_ITER_SOREN = / 20           [DEFAULT]
                             / nso           [I]
          ◇ PARA_ORTHO_SOREN = / 0.717       [DEFAULT]
                             / porso        [I]

#        Si METHODE = 'TRI_DIAG'
          ◇ PREC_ORTHO =     / 1.E-12       [DEFAULT]
                             / po            [R]
          ◇ NMAX_ITER_ORTHO = / 5           [DEFAULT]
                             / nio           [I]
          ◇ PREC_LANCZOS =    / 1.E-8       [DEFAULT]
                             / pl            [R]
          ◇ NMAX_ITER_QR =    / 30        [DEFAULT]
                             / nim           [I]

#        Si METHODE = 'JACOBI'
          ◇ PREC_BATHE =     / 1.E-10       [DEFAULT]
                             / pbat           [R]
          ◇ NMAX_ITER_BATHE = / 40           [DEFAULT]
                             / nbat           [I]
          ◇ PREC_JACOBI =    / 1.E-2       [DEFAULT]
                             / pjaco        [R]
          ◇ NMAX_ITER_JACOBI = / 12        [DEFAULT]
                             / njaco        [I]
          )

# POUR VERIFICATIONS FINALES
      ◇ VERI_MODE =    _F(
          ◇ STOP_ERREUR =    / 'OUI'        [DEFAULT]
                             / 'NON'        [R]
          ◇ SEUIL =        / 1.E-6        [DEFAULT]
                             / r            [R]
          ◇ PREC_SHIFT =    / 0.05        [DEFAULT]
                             / prs           [R]
          ◇ STURM =        / 'OUI'        [DEFAULT]
                             / 'NON'        [R]
          )
```

Titre : Opérateur *MODE_ITER_SIMULT*
Auteur(s) : **E. BOYERE, O. BOITEAU**

Clé : **U4.52.03-G** Date : **31/01/05**
Page : **4/16**

```
# SENSIBILITE
    ◇ SENSIBILITE = (
                                . . . voir [U4.50.02] . . .
    )

# DIVERS
    ◇ STOP_FREQ_VIDE =      /  'OUI'           [DEFAULT]
                           /  'NON'
    ◇ INFO =      /  1           [DEFAULT]
                  /  2           [I]
    ◇ TITRE = ti
);

# DONNEE RESULTAT
Si MATR_C =[matr_asse_DEPL_R]      alors [*] -> meca_c
Si TYPE_RESU = 'MODE_FLAMB'      alors [*] -> mode_flamb
Si MATR_A =[matr_asse_DEPL_C]      alors [*] -> meca_c
Si MATR_A =[matr_asse_DEPL_R]      alors [*] -> meca
Si MATR_A =[matr_asse_PRES_R]      alors [*] -> acou
Si MATR_A =[matr_asse_GENE_R]      alors [*] -> gene
```

3 Opérandes

3.1 Principes

Cet opérateur résout le problème généralisé aux valeurs propres suivant [R5.01.01] :

Trouver (λ, \mathbf{x}) tels que $\mathbf{A}\mathbf{x} = \lambda \mathbf{B}\mathbf{x}$, $\mathbf{x} \neq 0$, où \mathbf{A} et \mathbf{B} sont des matrices symétriques à coefficients réels. Pour modéliser un amortissement hystérétique dans l'étude des vibrations libres d'une structure, la matrice \mathbf{A} peut être complexe [U2.06.03] [R5.05.04]. Ce type de problème correspond, en mécanique, notamment à :

- **L'étude des vibrations libres d'une structure** non amortie et non tournante. Pour cette structure, on recherche les plus petites valeurs propres ou bien celles qui sont dans un intervalle donné pour savoir si une force excitatrice peut créer une résonance. Dans ce cas, la matrice \mathbf{A} est la matrice de rigidité matérielle, notée \mathbf{K} (réelle ou complexe), éventuellement augmentée de la matrice de rigidité géométrique notée \mathbf{K}_g , si la structure est précontrainte, et \mathbf{B} est la matrice de masse ou d'inertie notée \mathbf{M} . Les valeurs propres obtenues sont les carrés des pulsations associées aux fréquences cherchées.

Le système à résoudre peut s'écrire :
$$\underbrace{(\mathbf{K} + \mathbf{K}_g)}_{\mathbf{A}} \mathbf{x} = \lambda \underbrace{\mathbf{M}}_{\mathbf{B}} \mathbf{x} \text{ où } \lambda = (2\pi f)^2 \text{ est le carré de la}$$

pulsation ω , f la fréquence propre et \mathbf{x} le vecteur de déplacement propre associé.

Si \mathbf{K} est complexe, ω et f le sont aussi.

- **La recherche de mode de flambement linéaire.** Dans le cadre de la théorie linéarisée, en supposant a priori que les phénomènes de stabilité sont convenablement décrits par le système d'équations obtenu en supposant la dépendance linéaire du déplacement par rapport au niveau de charge critique, la recherche du mode de flambement \mathbf{x} associé à ce niveau de charge critique $\mu = -\lambda$, se ramène à un problème généralisé aux valeurs propres de la forme :
$$(\mathbf{K} + \mu \mathbf{K}_g) \mathbf{x} = \mathbf{0} \Leftrightarrow \underbrace{\mathbf{K}}_{\mathbf{A}} \mathbf{x} = \lambda \underbrace{\mathbf{K}_g}_{\mathbf{B}} \mathbf{x} \text{ avec } \mathbf{K} \text{ matrice de rigidité matérielle et}$$

\mathbf{K}_g matrice de rigidité géométrique.

Attention :

Dans le code, on ne traite que les valeurs propres du problème généralisé, les λ . Pour obtenir les véritables charges critiques, les μ , il faut les multiplier par -1 .

Cet opérateur permet aussi l'étude de la **stabilité dynamique d'une structure en présence d'amortissements visqueux (et/ou quadratique) et d'effets gyroscopiques**. Cela conduit à la résolution d'un problème modal d'ordre plus élevé, dit quadratique [R5.01.02]. On recherche alors des valeurs et vecteurs propres complexes par la méthode de Lanczos après avoir effectué une réduction linéaire du problème.

- Le problème consiste à trouver $(\lambda, \mathbf{x}) \in (C, C^N)$ tels que $(\lambda^2 \mathbf{B} + \lambda \mathbf{C} + \mathbf{A}) \mathbf{x} = \mathbf{0}$ où typiquement, en mécanique linéaire, $\mathbf{A} = \mathbf{K}$ sera la matrice de rigidité, $\mathbf{B} = \mathbf{M}$ la matrice de masse et \mathbf{C} la matrice d'amortissement. Les matrices \mathbf{K} , \mathbf{M} et \mathbf{C} sont des matrices à coefficients réels. La valeur propre complexe λ est liée à la fréquence propre f et à l'amortissement réduit ξ par : $\lambda = \xi (2\pi f) \pm i (2\pi f) \sqrt{1 - \xi^2}$.

\mathbf{K} peut être aussi complexe pour simuler, en plus, un amortissement hystérétique [U2.06.03] [R5.05.04].

Pour résoudre ces problèmes modaux généralisés ou quadratiques, le *Code_Aster* propose différentes approches. Au delà de leurs spécificités numériques et fonctionnelles qui sont reprises dans le document [R5.01.01], on peut les synthétiser sous la forme du tableau ci-dessous (**les valeurs par défaut sont matérialisées en gras**).

Opérateur/ Périmètre d'application	Algorithme	Mot-clé	Avantages	Inconvénients
MODE_ITER_INV				
<i>1^{ère} phase (heuristique)</i>				
Calcul de quelques modes	Bissection	'SEPRE'		
Calcul de quelques modes	Bissection + Sécante (gén.) Muller (quad.)	'AJUSTE'	Meilleure précision	Coût calcul
Amélioration de quelques estimations	Initialisation par l'utilisateur	'PROCHE'	Reprise de valeurs propres estimées par un autre processus. Coût calcul de cette phase quasi-nul	Pas de capture de multiplicité
<i>2^{ème} phase (méthode des puissances proprement dite)</i>				
Méthode de base	Puissances inverses	'DIRECT'	Très bonne construction de vecteurs propres	Peu robuste
Option d'accélération	Quotient de Rayleigh	'RAYLEIGH'	Améliore la convergence	Coût calcul Non porté en quadratique
MODE_ITER_SIMULT				
Calcul d'une partie du spectre	Bathe & Wilson	'JACOBI'		Peu robuste Non porté en quadratique
	Lanczos (Newman- Pipano)	'TRI_DIAG'		Peu robuste
	IRAM (Sorensen)	'SORENSEN'	Robustesse accrue. Meilleures complexités calcul et mémoire. Contrôle de la qualité des modes.	

Tableau 3.1-1 : Récapitulatif des méthodes modales du *Code_Aster*

Lorsqu'il s'agit de déterminer quelques valeurs propres simples bien discriminées ou d'affiner quelques estimations, l'opérateur `MODE_ITER_INV`, est souvent bien indiqué. Par contre, pour capturer une partie significatif du spectre, on a recourt à `MODE_ITER_SIMULT`, via les méthodes dites « de sous-espace ».

C'est cette classe de méthode qui va nous intéresser ici.

Elle consiste à projeter à bon escient l'opérateur de travail afin d'obtenir un problème modal standard de taille plus réduite et comportant une matrice de forme canonique (tridiagonale ou de Hessenberg supérieure). C'est sur cette dernière que des solveurs modaux globaux vont pouvoir alors opérer (algorithmes **QR**, **QL** ou Jacobi). Ils sont en général très robustes, mais ils fournissent tout le spectre de l'opérateur traité et ils sont très coûteux. D'où l'idée de continger leurs efforts sur seulement un spectre « projeté ».

Il est d'ailleurs tout à fait recommandé de profiter des points forts des deux classes de méthode en affinant les vecteurs propres obtenus par `MODE_ITER_SIMULT`, via `MODE_ITER_INV` (`OPTION='PROCHE'`). Cela permettra de réduire la norme du résidu final (cf. [§3.7.2]).

Remarque :

On conseille fortement une lecture préalable des documentations de référence [R5.01.01], [R5.01.02]. Elle donne à l'utilisateur les propriétés et les limitations, théoriques et pratiques, des méthodes modales abordées tout en reliant ces considérations, qui peuvent parfois paraître un peu éthérées, à un paramétrage précis des options.

3.2 Opérandes `MATR_A`, `_B`, `_C`

- ◆ `MATR_A = A`
Matrice assemblée de concept [`matr_asse*_R/C`] du système à résoudre.
- ◆ `MATR_B = B`
Matrice assemblée de concept [`matr_asse*_R`] du système à résoudre.
- ◇ `MATR_C = C`
Matrice assemblée de concept [`matr_asse*_R`] du système quadratique à résoudre.

3.3 Mot clé `TYPE_RESU`

- ◇ `TYPE_RESU = / 'DYNAMIQUE' [DEFAULT]
 / 'MODE_FLAMB'`

Ce mot-clé permet de définir la nature du problème modal à traiter : recherche de fréquences de vibration (cas classique de dynamique avec ou sans amortissement) ou recherche de charges critiques (cas de la théorie du flambement linéaire). Suivant cette classe d'appartenance, les résultats sont affichés et stockés différemment dans la structure de données :

- **En dynamique**, les fréquences sont ordonnées par ordre croissant du module de leur écart au shift (cf. [§2.9], [§4.4] [R5.01.01]). C'est la valeur de la variable d'accès `NUM_ORDRE` de la structure de donnée. L'autre variable d'accès, `NUM_MODE`, est égale à la véritable position modale dans le spectre de la valeur propre (déterminée par le test de Sturm cf. [§2.5], [§2.6] [R5.01.01]).
- **En flambement**, les valeurs propres sont stockées par ordre croissant algébrique. Les variables `NUM_ORDRE` et `NUM_MODE` prennent la même valeur égale à cette ordre.

3.4 Mot clé METHODE

Trois méthodes de résolution sont disponibles pour le problème aux valeurs propres

- La méthode IRA (dite de Sorensen), permet de traiter les deux types de problèmes généralisé et quadratique. Elle est la méthode par défaut et est basée sur :
 - l'obtention d'une matrice de Hessenberg en utilisant une factorisation de type Arnoldi
 - le calcul des valeurs propres de ce problème projeté par une méthode QR
 - un certain nombre de redémarrages permettant d'affiner les valeurs propres cherchées par l'utilisateur, les autres valeurs propres nécessaires à la méthode servant de valeurs auxiliaires.
- La méthode de Lanczos, permet de traiter les deux types de problèmes généralisé et quadratique. Elle est basée sur :
 - l'obtention d'une matrice tridiagonale projetée via la méthode de Lanczos,
 - la résolution du système tridiagonal réduit par une méthode **QR**,
- La méthode itérative de Bathe et Wilson valable seulement pour le problème généralisé, est basée sur :
 - la construction à chaque itération d'un problème généralisé projeté de plus petite taille,
 - le calcul des valeurs propres de ce problème projeté par une méthode de Jacobi.

◇ METHODE = / 'SORENSEN' [DEFAULT]

On utilise la méthode de Sorensen (cf. [§5] [R5.01.01]) pour calculer les valeurs et vecteurs propres du problème généralisé ou quadratique. Cette option ne peut pas être utilisée pour un problème quadratique.

◇ / 'TRI_DIAG'

On utilise la méthode de Lanczos (puis la méthode **QR** sur le système projeté) pour calculer les valeurs et vecteurs propres du problème généralisé ou quadratique (cf. [§4] [R5.01.01]).

◇ / 'JACOBI'

On utilise la méthode de Bathe & Wilson (cf. [§6] [R5.01.01]) (puis la méthode de Jacobi sur le système projeté) pour calculer les valeurs et vecteurs propres du problème généralisé. Cette option ne peut pas être utilisée pour un problème quadratique.

3.5 Mot clé OPTION

◇ OPTION = / 'MODE_RIGIDE'
/ 'SANS' [DEFAULT]

Mot-clé utilisable seulement avec la méthode de Lanczos pour un problème modal généralisé. Il permet de détecter et de calculer au préalable, par une méthode algébrique les modes de corps de rigide (modes associés à une valeur propre nulle) (cf. [§5.5.4] [R5.01.01]). Ils sont utilisés par la suite pour calculer les autres modes avec l'algorithme de Lanczos. Ils sont fournis à l'utilisateur seulement s'ils font partie des modes demandés. Si les modes de corps rigide sont calculés sans utiliser cette option, les valeurs propres calculées par l'algorithme de Lanczos ne sont pas nulles mais très voisines de zéro.

3.6 Mot clé CALC_FREQ

◇ CALC_FREQ = _F(...

Mot-clé facteur pour la définition des paramètres de calcul des valeurs propres et de leur nombre.

3.6.1 Opérande OPTIONS

◇ OPTION =

'BANDE'

On recherche toutes les valeurs propres dans une bande donnée. Cette bande est définie par l'argument de `FREQ` : (f_1 f_2) ou par celui de `CHAR_CRIT` : (λ_1 λ_2).

'CENTRE'

Cette option n'est pas utilisable avec un problème modal quadratique. On recherche les `NMAX_FREQ` valeurs propres les plus proches de la fréquence f (argument du mot-clé `FREQ` : f) ou les plus proches de la charge critique λ (argument du mot-clé `CHAR_CRIT` : λ).

'PLUS_PETITE'
[DEFAULT]

On recherche les `NMAX_FREQ` plus petites valeurs propres.

Voir [§2.9] et [§4.4] [R5.01.01].

3.6.2 Opérande APPROCHE

◇ APPROCHE = / 'REEL' [DEFAULT]
/ 'IMAG'
/ 'COMPLEXE'

Ce mot-clé définit le type d'approche (réelle, imaginaire ou complexe) pour le choix du pseudo-produit scalaire du problème quadratique(cf. [§5.5.2] [R5.01.02]). En général la valeur par défaut (réel) est valide.

Cet opérande n'a de sens que pour l'analyse des vibrations libres d'une structure amortie (modes propres complexes ; le mot-clé `MATR_C` doit être renseigné). En flambement, cela n'a aucun intérêt.

3.6.3 Opérande FREQ

◇ FREQ = l_f

Liste des fréquences (ne peut être utilisé que si `TYPE_RESU` = 'DYNAMIQUE') : son utilisation dépend de l'OPTION choisie.

OPTION = 'BANDE'

On attend deux valeurs (f_1 f_2) qui définissent la bande de recherche,

OPTION = 'CENTRE'

On attend une seule valeur de fréquence,

Les valeurs stipulées sous ce mot-clé doivent être positives.

3.6.4 Opérande AMOR_REDUIT

◇ AMOR_REDUIT = l_a

Valeur de l'amortissement réduit qui permet de définir la valeur propre complexe autour de laquelle on cherche les valeurs propres les plus proches. (ne peut être utilisé que si `TYPE_RESU` = 'DYNAMIQUE' et `MATR_C` renseigné).

OPTION = 'CENTRE'

On attend une seule valeur d'amortissement réduit,

La valeur stipulée sous ce mot-clé doit être positive et être comprise entre 0 et 1. En flambement, cela n'a aucun intérêt.

3.6.5 Opérande CHAR_CRIT

◇ CHAR_CRIT = l_c

Liste des charges critiques (ne peut être utilisé que si TYPE_RESU = 'MODE_FLAMB') : son utilisation dépend de l'OPTION choisie.

OPTION = 'BANDE' On attend deux valeurs (λ_1 λ_2) qui définissent la bande de recherche,
OPTION = 'CENTRE' On attend une seule valeur de charge critique,

Les valeurs stipulées sous ce mot-clé sont positives ou négatives.

3.6.6 Opérande NMAX_FREQ

◇ NMAX_FREQ = nf (10) [DEFAULT]

Nombre maximum de valeurs propres à calculer.

Ce mot-clé est ignoré avec l'option 'BANDE' car on calcule alors toutes les valeurs propres contenues dans la bande stipulée.

Dans les deux cas, si nf est strictement supérieur au nombre de « ddl actifs », nactif (cf. [§2.2] [R5.01.01]), alors on le force à prendre cette valeur plafond.

3.6.7 Opérande DIM_SOUS_ESPACE

◇ DIM_SOUS_ESPACE = des
◇ COEF_DIM_ESPACE = mse
EXCLUS ('DIM_SOUS_ESPACE' , 'COEF_DIM_ESPACE')

Si le mot-clé DIM_SOUS_ESPACE n'est pas renseigné ou est initialisé à une valeur strictement inférieure au nombre de fréquences demandées nf, l'opérateur calcule automatiquement une dimension admissible pour le sous-espace de projection (cf. [§5.2] de ce document et [§4.3], [§5.5.2], [§6.5.3], [§7.3.1] [R5.01.01]) à l'aide COEF_DIM_ESPACE.

Grâce à la donné de ce facteur multiplicatif, mse, on peut projeter sur un espace dont la taille est proportionnelle au nombre de fréquences contenues dans l'intervalle d'étude. Dans l'encapsulation de MODE_ITER_SIMULT, MACRO_MODE_MECA [U4.52.02], on peut donc optimiser la taille des sous-espaces qui reste proportionnelle au nombre de fréquences recherchées : les sous-espaces riches en valeurs propres ne pénalisent ainsi pas les plus pauvres (en terme de CPU).

On peut cependant fixer arbitrairement la taille de ce sous-espace, via la valeur des prise par le mot-clé DIM_SOUS_ESPACE (qui doit être supérieure à nf pour être prise en compte).

Dans les deux cas, si la taille du sous-espace de projection ndim est strictement supérieure au nombre de « ddl actifs », nactif (cf. [§2.2] [R5.01.01]), alors on la force à prendre cette valeur plafond.

Remarques :

- Si on utilise la méthode de Sorensen (IRAM) et que $ndim - nf < 2$, des impératifs numérico-informatiques forcent à imposer $ndim = nf + 2$.
- En quadratique on travaille sur un problème réel de taille double : $2*nf$, $2*ndim$.

3.6.8 Opérandes d'IRAM (si METHODE = 'SORENSEN')

◇ PREC_SOREN = pso (0.) [DEFAULT]

Remarque :

La méthode considère alors qu'elle doit travailler avec la plus petite précision possible, le « zéro machine ». Pour en avoir un ordre de grandeur, en double précision sur les machines standards, cette valeur est proche de $2.22 \cdot 10^{-16}$

◇ NMAX_ITER_SOREN = nso (20) [DEFAULT]

◇ PARA_ORTHO_SOREN = porso (0.717) [DEFAULT]

Il s'agit de paramètres d'ajustement de la précision requise sur les modes (par défaut, la précision machine est choisie), du nombre de redémarrages autorisé de la méthode de Sorensen (cf. [§5.4.2] et [§6.4] [R5.01.01]) et du coefficient d'orthogonalisation de l'IGSM de Kahan-Parlett (cf. [§11.4] [R5.01.01]).

Remarque :

Lors des premiers passages, il est fortement conseillé de ne pas modifier ces paramètres qui concernent plutôt les arcanes de l'algorithme et qui sont initialisés empiriquement à des valeurs standards.

3.6.9 Opérandes de la méthode de Lanczos (si METHODE = 'TRI_DIAG')

◇ PREC_ORTHO = po (1.10^{-12}) [DEFAULT]

◇ NMAX_ITER_ORTHO = nio (5) [DEFAULT]

◇ PREC_LANCZOS = pl (1.10^{-8}) [DEFAULT]

◇ NMAX_ITER_QR = nim (30) [DEFAULT]

Les deux premiers paramètres permettent, respectivement, d'ajuster la précision d'orthogonalisation et le nombre de réorthogonalisations dans la méthode de Lanczos pour obtenir des vecteurs indépendants engendrant le sous-espace (cf. [§5.5.1] [R5.01.01]).

Le troisième est un paramètre d'ajustement pour déterminer la nullité d'un terme sur la surdiagonale de la matrice tridiagonale caractérisant le problème réduit obtenu par la méthode de Lanczos. C'est juste un critère de déflation et non, contrairement à ce que pourrait laisser croire son nom, un critère de qualité des modes (cf. [§5.4.1] [R5.01.01]).

Le dernier fixe le nombre d'itérations maximum pour la résolution du système réduit par la méthode QR ([§5.5.2] et [§10] [R5.01.01]).

Remarque :

Lors des premiers passages, il est fortement conseillé de ne pas modifier ces paramètres qui concernent plutôt les arcanes de l'algorithme et qui sont initialisés empiriquement à des valeurs standards.

3.6.10 Opérandes de la méthode de Bathe & Wilson (si METHODE = 'JACOBI')

◇ PREC_BATHE = pbat (1.10^{-10}) [DEFAULT]

◇ NMAX_ITER_BATHE = nbat (40) [DEFAULT]

◇ PREC_JACOBI = pjaco (1.10^{-2}) [DEFAULT]

◇ NMAX_ITER_JACOBI = njaco (12) [DEFAULT]

Les deux premiers paramètres permettent, respectivement, d'ajuster la précision de convergence et le nombre maximum d'itérations permises de la méthode de Bathe & Wilson (cf. [§7] [R5.01.01]).

Les deux autres permettent d'ajuster la précision de la convergence et le nombre maximum d'itérations permises par la méthode de JACOBI (cf. [§12] [R5.01.01]) qui permet d'exhumer les modes propres de la matrice projetée par la méthode précédente.

Remarque :

Lors des premiers passages, il est fortement conseillé de ne pas modifier ces paramètres qui concernent plutôt les arcanes de l'algorithme et qui sont initialisés empiriquement à des valeurs standards.

3.6.11 Opérands SEUIL_FREQ, PREC_SHIFT et NMAX_ITER_SHIFT

◇ PREC_SHIFT = ps (0.05) [DEFAULT]
 ◇ SEUIL_FREQ = sf (0.01) [DEFAULT]
 ◇ NMAX_ITER_SHIFT = ns (5) [DEFAULT]

Pour les trois options possibles 'PLUS_PETITE', 'BANDE' ou 'CENTRE', on effectue une factorisation LDL^T de la matrice $\left(A - (2\pi f_*)^2 B \right)$. f_* dépend de la méthode utilisée. Si f_* est détectée comme étant une fréquence propre ou étant située à proximité de fréquences propres (perte de plus de $\text{ndeci}=8$ décimales lors de la factorisation des matrices), la fréquence f_* est alors modifiée (cf. [§2.6] et [§2.9] [R5.01.01]):

$$f_*^- = f_* \times (1 - ps) \text{ ou } f_*^+ = f_* \times (1 + ps)$$

Dans le cas où $\left(A - (2\pi f_*)^2 B \right)$ est non factorisable LDL^T et $(|f_*| \leq sf)$, on effectue la modification suivante : $f_*^- = -sf$. On considère alors que f_* est associée à un mode de corps rigide. La modification de cette fréquence permet a priori de comptabiliser tous les modes de corps rigide. On n'effectue pas plus de ns modifications de la valeur f_* .

Dans le cas du flambement linéaire, la transposition est immédiate en remplaçant f_* (fréquence de vibration) par λ_* (charge critique), $(2\pi f_*)^2$ par λ_* et sf par $(2\pi sf)^2$.

Remarque :

Lors des premiers passages, il est fortement conseillé de ne pas modifier ces paramètres qui concernent plutôt les arcanes de l'algorithme et qui sont initialisés empiriquement à des valeurs standards.

3.6.12 Opérande NPREC_SOLVEUR

◇ NPREC_SOLVEUR = ndeci (8) [DEFAULT]

ndeci représente le nombre de décimales qu'on s'autorise à perdre lors de la factorisation de la matrice shiftée $\left(A - (2\pi f_*)^2 B \right)$ ou $\left(A - \lambda B \right)$. Si on perd plus de ndeci décimales, la matrice est considérée comme non inversible (cf. [§2.6] et [§2.9] [R5.01.01]).

Remarque :

Lors des premiers passages, il est fortement conseillé de ne pas modifier ce paramètre qui concerne plutôt une arcane de l'algorithme et qui est initialisé empiriquement à une valeur standard.

3.7 Mot clé VERI_MODE

◇ VERI_MODE = _F(...

Mot clé facteur pour la définition des paramètres de la vérification des modes propres ([§2.9] [R5.01.01]).

3.7.1 Opérateur STOP_ERREUR

◇ STOP_ERREUR = / 'OUI' [DEFAULT]
/ 'NON'

Permet d'indiquer à l'opérateur s'il doit s'arrêter ('OUI') ou continuer ('NON') dans le cas où l'un des critères SEUIL ou STURM n'est pas vérifié.
Par défaut le concept de sortie n'est pas produit.

3.7.2 Opérateur SEUIL

◇ SEUIL = r (1.10⁻⁶) [DEFAULT]

Seuil de tolérance pour la norme d'erreur relative du mode au dessus duquel le mode est considéré comme faux.
La norme d'erreur relative du mode est :

$$\frac{\|(\mathbf{A} - \lambda \mathbf{B})\mathbf{x}\|_2}{\|\mathbf{Ax}\|_2}, \text{ pour } \lambda \neq 0 \text{ pour le problème généralisé et}$$

$$\frac{\|(\lambda^2 \mathbf{B} + \lambda \mathbf{C} - \mathbf{A})\mathbf{x}\|_2}{\|\mathbf{Ax}\|_2}, \text{ pour le problème quadratique}$$

3.7.3 Opérateur STURM

◇ STURM = / 'OUI' [DEFAULT]
/ 'NON'

Vérification dite de STURM ('OUI') permettant de s'assurer que l'algorithme utilisé dans l'opérateur a déterminé le nombre exact de valeurs propres dans l'intervalle de recherche ([§2.5] [§2.6] [R5.01.01]).

3.7.4 Opérateur PREC_SHIFT

◇ PREC_SHIFT = prs (0.05) [DEFAULT]

Ce paramètre (qui est un pourcentage) permet de définir un intervalle contenant les valeurs propres calculées, pour lequel la vérification de Sturm sera effectuée([§2.6] [R5.01.01]).

3.8 Opérandes SENSIBLITE

◇ SENSIBLITE =

Active le calcul de la dérivée des modes par rapport à un paramètre sensible du problème.
Il est à noter qu'à l'heure actuelle, la dérivée des modes multiples n'est pas disponible, car elle pose des problèmes théoriques et pratiques particuliers.

Le document [U4.50.02] précise le fonctionnement du mot clé.

3.9 Opérande STOP_FREQ_VIDE

◇ STOP_FREQ_VIDE = / 'OUI' [DEFAULT]
/ 'NON'

'OUI' arrête le calcul si aucune valeur propre n'est détectée dans la bande stipulée par l'utilisateur : une exception (nommée *BandeFrequenceVide*) est émise. Elle peut être traitée pour continuer le déroulement de l'étude. On peut trouver un exemple sous le cas test SDLL11a :

```
try:
    MODEL=MODE_ITER_SIMULT( MATR_A=K_ASSE, MATR_B=M_ASSE,
                             CALC_FREQ=_F(
OPTION = 'BANDE',
FREQ = ( 100., 200. )))
except aster.BandeFrequenceVideError:
    MODEL=MODE_ITER_SIMULT( MATR_A=K_ASSE, MATR_B=M_ASSE,
                             CALC_FREQ=_F(
OPTION = 'BANDE',
FREQ = ( 200., 3500., )))
```

'NON' n'arrête le calcul (émission seulement d'une ALARME) si aucune valeur propre n'est détectée dans la bande stipulée par l'utilisateur.

Ce mot-clé est utilisé dans la macro-commande MACRO_MODE_MECA [U4.52.02] afin de permettre l'absence de valeurs propres dans une bande de recherche.

3.10 Opérande INFO

◇ INFO = / 1 [DEFAULT]
/ 2

Indique le niveau d'impression dans le fichier MESSAGE.

- 1 : Impression sur le fichier 'MESSAGE' des valeurs propres, de leur position modale, de l'amortissement réduit, de la norme d'erreur a posteriori et de certains paramètres utiles pour suivre le déroulement du calcul (Cf. [§5.2])
- 2 : Impression plutôt réservée aux développeurs.

3.11 Opérande TITRE

◇ TITRE = ti

Titre attaché au concept produit par cet opérateur [U4.03.01].

4 Phase de vérification

On vérifie selon les options :

OPTION = 'BANDE'

l'argument du mot-clé FREQ ou du mot-clé CHAR_CRIT doit fournir exactement **deux** valeurs,

OPTION = 'CENTRE'

l'argument du mot-clé FREQ ou du mot-clé CHAR_CRIT doit fournir exactement **une** seule valeur,

OPTION = 'PLUS_PETITE'

l'argument du mot clé FREQ ou du mot-clé CHAR_CRIT, est ignoré.

Si les précisions et les nombres maximaux d'itérations sont irréalistes (par exemple des précisions inférieures à la précision machine ou des nombres d'itérations négatifs), on n'effectue pas le calcul.

5 Phase d'exécution

5.1 Vérification

Les matrices **A**, **B** (et **C**) arguments des mots-clé `MATR_A` et `MATR_B` (et `MATR_C`), doivent être cohérentes entre elles (c'est à dire s'appuyer sur la même numérotation et le même mode de stockage).

5.2 Actions par défaut

Si le mot-clé `DIM_SOUS_ESPACE` n'est pas renseigné ou est initialisé à une valeur strictement inférieure au nombre de fréquences demandées `nf` (opérande `NMAX_FREQ`), l'opérateur calcule automatiquement une dimension admissible pour le sous-espace de projection via les formules empiriques (cf [§3.6.7]) :

```
METHODE = 'SORENSEN'  
ndim = MIN ( MAX ( 2+nf , mse*nf ) , nactif ) avec mse = 2 par défaut.
```

```
METHODE = 'TRI_DIAG'  
ndim = MIN ( MAX ( 7+nf , mse*nf ) , nactif ) avec mse = 4 par défaut.
```

```
METHODE = 'JACOBI'  
ndim = MIN ( MAX ( 7+nf , mse*nf ) , nactif ) avec mse = 2 par défaut.
```

où `nactif` est le nombre de ddl actifs (c'est-à-dire le nombre total de ddl moins le nombre de ddls de LAGRANGE et moins le nombre de relations linéaires qui lient des ddls entre eux, cf. [§2.2] [R5.01.01]) et `mse` est le facteur de proportionnalité fixé par `COEF_DIM_ESPACE`.

Si l'on résout un problème quadratique aux valeurs propres, la dimension du sous-espace est doublée.

Les valeurs de ces différents paramètres sont imprimées dans le fichier `MESSAGE`.

6 Paramètres modaux / Norme des modes / Position modale

En sortie de cet opérateur, les modes propres réels ou complexes sont normalisés à la plus grande des composantes qui n'est pas un multiplicateur de LAGRANGE. Pour choisir une autre norme, il faut utiliser la commande `NORM_MODE` [U4.52.11].

Dans le cas d'un calcul dynamique, la structure de données `mode_meca_*`, contient, en plus des fréquences de vibration et des déformées modales associées, des paramètres modaux (masse généralisée, raideur généralisée, facteur de participation, masse effective). On trouvera la définition de ces paramètres dans [R5.01.03].

Dans le cas d'un calcul de flambement linéaire, la structure de données `mode_flamb`, ne contient que les charges critiques et les déformées associées.

Dans le cas d'un calcul dynamique, la position modale des modes correspond à la position du mode dans l'ensemble du spectre défini par les matrices initiales.

Dans le cas d'un calcul de flambement linéaire, les positions modales des charges critiques sont attribuées de 1 à `nf` (`nf` étant le nombre de charges critiques calculées) en classant les charges critiques par ordre croissant algébrique. Toutes les positions modales sont donc positives.

7 Impression des résultats

Pour afficher les paramètres modaux associés à chaque mode et les coordonnées des modes, il faut utiliser l'opérateur `IMPR_RESU` [U4.91.01] de la manière suivante :

- Affichage des paramètres modaux seulement sous forme de table :

```
IMPR_RESU      (  RESU = _F( RESULTAT = mode,
                          TOUT_PARA = 'OUI',
                          TOUT_CHAM = 'NON' ) ) ;
```

- Affichage des paramètres modaux et des vecteurs propres :

```
IMPR_RESU      (  RESU = _F( RESULTAT = mode,
                          TOUT_PARA = 'OUI',
                          TOUT_CHAM = 'OUI' ) ) ;
```

8 Tri de modes / Caractérisation de `mode_meca_*`

Par exemple, lors de sollicitations sismiques en analyse modale, la base modale utilisée doit contenir les modes qui ont une masse effective unitaire importante dans la direction du séisme.

La commande `EXTR_MODE` [U4.52.12] permet d'extraire dans une structure de données de type `mode_meca_*` des modes qui vérifient un certain critère et de concaténer plusieurs structures de données de type `mode_meca_*`.

Une macro-commande, permettant d'enchaîner les commandes `MODE_ITER_SIMULT`, `NORM_MODE` et `EXTR_MODE` a été créée : `MACRO_MODE_MECA` [U4.52.02].

9 Exemples

9.1 Calcul des 5 modes propres les plus proches d'une fréquence donnée (100 Hz)

```
mode = MODE_ITER_SIMULT      (  MATR_A = rigid,
                                MATR_B = masse,
                                CALC_FREQ = _F(  OPTION      = 'CENTRE',
                                                  FREQ          = 100.,
                                                  NMAX_FREQ     = 5 )
                                ) ;
```

9.2 Calcul des charges critiques contenues dans une bande

```
mode = MODE_ITER_SIMULT      (  MATR_A = rigid,
                                MATR_B = riggeo,
                                TYPE_RESU = 'MODE_FLAMB',
                                CALC_FREQ = _F(  OPTION      = 'BANDE',
                                                  CHAR_CRIT    = (-1.E8 , 1.5E8)
                                )
                                ) ;
```