Opérateur INFO_MODE

1 But

Cette commande permet de **compter** et de sauvegarder (dans une sd_table) **le nombre de valeurs propres comprises** dans:

- Un **intervalle de l'axe réel** (pour les problèmes généralisés GEPs standards à matrices réelles symétriques).
- Un **disque du plan complexe** (pour tous les autres cas de figure, autres GEPs et tous les problèmes quadratiques QEPs).

Lorsqu'on traite des GEPs standards, l'exécution de cette procédure ne nécessite généralement qu'une factorisation LDLT par mode. Lorsqu'elles sont disséminées dans le plan complexe, la procédure est beaucoup plus coûteuse. Dans ce cas de figure, elle est donc plutôt à réserver aux problèmes simplifiés de petites tailles (inférieurs à 1000 degrés de liberté).

L'appel à cet opérateur est préconisé comme **vérification/calibration** *a priori* **du modèle.** Il permet aussi de définir des **intervalles de recherche équilibrés et contenant un nombre raisonnable de valeurs propres** (typiquement quelques dizaines). Et ce, afin d'optimiser les coûts des opérateurs modaux ultérieurs et de ne pas dégrader la précision de leurs résultats.

Pour traiter efficacement un GEP standard[1], on propose de procéder en plusieurs étapes:

Calibrer les zones d'intérêt par des pré-calculs utilisant uniquement INFO_MODE (en séquentiel ou, si possible, en parallèle) sur des listes de fréquences (resp. charges critiques) données;

Relancer un ou plusieurs calculs MACRO_MODE_MECA ou MODE_ITER_SIMULT+OPTION='BANDE' en se basant sur ces calibrations initiales.

Pour le calcul intensif, l'opérateur INFO_MODE[2], peut bénéficier de deux niveaux de parallélisme. Les gains en temps peuvent atteindre un facteur 70 sur une centaine de processeurs, et les gains en pic mémoire un facteur 2. La mise en œuvre du parallélisme est décrite dans le paragraphe dédié de la documentation [U2.06.01].

Cet opérateur produit un concept sd_table.
Dans une première approche on peut se contenter de renseigner les paramètres: MATR_* , TYPE_MODE et FREQ (ou CHAR_CRIT ou *_CONTOUR).
Cet opérateur est complémentaire des opérateurs MACRO_MODE_MECA[U4.52.02], MODE_ITER_SIMULT [U4.52.01] et MODE_ITER_INV [U4.52.04].

1 Actuellement, la très grande majorité des cas de figures.
2 Tout comme MACRO_MODE_MECA.

Table des Matières

2 Syntaxe

```
tb [table_sdaster] = INFO_MODE

# Type de problème
              ◊  TYPE_MODE=    /   'DYNAMIQUE',          [DEFAUT]
                               /   'MODE_FLAMB',
                               /   'MODE_COMPLEXE',
                               /   'GENERAL'.


# Si TYPE_MODE='DYNAMIQUE'

# Caractéristiques du calcul
              ♦  MATR_RIGI=    A  ,                /   [matr_asse_depl_r]
                                                   /   [matr_asse_depl_c]
                                                   /   [matr_asse_temp_r]
                                                   /   [matr_asse_pres_r]
                                                   /   [matr_asse_gene_r]
                                                   /   [matr_asse_gene_c]
              ♦  MATR_MASS=    B  ,                /   [matr_asse_depl_r]
                                                   /   [matr_asse_temp_r]
                                                   /   [matr_asse_pres_r]
                                                   /   [matr_asse_gene_r]
# Le nombre de fréquences délimitant les intervalles de calcul est noté nb_freq
              ♦  FREQ      =   l_f                     [l_R]

# Paramètres de la méthode de comptage de STURM
              ◊  COMPTAGE=_F(
                 ◊  METHODE       =   /   'AUTO',       [DEFAUT]
                                      /   'STURM'.
                 ◊  SEUIL_FREQ    =   /   f_seuil       [R]
                                       /   0.01         [DEFAUT]
                 ◊  PREC_SHIFT    =   /   p_shift       [R]
                                      /    0.05         [DEFAUT]
                 ◊  NMAX_ITER_SHIFT=/   n_shift         [I]
                                      /   3             [DEFAUT]
                           )


# Si
TYPE_MODE='MODE_FLAMB'

# Caractéristiques du calcul
              ♦  MATR_RIGI=    A  ,                /   [matr_asse_depl_r]
                                                   /   [matr_asse_depl_c]
                                                   /   [matr_asse_temp_r]
                                                   /   [matr_asse_pres_r]
                                                   /   [matr_asse_gene_r]
                                                   /   [matr_asse_gene_c]
              ♦  MATR_RIGI_GEOM=  B  ,             /   [matr_asse_depl_r]
                                                   /   [matr_asse_temp_r]
                                                   /   [matr_asse_pres_r]
                                                   /   [matr_asse_gene_r]
# Le nombre de charges critiques délimitant les intervalles de calcul est noté nb_mode_flamb
              ♦  CHAR_CRIT=l_c                         [l_R]

# Paramètres de la méthode de comptage de STURM
```

```
◊  COMPTAGE=_F(
   ◊  METHODE          =  /  'AUTO',    [DEFAUT]
                          /  'STURM'.
   ◊  SEUIL_CHAR_CRIT  =  /  c_seuil    [R]
                          /   0.01      [DEFAUT]
   ◊  PREC_SHIFT       =  /  p_shift    [R]
                          /   0.05      [DEFAUT]
   ◊  NMAX_ITER_SHIFT  =  /  n_shift    [I]
                          /  3          [DEFAUT]
                )
```

Si TYPE_MODE='MODE_COMPLEXE'

Caractéristiques du calcul

```
♦  MATR_RIGI=   A  ,                 /  [matr_asse_depl_r]
                                     /  [matr_asse_depl_c]
                                     /  [matr_asse_temp_r]
                                     /  [matr_asse_pres_r]
                                     /  [matr_asse_gene_r]
                                     /  [matr_asse_gene_c]
♦  MATR_MASS=   B  ,                 /  [matr_asse_depl_r]
                                     /  [matr_asse_temp_r]
                                     /  [matr_asse_pres_r]
                                     /  [matr_asse_gene_r]
◊  MATR_AMOR=   C  ,(uniquement en QEP)/  [matr_asse_depl_r]
                                     /  [matr_asse_temp_r]
                                     /  [matr_asse_pres_r]
                                     /  [matr_asse_gene_r]
◊  TYPE_CONTOUR        =     'CERCLE'  [DEFAUT]
♦  RAYON_CONTOUR       =     rayon     [R]
◊  CENTRE_CONTOUR      =  /  centre    [C]
                          /  0.0+0.0j  [DEFAUT]
```

Paramètres de la méthode de comptage APM

```
◊  COMPTAGE=_F(
   ◊  METHODE          =  /  'AUTO',    [DEFAUT]
                          /  'APM'.
   ◊  NBPOINT_CONTOUR  =  /  nb_point   [I]
                          /   40        [DEFAUT ]
   ◊  NMAX_ITER_CONTOUR=  /  n_contour  [I]
                          /  3          [DEFAUT]
            )
```

Si TYPE_MODE='GENERAL'

Caractéristiques du calcul

```
♦  MATR_A=   A  ,                    /  [matr_asse_depl_r]
                                     /  [matr_asse_depl_c]
                                     /  [matr_asse_temp_r]
                                     /  [matr_asse_pres_r]
                                     /  [matr_asse_gene_r]
                                     /  [matr_asse_gene_c]
♦  MATR_B=   B  ,                    /  [matr_asse_depl_r]
                                     /  [matr_asse_temp_r]
                                     /  [matr_asse_pres_r]
                                     /  [matr_asse_gene_r]
◊  MATR_C=   C  ,  (uniquement en QEP)/  [matr_asse_depl_r]
```

```
                                                    /   [matr_asse_temp_r]
                                                    /   [matr_asse_pres_r]
                                                    /   [matr_asse_gene_r]
              ◊  CHAR_CRIT =l_c                         [l_R]

# Si CHAR_CRIT==None
              ◊  TYPE_CONTOUR          =      'CERCLE'  [DEFAUT]
              ♦  RAYON_CONTOUR         =      rayon     [R]
              ◊  CENTRE_CONTOUR        =  /   centre    [C]
                                          /    0.0+0.0j [DEFAUT]
              ◊  COMPTAGE=_F(
                 ◊  METHODE            =  /   'AUTO',   [DEFAUT]
                                          /   'STURM',   (si CHAR_CRIT )
                                          /   'APM'.     (si *_CONTOUR)

# Paramètres de la méthode de comptage
# Si STURM
                 ◊  SEUIL_CHAR_CRIT    =  /   c_seuil   [R]
                                           /   0.01      [DEFAUT]
                 ◊  PREC_SHIFT         =  /   p_shift   [R]
                                          /    0.05      [DEFAUT]
                 ◊  NMAX_ITER_SHIFT    =  /   n_shift   [I]
                                          /   3         [DEFAUT]
# Si APM
                 ◊  NBPOINT_CONTOUR    =  /   nb_point  [I]
                                           /   40        [DEFAUT ]
                 ◊  NMAX_ITER_CONTOUR=    /   n_contour [I]
                                          /   3         [DEFAUT]
                       )
```

 # Solveur linéaire et parallélisme

```
              ◊  SOLVEUR=_F(Pour plus de détails voir le document [U4.50.01]).
# En parallèle, on conseille particulièrement le paramétrage METHODE='MUMPS'+RENUM='QAMD' .

              ◊  NIVEAU_PARALLELISME =  /   'COMPLET',[DEFAUT]
                                        /   'PARTIEL'.
# Activé uniquement en mode parallèle ( nb_proc >1) et licite si METHODE='STURM' .
# L 'option ' COMPLET ' fonctionne quelque soit le solveur linéaire direct, si nb_proc = nb_freq
-1 (resp. nb_mode_flamb -1). Avec l'option 'PARTIEL , seul '
SOLVEUR=_F(METHODE='MUMPS') est licite.
```

 # Divers

```
              ◊  INFO=      /   1                     [DEFAUT]
                            /   2
              ◊  TITRE=     tit                       [Kn]

              )  ;
```

3 Opérandes

3.1 Principes

Cette commande permet de compter le nombre de valeurs propres comprises dans (cf. figure 3ab):

Un intervalle de l'axe réel (pour les problèmes généralisés GEPs standards à matrices réelles symétriques): mot-clé TYPE_MODE='DYNAMIQUE' ou 'MODE_FLAMB' ou 'GENERAL'.

Un disque du plan complexe (pour tous les autres cas de figure, autres GEPs et tous les problèmes quadratiques QEPs): mot-clé TYPE_MODE='MODE_COMPLEXE' ou 'GENERAL'.

Ce nombre de modes propres est tracé dans le fichier message pour chaque intervalle de l'axe réel ou pour la portion du plan complexe choisie (cf. figure 1). On rappelle aussi les caractéristiques de ces zones d'intérêt.

```
-----------------------------------------------------------------------
   VERIFICATION DU SPECTRE DE FREQUENCES (METHODE DE STURM)

 LE NOMBRE DE FREQUENCES DANS LA BANDE N   1 DE
BORNES ( 0.000E+00, 1.000E+01) EST   20
-----------------------------------------------------------------------

 LE NOMBRE DE FREQUENCES DANS LA BANDE N   2 DE
BORNES ( 1.000E+01, 2.000E+01) EST   43
…..

 LE NOMBRE DE FREQUENCES DANS LA BANDE N   6 DE
BORNES ( 5.000E+01, 6.000E+01) EST  139
-----------------------------------------------------------------------
```

Figure 1._ Affichages dans le fichier message d'un INFO_MODE sur une série de fréquences FREQ=(0.,10., 20., ...,50., 60.) pour un problème de DYNAMIQUE.

Tous ces éléments sont stockés dans une sd_table (cf. figure 2ab) que l'on peut imprimer (*via* un IMPR_TABLE) ou réutiliser (par des chaînages avec MODE_ITER_SIMULT). Le surcoût de cet opérateur peut alors devenir presque négligeable puisqu'on mutualise l'information globale dégagée avec d'autres opérateurs modaux.

```
#TABLE_SDASTER
 FREQ_MIN     FREQ_MAX     BORNE_MIN_EFFECT BORNE_MAX_EFFECT NB_MODE
  0.00000E+00  1.00000E-02      0.00000E+00      1.00000E-02           3
  1.00000E-02  5.00000E-02      1.00000E-02      5.00000E-02           3
  5.00000E-02  5.50000E-02      5.00000E-02      5.50000E-02           1
  5.50000E-02  6.00000E-02      5.50000E-02      6.00000E-02           0
  6.00000E-02  1.00000E-01      6.00000E-02      1.00000E-01           3
```

```
#TABLE_SDASTER
 CHAR_CRIT_MIN CHAR_CRIT_MAX BORNE_MIN_EFFECT BORNE_MAX_EFFECT NB_MODE
  -1.00000E+06  -5.00000E+05     -1.00000E+06   -5.00000E+05           1
  -5.00000E+05   0.00000E+00     -5.00000E+05    0.00000E+00           3
   0.00000E+00   1.00000E+05      0.00000E+00    1.00000E+05           0
   1.00000E+05   1.10000E+06      1.00000E+05    1.10000E+06           5
```

Figure 2ab._ Exemple n°1 de table générée par un INFO_MODE sur une série de fréquences FREQ=(0.,1.E-2,...,1.E-1) pour un problème de dynamique. Extrait de sdlx201a.mess. Exemple n°2 de table générée par un INFO_MODE sur une série de charges critiques CHAR_CRIT=(-1.E+6,-5.E+5,...,1.1E+6) pour un problème de flambement. Extrait de sdls504a.mess

L'appel à cet opérateur est préconisé comme vérification/calibration *a priori* du modèle. Il permet aussi de définir des intervalles de recherche contenant un nombre raisonnable de valeurs propres (typiquement une quarantaine de modes) afin d'optimiser les coûts des opérateurs modaux ultérieurs et de ne pas dégrader la précision de leurs résultats.

Typiquement, on conseille de rechercher les modes par paquets de quarante environ[3]. Au delà, les consommations en temps et en mémoire ne sont plus optimales et la qualité des modes obtenus se dégrade. Il vaut alors mieux utiliser plusieurs bandes fréquentielles (ou de charges critiques) directement en plusieurs appels à des MODE_ITER_SIMULT[U4.52.03], ou, indirectement, *via* la macro MACRO_MODE_MECA[U4.52.02].

Idéalement, pour traiter efficacement un GEP standard, on propose de procéder en plusieurs étapes:

Calibrer les zones d'intérêt par des pré-calculs utilisant uniquement INFO_MODE (en séquentiel ou, si possible, en parallèle) sur des listes de fréquences (resp. charges critiques) données;

Relancer un ou plusieurs calculs MACRO_MODE_MECA ou MODE_ITER_SIMULT+OPTION='BANDE' en se basant sur ces calibrations initiales.

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. Ce chaînage peut ainsi rendre le surcoût d'INFO_MODE négligeable et guider efficacement le calcul modal.
Contrairement à MODE_ITER_SIMULT, MACRO_MODE_MECA ne peut **réutiliser** la sd_table générée à l'étape n°1. Par contre, en séquentiel et surtout en parallèle, il permet des gains notables en temps, en pic mémoire et en précision lorsqu'on traite de gros calculs (en taille de maillage et/ou en nombre de modes recherchés).

Coté calcul intensif, l'opérateur INFO_MODE, tout comme MACRO_MODE_MECA, bénéficie potentiellement de deux niveaux de parallélisme (cf. mot-clé NIVEAU_PARALLELISME). D'où des gains en temps d'un facteur jusqu'à 70 sur une centaine de processeurs et des gains en pic mémoire jusqu'à un facteur 2.

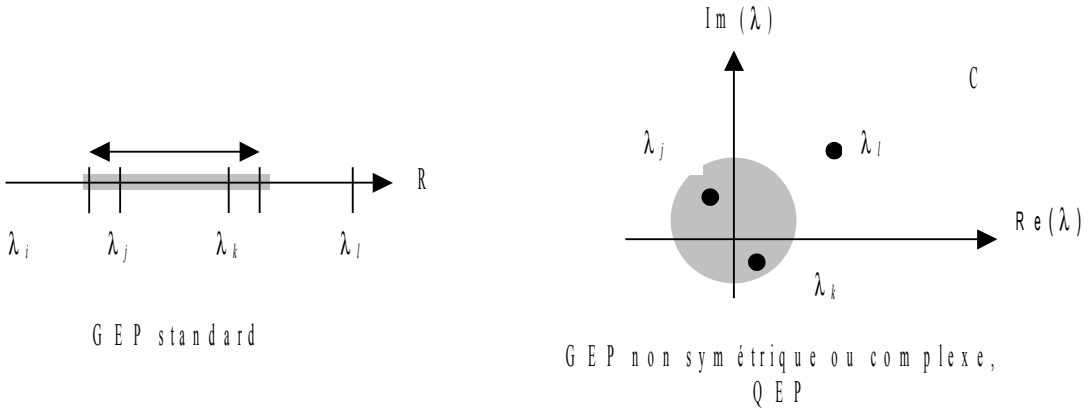


Figure 3ab._ Deux problématiques de dénombrement distinctes: dans un segment de l'axe réel et dans une portion finie du plan complexe (pour l'instant qu'un disque).

Dans le premier cas, lorsqu'on cherche à dénombrer les valeurs propres strictement incluses dans un segment de l'axe réel, on utilise la méthode dite de «Sturm», dans le second cas la méthode APM. Leurs théories et algorithmies sont détaillées dans la documentation de référence [R5.01.04].

Lorsqu'on traite des GEPs standards, l'exécution de cette procédure ne nécessite généralement[4] qu'une factorisation LDLT par mode (METHODE='STURM'). Lorsqu'elles sont disséminées dans le plan complexe, la procédure est beaucoup plus coûteuse (METHODE='APM') car elle requiert, en général,

3 En mode séquentiel. Avec MACRO_MODE_MECA parallèle, on peut obtenir un calcul plus rapidement en ne prenant qu'une vingtaine de modes.

4 Sauf si la valeur est proche d'un mode propre, auquel cas on la «décale» (cf. mot-clé NMAX_ITER_SHIFT) pour limiter certaines instabilités numériques.

des centaines de factorisations. Dans ce cas de figure, elle est donc plutôt à réserver aux problèmes simplifiés de petites tailles (inférieurs à 1000 degrés de liberté).

Notons que cette dernière méthode APM est encore l'objet de recherches actives. Sa robustesse n'est donc, pour l'instant, pas garantie. Elle est à utiliser en ayant bien pris soin de lire la documentation associée.

3.2 Caractéristiques du problème modal: 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_MODE.

TYPE_MODE			
'DYNAMIQUE'	'MODE_FLAMB'	'MODE_COMPLEXE'	'GENERAL'
♦ MATR_RIGI = A	♦ MATR_RIGI = A	♦ MATR_RIGI = A	♦ MATR_A = A
♦ MATR_MASS = B	♦ MATR_RIGI_GEOM = B	♦ MATR_MASS = B	♦ MATR_B = B
Sans Objet	Sans Objet	◊ MATR_AMOR = C	◊ MATR_C = C

Ces opérandes permettent de renseigner les matrices (assemblées ou généralisées[5]) caractérisant le problème modal. Ces matrices peuvent être symétriques ou non. Elles sont réelles sauf la matrice $\mathbf{A}$ qui peut être, soit réelle, soit complexe.

La donnée de $\mathbf{A}$ et $\mathbf{B}$ permet de définir le problème modal généralisé (GEP) (cf. [R5.01.01]) étudié

$$\mathbf{A}\,\mathbf{u}=\lambda\,\mathbf{B}\,\mathbf{u} \quad (\mathrm{GEP}_{\mathrm{dynamique}})$$

Dans le cas classique de la dynamique (TYPE_MODE='DYNAMIQUE'), $\mathbf{A}$ est la matrice de rigidité (symétrique réelle) et $\mathbf{B}$ et la matrice de masse (symétrique réelle). La valeur propre λ est alors reliée à la fréquence propre f par la formule: $\lambda=(2\pi f)^2$. Elle est réelle positive.

Dans le cas de la théorie du flambement linéaire (TYPE_MODE='MODE_FLAMB'),

$$\mathbf{A}\,\mathbf{u}-\lambda\,\mathbf{B}\,\mathbf{u}=\mathbf{0} \quad (\mathrm{GEP}_{\mathrm{flambement}})$$

$\mathbf{A}$ est la matrice de rigidité (symétrique réelle) et $\mathbf{B}$ la matrice de rigidité géométrique (symétrique réelle). La valeur propre λ est appelée charge critique. Elle est réelle.

Comme mentionné dans le tableau précédent, ces matrices seront renseignées, respectivement, pour la dynamique par MATR_RIGI/MATR_MASS et, en flambement, par MATR_RIGI/ MATR_RIGI_GEOM. Seule la méthode dite de 'STURM' est licite dans ces deux cas de figure.

Lorsqu'une des deux matrices n'est plus symétrique ou comporte des termes complexes (par ex. pour prendre en compte de l'amortissement hystérétique), les valeurs propres sont potentiellement complexes. Pour traiter ce problème il faut initialiser TYPE_MODE à 'MODE_COMPLEXE' et seule la méthode APM est alors licite.

La donnée de $\mathbf{A}$, $\mathbf{B}$ et $\mathbf{C}$ permet de définir le problème modal quadratique (QEP) (cf. [R5.01.02]) étudié

$$(\mathbf{A}+\lambda\,\mathbf{C}+\lambda^2\,\mathbf{B})\,\mathbf{u}=\mathbf{0} \quad (\mathrm{QEP})$$

5 Cette notion de matrice généralisée n'a rien à voir avec celle de problème modal généralisé !

Dans ce cas de figure, les valeurs propres sont potentiellement complexes. Pour traiter ce problème il faut initialiser TYPE_MODE à 'MODE_COMPLEXE'. Souvent $\mathbf{A}$ est la matrice de rigidité, $\mathbf{C}$ une matrice prenant en compte l'amortissement visqueux et/ou des effets gyroscopiques et $\mathbf{B}$ la matrice de rigidité géométrique. Comme mentionné dans le tableau précédent, ces matrices seront renseignées par MATR_RIGI/MATR_MASS/MATR_AMOR et seule la méthode APM est licite.

Le TYPE_MODE='GENERAL' permet de résoudre un problème de valeurs propres dans le cas générale. On fait alors jouer aux matrices $\mathbf{A}$, $\mathbf{B}$ et/ou $\mathbf{C}$ le rôle que l'on souhaite. Comme mentionné dans le tableau précédent, ces matrices seront renseignées par MATR_A/MATR_B/MATR_C. Suivant les cas de figure, seule la méthode de Sturm (si MATR_C n'est pas renseignée et si les deux autres matrices sont réelles symétriques) ou la méthode APM (tous les autres cas) sont licites.

3.3 Opérande TYPE_MODE

◊ TYPE_MODE= / **'DYNAMIQUE'** **[DEFAUT]**
/ 'MODE_FLAMB'
/ 'MODE_COMPLEXE'
/ 'GENERAL'

Ce mot-clé permet de définir la nature du problème modal à traiter: pré-estimer le spectre de fréquences de vibration (cas classique de 'DYNAMIQUE'), le faire en terme de de charges critiques (cas de la théorie du flambement linéaire 'MODE_FLAMB') ou calibrer celui d'un GEP non standard ou d'un QEP dans une portion du plan complexe ('MODE_COMPLEXE'). On peut aussi rechercher les valeurs propres et les modes associés d'un système matriciel général ('GENERAL'). Pour plus de précisions on pourra se reporter au descriptif du §3.2.

A l'issu de ce choix, l'utilisateur doit spécifier les caractéristiques de sa zone de contrôle:
pour 'DYNAMIQUE': liste de fréquences *via* FREQ,
pour 'MODE_FLAMB': liste de charges critiques *via* CHAR_CRIT,
pour 'MODE_COMPLEXE' : zone du plan complexe *via* TYPE/RAYON/CENTRE_CONTOUR ,
pour 'GENERAL' : suivant le cas de figure, soit on procède comme pour 'MODE_FLAMB' (si MATR_C n'est pas renseignée et si les deux autres matrices sont réelles symétriques) , soit comme pour 'MODE_COMPLEXE' (tous les autres cas).

3.4 Opérande FREQ

♦ FREQ = l_f

Liste de fréquences (en Hertz) définissant les intervalles que l'on veut calibrer $\mathtt{l\_f}=(f_i)_i$ (on note nb_freq le nombre de fréquences de cette liste). On recherche alors le nombre de valeurs propres dans les sous-bandes $[\lambda_i, \lambda_{i+1}]$ avec $\lambda_*=(2\pi f_*)^2$ .Cette liste doit comporter au moins deux valeurs. Ces valeurs doivent être rangées par ordre strictement croissant et toutes positives.
Ce mot-clé doit être utilisé si TYPE_MODE='DYNAMIQUE'. Seule la méthode de comptage de type STURM est alors licite.

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 (éventuellement plusieurs fois cf. paramètres PREC_SHIFT/NMAX_ITER_SHIFT etc). Ce décalage «du shift» est toujours opérée vers l'extérieure de la sous-bande considérée.

Si le décalage de la borne inférieure la rend plus grande que la borne supérieure, le processus s'arrête en ERREUR_FATALE.

Ce mode de décalage rend plus robuste et consistant les réutilisations ultérieures de cette sd_table par un ou plusieurs MODE_ITER_SIMULT. On évite en particulier les recouvrements de sous-bandes qui pouvaient se produire jusqu'à la V10.

Les bornes initiales de chacune de ces sous-bandes sont sauvegardées dans les champs FREQ_MIN/MAX de chaque ligne de la table produite (cf. figure 2a). Pour chacun de ces couples, on mentionne les bornes effectivement utilisées par le test de Sturm (c'est-à-dire après d'éventuels décalages), BORNE_MIN/MAX_EFFECT, ainsi que le nombre de modes trouvés: NB_MODE.

3.5 Opérandes CHAR_CRIT

CHAR_CRIT = l_c

Liste de charges critiques définissant les intervalles que l'on veut calibrer $l\_c=(\lambda_i)_i$ (on note nb_mode_flamb le nombre de charges critiques de cette liste). On recherche alors le nombre de valeurs propres dans les sous-bandes $[\lambda_i, \lambda_{i+1}]$ .Cette liste doit comporter au moins deux valeurs. Ces valeurs doivent être rangées par ordre strictement croissant. Elles doivent être de type réel.
Ce mot-clé doit être utilisé si TYPE_MODE='MODE_FLAMB' ou si TYPE_MODE='GENERAL' (si MATR_C n'est pas renseignée et si les deux autres matrices sont réelles symétriques). Seule la méthode de comptage de type STURM est licite.

Remarques:

Chaque charge critique 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 charge est jugée trop proche d'une valeur propre, on la décale (éventuellement plusieurs fois cf. paramètres PREC_SHIFT/NMAX_ITER_SHIFT etc). Ce décalage « du shift » est toujours opérée vers l'extérieure de la sous-bande considérée.

Si le décalage de la borne inférieure la rend plus grande que la borne supérieure, le processus s'arrête en ERREUR_FATALE.

Ce mode de décalage rend plus robuste et consistant les réutilisations ultérieures de la sd_table par un ou plusieurs MODE_ITER_SIMULT. On évite en particulier les recouvrements de sous-bandes qui pouvaient se produire jusqu'à la V10.

Les bornes initiales de chacune de ces sous-bandes sont sauvegardées dans les champs CHAR_CRIT_MIN/MAX de chaque ligne de la table produite (cf. figure 2b). Pour chacun de ces couples, on mentionne les bornes effectivement utilisées par le test de Sturm (c'est-à-dire après d'éventuels décalages), BORNE_MIN/MAX_EFFECT, ainsi que le nombre de modes trouvés: NB_MODE.

3.6 Opérandes TYPE/RAYON/CENTRE_CONTOUR

Si TYPE_MODE='MODE_COMPLEXE' ou si TYPE_MODE='GENERAL' (si MATR_C est renseignée ou/et si une des matrices est non symétrique ou complexe)

◊	TYPE_CONTOUR =		**'CERCLE'**	**[DEFAUT]**
♦	RAYON_CONTOUR=		rayon	[R]
◊	CENTRE_CONTOUR =	/	centre	[C]
		/	**0.0+0.0j**	**[DEFAUT]**

Ces mots-clés définissent le type de contour à l'intérieur duquel on cherche les valeurs propres ainsi que ses caractéristiques. Pour l'instant on ne peut choisir qu'un contour de type 'CERCLE' (qui délimite donc le disque de contrôle). On mentionne son centre (un nombre complexe) et son rayon (un réel positif).

Remarques:

En toute rigueur, ce rayon doit être supérieur au «zéro modal». C'est-à-dire supérieur à la valeur en dessous de laquelle on considère qu'une valeur propre est nulle ou que deux valeurs propres sont confondues si on parle de leur écart.

Contrairement aux valeurs de FREQ , les unités ne sont pas ici en fréquence, mais en pulsation si on fait l'analogie avec le problème DYNAMIQUE. Comme pour un TYPE_MODE='MODE_FLAMB', l'unité des caractéristiques du contour ne subit aucune transformation. Cela permet à la méthode APM de traiter tous les types de problèmes.

La table produite avec l'option TYPE_MODE='MODE_COMPLEXE' contient, en plus du nombre de valeurs propres (NB_MODE), trois réelles permettant de décrire les critères de recherche pour la méthode APM. Les deux premiers sont CENTRE_R et CENTRE_I , les parties réelle et imaginaire du nombre complexe qui correspond au centre du disque de recherche. Le troisième paramètre, RAYON, est le rayon (réel strictement positif>1.D-2) du disque.

3.7 Mot-clé facteur COMPTAGE

Une fois défini le type de zone de contrôle (segment ou disque), il reste à fixer la méthode de dénombrement et ses paramètres.

```
◊  METHODE = /  'AUTO',      [DEFAUT]
             /  'STURM',
             /  'APM'.
```

Si on traite un problème de type 'DYNAMIQUE' ou 'MODE_FLAMB', seule les méthodes 'STURM' et 'AUTO' sont autorisées.
De même pour 'MODE_COMPLEXE' et 'APM'/'AUTO'.
Dans le cas 'GENERAL', les trois méthodes sont disponibles et la valeur 'AUTO' prend alors tout son sens ! Elle choisit pour l'utilisateur entre STURM et APM suivant les caractéristiques du problème modal:
Soit il s'agit d'un GEP standard (deux matrices symétriques réelles): méthode de STURM .
Dans tous les autres cas (GEP atypique avec matrice non symétrique et/ou complexe ou QEP): méthode APM .

Si on laisse le mode 'AUTO' paramétré par défaut, l'opérateur choisit automatiquement la méthode suivant le type d'étude choisie. On a laissé sciemment cette valeur, même dans les cas qui ne comportent qu'une seule alternative, pour faciliter l'ergonomie.

Si on se trompe de METHODE, une alarme apparaît et l'opérateur choisit automatiquement la méthode la plus appropriée ou s'arrête en ERREUR_FATALE, suivant les cas de figure.

Remarque:

On peut utiliser la méthode APM pour dénombrer les modes d'un GEP standard en posant TYPE_MODE='MODE_COMPLEXE'. Cela permet de comparer et valider les deux méthodes de comptage (cf. cas-test SDLL123a).

```
# Si TYPE_MODE='DYNAMIQUE'

      ◊  SEUIL_FREQ =        /  f_seuil      [R]
                             /  0.01         [DEFAUT]
      ◊  PREC_SHIFT =        /  p_shift      [R]
                             /  0.05         [DEFAUT]
      ◊  NMAX_ITER_SHIFT =   /  n_shift      [I]
                             /  3            [DEFAUT]
```

Si TYPE_MODE='MODE_FLAMB' ou 'GENERAL' (si MATR_C n'est pas renseignée et si les deux autres matrices sont symétriques réelles)

```
      ◊  SEUIL_CHAR_CRIT =   /  c_seuil      [R]
```

```
                              /  0.01          [DEFAUT]
◊  PREC_SHIFT =               /  p_shift       [R]
                              /  0.05          [DEFAUT]
◊  NMAX_ITER_SHIFT =          /  n_shift       [I]
                              /  3             [DEFAUT]
```

Ces paramètres ne sont utilisés que par la méthode de STURM.
Les paramètres SEUIL_* permettent de définir le « zéro modal », c'est-à-dire la valeur en deçà de laquelle on considère que la valeur propre est nulle. Si on est en dynamique on transforme cette valeur en pulsation

$$omecor = (2\pi \, \text{f\_seuil})^2$$

tandis qu'en flambement on la garde telle quelle

$$omecor = \text{c\_seuil}.$$

Les autres paramètres, PREC_SHIFT et NMAX_ITER_SHIFT, sont liés à l'algorithme de décalage des bornes de l'intervalle (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 re-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-censu* , 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.

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 soit 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.

Si TYPE_MODE='MODE_COMPLEXE' ou 'GENERAL' (si MATR_C est renseignée ou/et si une des deux autres matrices est non symétrique ou complexe)

```
◊  NBPOINT_CONTOUR =    /  nb_point      [I]
                        /  40            [DEFAUT]
◊  NMAX_ITER_CONTOUR =  /  n_contour     [I]
                        /  3             [DEFAUT]
```

Ces paramètres ne sont utilisés que par la méthode APM .

Les deux paramètres précédent, NB_POINT_CONTOUR et NMAX_ITER_CONTOUR, ne sont utilisés qu'avec la méthode APM. Ils sont cruciaux pour sa robustesse et ils dimensionnent son temps calcul (un calcul avec nb_point=80 durera plus longtemps qu'un calcul avec nb_point=40).
La valeur nb_point donne le nombre de point de discrétisation qui sont positionnés le long du contour. Cette discrétisation doit être assez fine pour «capturer» toute l'information spectrale. Idéalement, cette valeur doit être fixée à au moins six fois le nombre de modes que l'on escompte trouver dans la cercle. Si on n'a aucune idée *a priori* de ce nombre, il faut le fixer à une valeur pas trop faible: par exemple, 40 ou 60.

Remarque:
Attention, si ce chiffre est trop grand (par exemple 1000), suivant la taille du problème, comme il implique autant de factorisations LDLT, le calcul peut être très long !

L'algorithme APM implanté ne convergera que lorsque 3 évaluations successives du nombre de valeurs propres fourniront le même résultat. La différences entre ces trois évaluations réside uniquement dans le degré de discrétisation du contour. On commence par discrétiser avec $k_- = \text{nb\_point}/2$, puis avec $k = \text{nb\_point}$ et enfin avec $k_+ = 2\,\text{nb\_point}$. Si ces trois niveaux de discrétisation du contour produisent une estimation du nombre de valeurs propres identiques, l'algorithme est considéré comme convergé. Son résultat est alors ce nombre entier.
Sinon, on double ces trois niveaux de discrétisations suivant la permutation suivante

$$k_- \leftarrow k \quad , \quad k \leftarrow k_+ \quad \text{et} \quad k_+ \leftarrow 2\,k_+$$

et on réévalue les nombres de valeurs propres avec ces trois niveaux de discrétisation[6]. Si ils sont identiques, la convergence est atteinte, sinon on continue. On réitère cette heuristique dichotomique au maximum n_contour fois.
Si ce processus n'a pas convergé ou si son résultat est incohérent (entier strictement négatif), on s'arrête en ERREUR_FATALE.

3.8 Opérande INFO

◊ INFO = / 1 **[DEFAUT]**
/ 2

Indique le niveau d'impression dans le fichier MESSAGE.
1: Impression du résultat (et des étapes principales de l'algorithme si APM).
2: Impression plus détaillée plutôt pour développeur.

3.9 Opérande TITRE

Le titre qui sera donné à la table produite.

3.10 Mot-clé facteur SOLVEUR

◊ SOLVEUR=_F(),

On a accès à tous les paramètres des solveurs linéaires directs (METHODE='LDLT'/'MULT_FRONT'/'MUMPS') sauf ceux explicitement liés à l'étape finale de descente-remontée. Cette restriction ne concernent que les deux paramètres suivant du solveur MUMPS: POSTTRAITEMENTS et RESI_RELA.
En parallèle, on conseille particulièrement le paramétrage [7] METHODE='MUMPS' et RENUM='QAMD' .

Pour plus de détails sur les solveurs, on pourra consulter le document [U4.50.01].

6 Bien sûr pour économiser du temps calcul, on ne réévalue que pour la nouvelle discrétisation, la plus fine.
7 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.

3.11 Opérande NIVEAU_PARALLELISME

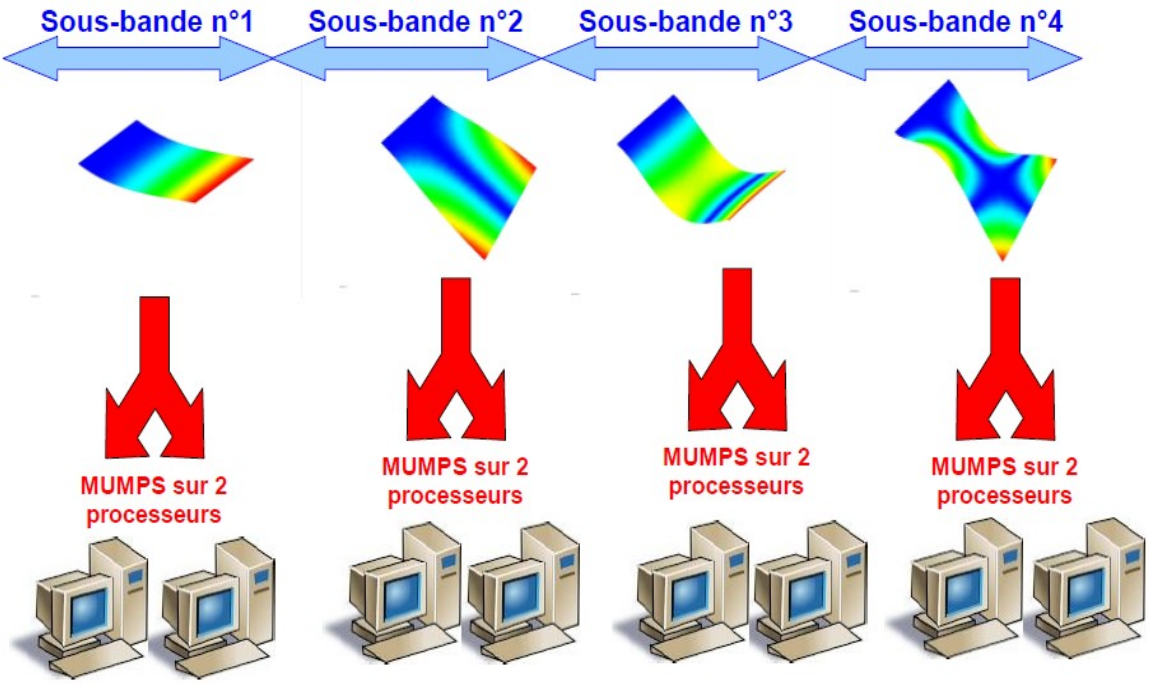
◊ NIVEAU_PARALLELISME = / 'COMPLET' [DEFAUT]
/ 'PARTIEL'

Lorsqu'on traite des problèmes **de tailles moyennes ou grandes** (> 0.5M degrés de liberté) et/ou que l'on cherche une **bonne partie de leurs spectres** (> 50 modes), le recours au parallélisme procure des gains appréciables en temps et en mémoire. Et ce, avec un comportement fonctionnel et des précisions de résultats inchangés par rapport au mode séquentiel.

Le découpage naturel du calcul INFO_MODE en nb_sbande indépendantes (nombre de sous-bandes=nb_freq-1 ou nb_mode_flamb-1), procure un premier niveau de parallélisme très efficace (en temps). Il n'impacte pas les consommations mémoire[8].
D'autre part, comme l'essentiel du coût calcul est dû à la phase de factorisation numérique[9] du solveur linéaire, si on parallélise cette étape *via* MUMPS (METHODE='MUMPS'), on rajoute des gains notables en temps et en mémoire. C'est un deuxième niveau de parallélisme activable.

Avec la valeur par défaut de ce mot-clé (NIVEAU_PARALLELISME='COMPLET'), lorsque le calcul est parallèle, les deux niveaux de parallélisme se cumulent. Comme le second (celui du solveur linéaire) est moins efficace que le premier, il ne se déclenche que si le nombre de processeurs le permet (nb_proc>nb_sbande) et que si le solveur linéaire sélectionné supporte du parallélisme MPI (SOLVEUR=_F(METHODE='MUMPS')).
Si on sélectionne l'autre valeur du mot-clé (NIVEAU_PARALLELISME='PARTIEL'), on ne profite que du parallélisme du solveur linéaire (le second niveau). Cela peut avoir de l'intérêt pour tester des méthodes ou pour axer les gains sur la réduction du pic mémoire (après avoir essayé les autres bras de levier proposés dans le mot-clé SOLVEUR).

Pour plus d'informations sur la mise en œuvre du parallélisme, on se reportera à la documentation [U2.08.06] ou au paragraphe dédié de la documentation [U2.06.01].



8 Ce qui est le cas pour MACRO_MODE_MECA parallèle.
9 Etape du solveur linéaire qui tire le plus de profit du parallélisme.

Figure 3.11-1. Exemple de distribution des calculs d'INFO_MODE sur 8 processeurs avec un découpage en 4 sous-bandes fréquentielles.

Ainsi, en activant seulement le premier niveau de parallélisme, un calcul découpé en nb_sbande et parallélisé sur nb_proc=nb_sbande peut gagner en temps au moins un facteur nb_sbande/2 (sans surcoût mémoire et sans perte de précision).
Si on distribue le calcul sur plus de processeurs (nb_proc>nb_sbande) en utilisant le solveur MUMPS (en ajoute ainsi le 2nd niveau), les gains en temps vont être améliorés d'un facteur proche de 2 pour chaque 2*nb_sbande processeurs supplémentaires[10]. Et ce, avec des gains en mémoire pouvant aller jusqu'à un facteur 2[11].
Afin de préserver l'efficacité des calculs parallèle, on conseille:

•Avec 'COMPLET', de sélectionner un nombre de processeurs qui est un multiple du nombre de sous-bandes. Typiquement 2, 4 ou 8. Cela limite les déséquilibre de charge.
•Avec 'PARTIEL', de réserver au moins 10^5 degrés de liberté par processeur afin d'alimenter suffisamment le solveur linéaire MUMPS.

Les règles fonctionnelles sont les suivantes, en notant nb_proc le nombre de processeurs paramétré (onglet option/mpi_nbcpu d'Astk) et nb_sbande le nombre de sous-bandes non vides:

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

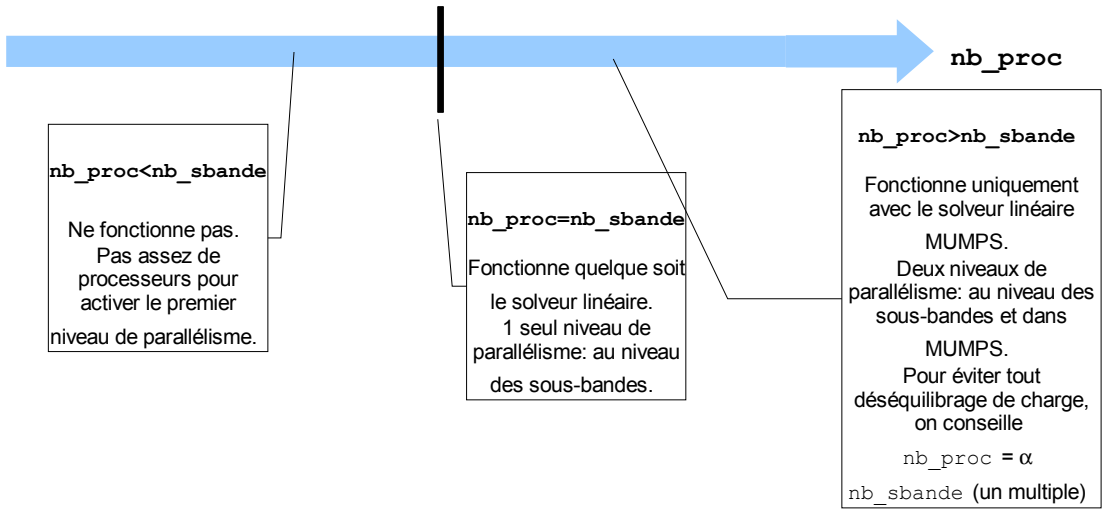
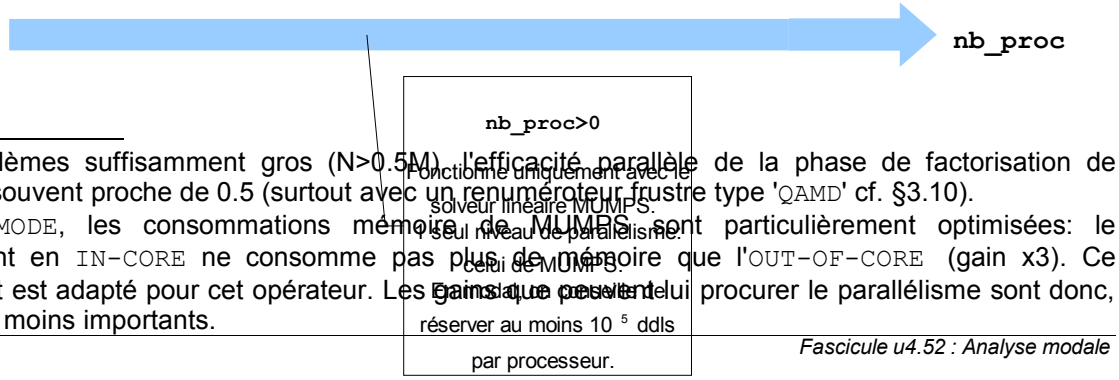


Figure 3.11-2. Périmètre d'utilisation avec NIVEAU_PARALLELISME='COMPLET'.

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



10 Sur des problèmes suffisamment gros (N>0.5M), l'efficacité parallèle de la phase de factorisation de MUMPS est souvent proche de 0.5 (surtout avec un renumeroteur frustre type 'QAMD' cf. §3.10).

11 Dans INFO_MODE, les consommations mémoire de MUMPS sont particulièrement optimisées: le fonctionnement en IN-CORE ne consomme pas plus de mémoire que l'OUT-OF-CORE (gain x3). Ce comportement est adapté pour cet opérateur. Les gains que peuvent lui procurer le parallélisme sont donc, en proportion, moins importants.

Figure 3.11-3. Périmètre d'utilisation avec `NIVEAU_PARALLELISME='PARTIEL'`.

4 Exemples

4.1 Exemple n°1

Comparaison des deux méthodes STURM et APM sur le GEP standard de SDLD02a (pour STURM les bornes sont mentionnées en fréquence). Dans le premier calcul, on chercher à compter le nombre de modes contenu dans la bande fréquentielle $[0;5]$ avec la méthode usuelle: STURM. Dans le second, on fait la même chose avec la méthode APM, mais dans le disque centré à l'origine (centre=0+0j) et de rayon rayon= $(2\pi.5)^2$ (car il n'y a pas de changement d'unité avec APM). Les deux résultats sont affichés dans le fichier MESSAGE.

```
f1=5.0
nbmod01 = INFO_MODE(MATR_RIGI=MATASSR,MATR_MASS=MATASSM,TYPE_MODE='DYNAMIQUE',
                    FREQ=(0.,f1),COMPTAGE=_F(METHODE='STURM'),)
w1=(2*pi*f1)**2
nbmod11 = INFO_MODE(MATR_RIGI=MATASSR,MATR_MASS=MATASSM,TYPE_MODE='MODE_COMPLEXE',
                    TYPE_CONTOUR='CERCLE',CENTRE_CONTOUR=0.0+0.0j,
                    RAYON_CONTOUR=w1,COMPTAGE=_F(METHODE='APM'),)
```

Avec INFO=1, cela provoque les affichages suivant dans le fichier MESSAGE:

```
-----------------------------------------------------------------------
   VERIFICATION DU SPECTRE DE FREQUENCES (METHODE DE STURM)
 PAS DE FREQUENCE DANS LA BANDE ( 0.000E+00, 5.000E+00)
-----------------------------------------------------------------------

….............

(METHODE APM) POUR LES 3 NIVEAUX DE DISCRETISATION SUIVANTS
 ---    20 ---    40 ---    80 ---
 NOMBRE DE VALEURS PROPRES DETECTEES
 ---     0 ---     0 ---     0 ---
(METHODE APM) CONVERGENCE DE L'HEURISTIQUE
-----------------------------------------------------------------------
   VERIFICATION DU SPECTRE EN FREQUENCE (METHODE DE L'ARGUMENT PRINCIPAL)
 PAS DE FREQUENCE DANS LE DISQUE CENTRE EN ( 0.000E+00, 0.000E+00)
 ET DE RAYON  9.870E+02
```

Ici, la méthode de Sturm a requis seulement deux factorisations. La méthode APM a convergé immédiatement à la première itération. Mais celle-ci a nécessité $20+40+80=140$ factorisations. Le dénombrement de valeurs propres dans le plan complexe a un prix (que l'on arrive pas pour l'instant à réduire) !

Le nombre des fréquences propres (0 dans ce cas) ainsi que les critères de recherche sont sauvegardés dans une table. L'impression, par IMPR_TABLE, des concepts NBMOD01 et NBMOD11 produits par INFO_MODE dans l'exemple précédent montre la composition suivante :

```
nbmod01

FREQ_MIN      FREQ_MAX      NB_MODE
0.00000E+00   5.00000E+00     0

...

nbmod11

CENTRE_R      CENTRE_I      RAYON         NB_MODE
0.00000E+00   0.00000E+00   9.86960E+02     0
```

Quand INFO_MODE est appelé avec l'option TYPE_MODE='MODE_FLAMB', les tables produites contiennent trois colonnes: NB_MODE (le nombre de valeurs propres) ainsi que CHAR_CRIT_MIN et CHAR_CRIT_MAX, les critères de recherche pour les problèmes dynamiques avec flambement linéaire.

4.2 Exemple n°2

Dénombrement concernant le QEP de SDLL123a. Cette fois, seule la méthode APM s'avère licite. On compte le nombre de modes contenus dans le cercle centré à l'origine ($centre = 0 + 0\mathrm{j}$) et de rayon $rayon = 124 \times 2\pi$.

```
f1=124.*2.*pi
nbmod4=INFO_MODE(MATR_RIGI=RIGIDITE,MATR_MASS=MASSE,MATR_C=GYOM,TYPE_MODE='MODE_COMPLEXE',
                 TYPE_CONTOUR='CERCLE',CENTRE_CONTOUR=0.0+0.0j,
                 RAYON_CONTOUR=f1, COMPTAGE=_F(METHODE='APM',),)
```

En INFO=1 cela provoque les affichages suivant dans le fichier MESSAGE:

```
(METHODE APM) POUR LES 3 NIVEAUX DE DISCRETISATION SUIVANTS
 ---    20 ---    40 ---    80 ---
 NOMBRE DE VALEURS PROPRES DETECTEES
 ---     4 ---     4 ---     4 ---
(METHODE APM) CONVERGENCE DE L'HEURISTIQUE
-------------------------------------------------------------------------
   VERIFICATION DU SPECTRE EN FREQUENCE (METHODE DE L'ARGUMENT PRINCIPAL)
 LE NOMBRE DE FREQUENCES DANS LE DISQUE CENTRE EN ( 0.000E+00, 0.000E+00)
 ET DE RAYON  7.791E+02 EST    4
```

Et l'impression de la table produite (nbmod4) par IMPR_TABLE donne:

```
CENTRE_R     CENTRE_I     RAYON        NB_MODE
0.00000E+00  0.00000E+00  7.79115E+02     4
```