

---

## Opérateur `MODE_ITER_SIMULT`

---

### 1 But

---

Que cela soit pour étudier les vibrations d'une structure (éventuellement amortie ou tournante) ou rechercher ses modes de flambement, le mécanicien doit souvent résoudre un problème modal: soit **généralisé (GEP)** [R5.01.01], soit **quadratique (QEP)** [R5.01.02]. Pour ce faire, *Code\_Aster* propose deux opérateurs (et leurs macros): `MODE_ITER_SIMULT` et `MODE_ITER_INV` [U4.52.04].

Le **premier**, qui nous occupe dans cette note, **est plutôt à utiliser lorsqu'on cherche une partie significative du spectre** (méthodes de sous-espace ou méthode globale). Le second est à privilégier lorsqu'on s'intéresse à seulement quelques modes propres (typiquement une demi-douzaine) ou lorsqu'on souhaite affiner quelques estimations (éventuellement provenant de `MODE_ITER_SIMULT`).

`MODE_ITER_SIMULT` détermine un **ensemble de modes propres**, soit par une méthode de type sous-espace (Bathe & Wilson, Lanczos ou Sorensen) soit par une méthode globale de type QR (QZ pour les petits problèmes). Ses quatre méthodes sont accessibles pour traiter un **GEP standard** (symétrique réel): calcul dynamique classique (sans amortissement et sans effet gyroscopique) ou problème de flambement d'Euler.

Pour traiter efficacement ce type de GEP, on propose d'ailleurs de procéder en plusieurs étapes :

- **Calibrer** les zones d'intérêt par un `INFO_MODE` [U4.52.01] initial sur une liste de fréquences (resp. charges critiques) donnée,
- Regarder les nombres de modes propres associés,
- **Relancer** un ou plusieurs calculs `MODE_ITER_SIMULT+OPTION='BANDE'` en essayant d' **équilibrer les sous-bandes et en réutilisant** la `sd_table` générée par `INFO_MODE`. Pour optimiser les performances CPU, il est aussi possible de paralléliser les calculs sur les sous-bandes grâce à l'opérateur `CALC_MODES + OPTION='BANDE'` [U4.52.02] en parallèle, qui peut procurer un gain d'un facteur 10 à 20 sur le temps de calcul, et de plusieurs dizaines de pourcents sur le pic de mémoire.

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+MODE_ITER_SIMULT` n'est alors plus possible. Certains mots-clés ou valeurs deviennent sans objet ( `OPTION='BANDE'`, `VERI_MODE/STURM...` ) et seules les méthodes Sorensen, Lanczos ou QZ restent disponibles.

Cet opérateur produit un concept `mode_meca_*` (cas dynamique) ou `mode_flamb` (cas flambement d'Euler, seulement en QEP) suivant la valeur renseignée dans le mot-clé `TYPE_RESU`.

Dans une première approche on peut se contenter de renseigner les paramètres: `MATR_*` , `TYPE_RESU`, `CALC_*/OPTION` et `FREQ` (ou `CHAR_CRIT` ).

Dans certains cas de figure, il peut être plus intéressant d'utiliser la macro-commande `CALC_MODES` avec l'option '`BANDE`' découpée en plusieurs sous-bandes [U4.52.02].

## Table des Matières

1 But.....	1
2 Syntaxe.....	5
3 Opérandes.....	11
3.1 Principes .....	11
3.2 Opérandes MATR_RIGI/MATR_A/MATR_MASS/MATR_RIGI_GEOM /MATR_B/MATR_AMOR/MATR_C.....	14
3.3 Mot clé TYPE_RESU.....	14
3.4 Mot clé METHODE.....	15
3.4.1 Opérandes d'IRAM (si METHODE='SORENSEN').....	16
3.4.2 Opérandes de la méthode de Lanczos (si METHODE='TRI_DIAG').....	16
3.4.3 Opérandes de la méthode de Bathe & Wilson (si METHODE='JACOBI').....	16
3.4.4 Opérandes de la méthode QZ (si METHODE='QZ').....	17
3.5 Mot clé CALC_FREQ.....	17
3.5.1 Opérande FREQ.....	17
3.5.2 Opérande AMOR_REDUIT.....	18
3.5.3 Opérande NMAX_FREQ.....	18
3.6 Mot clé CALC_CHAR_CRIT.....	19
3.6.1 Opérande CHAR_CRIT.....	19
3.6.2 Opérande NMAX_CHAR_CRIT.....	20
3.7 Opérande APPROCHE.....	20
3.8 Opérande OPTION.....	20
3.9 Opérandes SEUIL_FREQ / SEUIL_CHAR_CRIT, PREC_SHIFT, NMAX_ITER_SHIFT.....	21
3.10 Opérande DIM_SOUS_ESPACE.....	23
3.11 Mot-clé facteur VERI_MODE.....	23
3.11.1 Opérande STOP_ERREUR.....	23
3.11.2 Opérande SEUIL.....	23
3.11.3 Opérande STURM.....	23
3.11.4 Opérande PREC_SHIFT.....	24
3.12 Opérande STOP_BANDE_VIDE.....	24
3.13 Opérande INFO.....	24
3.14 Opérande TITRE.....	24
4 Phase de vérification.....	26
5 Phase d'exécution.....	27
5.1 Vérification.....	27
5.2 Actions par défaut.....	27
6 Paramètres modaux/ Norme des modes/ Position modale.....	28
7 Optimisation des performances CPU.....	28
7.1 Parallélisme du solveur linéaire.....	28
7.2 Calcul des modes par sous-bandes.....	28

<a href="#">8 Impression des résultats .....</a>	<a href="#">29</a>
<a href="#">9 Tri de modes/ Caractérisation de mode_meca_*</a>	<a href="#">30</a>
<a href="#">10 Exemples.....</a>	<a href="#">31</a>
<a href="#">10.1 Calcul des 5 modes propres les plus proches d'une fréquence donnée ().....</a>	<a href="#">31</a>
<a href="#">10.2 Calcul des charges critiques contenues dans une bande.....</a>	<a href="#">31</a>
<a href="#">10.3 Chaînage INFO_MODE+MODE_ITER_SIMULT (extrait de SDLS504a).....</a>	<a href="#">31</a>

## 2 Syntaxe

```
mode_ [*]=MODE_ITER_SIMULT
```

### # Type de problème

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

```
# 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_MASS=B [matr_asse_DEPL_R]
                [matr_asse_PRES_R]
                [matr_asse_GENE_R]
  ◇ MATR_AMOR=C (uniquement en QEP) [matr_asse_DEPL_R]
                [matr_asse_GENE_R]
  ◇ CALC_FREQ=_F(◇ OPTION=/'CENTRE'
                  /'BANDE'
                  /'PLUS_PETITE' [DEFAULT]
                  /'PLUS_GRANDE'
                  /'TOUT'(uniquement avec QZ))
```

#### # Si OPTION='PLUS\_PETITE'

```
◇ NMAX_FREQ=/10 [DEFAULT]
/nf [I]
```

#### # Si OPTION='PLUS\_GRANDE'

```
◇ NMAX_FREQ=/1 [DEFAULT]
/nf [I]
```

#### # Si OPTION='CENTRE'

```
◆ FREQ=1_f [1_R]
◇ AMOR_REDUIT=1_a [1_R]
◇ NMAX_FREQ=/10 [DEFAULT]
/nf [I]
```

#### # Si OPTION='BANDE'

```
◆ FREQ=1_f [1_R]
◇ TABLE_FREQ=table_f table_sdaster)
)
```

### # Paramètre propre au QEP

```
◇ APPROCHE=/'REEL' [DEFAULT]
              /'IMAG'
              /'COMPLEXE' (uniquement Sorensen)
```

### # Caractéristiques de l'espace de projection

```
◇ DIM_SOUS_ESPACE=dse [I]
◇ COEF_DIM_ESPACE=mse [I]
EXCLUS ('DIM_SOUS_ESPACE', 'COEF_DIM_ESPACE')
```

### # Pour pré et post-traitements

```
◇ PREC_SHIFT=/0.05 [DEFAULT]
```

	/p_shift	[R]
◇	NMAX_ITER_SHIFT=/3	<b>[DEFAULT]</b>
	/n_shift	[I]
◇	SEUIL_FREQ=/1.E-2	<b>[DEFAULT]</b>
	/f_seuil	[R]

```
# 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( ♦ OPTION='CENTRE'  
    / 'BANDE'  
    / 'PLUS_PETITE' [DEFAULT]
```

### # Si OPTION='PLUS\_PETITE'

```
    ♦ NMAX_CHAR_CRIT= /10 [DEFAULT]  
    /nf [I]
```

### # Si OPTION='CENTRE'

```
    ♦ CHAR_CRIT= l_c [1_R]  
    ♦ NMAX_CHAR_CRIT= /10 [DEFAULT]  
    /nf [I]
```

### # Si OPTION='BANDE'

```
    ♦ CHAR_CRIT=l_c [1_R]  
    ♦ TABLE_CHAR_CRIT=table_c [table_sdaster]  
    )
```

## # Paramètre propre au QEP

```
    ♦ APPROCHE= /'REEL' [DEFAULT]  
    /'IMAG'
```

## # Caractéristiques de l'espace de projection

```
    ♦ DIM_SOUS_ESPACE=dse [I]  
    ♦ COEF_DIM_ESPACE=mse [I]  
    EXCLUS('DIM_SOUS_ESPACE', 'COEF_DIM_ESPACE')
```

## # Pour pré et post-traitements

```
    ♦ PREC_SHIFT=/0.05 [DEFAULT]  
    /p_shift [R]  
    ♦ NMAX_ITER_SHIFT=/3 [DEFAULT]  
    /n_shift [I]  
    ♦ SEUIL_CHAR_CRIT=/1.E-2 [DEFAULT]  
    /c_seuil [R]
```

```
# 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( ◊ OPTION=/'CENTRE'
                        / 'BANDE'
                        / 'PLUS_PETITE' [DEFAULT]
                        / 'TOUT' (uniquement avec QZ)

# Si OPTION='PLUS_PETITE'
  ◊ NMAX_CHAR_CRIT= /10 [DEFAULT]
                        /nf [I]

# Si OPTION='CENTRE'
  ♦ CHAR_CRIT=l_c [l_R]
  ◊ NMAX_CHAR_CRIT= /10 [DEFAULT]
                        /nf [I]

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

## # Caractéristiques de l'espace de projection

```
◊ DIM_SOUS_ESPACE=dse [I]
◊ COEF_DIM_ESPACE=mse [I]
  EXCLUS('DIM_SOUS_ESPACE', 'COEF_DIM_ESPACE')
```

## # Pour pré et post-traitements

```
◊ PREC_SHIFT=/0.05 [DEFAULT]
  /p_shift [R]
◊ NMAX_ITER_SHIFT=/3 [DEFAULT]
  /n_shift [I]
◊ SEUIL_CHAR_CRIT=/1.E-2 [DEFAULT]
  /c_seuil [R]
```

## # Choix de la méthode

```
◇ METHODE= /'SORENSEN' [DEFAULT]
              /'TRI_DIAG' (uniquement GEP/QEP symétriques réels)
              /'JACOBI' (sauf en QEP)
              /'QZ' (problème de petites tailles < 103 degrés de liberté)
```

## # Paramétrage interne des méthodes

### # Si METHODE='SORENSEN'

```
◇ PREC_SOREN=/0 [DEFAULT]
              /pso [R]
◇ NMAX_ITER_SOREN= /20 [DEFAULT]
              /nso [I]
◇ PARA_ORTHO_SOREN=/0.717 [DEFAULT]
              /porso [I]
```

### # Si METHODE='TRI\_DIAG'

```
◇ PREC_ORTHO=/1.E-12 [DEFAULT]
              /po [R]
◇ NMAX_ITER_ORTHO=/5 [DEFAULT]
              /nio [I]
◇ PREC_LANCZOS=/1.E-8 [DEFAULT]
              /pl [R]
◇ NMAX_ITER_QR=/30 [DEFAULT]
              /nim [I]
◇ OPTION=/'SANS' [DEFAULT]
              /'MODE_RIGIDE'
```

### # Si METHODE='JACOBI'

```
◇ PREC_BATHE=/1.E-10 [DEFAULT]
              /pbat [R]
◇ NMAX_ITER_BATHE=/40 [DEFAULT]
              /nbat [I]
◇ PREC_JACOBI=/1.E-2 [DEFAULT]
              /pjaco [R]
◇ NMAX_ITER_JACOBI=/12 [DEFAULT]
              /njaco [I]
```

### # Si METHODE='QZ'

```
◇ TYPE_QZ=/'QZ_SIMPLE' [DEFAULT]
              /'QZ_EQUI'
              /'QZ_QR' (si GEP à matrices SPD)
```

## # Pour vérifications finales

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

## # Divers

```
◇ STOP_BANDE_VIDE=/'OUI' [DEFAULT]
                  /'NON'
```

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

◇ INFO=/1                                     [DEFAULT]
  /2                                           [I]
◇ TITRE=ti

);
```

## # Résultats du problème modal

```
Si MATR_AMOR ou MATR_C=[matr_asse_DEPL_R] alors [*]->mode_meca_c
Si TYPE_RESU='MODE_FLAMB' alors [*]->mode_flamb
Si TYPE_RESU='GENERAL' alors [*]->mode_flamb
Si MATR_RIGI ou MATR_A=[matr_asse_DEPL_C] alors [*]->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
```

## 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

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

où  $\lambda = (2\pi f)^2$  est le carré de la pulsation  $\omega$ ,  $f$  la fréquence propre et  $\mathbf{x}$  le vecteur de déplacement propre associé. Les modes propres manipulés  $(\lambda, \mathbf{x})$  sont à valeurs réelles. Ce type de problématique est activé par le mot-clé `TYPE_RESU='DYNAMIQUE'` et génère une structure de données Aster 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  $\mathbf{x}$  associé à ce niveau de charge critique  $\mu = -\lambda$ , se ramène à un problème généralisé aux valeurs propres de la forme

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

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

#### Attention:

- Dans le code, on ne traite que les valeurs propres du problème généralisé, les variables  $\lambda$ . Pour obtenir les véritables charges critiques, les variables  $\mu$ , 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 `MODE_ITER_SIMULT (METHODE='SORENSEN'/'OZ')`.

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  $(l, \mathbf{x})$ .

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

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

où typiquement, en mécanique linéaire, **A** sera la matrice de rigidité, **B** la matrice de masse et **C** la matrice d'amortissement. Les matrices **A**, **B** et **C** sont des matrices symétriques et réelles. La valeur propre complexe  $\lambda$  est reliée à la fréquence propre  $f$  et à l'amortissement réduit  $\xi$  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 Aster de type `mode_meca_c`.

### Attention:

- En QEP, pour traiter des problèmes à matrices non symétriques et/ou à valeurs complexes, il faut utiliser `MODE_ITER_SIMULT (METHODE='SORENSEN'/'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 [R5.01.01/02], on peut les synthétiser sous la forme du tableau ci-dessous (les valeurs par défaut sont matérialisées en gras).

Opérateur/ Périmètre d'application	Algorithme	Mot-clé	Avantages	Inconvénients
<b>MODE_ITER_INV</b>				Uniquement symétrique réel (GEP et QEP).
<i>1<sup>ère</sup> phase (heuristique)</i>				
Calcul de quelques modes	Bissection (sans objet en QEP).	'SEPRE'		
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
<i>2<sup>ième</sup> phase (méthode des puissances inverses)</i>				
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
<b>MODE_ITER_SIMULT</b>				
Calcul d'une partie du spectre	Bathe & Wilson	'JACOBI'		Peu robuste Uniquement symétrique

Opérateur/ Périmètre d'application	Algorithme	Mot-clé	Avantages	Inconvénients
				<i>réel (GEP)</i>
	Lanczos (Newman-Pipano en GEP et Jennings en QEP)	'TRI_DIAG'	Détection spécifique des modes rigides.	<i>Uniquement symétrique réel (GEP et QEP)</i>
	IRAM (Sorensen)	'SORENSEN'	Robustesse accrue. Meilleures complexités calcul et mémoire. Contrôle de la qualité des modes.	Méthode par défaut. <i>Portée en non symétrique et avec A complexe symétrique.</i>
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. A réserver aux petits cas ( $<10^3$ degrés de liberté). <i>Portée en non symétrique et avec A complexe symétrique.</i>

Tableau 3.1-1. Récapitulatif des méthodes modales de Code\_Aster

Lorsqu'il s'agit de déterminer **quelques valeurs propres simples bien discriminées** ou d'affiner quelques estimations, l'opérateur `MODE_ITER_INV` (heuristique + puissance inverse), est souvent bien indiqué. Par contre, pour capturer une partie significatif du spectre, on a recourt à `MODE_ITER_SIMULT`, via les méthodes de sous-espace (Lanczos, IRAM, Jacobi) ou la méthode globale QZ (méthode très robuste mais coûteuse; à réserver aux petits cas).

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

Pour les **méthodes de sous-espace**, elle consiste à projeter le problème sur un espace dont la taille est supérieure au nombre de valeurs propres souhaitées mais très inférieure à celle du problème. On s'arrange pour que ce problème ait un spectre très proche de celle du problème initial et qu'il prenne une forme canonique (tridiagonale, Hessenberg etc.). Puis on applique un solveur modal global (Jacobi pour Bathe & Wilson, QR pour Lanczos/IRAM) sur ce problème simplifié. Enfin on convertit les modes obtenus dans l'espace de travail initial.

Quant à la méthode globale QZ, elle résoud directement et entièrement le problème initial (GEP ou QEP linéarisé) pour améliorer la robustesse du processus. Elle présente toutefois l'inconvénient de calcul tout le spectre. Elle est donc à réserver aux petits cas ( $<10^3$  degrés de liberté).

**Il est d'ailleurs tout à fait recommandé de profiter des points forts des deux classes de méthode en affinant les vecteurs propres obtenus par `MODE_ITER_SIMULT`, via `MODE_ITER_INV(OPTION='PROCHE')`.** Cela permettra de réduire la norme du résidu final (cf. §3.6.2).

D'autre part, pour traiter plus efficacement un GEP standard, on propose de procéder en plusieurs étapes:

- **Calibrer** les zones d'intérêt par un appel initial à `INFO_MODE` sur une liste de fréquences <sup>1</sup> (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 `MODE_ITER_SIMULT+OPTION='BANDE'` (ou `CALC_MODES`) en essayant d'**équilibrer** les bandes.

Pour gagner du temps, on peut même mutualiser (et c'est fortement conseillé !) une partie du coût calcul de l'`INFO_MODE` initial en notifiant aux `MODE_ITER_SIMULTs` 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+MODE_ITER_SIMULT` n'est alors plus possible. Certains mots-clés ou valeurs deviennent sans objet (`OPTION='BANDE'`, `VERI_MODE/STURM...`) et seules les méthodes Sorensen, Lanczos ou QZ restent disponibles.

**Remarque:**

- On conseille fortement une lecture préalable des documentations de référence [R5.01.01] [R5.01.02] [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, qui peuvent parfois paraître un peu éthérées, à un paramétrage précis des options.

## 3.2 Opérandes `MATR_RIGI/MATR_A/MATR_MASS/MATR_RIGI_GEOM/MATR_B/MATR_AMOR/MATR_C`

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

TYPE_RESU		
'DYNAMIQUE'	'MODE_FLAMB'	'GENERAL'
◆ <code>MATR_RIGI = A</code>	◆ <code>MATR_RIGI = A</code>	◆ <code>MATR_A = A</code>
◆ <code>MATR_MASS = B</code>	◆ <code>MATR_RIGI_GEOM = B</code>	◆ <code>MATR_B = B</code>
◇ <code>MATR_AMOR = C</code>	Sans objet	Hors périmètre actuel

- ◆ `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.
- ◆ `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/MATR_C=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 réelles, seule les méthodes de Sorensen et QZ sont licites. On ne peut alors utiliser l'option de calcul '`BANDE`' et, pour le cas où la matrice `A` est complexe, une borne fréquentielle nulle (`OPTION=PLUS_PETITE` ou '`CENTRE`' avec  $f = 0$  ).

<sup>1</sup> Pour des raisons de coûts calcul, il vaut mieux limiter la liste à une douzaine de valeurs, au maximum.

## 3.3 Mot clé `TYPE_RESU`

```
◇ TYPE_RESU=/ 'DYNAMIQUE'           [DEFAULT]
              / '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 la spectre de la valeur propre (déterminée par le test de Sturm cf. §3.6 [R5.01.01]). 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'` ou `TYPE_RESU='GENERAL'`), les valeurs propres sont stockées par ordre croissant algébrique. Les variables `NUME_ORDRE` et `NUME_MODE` prennent la même valeur égale à cette ordre.

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/B` plutôt que `MATR_RIGI/MATR_RIGI_GEOM`.

## 3.4 Mot clé `METHODE`

Quatre méthodes de résolution sont disponibles pour résoudre le problème aux valeurs propres (cf. tableau 3.1.1):

```
◇ METHODE=/ 'SORENSEN'           [DEFAULT]
```

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.

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

Mot-clé utilisable seulement avec la méthode de Lanczos pour un GEP. Il 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 valeurs propres calculées par l'algorithme de Lanczos ne sont pas nulles mais très voisines de zéro.

```
◇           / '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, 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).

### 3.4.1 Opérandes d'IRAM (si `METHODE= 'SORENSEN'` )

◇ `PREC_SOREN=pso` (0.) [DEFAULT]

#### Remarque :

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

◇ `NMAX_ITER_SOREN=nso` (20) [DEFAULT]  
◇ `PARA_ORTHO_SOREN=porso` (0.717) [DEFAULT]

Il s'agit de paramètres d'ajustement de la précision requise sur les modes (par défaut, la précision machine est choisie), du nombre de redémarrages autorisé de la méthode de Sorensen (cf. [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.

### 3.4.2 Opérandes de la méthode de Lanczos (si `METHODE= 'TRI_DIAG'` )

◇ `PREC_ORTHO =po` ( $1 \cdot 10^{-12}$ ) [DEFAULT]  
◇ `NMAX_ITER_ORTHO=nio` (5) [DEFAULT]  
◇ `PREC_LANCZOS=pl` ( $1 \cdot 10^{-8}$ ) [DEFAULT]  
◇ `NMAX_ITER_QR=nim` (30) [DEFAULT]

Les deux premiers paramètres permettent, respectivement, d'ajuster la précision d'orthogonalisation et le nombre de réorthogonalisations dans la méthode de Lanczos pour obtenir des vecteurs indépendants engendrant le sous-espace (cf. [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 dernier fixe le nombre d'itérations maximum pour la résolution du système réduit pour la méthode QR ([R5.01.01] annexe 1).

#### 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.

### 3.4.3 Opérandes de la méthode de Bathe & Wilson (si `METHODE='JACOBI'`)

◇	<code>PREC_BATHE =pbat</code>	(1.10 <sup>-10</sup> )	[DEFAULT]
◇	<code>NMAX_ITER_BATHE=nbat</code>	(40)	[DEFAULT]
◇	<code>PREC_JACOBI=pjaco</code>	(1.10 <sup>-2</sup> )	[DEFAULT]
◇	<code>NMAX_ITER_JACOBI=njaco</code>	(12)	[DEFAULT]

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

### 3.4.4 Opérandes de la méthode QZ (si `METHODE='QZ'`)

```
◇ TYPE_QZ = /'QZ_SIMPLE' [DEFAULT]
           /'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 Mot-clé `CALC_FREQ`

```
◇ CALC_FREQ=_F(...
```

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

### 3.5.1 Opérande `FREQ`

```
◆ FREQ=l_f
◇ TABLE_FREQ=table_f
```

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

<code>OPTION='BANDE'</code>	On attend deux valeurs ( $f_{min}, f_{max}$ ) qui définissent la bande de recherche
<code>OPTION='CENTRE'</code>	On attend une seule valeur de fréquence

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

Avec l'option '`BANDE`', on commence par opérer un coûteux<sup>2</sup> 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

<sup>2</sup> En moyenne 20/30% du coût calcul total de l'opérateur.

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

$$\text{INFO\_MODE+FREQ} = (f_0, f_1, f_2, f_3, f_4, f_5) ,$$

on peut en mutualiser une partie du coût calcul en posant dans `MODE_ITER_SIMULT+'BANDE'`, `FREQ=(f_min=f_1, f_max=f_4)`. L'étape de prétraitement de `MODE_ITER_SIMULT` 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:

$$[f_1, f_2] \cup [f_2, f_3] \cup [f_3, f_4] .$$

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`. Celle-ci peut devenir la règle avant tout calcul de modes.

## Remarques à propos du chaînage `INFO_MODE+MODE_ITER_SIMULT+'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`<sup>3</sup>).
- La sélection des lignes de la table s'effectue par rapport aux valeurs initiales des fréquences. Mais si elles ont subies 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 `MODE_ITER_SIMULT`. Cette stratégie préserve ainsi, à la fois, l'ergonomie de l'option et la consistance des comportements logicielles: un `MODE_ITER_SIMULT+'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, faute de quoi 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 !

## 3.5.2 Opérande `AMOR_REDUIT`

◇ `AMOR_REDUIT=1_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

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

## 3.5.3 Opérande `NMAX_FREQ`

◇ `NMAX_FREQ=nf (10 si OPTION='PLUS_PETITE', 1 si OPTION='PLUS_GRADE')`  
`[DEFAULT]`

Nombre maximum de valeurs propres à calculer.

<sup>3</sup> Le critère de tri est en relatif, sauf lorsque la borne recherchée est proche de zéro. Il devient alors un critère absolu.

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

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

## 3.6 Mot-clé `CALC_CHAR_CRIT`

◇ `CALC_CHAR_CRIT=_F(...`

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

### 3.6.1 Opérande `CHAR_CRIT`

◆ `CHAR_CRIT=l_c`  
◇ `TABLE_CHAR_CRIT=table_c`

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

<code>OPTION='BANDE'</code>	On attend deux valeurs $(\lambda_{min}, \lambda_{max})$ qui définissent la bande de recherche
<code>OPTION='CENTRE'</code>	On attend une seule valeur de charge critique

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

Avec l'option 'BANDE', on commence par opérer un coûteux<sup>4</sup> 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=`  $(\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$  ,

on peut en mutualiser une partie du coût calcul en posant dans `MODE_ITER_SIMULT+'BANDE'`, `CHAR_CRIT=`  $(\lambda_{min}=\lambda_1, \lambda_{max}=\lambda_4)$  . L'étape de prétraitement de `MODE_ITER_SIMULT` 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:

$[\lambda_1, \lambda_2] \cup [\lambda_2, \lambda_3] \cup [\lambda_3, \lambda_4]$  .

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`. Celle-ci peut devenir la règle avant tout calcul de modes.

#### Remarques à propos du chaînage `INFO_MODE+MODE_ITER_SIMULT+'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<sup>5</sup>).
- La sélection des lignes de la table s'effectue par rapport aux valeurs initiales des fréquences. Mais si elles ont subies 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

4 En moyenne 20/30% du coût calcul total de l'opérateur.

5 Le critère de tri est en relatif, sauf lorsque la borne recherchée est proche de zéro. Il devient alors un critère absolu.

dernières valeurs décalées qui sont transmises à l'algorithmie de `MODE_ITER_SIMULT`. Cette stratégie préserve ainsi, à la fois, l'ergonomie de l'option et la consistance des comportements logicielles: un `MODE_ITER_SIMULT+'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, faute de quoi 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 !

## 3.6.2 Opérande `NMAX_CHAR_CRIT`

◇ `NMAX_CHAR_CRIT=nf` (10) [DEFAULT]

Nombre maximum de charges critiques à calculer.

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

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

## 3.7 Opérande `APPROCHE`

◇ `APPROCHE= /'REEL'` [DEFAULT]  
/ '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 libres d'une structure amortie ou tournantes (modes propres complexes; le mot-clé `MATR_C` doit être renseigné). En flambement, (`TYPE_RESU='MODE_FLAMB'`) cela n'a aucun intérêt.

### Remarque:

- 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.

## 3.8 Opérande `OPTION`

◇ `OPTION=`  
`'BANDE'`

On recherche toutes les valeurs propres dans une bande donnée. Cette bande est définie par l'argument de `FREQ=(fmin, fmax)` ou par celui de `CHAR_CRIT=(λmin, λmax)`. Option uniquement disponible en GEP à matrices réelles symétriques.

`'CENTRE'`

On recherche les `NMAX_FREQ` valeurs propres les plus proches de la fréquence  $f$  (argument du mot-clé `FREQ=f`) ou les plus proches de la charge critique  $\lambda$  (argument du mot-clé `CHAR_CRIT=λ`).

<code>'PLUS_PETITE'</code> [DEFAULT]	On recherche les <code>NMAX_FREQ</code> plus petites valeurs propres ou les <code>NMAX_CHAR_CRIT</code> les plus petites charges critiques.
<code>'PLUS_GRADE'</code>	On recherche les <code>NMAX_FREQ</code> plus grandes valeurs propres. Option utilisable seulement dans le cas <code>TYPE_RESU='DYNAMIQUE'</code> , 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 <code>VERI_MODE</code> . 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.
<code>'TOUT'</code>	On cherche tous les modes associés à des degrés de liberté physiques. Option utilisable seulement avec la méthode QZ.

Voir [R5.01.01/02] §2.5/3.8.

### 3.9 Opérandes `SEUIL_FREQ` / `SEUIL_CHAR_CRIT`, `PREC_SHIFT`, `NMAX_ITER_SHIFT`

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

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

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

$$Q(\lambda) := A - \lambda B \quad (\text{GEP})$$

$$Q(\lambda) := \lambda^2 B + \lambda C + A \quad (\text{QEP})$$

Ces factorisations sont tributaires d'instabilités numériques lorsque le shift  $\lambda$  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`<sup>6</sup>, 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  $\lambda_1, \lambda_2$  sont considérés comme conjugués si

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

<sup>6</sup> Valeur fixée via le paramètre `NPREC` du mot-clé `SOLVEUR` (par défaut `ndeci=8`).

## 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  $100 \times \text{omecor}$  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  $\lambda_{min}$  en flambement) ou  $f_{max}$  (resp.  $\lambda_{max}$ ) sont alors décalées vers l'extérieur du segment de `p_shift%`. Si la matrice dynamique ainsi reconstruite est toujours jugée "numériquement singulière", on re-décale à nouveau après avoir émis une `ALARME`. On tente ce décalage `n_shift` fois.

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

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,  $2 \times p_{shift}\%$  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.

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 dès le départ, la borne jaugée est inférieure à ce "zéro"  $|\lambda_*| < \text{omecor}$  (en valeur absolue) on la fixe à plus ou moins cette valeur (suivant que cette borne soit positive ou négative). On ne permet alors plus aucun décalage.

## Remarques:

- Une borne de l'intervalle  $\sigma$  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`.

## 3.10 Opérande `DIM_SOUS_ESPACE`

- ◇ `DIM_SOUS_ESPACE=des`
- ◇ `COEF_DIM_ESPACE=mse`

```
EXCLUS ('DIM_SOUS_ESPACE', 'COEF_DIM_ESPACE')
```

Si le mot-clé `DIM_SOUS_ESPACE` n'est pas renseigné ou est initialisé à une valeur strictement inférieure au nombre de fréquences demandées `nf`, l'opérateur calcule automatiquement une dimension admissible pour le sous-espace de projection (cf. §5.2 de ce document et [R5.01.01] §5.3) à l'aide `COEF_DIM_ESPACE`.

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. Dans l'encapsulation de `MODE_ITER_SIMULT`, `CALC_MODES` [U4.52.02], on peut donc optimiser la taille des sous-espaces qui reste proportionnelle au nombre de fréquences recherchées: les sous-espaces riches en valeurs propres ne pénalisent ainsi pas les plus pauvres (en terme de CPU).

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

Dans les deux cas, si la taille du sous-espace de projection `ndim` est strictement supérieure au nombre de «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'.

## 3.11 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 le comptage des valeurs propres (cf [R5.01.01] §3.7.4 et [R5.01.02] §2.5.4).

### 3.11.1 Opérande `STOP_ERREUR`

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

Permet d'indiquer à l'opérateur s'il doit s'arrêter ('OUI') ou continuer ('NON') dans le cas où l'un des critères `SEUIL` ou `STURM` (uniquement avec `MODE_ITER_SIMULT`) n'est pas vérifié.

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

### 3.11.2 Opérande `SEUIL`

◇ `SEUIL = r (1.10-6) [DEFAULT]`

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`.

### 3.11.3 Opérande `STURM`

◇ `STURM = / 'OUI' [DEFAULT]`  
`/ 'NON'`

Vérification dite de STURM ('OUI') permettant de s'assurer que l'algorithme utilisé dans l'opérateur a déterminé le nombre exact de valeurs propres dans l'intervalle de recherche ([R5.01.01] §3.5 et 3.7.4). Cette option n'a d'intérêt qu'en GEP à modes réels (donc pas avec **K** complexe et avec des matrices non symétriques). Voir aussi paramètre `STOP_ERREUR`.

## 3.11.4 Opérande `PREC_SHIFT`

◇ `PREC_SHIFT=prs` (0.05) [DEFAULT]

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

## 3.12 Opérande `STOP_BANDE_VIDE`

◇ `STOP_BANDE_VIDE=/'OUI'` [DEFAULT]  
/ 'NON'

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

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

Ce mot-clé est utilisé dans la macro-commande `CALC_MODES` [U4.52.02] afin de permettre l'absence de valeurs propres dans une bande de recherche. Cette option n'a pas d'intérêt avec la méthode QZ.

## 3.13 Opérande `INFO`

◇ `INFO=/1` [DEFAULT]  
/2

Indique le niveau d'impression dans le fichier `MESSAGE`.

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

## 3.14 Opérande `TITRE`

◇ `TITRE=ti`  
Titre attaché au concept produit par cet opérateur [U4.03.01].

## 4 Phase de vérification

---

On vérifie selon les options:

`OPTION='BANDE'`

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

`OPTION='CENTRE'`

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

`OPTION='PLUS_PETITE'`

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

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

## 5 Phase d'exécution

---

### 5.1 Vérification

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

### 5.2 Actions par défaut

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

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

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

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

où `nactif` est le nombre de 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`.

## 6 Paramètres modaux/ Norme des modes/ Position modale

---

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

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

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

Dans le cas d'un calcul dynamique 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 dans ASTK (onglet 'Options'). 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 `MODE_ITER_SIMULT` pour chaque sous-bande.

Pour une meilleure ergonomie, ces appels à `MODE_ITER_SIMULT` pour chaque sous-bande peuvent être réalisés de manière automatique et transparente pour l'utilisateur grâce à la macro-commande `CALC_MODES` [U4.52.02] avec l'option '`BANDE`' découpée en sous-bandes. Cette macro-commande 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.

## 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' ) );
```

## 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.

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

La macro-commande `CALC_MODES[U4.52.02]` permet d'enchaîner les commandes `MODE_ITER_SIMULT`, `NORM_MODE` et `EXTR_MODE`.

## 10 Exemples

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

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

### 10.2 Calcul des charges critiques contenues dans une bande

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

### 10.3 Chaînage `INFO_MODE+MODE_ITER_SIMULT` (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=MODE_ITER_SIMULT(...  
          CALC_CHAR_CRIT=_F(OPTION='BANDE',TABLE_CHAR_CRIT=nbmod1,  
                           CHAR_CRIT=(-1.E+06,1.E+05)),  
          TYPE_RESU='MODE_FLAMB'  
          )
```