Opérateur CALC_MODES

1 But

Calculer les modes propres de vibrations ou les modes de flambement d'Euler, d'une structure mécanique.

L'opérateur CALC_MODES résout pour cela un problème modal qui peut être soit généralisé (GEP) [R5.01.01], soit quadratique (QEP) [R5.01.02].

L'utilisateur précise le critère de recherche des modes (par exemple : sur une bande ; à proximité de valeurs données ; etc.) grâce au mot-clé OPTION. Selon la valeur de ce mot-clé, la méthode générale de calcul (itérations simultanées ou puissances inverses) est alors déterminée automatiquement.

Au sein de chacune de ces deux méthodes générales, des variantes existent :
- cas des itérations simultanées : méthode de type sous-espace (Bathe & Wilson, Lanczos ou Sorensen), méthode globale de type QR (QZ pour les petits problèmes) ;
- cas des puissances inverses : méthode directe ou accélérée par le quotient de Rayleigh.

Ces variantes sont accessibles avec un mot-clé facteur SOLVEUR_MODAL.

Dans le cas d'un calcul de modes propres de vibrations, des post-traitements peuvent également être réalisés : normalisation des modes selon un critère donné, filtrage des modes selon un critère donné, ...

Cet opérateur produit un concept mode_meca_* (cas dynamique) ou mode_flamb (cas flambement d'Euler, seulement en GEP) ou mode_gene suivant la valeur renseignée dans le mot-clé TYPE_RESU et le type des matrices d'entrée du problème modal.
Dans une première approche on peut se contenter de renseigner les paramètres suivants : OPTION pour définir le critère de recherche des modes, TYPE_RESU, les matrices d'entrée du problème modal MATR_*, CALC_FREQ (ou CALC_CHAR_CRIT).

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

Table des Matières

1 But....1
2 Syntaxe....4
3 Opérandes....11
3.1 Principes....11
3.2 Mot-clé TYPE_RESU....15
3.3 Opérandes MATR_RIGI/MATR_A/MATR_MASS/MATR_RIGI_GEOM /MATR_B/MATR_AMOR 15
3.4 Opérande OPTION....16
3.5 Mot-clé facteur SOLVEUR_MODAL....18
3.5.1 Mots-clés associés à la méthode des itérations simultanées....18
3.5.1.1 Mot-clé METHODE....18
3.5.1.2 Paramètres liés à la méthode de résolution....19
3.5.1.3 Mot-clé APPROCHE....20
3.5.1.4 Mots-clé DIM_SOUS_ESPACE et COEF_SOUS_ESPACE....21
3.5.2 Paramètres associés à la méthode des puissances inverses....21
3.5.2.1 Opérandes de la bissection (si OPTION='SEPARE' ou 'AJUSTE')....21
3.5.2.2 Opérandes de la sécante (si OPTION='AJUSTE')....21
3.5.2.3 Paramètres de calcul de la deuxième phase de calcul de la méthode des puissances inverses....22
3.6 Mot-clé CALC_FREQ (si TYPE_RESU='DYNAMIQUE')....22
3.6.1 Opérande FREQ (uniquement si OPTION='BANDE' ou 'CENTRE' ou 'PROCHE' ou 'SEPARE' ou 'AJUSTE')....22
3.6.2 Opérande AMOR_REDUIT (uniquement si OPTION='CENTRE' ou 'PROCHE')....24
3.6.3 Opérande NMAX_FREQ (uniquement si OPTION='PLUS_PETITE' ou 'PLUS_GRANDE' ou 'CENTRE' ou 'SEPARE' ou 'AJUSTE')....24
3.7 Mot-clé CALC_CHAR_CRIT (si TYPE_RESU='MODE_FLAMB' ou 'GENERAL')....24
3.7.1 Opérande CHAR_CRIT (uniquement si OPTION='BANDE' ou 'CENTRE' ou 'PROCHE' ou 'SEPARE' ou 'AJUSTE')....24
3.7.2 Opérande NMAX_CHAR_CRIT (uniquement si OPTION='PLUS_PETITE' ou 'CENTRE' ou 'SEPARE' ou 'AJUSTE')....26
3.8 Opérandes communes à CALC_FREQ et CALC_CHAR_CRIT : SEUIL_FREQ / SEUIL_CHAR_CRIT, PREC_SHIFT, NMAX_ITER_SHIFT....26
3.9 Mot-clé facteur SOLVEUR....28
3.10 Mot-clé VERI_MODE....28
3.10.1 Opérande STOP_ERREUR....28
3.10.2 Opérande SEUIL....29
3.10.3 Opérande STURM....29
3.10.4 Opérande PREC_SHIFT (uniquement pour la méthode des itérations simultanées)....30

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

3.11 Opérande STOP_BANDE_VIDE (uniquement pour la méthode des itérations simultanées)....30
3.12 Opérande NIVEAU_PARALLELISME....31
3.13 Mot-clé AMELIORATION....35
3.14 Mots-clés pour le post-traitement : NORM_MODE, FILTRE_MODE, IMPRESSION....35
3.14.1 Mot-clé facteur NORM_MODE....35
3.14.2 Mot-clé facteur FILTRE_MODE....35
3.14.3 Mot-clé IMPRESSION....35
3.15 Opérande INFO....36
3.16 Opérande TITRE....36
4 Phase de vérification....37
5 Phase d'exécution....38
6 Paramètres modaux/ Norme des modes/ Position modale....39
7 Optimisation des performances CPU....39
7.1 Parallélisme du solveur linéaire....39
7.2 Calcul des modes par sous-bandes....39
8 Impression des résultats....40
9 Tri de modes / Caractérisation de mode_meca_*....41
10 Exemples....42
10.1 Calcul des 5 modes propres les plus proches d'une fréquence donnée....42
10.2 Calcul des charges critiques contenues dans une bande....42
10.3 Chaînage INFO_MODE+CALC_MODES (extrait de SDLS504a)....42
10.4 Calcul des fréquences propres contenues dans la bande [50 ; 150] Hz....42
10.5 Calcul des fréquences propres les plus proches de 20 et 50 Hz....42
10.6 Découpage en plusieurs sous-bandes....43

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

2 Syntaxe

```
mode_  [*]=CALC_MODES (

# Type de problème
      ◊  TYPE_RESU= /'DYNAMIQUE'                                  [DEFAUT]
                    /'MODE_FLAMB'
                    /'GENERAL'

# Critère de recherche des modes
      ◊  OPTION= /'CENTRE'
                 /'BANDE'
                 /'PLUS_PETITE'                                   [DEFAUT]
                 /'PLUS_GRANDE'
                 /'TOUT'
                 /'PROCHE'              (si pas de mode multiple)
                 /'SEPARE'              (uniquement en GEP)
                 /'AJUSTE'              (uniquement en GEP)

# Divers
      ◊  STOP_BANDE_VIDE= /'OUI'                                  [DEFAUT]
                          /'NON'
      ◊  AMELIORATION= /'NON'                                     [DEFAUT]
                       /'OUI'

# Normalisation des modes
      ◊  NORM_MODE = _F   (
                             ◊  /  NORME =   /  'EUCL_TRAN'
                                             /  'MASS_GENE'
                                             /  'RIGI_GENE'
                                             /  'TRAN'
                                             /  'TRAN_ROTA'      [DEFAUT]
                                             /  'EUCL'
                             ◊  INFO =       /  1                [DEFAUT]
                                             /  2
                          )
```

Si TYPE_RESU='DYNAMIQUE'

Caractéristiques du calcul

```
   (        ♦  MATR_RIGI=A                    /[matr_asse_DEPL_R]
                                              /[matr_asse_DEPL_C]
                                              /[matr_asse_PRES_R]
                                              /[matr_asse_GENE_R]
                                              /[matr_asse_GENE_C]
                                              /[matr_asse_ TEMP _R]
            ♦  MATR_MASS=B                    /[matr_asse_DEPL_R]
                                              /[matr_asse_PRES_R]
                                              /[matr_asse_GENE_R]
                                              /[matr_asse_ TEMP _R]
            ◊  MATR_AMOR=C                    /[matr_asse_DEPL_R]
                                              /[matr_asse_GENE_R]

            ◊  CALC_FREQ=_F(
      # Si OPTION='PLUS_PETITE'
                  ◊  NMAX_FREQ=/10                          [DEFAUT]
                               /nf                          [I]
      # Si OPTION='PLUS_GRANDE'
```


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

```
                ◊  NMAX_FREQ=/1                          [DEFAUT]
                           /nf                           [I]
# Si OPTION='CENTRE'
                ♦  FREQ=l_f                              [l_R]
                ◊  AMOR_REDUIT=l_a                       [l_R]
                ◊  NMAX_FREQ=/10                         [DEFAUT]
                           /nf                           [I]
# Si OPTION='BANDE'
                ♦  FREQ=l_f                              [l_R]
                ◊  TABLE_FREQ=table_f                    [table_sdaster]

# Si OPTION='PROCHE'
        ou 'SEPARE'
        ou 'AJUSTE'
                ♦  FREQ=l_f                              [l_R]
                ◊  NMAX_FREQ=nf                          [I]
                ◊  AMOR_REDUIT=l_amor                    [l_R]


# Pour pré et post-traitements
        ◊  PREC_SHIFT=/0.05                              [DEFAUT]
                    /p_shift                             [R]
        ◊  NMAX_ITER_SHIFT=/3                            [DEFAUT]
                           /n_shift                      [I]
        ◊  SEUIL_FREQ=/1.E-2                             [DEFAUT]
                    /f_seuil                             [R]

                          )
```


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

Si TYPE_RESU='MODE_FLAMB'

Caractéristiques du calcul

```
(       ♦ MATR_RIGI=A                          /[matr_asse_DEPL_R]
                                               /[matr_asse_PRES_R]
                                               /[matr_asse_GENE_R]
        ♦ MATR_RIGI_GEOM=B                     /[matr_asse_DEPL_R]
                                               /[matr_asse_PRES_R]

        ◊ CALC_CHAR_CRIT =_F(
# Si OPTION='PLUS_PETITE'
            ◊ NMAX_CHAR_CRIT= /10                          [DEFAUT]
                              /nf                               [I]
# Si OPTION='CENTRE'
            ♦ CHAR_CRIT= l_c                                 [l_R]
            ◊ NMAX_CHAR_CRIT= /10                          [DEFAUT]
                              /nf                               [I]

# Si OPTION='BANDE'
            ♦ CHAR_CRIT=l_c                                  [l_R]
            ◊ TABLE_CHAR_CRIT=table_c              [table_sdaster]

# Si OPTION='PROCHE'
       ou 'SEPARE'
       ou 'AJUSTE'
            ♦ CHAR_CRIT=l_f                                  [l_R]
            ◊ NMAX_CHAR_CRIT=nf                              [I]


# Pour pré et post-traitements
            ◊ PREC_SHIFT=/0.05                             [DEFAUT]
                        /p_shift                           [R]
            ◊ NMAX_ITER_SHIFT=/3                           [DEFAUT]
                             /n_shift                      [I]
            ◊ SEUIL_CHAR_CRIT=/1.E-2                       [DEFAUT]
                             /c_seuil                      [R]
                          )
```


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

Si TYPE_RESU='GENERAL'

Caractéristiques du calcul

```
(       ♦ MATR_A=A                      /[matr_asse_DEPL_R]
                                        /[matr_asse_PRES_R]
                                        /[matr_asse_GENE_R]
        ♦ MATR_B=B                      /[matr_asse_DEPL_R]
                                        /[matr_asse_PRES_R]
                                        /[matr_asse_GENE_R]

  ◊ CALC_CHAR_CRIT =_F(
# Si OPTION='PLUS_PETITE'
      ◊ NMAX_CHAR_CRIT= /10             [DEFAUT]
                        /nf             [I]
# Si OPTION='CENTRE'
      ♦ CHAR_CRIT=l_c                   [l_R]
      ◊ NMAX_CHAR_CRIT= /10             [DEFAUT]
                        /nf             [I]

# Si OPTION='BANDE' (uniquement GEP à matrices symétriques réelles)
      ♦ CHAR_CRIT=l_c                   [l_R]
      ◊ TABLE_CHAR_CRIT=table_c         [table_sdaster]

# Si OPTION='PROCHE'
     ou 'SEPARE'
     ou 'AJUSTE'
      ♦ CHAR_CRIT=l_f                   [l_R]
      ◊ NMAX_CHAR_CRIT=nf               [I]

 # Pour pré et post-traitements
      ◊ PREC_SHIFT=/0.05                [DEFAUT]
                  /p_shift              [R]
      ◊ NMAX_ITER_SHIFT=/3              [DEFAUT]
                       /n_shift         [I]
      ◊ SEUIL_CHAR_CRIT=/1.E-2          [DEFAUT]
                       /c_seuil         [R]
                    )
```


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

 # Solveur modal

◊ SOLVEUR_MODAL =_F(

 # Si OPTION = 'PLUS_PETITE', 'PLUS_GRANDE', 'CENTRE', 'BANDE', 'TOUT' : choix de la méthode des itérations simultanées

◊ METHODE= /'SORENSEN' **[DEFAUT]**
/'TRI_DIAG' (uniquement GEP/QEP symétriques réels)
/'JACOBI' (sauf en QEP)
/'QZ' (problème de petites tailles $<10^3$ degrés de liberté)

Paramétrage interne des méthodes

Si METHODE='SORENSEN'
◊ PREC_SOREN=/0 **[DEFAUT]**
/pso [R]
◊ NMAX_ITER_SOREN= /20 **[DEFAUT]**
/nso [I]
◊ PARA_ORTHO_SOREN=/0.717 **[DEFAUT]**
/porso [R]

Si METHODE='TRI_DIAG'
◊ PREC_ORTHO=/1.E-12 **[DEFAUT]**
/po [R]
◊ NMAX_ITER_ORTHO=/5 **[DEFAUT]**
/nio [I]
◊ PREC_LANCZOS=/1.E-8 **[DEFAUT]**
/pl [R]
◊ NMAX_ITER_QR=/30 **[DEFAUT]**
/nim [I]
◊ MODE_RIGIDE=/'NON' **[DEFAUT]**
/'OUI'

Si METHODE='JACOBI'
◊ PREC_BATHE=/1.E-10 **[DEFAUT]**
/pbat [R]
◊ NMAX_ITER_BATHE=/40 **[DEFAUT]**
/nbat [I]
◊ PREC_JACOBI=/1.E-2 **[DEFAUT]**
/pjaco [R]
◊ NMAX_ITER_JACOBI=/12 **[DEFAUT]**
/njaco [I]

Si METHODE='QZ'
◊ TYPE_QZ=/'QZ_SIMPLE' **[DEFAUT]**
/'QZ_EQUI'
/'QZ_QR' (si GEP à matrices SPD)

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

 # Si OPTION = 'PROCHE', 'SEPARE', 'AJUSTE' : choix des paramètres de la méthode des puissances inverses

Si OPTION='SEPARE' ou 'AJUSTE' (uniquement en GEP)
◊ NMAX_ITER_SEPARE= /30 **[DEFAUT]**
/nis [I]

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

```
      ◊    PREC_SEPARE= /1.E-4                          [DEFAUT]
                        /ps                            [R]

    # Si OPTION='AJUSTE'
      ◊    NMAX_ITER_AJUSTE= /15                        [DEFAUT]
                             /nia                      [I]
      ◊    PREC_AJUSTE= /1.E-4                          [DEFAUT]
                        /pa                            [R]

     # Paramètres de la seconde phase de calcul des puissances inverses
      ◊    OPTION_INV= /'DIRECT'                        [DEFAUT]
                      /'RAYLEIGH'  (uniquement en GEP)
      ◊    NMAX_ITER_INV= /30                           [DEFAUT]
                          /nim                         [I]
      ◊    PREC_INV= /1.E-5                             [DEFAUT]
                     /pm                               [R]
          )
```

 # Pour vérifications finales

```
      ◊  VERI_MODE=_F(
                    ◊  STOP_ERREUR=/'OUI'               [DEFAUT]
                                   /'NON'
                    ◊  SEUIL= /1.E-6    [DEFAUT pour 'PLUS_PETITE'/
'PLUS_GRANDE'/'CENTRE'/'BANDE'/'TOUT']
                              /1.E-2    [DEFAUT pour 'PROCHE'/'SEPARE'/
'AJUSTE']
                              /r                        [R]
                    ◊  PREC_SHIFT=/0.05                 [DEFAUT]
                                  /prs                  [R]
                    ◊  STURM= /'OUI'       [DEFAUT pour 'PLUS_PETITE'/
'PLUS_GRANDE'/'CENTRE'/'BANDE'/'TOUT']
                              /'NON'       [DEFAUT pour 'PROCHE'/'SEPARE'/
'AJUSTE']
                          # si TYPE_RESU='DYNAMIQUE' et OPTION='BANDE' :
                              /'GLOBAL'
                              /'LOCAL'
                  )
```

 # Post-traitements NORM_MODE : pour tous les cas

```
       ◊ NORM_MODE = _F   (
                    ◊ /   NORME =    /   'EUCL_TRAN'
                                     /   'MASS_GENE'
                                     /   'RIGI_GENE'
                                     /   'TRAN'
                                     /   'TRAN_ROTA'        [DEFAUT]
                                     /   'EUCL'
                    ◊   INFO =       /   1                  [DEFAUT]
                                     /   2
                        )
```

 # Autres post-traitements : uniquement pour le cas de modes dynamiques et physiques (matrices d'entrées de type matr_asse_DEPL_R) :

```
      ◊   FILTRE_MODE =_F  (
                    ◊   CRIT_EXTR =  /   'MASS_EFFE_UN'     [DEFAUT]
                                     /   'MASS_GENE'
                    ◊   SEUIL =      /   0.001              [DEFAUT]
                                     /   rseuil             [R]
                        )
```


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

```
      ◊  IMPRESSION = _F  (
                  ◊  CUMUL =    /  'OUI'                    [DEFAUT]
                                /  'NON'
                  ◊  CRIT_EXTR =  /  'MASS_EFFE_UN'         [DEFAUT]
                                /  'MASS_GENE'
                  ◊  TOUT_PARA =  /  'OUI'                  [DEFAUT]
                                /  'NON'
                        )
```

 # Solveur linéaire et parallélisme

```
SOLVEUR=_F().
```

◊ SOLVEUR=_F (Pour plus de détails voir le document [U4.50.01]).

Uniquement si TYPE_RESU='DYNAMIQUE' et OPTION='BANDE' et FREQ est une liste d'au moins 3 valeurs :

```
      ◊  NIVEAU_PARALLELISME =  /  'COMPLET'                [DEFAUT]
                                /  'PARTIEL'

      ◊  INFO=/1                                            [DEFAUT]
               /2                                           [I]

      ◊  TITRE=ti
   );
```

 # Résultats du problème modal

Si	TYPE_RESU='MODE_FLAMB'	alors	[*]->mode_flamb
Si	TYPE_RESU='GENERAL'	alors	[*]->mode_flamb
Si	MATR_AMOR	alors	[*]->mode_meca_c
Si	MATR_RIGI ou MATR_A=[matr_asse_DEPL_C]	alors	[*]->mode_meca_c
Si	MATR_RIGI ou MATR_A=[matr_asse_DEPL_R]	alors	[*]->mode_meca
Si	MATR_RIGI ou MATR_A=[matr_asse_PRES_R]	alors	[*]->mode_acou
Si	MATR_RIGI ou MATR_A=[matr_asse_GENE_R]	alors	[*]->mode_gene
Si	MATR_RIGI ou MATR_A=[matr_asse_GENE_C]	alors	[*]->mode_gene
Si	MATR_RIGI ou MATR_A=[matr_asse_TEMP_R]	alors	[*]->mode_meca

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

3 Opérandes

3.1 Principes

Cet opérateur résout le **problème généralisé (GEP)** aux valeurs propres suivant [R5.01.01] :
Trouver $(\lambda, \mathbf{x})$ tels que $\mathbf{A}\,\mathbf{x}=\lambda\,\mathbf{B}\,\mathbf{x}$, $\mathbf{x}\neq 0$, où $\mathbf{A}$ et $\mathbf{B}$ sont des matrices réelles, symétriques ou non. Pour modéliser un amortissement hystérétique dans l'étude des vibrations libres d'une structure, la matrice $\mathbf{A}$ peut être complexe symétrique [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 $\mathbf{K}$, symétrique réelle (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 $\mathbf{M}$ (symétrique réelle). Les valeurs propres obtenues sont les carrés des pulsations associées aux fréquences cherchées. Le système à résoudre peut s'écrire

$$\underbrace{\left(\mathbf{K}+\mathbf{K}_g\right)}_{\mathbf{A}}\mathbf{x}=\lambda\underbrace{\mathbf{M}}_{\mathbf{B}}\mathbf{x}$$

  où $\lambda=(2\pi f)^2$ est le carré de la pulsation ω, f la fréquence propre et $\mathbf{x}$ le vecteur de déplacement propre associé. Les modes propres manipulés (λ, x) sont à valeurs réelles. Ce type de problématique est activé par le mot-clé `TYPE_RESU='DYNAMIQUE'` et génère une structure de données de type `mode_meca`, `mode_acou` ou `mode_gene` (suivant le type des données d'entrée).

- **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 **x** associé à ce niveau de charge critique $\mu=-\lambda$, se ramène à un problème généralisé aux valeurs propres de la forme

$$\left(\mathbf{K}+\mu\,\mathbf{K}_g\right)x=0\Leftrightarrow\underbrace{\mathbf{K}}_{\mathbf{A}}\mathbf{x}=\lambda\underbrace{\mathbf{K}_g}_{\mathbf{B}}\mathbf{x}$$

  avec $\mathbf{K}$ matrice de rigidité matérielle et $\mathbf{K}_g$ matrice de rigidité géométrique. Les modes propres manipulés (λ, x) sont à valeurs réelles. Ce type de problématique est activé par le mot-clé `TYPE_RESU='MODE_FLAMB'` et génère une structure de données de type `mode_flamb`.

Attention:

- *Dans le code, on ne traite que les valeurs propres du problème généralisé, les variables λ. Pour obtenir les véritables charges critiques, les variables μ, il faut les multiplier par –1.*
- *En GEP, pour traiter des problèmes à modes complexes (matrices non symétriques et/ou à valeurs complexes), il faut utiliser la méthode des itérations simultanées et la méthode de résolution `METHODE='SORENSEN'` ou `'QZ'`.*

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

- Le problème consiste à trouver $(\lambda, \mathbf{x})\in(C, C^N)$ tels que

$$\left(\lambda^2\mathbf{B}+\lambda\,\mathbf{C}+\mathbf{A}\right)\mathbf{x}=\mathbf{0}$$

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

où typiquement, en mécanique linéaire, $\mathbf{A}$ sera la matrice de rigidité, $\mathbf{B}$ la matrice de masse et $\mathbf{C}$ la matrice d'amortissement. Les matrices $\mathbf{A}$, $\mathbf{B}$ et $\mathbf{C}$ sont des matrices symétriques et réelles. La valeur propre complexe λ est reliée à la fréquence propre f et à l'amortissement réduit ξ par $\lambda=\xi(2\pi f)\pm i(2\pi f)\sqrt{1-\xi^2}$. Ce type de problématique est activé par le mot-clé `TYPE_RESU='DYNAMIQUE'` et génère une structure de données de type `mode_meca_c`.

Attention:

- *En QEP, pour traiter des problèmes à matrices non symétriques et/ou à valeurs complexes, il faut utiliser la méthode des itérations simultanées et la méthode de résolution `METHODE='SORENSEN'` ou `'QZ'`.*
- *Le flambement (`TYPE_RESU='MODE_FLAMB'`) n'est pas licite en QEP.*
- *Le test de Sturm n'est opérant qu'en GEP à matrices symétriques réelles. En dehors de ce cadre (QEP, GEP à matrices réelles non symétriques ou à matrice A complexe symétrique), l'option 'BANDE' est proscrite et la post-vérification basée sur Sturm n'est pas activée (paramètre `VERI_MODE/STURM` inopérant).*

Pour résoudre ces problèmes modaux généralisés ou quadratiques, code_aster propose différentes approches. Au delà de leurs spécificités numériques et fonctionnelles qui sont reprises dans les documents [R5.01.01] [R5.01.02], on peut les synthétiser sous la forme du tableau ci-dessous (**les valeurs par défaut sont matérialisées en gras**).

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

Opérateur/ Périmètre d'application	Algorithme	Mot-clé	Avantages	Inconvénients
Méthode des puissances inverses ***OPTION=*** **'SEPARE', 'AJUSTE' *ou* 'PROCHE'**		OPTION=		*Uniquement symétrique réel (GEP et QEP).*
1 ère phase (heuristique)				
Calcul de quelques modes	Bissection (sans objet en QEP).	'SEPARE'		
Calcul de quelques modes	Bissection+ Sécante (GEP) ou Müller-Traub (QEP).	'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 des modes
2 ème phase (méthode des puissances inverses)				*Uniquement symétrique réel (GEP et QEP)*
Méthode de base	Puissances inverses	**'DIRECT'**	Très bonne construction de vecteurs propres	Peu robuste
Option d'accélération	Quotient de Rayleigh (sans objet en QEP)	'RAYLEIGH'	Améliore la convergence	Coût calcul
Méthode des itérations simultanées ***OPTION=*** **'PLUS_*', 'CENTRE', *'BANDE'* *ou* 'TOUT'**		SOLVEUR_MODAL=_F(METHODE=		
Calcul d'une partie du spectre	Bathe & Wilson	'JACOBI'		Peu robuste *Uniquement symétrique réel (GEP)*
	Lanczos (Newman-Pipano en GEP et Jennings en QEP)	'TRI_DIAG'	Détection spécifique des modes rigides.	*Uniquement symétrique réel (GEP et QEP)*
	IRAM (Sorensen)	**'SORENSEN'**	Robustesse accrue. Meilleures complexités calcul et mémoire. Contrôle de la qualité des modes.	Méthode par défaut. *Portée en non symétrique et avec A complexe symétrique.*
Calcul de tout le spectre puis filtrage d'une partie.	QZ	'QZ'	Méthode de référence en terme de robustesse.	Très coûteuse en CPU et en mémoire.

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

Opérateur/ Périmètre d'application	Algorithme	Mot-clé	Avantages	Inconvénients
				A réserver au petits cas (<10^3 degrés de liberté). *Portée en non symétrique et avec A complexe symétrique.*

Tableau 3.1-1 - Récapitulatif des méthodes modales de Code_Aster

Pour capturer une partie significative du spectre, il est préférable d'utiliser les valeurs **'PLUS_PETITE' , 'PLUS_GRANDE' , 'CENTRE'** ou **'BANDE'** du mot-clé **OPTION ,** qui utilisent une méthode de type **« itérations simultanées »** : méthodes de sous-espace (Lanczos, IRAM, Jacobi) ou la méthode globale QZ (méthode très robuste mais coûteuse ; à réserver aux petits problèmes).
Par contre, **lorsqu'il s'agit de déterminer quelques valeurs propres simples bien discriminées ou d'affiner quelques estimations,** les valeurs **'SEPARE', 'AJUSTE' ou 'PROCHE'** du mot-clé OPTION (qui utilisent une méthode de type « **puissances inverses »**) sont souvent bien indiquées.
Il est d'ailleurs tout à fait recommandé de profiter des points forts des deux classes de méthode en affinant les vecteurs propres obtenus préalablement par une méthode d'itérations simultanées, *via* la méthode des puissances inverses**.** Cela permettra de réduire la norme du résidu final. C'est ce que permet de faire le post-traitement 'AMELIORATION'.

Ainsi, même si les options des méthodes d'itérations simultanées (**'PLUS_PETITE/GRANDE', 'CENTRE'** et **'BANDE'**) sont souvent préférables, on pour calculer jusqu'à une dizaine de modes avec les options des méthodes de type puissance :'SEPARE', 'AJUSTE' ou 'PROCHE'. Pour calculer des dizaines voire des centaines de modes, il faut privilégier, lorsque c'est possible, l'option 'BANDE'. Cela améliore la robustesse, la qualité et les performances du calcul.
Dans le cas standard d'un GEP symétrique réel, idéalement il faudrait organiser son calcul en plusieurs sous-bandes comportant chacune entre 20 et 60 modes. Avec, si possible un découpage homogène en nombre de modes (un déséquilibrage inférieur à X3 [1]).

Pour ce faire, on peut procéder en plusieurs étapes:

- **Calibrer** les zones d'intérêt par un appel initial à INFO_MODE sur une liste de fréquences[2] (resp. charges critiques) donnée,
- **Regarder** les nombres de modes propres affichés dans le fichier message (ou dans la sd_table générée),
- **Relancer** un ou plusieurs calculs CALC_MODES avec OPTION='BANDE' en essayant d' **équilibrer** les bandes.

Si on ne calcule finalement qu'une seule bande, pour gagner du temps, on peut même mutualiser une partie du coût calcul de l'INFO_MODE initial en notifiant à CALC_MODES le nom de la sd_table générée (cf. mots-clés TABLE_*). Ce chaînage peut ainsi rendre le surcoût d'INFO_MODE négligeable et guider efficacement le calcul modal.
Par contre, dès qu'on traite un QEP ou un GEP atypique (matrice complexe et/ou non symétrique), le spectre devient complexe. Le chaînage INFO_MODE + CALC_MODES n'est alors plus possible. Certains mots-clés ou valeurs deviennent sans objet (OPTION='BANDE', VERI_MODE/STURM...).

Remarque :

1 Si le sous-bande la moins remplie contient seulement 20 modes, la plus fournie ne devrait pas contenir idéalement plus de 60 modes. Cela permet d'optimiser les coûts du calcul, la robustesse ainsi que la qualité des modes obtenus.
2 Pour des raisons de coûts calcul, il vaut mieux limiter la liste à une douzaine de valeurs. Quitte à relancer plusieurs CALC_MODES successifs pour balayer toutes les sous-bandes recherchées.

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

- *On conseille fortement une lecture préalable des documentations de référence [R5.01.01] [R5.01.02] et [R5.01.04]. Elle donne à l'utilisateur les propriétés et les limitations, théoriques et pratiques, des méthodes modales abordées tout en reliant ces considérations à un paramétrage précis des options.*

3.2 Mot-clé TYPE_RESU

◊ TYPE_RESU=/**'DYNAMIQUE'** **[DEFAUT]**
/'MODE_FLAMB'
/'GENERAL'

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 et effets gyroscopiques) ou recherche de charges critiques (cas de la théorie du flambement linéaire, uniquement en GEP). Suivant cette classe d'appartenance, les résultats sont affichés et stockés différemment dans la structure de données :

- **En dynamique** (TYPE_RESU='DYNAMIQUE', les fréquences sont ordonnées par ordre croissant du module de leur écart au shift (cf. [R5.01.01/02], §3.8/2.5). C'est la valeur de la variable d'accès NUME_ORDRE de la structure de donnée. L'autre variable d'accès, NUME_MODE, est égale à la véritable position modale dans le spectre de la valeur propre (déterminée par le test de Sturm cf. [R5.01.04]). Ce test de Sturm n'est licite qu'en GEP à modes réels (matrices symétriques réelles), dans les autres cas de figures (GEP à modes complexes et QEP), on pose NUME_MODE=NUME_ORDRE.
- **En flambement** (TYPE_RESU='MODE_FLAMB'), les valeurs propres sont stockées par ordre croissant algébrique. Les variables NUME_ORDRE et NUME_MODE prennent la même valeur égale à cet ordre.
- Cas général (TYPE_RESU='GENERAL') : idem que pour le cas du flambement.

Remarque : Le TYPE_RESU='GENERAL' permet de résoudre un problème de valeurs propres dans le cas d'un **système matriciel général**. Pour l'instant son périmètre est limité aux GEPs standards (matrices réelles symétriques). Sa seule différence avec MODE_FLAMB n'est donc que dans la dénomination des matrices : MATR_A / MATR_B plutôt que MATR_RIGI / MATR_RIGI_GEOM.

3.3 Opérandes MATR_RIGI/MATR_A/MATR_MASS/MATR_RIGI_GEOM /MATR_B/MATR_AMOR

Le tableau ci-dessous représente les opérandes à utiliser en fonction type du mot-clé TYPE_RESU.

TYPE_RESU		
'DYNAMIQUE'	'MODE_FLAMB'	'GENERAL'
♦ MATR_RIGI = A	♦ MATR_RIGI = A	♦ MATR_A = A
♦ MATR_MASS = B	♦ MATR_RIGI_GEOM = B	♦ MATR_B = B
◊ MATR_AMOR = C	Sans objet	Hors périmètre actuel

Tableau 3.3-1 - Nom des matrices d'entrée en fonction du TYPE_RESU

♦ MATR_RIGI / MATR_A = A
Matrice assemblée, réelle (symétrique ou non) ou complexe symétrique, de type [matr_asse_*_R/C] du GEP/QEP à résoudre.

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

♦ MATR_MASS / MATR_RIGI_GEOM / MATR_B = B
Matrice assemblée, réelle (symétrique ou non), de type [matr_asse_*_R] du GEP/QEP à résoudre.

◊ MATR_AMOR = C
Matrice assemblée, réelle (symétrique ou non), de type [matr_asse_*_R] du QEP à résoudre.

Remarque :

Si la matrice A est complexe symétrique ou si une des matrices A, B ou C est non symétrique réelle, seuls certains jeux de paramètres sont licites. En particulier :
- les options 'BANDE', 'PROCHE', 'SEPARE', 'AJUSTE' ne sont pas utilisables
- si A est complexe : 'PLUS_PETITE' n'est pas utilisable, ni 'CENTRE' si la fréquence cible est 0
- les méthodes de résolution 'JACOBI' et 'TRI_DIAG' (dans SOLVEUR_MODAL / METHODE) ne sont pas utilisables.

3.4 Opérande OPTION

◊ OPTION=

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

'BANDE' On recherche toutes les valeurs propres dans une bande donnée. Cette bande est définie par l'argument de FREQ= (f_{min}, f_{max}) ou par celui de CHAR_CRIT= $(\lambda_{min}, \lambda_{max})$. Option uniquement disponible en GEP à matrices réelles symétriques.
Dans le cas TYPE_RESU='DYNAMIQUE', le mot-clé FREQ peut être une liste de n>2 valeurs : FREQ= $(f_1 = f_{min}, , ..., f_i, ..., f_n = f_{max})$. Dans ce cas, la bande de recherche globale est découpée en sous-bandes $[f_i, f_{i+1}]$ presque indépendantes (cf §3.6.1).

'CENTRE' On recherche les NMAX_FREQ valeurs propres les plus proches de la fréquence f (argument du mot-clé FREQ= f) ou les NMAX_CHAR_CRIT charges critiques les plus proches de la charge λ (argument du mot-clé CHAR_CRIT= λ).

'PLUS_PETITE' [DEFAUT] On recherche les NMAX_FREQ plus petites fréquences propres (cas TYPE_RESU='DYNAMIQUE') ou les NMAX_CHAR_CRIT plus petites charges critiques (TYPE_RESU='MODE_FLAMB' ou 'GENERAL').

'PLUS_GRANDE' On recherche les NMAX_FREQ plus grandes valeurs propres.
Option utilisable seulement dans le cas TYPE_RESU='DYNAMIQUE', pour un problème généralisé, avec matrices réelles symétriques.
Les rôles des matrices de rigidité et de masse sont intervertis de manière transparente pour l'utilisateur.
Rq : il peut être utile de débrancher le test de STURM dans l'opérande VERI_MODE. En effet, au cœur de l'algorithme, avant d'être converties en fréquences propres physiques, les valeurs propres peuvent être très petites et très proches.

'TOUT' On cherche tous les modes associés à des degrés de liberté physiques.
Option utilisable seulement avec la méthode de résolution QZ (cf §3.5.1.1).

'PROCHE' On recherche les modes dont les valeurs propres sont les plus proches de valeurs données. Ces valeurs sont indiquées par :

- l'argument lfreq du mot-clé FREQ pour un GEP de type dynamique (TYPE_RESU='DYNAMIQUE') .
- l'argument lcharc du mot-clé CHAR_CRIT pour un GEP de type flambement linéaire (TYPE_RESU='MODE_FLAMB') .
- les arguments lfreq et lamor des mot-clés FREQ et AMOR_REDUIT pour un QEP de type dynamique (TYPE_RESU='DYNAMIQUE') .

Il y a autant de recherches de modes que de termes dans cette liste (ou ces listes).
Rq : Si on souhaite calculer un mode multiple, il ne faut pas utiliser cette option car on ne trouvera qu'un seul mode pour chaque valeur donnée de la liste.

'SEPARE' On sépare les valeurs propres par une méthode de bissection basée sur le critère de Sturm. Les bornes de l'intervalle de recherche sont :

- les valeurs de la liste lfreq du mot-clé FREQ pour un problème généralisé ou quadratique de type dynamique (TYPE_RESU='DYNAMIQUE') .
- les valeurs de la liste lcharc du mot-clé CHAR_CRIT pour un problème généralisé de type flambement linéaire (TYPE_RESU= 'MODE_FLAMB') .

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

'AJUSTE'	Fonctionnement similaire à l'option 'SEPARE' précédente. Après avoir séparé les fréquences propres via l'option 'SEPARE' (en GEP uniquement), on effectue des itérations supplémentaires soit par la méthode de la sécante (GEP) soit par la méthode de Müller-Traub (QEP) pour obtenir une meilleure précision sur les valeurs propres.

Tableau 3.4-1 - Valeurs possibles du mot-clé OPTION

Il est important de se rappeler que le choix d'une de ces options entraîne l'utilisation d'une méthode particulière :
- OPTION= 'BANDE' , 'CENTRE' , 'PLUS_PETITE' , 'PLUS_GRANDE' ou 'TOUT' implique l'utilisation d'une méthode des itérations simultanées
- OPTION='PROCHE', 'SEPARE', 'AJUSTE' implique l'utilisation de la méthode des puissances inverses.

Ce choix a des conséquences sur le reste des mot-clés accessibles dans la commande, notamment sur le paramétrage du SOLVEUR_MODAL.

Voir [R5.01.01/02] §2.5/3.8.

3.5 Mot-clé facteur SOLVEUR_MODAL

Mot-clé permettant de régler les algorithmes et les paramètres du solveur modal.

3.5.1 Mots-clés associés à la méthode des itérations simultanées

Ces mots-clés ne sont utilisables qui si la valeur du mot-clé OPTION est parmi 'PLUS_PETITE', 'PLUS_GRANDE', 'BANDE', 'CENTRE', 'TOUT'.

3.5.1.1 Mot-clé METHODE

Quatre méthodes de résolution sont alors disponibles pour résoudre le problème aux valeurs propres (cf. aussi Tableau 3.1-1) :

◊ METHODE=/**'SORENSEN' [DEFAUT]**

On utilise la méthode de Sorensen (package externe ARPACK) pour calculer les modes propres du GEP ou du QEP (cf. [R5.01.01/02] §7/4). Son périmètre englobe les matrices réelles, symétriques ou non, voire une matrice A complexe symétrique.

/'TRI_DIAG'

On utilise la méthode de Lanczos (variante de Newmann-pipano en GEP, de Parlett & Saad en QEP) pour calculer les modes propres du GEP ou du QEP (cf. [R5.01.01/02] §6/4). Son périmètre est limité aux matrices symétriques réelles.

/'JACOBI'

On utilise la méthode de Bathe & Wilson (puis la méthode de Jacobi sur le système projeté) pour calculer les modes propres du GEP (cf. [R5.01.01] §8). Son périmètre est limité aux matrices symétriques réelles.

/'QZ'

On utilise la méthode QZ de la bibliothèque externe LAPACK pour calculer les modes propres du GEP ou du QEP (cf. [R5.01.01/02] §9/5). Son périmètre englobe les matrices réelles, symétriques ou non,

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

voire une matrice A complexe symétrique. Cette méthode de référence très coûteuse est à réserver aux problèmes de petites tailles (<10^3 degrés de liberté).

3.5.1.2 Paramètres liés à la méthode de résolution

Chacune de ces méthodes de résolution dispose de différents réglages :

- si METHODE='SORENSEN' :

◊ PREC_SOREN=pso **(0.)** **[DEFAUT]**

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 .10 $^{-16}$.*

◊ NMAX_ITER_SOREN=nso **(20)** **[DEFAUT]**
◊ PARA_ORTHO_SOREN=porso **(0.717)** **[DEFAUT]**

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.* [R5.01.01] §7) et du coefficient d'orthogonalisation de l'IGSM de Kahan-Parlett (cf. [R5.01.01] annexe 2).
Si le coefficient porso est négatif, la réorthogonalisation est effectuée sur tous les modes calculés au lieu de cibler uniquement les modes appartenant au même espace propre. Le calcul peut alors être deux ou trois fois plus coûteux.
Pour plus d'informations sur le mode de sélection des espaces propres on pourra consulter les paramètres SEUIL_FREQ/CHAR_CRIT.

Remarque :

- *Lors des premiers passages, il est fortement conseillé de ne pas modifier ces paramètres avancés de l'algorithme et qui sont initialisés empiriquement à des valeurs standards.*

- si METHODE='TRI_DIAG' :

◊ PREC_ORTHO =po **(1.10 $^{-12}$)** **[DEFAUT]**
◊ NMAX_ITER_ORTHO=nio **(5)** **[DEFAUT]**
◊ PREC_LANCZOS=pl **(1.10^{-8})** **[DEFAUT]**
◊ NMAX_ITER_QR=nim **(30)** **[DEFAUT]**
◊ MODE_RIGIDE=/'OUI'
/**'NON'** **[DEFAUT]**

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. [R5.01.01] §6).

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.

Le quatrième fixe le nombre d'itérations maximum pour la résolution du système réduit pour la méthode QR ([R5.01.01] annexe 1).

Le mot-clé MODE_RIGIDE permet de détecter et de calculer au préalable, par une méthode algébrique, les modes de corps de rigide. 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

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

valeurs propres calculées par l'algorithme de Lanczos ne sont pas nulles mais très voisines de zéro.

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.*

- si METHODE='JACOBI' :

◊	PREC_BATHE =pbat	**(1.10^{-10})**	**[DEFAUT]**
◊	NMAX_ITER_BATHE=nbat	**(40)**	**[DEFAUT]**
◊	PREC_JACOBI=pjaco	**(1.10-2)**	**[DEFAUT]**
◊	NMAX_ITER_JACOBI=njaco	**(12)**	**[DEFAUT]**

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. [R5.01.01] §8).
Les deux autres ajustent la précision de la convergence et le nombre maximum d'itérations de la méthode de JACOBI (cf. [R5.01.01] annexe 3). Ce solveur modal global est utilisé pour calculer les modes propres de la matrice projetée par Bathe & Wilson.

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.*

- si METHODE='QZ' :

◊ TYPE_QZ = /**'QZ_SIMPLE' [DEFAUT]**
/'QZ_EQUI'
/'QZ_QR'

Ce paramètre permet de choisir une des variantes de l'algorithme QZ proposé par LAPACK. Le premier choix ('QZ_SIMPLE') désigne la méthode de base, le second ('QZ_EQUI') lui rajoute un pré-traitement d'équilibrage des termes de la matrice. Cela améliore souvent la qualité des modes mais, *a contrario*, si la matrice présente des termes très petits dus à des erreurs d'arrondis, cette phase engendre alors des modes parasites.

Quant au troisième choix ('QZ_QR'), il est réservé au cas symétrique défini positif (matrice de raideur réelle, condition de Dirichlet sans Lagrange, pas de flambement ou d'amortissement). Il est beaucoup plus rapide que les options précédentes.

3.5.1.3 Mot-clé APPROCHE

◊ APPROCHE= /**'REEL' [DEFAUT]**
/'IMAG'
/'COMPLEXE' (uniquement avec Sorensen)

Ce mot-clé définit le type d'approche (réelle, imaginaire ou complexe) pour le choix du pseudo-produit scalaire du QEP utilisé avec les méthode de Lanczos ou avec celle de Sorensen (cf. [R5.01.02]).
Cet opérande n'a de sens que pour l'analyse des vibrations (TYPE_RESU='DYNAMIQUE') libres d'une structure amortie ou tournante (modes propres complexes ; le mot-clé MATR_AMOR doit être renseigné). En flambement, (TYPE_RESU='MODE_FLAMB') cela n'a aucun intérêt.

Remarques :
- *En quadratique, avec la méthode de Lanczos seule l'approche 'IMAG' est compatible avec une borne fréquentielle nulle ('OPTION=PLUS_PETITE' ou 'CENTRE' avec $f=0$).*
- *Avec Sorensen, aucune n'est compatible.*

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

3.5.1.4 Mots-clé DIM_SOUS_ESPACE et COEF_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 à l'aide COEF_DIM_ESPACE (cf. §5 de ce document et [R5.01.01] §5.3).

Grâce à la donnée 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.
Si on recherche des modes propres sur une bande découpée en plusieurs sous-bandes (OPTION='BANDE' et FREQ est une liste d'au moins 3 valeurs), on peut donc optimiser la taille des sous-espaces qui reste proportionnelle au nombre de fréquences recherchées sur chaque sous-bande : 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 «degrés de liberté actifs», nactif (cf. [R5.01.01] §3.2), 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 .*
- *Ces paramètres sont inutiles pour la méthode 'QZ'.*

3.5.2 Paramètres associés à la méthode des puissances inverses

Ces paramètres sont donc disponibles uniquement si la valeur du mot-clé OPTION est parmi 'PROCHE', 'SEPARE', 'AJUSTE'.

3.5.2.1 Opérandes de la bissection (si OPTION='SEPARE' ou 'AJUSTE')

◊ NMAX_ITER_SEPARE=nis **(30)** **[DEFAUT]**
◊ PREC_SEPARE =ps **(1.10^{-4})** **[DEFAUT]**

Paramètres d'ajustement du nombre d'itérations et de la précision de séparation pour la recherche par dichotomie. Ces opérandes sont ignorés pour l'option 'PROCHE' (Cf. [R5.01.01] §4.2).

3.5.2.2 Opérandes de la sécante (si OPTION='AJUSTE')

◊ NMAX_ITER_AJUSTE = nia **(15)** **[DEFAUT]**
◊ PREC_AJUSTE = pa **(1.10^{-4})** **[DEFAUT]**

Paramètres d'ajustement du nombre d'itérations et de la précision de séparation pour la méthode de la sécante. Ces opérandes ne servent qu'à l'option 'AJUSTE' (Cf. [R5.01.01] §4.2).

Remarque :

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

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.

3.5.2.3 Paramètres de calcul de la deuxième phase de calcul de la méthode des puissances inverses

◊ OPTION_INV= / **'DIRECT'** **[DEFAUT]**
/ 'RAYLEIGH'

Définition de la méthode des puissances inverses (confer [R5.01.01/02] §4.3/3.3) :

'DIRECT' [DEFAUT]	méthode standard en GEP ou variante de Jennings en QEP.
'RAYLEIGH'	Accélération via le quotient de Rayleigh (uniquement en GEP)

Tableau 3.5.2.3-1 - Fonctionnement de OPTION_INV selon sa valeur

◊ NMAX_ITER_INV=nim **(30)** **[DEFAUT]**

Nombre maximum d'itérations de la méthode des puissances inverses pour la recherche des modes propres.

◊ PREC_INV=pm **(1.10^{-5})** **[DEFAUT]**

Test d'arrêt de la méthode des puissances inverses.

3.6 Mot-clé CALC_FREQ (si TYPE_RESU='DYNAMIQUE')

◊ CALC_FREQ=_F(…

Mot-clé facteur qui précise les paramètres de calcul des modes propres et leur nombre, en fonction de l'OPTION choisie.

3.6.1 Opérande FREQ (uniquement si OPTION='BANDE' ou 'CENTRE' ou 'PROCHE' ou 'SEPARE' ou 'AJUSTE')

♦ FREQ=l_f
◊ TABLE_FREQ=table_f

Liste des fréquences : son utilisation dépend de l'OPTION choisie.

OPTION='BANDE'	On attend une liste de n ≥ 2 valeurs $(f_{min}=f_1,\ldots,f_i,\ldots,f_{max}=f_n)$ qui définissent la bande de recherche. Si n > 2, la bande de recherche globale $[f_{min},f_{max}]$ est découpée en sous-bandes $[f_i,f_{i+1}]$.
OPTION='CENTRE'	On attend une seule valeur de fréquence.
OPTION='PROCHE'	On attend une liste de n ≥ 1 valeurs $(f_1,\ldots,f_i,\ldots f_n)$ qui définissent les fréquences autour desquelles on cherche la fréquence propre la plus proche.
OPTION='SEPARE' ou 'AJUSTE'	On attend une liste de n ≥ 2 valeurs $(f_1,\ldots,f_i,\ldots f_n)$ qui définissent les bornes des intervalles de recherche $[f_i,f_{i+1}]$.

Tableau 3.6.1-1 - Utilisation du mot-clé FREQ en fonction de l'OPTION choisie

- Avec l'option ' BANDE ': les valeurs stipulées sous ce mot-clé doivent être positives strictement croissantes.

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

- Si n = 2 :

on commence par opérer un coûteux[3] test de Sturm afin de déterminer le nombre de modes contenus dans la bande (cf. [R5.01.04]). Si on a, au préalable, effectué une calibration de la zone d'intérêt par un INFO_MODE, on peut économiser une partie de ce coût du calcul. Pour ce faire, on réutilise la table générée par l'INFO_MODE. Les bornes (f_{min}, f_{max}) définies ci-dessus permettent de sélectionner une ou plusieurs lignes de ladite table.
Par exemple, si la table a été générée par un

$$\text{INFO_MODE+FREQ=}\left(f_{im0}, f_{im1}, f_{im2}, f_{im3}, f_{im4}, f_{im5}\right),$$

on peut en mutualiser une partie du coût calcul en posant dans CALC_MODES+'BANDE', FREQ= $(f_{min}=f_{im1}, f_{max}=f_{im4})$. L'étape de prétraitement de CALC_MODES ne va alors pas effectuer le test de Sturm mais, à la place, détecter dans la table les sous-bandes incluses dans l'intervalle. Soit, ici :

$$[f_{im1}, f_{im2}]\cup[f_{im2}, f_{im3}]\cup[f_{im3}, f_{im4}].$$

On va alors juste sommer les nombres de modes correspondant à chaque sous-intervalle pour en déduire le nombre de modes total à rechercher.
Ce chaînage permet vraiment de réduire le surcoût de la calibration initiale par INFO_MODE et est donc à utiliser chaque fois que c'est possible.

Remarques à propos du *chaînage* INFO_MODE+CALC_MODES+'BANDE':

- *Les bornes de sélection (f_{min}, f_{max}) doivent correspondre exactement à celles ayant servies à générer l'* INFO_MODE *initial (à* VERI_MODE/PREC_SHIFT *% près [4]).*
- *La sélection des lignes de la table s'effectue par rapport aux valeurs initiales des fréquences. Mais si elles ont subi des décalages (car elles étaient trop proches de modes propres), la table trace aussi ces valeurs après décalages (valeurs* FREQ_MIN/MAX *versus* BORNE_MIN/MAX_EFFECT *). Ce sont bien sûr ces dernières valeurs décalées qui sont transmises à l'algorithmie de* CALC_MODES*. Cette stratégie préserve ainsi, à la fois, l'ergonomie de l'option et la consistance des comportements logiciels : un* CALC_MODES+'BANDE' *fournit le même résultat, qu'il démarre sa phase de prétraitement avec ou sans table précalculée.*
- *Si une des bornes de sélection a dû être décalée (dans l'* INFO_MODE *préalable), la phase de pré-traitement émet une* ALARME *pour reproduire le même comportement que pour un calcul standard.*
- *La table ne doit comporter ni trou, ni recouvrement, sinon elle est rejetée. Mais ce cas de figure ne peut normalement pas se produire avec une carte issue de* INFO_MODE *. Cette règle permet de préserver la robustesse du schéma algorithmique : on ne veut rater aucune fréquence.*

- Si n > 2 :

la calibration de chaque sous-bande par l'opérateur INFO_MODE est faite automatiquement au sein de l'opérateur CALC_MODES. En outre, les recherches sur chacune des sous-bandes peuvent être parallélisées afin de réduire les temps de calcul (cf §3.12).

Remarques :

- *Chaque fréquence n'est traitée qu'une seule fois : en tant que borne inférieure de la première sous-bande pour la première de la liste, en tant que borne supérieure des sous-bandes qui suivent pour les autres fréquences. En particulier, si cette fréquence est jugée trop proche d'une valeur propre, on la décale (cf. [U4.52.01] et [R5.01.04]).*
- *Le décalage éventuel d'une borne de fréquence ne s'opère plus qu'une seule fois dans l'*INFO_MODE *initial. Il n'y a donc plus de risque de chevauchement d'intervalles décalés comme jusqu'en version v10. On ne risque donc plus de calculer par erreur deux fois le même mode.*

- Avec l'option 'PROCHE' : les valeurs stipulées sous ce mot-clé doivent être positives strictement croissantes. c'est la liste des fréquences dont on cherche le mode le plus proche.

3 En moyenne 20/30% du coût calcul total de l'opérateur.
4 Le critère de tri est en relatif, sauf lorsque la borne recherchée est proche de zéro. Il devient alors un critère absolu.

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

- Avec l'option ' SEPARE ' ou ' AJUSTE' : les valeurs sont les bornes des intervalles de recherche. On cherchera à séparer les fréquences dans les intervalles

$$[f_1, f_2], [f_2, f_3] \ldots [f_{n-2}, f_{n-1}], [f_{n-1}, f_n]$$

La liste a au moins deux éléments. Les fréquences doivent être positives et dans l'ordre croissant.

3.6.2 Opérande AMOR_REDUIT (uniquement si OPTION='CENTRE' ou 'PROCHE')

◊ AMOR_REDUIT=l_a

Valeur de l'amortissement réduit qui permet de définir la valeur propre complexe (le «shift») autour de laquelle on cherche les valeurs propres les plus proches (cf. [R5.01.01] §5.4). Cette option ne peut être utilisée que dans le cadre d'un problème modal à modes complexes : QEP ou GEP à matrices réelles non symétriques ou avec A complexe symétrique.

OPTION='CENTRE'	On attend une seule valeur d'amortissement réduit
OPTION='PROCHE'	On attend une liste de valeurs d'amortissement réduit, de même taille que la liste donnée sous le mot-clé FREQ.

Tableau 3.6.2-1 - fonctionnement du mot-clé AMOR_REDUIT selon l'OPTION choisie

La valeur stipulée sous ce mot-clé doit être positive et être comprise entre 0 et 1.

3.6.3 Opérande NMAX_FREQ (uniquement si OPTION='PLUS_PETITE' ou 'PLUS_GRANDE' ou 'CENTRE' ou 'SEPARE' ou 'AJUSTE')

◊ NMAX_FREQ=nf **(10** si OPTION='PLUS_PETITE'**, 1** si OPTION='PLUS_GRANDE', **0** si OPTION='SEPARE' ou 'AJUSTE'**)** **[DEFAUT]**

Nombre maximum de valeurs propres à calculer.

- Si OPTION='PLUS_PETITE' ou 'PLUS_GRANDE' ou 'CENTRE' :

Si nf est strictement supérieur au nombre de «degrés de liberté-actifs», nactif (cf. [R5.01.01] §3.2), alors on le force à prendre cette valeur plafond.

- Si OPTION='PROCHE' :

La valeur est ignorée.

- Si OPTION='SEPARE' ou 'AJUSTE' :

Si l'utilisateur ne renseigne pas ce mot-clé, toutes les valeurs propres contenues dans les intervalles précisés par l'utilisateur sont calculées. Sinon, les NMAX_FREQ premières valeurs propres, donc les plus basses, sont calculées.

3.7 Mot-clé CALC_CHAR_CRIT (si TYPE_RESU='MODE_FLAMB' ou 'GENERAL')

◊ CALC_CHAR_CRIT=_F(…

Mot-clé facteur qui précise les paramètres de calcul des modes propres et leur nombre, en fonction de l'OPTION choisie.

3.7.1 Opérande CHAR_CRIT (uniquement si OPTION='BANDE' ou 'CENTRE' ou 'PROCHE' ou 'SEPARE' ou 'AJUSTE')

♦ CHAR_CRIT=l_c

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

◊ TABLE_CHAR_CRIT=table_c

Liste des charges critiques : son utilisation dépend de l'option choisie.

OPTION='BANDE'	On attend deux valeurs $(\lambda_{min}, \lambda_{max})$ qui définissent la bande de recherche
OPTION='CENTRE'	On attend une seule valeur de charge critique
OPTION='PROCHE'	On attend une liste de n ≥ 1 valeurs qui définissent les $(\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, \lambda_n)$ charges autour desquelles on cherche la charge critique la plus proche.
OPTION='SEPARE' ou 'AJUSTE'	On attend une liste de n ≥ 2 valeurs qui définissent les $(\lambda_1, \lambda_2, \ldots, \lambda_{n-1}, \lambda_n)$ bornes des intervalles de recherche $[\lambda_i, \lambda_{i+1}]$.

Tableau 3.7.1-1 - Utilisation du mot-clé CHAR_CRIT en fonction de l'OPTION choisie

Les valeurs stipulées sous ce mot-clé sont positives ou négatives. En QEP cela n'a aucun intérêt.

- Avec l'option ' BANDE ', on commence par opérer un coûteux [5] test de Sturm afin de déterminer le nombre de modes contenus dans la bande (cf. [R5.01.04]). Si on a, au préalable, effectué une calibration de la zone d'intérêt par un INFO_MODE , on peut économiser une partie de ce coût du calcul. Pour ce faire, on réutilise la table générée par l'INFO_MODE. Les bornes $(\lambda_{min}, \lambda_{max})$ définies ci-dessus permettent de sélectionner une ou plusieurs lignes de ladite table.

Par exemple, si la table a été générée par un
INFO_MODE+CHAR_CRIT= $(\lambda_{im0}, \lambda_{im1}, \lambda_{im2}, \lambda_{im3}, \lambda_{im4}, \lambda_{im5})$,
on peut en mutualiser une partie du coût calcul en posant dans CALC_MODES+'BANDE', CHAR_CRIT= $(\lambda_{min}=\lambda_{im1}, \lambda_{max}=\lambda_{im4})$. L'étape de pré-traitement de CALC_MODES ne va alors pas effectuer le test de Sturm mais, à la place, détecter dans la table les sous-bandes incluses dans l'intervalle. Soit, ici :
$$[\lambda_{im1}, \lambda_{im2}] \cup [\lambda_{im2}, \lambda_{im3}] \cup [\lambda_{im3}, \lambda_{im4}] .$$
On va alors juste sommer les nombres de modes correspondant à chaque sous-intervalles pour en déduire le nombre de modes totale à rechercher.
Ce chaînage permet vraiment de réduire le surcoût de la calibration initiale par INFO_MODE et est donc à utiliser chaque fois que c'est possible.

Remarques à propos du *chaînage INFO_MODE+CALC_MODES+'BANDE'*:

- *Les bornes de sélection $(\lambda_{min}, \lambda_{max})$ doivent correspondre exactement à celles ayant servies à générer l'INFO_MODE initial (à VERI_MODE/PREC_SHIFT % près [6]).*
- *La sélection des lignes de la table s'effectue par rapport aux valeurs initiales des fréquences. Mais si elles ont subi des décalages (car elles étaient trop proches de modes propres), la table trace aussi ces valeurs après décalages (valeurs CHAR_CRIT_MIN/MAX versus BORNE_MIN/MAX_EFFECT). Ce sont bien sûr ces dernières valeurs décalées qui sont transmises à l'algorithmie de CALC_MODES. Cette stratégie préserve ainsi, à la fois l'ergonomie de l'option et la consistance des comportements logiciels : un CALC_MODES+'BANDE' fournit le même résultat, qu'il démarre sa phase de pré-traitement avec ou sans table précalculée.*
- *Si une des bornes de sélection a dû être décalée (dans l'INFO_MODE préalable), la phase de pré-traitement émet une ALARME pour reproduire le même comportement que pour un calcul standard.*

5 En moyenne 20/30% du coût calcul total de l'opérateur.
6 Le critère de tri est en relatif, sauf lorsque la borne recherchée est proche de zéro. Il devient alors un critère absolu.

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

> • *La table ne doit comporter ni trou, ni recouvrement, sinon elle est rejetée. Mais ce cas de figure ne peut normalement pas se produire avec une carte issue de INFO_MODE . Cette règle permet de préserver la robustesse du schéma algorithmique : on ne veut rater aucune fréquence.*

- Avec l'option 'PROCHE': les valeurs stipulées sous ce mot-clé doivent être positives strictement croissantes. c'est la liste des charges dont on cherche le mode le plus proche.

- Avec l'option ' SEPARE ' ou ' AJUSTE' : les valeurs sont les bornes des intervalles de recherche. On cherchera à séparer les charges critiques dans les intervalles
$$[\lambda_1,\lambda_2],[\lambda_2,\lambda_3]...[\lambda_{n-2},\lambda_{n-1}],[\lambda_{n-1},\lambda_n]$$
La liste a au moins deux éléments. Les fréquences doivent être positives et dans l'ordre croissant.

3.7.2 Opérande NMAX_CHAR_CRIT (uniquement si OPTION='PLUS_PETITE' ou 'CENTRE' ou 'SEPARE' ou 'AJUSTE')

◊ NMAX_CHAR_CRIT=nf **(10) [DEFAUT]**

Nombre maximum de charges critiques à calculer.

- Si OPTION='PLUS_PETITE' ou 'CENTRE' :

Si nf est strictement supérieur au nombre de «degrés de liberté actifs», nactif (cf. [R5.01.01] §3.2), alors on le force à prendre cette valeur plafond.

- Si OPTION='PROCHE' :

La valeur est ignorée.

- Si OPTION='SEPARE' ou 'AJUSTE' :

Si l'utilisateur ne renseigne pas ce mot-clé, toutes les valeurs propres contenues dans les intervalles précisés par l'utilisateur sont calculées. Sinon, les NMAX_CHAR_CRIT premières valeurs propres, donc les plus basses, sont calculées.

3.8 Opérandes communes à CALC_FREQ et CALC_CHAR_CRIT : SEUIL_FREQ / SEUIL_CHAR_CRIT, PREC_SHIFT, NMAX_ITER_SHIFT

Ces opérandes sont situées à l'intérieur du mot-clé CALC_FREQ (cas TYPE_RESU='DYNAMIQUE') ou CALC_CHAR_CRIT (cas TYPE_RESU='MODE_FLAMB' ou 'GENERAL').

```
# SI TYPE_RESU='DYNAMIQUE'
      ◊  PREC_SHIFT      =  p_shift        (0.05)    [DEFAUT]
      ◊  SEUIL_FREQ      =  f_seuil        (0.01)    [DEFAUT]
      ◊  NMAX_ITER_SHIFT =  n_shift        (3)       [DEFAUT]

# SI TYPE_RESU='MODE_FLAMB' ou 'GENERAL'
      ◊  PREC_SHIFT      =  p_shift        (0.05)    [DEFAUT]
      ◊  SEUIL_CHAR_CRIT =  c_seuil        (0.01)    [DEFAUT]
      ◊  NMAX_ITER_SHIFT =  n_shift        (3)       [DEFAUT]
```

Le déroulement d'un calcul modal dans cet opérateur requiert la factorisation LDL^T de matrices dynamiques $\boldsymbol{Q}(\lambda)$ du type (cf. [R5.01.01/02] §2.5/3.8)

$$\boldsymbol{Q}(\lambda):=\boldsymbol{A}-\lambda\boldsymbol{B} \text{ (en GEP)}$$
$$\boldsymbol{Q}(\lambda):=\lambda^2\boldsymbol{B}+\lambda\boldsymbol{C}+\boldsymbol{A} \text{ (en QEP)}$$

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

Ces factorisations sont tributaires d'instabilités numériques lorsque le shift λ est proche d'une valeur propre du problème. Cette détection s'opère en comparant la perte de décimales des termes diagonaux de cette factorisée par rapport à leurs valeurs initiales (en valeur absolue). Si le maximum de cette perte est supérieure à ndeci[7], la matrice est supposée singulière et on cherche une valeur décalée du shift procurant une matrice inversible.

Pour les GEPs, les paramètres SEUIL_* permettent de définir le « zéro modal », c'est-à-dire la valeur en deçà de laquelle on considère qu'une valeur propre est nulle. En corollaire, dans certains traitements de l'opérateur, si l'écart entre deux valeurs propres est inférieur à ce chiffre, on considère qu'elles sont confondues. Il faut donc ajuster cette valeur suivant l'amplitude moyenne des modes recherchés.
Si on est en dynamique on transforme cette valeur en pulsation

$$omecor = (2\pi \, f_seuil)^2$$

tandis qu'en flambement on la garde telle quelle

$$omecor = c_seuil .$$

Pour les QEPs, cette valeur du « zéro modal » est utilisée lors du tri effectué à l'issue du calcul modal. Lors de ce tri, on cherche à déterminer si un mode est réel (on ne le retient pas), complexe conjugué (on garde celui de partie imaginaire positive) ou complexe dépareillé (on ne le retient pas). Deux modes λ_1, λ_2 sont considérés comme conjugués si

$$|\lambda_1 - \overline{\lambda_2}| < omecor$$

Remarques :

- *Un mode est considéré comme réel si sa partie imaginaire est inférieure à SEUILR=1E-7 (valeur en dur initialisée dans les routines de tri).*
- *Lorsque des modes ont été triés comme purement réels ou complexes dépareillés, un message informatif ou une alarme apparaît (ALGELINE4_87/88) suivant les cas de figure.*
- *Avec les méthodes de Sorensen et de QZ, en GEP standard (symétrique réel), les paramètres *_seuil servent à déterminer si deux modes doivent être orthogonalisés ou non (lorsque l'orthogonalisation sélective est activée comme c'est le cas par défaut, cf. PARA_ORTHO_SOREN). Deux modes sont considérés comme « multiples », donc à réorthogonaliser, si leurs modules sont tous les deux inférieurs à 100*omecor ou, dans le cas contraire, si leur écart relatif est inférieur à omecor. Cette réorthogonalisation est coûteuse mais indispensable pour les projections ultérieures sur base modale, d'où le besoin de valeurs équilibrées pour ce critère. Normalement les valeurs fixées par défaut sont suffisantes et elles n'ont pas à être modifiées souvent.*

Les autres paramètres, PREC_SHIFT et NMAX_ITER_SHIFT, sont liés à l'algorithme de décalage des bornes de l'intervalle $[f_{min}, f_{max}]$ (cf. [R5.01.04] §3.2), lorsqu'on s'aperçoit que celles-ci sont très proches d'une valeur propre. Grossièrement ces bornes f_{min} (ou λ_{min} en flambement) ou f_{max} (resp. λ_{max}) sont alors décalées vers l'extérieur du segment de p_shift%. Si la matrice dynamique ainsi reconstruite est toujours jugée “numériquement singulière”, on décale à nouveau après avoir émis une ALARME. On tente ce décalage n_shift fois.

$$\lambda^-_{min} = \lambda_{min} - max(omecor, 2^{(i-1)} \times p_{shift} \times |(\lambda_{min})|) \quad (i\text{ème tentative})$$
$$\lambda^-_{max} = \lambda_{max} + max(omecor, 2^{(i-1)} \times p_{shift} \times |(\lambda_{max})|) \quad (i\text{ème tentative})$$

En fait, en dynamique comme en flambement, le décalage s'opère de la même manière. *Stricto sensu*, en dynamique ce n'est donc pas les fréquences qu'on décale, mais les pulsations.
Autre précision, le décalage est en fait, par soucis d'efficacité, dichotomique : p_shift% la première fois, 2xp_shift% la seconde fois etc. Ce procédé doit permettre de rapidement s'éloigner de la “zone de singularité” à moindre coût. *A contrario*, il ne faut pas trop augmenter les valeurs de ces paramètres, car à force de décalages, les bornes résultantes peuvent s'avérer être très différentes des bornes initiales.

7 Valeur fixée *via* le paramètre NPREC du mot-clé SOLVEUR (par défaut ndeci=8).

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

De plus, pour rester cohérent avec le “zéro modal” (noté ici omecor) :

- on ne décale pas d'une valeur inférieure à ce minimum (d'où le max dans les formules ci-dessus) ;
- si dés le départ, la borne jaugée est inférieure à ce “zéro” $|\lambda_*| < omecor$ (en valeur absolue) on la fixe à plus ou moins cette valeur (suivant que cette borne est positive ou négative). On ne permet alors plus aucun décalage.

Remarques:

- *Une borne de l'intervalle σ est proche d'une valeur propre, lorsque la factorisation LDLT de la matrice dynamique associée à cette borne (par exemple celle d'un GEP s'écrit $\mathbf{Q}(\sigma) := \mathbf{A} - \sigma \mathbf{B}$), conduit à une perte de décimale de plus de NPREC digits (valeur paramétrée sous le mot-clé SOLVEUR). En jouant sur la valeur de ce paramètre (NPREC=7, 8 ou 9), on peut alors éviter les coûteuses refactorisations qu'impliquent ces décalages lorsque cette singularité numérique est peu prononcée.*
- *De même, en jouant sur les paramètres numériques des solveurs linéaires (par exemple: METHODE, RENUM, PRETRAITEMENTS...), on peut aussi influencer ce critère de singularité.*
- *On met en œuvre cette technique de décalage dans deux cas de figure : calcul d'un test de Sturm (pré et/ou post-traitement) et construction de la matrice dynamique de travail. En cas d'échec de l'algorithme de décalage : dans le premier cas, on émet une ALARME, dans le second, on s'arrête en ERREUR_FATALE.*

3.9 Mot-clé facteur SOLVEUR

◊ SOLVEUR=_F(),

On a accès à tous les paramètres des solveurs linéaires directs (METHODE='LDLT'/'MULT_FRONT'/'MUMPS').

En mode parallèle, on conseille particulièrement le paramétrage[8] METHODE='MUMPS' et RENUM='QAMD'.
Pour plus de détails sur les solveurs, on pourra consulter le document [U4.50.01]. Concernant le parallélisme, on renvoie au document [U2.08.06] et au paragraphe dédié du document [U2.06. 01].

3.10 Mot-clé VERI_MODE

◊ VERI_MODE=_F(…
Mot-clé facteur pour la définition des post-traitements de vérification des modes propres. Ces post-traitements concernent la norme du résidu des modes et éventuellement le comptage des valeurs propres (cf [R5.01.01] §3.7.4 et [R5.01.02] §2.5.4).

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.

3.10.1 Opérande STOP_ERREUR

◊ STOP_ERREUR=/**'OUI'** **[DEFAUT]**
/'NON'

8 Afin de réduire au minimum le coût en temps de la phase d'analyse (séquentielle) de MUMPS. Ce paramétrage se fait cependant au détriment de la consommation mémoire. Mais ce surcoût s'avère rapidement compensé par la distribution des données sur les processeurs qu'implique le parallélisme.

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

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` (branché par défaut uniquement avec la méthode des itérations simultanées) n'est pas vérifié.

Par défaut le concept de sortie n'est pas produit.

3.10.2 Opérande SEUIL

◊ `SEUIL=r` **(1.10^{-6}) [DEFAUT] pour la méthode des itérations simultanées**
(1.10^{-2}) [DEFAUT] pour la méthode des puissances inverses

Seuil de tolérance pour la norme d'erreur relative du mode au dessus duquel il est considéré comme faux ou trop approximé (cf. [R5.01.01/02] algorithme n°2/n°1). Voir aussi paramètre `STOP_ERREUR`.

3.10.3 Opérande STURM

- si `TYPE_RESU='DYNAMIQUE'` et `OPTION='BANDE'` :

◊ `STURM=` /**`'GLOBAL' [DEFAUT]`**
/`'LOCAL'`
/`'OUI'`
/`'NON'`

`'GLOBAL'` et `'LOCAL'` ont un sens s'il y a au moins deux sous-bandes de calcul. S'il y a une seule bande de calcul, ces deux valeurs sont équivalentes à `'OUI'`.
Vérification dite de `STURM` permettant de s'assurer que l'algorithme utilisé dans l'opérateur a déterminé le nombre exact de valeurs propres, sous-bande par sous-bande (`'LOCAL'`) ou uniquement dans la bande globale[9] (`'GLOBAL'`) (cf. [U4.52.01][R5.01.04]). La deuxième variante est la plupart du temps amplement suffisante et beaucoup moins coûteuse que la première.
Cependant, lorsque les bornes fournies au test de Sturm sont proches d'une valeur propre, il faut les décaler (pour préserver la robustesse du processus). Parfois ce décalage est trop prononcé et il va donc conduire le test de Sturm à englober un intervalle trop grand comportant des fréquences non calculées (et non souhaitées).
Le test va alors alerter l'utilisateur parfois inutilement. Après s'être assuré qu'il ne s'agissait pas de fréquences multiples ratées proches des bornes de la bande, on peut alors le débrancher (`'NON'`) ou réduire les paramètres de décalage (passer de `PREC_SHIFT`=5% à 2% par exemple).

Par exemple, on teste l'intervalle $[100,500]$ et 499.5 et 520 sont des valeurs propres du problème. Du fait de la proximité de la valeur propre 499.5 avec la borne maximum 500, le test de Sturm va devoir décaler cette dernière. Par défaut elle va prendre la valeur 525. Cette nouvelle bande de test $[100,525]$ est maintenant trop importante car elle englobe la valeur 520 : le test va conclure, faussement, qu'il y' a un problème en comptant une fréquence en trop.
A contrario, si 500.1 avait été une valeur propre, le test de Sturm aurait sans doute bien fait d'alerter l'utilisateur.

Remarques:

- *En mode parallèle standard (`NIVEAU_PARALLELISME='COMPLET'`), il n'y a pas de possibilité de test de Sturm local. `STURM='GLOBAL'` ou `' LOCAL'` effectuent le même traitement : ils vérifient la validité du test de Sturm sur l'ensemble des sous-bandes de calcul.*
- *Ce test de post-vérification s'effectue en plus d'autres tests (non débrayables et indispensables) :*

9 Cf. Exemple du §10.

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

** Tests de convergence internes [10] au solveur modal ('SORENSEN', 'TRI_DIAG' et 'JACOBI') modulable via les mots-clés PREC_*.*
** Vérification des résidus (cf. [R5.01.01/02] algorithme n°2/n°1) de chaque mode calculé (cf. mot-clés SEUIL_FREQ et SEUIL).*
** On s'assure enfin que les fréquences exhumées pour chaque sous-bande appartiennent bien à l'intervalle choisi (à VERI_MODE/PREC_SHIFT % près).*

- dans les autres cas :

◊ STURM= /**'OUI'** **[DEFAUT pour 'PLUS_PETITE'/'PLUS_GRANDE'/'CENTRE'/'TOUT']**
/**'NON'** **[DEFAUT pour 'PROCHE'/'SEPARE'/'AJUSTE']**

Vérification dite de STURM permettant de s'assurer que l'algorithme utilisé dans l'opérateur a déterminé le nombre exact de valeurs propres dans l'intervalle de recherche (§3.5/6/8 [R5.01.01]). Cette option n'a d'intérêt qu'en GEP à modes réels (donc pas avec K complexe et avec des matrices non symétriques).
Pour la famille d'options de calcul **'PLUS_PETITE'/'PLUS_GRANDE'/'CENTRE'/'BANDE'/'TOUT',** le test de Sturm est réalisé à partir des bornes de sous-bandes fournies ou des valeurs extrêmes des modes calculés. Tandis que pour la seconde famille, **'PROCHE'/'SEPARE'/'AJUSTE'**, ces bornes sont déduites des valeurs extrêmes des listes fournies.
Notons que la première famille d'options doit calculer et vérifier un ou plusieurs paquets de modes contigus alors, que la seconde famille, affine elle une liste de valeurs fournies. La première famille ne tolère donc pas de trou dans le spectre calculé. Il ne faut ici rater aucun mode ! Alors, qu'il peut-être volontaire, avec la seconde famille, de fournir des valeurs éparses. Dans le premier cas, l'activation du test de Sturm est fortement conseillée (valeur par défaut), dans le second, elle est facultative et laisser à l'appréciation de l'utilisateur.
Voir aussi paramètre STOP_ERREUR .

3.10.4 Opérande PREC_SHIFT (uniquement pour la méthode des itérations simultanées)

◊ PREC_SHIFT=prs **(0.05)** **[DEFAUT]**

Ce paramètre (qui est un pourcentage) permet de définir un intervalle contenant les valeurs propres calculées, pour lequel la vérification de Sturm sera effectuée ([R5.01.01] algorithme n°2). Il est aussi utilisé pour sélectionner les lignes de la table fournie en cas de chaînage INFO_MODE+CALC_MODES+'BANDE' (cf. mots-clés TABLE_FREQ/TABLE_CHAR_CRIT).
Cette option n'a d'intérêt qu'en GEP à modes réels.

3.11 Opérande STOP_BANDE_VIDE (uniquement pour la méthode des itérations simultanées)

◊ STOP_BANDE_VIDE=/'NON' **[DEFAUT si TYPE_RESU='DYNAMIQUE' et OPTION='BANDE' et FREQ comporte n>2 fréquences]**
/'OUI' **[DEFAUT dans les autres cas]**

'OUI' arrête le calcul si aucune valeur propre n'est détectée dans la bande stipulée par l'utilisateur (ou les sous-bandes dans le cas où le mot-clé FREQ est une liste de n>2 valeurs) : une

10 Ces tests sont exprimés dans le contexte du « problème de travail » fournit pour chaque sous-bande au solveur modal. Souvent ce problème transformé est différent du problème initial. La bonne convergence de cette étape n'assure donc pas à 100% celle du problème initial.

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

exception (nommée BandeFrequenceVide) est émise. Elle peut être traitée pour continuer le déroulement de l'étude. On peut trouver un exemple dans le cas test SDLL11a :

```
try:
   MODE1=CALC_MODES( MATR_RIGI=K_ASSE,
                     MATR_MASS=M_ASSE,
                     OPTION='BANDE',
                     CALC_FREQ=_F(FREQ=(100.,200.)))
except aster.BandeFrequenceVideError:
   MODE1=CALC_MODES( MATR_RIGI=K_ASSE,
                     MATR_MASS=M_ASSE,
                     OPTION='BANDE',
                     CALC_FREQ=_F(FREQ=(200.,3500.,)))
```

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

Cette option n'a pas d'intérêt avec la méthode QZ.

3.12 Opérande NIVEAU_PARALLELISME

Ce mot-clé est disponible uniquement dans le cas de modes propres de vibrations (TYPE_RESU='DYNAMIQUE') et si le calcul est réalisé sur une bande fréquentielle (OPTION='BANDE') découpée en au moins 2 sous-bandes (FREQ est une liste de nb_freq > 2 valeurs).

◊ NIVEAU_PARALLELISME=/**'COMPLET'** **[DEFAUT]**
/'PARTIEL'

Le découpage en plusieurs sous-bandes est à privilégier lorsqu'on traite des problèmes **de tailles moyennes ou grandes** (> 0.5M ddls) et/ou que l'on cherche une **bonne partie de leurs spectres** (> 50 modes).
On découpe alors le calcul en plusieurs sous-bandes fréquentielles (cf. opérande FREQ). Sur chacune de ces sous-bandes, un solveur modal effectue la recherche de modes associée. Pour ce faire, ce solveur modal utilise intensivement un solveur linéaire.
Ces deux briques de calcul (solveur modal et solveur linéaire) sont les **étapes dimensionnantes** du calcul en terme de consommation mémoire et temps. C'est sur elles qu'il faut mettre l'accent si on veut réduire significativement les coûts de calcul de cet opérateur.

Or l'organisation du calcul modal sur des sous-bandes distinctes offre ici un cadre idéal de parallélisme : **distribution de gros calculs presque indépendants**[11]. Son parallélisme permet de gagner beaucoup en temps mais au prix d'un surcoût en mémoire[12].
Si on dispose d'un nombre de processeurs suffisant (supérieur au nombre de sous-bandes non vides), on peut alors enclencher un **deuxième niveau de parallélisme *via* le solveur linéaire** (si on a choisit METHODE='MUMPS'). Celui-ci permettra de continuer à gagner en temps mais surtout, il permettra de compenser le surcoût mémoire du premier niveau voire de diminuer notablement le pic mémoire séquentiel.

11 Aux coûteuses communications de vecteurs propres près.
12 Du fait des buffers MPI requis par les communications de vecteurs propres en fin de CALC_MODES.

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

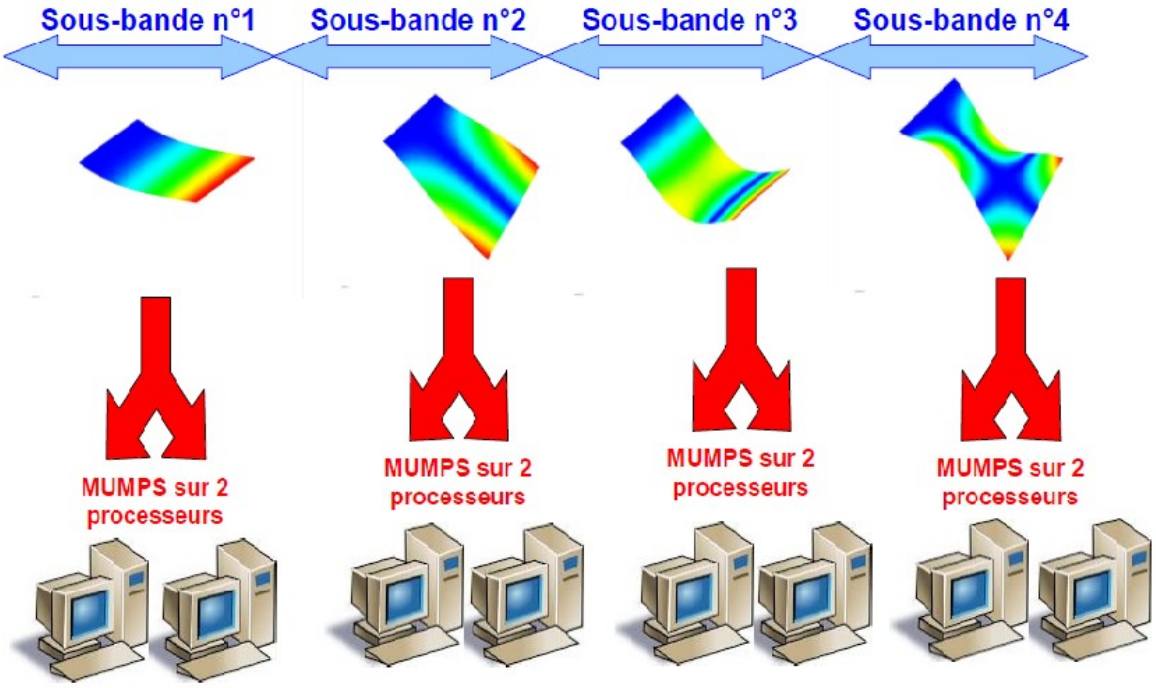


Image 3.12-1 - *Exemple de distribution des calculs de CALC_MODES sur 8 processeurs avec un découpage en 4 sous-bandes fréquentielles*

Ce double niveau de parallélisme (activé par défaut *via* le mot-clé NIVEAU_PARALLELISME='COMPLET') permet alors de tirer profit, au mieux, des deux aspects.
Lorsqu'on souhaite véritablement gagner en pic mémoire parce que le calcul ne passe pas sur la machine et que l'on a essayé, sans succès, tous les autres bras de levier[13], on peut choisir sciemment de limiter le parallélisme uniquement au niveau du solveur linéaire[14] : NIVEAU_PARALLELISME='PARTIEL'. Cela fonctionne uniquement avec le solveur linéaire parallèle MUMPS.

Les règles fonctionnelles sont les suivantes, en notant nbproc le nombre de processeurs paramétré et nb_sbande le nombre de sous-bandes non vides (=nb_freq-1) :

- Avec NIVEAU_PARALLELISME='COMPLET' (**défaut**) : très gros gain en temps/amélioration ou détérioration moyenne du pic mémoire RAM.

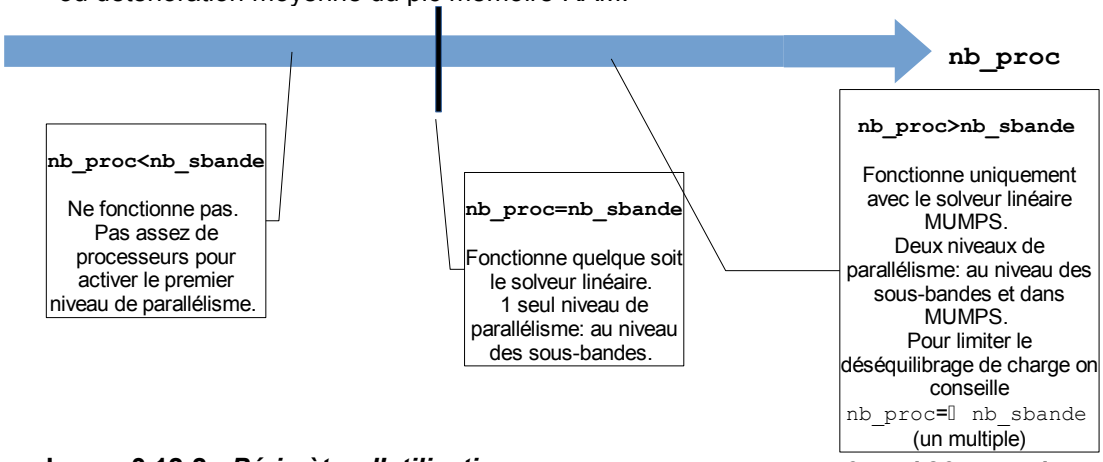


Image 3.12-2 - *Périmètre d'utilisation avec NIVEAU_PARALLELISME='COMPLET'.*

13 Découper en plus de sous-bandes, utiliser le solveur modal SORENSEN, réduire la taille de l'espace de projection *via* COEF_DIM_ESPACE, utiliser le solveur linéaire MUMPS en OUT_OF_CORE et/ou avec METIS...
14 C'est ce type de parallélisme qui est déployé dans le reste du code.

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

- Avec NIVEAU_PARALLELISME='PARTIEL' : gain modéré en temps/gain important sur le pic mémoire RAM.

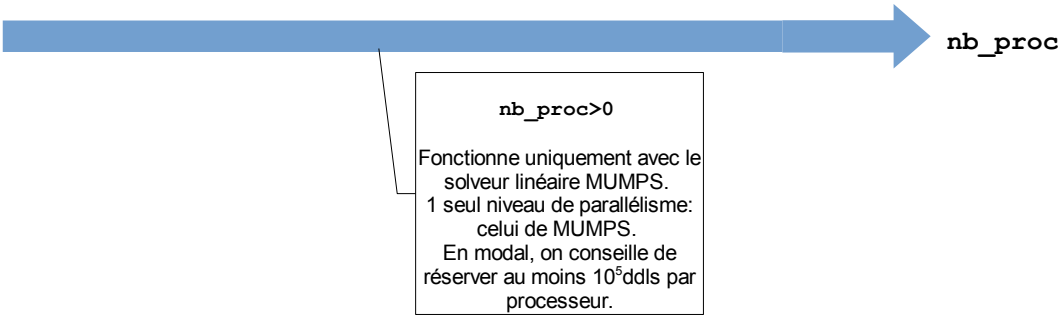


Image 3.12-3 - *Périmètre d'utilisation avec NIVEAU_PARALLELISME='PARTIEL'.*

Pour un **usage optimal** du parallélisme, il est donc conseillé de :

- Construire des sous-bandes de calcul relativement équilibrées. Pour ce faire, on peut donc, au préalable, calibrer le spectre étudié *via* un ou plusieurs appel à INFO_MODE[U4.52.01]. Si possible en mode parallèle. Puis lancer le calcul CALC_MODES parallèle en fonction du nombre de sous-bandes choisies et du nombre de processeurs disponibles.

- De prendre des sous-bandes plus fines qu'en séquentiel, entre 10 et 20 modes au lieu de 40 à 80 modes en séquentiel. La qualité des modes et la robustesse du calcul s'en trouvera accrue. Le pic mémoire en sera diminué. Il faut cependant avoir suffisamment de processeurs disponibles (et avec assez de mémoire).

- Sélectionner un nombre de processeurs qui est un multiple du nombre de sous-bandes (non vides). Ainsi, on réduit les déséquilibrages de charges qui nuisent aux performances.

Pour réduire le pic mémoire d'un calcul, on dispose de plusieurs bras de levier : réduire la taille des sous-bandes, utiliser le solveur linéaire MUMPS (éventuellement en OUT_OF_CORE[U4.50.01]) et/ou ne paralléliser que cette brique de calcul (NIVEAU_PARALLELISME='PARTIEL').

Pour utiliser efficacement **CALC_MODES en parallèle** , on propose donc de procéder en **trois étapes :**

- **Pré-calibrations modales** préalables *via* INFO_MODE. Si possible, en mode parallèle (Gains potentiels en temps x70 sur une centaines de processeurs. Gain en pic mémoire RAM jusqu'à x2).

- Examiner les résultats produits et décomposer le calcul en sous-bandes de tailles modestes (par ex. 20 modes) et équilibrées, en fonction du nombre de processeurs disponibles.

- Lancer en mode POURSUITE, le calcul CALC_MODES parallèle proprement dit.

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

Cas-test perf016a (N=4.0M, 50 modes) découpage en **8 sous-bandes**	**Temps elapsed**	**Pic mémoire RAM**
1 processeur	5524s	16.9Go
8 processeurs	1002s	19.5Go
32 processeurs	643s	13.4Go
découpage en **4 sous-bandes**		
1 processeur	3569s	17.2Go
4 processeurs	1121s	19.5Go
16 processeurs	663s	12.9Go

Tableau 3.12-1 - Résultats de *CALC_MODES* parallèle avec les paramètres par défaut (+ *SOLVEUR=MUMPS* en *IN_CORE* et *RENUM='QAMD'*) sur le cas-test PERF016A. Obtenu avec Code_Aster v11.3.11 sur la machine IVANOE (1 ou 2 processus MPI par noeud).

Étude sismique (N=0.7M, 450 modes) découpage en **20 sous-bandes**	**Temps elapsed**	**Pic mémoire RAM**
1 processeur	5200s	10.5Go
20 processeurs	407s	12.1Go
80 processeurs	270s	9.4Go
découpage en **5 sous-bandes**		
1 processeur	4660s	8.2Go
5 processeurs	1097s	11.8Go
20 processeurs	925s	9.5Go

Tableau 3.12-2 - Résultats de *CALC_MODES* parallèle avec les paramètres par défaut (+ *SOLVEUR=MUMPS* en *IN_CORE* et *RENUM='QAMD'*) sur une étude sismique. Obtenu avec Code_Aster v11.3.11 sur la machine IVANOE (1 ou 2 processus MPI par noeud).

Remarques:

- *En mode NIVEAU_PARALLELISME='COMPLET', si le nombre de processeurs n'est pas un multiple du nombre de sous-bandes (non vides), on distribue le reliquat de processeurs en privilégiant les premières sous-bandes. Un message avertit l'utilisateur du potentiel déséquilibrage de charge et du caractère sous-optimal du calcul.*
- *En mode NIVEAU_PARALLELISME='COMPLET' , on a désactivé le parallélisme des calculs élémentaires et des assemblages qui peuvent s'opérer dans NORM_MODE . Leur coût est de toute manière marginal.*
  Cette désactivation est temporaire et juste limitée à CALC_MODES. Cependant si la distribution parallèle des données, générée avant le CALC_MODES (via AFFE_MODELE/MODI_MODELE), a été produite avec d'autres paramètres que les valeurs par défaut (activation des mots-clés DISTRIBUTION + CHARGE_PROC0_MA ou NB_SOUS_DOMAINE/PARTITIONNEUR/CHARGE_PROC0_SD) et si l'on veut retrouver exactement la même distribution, il faut effectuer un appel explicite à MODI_MODELE à l'issue du CALC_MODES.
  Si on ne le fait pas, le partitionnement parallèle qui suivra le CALC_MODES reprendra les valeurs par défaut. C'est souvent le scénario le plus performant.

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

- *En mode NIVEAU_PARALLELISME='COMPLET', on communique tous les vecteurs propres exhumés en fin de CALC_MODES. Donc la distinction[15] entre les valeurs STURM='LOCAL' ou 'GLOBAL' n'a plus lieu d'être fonctionnellement. Ce n'est pas grave car le mode par défaut à privilégier est le mode 'GLOBAL'.*

Pour la mise en œuvre pratique du parallélisme, on se reportera au documents générique [U2.08.06] sur le parallélisme, et au paragraphe dédié de [U2.06.01] sur le calcul modal.

3.13 Mot-clé AMELIORATION

◊ AMELIORATION=/'NON' [DEFAUT]
/'OUI'

Mot-clé permettant d'améliorer de manière automatique la qualité des modes calculés : principalement les déformées modales (ce qui se traduit par une diminution de la norme d'erreur) et valeurs propres. Cette amélioration est réalisée par un chaînage de deux calculs modaux : un premier, de préférence réalisé par une méthode des itérations simultanées (mot-clé OPTION='BANDE' ou 'PLUS_PETITE' ou 'CENTRE' ou 'TOUT'); le second est réalisé de manière transparente pour l'utilisateur, avec la méthode des itérations inverses (OPTION='PROCHE') avec comme entrée les valeurs propres estimées grâce au premier calcul.

Remarques:
- *La qualité des modes est généralement déjà suffisante à l'issue d'un premier calcul. Cette option, malgré son coût, peut cependant s'avérer très intéressante pour des modèles compliqués.*
- *Lorsqu'on active cette option d'amélioration, les tests de post-vérification, normalement réalisés à l'issue de la première étape (suivant le paramétrage de VERI_MODE), ne sont effectués ici qu'à l'issue de la seconde.*

3.14 Mots-clés pour le post-traitement : NORM_MODE, FILTRE_MODE, IMPRESSION

Le mot-clé NORM_MODE est disponible pour tous les types de résultat, mais les deux autres le sont uniquement pour le cas de modes propres de vibrations (TYPE_RESU='DYNAMIQUE') et physiques réels (matrices d'entrée de type matr_asse_depl_r).

3.14.1 Mot-clé facteur NORM_MODE

Sert à définir les arguments pour la normalisation des modes. Tous les modes sont normés de la même façon. Les arguments sont les mêmes que pour la commande NORM_MODE [U4.52.11].

3.14.2 Mot-clé facteur FILTRE_MODE

S'il est présent, ce mot-clé sert à introduire les arguments de filtrage des modes selon un critère donné. Les arguments sont les mêmes que pour la commande EXTR_MODE [U4.52.12].

3.14.3 Mot-clé IMPRESSION

Permet d'afficher éventuellement le cumul de valeurs d'un paramètre modal choisi. Les mots-clés internes ont la même signification que dans la commande EXTR_MODE [U4.52.12].

15 La distinction entre les deux modes est juste ici d'ordre informatique : dans le cas 'GLOBAL', le test de Sturm est mis en œuvre au niveau du fichier PYTHON de la macro, alors que dans le cas 'LOCAL', il est opéré dans les sources Fortran de l'opérateur (invisible pour l'utilisateur) réalisant le calcul modal par la méthode des itérations simultanées.

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

Le paramètre modal choisi peut ne pas être le même que celui qui a servi éventuellement à filtrer les modes calculés.

Le mot-clé TOUT_PARA permet d'afficher, après normalisation éventuelle, la valeur de tous les paramètres modaux contenus dans la structure de données produite (fréquence, masses effectives, ...). Cet affichage est activé dans le cas de modes généralisés.

3.15 Opérande INFO

◊ INFO=/**1** **[DEFAUT]**
/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.

2: Impression plutôt réservée aux développeurs.

Tableau 3.15-1 - Fonctionnement du mot-clé INFO selon sa valeur.

3.16 Opérande TITRE

◊ TITRE=ti

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

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

4 Phase de vérification

Les matrices $\mathbf{A}$, $\mathbf{B}$ et $\mathbf{C}$, arguments des mot-clés (MATR_A/MATR_RIGI), (MATR_MASS/MATR_RIGI_GEOM/MATR_B) et (MATR_AMOR), doivent être cohérentes entre elles (c'est à dire s'appuyer sur la même numérotation et le même mode de stockage).

On vérifie, selon l'option de recherche choisie :

OPTION='BANDE'
L'argument du mot-clé FREQ (cas TYPE_RESU='DYNAMIQUE') doit fournir au moins **deux** valeurs.
L'argument du mot-clé CHAR_CRIT (cas TYPE_RESU='MODE_FLAMB' ou 'GENERAL') 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='SEPARE' ou 'AJUSTE'
L'argument du mot-clé FREQ ou du mot-clé CHAR_CRIT doit fournir au moins **deux** valeurs.

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.

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

5 Phase d'exécution

• Dans le cas où c'est la méthode des itérations simultanées qui est utilisée (i.e. si OPTION='BANDE' ou 'CENTRE' ou 'PLUS_PETITE' ou 'PLUS_GRANDE' ou 'TOUT') :
si le mot-clé DIM_SOUS_ESPACE (sous le mot-clé facteur SOLVEUR_MODAL) n'est pas renseigné ou est initialisé à une valeur strictement inférieure au nombre de modes demandés nf (opérande NMAX_FREQ ou NMAX_CHAR_CRIT), l'opérateur calcule automatiquement une dimension admissible pour le sous-espace de projection *via* les formules empiriques (cf §3.5.1.4) :

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 degrés de liberté actifs (c'est-à-dire le nombre total de degrés de liberté moins le nombre de degrés de liberté de LAGRANGE et moins le nombre de relations linéaires qui lient des degrés de liberté entre eux, [R5.01.01] §3.2) et mse est le facteur de proportionnalité fixé par COEF_DIM_ESPACE.

Si l'on résout un GEP, la dimension du sous-espace est doublée. Les valeurs de ces différents paramètres sont imprimées dans le fichier MESSAGE.

• Pour l'option 'SEPARE' : ayant obtenu un intervalle cernant une valeur propre, on prend pour le calcul du mode le milieu de l'intervalle. Lors du calcul du mode, la valeur de la valeur propre est encore affinée. C'est le résultat de l'itération inverse proprement dit.

• Pour l'option 'AJUSTE
' : si la séparation n'est pas possible et que dans un intervalle donné il y a plus d'une valeur de valeur propre, on n'applique pas la méthode d'ajustement à cet intervalle. Par contre, on effectuera lors du calcul des modes des réorthogonalisations par rapport aux modes précédents contenus dans l'intervalle (ceci permet de calculer des modes associés à une valeur propre multiple).

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

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 le mot-clé facteur NORM_MODE à l'intérieur de CALC_MODES, ou la commande NORM_MODE [U4.52.11] à la suite de CALC_MODES (on a dans ce accès à plus de fonctionnalités).

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 généralisé à matrices réelles symétriques, la position modale correspond à la position du mode dans l'ensemble du spectre défini par les matrices initiales.
Dans tous les autres cas, les positions modales sont attribuées de 1 à nf (nf étant le nombre de modes retenus) en les classant par ordre croissant algébrique. Toutes les positions modales sont donc positives.

7 Optimisation des performances CPU

7.1 Parallélisme du solveur linéaire

Le calcul modal fait appel à un solveur linéaire. Si on choisit le solveur linéaire MUMPS avec le mot-clé facteur SOLVEUR, opérande METHODE='MUMPS', on peut activer son fonctionnement en parallèle. Il faut pour cela utiliser une version parallèle de *Code_Aster*, et renseigner le nombre de processeurs à utiliser. Empiriquement, l'utilisation d'un processeur par tranche de 10^5 à 10^6 degrés de liberté que comporte le modèle, donne de bonnes performances. En parallèle, on recommande aussi le paramétrage RENUM='QAMD'.

7.2 Calcul des modes par sous-bandes

Si le problème à traiter est un GEP et qu'on recherche beaucoup de modes sur une bande donnée ou que le nombre de degrés de liberté du modèle est élevé (> 10^5), il est conseillé de découper la bande de recherche globale en plusieurs sous-bandes, de 40 à 80 modes, les plus équilibrées possibles. Le calcul sera ainsi plus robuste et plus rapide, même en appelant séquentiellement un CALC_MODES pour chaque sous-bande.
Pour une meilleure ergonomie, les calculs sur chaque sous-bande peuvent être réalisés de manière automatique et transparente pour l'utilisateur, en donnant au mot-clé FREQ un liste de n>2 valeurs. Cela permet en outre de paralléliser le traitement des différentes sous-bandes, engendrant des gains considérables sur les performances CPU (facteur pouvant atteindre 10 à 20 sur le temps de calcul, et plusieurs dizaines de pourcents sur le pic mémoire).

Pour plus de détails, on se reportera aux documentations génériques [U2.08.06] sur le parallélisme, et [U2.06.01] sur le calcul de modes propres.

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

8 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'));
```


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

9 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.
Le mot-clé facteur FILTRE_MODE permet d'extraire dans la structure de données de type mode_meca_* des modes qui vérifient un certain critère. On peut aussi utiliser la commande a commande EXTR_MODE[U4.52.12] après CALC_MODES, afin par exemple de concaténer plusieurs structures de données de type mode_meca_*.

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

10 Exemples

10.1 Calcul des 5 modes propres les plus proches d'une fréquence donnée

```
mode=CALC_MODES( MATR_RIGI=rigid,
                 MATR_MASS=masse,
                 OPTION='CENTRE',
                  CALC_FREQ=_F( FREQ=100.,
                                NMAX_FREQ=5)
               );
```

10.2 Calcul des charges critiques contenues dans une bande

```
mode=CALC_MODES( TYPE_RESU='MODE_FLAMB',
                  MATR_RIGI=rigid,
                  MATR_RIGI_GEOM=riggeo,
                 OPTION='BANDE',
                  CALC_FREQ=_F(
                            CHAR_CRIT=(-1.E8,1.5E8))
                           );
```

10.3 Chaînage INFO_MODE+CALC_MODES (extrait de SDLS504a)

```
nbmod1 = INFO_MODE( TYPE_MODE='MODE_FLAMB',
                    ...
                    CHAR_CRIT=(-1.E+6,-5.E+5,0.0,1.E+5,1.1E+6),)

RESULT0=CALC_MODES( TYPE_RESU='MODE_FLAMB',
                    ...,
                    OPTION='BANDE',
                    CALC_CHAR_CRIT=_F( TABLE_CHAR_CRIT=nbmod1,
                                       CHAR_CRIT=(-1.E+06,1.E+05)),
                  )
```

10.4 Calcul des fréquences propres contenues dans la bande [50 ; 150] Hz

avec la méthode des puissances inverses

```
   mode=CALC_MODES(MATR_RIGI=rigidite,
                   MATR_MASS=masse,
                   OPTION='AJUSTE',
                   CALC_FREQ=_F( FREQ=(50.,150.))
                  )
```

10.5 Calcul des fréquences propres les plus proches de 20 et 50 Hz

avec la méthode des puissances inverses, avec accélération de convergence par le coefficient de Rayleigh

```
   mode= CALC_MODES( MATR_RIGI=rigidite,
                     MATR_MASS=masse,
                     OPTION='PROCHE',
                     CALC_FREQ=_F( FREQ=(50.,150.)),
                     SOLVEUR_MODAL=_F(OPTION_INV='RAYLEIGH')
```


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

```
                          )
```

10.6 Découpage en plusieurs sous-bandes

Soit la séquence suivante :

```
    mode=CALC_MODES(
          MATR_RIGI=rigi,
          MATR_MASS=masse,
          OPTION='BANDE',
          CALC_FREQ=_F( FREQ=(1.,3.,5.),)
          VERI_MODE=_F(),

          NORM_MODE=_F(NORME='TRAN_ROTA',),

          FILTRE_MODE=_F(CRIT_EXTR='MASS_EFFE_UN'),

          IMPRESSION=_F(CUMUL='OUI',
                        CRIT_EXTR='MASS_EFFE_UN')
                              );
```

On veut donc chercher tous les modes compris dans la bande globale $[1.,5.]$ Hz en la découpant en deux sous-bandes fréquentielles : $[1.,3.]$ et $[3.,5.]$.
Une fois interprétée, la macro-commande consiste en l'enchaînement de commandes usuelles décrit ci-dessous.

- Etape 1 : Détermination du nombre de fréquences dans chaque sous-bandes

```
      table1=INFO_MODE( MATR_RIGI=rigi,
                        MATR_MASS=masse,
                        FREQ=(1.,3.,5.))
```

Calcul du nombre de fréquences théoriques de la bande globale[16] : nbmodeth
Si la bande globale est vide : ALARME ou ERREUR_FATALE suivant la valeur de CALC_FREQ/STOP_BANDE_VIDE.

- Etape 2 : Calcul et normalisation des modes dans chaque sous-bande

Pour économiser les coûts calcul, on réutilise la table générée précédemment [17] et, par défaut, on ne fait pas localement à chaque sous-bande le test de Sturm de post-vérification.
Si la sous-bande locale est vide : ALARME ou ERREUR_FATALE suivant la valeur de CALC_FREQ/STOP_BANDE_VIDE.

```
      mode_1=CALC_MODES( MATR_RIGI=rigi,
                         MATR_MASS=masse,
                         OPTION='BANDE',
                         CALC_FREQ=_F( FREQ=(1.,3.),
                                       TABLE_FREQ=table1),),
                         VERI_MODE(STURM='NON'));
      mode_1=NORM_MODE( MODE=mode_1,
                        reuse=mode1,
                       NORME='TRAN_ROTA',);
      mode_2=CALC_MODES( MATR_RIGI=rigi,
                         MATR_MASS=masse,
                         OPTION='BANDE',
                         CALC_FREQ=_F( FREQ=(3.,5.),
                                       TABLE_FREQ=table1),),
```

16 On somme juste les nombres de fréquences calculés précédemment et stockés dans table1.
17 Pour ne pas refaire le test de Sturm de prétraitement propre à chaque sous-bande.

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

```
                          VERI_MODE(STURM='NON'));
      mode_2=NORM_MODE( MODE=mode_2,
                        reuse=mode2,
                            NORME='TRAN_ROTA',);
```

Vérification par un test de Sturm global du bon nombre de fréquences calculées
Détermination de la plus petite (resp. grande) fréquence de la première (resp. dernière) sous-bande non vide : freq_ini (resp. freq_fin).
Calcul du nombre de fréquences comprises dans l'intervalle : [freq_ini,freq_fin] : nbmodeef.

```
      table2=INFO_MODE( MATR_RIGI=rigi,
                        MATR_MASS=masse,
                        FREQ=(freq_ini,freq_fin))
```

Si ce nombre de modes est différent du nombre de modes prévu initialement : ERREUR_FATALE .

- Étape 3 : Filtrage, concaténation et impression des modes calculés.

```
      mode=EXTR_MODE( FILTRE_MODE=_F( MODE=mode_1,
                                      CRIT_EXTR='MASS_EFFE_UN'),
                      FILTRE_MODE=_F( MODE=mode_2,
                                      CRIT_EXTR='MASS_EFFE_UN'),
                      IMPRESSION=_F( CUMUL='OUI',
                                     CRIT_EXTR='MASS_EFFE_UN'),
                    );
```


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