Notice de mise en œuvre de calculs de rotors

Résumé :

Ce document constitue une aide à l'utilisateur pour la réalisation des études machines tournantes dans *Code_Aster*. Il doit lui permettre de mener à bien un calcul de machines tournantes (analyses statiques, modales, harmoniques et transitoires) cohérent et correct et ce quel que soit l'axe de rotation de la ligne d'arbres.

Ce document a vocation d'être évolutif pour intégrer la mise en œuvre des futures fonctionnalités « machines tournantes » de *Code_Aster*.

La liste des cas-tests de machines tournantes dans *Code_Aster* est mise en annexe.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des Matières

1 Mise en équation....4
1.1 Équations du mouvement....4
2 Modélisation numérique des machines tournantes....5
2.1 Modélisation des parties tournantes....5
2.2 Modélisation des paliers de supportage....6
2.2.1 Paliers linéaires....7
2.2.2 Paliers non linéaires....7
2.3 Modélisation de la table de groupe....8
2.3.1 Appuis infiniment rigides....8
2.3.2 Supports simplifiés....8
2.3.3 Supports tridimensionnels....8
2.3.4 Supports généralisés....8
2.3.5 Supports de type impédance....8
3 Mise en données des études « machines tournantes »....9
3.1 Affectation des modèles....9
3.2 Affectation des caractéristiques élémentaires....9
3.3 Axe de rotation....10
3.3.1 Passage Repère global / Repère local....10
3.3.2 Convention de signe....10
3.3.3 De la nécessité d'imposer un axe de rotation....11
3.4 Conditions aux limites....11
3.4.1 Exemple du délignage paliers....12
3.4.2 Exemple de conditions aux limites libre-libre et appuyé-appuyé....12
3.5 Calcul des matrices élémentaires et assemblage....12
3.5.1 Matrices de raideur, de masse et d'amortissement mécaniques....12
3.5.2 Matrices gyroscopiques....13
4 Calculs de type « machines tournantes »....14
4.1 Calcul modal du système quadratique....14
4.1.1 Calcul direct pour un faible nombre de degrés de liberté....14
4.1.2 Calcul en 2 étages pour un grand nombre de degrés de liberté....14
4.1.3 Exemple de génération d'une base modale utile....15
4.1.4 Modélisation de l'amortissement en dynamique linéaire....17
4.1.5 Autres opérateurs de calcul modal....17
4.2 Calcul statique....17
4.3 Calcul de la réponse au balourd....18
4.3.1 Définition d'un effort tournant de type balourd en harmonique....18
4.3.2 Définition d'un effort tournant de type balourd en transitoire....19
4.3.3 Calcul de la réponse harmonique au balourd....19
4.3.3.1 Réponse harmonique sur base physique....20

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.3.3.2 Réponse harmonique sur base modale....20
4.3.4 Calcul de la réponse transitoire au balourd....21
4.3.4.1 Réponse transitoire sur base physique....21
4.3.4.2 Réponse transitoire sur base modale....21
4.3.4.3 Calcul de ralentissement à vitesse imposée....22
4.4 Calcul de rotor fissuré....23
4.4.1 Démarche....23
4.4.2 Détermination de la loi de raideur de la fissure....24
4.4.3 Introduction de la fissure dans le maillage....25
4.4.4 Modification du fichier de commandes....25
4.4.5 Détermination de la base modale....26
4.4.6 Réalisation d'un calcul transitoire....28
5 Post-traitements pour les études « machines tournantes »....29
5.1 Sensibilités modales au balourd....29
5.2 Coefficients de participation....30
5.3 Détermination des phases et amplitudes en harmonique....31
5.4 Détermination des ellipses des rotors....32
5.5 Diagramme de Campbell....33
6 Références bibliographiques....33
7 Annexes....35
7.1 Tableau des cas-tests machines tournantes dans Code_Aster....35

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Mise en équation

1.1 Équations du mouvement

Cette partie a pour objectif de présenter les aspects essentiels d'un calcul de ligne d'arbres. Pour cela, nous allons étudier le modèle simple de rotor de la Figure 1.1-a.

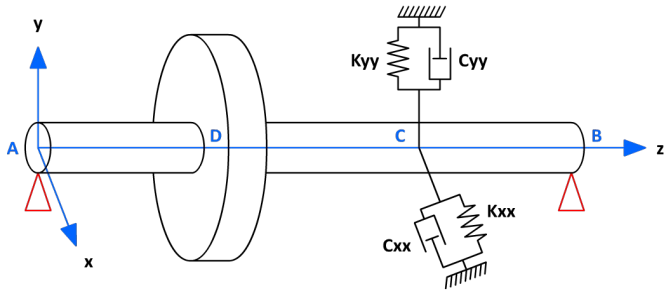

Figure 1.1-a : Modèle simplifié

Le rotor est modélisé par une poutre qui comporte un disque au tiers de sa longueur. Il repose, d'une part, sur deux appuis infiniment rigides à chaque extrémité et, d'autre part, sur un palier linéaire dans les directions X et Y situé aux deux tiers de la poutre.

Les expressions des énergies cinétiques de l'arbre, des disques et des paliers sont nécessaires pour caractériser ces dernier. L'énergie de déformation est nécessaire à la caractérisation de l'arbre. Nous utilisons aussi les forces dues aux paliers afin de calculer le travail virtuel, et ainsi obtenir les efforts correspondants exercés sur l'arbre. Nous ne présenterons ici que les principales équations de comportement, sans nécessairement détailler les calculs pour y arriver.

Les équations générales du système sont donc obtenues selon le schéma suivant : on calcule pour les éléments du système, l'énergie cinétique T, l'énergie de déformation U et le travail virtuel des efforts extérieurs, puis on applique les équations de Lagrange sous la forme :

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right)-\frac{\partial T}{\partial q_i}+\frac{\partial U}{\partial q_i}=F_{q_i} \quad \text{avec} \quad 1<i<N \qquad \textbf{éq 1.1-1}$$

où N est le nombre de degrés de liberté, q_i sont les coordonnées indépendantes généralisées et F_{q_i} les forces généralisées.

On suppose que le disque est indéformable, et on se limite donc à sa caractérisation par son énergie cinétique. Ainsi, le disque est caractérisé par sa matrice d'inertie en son centre de gravité :

$$[I]=\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \qquad \textbf{éq 1.1-2}$$

L'arbre est présenté comme une poutre à section circulaire constante et est caractérisé par ses énergies cinétique et de déformation.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les termes de raideurs et d'amortissement sont supposés connus (cf. sous-section 6 pour plus de détails sur l'identification de ces coefficients). Les forces exercées par le palier sur l'arbre peuvent donc s'exprimer comme suit :

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = -\begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} C_{xx} & K_{xy} \\ C_{yx} & K_{yy} \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \qquad \textbf{éq 1.1-3}$$

La Figure 1.1-b présente le modèle et les notations adoptés pour les paliers tandis que la Figure 1.1-a présentent une vue schématique de la ligne d'arbres.

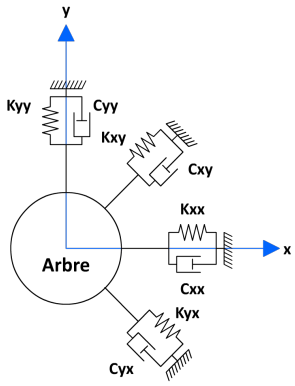

Figure 1.1-b : Modélisation linéaire des paliers

A partir du calcul des énergies cinétiques et de déformations de tous les éléments, et de l'équation éq 1.1-1 de Lagrange, on obtient des équations de comportement qui peuvent être réécrites sous une forme matricielle (voir équation éq 1.1-4). Nous omettons volontairement le détail des calculs et renvoyons à l'ouvrage de Lalanne et Ferraris ([1]).

$$M.\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + C.\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \Omega.G.\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + K.\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} \qquad \textbf{éq 1.1-4}$$

Tableau 1.1-1

où M, C, G et K sont, respectivement, les matrices carrées de masse, d'amortissement, de gyroscopie et de raideur. On remarquera que la matrice de gyroscopie est multipliée par la vitesse de rotation du rotor Ω. Le comportement de la ligne d'arbre dépend donc de sa vitesse de rotation. La solution homogène de cette équation permet de trouver les fréquences et modes propres du rotor.

2 Modélisation numérique des machines tournantes

Une ligne d'arbre comporte un nombre d'éléments dimensionnels (géométrie, caractéristiques des matériaux différentes) relativement important, il est donc délicat de déterminer un profil qui soit à la fois simple et représentatif de son comportement élastique.

2.1 Modélisation des parties tournantes

Les rotors se prêtent plutôt bien à une modélisation filaire. D'un point de vue dynamique, les lignes d'arbres sont des éléments flexibles et il est nécessaire de prendre en compte le cisaillement dans le calcul du comportement. C'est pourquoi, une modélisation par des éléments de type poutre de Timoshenko, prenant en compte l'effet gyroscopique, la déformation d'effort tranchant et l'inertie de rotation des sections, a été choisie.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

La théorie des poutres ne permet pas de traiter correctement les changements brusques de section. En présence d'un changement de section important, la masse mise en jeu dans le mouvement correspond bien à la géométrie de la section. En revanche, en ce qui concerne la déformation, celle-ci est plus limitée et l'utilisation des dimensions de la structure pour la définition des sections des éléments aboutit à une forte sur-évaluation de la rigidité de la zone considérée.

Code_Aster doit donc permettre, pour chaque élément de rotor, un traitement dissocié des termes de masse et de rigidité. A chaque section d'un élément sont associées une section masse utilisée pour le calcul des termes élémentaires de masse et une section raideur utilisée pour le calcul des termes élémentaires de rigidité. La mise en données actuelle des modèles de lignes d'arbres se fait donc par 2 géométries duales (diamètre en raideur / diamètre en masse). Cette dualité diamètre de masse / diamètre de raideur s'explique par la règle empirique des $17°$.

Chaque élément de rotor s'appuie donc sur deux mailles ayant les mêmes deux nœuds mais pouvant avoir des caractéristiques géométriques (diamètres) identiques ou différentes. De manière générale, la section masse correspond à la géométrie de la structure. Les sections raideur peuvent ne pas être en cohérence avec la géométrie de la structure du rotor notamment dans le cas de changement brusque de section.

Les disques ailettés sont considérés comme rigides et modélisés par des masses et inerties équivalentes. Cela permet de négliger les modes locaux dus aux ailettes des rotors et de simplifier le modèle de la structure. La représentation schématique d'une ligne d'arbres CP0-CP1 est visible sur la figure ci-dessous.

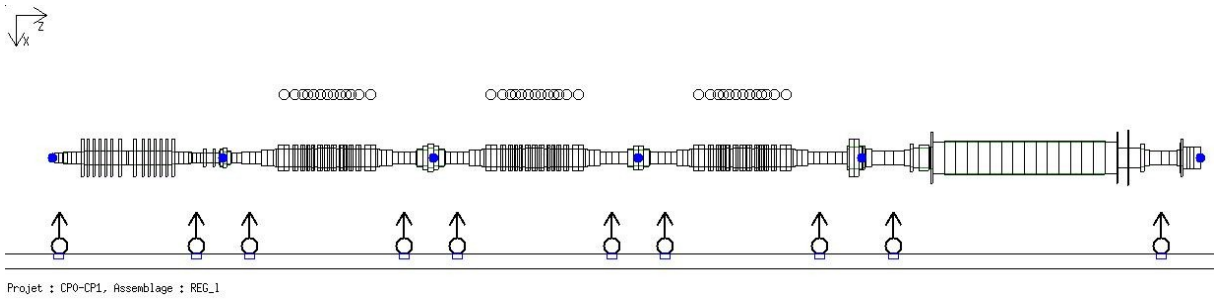

Figure 2.1-a: Représentation éléments finis du CP0-CP1

2.2 Modélisation des paliers de supportage

La Figure 2.2-a présente les modèles et les notations adoptés.

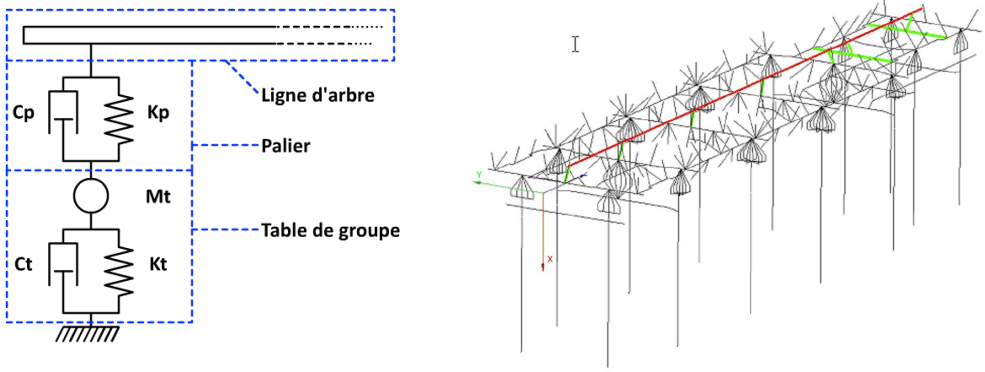

Figure 2.2-a : Représentations des paliers et de la table de groupe

Dans une turbine du parc nucléaire, la ligne d'arbres est supportée par des paliers qui reposent sur le massif.

2.2.1 Paliers linéaires

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Dans une première approximation, un comportement linéaire peut être retenu pour les paliers, ce comportement étant fonction de la vitesse de rotation. En particulier, avec l'hypothèse des petits déplacements, les coefficients de raideur et d'amortissement peuvent être calculés en linéarisant les équations de Reynolds autour de la position d'équilibre. Puis en calculant le travail virtuel $W_{paliers}$ des forces externes agissant sur l'arbre, il vient :

$$\delta W_{paliers} = \begin{bmatrix} F_x & F_y \end{bmatrix} \begin{Bmatrix} \delta x \\ \delta y \end{Bmatrix}$$

avec F_x et F_y les composantes de la force généralisée agissant sur les paliers.

En supposant que le rotor est parfaitement rigide, que les déplacements de l'arbre se limitent au voisinage d'une position d'équilibre statique (indicée 0) et en considérant le cadre des petites perturbations (petits déplacements (x, y) et petites vitesses $(\dot{x}, \dot{y})$), les composantes des forces aux paliers s'écrivent :

$$\begin{cases} F_x(x+x_0, y+y_0, \dot{x}, \dot{y}) = F_x(x_0, y_0, 0, 0) + x\left[\frac{\partial F_x}{\partial x}\right]_0 + y\left[\frac{\partial F_x}{\partial y}\right]_0 + \dot{x}\left[\frac{\partial F_x}{\partial \dot{x}}\right]_0 + \dot{y}\left[\frac{\partial F_x}{\partial \dot{y}}\right]_0 + \dots \\ F_y(x+x_0, y+y_0, \dot{x}, \dot{y}) = F_y(x_0, y_0, 0, 0) + x\left[\frac{\partial F_y}{\partial x}\right]_0 + y\left[\frac{\partial F_y}{\partial y}\right]_0 + \dot{x}\left[\frac{\partial F_y}{\partial \dot{x}}\right]_0 + \dot{y}\left[\frac{\partial F_y}{\partial \dot{y}}\right]_0 + \dots \end{cases}$$

avec $k_{ij} = -\frac{\partial F_i}{\partial x}$ et $c_{ij} = -\frac{\partial F_i}{\partial \dot{x}_j}$ qui correspondent aux raideurs et aux amortissements dus au film d'huile. Ces coefficients peuvent être calculés analytiquement en fonction de la géométrie du palier, que l'on peut retrouver dans la littérature spécialisée pour des géométries très simples de paliers.
En se limitant au premier ordre, les forces exercées par le fluide sur le rotor, suite aux déplacements du centre du rotor, peuvent se mettre sous forme matricielle :

$$\begin{Bmatrix} F_x \\ F_y \end{Bmatrix} = -\begin{bmatrix} k_{xx} & k_{xy} \\ k_{yx} & k_{yy} \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} - \begin{bmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{bmatrix} \begin{Bmatrix} \dot{x} \\ \dot{y} \end{Bmatrix}$$

avec les termes de matrices raideur et amortissement décrits par des fonctions linéaires par morceaux de la vitesse de rotation. Les valeurs des coefficients sont habituellement calculées par le *Code_Edyos*.

2.2.2 Paliers non linéaires

En général, le comportement non linéaire des paliers peut être approché par une représentation linéaire de l'équation de Reynolds autour d'un point de fonctionnement. Or, d'après le REX des études EDF, pour certains paliers fluides, où les amplitudes des vibrations deviennent excessives, il est préférable d'adopter un comportement non linéaire, notamment lors du passage des vitesses critiques où le film d'huile est fortement écrasé.

Dans le cas des groupes turbo-alternateurs, il est supporté par des paliers hydrodynamiques qui ne peuvent être considérés comme passifs mais comme des éléments qui influent dans la dynamique de la ligne d'arbres. En effet, le film d'huile a des propriétés de raideur et d'amortissement qui varient selon le régime de fonctionnement de la turbine, *i.e.* de l'excentricité du rotor et de la vitesse de rotation (notamment aux vitesses critiques), et contrôlent la stabilité de l'ensemble.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les réactions aux paliers sont obtenues après intégration du champ de pression calculé à partir de l'équation non linéaire de Reynolds. La résolution simultanée des équations du mouvement du rotor et du comportement hydrodynamique de chaque palier peut s'avérer complexe et coûteuse en temps de calcul. Cette fonctionnalité de couplage n'est pas actuellement disponible dans une version stable de Code_Aster.

2.3 Modélisation de la table de groupe

Les paliers, via les chaise palières, reposent sur un élément nommé table de groupe. Cette table est composée d'une multitude d'éléments, et la modéliser est relativement délicat. La modélisation de la table de groupe utilisée dans la majorité des études est très simpliste. Il s'agit soit d'une représentation ponctuelle au droit de chaque palier caractérisée par un système masse-ressort-amortisseur (voir Figure 2.2-a à gauche) soit de l'encastrement des paliers (table de groupe indéformable). Des représentations 3D complexes ou massif généralisé (éléments filaires ou volumiques) issues du génie civil existent aussi (voir Figure 2.2-a à droite). Les supports peuvent donc être considérés comme infiniment rigides, modélisés de façon simplifiée ou modélisés de façon fine et complète.

2.3.1 Appuis infiniment rigides

Dans ce cas, le support n'est pas modélisé et le palier (linéaire ou non-linéaire) est relié à un nœud encastré. Il suffit alors d'imposer des conditions aux limites de type appui.

2.3.2 Supports simplifiés

Dans ce cas de figure, le support est modélisé par un élément discret dont il faut définir les caractéristiques élémentaires suivantes :

1) la matrice de masse ;
2) la matrice de rigidité ;
3) la matrice d'amortissement.

2.3.3 Supports tridimensionnels

Le support 3D issu du Génie Civil est modélisé par un réseau de poutre (modélisation de la table de groupe et de ses parties statiques par des éléments 1D) ou par un modèle 3D, et le palier est relié directement au support.

2.3.4 Supports généralisés

Dans ce cas, le comportement du support est défini par une base modale calculée au préalable, par exemple à partir d'une modélisation 1D, 3D ou hybride 1D-3D de la table de groupe et de ses parties statiques. La prise en compte du massif généralisé peut se faire soit par calcul direct soit par sous-structuration de type Craig-Bampton ou Mac-Neal (cf. doc [U2.06.04] – Notice de construction des modèles réduits en dynamique). A titre d'exemple, le cas-test SDLV132 [U2.04.132] (cf. en annexe le tableau des cas-tests machines tournantes dans *Code_Aster*) illustre la mise en œuvre d'un calcul modal de ligne d'arbres avec son massif généralisé, ce dernier étant pris en compte par sous-structuration. La procédure employée garantit aussi la possibilité de prise en compte des spécificités technologiques (condenseur lié aux corps externes de turbine, par exemple).

2.3.5 Supports de type impédance

Le support peut également être modélisé par une matrice des impédances directes et croisées du support mesuré au niveau des paliers. Cette méthode présente l'avantage de pouvoir s'appuyer sur des relevés expérimentaux réalisés sur site.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Mise en données des études « machines tournantes »

Les opérateurs utilisés actuellement pour les études machines tournantes concernent l'affectation des modèles (AFFE_MODELE) et l'affectation des caractéristiques élémentaires (AFFE_CARA_ELEM).

3.1 Affectation des modèles

Après la lecture du maillage (via l'opérateur LIRE_MAILLAGE), la création d'un modèle simple du rotor représenté par la Figure 1.1-a se fait par l'opérateur AFFE_MODELE comme suit :

```
MODELE=AFFE_MODELE(MAILLAGE=MAIL,
                   AFFE=(_F(GROUP_MA='ROTOR',
                            PHENOMENE='MECANIQUE',
                            MODELISATION='POU_D_T',),
                         _F(GROUP_MA=('DISQUE'),
                            PHENOMENE='MECANIQUE',
                            MODELISATION='DIS_TR',),),);
```

3.2 Affectation des caractéristiques élémentaires

Après la définition (DEFI_MATERIAU) et l'affectation (AFFE_MATERIAU) des matériaux, l'affectation des caractéristiques géométriques pour la mise en donnée du modèle (le rotor, les paliers et éventuellement le support) est effectuée via l'opérateur AFFE_CARA_ELEM comme suit :

```
CARELEM=AFFE_CARA_ELEM(MODELE=MODELE,
                       POUTRE=_F(GROUP_MA='ROTOR',
                                 SECTION='CERCLE',
                                 CARA='R',
                                 VALE=(.025),),
                       DISCRET=_F(GROUP_MA='DISQUE',
                                  CARA='M_TR_D_N',
                                  VALE=(0.0383,3.E-6,1.8E-6,1.8E,
                                        6,0.,0.,0.,0.,0.,0.), ),);
```

Code_Aster permet également des options évoluées de mise en donnée de modèles de lignes d'arbres. A titre d'exemple, il s'agit de :

•***Rotor à section (circulaire) variable :***

La fonctionnalité amortissement gyroscopique est disponible dans *Code_Aster* pour les poutres de Timoshenko/Bernoulli à section circulaire variable (section à variation homothétique). Elle est introduite en faisant l'approximation que l'inertie moyenne de la poutre est la moyenne de ses inerties aux deux extrémités. La méthode de validation utilisée pour mesurer l'effet de cette hypothèse est purement numérique (cf. documentation [V2.02.127]). En effet, les résultats de référence sont obtenus par une discrétisation en éléments de poutre à section circulaire constante. Les valeurs de référence ont été encadrées par les résultats de deux modélisations :
•La première, approchant la solution exacte par valeurs inférieures, consiste à affecter à chaque élément la section circulaire constante de diamètre égal au diamètre du rotor étudié au droit du nœud initial de l'élément.
•La deuxième, approchant la solution exacte par valeurs supérieures, consiste à affecter à chaque élément la section circulaire constante de diamètre égal au diamètre du rotor étudié au droit du nœud final de l'élément.

Cette hypothèse peut s'avérer légitime dans une première approximation, mais il conviendra d'affiner le traitement si besoin.

•***Paliers à caractéristiques variables :***

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Il est possible de faire un calcul des modes en rotation d'un système d'arbres tournant en présence de paliers dont les caractéristiques en raideur et en amortissement dépendent de la vitesse de rotation. Pour un exemple mettant en œuvre cette fonctionnalité, le lecteur est invité à consulter la documentation de validation [V2.02.128].

3.3 Axe de rotation

Il est possible de mettre en données de modèles de lignes d'arbres orientées selon un axe de rotation quelconque défini par ses angles nautiques.

3.3.1 Passage Repère global / Repère local

Par ailleurs, il est permis une mise en donnée de modèles de lignes d'arbres orientées selon un axe de rotation quelconque défini par ses angles nautiques. Comme la mise en donnée des poutres du rotor est réalisée à partir de l'orientation imposée par le maillage construit (donc dans le repère global), il faut impérativement veiller à ce que les éléments discrets modélisant les disques, paliers et supports soient affectés dans le repère local en précisant `REPERE='LOCAL'`, en définissant ce repère local (mot-clé `ORIENTATION`), l'option `MECA_GYRO` permettant le changement effectif de repère.

```
CARELEM=AFFE_CARA_ELEM(MODELE=MODELE,
                       POUTRE=_F(GROUP_MA='ROTOR',
                                 SECTION='CERCLE',
                                 CARA='R',
                                 VALE=(.05),),
                       DISCRET=(_F(GROUP_MA='DISQUE',
                                   CARA='M_TR_D_N',
                                   REPERE='LOCAL',
                                   VALE=(0.0383,3.E-6,1.8E-6,
                                         1.8E,6,0.,0.,0.,0.,0.,0.),)
                                _F(GROUP_MA='PALIER',
                                   CARA='K_TR_D_N',
                                   REPERE='LOCAL',
                                   VALE=(0.,5.E7,7.E7,0.,0.,0.),
                                _F(GROUP_MA='PALIER',
                                   CARA='A_TR_D_N',
                                   REPERE='LOCAL',
                                   VALE=(0.,5.E3,7.E3,0.,0.,0.),),),),
                       ORIENTATION=_F(GROUP_MA=('DISQUE','PALIER'),
                                      CARA='ANGL_NAUT',
                                      VALE= (α,β,γ) ,),),
```

Par exemple,

- le passage d'une ligne d'arbres d'axe de rotation X à une ligne d'arbres d'axe de rotation Y se fait en utilisant les angles nautiques $(\alpha=90,\beta=0,\gamma=0)$;
- le passage d'une ligne d'arbres d'axe de rotation X à une ligne d'arbres d'axe de rotation Z se fait en utilisant les angles nautiques $(\alpha=0,\beta=-90,\gamma=0)$.

3.3.2 Convention de signe

La convention de sens de rotation des machines tournantes prise dans *Code_Aster* implique de choisir un axe de rotation et de définir le sens positif suivant l'axe du rotor comme étant le sens usuel trigonométrique (anti-horaire).

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Si on crée un modèle de GTA avec des nœuds d'abscisses croissantes en parcourant les lignes d'arbres du corps HP vers l'alternateur, le sens de rotation de la machine est positif suivant l'axe utilisé. Si au contraire on modélise en partant de l'alternateur vers le corps HP, le sens de rotation se trouve être inversé. Par souci d'exhaustivité et de généricité, il est donc nécessaire de pouvoir définir des vitesses de rotation aussi bien positives que négatives suivant n'importe quel axe de rotation. Cela nous conduit, par exemple, à pouvoir réaliser des diagrammes de Campbell ([U4.42.51] et [U4.52.52]) pour des vitesses de rotation négatives.

Le sens de rotation positif est une convention de signe. En d'autres termes, le respect de cette convention par un utilisateur standard permet de garantir la cohérence et la conformité des résultats obtenus avec la mise en donnée réalisée. L'utilisateur avancé, quant à lui, pourra choisir sa propre convention de signe (ie. faire tourner négativement sa ligne d'arbres selon son axe de rotation) tout en veillant à la cohérence entre sa modélisation, sa mise en donnée, son calcul et l'interprétation des résultats obtenus.

3.3.3 De la nécessité d'imposer un axe de rotation

Pour la commodité d'utilisation, il va être nécessaire d'imposer l'axe de rotation, de définir le sens positif suivant cet axe comme étant le sens usuel de rotation (sens trigonométrique) et de respecter ce choix du début jusqu'à la fin de l'étude. Cette imposition de l'axe de rotation ne conduit pas à s'écarter de la logique éléments finis de *Code_Aster*, mais permet de définir une logique d'utilisation, d'homogénéiser les modèles, de minimiser les erreurs de signes lors des différents échanges de données (entre *Code_Aster* et Edyos en l'occurrence). En effet, Edyos ne permettant pas le choix du repère, l'axe de rotation est forcément colinéaire à la direction z et positif suivant cet axe. Le couplage impose de renvoyer au code Edyos les déplacements et vitesses suivant les directions radiales des nœuds paliers. En retour, *Code_Aster* reçoit des efforts suivant des directions radiales à appliquer sur les nœuds palier.

Par ailleurs, les résultats en terme de raideur et d'amortissement de la base de modèle de paliers des lignes d'arbres CP2 et P4-P'4 seront directement exploitables dans *Code_Aster* , et ceci sans l'utilisation d'un changement de base qui occasionne bien souvent des erreurs de signes sur les termes croisés.

3.4 Conditions aux limites

Pour modéliser les appuis au niveau des paliers, on bloque les degrés de liberté de translation DX, DY et DZ des groupes de nœuds correspondants (ici PALIER_A et PALIER_B).

Exemple avec un rotor suivant Z ***:***

```
CONDLIM=AFFE_CHAR_MECA(MODELE=MODELE,
                       DDL_IMPO=(_F(GROUP_NO=('PALIER_A','PALIER_B'),
                                    DX=0,DY=0,DZ=0,),
                                 _F(GROUP_NO=('PALIER_A'),
                                    DRZ=0,),);
```

En plus des conditions aux limites habituelles, on peut être amené à bloquer les mouvements de corps rigides pour éviter les problèmes de pivots nuls lors de la résolution du système matriciel par *Code_Aster*. Ici, il s'agira, par exemple, de bloquer le degré de liberté de rotation DRZ au premier nœud de l'arbre.

Par ailleurs, l'opérateur d'affectation du chargement mécanique permet de tenir compte du délignage paliers.

3.4.1 Exemple du délignage paliers

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Pour les calcul sans prise en compte de la chaînette de la ligne d'arbres, on impose un appui rigide au droit des paliers. Pour le calcul en tenant compte de la chaînette, on impose l'altimétrie des différents nœuds paliers. Ces conditions aux limites en déplacement pour les nœuds paliers (avec ou sans chaînette) sont réalisées à l'aide de l'opérateur DDL_IMPO.

```
# Conditions aux limites sous poids propre avec chaînette
POIDS=AFFE_CHAR_MECA(MODELE=MODELE,
                     DDL_IMPO=(_F(GROUP_NO='N_ROTOR',  DRX=0.,),
                               _F(GROUP_NO='N_PALIER', DX=0., DZ=0.,),
                               _F(GROUP_NO='N_SUPPOR', DX=0., DZ=0.,),
                               _F(NOEUD='N2',   DY=-0.0160, ),
                               _F(NOEUD='N40',  DY=-0.0049, ),
                               _F(NOEUD='N110', DY= 0.0040, ),
                               _F(NOEUD='N180', DY= 0.0020, ),
                               _F(NOEUD='N247', DY=-0.0059, ),
                               _F(NOEUD='N258', DY=-0.0089, ),
                               _F(NOEUD='N291', DY=-0.0210, ),),
                     PESANTEUR=(_F(GRAVITE=9.81,
                                   DIRECTION=(1.,0.,0.,),),),);
```

3.4.2 Exemple de conditions aux limites libre-libre et appuyé-appuyé

```
# Conditions aux limites appuyé-appuyé
#
CLI_APP=AFFE_CHAR_CINE(MODELE=MODELE,
             MECA_IMPO=(_F(GROUP_NO='N_ROTOR', DX=0.,DRX=0.,),
                        _F(GROUP_NO='N_PALIER',DX=0.,DY=0.,DZ=0.,),
                        _F(GROUP_NO='N_SUPPOR',DX=0.,DY=0.,DZ=0.,),
                       ),);

# Conditions aux limites libre-libre
#
CLI_LIB=AFFE_CHAR_CINE(MODELE=MODELE,
                     MECA_IMPO=(_F(GROUP_NO='N_ROTOR', DX=0.,DRX=0.,),
                                _F(GROUP_NO='N_PALIER',DX=0.,),
                                _F(GROUP_NO='N_SUPPOR',DX=0.,),
                               ),);

# Conditions aux limites réelles
#
CLI_RE = AFFE_CHAR_CINE(MODELE=MODELE,
              MECA_IMPO=(_F(GROUP_NO='N_ROTOR', DX=0.,DRX=0.,),
                         _F(GROUP_NO='N_PALIER',DX=0.,),
                         _F(GROUP_NO='N_SUPPOR',DX=0.,DY=0.,DZ=0.,),
                        ),);
```

3.5 Calcul des matrices élémentaires et assemblage

3.5.1 Matrices de raideur, de masse et d'amortissement mécaniques

Le calcul des matrices élémentaires de raideur et de masse se fait via l'opérateur CALC_MATR_ELEM, options RIGI_MECA et MASS_MECA.

```
#
RIGI_ELE=CALC_MATR_ELEM(OPTION='RIGI_MECA',
                        MODELE=MODELE,
                        CHAM_MATER=CHMAT,
                        CARA_ELEM=CARELEM,
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                              CHARGE=CONDLIM,);

MASS_ELE=CALC_MATR_ELEM(OPTION='MASS_MECA',
                              MODELE=MODELE,
                              CHAM_MATER=CHMAT,
                              CARA_ELEM=CARELEM,
                              CHARGE=CONDLIM,);
```

Ensuite un concept sd-nume_ddl doit être crée pour générer la numérotation de degrés de liberté:

```
#
NUMEDDL=NUME_DDL(MATR_RIGI=RIGI_ELE, );
```

Les matrices élémentaires sont ensuite assemblées avec l'opérateur ASSE_MATRICE:

```
#
RIGIDITE=ASSE_MATRICE(MATR_ELEM=RIGI_ELE,
                      NUME_DDL=NUMEDDL,);

MASSE=ASSE_MATRICE(MATR_ELEM=MASS_ELE,
                   NUME_DDL=NUMEDDL,);
```

Remarque:
L'ensemble des trois opérations décrites ci-dessus peut être réalisé via la macro-commande ASSEMBLAGE.

La matrice d'amortissement mécanique (dans les paliers) est générée de la même façon :

```
AMOR_ELE=CALC_MATR_ELEM(OPTION='AMOR_MECA',
                              MODELE=MODELE,
                              CHAM_MATER=CHMAT,
                              CARA_ELEM=CARELEM,
                              RIGI_MECA=RIGI_ELE,
                              MASS_MECA=MASS_ELE,
                              CHARGE=CONDLIM,);

AMOR=ASSE_MATRICE(MATR_ELEM=AMOR_ELE,
                  NUME_DDL=NUMEDDL,);
```

3.5.2 Matrices gyroscopiques

La gyrocopie est disponibles pour les éléments discrets et les éléments de poutre droite de type Euler et Timoshenko, éventuellement multi-fibre et avec gauchissement. Les matrices d'amortissement et de raideur gyroscopiques sont calculées comme suit :

```
GYRO_ELE=CALC_MATR_ELEM(OPTION='MECA_GYRO',
                        MODELE=MODELE,
                        CHAM_MATER=CHMAT,
                        CARA_ELEM=CARELEM,);

GYRO=ASSE_MATRICE(MATR_ELEM=GYRO_ELE,
                  NUME_DDL=NUMEDDL,);

RIGY_ELE=CALC_MATR_ELEM(OPTION='RIGI_GYRO',
                        MODELE=MODELE,
                        CHAM_MATER=CHMAT,
                        CARA_ELEM=CARELEM,);

RIGY=ASSE_MATRICE(MATR_ELEM=RIGY_ELE,
                  NUME_DDL=NUMEDDL,);
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les contributions des matrices d'amortissement et de raideur gyroscopiques doivent être respectivement multipliées par la vitesse et l'accélération angulaires, exprimées en rad/s et rad/s/s, et sont combinées après assemblage aux matrices d'amortissement et de rigidité classiques par l'opérateur **COMB_MATR_ASSE** comme suit :

```
OM=1500.*pi/30. # Pour obtenir la vitesse de rotation en rad/s
GYOM=COMB_MATR_ASSE(COMB_R=(_F(MATR_ASSE=GYRO, COEF_R=OM,),
                            _F(MATR_ASSE=AMOR, COEF_R=1.,),))
```

Pour plus de détails sur l'utilisation de l'amortissement et de la raideur gyroscopique dans *Code_Aster*, le lecteur peut se référer à la documentation [U2.06.31].

4 Calculs de type « machines tournantes »

Quatre grandes familles de calcul de lignes d'arbres sont d'ores et déjà possibles dans *Code_Aster* :

1) Calcul statique
2) Calcul de modes propres
3) Calcul de réponse harmonique
4) Calcul de réponse transitoire linéaire

En plus des calculs sur base modale (validés avec des cas-tests de type machines tournantes de la base de tests), *Code_Aster* offre aussi la possibilité de calcul sur base physique (ce qui permet, le cas échéant, de corriger le biais lié à l'effet de la troncature modale).

Les opérateurs utilisés actuellement pour les calculs de type machines tournantes sont l'opérateur de calcul modal (MODE_ITER_SIMULT), l'opérateur de calcul statique (MECA_STATIQUE), l'opérateur de calcul harmonique, de calculs dynamiques transitoires sur bases physique et de calculs dynamiques transitoires sur bases modales (DYNA_VIBRA).

4.1 Calcul modal du système quadratique

Selon la taille du modèle considéré (nombre de degrés de liberté), le calcul modal peut être mené de deux façons différentes.

4.1.1 Calcul direct pour un faible nombre de degrés de liberté

Pour les systèmes à un nombre réduit de degrés de liberté (quelques milliers de degrés de liberté), un calcul modal direct peut être réalisé avec l'un ou l'autre des opérateurs MODE_ITER_SIMULT [U4.52.03] / MODE_ITER_INV [U4.52.04] et avec l'un ou l'autre des algorithmes de résolution (ici opérateur MODE_ITER_SIMULT et algorithme SORENSEN par exemple) :

```
MODEG=MODE_ITER_SIMULT(MATR_RIGI=RIGIDITE,
                       MATR_MASS=MASSE,
                       MATR_AMOR=GYOM,
                       METHODE='SORENSEN',
                       VERI_MODE=_F(SEUIL=1.E-3),
                       CALC_FREQ=_F(OPTION='CENTRE',
                                    FREQ=1.,
                                    NMAX_FREQ=16))
```

Selon que l'on cherche à déterminer l'intégralité ou une partie du spectre, on utilisera respectivement les algorithmes de résolution modale de type QZ ou SORENSEN.

4.1.2 Calcul en 2 étages pour un grand nombre de degrés de liberté

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Pour les analyses en base modale de systèmes à grand nombre de degrés de liberté (plus que 300 degrés de liberté), il faut procéder à un calcul en deux étages (nommé aussi calcul modal en coordonnées généralisées) :
•Calcul des matrices assemblées sur base physique :
•Calcul d'une base réduite de modes propres non amortis
•Projection des matrices de masse, masse et amortissement total dans cette base
•Calcul modal sur les matrices généralisées

Le calcul final est réalisé à partir de ces matrices projetées à l'aide de l'algorithme de résolution QZ.

4.1.3 Exemple de génération d'une base modale utile

Dans les calculs de ligne d'arbres, il est fréquent de recourir à des calculs sur base modale dans le cadre de l'approche de la réduction du modèle. Pour cela, il est d'usage de générer une base modale mixte incluant des modes appuyés-appuyés au droit des paliers et des modes libres-libres, la différence étant donnée par les jeux de conditions aux limites qui sont différentes. L'intérêt de cette base composée est de créer un espace de projection le plus complet possible pour représenter les solutions admissibles.
La faisabilité de cette procédure est montrée dans ce qui suit sur la base d'un cas industriel. Les conditions aux limites libres-libres CLI_LIB , appuyés-appuyés CLI_APP , ou réelles CLI_RE sont imposées par le bais de l'opérateur AFFE_CHAR_CINE et définies dans le paragraphe 3.3.2.

Génération de la base complète
1) Base appuyé-appuyé:
A l'aide de l'opérateur MODE_ITER_SIMULT, on calcule les modes appuyés-appuyés conservatifs (donc sans tenir compte des matrices d'amortissement gyroscopique ou non) qui ont une fréquence inférieure à une valeur fixée par l'utilisateur (ici $100\,Hz$).

```
ASSEMBLAGE(     MODELE=MODELE,
                CHAM_MATER=CHMAT,
                CARA_ELEM=CARAELEM,
                CHAR_CINE=CLI_APP,
                NUME_DDL=CO('nddl_app'),
                MATR_ASSE=(_F(MATRICE=CO('M_app'),
                              OPTION='MASS_MECA',),
                           _F(MATRICE=CO('R_app'),
                              OPTION='RIGI_MECA',),
                      ),);

base_app=MODE_ITER_SIMULT(MATR_RIGI=R_app,
                          MATR_MASS=M_app,
                          CALC_FREQ=_F(OPTION='BANDE',
                                       FREQ=(-1.,100.,),),);
```

2) Base libre-libre:
A l'aide de l'opérateur MODE_ITER_SIMULT , on calcule les nb_pal premiers modes libres-libres conservatifs, nb_pal étant égal à deux (pour tenir compte des deux directions de déplacement radiaux des nœuds) fois le nombre de paliers. Ils sont calculés à l'aide de l'opérateur MODE_ITER_SIMULT .

```
nb_pal=7 # nombre de paliers

ASSEMBLAGE(     MODELE=MODELE,
                CHAM_MATER=CHMAT,
                CARA_ELEM=CARAELEM,
                CHAR_CINE=CLI_LIB,
                NUME_DDL=CO('nddl_lib'),
                MATR_ASSE=(_F(MATRICE=CO('M_lib'),
                              OPTION='MASS_MECA',),
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                              _F(MATRICE=CO('R_lib'),
                                 OPTION='RIGI_MECA',),
                           ),);

base_lib=MODE_ITER_SIMULT(MATR_RIGI=R_lib,
                          MATR_MASS=M_lib,
                          CALC_FREQ=_F(OPTION='PLUS_PETITE',
                                       NMAX_FREQ=(2*nb_pal),
                              SEUIL_FREQ = 0.,),
                           );
```

3) Assemblage des matrices:
La numérotation des degrés de liberté choisie pour l'assemblage est celle correspondant aux conditions aux limites réelles.

```
ASSEMBLAGE(     MODELE=MODELE,
                CHAM_MATER=CHMAT,
                CARA_ELEM=CARAELEM,
                CHAR_CINE=CLI_RE,
                NUME_DDL=CO('nddl_re'),
                MATR_ASSE=(_F(MATRICE=CO('M_re'),
                              OPTION='MASS_MECA',),
                           _F(MATRICE=CO('R_re'),
                              OPTION='RIGI_MECA',),
                           _F(MATRICE=CO('A_re'),
                              OPTION='AMOR_MECA',),
                           _F(MATRICE=CO('GY_re'),
                              OPTION='MECA_GYRO',),
                          ),);

A_GY_re=COMB_MATR_ASSE(COMB_R=(_F(MATR_ASSE=GY_re,
                                  COEF_R=OM,),
                               _F(MATR_ASSE=A_re,
                                  COEF_R=1.0,),),);
```

Les deux bases modales libres-libres, ou appuyés-appuyés sont ensuite concaténées à l'aide de l'opérateur DEFI_BASE_MODALE en une unique base de type RITZ :

```
base_tot=DEFI_BASE_MODALE(RITZ=(_F( MODE_MECA = base_lib,
                                     NMAX_MODE = 2*nb_pal),
                                 _F( MODE_INTF = base_app,
                                     NMAX_MODE = 1000),),
# on prend la numérotation de la base réelle
                          NUME_REF=nddl_re,
                         );
```

Projection des matrices sur la base complète:

```
ngen_tot=NUME_DDL_GENE(BASE= base_tot,
                       STOCKAGE= 'PLEIN',)

M_pr_tot=PROJ_MATR_BASE(BASE=base_tot,
                        NUME_DDL_GENE=ngen_tot,
                        MATR_ASSE=M_re,);

R_pr_tot=PROJ_MATR_BASE(BASE=base_tot,
                        NUME_DDL_GENE=ngen_tot,
                        MATR_ASSE=R_re,);

A_pr_tot=PROJ_MATR_BASE(BASE=base_tot,
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                NUME_DDL_GENE=ngen_tot,
                                MATR_ASSE=A_GY_re,);
```

Calcul modal sur base réduite

```
MOD2ETAG=MODE_ITER_SIMULT(MATR_RIGI=R_pr_tot,
                          MATR_MASS=M_pr_tot,
                          MATR_AMOR=A_pr_tot,
                          METHODE='QZ',
                          CALC_FREQ=_F(OPTION='TOUT'),)
```

4.1.4 Modélisation de l'amortissement en dynamique linéaire

En plus de l'amortissement apporté par les paliers hydrodynamiques, il existe une autre source d'amortissement dans les machines tournantes. Il s'agit de l'amortissement matériau (béton du Génie Civil, par exemple). Il existe plusieurs modélisations de l'amortissement :
•l'amortissement visqueux
•l'amortissement hystérétique (dit aussi « amortissement structural »)

Pour plus de détails, le lecteur est renvoyé à la documentation de référence [R5.05.04].
En ce qui concerne les études de machines tournantes, il est également d'usage d'effectuer des analyses linéaires en utilisant les méthodes de réponse dynamique par recombinaison modale, avec une base modale de modes propres réels. Dans ce cas, l'amortissement est utilisé pour la stabilisation numérique ou alors pour simuler des résultats expérimentaux (recalage).

Dans *Code_Aster*, il est possible d'introduire des coefficients d'amortissement modaux. Dans les opérateurs DEFI_BASE_MODALE , DYNA_LINE_HARM, DYNA_TRAN_MODAL et DYNA_VIBRA, les mot-clés AMOR_REDUIT et/ou LIST_AMOR permettent d'ajouter des amortissements réduits (pourcentages de l'amortissement critique) pour les modes même si à l'origine ils sont des modes réels.

Dans le cas d'un calcul par sous-structuration dynamique, les amortissements réduits sont définis pour chaque sous-structure séparément (opérateur MACR_ELEM_DYNA [U4.65.01]).

4.1.5 Autres opérateurs de calcul modal

Nous signalons, à titre d'information, que d'autres opérateurs de *Code_Aster* peuvent être utilisés pour le calcul modal. Il s'agit de :

1) MACRO_MODE_MECA (cf [U4.52.02]) pour lancer une succession de calculs de modes propres ;
2) CALC_MODAL (cf [U4.52.10]) pour lancer dans une seule commande l'intégralité de la chaîne de calcul de modes propres, ie CALC_MOTR_ELEM, NUME_DDL, ASSE_MATRICE et MODE_ITER_SIMULT ;

4.2 Calcul statique

En parallèl e avec un simple calcul modal (calcul des fréquences propres et des déformées modales), une première validation élémentaire du modèle de ligne d'arbres construit peut être réalisée, par exemple, par un calcul de déformée statique sous l'action de forces ou de moments et avec ou sans gravité. L e chargement mécanique s'effectue alors via l'opérateur AFFE_CHAR_MECA avec ou sans le mot clé GRAVITE . Ce type de calcul sert généralement à la validation des modèles numériques.

Par exemple, le calcul des efforts résultants de la configuration chaînette de ligne d'arbres se fait via l'opérateur MECA_STATIQUE [U4.51.01] comme suit :

```
STAT=MECA_STATIQUE(MODELE=MODELE,
                   CHAM_MATER=CHMAT,
                   CARA_ELEM= CARELEM,
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                    EXCIT=(_F(CHARGE= POIDS,),),
                    );
```

Le calcul des efforts aux appuis paliers dans cette configuration se fait en utilisant les opérateurs de post-traitement des résultats. L'obtention des résultats fait appel à l'opérateurs de post-traitement CALC_CHAMP, avec l'option REAC_NODA permettant d'obtenir les efforts aux appuis paliers dans cette configuration :

```
STAT=CALC_CHAMP(reuse=STAT,
                RESULTAT=STAT,
                FORCE=('REAC_NODA'),);
```

Les réactions d'appui au niveau des nœuds paliers sous poids propre permettent de connaître la charge supportée par chaque palier. À l'aide d'une étude Edyos, il reste à déterminer la position correspondante et les coefficients linéarisés.

4.3 Calcul de la réponse au balourd

4.3.1 Définition d'un effort tournant de type balourd en harmonique

Pour créer un effort de type balourd, il est nécessaire de créer un effort de module proportionnel à ω^2, et d'en effectuer une décomposition en une composante en cosinus et une composante en sinus suivant les deux directions radiales. Dans un calcul harmonique, le chargement est la partie réelle de l'excitation et donc en cosinus. Comme $\cos(\omega \cdot t - \pi/2) = \sin(\omega \cdot t)$. Le chargement suivant l'autre direction radiale est donc affecté d'une phase de $-90°$ en plus du déphasage.

Exemple de rotation suivant l'axe OX ***:***

```
balou = 0.05;
```
Valeur du balourd en $kg.m$

```
PHI = 0.0;
```
angle entre la direction x et celle du balourd à l'instant initial.

```
NTRIM = 15000.0;
```
vitesse de rotation en $tours/min$

```
FREQ = (NTRIM / 60.0);
```
fréquence de rotation en Hz

```
OM = ((FREQ * 2) * pi);
```
vitesse de rotation en rad/s

```
AMOGYRO=COMB_MATR_ASSE(COMB_R=(_F(MATR_ASSE=GYRO,
                                  COEF_R=OM,),
                               _F(MATR_ASSE=AMOR,
                                  COEF_R=1.0,),),);
```

Création d'un balourd de module $balou \times OM^2$ sur le nœud N_DIS

```
bal_FY=AFFE_CHAR_MECA(MODELE=MODELE,
                      FORCE_NODALE=_F(GROUP_NO='N_DIS2',
                                      FY=(balou * (OM ** 2)),),);

bal_FZ=AFFE_CHAR_MECA(MODELE=MODELE,
                      FORCE_NODALE=_F(GROUP_NO='N_DIS2',
                                      FZ=(balou * (OM ** 2)),),);
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

La dérivation du vecteur élémentaire correspondant au chargement mécanique (option CHAR_MECA) se fait à l'aide de l'opérateur CALC_VECT_ELEM .

```
bal_FYe=CALC_VECT_ELEM(OPTION='CHAR_MECA',
                       CHARGE=bal_FY,);

bal_FZe=CALC_VECT_ELEM(OPTION='CHAR_MECA',
                       CHARGE=bal_FZ,);
```

L'assemblage se fait via l'opérateur ASSE_VECTEUR selon bien évidement la même numérotation des degrés de liberté que celle des matrices assemblées.

```
bal_Yass=ASSE_VECTEUR(VECT_ELEM=bal_FYe,
                      NUME_DDL=NUMEDDL,);

bal_Zass=ASSE_VECTEUR(VECT_ELEM=bal_FZe,
                      NUME_DDL=NUMEDDL,);
```

4.3.2 Définition d'un effort tournant de type balourd en transitoire

Pour créer un effort de type balourd, il est nécessaire de créer un effort de module proportionnel à ω^2 et d'en effectuer une décomposition en une composante en cosinus et une composante en sinus suivant les deux directions radiales. L'étape de construction des vecteurs de chargement assemblées est la même qu'en harmonique. La différence pour le transitoire réside dans la nécessité de définir les deux fonctions cosinus et sinus ci-dessous pour appliquer les efforts.

Exemple de rotation suivant l'axe OX ***:***

```
Fory=FORMULE(VALE='cos(OM*INST+PHI)',
       NOM_PARA='INST',);

Forz=FORMULE(VALE='sin(OM*INST+PHI)',
     NOM_PARA='INST',);
```

Suivant l'axe de rotation :

Rotation	Suivant X		Suivant Y		Suivant Z	
Phase mesurée par rapport à	Y		Z		X	
Effort	Y	$\cos(\omega t+\phi)$	Z	$\cos(\omega t+\phi)$	X	$\cos(\omega t+\phi)$
Effort	Z	$\sin(\omega t+\phi)$	X	$\sin(\omega t+\phi)$	Y	$\sin(\omega t+\phi)$

Tableau 4.3.2-1

4.3.3 Calcul de la réponse harmonique au balourd

Le calcul de réponse harmonique consiste à déterminer rigoureusement la réponse d'une machine en régime permanent, tournant à vitesse constante et soumise aux excitations naturelles (balourd et défaut de fibre neutre) ou à des efforts harmoniques d'origine extérieure (excitateur électromagnétique par exemple). Il suppose que le transitoire de chargement est amorti. Cette hypothèse implique que tous les modes propres ont un amortissement strictement positif. Il suppose également que le comportement de la machine est linéaire et que la structure est soumise à des chargements harmoniques de même pulsation d'excitation.

Afin de déterminer les vitesses critiques de rotation, il est nécessaire de rechercher les vitesses de rotation pour lesquelles l'amplitude de vibration est maximale. Pour ce faire, la résolution du problème

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

harmonique de réponse à balourd peut être effectuée en considérant l'une après l'autre, toutes les fréquences d'excitation contenues dans une liste définie par l'utilisateur. On suppose un chargement synchrone avec la vitesse de rotation de la machine et on réalise un balayage des fréquences d'excitation (et donc des vitesse de rotation) avec un pas de discrétisation. La valeur de ce pas est choisie judicieusement pour permettre de capter toutes les vitesses critiques dans la plage étudiée.

4.3.3.1 Réponse harmonique sur base physique

Le calcul de la réponse harmonique à balourd se fait via l'opérateur DYNA_LINE_HARM .

```
FUN=DEFI_CONSTANTE(VALE=1.0,);

DHAM=DYNA_LINE_HARM(MATR_MASS=MASSE,
                    MATR_RIGI=RIGIDITE,
                    MATR_AMOR=AMOGYRO,
                    FREQ=FREQ,
                    EXCIT=(_F(VECT_ASSE=bal_Yass,
                              FONC_MULT=FUN,
                              PHAS_DEG= PHI,),
                           _F(VECT_ASSE=bal_Zass,
                              FONC_MULT=FUN,
                              PHAS_DEG= -90.0+PHI,),
                           ), );
```

4.3.3.2 Réponse harmonique sur base modale

Le calcul harmonique nécessite une résolution de type statique qui est peu coûteuse. L'intérêt de la réduction modale pour un calcul harmonique est donc limité étant donné la rapidité d'un tel calcul. Toutefois, les deux calculs, sur bases physique et modale, peuvent être menés et conduisent aux mêmes résultats du moment que la base modale est suffisamment représentative.
En plus de la constitution de la base modale et de la projection des matrices assemblées, il est nécessaire de projeter les chargements mécaniques (balourd dans ce cas). La projection du chargement est effectuée via l'opérateur PROJ_BASE.

```
PROJ_BASE(BASE=MODES,
                PROFIL='PLEIN',
                MATR_ASSE_GENE=(_F(MATRICE=CO('MAGE'),
                                   MATR_ASSE=MASSE,),
                                _F(MATRICE=CO('RIGE'),
                                   MATR_ASSE=RIGIDITE,),
                                _F(MATRICE=CO('AGGE'),
                                   MATR_ASSE=AMOGYRO,),),
                VECT_ASSE_GENE=(_F(VECTEUR=CO('BYGE'),
                                   VECT_ASSE=bal_Yass,),
                                _F(VECTEUR=CO('BZGE'),
                                   VECT_ASSE=bal_Zass,),),);

DHAM=DYNA_LINE_HARM(MATR_MASS=MAGE,
                    MATR_RIGI=RIGE,
                    MATR_AMOR=AGGE,
                    FREQ=FREQ,
                    EXCIT=(_F(VECT_ASSE=BYGE,
                              FONC_MULT=FUN,
                              PHAS_DEG= PHI,),
                           _F(VECT_ASSE=BZGE,
                              FONC_MULT=FUN,
                              PHAS_DEG= -90.0+PHI,),
                           ),
                    );
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Le tableau pour choisir la phase suivant l'axe de rotation :

Rotation	Suivant X		Suivant Y		Suivant Z	
Phase mesurée par rapport à	Y		Z		X	
Effort et phase	Y	ϕ	Z	ϕ	X	ϕ
Effort et phase	Z	$-90°+\phi$	X	$-90°+\phi$	Y	$-90°+\phi$

Tableau 4.3.3.2-1

4.3.4 Calcul de la réponse transitoire au balourd

Le calcul de la réponse transitoire consiste à déterminer, en fonction du temps, la réponse d'une structure soumise à divers chargements.

Contrairement au calcul de la réponse harmonique en régime établi présenté dans le chapitre précédent, aucune hypothèse n'est faite ni sur les caractéristiques des chargements appliqués, ni sur la forme de la réponse correspondante.

4.3.4.1 Réponse transitoire sur base physique

```
tdeb = 0.0;
tfin = 5.0;
pdtf = 1e-04;

LINST=DEFI_LIST_REEL(DEBUT=tdeb,
                     INTERVALLE=_F(JUSQU_A=tfin,
                                   PAS=pdtf,),);

TRANSI=DYNA_LINE_TRAN(MODELE=MODELE,
                      CHAM_MATER=CHMAT,
                      CARA_ELEM=CARELEM,
                      MATR_MASS=MASSE,
                      MATR_RIGI=RIGIDITE,
                      MATR_AMOR=AMOGYRO,
                      NEWMARK=_F(),
                      EXCIT=(_F(VECT_ASSE=bal_Yass,
                                FONC_MULT=fory,),
                             _F(VECT_ASSE=bal_Zss,
                                FONC_MULT=forz),),
                      INCREMENT=_F(LIST_INST=LINST,),);
```

4.3.4.2 Réponse transitoire sur base modale

La projection des matrices et du chargement est ici effectuée via les opérateurs PROJ_MATR_BASE et PROJ_VECT_BASE. L'opérateur DYNA_TRAN_MODAL permet ensuite la résolution du problème transitoire sur base modale.

```
MAGE = PROJ_MATR_BASE(BASE= MODES,
                      NUME_DDL_GENE=NUMEGE,
                      MATR_ASSE=MASSE,);
RIGE = PROJ_MATR_BASE(BASE= MODES,
                      NUME_DDL_GENE=NUMEGE,
                      MATR_ASSE=RIGIDITE,);
RYGE = PROJ_MATR_BASE(BASE= MODES,
                      NUME_DDL_GENE=NUMEGE,
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                          MATR_ASSE=RIGY,);
AMGE = PROJ_MATR_BASE(BASE= MODES,
                          NUME_DDL_GENE=NUMEGE,
                          MATR_ASSE=GYRO,);
AGGE = PROJ_MATR_BASE(BASE= MODES,
                          NUME_DDL_GENE=NUMEGE,
                          MATR_ASSE=AMOGYRO,);
BYGE = PROJ_VECT_BASE(BASE= MODES,
                        NUME_DDL_GENE=NUMEGE,
                        VECT_ASSE=bal_Xass,);
BZGE = PROJ_VECT_BASE(BASE= MODES,
                        NUME_DDL_GENE=NUMEGE,
                        VECT_ASSE=bal_Yass,);

TRANSMO=DYNA_TRAN_MODAL(METHODE='NEWMARK',
                          MASS_GENE=MAGE,
                          RIGI_GENE=RIGE,
                          AMOR_GENE=AGGE,
                          INCREMENT=_F(INST_INIT=tdeb,
                                       INST_FIN=tfin,
                                       PAS=pdtf,),
                          EXCIT=(_F(VECT_GENE=BYGE,
                                    FONC_MULT=fory,),
                                 _F(VECT_GENE=BZGE,
                                    FONC_MULT=forz,),),);
```

Remarques :

1) les calculs harmoniques et transitoires présentés ci-dessus peuvent être lancés également via le nouveau opérateur unique pour la dynamique vibratoire DYNA_VIBRA (macro qui appelle les opérateurs historiques):

- harmonique sur base physique: TYPE_CALCUL = 'HARM', BASE_CALCUL='PHYS';
- harmonique sur base modale: TYPE_CALCUL = 'HARM', BASE_CALCUL='GENE';

2) Pour tous les calculs sur base modale (harmonique ou transitoire), il est nécessaire à la fin du calcul de restituer les résultats c'est à dire les déplacements aux nœuds sur la base physique. Ceci est effectué par l'intermédiaire de l'opérateur REST_GENE_PHYS.

4.3.4.3 Calcul de ralentissement à vitesse imposée

La modélisation du ralentissement transitoire d'une machine tournante nécessite la prise en compte de la dépendance des matrices d'amortissement et de raideur gyroscopiques (options MECA_GYRO et RIGI_GYRO) et des forces tournantes de balourd à la vitesse et l'accélération angulaires.
Dans le cas de consigne connue d'avance (ie. vitesse imposée ou pilotée), les transitoires lents (évolution « lente » par paliers de la vitesse de rotation) peuvent être traités comme suit :

- on définit un pas de vitesse de rotation pour la mise à jour des matrices et des efforts (vitesse de rotation constante par blocs),
- on réalise les calculs transitoires par intervalle de vitesse,
- on construit la solution par poursuites successives.

Dans le cas de consigne connue d'avance (soit vitesse imposée ou pilotée), les transitoires rapides de (évolution « rapide » de la vitesse de rotation) peuvent être traitées en ajoutant le mot-clé VITESSE_VARIABLE = 'OUI' dans les opérateurs de calcul sur base modale DYNA_TRAN_MODAL ou DYNA_VIBRA. On doit alors définir les lois de vitesse et d'accélération angulaires imposées par des formules ou des fonctions dans *Code_Aster* comme suit :

```
NTRIMdeb = 0.0;

OMdeb = 2 * pi * NTRIMdeb / 60.0; # vitesse de rotation initiale en rad/s
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
tdeb = 0.0; # instant initial en s

NTRIMfin = 5000.0;

OMfin = 2 * pi * NTRIMfin / 60.0; # vitesse de rotation finale en rad/s

tfin = 12.0; # instant final en s
```

Exemple de loi d'évolution linéaire de la vitesse de rotation

```
A = (OMdeb*tfin-OMfin*tdeb)/(tfin-tdeb);

B = (OMdeb-OMfin)/(tdeb-tfin);

C = PHIdeb - A*tdeb - 0.5*B*(tdeb**2);

VITE=FORMULE(VALE='A+B*INST',
             NOM_PARA='INST',);

ACCE=FORMULE(VALE='B',
             NOM_PARA='INST',);
```

Exemple de loi d'évolution exponentielle de la vitesse de rotation

```
alpha = 3.273e-01;

A=((exp(alpha*tfin)*OMfin)-(exp(alpha*tdeb)*OMdeb))/(exp(alpha*tfin)-
exp(alpha*tdeb));

B = (OMfin-OMdeb)/(exp(-alpha*tfin)-exp(-alpha*tdeb));

C = PHIdeb - A*tdeb - ((B/alpha)*exp(-alpha*tdeb));

VITE=FORMULE(VALE='A+B*exp(-alpha*INST)',
             NOM_PARA='INST',);

ACCE=FORMULE(VALE='-alpha*B*exp(-alpha*INST)',
             NOM_PARA='INST',);
```

Ensuite, on alimente l'opérateur de calcul par les lois angulaires VITE et ACCE ainsi que les matrices gyroscopiques assemblées AMGE et GYGE sous les mot-clés VITE_ROTA, ACCE_ROTA, MATR_GYRO et MATR_RIGY.

```
TRANSMO=DYNA_TRAN_MODAL(METHODE='NEWMARK',
                        MASS_GENE=MAGE,
                        RIGI_GENE=RIGE,
                        AMOR_GENE=AGGE,
                        VITESSE_VARIABLE='OUI',
                        VITE_ROTA=VITE,
                        MATR_GYRO=AMGE,
                        ACCE_ROTA=ACCE,
                        MATR_RIGY=RYGE,
                        INCREMENT=_F(INST_INIT=tdeb,
                                     INST_FIN=tfin,
                                     PAS=pdtf,),
                        EXCIT=(_F(VECT_GENE=BYGE,
                                  FONC_MULT=fory,),
                               _F(VECT_GENE=BZGE,
                                  FONC_MULT=forz,),),);
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Pour plus de détails, le lecteur est renvoyé au document de validation [V2.02.144] du cas-test SDLL144 mettant en œuvre un transitoire de vitesse de rotation ainsi qu'aux documents d'utilisation [U4.53.03] et [U4.53.21] des opérateurs DYNA_TRAN_MODAL et DYNA_VIBRA .

4.4 Calcul de rotor fissuré

La procédure de mise en œuvre d'un calcul de rotor sain est exposé ci-dessous. Dans cette section, on se propose de présenter les modifications à apporter à un modèle sain pour tenir compte de la présence d'une fissure.

4.4.1 Démarche

La démarche de calcul dynamique d'une ligne d'arbre fissurée dans *Code_ASTER* est représentée par le schéma ci-dessous (cf. [R5.05.08])

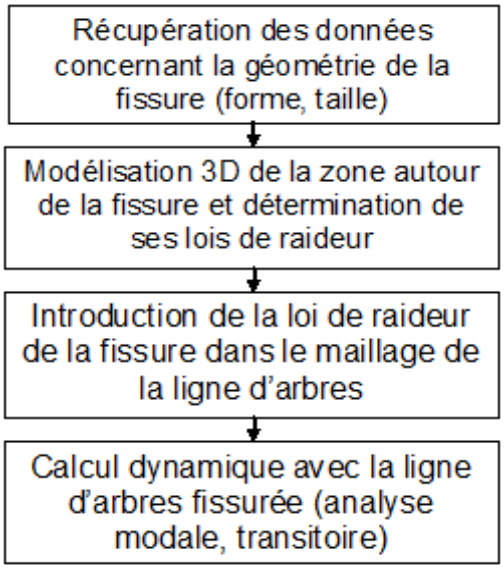

Figure 4.4.1-a: Démarche de calcul d'une ligne d'arbres fissurée dans *Code_Aster*

Le cas-test SDNL133 ([V5.02.133]) de la base de *Code_Aster* exemplifie cette fonctionnalité.

4.4.2 Détermination de la loi de raideur de la fissure

Le document de référence [R5.05.08] aborde la manière de déterminer la raideur équivalente d'une fissure. Le calcul de la loi de raideur de la fissure se fait sur un barreau cylindrique fissuré de dimensions unitaires (diamètre = 1 et longueur = 4 *diamètre). Lors de la génération du maillage du barreau fissuré plusieurs formes de fissures peuvent être modélisées (fond droit, semi-elliptique concave et convexe).

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

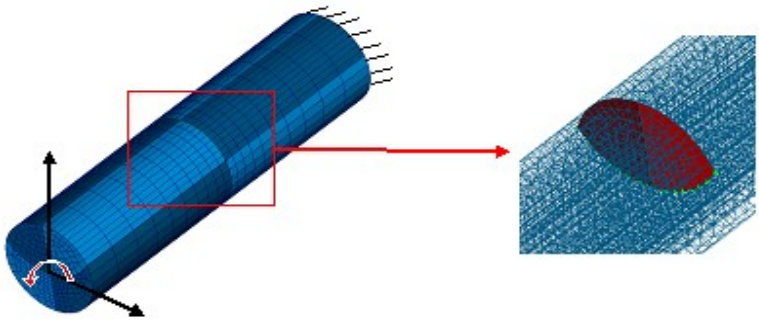

Figure 4.4.2-a: Modélisation de l'éprouvette fissurée

Le barreau fissuré est encastré sur une extrémité et un moment est appliqué sur la deuxième extrémité. L'orientation du moment évolue au cours du calcul de manière à parcourir une l'intervalle complète entre 0° et 360° avec un pas de 10°, par exemple.
Un calcul statique est ensuite réalisé sur le barreau afin de déterminer la loi de raideur de la fissure ainsi que sa dérivée. À l'issu du calcul, un traitement Python, capitalisé dans le cas-test SDNLL133D ([V5.02.133]) permet de déterminer l'évolution de la raideur ainsi que de sa dérivée avec un pas de 0.2°. Les résultats se présentent sous forme de fichier à 3 colonnes (angle, raideur, dérivée de la raideur). La Figure 4.4.2-a montre un exemple d'évolution de la raideur ainsi que de sa dérivée.

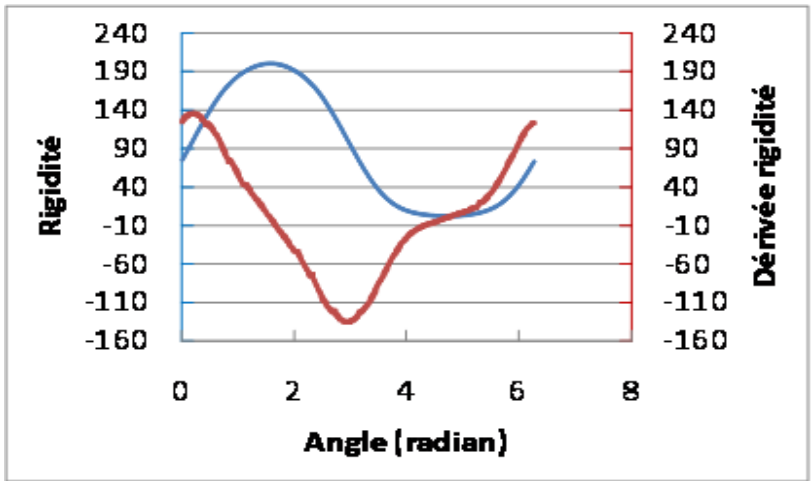

Figure 4.4.2-b: Exemple de loi de raideur de fissure

Comme les dimensions de l'éprouvette sont unitaires, la raideur ainsi que sa dérivée sont adimensionnées. Leurs niveaux seront ensuite adaptés aux dimensions de l'arbre lors du calcul dynamique.

4.4.3 Introduction de la fissure dans le maillage

L'introduction de la fissure dans le modèle se fait dans le fichier maillage du rotor. L'élément modélisant la fissure est introduit à l'emplacement de la fissure réelle (Figure 4.4.3-a) en dédoublant le nœud où se trouve la fissure et en insérant un élément discret (de type SEG2) entre les 2 nœuds générés. Dans l'exemple de la Figure 4.4.3-a , le groupe de maille de l'élément fissuré est nommé « FISSURE » et les groupes de nœuds l'entourant de part et d'autre « FISS_G » et « FISS_D ».

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

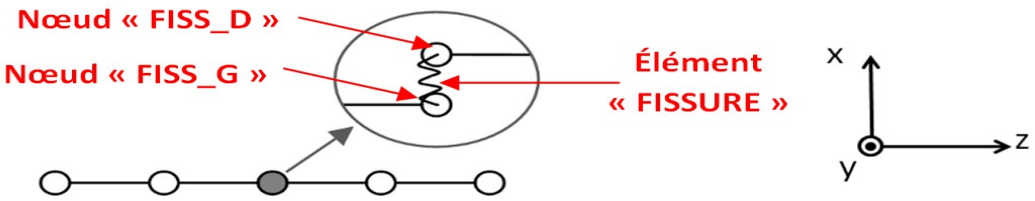

Figure 4.4.3-a : Élément discret inséré dans le modèle poutre de la ligne d'arbres saine

4.4.4 Modification du fichier de commandes

Nous supposons que l'axe de rotation de la ligne d'arbres est selon Z.
Dans un premier temps il faut affecter le type d'élément à la maille « FISSURE » déjà créée dans le fichier maillage :

```
MOD_ROT=AFFE_MODELE(MAILLAGE=ARBRE,
             AFFE=(….,
                ….,
                 _F(GROUP_MA=('FISSURE'),
                           PHENOMENE='MECANIQUE',
                           MODESILATION='DIS_TR'),
                    …,
             ) ) ;
```

Les propriétés de l'élément « FISSURE » sont ensuite définies en dédoublant l'opérateur AFFE_CARA_ELEM déjà existant dans le fichier de calcul sur le modèle sain. Il faut créer 2 affectations des caractéristiques élémentaires, une première avec fissure fermée (ou rigide) nommée CARAELEM et une deuxième tenant compte du point anguleux de l'arbre à l'endroit de fissure (arbre rotulé à cet endroit) nommée CARAELEF. La loi de raideur de la fissure n'est pas intégrée à ce stade, il s'agit ici de modéliser la fissure comme une rotule suivant les 2 axes de flexion x et y afin de déterminer la base modale.
Dans CARAELEM, les masses et raideurs sont définies comme suit :

```
CARAELEM=AFFE_CARA_ELEM(MODELE=MOD_ROT,
              …,
              _F(CARA='K_TR_D_L',
                 GROUP_MA='FISSURE,
                 VALE=(raidmax,raidmax,raidmax,raidmax,raidmax,raidmax),
               _F(CARA='M_TR_D_L',
                 GROUP_MA='FISSURE,
                 VALE=(0,0,0,0,0,0),
               …,
               );
```

La rigidité de la fissure 'raidmax' est choisie suffisamment élevée par rapport à celle de la ligne d'arbre (100 plus grande).
Dans CARAELEF, les valeurs des raideurs sont nulles suivant les 2 axes de flexion de l'arbre X et Y (afin de tenir compte de la présence de la fissure) :

```
CARAELEF=AFFE_CARA_ELEM(MODELE=MOD_ROT,
              …,
              _F(CARA='K_TR_D_L',
                 GROUP_MA='FISSURE,
                 VALE=(raidmax,raidmax,raidmax,0,0,raidmax),
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
_F(CARA='M_TR_D_L',
   GROUP_MA='FISSURE,
   VALE=(0,0,0,0,0,0),
 …,
 );
```

4.4.5 Détermination de la base modale

La base modale d'un rotor sain, décrite dans le paragraphe 15 et constituée de modes libre-libre et appuyés au droit des paliers, ne permet pas de représenter la cassure de la poutre au droit de la fissure. Par conséquent, la combinaison linéaire de ces modes propres ne peut pas rendre compte d'une déformée présentant un point anguleux puisque tous les modes de cette base modale sont continus et dérivables en ce point. La solution adoptée, pour la constitution de la base modale, est le découpage « virtuel » du rotor, au droit de la fissure, en deux demi-rotors (liaison rotule au niveau de la fissure). Ainsi le point anguleux caractérisé par des tangentes à la déformée, différentes à droite et à gauche de la fissure, est facilement représentable en utilisant un mode pour la partie gauche et un autre pour la partie droite.
Les différents groupes de nœuds de la ligne d'arbre où sont appliquées les conditions aux limites sont décrits sur la Figure 4.4.5-a.

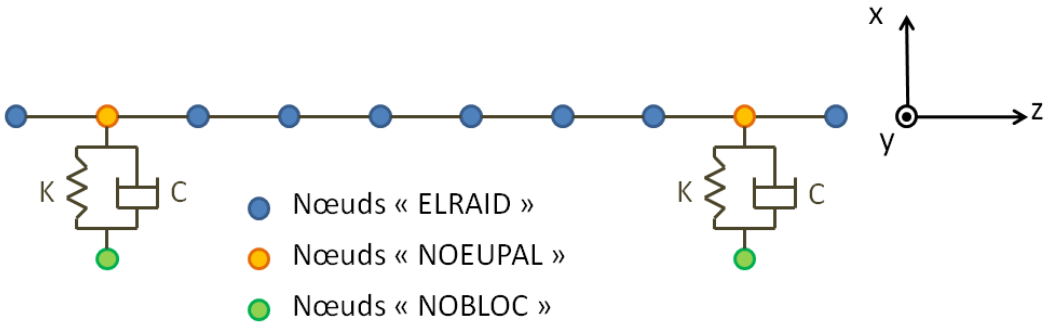

Figure 4.4.5-a : N œuds de la ligne d'arbres où sont appliquées les conditions aux limites

Les conditions aux limites libres-libres, appuyées-appuyées, ou réelles sont imposées par le bais de l'opérateur AFFE_CHAR_CINE. Dans le fichier de commande de *Code_Aster*, ces groupes de nœuds sont bloqués différemment pour chacune de ces conditions aux limites :

En Appuyé/Appuyé :

```
CLI_APP = AFFE_CHAR_CINE(MODELE=MOD_ROT,
                  MECA_IMPO=(_F(GROUP_NO='ELERAID', DZ=0., DRZ=0.,),
                             _F(GROUP_NO='NOEUPAL', DX=0., DY=0., DZ=0.,),
                             _F(GROUP_NO='NOBLOC',  DX=0., DY=0., DZ=0.,),
                             ),);
```

En Libre/Libre :

```
CLI_LIB = AFFE_CHAR_CINE(MODELE=MOD_ROT,
                         MECA_IMPO=(_F(GROUP_NO='ELERAID', DZ=0., DRZ=0.,),
                                    _F(GROUP_NO='NOEUPAL', DZ=0., ),
                                    _F(GROUP_NO='NOBLOC',  DZ=0., ),
                                    ),);
```

Pour les conditions aux limites réelles :

```
CLI_RE = AFFE_CHAR_CINE(MODELE=MOD_ROT,
                  MECA_IMPO=(_F(GROUP_NO='ELERAID', DZ=0., DRZ=0.,),
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                _F(GROUP_NO='NOEUPAL', DZ=0.,),
                                _F(GROUP_NO='NOBLOC',  DX=0., DY=0., DZ=0.,),
                                ),);
```

A ce niveau, la base modale tenant compte de la présence d'une fissure peut être calculée en combinant plusieurs configurations de « conditions aux limites » et de « géométrie » (Figure 4.4.5-b).

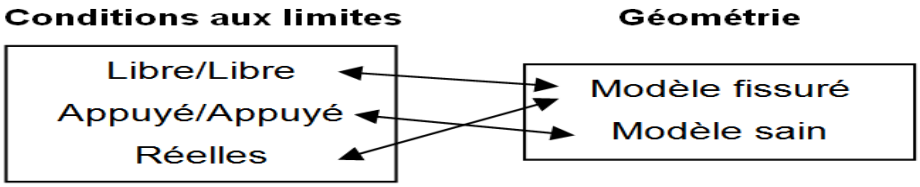

Figure 4.4.5-b : Différentes configuration de calcul de la base modale

Par exemple, cette base modale peut être créée en combinant les 2 bases suivantes :

- Une base avec des conditions aux limites **Libre/Libre** sur la ligne d'arbre **fissurée**. Nous calculons un nombre de modes égal à deux fois le nombre de paliers (pour tenir compte des deux directions de déplacement radiaux des nœuds) plus deux fois le nombre de fissures.
- Une base avec des conditions aux limites **Appuyé/Appuyé** (appuis simples sur tous les paliers) au droit des paliers sur la ligne d'arbre **saine**. Les modes calculés sont ceux ayant une fréquence inférieure à une valeur fixée par l'utilisateur.

Tous ces modes sont des modes conservatifs (ie. ne tenant pas compte des matrices d'amortissement gyroscopique ou non) et sont calculés à l'aide de l'opérateur MODE_ITER_SIMULT.

Les deux bases modales **libre-libre** (structure **fissurée**) et **appuyé-appuyé** (structure **saine**) sont ensuite assemblées à l'aide de l'opérateur DEFI_BASE_MODALE en une base de type RITZ. La numérotation des DDL choisie pour l'assemblage est celle correspondant aux conditions aux limites réelles. Une base globale obtenue est nommée « BMOD » :

```
BMOD=DEFI_BASE_MODALE(RITZ=(_F(MODE_MECA = (base_lib),
                               NMAX_MODE = 2*(nb_pal+nb_fis),),
                            _F(MODE_INTF = base_app,
                               NMAX_MODE = 1000),),
                            NUME_REF=nddl_re,
                            INFO=2,);
```

Dans un dernier temps, toutes les matrices réelles (masse, raideur, amortissement et gyroscopie) sont projetées via l'opérateur PROJ_MATR_BASE sur la base complète de RITZ (base BMOD). Pour le calcul des modes propres, le calcul modal final est réalisé à partir de ces matrices projetées, dites généralisées, à l'aide de l'algorithme de résolution de type QZ.

4.4.6 Réalisation d'un calcul transitoire

Dans un premier temps, il faut déterminer le coefficient multiplicateur de la loi de raideur qui tient compte des dimensions réelles du rotor. Ce coefficient est donné par l'équation suivante :

$$COEFF = \frac{E * I}{2 * L}$$

avec $I = \frac{\pi * D^4}{64}$ et $L = 4 * D$

E étant le module de Young du matériau, D le diamètre de l'arbre, L sa longueur et I son moment quadratique en flexion.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
YOUNG = 208.e9;
DIAM = 1.078;
LONG = (2.0 * DIAM);
I = (((((pi * DIAM) * DIAM) * DIAM) * DIAM) / 64.0);
COEFF = YOUNG*I/2/LONG
```

Les lois d'évolution de la raideur et de sa dérivée sont ensuite lues et multipliées par ce coefficient géométrique :

Loi de raideur :

```
F_K1=LIRE_FONCTION(UNITE=29,
                   NOM_PARA='ABSC',
                   INDIC_PARA=[1,1],
                   INDIC_RESU=[1,2],
                   PROL_DROITE="LINEAIRE",
                   PROL_GAUCHE="LINEAIRE",
                   INTERPOL='LIN');
```

Dérivée de la raideur :

```
F_KP1=LIRE_FONCTION(UNITE=29,
                    NOM_PARA='ABSC',
                    INDIC_PARA=[1,1],
                    INDIC_RESU=[1,3],
                    PROL_DROITE="LINEAIRE",
                    PROL_GAUCHE="LINEAIRE",
                    INTERPOL='LIN')
```

Multiplication par le coefficient géométrique :

```
F_K= CALC_FONCTION(COMB=(_F(FONCTION=F_K1,  COEF=COEFF)))

F_KP=CALC_FONCTION(COMB=(_F(FONCTION=F_KP1, COEF=COEFF)))
```

La résolution du calcul transitoire tenant compte de la présence de la fissure se fait ensuite avec la commande DYNA_VIBRA :

```
TRANGEN=DYNA_VIBRA(MATR_MASS=MASGEN,
            MATR_RIGI=RIGGEN,
            MATR_AMOR=GYOGEN,
            BASE_CALCUL ='GENE',
            TYPE_CALCUL ='TRAN',
            SCHEMA_TEMPS=_F(SCHEMA='EULER'),
            INCREMENT=_F(PAS=pas,
                         INST_FIN=tfin),
            EXCIT=(_F(VECT_ASSE=vgen_po,
                      COEF_MULT=1.,
                     ),),
            ROTOR_FISS=_F(GROUP_NO_G = 'FISS_G',
                          GROUP_NO_D = 'FISS_D',
                          K_PHI=F_K,
                          VITE_ROTA=-1*OM,
                          DK_DPHI=F_KP,),
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                ARCHIVAGE=_F(PAS_ARCH=1000,));
```

La fissure est définie grâce au mot clé ROTOR_FISS, les lois de raideur (K_PHI) et de sa dérivée (DK_DPHI) sont appelées dans ce mot clé. Il faut également indiquer les 2 nœuds ('FISS_G' et 'FISS_D') délimitant la fissure.

5 Post-traitements pour les études « machines tournantes »

Pour un calcul modal, le *Code_Aster* permet de calculer les sensibilités modales au balourd ainsi que les coefficients de participation. Pour les calculs harmoniques, *Code_Aster* permet de tracer l'évolution de l'amplitude et de la phase d'un degré de liberté en fonction de la fréquence d'excitation, de tracer pour une fréquence d'excitation donnée la déformée du rotor et les ellipses trajectoires des nœuds. Il permet également de déterminer les caractéristiques de la trajectoire du nœud pour une fréquence d'excitation donnée.

5.1 Sensibilités modales au balourd

Il peut arriver qu'un mode (position de sa vitesse critique) se situe dans une zone estimée « à risque » sans que ce dernier ne pose de problème vibratoire particulier. En effet, lors de l'introduction de la notion d'amortissement, les racines de l'équation du comportement écrite sous forme matricielle ne sont plus purement complexes, et une partie réelle apparaît :

$$r_i = -\frac{\epsilon_i \omega_i}{\sqrt{1-\epsilon_i^2}} + j\,\omega_i$$

Où ϵ_i est l'amortissement réduit du mode et ω_i la fréquence propre du mode. On peut alors définir le facteur de sensibilité au balourd, M_i :

$$M_i = \frac{\left(\frac{\Omega}{\omega_i}\right)^2}{\sqrt{\left(1-\left(\frac{\Omega}{\omega_i}\right)^2\right)^2 + 4\,\epsilon_i^2\left(\frac{\Omega}{\omega_i}\right)^2}},$$

Où Ω est la vitesse de rotation de la ligne d'arbres.

Cette grandeur permet de déterminer si oui ou non le mode est critique. Ainsi, un mode parfaitement synchrone avec la vitesse de rotation peut s'avérer inoffensif pour la ligne d'arbres si son amortissement réduit possède une valeur proche de 1. C'est pourquoi, on considère que si la valeur du facteur de sensibilité est inférieure à un certain seuil, alors il n'y a pas de problème vibratoire particulier. Cette grandeur peut être calculée de la façon suivante à partir du concept MODE_MECA :

```
TABLE=RECU_TABLE(CO=MODES,
                 NOM_PARA=('NUME_MODE','FREQ','AMOR_REDUIT'),);
tab=TABLE.EXTR_TABLE()
fprop=tab.FREQ
amort=tab.AMOR_REDUIT
# Calcul pour le i ème mode
fp=fprop[i]
am=amort[i]
r=w/fp
Mn=r**2/sqrt((1.-r**2)**2.+(2.*am*r)**2)
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5.2 Coefficients de participation

Pour un mode donné, les coefficients de participation sont les contributions respectives des énergies de déformation du rotor et du support à l'énergie de déformation totale du système. L'énergie de déformation totale du mode k s'écrit :

$$E_k = \Phi_k^t K \Phi_k$$

En séparant les degrés de liberté relatifs au rotor et ceux relatifs au support, l'énergie s'exprime sous la forme :

$$E_k = \begin{bmatrix} \Phi_{kr}^t & \Phi_{ks}^t \end{bmatrix} \left(\begin{bmatrix} K_r & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} K_{rr}(\Omega) & K_{rs}(\Omega) \\ K_{sr}(\Omega) & K_{ss}(\Omega) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_s \end{bmatrix} \right) \begin{bmatrix} \Phi_{kr} \\ \Phi_{ks} \end{bmatrix}$$

Si l'on note E_{kr} et E_{ks} les énergies du rotor et du support, définies par :

$$E_{kr} = \Phi_{kr}^t K_r \Phi_{kr} \text{ et } E_{ks} = \Phi_{ks}^t K_s \Phi_{ks}$$

Les coefficients de participation relatifs au rotor et support sont définis respectivement par :

$$\rho_{kr} = \frac{E_{kr}}{E_k} \quad \text{et} \quad \rho_{ks} = \frac{E_{ks}}{E_k}$$

L'opérateur POST_ELEM permet d'accéder directement aux énergies potentielle, cinétique et de déformation élastique sur tout le système mécanique (option TOUT='OUI') ou des parties du système (option GROUP_MA='ROTOR' ou GROUP_MA='MASSIF'). L'exemple suivant permet de calculer ces énergies pour les 4 premiers modes de la ligne d'arbres.

```
EPOT_TOT=POST_ELEM(RESULTAT=MODES,
                   NUME_MODE=(1,2,3,4),
                   MODELE=MODELE,
                   CARA_ELEM=CARELEM,
                   CHAM_MATER=CHMAT,
                   ENER_POT=_F(TOUT='OUI'),
                );
epot_tot=EPOT_TOT.EXTR_TABLE()
tot_epot=epot_tot.TOTALE

EPOT_ROT=POST_ELEM(RESULTAT=MODES,
                   NUME_MODE=(1,2,3,4),
                   MODELE=MODELE,
                   CARA_ELEM=CARELEM,
                   CHAM_MATER=CHMAT,
                   ENER_POT=_F(GROUP_MA='ROTOR'),
                );
epot_rot=EPOT_ROT.EXTR_TABLE()
rot_epot=epot_rot.TOTALE

EPOT_MAS=POST_ELEM(RESULTAT=MODES,
                   NUME_MODE=(1,2,3,4),
                   MODELE=MODELE,
                   CARA_ELEM=CARELEM,
                   CHAM_MATER=CHMAT,
                   ENER_POT=_F(GROUP_MA='MASSIF'),
             );
epot_mas=EPOT_MAS.EXTR_TABLE()
mas_epot=epot_mas.TOTALE
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

L'obtention des coefficients de participation peut se faire avec Python de la même façon que pour les sensibilités modales au balourd M_n (cf. paragraphe 7.1).

```
# Calcul des participations du rotor et massif pour le premier mode
tot_pot1=tot_epot[1]
rot_pot1=rot_epot[1]
mas_pot1=mas_epot[1]
part_rot=rot_pot1/tot_pot1
part_mas=mas_pot1/tot_pot1
print 'part_rot', part_rot
print 'part_mas', part_mas
```

5.3 Détermination des phases et amplitudes en harmonique

La déformée (modale, réponse harmonique, etc.) est caractérisée par des déplacements complexes. Ce caractère complexe des déplacements correspond à l'introduction de déphasage entre les déplacements des nœuds du modèle. Ce déphasage est dû à l'amortissement et aux effets gyroscopiques liés à la rotation du rotor. À partir du concept résultat, les déplacements latéraux sont récupérés sous forme de fonctions comme suit :

```
DY_DIS2=RECU_FONCTION(RESULTAT=DHAM,
                      NOM_CHAM='DEPL',
                      NOM_CMP='DY',
                      GROUP_NO='N_DIS2',);

DZ_DIS2=RECU_FONCTION(RESULTAT=DHAM,
                      NOM_CHAM='DEPL',
                      NOM_CMP='DZ',
                      GROUP_NO='N_DIS2' ,);
```

Les amplitudes et les phases de déplacement suivant les axes Y et Z sont obtenues à partir des déplacements modaux complexes dans les mêmes directions comme suit :

```
MOD_Y_D2=CALC_FONCTION(EXTRACTION=_F(FONCTION=DY_DIS2,
                                     PARTIE='MODULE',),);

MOD_Z_D2=CALC_FONCTION(EXTRACTION=_F(FONCTION=DZ_DIS2,
                                     PARTIE='MODULE',),);

PHA_Y_D2=CALC_FONCTION(EXTRACTION=_F(FONCTION=DY_DIS2,
                                     PARTIE='PHASE',),);

PHA_Z_D2=CALC_FONCTION(EXTRACTION=_F(FONCTION=DZ_DIS2,
                                     PARTIE='PHASE',),);
```

Pour la commodité de mise en œuvre des opérations ultérieures de post-traitement, ces grandeurs sont stockées sous format de tables.

```
TABMODY=CREA_TABLE(FONCTION=_F(FONCTION=MOD_Y_D2, ))
TABMODZ=CREA_TABLE(FONCTION=_F(FONCTION=MOD_Z_D2, ))
TABPHAY=CREA_TABLE(FONCTION=_F(FONCTION=PHA_Y_D2, ))
TABPHAZ=CREA_TABLE(FONCTION=_F(FONCTION=PHA_Z_D2, ))

MODYDIS2=TABMODZ['DY',1]
MODZDIS2=TABMODY['DZ',1]
PHAYDIS2=TABPHAZ['DY',1]
```


Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
PHAZDIS2=TABPHAY['DZ',1]
```

5.4 Détermination des ellipses des rotors

Comme pour les modes propres, La trajectoire d'un nœud dans le plan perpendiculaire à la fibre neutre du rotor est donc une ellipse dont les caractéristiques sont définies de la manière suivante :

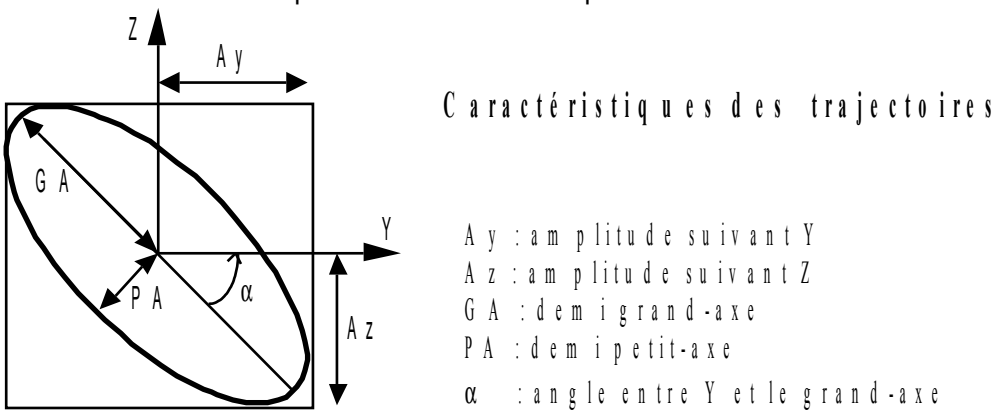

Figure 5.4-a : Illustration de la trajectoire d'un mode

Ensuite, on doit déterminer le grand-axe de la trajectoire elliptique. D'un point de vue vibratoire, c'est la valeur du demi grand-axe qui représente le niveau (crête) vibratoire à prendre en compte pour le dimensionnement ou la surveillance.

```
dax=ellipse(MODYDIS2,MODZDIS2,PHAYDIS2,PHAZDIS2)
```

La fonction ellipse est définie comme suit :

```
def ellipse(mody,modz,thety,thetz) :
    thety=thety*pi/180.
    thetz=thetz*pi/180.
    num = mody**2*sin(2*thety) + modz**2*sin(2*thetz)
    denum = mody**2*cos(2*thety) + modz**2*cos(2*thetz)
    if abs(denum) < 1e-33 :
       maxdemiaxe = 0.
    else :
       T=.5*atan(-num/denum)
       uy1 = mody*cos(T+thety)
       uz1 = modz*cos(T+thetz)
       uy2 = mody*cos(T+thety+pi/2.)
       uz2 = modz*cos(T+thetz+pi/2.)
       ayz1 = sqrt(uy1**2+uz1**2)
       ayz2 = sqrt(uy2**2+uz2**2)
       maxdemiaxe = max(ayz1,ayz2)
    return maxdemiaxe
```

Le sens de parcours de la trajectoire est appelé sens de précession. Si le sens de parcours de la trajectoire elliptique correspond au sens de rotation du rotor, la précession est dite directe . Dans le cas contraire la précession est dite inverse ou rétrograde.
On suppose sans le démontrer que, pour un même mode, tous les nœuds ont le même sens de précession. On parle alors de mode direct ou rétrograde.
Dans *Code_Aster* , l'identification du sens de précession se fait soit en fonction du signe de la plus grande orbite (le nœud dont le grand axe est maximum) dans un mode, soit en fonction du signe de la somme des signes de toutes les orbites. Pour plus de renseignements sur le calcul du sens de précession, le lecteur pourra se référer au document [R7.10.03].

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5.5 Diagramme de Campbell

Le diagramme de Campbell est une représentation graphique permettant le suivi des fréquences naturelles d'un système en rotation en fonction de sa vitesse de rotation ainsi que les zones d'instabilité de ces modes ([R7.10.03]). Les fréquences naturelles et les modes d'un système tournant sont obtenus par la résolution de l'équation d'équilibre dynamique d'un système d'arbres tournants, sans second membre et incluant les effets d'amortissement gyroscopique.

A cet effet, deux macro-commandes sont développées dans *Code_Aster*. La première macro-commande, CALC_MODE_ROTATION [U4.42.51], permet le calcul des fréquences et des modes sur le système complet en fonctions des vitesses de rotation. La deuxième macro-commande, IMPR_DIAG_CAMPBELL [U4.52.52], permet de classer les modes en flexion, en torsion et en traction compression, de normer ces modes, de déterminer le sens de précession des modes en flexion, de réaliser le tri des fréquences suivant différentes méthodes de suivi de modes et, enfin, de tracer le diagramme de Campbell.

6 Références bibliographiques

1 M. LALANNE, G. FERRARIS, " Rotordynamics Prediction in Engineering ", Second Edition, Wiley, 2001.

2 *CODE_ASTER*, www.code-aster.org .

3 CADYRO, manuel d'utilisation.

4 ROTORINSA, logiciel éléments finis destiné à prévoir le comportement dynamique de rotors en flexion, LaMCoS UMR5259, INSA-Lyon.

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

7 Annexes

7.1 Tableau des cas-tests machines tournantes dans *Code_Aster*

Nom du test		SDLL123			SDLL124	SDLL125	SDLL126				SDLL140	SDNL133			
		a	b	c	a	a	a	b	c	d	a	a	b	c	d
Poutre suivant		x	x	x	x	x	x	x	x	z	x	x	y	x	
Élément	Euler	Oui	Oui	Oui											
	Timoshenko	Oui			Oui		Oui	Oui	Oui	Oui		Oui	Oui	Oui	Non
disque		1	1		3	2	3	3	3	3	3	1	1	1	
Palier nb	rigides	2	2												
	linéaires			2	2	2	2	2	2	2	2	1	1	1	
	symétriques	Oui	Oui	Non	Oui	Non	Oui	Oui	Oui	Oui	Oui	1	1	1	
support	rigides	2	2									2	2	2	
	simplifiés			2											
	généralisés				Non	Non	Non	Non	Non	Non	Non				
Fréq. propres	QZ										Oui				
	Sorensen										Oui	Oui	Oui	Oui	
	2 étages										Oui				
	Campbell				Oui	Oui									
harmonique	DYNA_LINE_HARM								Oui						
Transitoire	DYNA_LINE_TRAN						Oui								
	DYNA_TRAN_MODAL							Oui		Oui		Oui	Oui	Oui	
Fissure		Non			Non	Non	Non	Non	Non	Non	Non	Oui	Oui	Oui	
validation	Non régression	Oui	Oui	Oui											
	Source externe				Oui	Oui					Oui				
	Analytique														
	Autre Aster						Oui	Oui	Oui	Oui		Oui	Oui	Oui	
Remarques															Détermination de la loi de fissure

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Nom du test		SDLL141				SHLL102	SHLL103	SDLL127	SDLL128	SDLL129	SDLV132
		a	b	c	d	a	a	a	a	a	a
Poutre suivant		x	$x+y$	$x+y$	x	x	x	x	x	x	x
Élément	Euler	Oui	Oui		Oui						
	Timoshenko					Oui	Oui	Oui		Oui	Oui
disque				répartis		3	3		3	3	Non
Palier nb	rigides	2	2	2	2	2	2				
	linéaires							2	2	2	2
	symétriques					Oui	Non	Oui	Var	Var	Oui
support	rigides	2	2	2	2	2	2				
	simplifiés										
	généralisés					Non	Non	Non	Non	Non	Oui
Fréq. propres	QZ							Oui			
	Sorensen	Oui	Oui	Oui	Oui						
	2 étages										
	Campbell									Oui	
harmonique	DYNA_LINE_HARM					Oui	Oui				
Transitoire	DYNA_LINE_TRAN										
	DYNA_TRAN_MODAL										
Fissure		Non	Non	Non	Non	Non	Non	Non	Non	Non	Non
validation	Non régression						Oui	Oui	Oui		Oui
	Source externe					Oui	Oui	Oui		Oui	Oui
	Analytique	Oui	Oui	Oui	Oui						
	Autre Aster							Oui			Oui
Remarques					Fausse section variable			Section variable			

Copyright 2015 EDF R&D - Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)