Algorithmes d'intégration temporelle de l'opérateur DYNA_TRAN_MODAL

Résumé

Ce document décrit les schémas d'intégration temporelle qui sont utilisés pour résoudre dans l'espace des modes des problèmes de dynamique transitoire en mécanique linéaire, avec, pour certains schémas, la prise en compte possible de non linéarités localisées de type chocs, frottements, relations effort-déplacement et effort-vitesse, et l'utilisation possible de la sous-structuration. Les schémas de NEWMARK (implicite, limité ici au linéaire), EULER, DEVOGELAERE, et quatre schémas explicites à pas de temps adaptatif, ADAPT_ORDRE1 et ADAPT_ORDRE2 RUNGE_KUTTA_32 et RUNGE_KUTTA_54 sont présentés.

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

Table des Matières

1 Introduction....3
2 Méthodes d'intégration temporelle d'un problème dynamique....3
2.1 Introduction....3
2.2 Méthode d'intégration implicite....5
2.2.1 Introduction....5
2.2.2 La méthode de NEWMARK [bib1]....5
2.2.2.1 Présentation du schéma....5
2.2.2.2 Algorithme complet de la méthode de NEWMARK....6
2.2.2.3 Conditions de stabilité du schéma de NEWMARK....6
2.2.2.4 Emploi....7
2.2.2.5 Amortissement numérique des schémas implicites....7
2.2.3 Algorithme TRBDF2....7
2.2.3.1 Présentation....7
2.2.3.2 Algorithme complet....8
2.2.3.3 Conditions de stabilité....9
2.3 Méthodes d'intégration explicites....9
2.3.1 Introduction....9
2.3.2 Schéma explicite d'Euler modifié d'ordre 1....9
2.3.2.1 Présentation....9
2.3.2.2 Ordre et stabilité du schéma....10
2.3.3 Méthode de Devogelaere-Fu....10
2.3.3.1 Présentation....10
2.3.3.2 Ordre et stabilité du schéma....11
2.3.3.3 Emploi....11
2.3.4 Schémas d'intégration à pas de temps adaptatif....11
2.3.4.1 Introduction : intérêt d'un pas de temps adaptatif....11
2.3.4.2 Les schémas à pas adaptatifs ADAPT_ORDRE1 et ADAPT_ORDRE2…12
2.3.4.2.1 Schéma des différences centrées à pas constant....12
2.3.4.2.2 Adaptation du schéma au pas de temps variable....13
2.3.4.2.3 Stabilité et précision du schéma....13
2.3.4.2.4 Critères d'adaptation du pas de temps....14
2.3.4.2.5 Algorithme du schéma des différences centrées à pas adaptatif....16
2.3.4.2.6 Commentaires sur les paramètres de l'algorithme....17
2.3.4.2.7 Performance de l'algorithme....17
2.3.4.3 Schémas à pas de temps adaptatifs de la famille de Runge-Kutta…....17
2.3.4.3.1 Schémas de Runge-Kutta emboîtés pour le contrôle du pas de temps adaptatif....18
3 Conclusion....21
4 Bibliographie....22
5 Description des versions du document....23

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

1 Introduction

Le but de l'analyse dynamique transitoire est de déterminer en fonction du temps la réponse d'une structure, étant donné un chargement extérieur ou des conditions aux limites fonctions du temps, dans des cas où les effets d'inertie ne peuvent être négligés.

Dans un certain nombre de configurations physiques, on ne peut se contenter d'une analyse modale ou harmonique et l'on doit réaliser une analyse transitoire. C'est notamment le cas si :

- l'histoire du phénomène a une importance dans l'étude,
- si le chargement extérieur est complexe (séisme, excitations multi-composantes, etc ...),
- si le système est non linéaire (plasticité, chocs, frottements, etc ...).

Les méthodes d'analyse transitoire qui peuvent être alors utilisées se divisent en deux grandes catégories :

- les méthodes dites d'intégration directe,
- les méthodes de Ritz, qui comprennent entre autres la recombinaison de projections modales.

Les méthodes d'intégration directe sont dénommées ainsi car aucune transformation n'est effectuée sur le système dynamique après la discrétisation par éléments finis. Elles sont présentées dans le document [R5.05.02], algorithmes d'intégration directe de l'opérateur DYNA_LINE_TRAN.
Les méthodes de Ritz, en revanche, procèdent à une transformation du système dynamique initial, par une projection sur un sous-espace de l'espace de discrétisation départ. La résolution se fait alors sur un système modifié, qui, s'il est réduit, ne permet d'accéder qu'à une approximation de la réponse du système réel.

Des algorithmes d'intégration temporelle sur un système en coordonnées généralisées sont utilisés pour résoudre les problèmes dynamiques en mécanique pour des structures linéaires, avec prise en compte éventuelle des non linéarités localisées telles les chocs, les frottements, et les relations effort-déplacement et effort-vitesse. Certains algorithmes permettent en outre la sous-structuration.
Ces algorithmes sont programmés dans l'opérateur DYNA_TRAN_MODAL de *Code_Aster* [U4.53.21].

2 Méthodes d'intégration temporelle d'un problème dynamique

2.1 Introduction

On suppose que les équations régissant l'équilibre dynamique des solides ont été discrétisées par éléments finis. On obtient un système discret d'équations qu'il s'agit d'intégrer dans le temps. Pour cela on choisit une discrétisation $\{t_i, i \in \mathbb{N}\}$ de l'intervalle de temps de l'étude $[0, T]$ et on écrit l'équilibre de la structure aux instants t_i .

De façon générale ces équations prennent la forme suivante :

$$\mathbf{M}.\ddot{X}_t + \mathbf{C}.\dot{X}_t + \mathbf{K}.X_t = \mathbf{R}_{ext}(t) + \mathbf{R}_{nl}(X_t, \dot{X}_t, \ddot{X}_t) \qquad \textbf{éq 2.1-1}$$

où

- $\mathbf{M}$ est la matrice de masse du système,
- $\mathbf{K}$ est la matrice de rigidité du système,
- $\mathbf{C}$ est la matrice d'amortissement du système,
- $\mathbf{R}_{ext}(t)$ est le vecteur des forces extérieures,
- $\mathbf{R}_{nl}(X_t, \dot{X}_t, \ddot{X}_t)$ est le vecteur des forces non linéaires.

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

La matrice d'amortissement $\mathbf{C}$ est en général difficile à évaluer car l'amortissement est souvent fonction de la fréquence. Il est toutefois fréquent de simplifier la prise en compte de l'amortissement en employant le modèle d'amortissement proportionnel, ou modèle de Rayleigh.

Les méthodes de réduction de Rayleigh-Ritz sont présentées dans le document [R5.06.01], Méthodes de Ritz en dynamique linéaire et non linéaire.

Dans le cas où le terme $\mathbf{R}_{nl}\left(\boldsymbol{X}_t, \dot{\boldsymbol{X}}_t, \ddot{\boldsymbol{X}}_t\right)$ n'est pas nul, la technique des pseudo-forces consiste à projeter sur la base du système linéaire et à maintenir les forces non linéaires au second membre. La technique des pseudo-forces est toujours associée à un schéma d'intégration explicite. De ce fait la prise en compte des non linéarités n'est disponible que pour des schémas explicites. L'ajout des non linéarités ne modifie pas la forme des équations.

Dans la méthode de Ritz, le champ de déplacement $\boldsymbol{X}_t$ est remplacé par sa projection sur la base modale telle que $\boldsymbol{X}_t = \Phi \eta_t$ où η_t est le vecteur des coordonnées généralisées et Φ est la base modale, le plus souvent réduite.

Le système dynamique projeté prend la forme suivante, avec $\eta_t \in \mathbb{R}^p$:

$$\Phi^t.\mathbf{M}.\Phi.\ddot{\eta}_t + \Phi^t.\mathbf{C}.\Phi.\dot{\eta}_t + \Phi^t.\mathbf{K}.\Phi.\eta_t = \Phi^t.\mathbf{R}_{ext}(t) + \Phi^t.\mathbf{R}_{nl}\left(\Phi.\eta_t, \Phi.\dot{\eta}_t, \Phi.\ddot{\eta}_t\right) \qquad \textbf{éq 2.1-2}$$

Lorsque l'hypothèse de Basile ne s'applique pas (amortissement non proportionnel), la matrice d'amortissement projetée n'est pas diagonale. L'intégration du système couplé se fait alors obligatoirement avec l'un des trois schémas suivants : le schéma implicite NEWMARK, le schéma explicite EULER ou les schémas explicites ADAPT_ORDRE1 et ADAPT_ORDRE2.

L'équation obtenue en η_t est de la même forme que l'équation en $\boldsymbol{X}_t$. De ce fait, dans la suite du document, on utilisera la notation $\boldsymbol{X}_t$ aussi bien pour le déplacement en coordonnées généralisées que pour le déplacement dans l'espace physique. Dans le cas de l'opérateur DYNA_TRAN_MODAL, il s'agit de coordonnées généralisées.

Deux classes de méthode peuvent être distinguées dans l'intégration pas à pas des équations d'équilibre, les méthodes d'intégration explicite et les méthodes d'intégration implicite.

Soit le système dynamique linéaire suivant à intégrer dans le temps :

$$\mathbf{M}.\ddot{\boldsymbol{X}}_t + \mathbf{C}.\dot{\boldsymbol{X}}_t + \mathbf{K}.\boldsymbol{X}_t = \mathbf{R}_{ext}(t) \qquad \textbf{éq 2.1-3}$$

Ce système différentiel du second ordre peut être ramené à un système du premier ordre :

$$\mathbf{N}.\dot{\boldsymbol{U}}_t = \mathbf{H}.\boldsymbol{U}_t + \mathbf{F}_t \qquad \textbf{éq 2.1-4}$$

où $\boldsymbol{U}_t = \begin{pmatrix} \boldsymbol{X}_t \\ \dot{\boldsymbol{X}}_t \end{pmatrix}$, $\mathbf{N} = \begin{pmatrix} \mathbf{Id} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{pmatrix}$, $\mathbf{H} = \begin{pmatrix} \mathbf{0} & \mathbf{Id} \\ -\mathbf{K} & -\mathbf{C} \end{pmatrix}$, $\mathbf{F}_t = \begin{pmatrix} \mathbf{0} \\ \mathbf{R}_t \end{pmatrix}$

Pour intégrer cet système d'équations différentielles, on utilise une discrétisation $\left\{t_i, i \in \mathbb{N}\right\}$, ainsi qu'une formule de différences finies pour exprimer la dérivée $\dot{\boldsymbol{U}}_t$.

On appellera méthodes d'intégration explicite les méthodes où seule la dérivée $\dot{\boldsymbol{U}}_t$ fait intervenir des inconnues au temps t_{i+1} . De cette façon la détermination des grandeurs cherchées à l'instant t_{i+1} ne résulte pas d'une inversion de système faisant intervenir l'opérateur $\mathbf{H}$. Si de plus, on réalise un « mass-lumping » de façon à rendre la matrice $\mathbf{M}$ diagonale, la détermination de $\boldsymbol{U}_t$ est particulièrement simple. Ce sont là les principales caractéristiques des méthodes d'intégration explicite.

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

Les méthodes implicites ou semi-implicites font intervenir la discrétisation de U_t à un instant postérieur à t_i, généralement t_{i+1}. La détermination des variables passe donc par la résolution d'un système faisant intervenir l'opérateur $\mathbf{H}$.

Deux notions sont importantes : la consistance, ou l'ordre du schéma d'intégration, et la stabilité.

Les approximations utilisées pour obtenir les opérateurs différentiels définissent la consistance, ou l'ordre du schéma d'intégration. On peut en effet considérer que l'approximation avec laquelle on obtient le déplacement à chaque pas de temps est liée à l'ordre d'approximation des dérivées premières et secondes par rapport au temps.

L'étude de stabilité d'un schéma consiste à analyser la propagation des perturbations numériques au cours du temps. Un schéma stable conserve une solution finie, malgré les perturbations, alors qu'un schéma instable conduit à une explosion numérique ou divergence de la solution.

Pour réaliser une étude de stabilité d'un schéma d'intégration, on met ce dernier sous la forme d'un système récursif linéaire et on détermine les caractéristiques propres de ce système. Si toutes les valeurs propres de l'opérateur de récursivité sont plus petites que 1 en module, le schéma est stable. Sinon il est instable.
Les schémas d'intégration explicite sont généralement conditionnellement stables, ce qui signifie que le pas de temps doit être suffisamment petit pour assurer la stabilité de l'intégration temporelle.
Certains algorithmes implicites ont la propriété d'être inconditionnellement stables, ce qui fait leur intérêt et permet d'utiliser un pas de temps arbitrairement grand.

Les schémas retenus pour l'opérateur DYNA_TRAN_MODAL sont un schéma implicite, NEWMARK, et quatre schémas explicites, EULER, DEVOGELAERE, ADAPT_ORDRE1 et ADAPT_ORDRE2 (à pas de temps adaptatif). Le choix se fait par le mot-clef METHODE : 'EULER', 'DEVOGE', 'NEWMARK', 'ADAPT_ORDRE1' ou 'ADAPT_ORDRE2'.

2.2 Méthode d'intégration implicite

2.2.1 Introduction

Les méthodes implicites font intervenir la résolution d'un système matriciel avec l'opérateur précédemment défini. Si les solides sont supposés élastiques linéaires, cela se traduit par la résolution d'un système linéaire à chaque pas de temps.

L'avantage de ces méthodes est leur stabilité inconditionnelle, qui leur permet d'intégrer les équations de la dynamique avec un pas de temps relativement important tout en représentant correctement le comportement des modes les plus bas en fréquence de la structure.
Un version implicite de la méthode de NEWMARK, qui a été programmée dans DYNA_TRAN_MODAL pour les problèmes linéaires.

2.2.2 La méthode de NEWMARK [bib1]

2.2.2.1 Présentation du schéma

NEWMARK [bib1] a introduit deux paramètres γ et β pour le calcul des positions et vitesses au pas de temps $t+\Delta t$:

$$\dot{\mathbf{X}}_{t+\Delta t}=\dot{\mathbf{X}}_t+\Delta t\left[(1-\gamma)\ddot{\mathbf{X}}_{\mathbf{t}}+\gamma\ddot{\mathbf{X}}_{t+\Delta t}\right]$$

$$\mathbf{X}_{t+\Delta t}=\mathbf{X}_t+\Delta t\,\dot{\mathbf{X}}_{\mathbf{t}}+\Delta t^2\left[(\frac{1}{2}-\beta)\ddot{\mathbf{X}}_{\mathbf{t}}+\beta\ddot{\mathbf{X}}_{t+\Delta t}\right]$$

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

Considérons les équations d'équilibre au temps $t+\Delta t$:

$$\mathbf{M}\cdot\ddot{\mathbf{X}}_{t+\Delta t}+\mathbf{C}\dot{\mathbf{X}}_{t+\Delta t}+\mathbf{K}\mathbf{X}_{t+\Delta t}=\mathbf{R}_{t+\Delta t}$$

Reportons les relations précédentes en éliminant $\dot{\mathbf{X}}_{t+\Delta t}$ et $\ddot{\mathbf{X}}_{t+\Delta t}$, il vient :

$$\tilde{\mathbf{K}}\cdot\mathbf{X}_{t+\Delta t}=\tilde{\mathbf{R}}_{t+\Delta t} \text{ où } \tilde{\mathbf{K}}=\mathbf{K}+a_0\mathbf{M}+a_1\mathbf{C}$$

$$\tilde{\mathbf{R}}=\mathbf{R}_{t+\Delta t}+\mathbf{C}\cdot\left(a_1\mathbf{X}_t+a_4\dot{\mathbf{X}}_t+a_5\ddot{\mathbf{X}}_t\right)+\mathbf{M}\cdot\left(a_0\mathbf{X}_t+a_2\dot{\mathbf{X}}_t+a_3\ddot{\mathbf{X}}_t\right)$$

avec :

$$a_0=\frac{1}{(\beta.\Delta t^2)} \quad a_1=\frac{\gamma}{(\beta.\Delta t)} \quad a_2=\frac{1}{(\beta.\Delta t)} \quad a_3=\frac{1}{2\beta}-1$$

$$a_4=\frac{\gamma}{\beta}-1 \quad a_5=\frac{\Delta t}{2}\left(\frac{\gamma}{\beta}-2\right) \quad a_6=\Delta t.(1-\gamma) \quad a_7=\gamma.\Delta t$$

2.2.2.2 Algorithme complet de la méthode de NEWMARK

a) initialisation

1) conditions initiales $\mathbf{X_0}$, $\dot{\mathbf{X}}_\mathbf{0}$ et $\ddot{\mathbf{X}}_\mathbf{0}$
2) choix de Δt γ et β et calcul des coefficients $a_{1,\ldots},a_8$ (cf ci-dessus)
3) assembler les matrices de raideur $\mathbf{K}$, de masse $\mathbf{M}$ et d'amortissement $\mathbf{C}$
4) former la matrice de rigidité effective $\tilde{\mathbf{K}}=\mathbf{K}+a_0\mathbf{M}+a_1\mathbf{C}$
5) factoriser $\tilde{\mathbf{K}}$

b) à chaque pas de temps

1) calculer le chargement effectif $\tilde{\mathbf{R}}$:
$$\tilde{\mathbf{R}}=\mathbf{R}_{t+\Delta t}+\mathbf{C}\cdot\left(a_1\mathbf{X}_t+a_4\dot{\mathbf{X}}_t+a_5\ddot{\mathbf{X}}_t\right)+\mathbf{M}\cdot\left(a_0\mathbf{X}_t+a_2\dot{\mathbf{X}}_t+a_3\ddot{\mathbf{X}}_t\right)$$
2) résoudre $\tilde{\mathbf{K}}\cdot\mathbf{X}_{t+\Delta t}=\tilde{\mathbf{R}}_{t+\Delta t}$
3) calculer les vitesses et accélérations au temps $t+\Delta t$
$$\ddot{\mathbf{X}}_{t+\Delta t}=a_0\left(\mathbf{X}_{t+\Delta t}-\mathbf{X}_t\right)-a_2\dot{\mathbf{X}}_t-a_3\ddot{\mathbf{X}}_t$$
$$\dot{\mathbf{X}}_{t+\Delta t}=\dot{\mathbf{X}}_t+a_6\ddot{\mathbf{X}}_t+a_7\ddot{\mathbf{X}}_{t+\Delta t}$$
4) calcul du pas de temps suivant : retour en b)1)

2.2.2.3 Conditions de stabilité du schéma de NEWMARK

La méthode de NEWMARK utilisée de façon assez répandue dans le domaine de la mécanique, car il permet de choisir l'ordre de l'intégration, d'introduire ou non de l'amortissement numérique, et possède une très bonne précision.

Il est inconditionnellement stable si : $\gamma>0.5$ et $\beta>\frac{(2\gamma+1)^2}{4}$

On introduit un amortissement numérique positif si $\gamma>\frac{1}{2}$ et négatif si $\gamma<\frac{1}{2}$.

Lorsque $\gamma=\frac{1}{2}$ et $\beta=0$, la formule de NEWMARK se réduit au schéma des différences centrées. C'est donc alors un schéma explicite.

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

Une combinaison très souvent employée est $\gamma=\frac{1}{2}$ et $\beta=\frac{1}{4}$, car elle conduit à un schéma d'ordre 2, inconditionnellement stable sans amortissement numérique. C'est le choix qui a été fait dans l'opérateur DYNA_TRAN_MODAL. Le schéma de Newmark de cet opérateur est donc implicite.

2.2.2.4 Emploi

Dans DYNA_VIBRA/BASE_CALCUL='GENE', ce schéma permet l'intégration de problèmes non-linéaires. Dans le cadre de la sous-structuration dynamique, il permet d'employer une base modale calculée par sous-structuration mais il ne supporte pas le calcul direct sur les bases modales des sous-structures.

2.2.2.5 Amortissement numérique des schémas implicites

L'avantage numérique des schémas d'intégration implicite directs réside dans le fait que le pas de temps peut être substantiellement grand par rapport à la plus petite période propre du système sans risquer de causer une instabilité des résultats.

Pour des modes de période propre de l'ordre du pas de temps ou inférieure au pas de temps, les algorithmes d'intégration introduisent un fort amortissement qui contribue à effacer la contribution de des modes élevés (cf [R5.05.02]).

Il n'y pas d'amortissement numérique dans le cas particulier de l'algorithme de NEWMARK avec $\beta=\frac{1}{4}$ et $\gamma=\frac{1}{2}$.

En revanche, les algorithmes implicites on un effet sensible d'allongement des périodes de la réponse de la structure. On constate que pour garantir une bonne précision sur l'amplitude et la phase des déplacements calculés, il faut respecter un critère voisin de :

$$\Delta t<\frac{1}{(10\times F_{max})} \text{ à } \frac{1}{(100\times F_{max})}$$

où $F_{\max}$ est la plus haute fréquence du mouvement que l'on souhaite capturer.

2.2.3 Algorithme TRBDF2

2.2.3.1 Présentation

Cette section traite le schéma d'intégration temporelle TRBDF2 pour Trapezoidal Rule (TR) Backward Differentiation Formula Order 2 (BDF2) et de son utilisation dans le cas de dynamique linéaire et non-linéaire des structures. Nous suivons dans ce qui suit la présentation de [GARDNER2021].

TRBDF2 est un schéma d'intégration temporel en deux étapes. Il a initialement été proposé pour la simulation numérique des microprocesseurs [BANK1985], considérés dans la littérature comme des problèmes raides. Pour illustrer son fonctionnement, on considère la forme canonique :

$$y'=f(y,t)$$

Ainsi pour intégrer cette équation entre $t=t_n$ et $t+\Delta t=t_{n+1}$, on enchaîne les 2 séquences suivantes :

- Tout d'abord, on utilise la règle du trapèze (TR) pour passer de t_n à $t_{n+\gamma}=t+\gamma\Delta t$, où $\gamma\in]0,1[$. Cela revient à résoudre l'équation suivante où $f_n=f(y_n,t_n)$:

$$y_{n+\gamma}-\gamma\frac{\Delta t}{2}f_{n+\gamma}=y_n+\gamma\frac{\Delta t}{2}f_n$$

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

- On utilise ensuite le schéma Backward Differentiation Formula d'ordre 2 (BDF2) pour passer de $t=t_n$ et $t+\Delta t=t_{n+1}$ en utilisant la solution précédente à $t_{n+\gamma}$:

$$y_{n+1}-\frac{1-\gamma}{2-\gamma}\Delta t f_{n+1}=\frac{1}{\gamma(2-\gamma)}y_{n+\gamma}-\frac{(1-\gamma)^2}{\gamma(2-\gamma)}y_n$$

Ces étapes sont représentées sur la Figure 2.2.3.1-1.

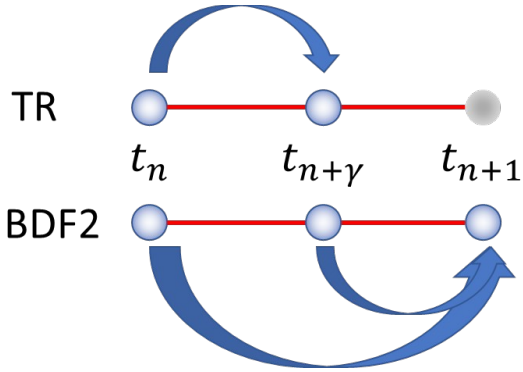


Figure 2.2.3.1-1 Représentation du schéma TRBDF2

2.2.3.2 Algorithme complet

a) initialisation

1) conditions initiales $\mathbf{X_0}$, $\dot{\mathbf{X}}_0$ et $\ddot{\mathbf{X}}_0$
2) assembler les matrices de raideur $\mathbf{K}$, de masse $\mathbf{M}$ et d'amortissement $\mathbf{C}$
3) calculer les paramètres $a_1=\frac{4}{(\gamma\Delta t)^2}$ $a_2=\frac{2}{\gamma\Delta t}$ $\gamma_1=\frac{1}{\gamma(2-\gamma)}$ $\gamma_2=\frac{-(1-\gamma)^2}{\gamma(2-\gamma)}$
$\gamma_3=\frac{1-\gamma}{2-\gamma}$ $\gamma=2-\sqrt{(2)}$
4) former la matrice de rigidité effective $\tilde{\mathbf{K}}=\left(\mathbf{K}+a_2\mathbf{C}+a_1\mathbf{M}\right)$
5) factoriser $\tilde{\mathbf{K}}$

b) à chaque pas de temps

 # 1ère étape par la méthode du trapèze $t\rightarrow t+\gamma\Delta t$

1) calculer une prédiction de l'accélération au temps $t+\gamma\Delta t$
$$\tilde{\ddot{\mathbf{X}}}_{t+\gamma\Delta t}=\mathbf{M}^{-1}\left(F^{ext}(t+\gamma\Delta t)-\mathbf{C}\dot{\mathbf{X}}_t-\mathbf{K}\mathbf{X}_t\right)$$
2) calculer le chargement effectif $\tilde{\mathbf{R}}_{t+\gamma\Delta t}$:
$$\tilde{\mathbf{R}}_{t+\gamma\Delta t}=\gamma\Delta t\left(\left(a_1\mathbf{M}+a_2\mathbf{C}\right)\dot{\mathbf{X}}_t+\frac{1}{2}a_2\mathbf{M}\left(\ddot{\mathbf{X}}_t+\tilde{\ddot{\mathbf{X}}}_{t+\gamma\Delta t}\right)\right)$$
3) résoudre $\tilde{\mathbf{K}}\cdot\delta\mathbf{X}_{t+\gamma\Delta t}=\tilde{\mathbf{R}}_{t+\gamma\Delta t}$
4) calculer les déplacements, vitesses au temps $t+\gamma\Delta t$
$$\mathbf{X}_{t+\gamma\Delta t}=\mathbf{X}_t+\delta\mathbf{X}_{t+\gamma\Delta t}$$
$$\dot{\mathbf{X}}_{t+\gamma\Delta t}=\dot{\mathbf{X}}_t+2\left(\frac{1}{\gamma\Delta t}\delta\mathbf{X}_{t+\gamma\Delta t}-\dot{\mathbf{X}}_t\right)$$

 # 2 ème étape par la méthode BDF2 $t\rightarrow t+\Delta t$

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

5) calculer une prédiction de l'accélération au temps $t+\Delta t$
$$\tilde{\ddot{\mathbf{X}}}_{t+\Delta t}=\mathbf{M}^{-1}\left(F^{ext}(t+\Delta t)-\mathbf{C}\dot{\mathbf{X}}_t-\mathbf{K}\mathbf{X}_t\right)$$

6) calculer le chargement effectif $\tilde{\mathbf{R}}_{t+\Delta t}$:
$$\begin{aligned}\tilde{\mathbf{R}}_{t+\Delta t} &=-a_2\mathbf{M}\left(\mathbf{X}_t-\gamma_1\mathbf{X}_{t+\gamma\Delta t}-\gamma_2\mathbf{X}_t-\gamma_3\dot{\mathbf{X}}_t\right)\\ &\quad-\left(a_1\mathbf{M}+a_2\mathbf{C}\right)\left(\dot{\mathbf{X}}_t-\gamma_1\dot{\mathbf{X}}_{t+\gamma\Delta t}-\gamma_2\dot{\mathbf{X}}_t-\gamma_3\tilde{\ddot{\mathbf{X}}}_t\right)\end{aligned}$$

7) résoudre $\tilde{\mathbf{K}}\cdot\delta\mathbf{X}_{t+\Delta t}=\tilde{\mathbf{R}}_{t+\Delta t}$

8) calculer les déplacements, vitesses et accélérations au temps $t+\Delta t$
$$\mathbf{X}_{t+\Delta t}=\mathbf{X}_t+\delta\mathbf{X}_{t+\Delta t}$$
$$\dot{\mathbf{X}}_{t+\Delta t}=\dot{\mathbf{X}}_t+\frac{1}{\gamma_3\Delta t}\left(\delta\mathbf{X}_{t+\Delta t}-\mathbf{X}_t+\gamma_1\mathbf{X}_{t+\gamma\Delta t}+\gamma_2\mathbf{X}_t+\gamma_3\dot{\mathbf{X}}_t\right)$$
$$\ddot{\mathbf{X}}_{t+\Delta t}=\mathbf{M}^{-1}\left(F^{ext}(t+\Delta t)-\mathbf{C}\dot{\mathbf{X}}_{t+\Delta t}-\mathbf{K}\mathbf{X}_{t+\Delta t}\right)$$

9) calcul du pas de temps suivant : retour en b)1)

Les détails d'obtention des ces expressions sont dans [TARDIEU2021].

2.2.3.3 Conditions de stabilité

Le schéma TRBDF2 est d'ordre 2, implicite et donc inconditionnellement stable. Par ailleurs, il est A-stable et L-stable.

2.3 Méthodes d'intégration explicites

2.3.1 Introduction

Plusieurs méthodes d'intégration explicite sont présentées : un schéma d'Euler modifié d'ordre 1, un schéma de Devogelaere-Fu d'ordre 4 et des schémas à pas de temps adaptatif ADAPT et Runge-Kutta. Ces trois méthodes sont disponibles dans l'opérateur DYNA_TRAN_MODAL. Les schémas sont présentés en ne considérant que des forces linéaires. Cependant la prise en compte des forces non linéaires s'en déduit facilement avec la technique des pseudo-forces.

2.3.2 Schéma explicite d'Euler modifié d'ordre 1

2.3.2.1 Présentation

Ce schéma est communément appelé « Euler modifié » car il s'agit d'une variante très simple mais conditionnellement stable du schéma d'Euler d'ordre 1, qui est, lui, instable. C'est donc un schéma souvent employé en explicite pour la mécanique. Dans *Code_Aster*, il est tout simplement appelé EULER.

Ce schéma fut utilisé dans le module TRANSIS de POUX [bib3], code éléments finis de poutre, et dans le code CADYRO [bib4] pour le calcul des lignes d'arbre en rotation.

Le schéma utilise le formule d'Euler d'ordre 1 pour estimer la dérivée en temps, avec une formule d'Euler avant pour la vitesse et une formule d'Euler arrière pour le déplacement, comme suit :

$$\dot{\mathbf{X}}_{n+1}=\dot{\mathbf{X}}_n+\Delta t\,\mathbf{M}^{-1}\left(\mathbf{R}-\mathbf{K}\mathbf{X}_n-\mathbf{C}\dot{\mathbf{X}}_n\right)+o(\Delta t)$$
$$X_{n+1}=X_n+\Delta t\,\dot{X}_{n+1}+o(\Delta t)$$

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

L'algorithme est donc le suivant :

a) initialisation : $X_0, \dot{X}_0$ donnés

b) à chaque pas de temps :

$$\dot{\boldsymbol{X}}_{n+1} = \dot{\boldsymbol{X}}_{\boldsymbol{n}} + \Delta t \,.\, \mathbf{M}^{-1} \,.\, \left(\mathbf{R} - \mathbf{K} \,.\, \boldsymbol{X}_{\boldsymbol{n}} - \mathbf{C} \,.\, \dot{\boldsymbol{X}}_{n} \right)$$

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_{\boldsymbol{n}} + \Delta t \,.\, \dot{\boldsymbol{X}}_{n+1}$$

2.3.2.2 Ordre et stabilité du schéma

les approximations utilisées dans l'obtention de ce schéma sont d'ordre 1. On peut donc considérer que l'approximation avec laquelle on obtient le déplacement à chaque pas de temps est d'ordre 1. Il s'agit de la consistance du schéma.

Si l'on met le schéma d'intégration sous forme récursive en éliminant les termes de vitesse, on obtient la relation de récurrence suivante (sans force extérieure, ni amortissement) :

$$\boldsymbol{X}_{n+1} + \left(\mathbf{M}^{-1} \,.\, \mathbf{K} \,.\, \Delta t^2 - 2 \right) \,.\, \boldsymbol{X}_{\boldsymbol{n}} + \boldsymbol{X}_{n-1} = \mathbf{0}$$

Les valeur propres de ce schéma sont pour un système à un degré de liberté :

$$\lambda = \frac{2 - \mathbf{M}^{-1} \boldsymbol{K} \, \Delta t^2 \pm i \sqrt{\mathbf{M}^{-1} \,.\, \mathbf{K} \,.\, \Delta t^2 \left(4 - \mathbf{M}^{-1} \,.\, \mathbf{K} \,.\, \Delta t^2 \right)}}{2} \text{ si } \Delta t < \frac{2}{\mathbf{M}^{-1} \,.\, \mathbf{K}}$$

Le module des valeurs propres λ vaut 1. On se rend compte que l'on est dans une situation limite mais favorable. Il n'y aura pas augmentation incontrôlée de l'erreur. Sans amortissement, on est juste à la borne de stabilité. Ce peut être un atout pour le schéma : il n'introduit pas de dissipation numérique.

Si $\Delta t < \frac{2}{\mathbf{M}^{-1} \,.\, \mathbf{K}}$ on peut montrer qu'une des deux valeurs propres a un module plus grand que l'unité et donc que le schéma est instable.

Le critère de stabilité du schéma EULER est donc : $\Delta t < \frac{2}{\mathbf{M}^{-1} \,.\, \mathbf{K}}$.

Cette étude peut être étendue à un système à nombre fini de degrés de liberté. Dans ce cas, le critère de stabilité devient :

$$\Delta t < \frac{2}{\omega_{\max}}$$

L'analyse peut être affinée en considérant un système avec amortissement [bib13].

2.3.3 Méthode de Devogelaere-Fu

2.3.3.1 Présentation

Pour présenter l'algorithme de Devogelaere-Fu, abrégé en DEVOGE dans *Code_Aster*, on met le problème dynamique sous la forme :

$\mathbf{M} \,.\, \ddot{\boldsymbol{X}}_t + \mathbf{C} \,.\, \dot{\boldsymbol{X}}_t = \mathbf{G}\left(t \,,\, \boldsymbol{X}_t \right)$ où la matrice $\mathbf{C}$ est supposée diagonale.

Les déplacements et les vitesses sont calculés comme suit :

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

a) initialisation

$$\boldsymbol{X}_{-\frac{1}{2}} = \boldsymbol{X}_0 - \frac{\Delta t}{2}\dot{\boldsymbol{X}}_0 + \frac{\Delta t^2}{8}\left(\mathbf{M}^{-1}.\mathbf{G}\left(t_0, \boldsymbol{X}_0\right) - \mathbf{M}^{-1}.\mathbf{C}.\dot{\boldsymbol{X}}_0\right)$$

$$\dot{\boldsymbol{X}}_{-\frac{1}{2}} = \left(4\boldsymbol{I} - \Delta t.\mathbf{M}^{-1}.\mathbf{C}\right)^{-1}\left(\left(4\boldsymbol{I} + \Delta t.\mathbf{M}^{-1}.\mathbf{C}\right).\dot{\boldsymbol{X}}_0 - \Delta t\left(\mathbf{G}\left(t_{-\frac{1}{2}}, \boldsymbol{X}_{-\frac{1}{2}}\right) + \mathbf{G}\left(t_0, \boldsymbol{X}_0\right)\right)\right)$$

b) à chaque pas de temps

$$\boldsymbol{X}_{n+\frac{1}{2}} = \boldsymbol{X}_n + \frac{\Delta t}{2}\dot{\boldsymbol{X}}_n + \frac{\Delta t^2}{24}\left(4\mathbf{M}^{-1}.\mathbf{G}\left(t_n, \boldsymbol{X}_n\right) - \mathbf{M}^{-1}.\mathbf{G}\left(t_{n-\frac{1}{2}}, \boldsymbol{X}_{\boldsymbol{n-\frac{1}{2}}}\right) - \mathbf{M}^{-1}.\mathbf{C}\left(4\dot{\boldsymbol{X}}_n - \dot{\boldsymbol{X}}_{n-\frac{1}{2}}\right)\right)$$

$$\dot{\boldsymbol{X}}_{n+\frac{1}{2}} = 4\left(4\boldsymbol{I} + \Delta t.\mathbf{M}^{-1}.\mathbf{C}\right)^{-1}\left(\dot{\boldsymbol{X}}_n + \frac{\Delta t}{4}\left(\mathbf{G}\left(t_n, \boldsymbol{X}_n\right) + \mathbf{G}\left(t_{n+\frac{1}{2}}, \boldsymbol{X}_{n+\frac{1}{2}}\right) - \mathbf{M}^{-1}.\mathbf{C}.\dot{\boldsymbol{X}}_n\right)\right)$$

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_n + \frac{\Delta t}{2}\dot{\boldsymbol{X}}_n + \frac{\Delta t_2}{6}\left(4\mathbf{M}^{-1}.\mathbf{G}\left(t_n, \boldsymbol{X}_n\right) + 2\mathbf{M}^{-1}.\mathbf{G}\left(t_{n-\frac{1}{2}}, \boldsymbol{X}_{n-\frac{1}{2}}\right) - \mathbf{M}^{-1}.\mathbf{C}\left(\dot{\boldsymbol{X}}_n + 2\dot{\boldsymbol{X}}_{n+\frac{1}{2}}\right)\right)$$

$$\dot{\boldsymbol{X}}_{n+1} = 6\left(6\boldsymbol{I} + \Delta t.\mathbf{M}^{-1}.\mathbf{C}\right)^{-1}.\left(\dot{\boldsymbol{X}}_n + \frac{\Delta t}{6}\left(\mathbf{G}\left(t_{n+1}, \boldsymbol{X}_{n+1}\right) + 4\mathbf{G}\left(t_{n+\frac{1}{2}}, \boldsymbol{X}_{n+\frac{1}{2}}\right) + \mathbf{G}\left(t_n, \boldsymbol{X}_n\right) - \mathbf{M}^{-1}.\mathbf{C}.\left(4\dot{\boldsymbol{X}}_{n+\frac{1}{2}} + \dot{\boldsymbol{X}}_n\right)\right)\right)$$

2.3.3.2 Ordre et stabilité du schéma

Le schéma est d'ordre 4, les approximations dans l'écriture des dérivées temporelles étant en . $o\left(\Delta t^4\right)$ Il possède donc une excellente aptitude à l'intégration de solutions régulières. Son intérêt est en revanche moins manifeste si les fonctions à intégrer présentent des discontinuités (chocs, frottement, etc.)

On peut montrer que pour un système linéaire non amorti le pas de temps garantissant la stabilité vaut : $\Delta t < \frac{2\sqrt{2}}{\omega_{\max}}$.

2.3.3.3 Emploi

Cette méthode est coûteuse en temps de calcul car elle nécessite deux fois l'évaluation du vecteur des forces internes $\mathbf{G}$, opération particulièrement lourde. Par conséquent elle est peu utilisée en mécanique pour l'intégration directe. Par contre elle est employée par le CEA [bib2] dans le cas des systèmes projetés sur base modale.

Ce schéma permet la prise en compte de non linéarités localisées de type chocs et frottements.
Dans le cadre de la sous-structuration dynamique, il permet d'employer une base modale calculée par sous-structuration mais il ne supporte pas le calcul direct sur les bases modales des sous-structures.

2.3.4 Schémas d'intégration à pas de temps adaptatif

2.3.4.1 Introduction : intérêt d'un pas de temps adaptatif

Réaliser l'intégration temporelle du transitoire d'une structure dans une phase non linéaire pose toujours des problèmes quant au choix du pas temps. L'estimation de l'erreur est rarement accessible lors de l'intégration.

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

Les schémas d'intégration explicite obligent à respecter un pas de temps maximal pour ne pas diverger. Dans le cas de comportement non linéaire, ce pas ne peut être déterminé *a priori* et peut changer à chaque itération. Lorsque la rigidité varie très fortement, un pas de temps constant et très fin pour conserver la stabilité du schéma conduit à un nombre d'itérations très grand et à un temps de calcul considérable.

Plusieurs algorithmes d'intégration à pas de temps adaptatif ont donc été développé pour DYNA_TRAN_MODAL : ADAPT_ORDRE1, ADAPT_ORDRE2, RUNGE_KUTTA_32 et RUNGE_KUTTA54. Les deux premiers s'appuient sur le schéma de différences centrées, d'ordre 2 et sur le schéma d'Euler, d'ordre 1. Les deux derniers, ce sont des schémas de la famille de Runge-Kutta avec contrôle de l 'erreur. Dans la suite de ce chapitre on ne représentera pas l'algorithme du schéma adaptatif d'ordre 1 car il est calqué sur le schéma d'Euler. La gestion de l'adaptation du pas de temps est, quant à elle, la même que pour le schéma adaptatif d'ordre 2.

On peut remarquer que ce type de schéma a aussi été programmé dans DYNA_LINE_TRAN (cf. [R5.05.02]).

2.3.4.2 Les schémas à pas adaptatifs ADAPT_ORDRE1 et ADAPT_ORDRE2

2.3.4.2.1 Schéma des différences centrées à pas constant

On présente d'abord le schéma des différences centrées à pas constant sur lequel le schéma ADAPT_ORDRE2 se base. Il s'écrit ainsi :

$$\dot{X}_{n+\frac{1}{2}} = \dot{X}_{n-\frac{1}{2}} + \Delta t . \ddot{X}_n\left(t_n, X_n, \dot{X}_n\right) + o\left(\Delta t^2\right)$$
$$X_{n+1} = X_n + \Delta t \dot{X}_{n+\frac{1}{2}} + o\left(\Delta t^2\right)$$

avec les notations suivantes :

On constate que la vitesse est exprimée à des indices demi entiers de la discrétisation en temps alors que les déplacements et accélérations sont exprimés aux indices entiers. Ecrit de cette façon le schéma est d'ordre 2. Toutefois l'accélération n'est pas immédiatement calculable car la vitesse n'est connue qu'au demi pas précédent. Pour contourner cette difficulté, on peut utiliser plusieurs approximations de la vitesse au pas de temps entier.

- méthode 1 : supposer que $\ddot{X}_n\left(X_n, \dot{X}_n, t_n\right) \equiv \ddot{X}_x\left(X_n, \dot{X}_{n-\frac{1}{2}}, t_n\right)$ ce qui constitue une approximation valide si l'amortissement est suffisamment faible $\dot{X}_n = \dot{X}_{n-\frac{1}{2}} + o(1)$. Si l'amortissement est important, le schéma perd alors sa précision d'ordre 2.
- méthode 2 : utiliser une approximation d'ordre 1 pour la vitesse : ce qui permet de $\dot{X}_n = \dot{X}_{n-\frac{1}{2}} + \frac{\Delta t}{2}\ddot{X}_{n-1} + o(\Delta t)$ conserver l'ordre 2 du schéma.
- méthode 3 : utiliser un schéma de type prédicteur/correcteur :

prédicteur :
$$\begin{cases} \dot{X}_{n^p} = \dot{X}_{n-\frac{1}{2}} + \gamma \Delta t \ddot{X}_{n-1} \\ \ddot{X}_{n^p} = \ddot{X}\left(t_n, X_n, \dot{X}_{n^p}\right) \end{cases}$$

correcteur :
$$\begin{cases} \dot{X}_n = \dot{X}_{n-\frac{1}{2}} + \frac{\Delta t}{2}\left(\beta \ddot{X}_{n^p} + (1-\beta)\ddot{X}_{n-1}\right) \\ \ddot{X}_n = \ddot{X}\left(t_n, X_n, \dot{X}_n\right) \end{cases}$$

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

où α et β sont deux paramètre à choisir. Park et Underwood [bib6] rapportent qu'effectuer des itérations supplémentaires n'améliore pas de façon sensible la stabilité du schéma.

2.3.4.2.2 Adaptation du schéma au pas de temps variable

Lorsque le pas de temps varie, les expressions du paragraphe précédent ne sont plus valables, l'accélération $\ddot{\boldsymbol{X}}_{\boldsymbol{n}}$ n'étant plus nécessairement exprimée au centre de l'intervalle $\left[\dot{\boldsymbol{X}}_{n-\frac{1}{2}}, \dot{\boldsymbol{X}}_{n+\frac{1}{2}}\right]$, comme on le voit sur le schéma ci-dessous.
Pour tenir compte de ceci, la vitesse est calculée ainsi :

$$\dot{\boldsymbol{X}}_{n+\frac{1}{2}} = \dot{\boldsymbol{X}}_{n-\frac{1}{2}} + \frac{\Delta t_{n-1} + \Delta t_n}{2} \ddot{\boldsymbol{X}}_n$$

Le schéma ADAPT_ORDRE2 complet s'écrit alors ainsi :

1. estimation de $\dot{\boldsymbol{X}}_{\boldsymbol{x}}$ selon les méthodes 1, 2 ou 3
2. $\dot{\boldsymbol{X}}_{n+\frac{1}{2}} = \dot{\boldsymbol{X}}_{n-\frac{1}{2}} + \frac{\Delta t_{n-1} + \Delta t_n}{2} \ddot{\boldsymbol{X}}_{\boldsymbol{n}}\left(t_n, \boldsymbol{X}_n, \dot{\boldsymbol{X}}_n\right) + o\left(\Delta t^2\right)$
3. $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n + \Delta t \dot{\boldsymbol{X}}_{n+\frac{1}{2}} + o\left(\Delta t^2\right)$

L'ordre du schéma n'est plus rigoureusement égal à 2, le schéma ayant perdu son caractère centré.
Plus Δt_n et Δt_{n+1} sont différents, plus l'ordre du schéma tend vers 1. De fortes variations du pas de temps conduisent donc à une perte de précision.

Il est possible de trouver des expressions plus complexes, qui utilisent la vitesse ou l'accélération à l'itération précédente [bib7]. Toutefois la formule présentée ici donne des résultats satisfaisants lorsque le pas de temps diminue mais elle fait baisser la limite de stabilité lorsque le pas de temps augmente. Le remède est de réguler le pas afin qu'il ne s'accroisse que lentement.

2.3.4.2.3 Stabilité et précision du schéma

Pour étudier le schéma, on s'est contenté de l'analyse d'un système à un seul degré de liberté, libre et linéaire, de pulsation propre ω et d'amortissement réduit ξ :

$$\ddot{\xi} + 2\omega\xi\dot{\xi} + \omega^2\xi = 0$$

La solution approchée, en utilisant le schéma à pas de constant, s'obtient par la relation de récurrence suivante :

$$\mathbf{A}.\boldsymbol{Y}_n + \mathbf{B}.\boldsymbol{Y}_{n-1} = \mathbf{0}$$

avec $\boldsymbol{Y}_{\boldsymbol{n}} = \begin{bmatrix} \ddot{x}_n \\ \dot{x}_{n+\frac{1}{2}} \\ x_{n+1} \end{bmatrix}$

$\mathbf{A}$ et $\mathbf{B}$ sont deux matrices qui dépendent de la méthode choisie pour calculer la contribution du terme d'amortissement. On cherche une solution de la forme : $\boldsymbol{Y}_n = \lambda \boldsymbol{Y}_{n-1}$.

λ est une valeur propre de $\mathbf{A}^{-1}.\mathbf{B}$ et peut s'écrire sous la forme suivante :
$\lambda_c = \exp\left(\omega_c \Delta t\left(-\xi_c \pm i\sqrt{1-\xi_{c^2}}\right)\right)$ où ω_c et ξ_c sont la pulsation et l'amortissement réduit calculés par l'algorithme.

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

On peut les comparer à la solution exacte, $\lambda_e = \exp\left(\omega \Delta t \left(-\xi \pm i\sqrt{1-\xi^2}\right)\right)$, ce qui permet d'évaluer l'erreur sur la pulsation et l'erreur sur l'amortissement : $\frac{|\omega_c - \omega|}{\omega}$ et . $\frac{|\xi_c - \xi|}{\xi}$.

On a étudié [bib 8] et [bib10] les propriétés du schéma selon la méthode employée pour estimer la vitesse aux pas entiers. Il a été empiriquement trouvé que la méthode 3 est à la fois plus précise et plus stable que les méthodes 1 et 2. La méthode, sans surcoût de calcul, permet d'accroître l'ordre du schéma et donne dans la plupart des cas une meilleure précision, sauf en cas d'amortissement faible. Elle est cependant moins stable que la méthode 1. C'est la méthode 2 qui a été finalement retenue dans le schéma ADAPT. Ces études ont permis en outre d'estimer le nombre de points par période nécessaire pour garantir une intégration stable. 20 est une valeur qui donne une bonne marge de sécurité. C'est la valeur choisie par défaut.

2.3.4.2.4 Critères d'adaptation du pas de temps

Les développements précédents permettent de quantifier les erreurs introduites lors du calcul d'un système libre et linéaire. Ces critères ne permettent cependant pas d'adapter le pas de temps. Ils sont en effet délicats à mettre en œuvre dans les cas non linéaires et ne tiennent pas compte des variations de l'excitation.

Un autre approche consiste à étudier l'erreur locale introduite par le schéma à l'aide de développements limités.

La solution exacte d'un système à un degré de liberté vérifie :

$$\begin{cases} X\left(t+\frac{\Delta t}{2}\right) = X(t) + \frac{\Delta t}{2}\dot{X}(t) + \frac{\Delta t^2}{8}\ddot{X}(t) + \frac{\Delta t^3}{48}\dddot{X}(t) + o\left(\Delta t^3\right) \\ X\left(t-\frac{\Delta t}{2}\right) = X(t) - \frac{\Delta t}{2}\dot{X}(t) + \frac{\Delta t^2}{8}\ddot{X}(t) - \frac{\Delta t^3}{48}\dddot{X}(t) + o\left(\Delta t^3\right) \end{cases}$$

$$\Rightarrow X\left(t+\frac{\Delta t}{2}\right) = X\left(t-\frac{\Delta t}{2}\right) + \Delta t\,\dot{X}(t) + \frac{\Delta t^3}{24}\dddot{X}(t) + o\left(\Delta t^3\right)$$

La formule d'intégration des différences conduit donc à une erreur de troncature valant :

$$E_n = \frac{\Delta t^3}{24}\dddot{X}(t_n) \equiv \frac{\Delta t^2}{12}\left(\ddot{X}_n - \ddot{X}_{n-1}\right)$$

On peut normer cette erreur pour obtenir une erreur relative :

$$e_n = \frac{\Delta t^2}{12}\frac{|\ddot{X}_n - \ddot{X}_{n-1}|}{X_n} \quad , \quad X_n \neq 0$$

Park et Underwood [bib7] ont interprété cette erreur en définissant une « pulsation apparente » :

$$\omega_{A_n^2} = \frac{\ddot{X}_n}{X_n}$$

Appliquée au schéma à différence centrée, cette définition permet d'interpréter l'erreur relative e_n comme une variation de la pulsation apparente :

$$e_n \equiv \frac{\Delta t^2}{12}|\omega_{A_n^2} - \omega_{A_{n-1}^2}|$$

De nombreux algorithmes utilisent un critère d'adaptation du pas de temps fondé sur l'erreur de troncature ([bib9], [bib11]). Cependant dans le cas d'un schéma conditionnellement stable, cette

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

méthode ni de s'assurer de la stabilité de l'intégration, ni de garantir une précision pour le calcul des transitoires.

D'autres méthodes utilisent une approximation de la pulsation propre instantanée du système [bib12], à l'aide des matrices de masse et de raideur. Elles ont le défaut de ne pas s'adapter aux forces extérieures et à leurs fluctuations en fréquence.

Il est donc utile de trouver un critère qui tienne compte des deux approches. C'est pourquoi Park et Underwood ont introduit la notion de « fréquence apparente perturbée » :

$$f_{AP_n}=\frac{1}{2\mathrm{p}}\sqrt{|\frac{\ddot{X}_n-\ddot{X}_{n-1}}{X_n-X_{n-1}}|}$$

Cette grandeur s'interprète comme la fréquence « instantanée » du système.

Dans le cas d'un système à plusieurs degrés de liberté, il faut calculer une fréquence apparente pour chaque degré de liberté et prendre le maximum. Le pas de temps peut être alors choisi pour respecter un minimum de points par période apparente.

Si le dénominateur de l'expression de la fréquence apparente tend vers zéro, celle-ci peut devenir très grande et ne plus avoir de signification. Ceci mène à un raffinement injustifié lorsque la vitesse s'annule. Pour y remédier on ajoute un critère du type :

$$\frac{|X_n-X_{n-1}|}{Dt}<\dot{X}_{\min}\Rightarrow f_{AP_n}=\frac{1}{2\mathrm{p}}\sqrt{|\frac{\ddot{X}_n-\ddot{X}_{n-1}}{\dot{X}_{\min}Dt}|}$$

C'est un intermédiaire entre la fréquence apparente perturbée et l'erreur de troncature. La valeur adéquate de $f_{AP_n}=\frac{1}{2\pi}\sqrt{|\frac{\ddot{X}_n-\ddot{X}_{n-1}}{X_n-X_{n-1}}|}$ est difficile à choisir *a priori* et une valeur inadaptée entraîne une diminution artificielle de la fréquence apparente. Dans le cas d'un système à plusieurs degrés de liberté on contourne cette difficulté en employant les degrés de liberté « voisins » :

$$f_{AP_n}=\max_{1\le i\le \mathrm{nb}\,ddl}\left(\frac{1}{2\pi}\sqrt{|\frac{\ddot{X}^i_n-\ddot{X}^i_{n-1}}{b^i_n}|}\right)$$

$$b^i_n=\max\left(\max_{|i-j|\le lb}|X^j_n-X^j_{\mathrm{n\text{-}1}}|,\dot{X}_{\min}\Delta t\right)$$

où lb désigne par exemple la largeur de bande de la matrice de raideur et où peut être $\dot{X}_{\min}$ choisi très petit.

Cette méthode se révèle très efficace dans le cas où les X^i_n désignent des composantes physiques (déplacements). Dans le cas d'une projection sur base modale il n'est pas pertinent d'employer les composantes voisines pour calculer la fréquence apparente. Dans ce cas il vaut mieux revenir au premier critère et utiliser une des deux méthodes suivantes, spécifiées par le mot clef VITE_MIN :

- si VITE_MIN : 'NORM' alors c'est un paramètre variable égal à $\frac{\|\dot{X}_n\|}{100}$ ($\|\dot{X}_n\|=\sqrt{\sum_{1\le i\le \mathrm{nb}\,ddl}(\dot{X}^i_n)^2}$). Cette méthode donne de bons résultats lorsque le nombre de degrés de liberté est grand et est inapplicable au cas à un seul degré de liberté. Elle n'est plus indiquée si l'ordre de grandeur de la vitesse est très différent d'un degré de liberté à un autre.
- si VITE_MIN : 'MAXI' alors c'est un paramètre variable et différent pour chaque degré de liberté, $\dot{X}^j_{\min}=\max_{1\le m\le n}\frac{|\dot{X}^j_m|}{1001}$. Cette méthode a l'avantage de fonctionner quel que soit le nombre de degré de liberté du système mais elle ne peut pas être utilisée si l'ordre de

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

grandeur de la vitesse varie trop au cours du calcul car, dans ce cas, on obtiendrait systématiquement : $\frac{|X_n^j - X_{n-1}^j|}{\Delta t} \leq \dot{X}_{\min}^j$.

2.3.4.2.5 Algorithme du schéma des différences centrées à pas adaptatif

Les règles évoquées plus haut permettent de fixer un nombre de pas de temps souhaité par période de la réponse en fonction de la précision voulue, N . Il est ajustable par le mot clef NB_POINT_PERIODE. Le pas de temps Δt_n doit alors être inférieur à $\frac{1}{Nf_{AP_n}}$. Le mot clef PAS donne le pas de temps initial, Δt_{ini} , et le mot clef PAS_MAXI le pas de temps maximal à ne pas dépasser, $\Delta t_{\max}$. Dans les anciennes versions du code (avant la version 10.1.20) le mot clef PAS définissait en même temps le pas maximal, le mot clef PAS_MAXI n'existant pas.

L'algorithme est décrit schématiquement ci-dessous :

1) initialisation : X_0 et $\dot{X}_0$ donnés
$\Delta t_{-1} = 0$, $\Delta t_0 = \Delta t_{ini}$ et $\ddot{X}_0 = \ddot{X}(t_0, X_0, \dot{X}_0)$

2) initialisation de $\dot{X}_{\min}$

3) A chaque pas de temps

1) initialisation de la recherche du pas de temps : $N_{iter} = 0$
2) calcul de $\dot{X}_{n+\frac{1}{2}}$ puis de X_{n+1} :

estimation de la vitesse (méthode 2) : $\dot{X}_n = \dot{X}_{n-\frac{1}{2}} + \frac{\Delta t}{2}\ddot{X}_{n-1}$

vitesse au mi pas : $\dot{X}_{n+\frac{1}{2}} = \dot{X}_{n-\frac{1}{2}} + \frac{\Delta t_{n-1} + \Delta t_n}{2}\ddot{X}_n(t_n, X_n, \dot{X}_n)$

déplacement : $X_{n+1} = X_n + \Delta t\, \dot{X}_{n+\frac{1}{2}}$

- calcul de l'accélération $\ddot{X}_{n+1}$
- calcul de la fréquence apparente :

$$\frac{|X_n - X_{n-1}|}{\Delta t} \geq \dot{X}_{\min} \Rightarrow f_{AP_n} = \frac{1}{2\pi}\sqrt{|\frac{\ddot{X}_n - \ddot{X}_{n-1}}{X_n - X_{n-1}}|}$$

$$\frac{|X_n - X_{n-1}|}{\Delta t} < \dot{X}_{\min} \Rightarrow f_{AP_n} = \frac{1}{2\pi}\sqrt{|\frac{\ddot{X}_n - \ddot{X}_{n-1}}{\dot{X}_{\min}\Delta t}|}$$

3) vérification de l'adéquation entre le pas de temps et la fréquence apparente :

calcul de l'indicateur $err = \Delta t_n Nf_{AP_n}$

- si $err \geq 1$ et $N_{itez} < N_{iter\ \max}$ alors réduction du pas de temps et nouvelle itération de recherche du pas : $\Delta t_n \leftarrow 0,75\Delta t_n$, $N_{iter} \leftarrow N_{iter} + 1$, retour en 2)
- si $err \leq 0,75$ depuis plus de 5 pas de temps consécutifs, alors accroissement du pas de temps $N_{iter} \leftarrow N_{iter} + 1$
- archivage de la solution, calcul éventuel de $\dot{X}_{\min}$ et retour en 1) pour l'itération suivante

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

2.3.4.2.6 Commentaires sur les paramètres de l'algorithme

Le fait de fixer une borne supérieure $N_{iter\ \max}$ par le mot clef NMAX_ITER_PAS au nombre de réductions du pas de temps permet de s'assurer de la convergence de l'algorithme dans les cas difficiles (par exemple en cas de discontinuité dans les forces extérieures).

Lorsque l'indicateur err est supérieur à 1, le pas de temps est multiplié par un facteur fixe (0,75 par défaut mais il peut être modifié par l'utilisateur grâce à l'opérande COEF_DIVI_PAS). Il aurait été possible d'écrire directement : $\Delta t_n \rightarrow \frac{1}{err}\Delta t_n = Nf_{AP_n}$, ce qui plus intuitif. Mais cette stratégie conduit à un raffinement excessif, la fréquence apparente calculée étant souvent largement supérieure à la fréquence réelle, lorsque l'erreur est grande. Toutefois, en un seul pas de temps, Δt_n peut être considérablement réduit (facteur $0{,}75^{N_{iter}}$).

En revanche l'augmentation du pas de temps est toujours beaucoup plus lente (coefficient de multiplication par défaut de 1,1 définissable par COEF_MULT_PAS) et n'a lieu que si l'indicateur est inférieur à 1 pendant cinq pas de temps consécutif. Ces restrictions se justifient par les risques de perte de stabilité ou de précision du schéma lorsque le pas de temps varie trop vite. Un coefficient de 1, 2 ou 1, 3 peut permettre un calcul plus rapide mais expose parfois à des risques d'erreur.

En résumé les valeurs par défaut ont été validées par de nombreux tests et donnent en général satisfaction en termes de précision et de stabilité [bib8].

2.3.4.2.7 Performance de l'algorithme

A précision égale, le nombre d'itérations effectuées par un schéma adaptatif est au moins cinq fois plus faible qu'avec un pas constant dans les phénomènes qui justifient l'utilisation d'un pas variable par l'aspect irrégulier de leur évolution (chocs, excitations discontinues, etc.).

Les études empiriques ont montré que le pas de temps adaptatif permet dans les cas favorables de gagner un facteur deux ou trois en temps de calcul. Ce schéma permet en outre de maîtriser la précision de l'intégration par le moyen du contrôle du nombre de points par période de la réponse.
Dans le cas des systèmes très amortis, les gains peuvent être encore plus importants (calculs cinq à dix fois plus rapides).

En revanche lorsque le pas de temps « idéal » est à peu près constant, l'utilisation du schéma adaptatif se révèle inutile.

Il permet bien sûr la prise en compte des non linéarités localisées de type chocs ou frottements.

En sous-structuration dynamique, il est compatible aussi bien avec l'analyse transitoire sur la base modale restituée sur le système entier ou le calcul transitoire sur les bases distinctes des sous-structures.

2.3.4.3 Schémas à pas de temps adaptatifs de la famille de Runge-Kutta

Les méthodes d'intégration explicites à un pas de type Runge-Kutta cherchent à déterminer une solution approchée du problème de Cauchy suivant :

$$\begin{cases} \dot{y}(t) = f(t, y(t)) \quad t \in [t_0, T] \\ y(t_0) = y_0 \quad y_0 \in \mathbb{R} \end{cases}$$

Pour cela, on subdivise l'intervalle $[t_0 : T] : t_0 < t_1 < t_2 < \cdots < t_N$ et on pose $\Delta t = t_{n+1} - t_n$. A chaque instant de l'intervalle la solution du problème de Cauchy est donnée par l'expression intégrale :

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_n + \Delta t} f(t, y(t))$$

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

La solution approchée $y_n \approx y(t_n)$ proposée par les schémas récursifs de Runge-Kutta prennent la forme :

$$y_{n+1} = y_n + \Delta t \Psi(t_n, y_n, \Delta t)$$

avec :

$$\Psi : [t_0, T] \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$$

où on identifie facilement le terme $\Delta t \Psi(t_n, y_n, \Delta t)$ comme l'approximation du terme intégrale $\int_{t_n}^{t_n+\Delta t} f(t, y(t))$.

Ainsi, une méthode de Runge-Kutta à s étages permettant d'approcher la solution du problème de Cauchy est donnée par la méthode de quadrature suivante :

$$\begin{cases} k_1 = f(t_n, y_n) \\ k_2 = f(t_n + c_2 \cdot \Delta t, y_n + a_{21} \cdot k_1) \\ \vdots \\ k_s = f(t_n + c_s \cdot \Delta t, y_n + a_{s1} \cdot k_1 + a_{s2} \cdot k_2 + \cdots + a_{s\ s-1} \cdot k_{s-1}) \\ y_{n+1} = y_n + \Delta t \cdot (b_1 \cdot k_1 + b_2 \cdot k_2 + \cdots + b_s \cdot k_s) \end{cases}$$

Un schéma d'intégration de Runge-Kutta est donc entièrement défini par les coefficients b_i, c_i et a_{ij} avec, par ailleurs, $c_i = \sum_j a_{ij}$. Dans le cas d'un schéma de Runge-Kutta explicite, les coefficients doivent également vérifier la condition $a_{ij} = 0 \; \forall \; j \geq i$.

En pratique, l'ensemble des coefficients d'un schéma de Runge-Kutta est présenté sous la forme d'un tableau de Butcher comme l'illustre la figure ci-dessous :

c_1				
c_2	a_{21}			
c_3	a_{31}	a_{32}		
$\vdots$	$\cdots$	$\cdots$	$\ddots$	
c_s	a_{s1}	a_{s2}	$\cdots$	$a_{s\ s-1}$
y_{n+1}	b_1	b_2	$\cdots$	b_s

De plus, on dira qu'une méthode est d'ordre p si $\forall n, 0 \leq n \leq N, e_n = O(\Delta t^{p+1}), \Delta t \to 0$

2.3.4.3.1 Schémas de Runge-Kutta emboîtés pour le contrôle du pas de temps adaptatif

Lors de la résolution numérique du problème de Cauchy décrit plus haut, le choix du pas de temps Δt est déterminant sur l'ordre de grandeur de l'erreur globale. Ainsi, le contrôle de l'erreur commise e_n lors du calcul de la solution $y_n \approx y(t_n)$ permet de déterminer un choix de pas de temps dit « optimal » de manière à garantir une majoration de l'erreur par une tolérance fournie par l'utilisateur.

Pour ce faire, une procédure classique consiste à employer deux méthodes de Runge-Kutta dites emboîtées. La première méthode d'ordre p à s étages sert à calculer la solution approchée y_{n+1}, alors que la seconde méthode d'ordre $\hat{p} < p$ sert à estimer l'erreur $e_n = \|y_{n+1} - \hat{y}_{n+1}\|$ pour le contrôle du pas de temps. En général, on a $\hat{p} = p - 1$ et on note la méthode $RK\ p(\hat{p})$.
L'avantage de cette approche est que l'approximation d'ordre la plus faible $\hat{p}$ utilise les mêmes évaluations de f et donc les mêmes coefficients a_{ij}.

Dans le cadre de l'opérateur DYNA_TRAN_MODAL, deux schémas d'intégration explicites de la famille Runge-Kutta sont disponibles :

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

1. Le schéma 'RUNGE_KUTTA_32' : Il s'agit ici du schéma de Bogacki-Shampine3(2). Ce schéma est d'ordre 3 avec 4 étages. Il intègre également une approximation d'ordre 2 permettant le contrôle de l'erreur. Bien qu'il y ait 4 étages, il utilise réellement 3 étages car il a la propriété FSAL (First Same As Last). Le tableau de Butcher associé est présenté ci-dessous :

0				
1/2	1/2			
3/4	0	3/4		
1	2/9	1/3	4/9	
y_{n+1}	2/9	1/3	4/9	0
$\hat{y}_{n+1}$	7/24	1/4	1/3	1/8

2. Le schéma 'RUNGE_KUTTA_54' : Il s'agit ici du schéma de Dormand-Prince connu également sous le nom de DOPRI5(4). Ce schéma est d'ordre 5 avec 7 étages. Avec une approximation d'ordre 4 permettant le contrôle de l'erreur. Comme le schéma précédent il a la propriété FSAL et il n'utilise réellement que 6 étages. Le tableau de Butcher associé est présenté ci-dessous :

c0	0							
c1	1/5	1/5						
c2	3/10	3/40	9/40					
c3	4/5	44/45	-56/15	32/9				
c4	8/9	19372/6561	-25360/2187	64448/6561	-212/729			
c5	1	9017/3168	-355/33	46732/5247	49/176	-5103/18656		
c6	1	35/384	0	500/1113	125/192	-2187/6784	11/84	
b	$\hat{y}_{n+1}$	5179/57600	0	7571/16695	393/640	-92097/339200	187/2100	1/40
d	y_{n+1}	35/384	0	500/1113	125/192	-2187/6784	11/84	0

Ces deux schémas programmés dans l'opérateur DYNA_TRAN_MODAL considèrent un vecteur d'état y_n comme étant la concaténation du vecteur de déplacements et des vitesses, à savoir :

$$y_n = \begin{Bmatrix} X_n \\ \dot{X}_n \end{Bmatrix}$$

Par ailleurs, la norme utilisée pour le contrôle de l'erreur relative est donnée par [bib14] :

$$err = \frac{1}{d} \sum_{k=1}^{d} \sqrt{\left(\frac{y_{n+1}^k - \hat{y}_{n+1}^k}{sc^k} \right)^2}$$

où d est la dimension du vecteur d'état y, y_{n+1}^k et $\hat{y}_{n+1}^k$ les k -ièmes composantes des vecteurs y_{n+1} et $\hat{y}_{n+1}$ respectivement. Enfin, sc^k est donné par:

$$sc^k = MAX\left(|y_n^k|, |y_{n+1}^k|\right) + \alpha$$

où α est un paramètre régularisation (valeur par défaut 0,001). Ainsi, l'algorithme contrôle la majoration de l'erreur relative par l'expression $err \leq tol$ où tol est une tolérance relative donné par l'utilisateur.

Enfin, l'expression du pas de temps optimal fonction de l'erreur commise est :

$$\Delta t_{opt} = 0{,}9 \cdot \Delta t_n \cdot \left(\frac{tol}{err}\right)^{\frac{1}{p+1}}$$

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

avec $p=5$ pour le schéma 'RUNGE_KUTTA_54' et $p=3$ pour le schéma 'RUNGE_KUTTA_32'. Pour une meilleure compréhension, les deux algorithmes sont présentés dans le tableau ci-dessous.

1) Initialisation : X_0 , $\dot{X}_0$, t_0 et Δt_0 donnés :
$n=0$, $\Delta t=\Delta t_0$ et $\ddot{X}_0=\ddot{X}\left(t_0, X_0, \dot{X}_0\right)$

2) Tant que la condition $t_n<T$ est satisfaite

3) Calcul de l'état suivant y_{n+1} et de l'erreur relative err par une méthode de Runge_Kutta

4) Si la condition $err \leq tol$ est satisfaite (le pas de temps est accepté) :

- Archivage de X_0 , $\dot{X}_0$ et $\ddot{X}_0$
- Passage à l'état suivant :
  - $y_n := y_{n+1}$
  - $t_n := t_n + \Delta t$
  - $n := n+1$

5) Calcul de Δt_{opt} sous la contrainte $0{,}2 \cdot \Delta t_n \leq \Delta t_{opt} \leq 5 \cdot \Delta t_n$ (afin d'éviter des changements brutaux)

6) Sélection du pas de temps $\Delta t := min(\Delta t_{opt}, t_n - T)$

On peut remarquer plusieurs choses, particulière à la méthode 'RUNGE_KUTTA_54':

- les étages 5 et 6, font le calcul à t+dt.
- que les valeurs de "d" correspondent aux coefficients du 6ème étage de la méthode.
- l'erreur est calculé à l'aide (b-d)*ki, et le pas est adapté au besoin. On peut donc prendre soit "b" soit "d" pour calculer la solution à t+dt.
- "d" permet de calculer la solution à l'ordre 4 avec un contrôle de l'erreur. On garde donc l'évaluation réalisée au 6ème étage de la méthode pour des raisons de performance CPU. Il n'y a plus besoin d'utiliser ni "b" ni "d" pour évaluer la solution à t+dt.

Pour des raisons de performance CPU et d'erreur numérique, on calcule sous forme rationnelle "b-d" dans la programmation, pour l'évaluation de l'erreur. Les coefficients "b" et "d" ne servent donc plus lors la programmation de cette méthode. (Pour tracer les opérations réalisées ils sont mis en commentaires dans le code).

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

3 Conclusion

En guise de conclusion, voici résumés les différentes possibilités d'intégration temporelle qu'offre l'opérateur :

- Euler (‘EULER’) explicite modifié pour assurer une stabilité conditionnelle,
- Schéma de Newmark (‘NEWMARK’) paramétré de façon à être implicite,
- Schéma de Devogelaere-Fu (‘DEVOGE’) d'ordre 4,
- Quatre schémas adaptatifs explicites :
  - ‘ADAPT_ORDRE1’ s'appuyant sur le schéma d'Euleur,
  - ‘ADAPT_ORDRE2’ basé sur le schéma de différences centrées,
  - deux schémas de la famille Runge-Kutta : ’RUNGE_KUTTA_32’ et RUNGE_KUTTA_54’.

Le schéma par défaut est EULER mais ce n'est pas systématiquement le plus adapté. Le schéma de NEWMARK disponible dans le *Code_Aster* est implicite et garantit une stabilité inconditionnelle mais n'est utilisable que pour des problèmes purement linéaires.

Le schéma DEVOGE est d'ordre 4 et donc est plus précis mais est il est coûteux en temps de calcul.

Les schémas adaptatifs sont plus particulièrement indiqués pour les problèmes avec des non-linéarités localisées, où le pas de temps « idéal » n'est pas constant au cours du transitoire. C'est donc l'expérience de la modélisation qui permet de choisir le schéma le mieux adapté au problème en fonction du rapport (temps de calcul)/précision.

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

4 Bibliographie

1) K.-J. BATHE, E.-L. WILSON : « Numerical Methods in Finite Element Analysis », Prentice Hall Inc.

2) J. ANTUNE, F. AXISA, H. BUNG, F. DOVEIL, E. de LANGRE : « Méthodes d'analyse en dynamique non linéaire des structure » - IPSI.

3) J.-R. LEVESQUE, P. LABBE et al. : « Module TRANSIS du code POUX in documentation de référence du code POUX » - Rapport interne EDF.

4) X. RAUD, P. RICHARD : « Structure et validation du calcul non linéaire des paliers hydrodynamiques du code CADYRO » – Note HP-61/92/041.

5) G.-D. HAHN : « A Modified Euler Method for Dynamic Analysis » - International Journal for Numerical Methods in Engineering Vol. 32 (1991), pp 943-955.

6) K.-C. PARK, P.-G. UNDERWOOD : « A Variable-Step Central Difference Method for Structural Dynamic Analysis – Part II : Implementation and performance evaluation » - Computer Methods in Applied Mechanics and Engineering Vol. 23 (1980) pp 259-279.

7) K.-C. PARK, P.-G. UNDERWOOD : « A Variable-Step Central Difference Method for Structural Dynamic Analysis – Part I : Theoritical Aspects » - Computer Methods in Applied Mechanics and Engineering Vol. 22 (1980) pp 241-258.

8) G. JACQUART, S. GARREAU : « Algorithme d'intégration à pas de temps adaptatif dans le *Code_Aster* » - Note HP-61/95/023.

9) O.C.. ZIENKIEWICZ, Y.M XIE : « A Posteriori Local Error Estimation and Adaptative Time-Stepping Procedure for Dynamic Analysis » Earthquake Engineering and Structural Dynamics Vol. 20(1991) pp. 871-887.

10) A.-C. LEGER, G. JACQUART : « Algorithmes d'intégration de l'opérateur DYNA_TRAN_MODAL du *Code_Aster* : documentation de référence » - Note HP-51/96/072.

11) R.M. THOMAS, I. GLADWELL : « Variable Order Variable Step Algorithms for Second Order systems » - International Journal for Numerical Methods in Engineering Vol. 26 (1998) pp. 39-53.

12) P.G. BERGAN, E. MOLLESTAD : «An Automatic Time-Stepping Algorithm for Dynamic Problem » - Computer Methods in Applied Mechanics and Engineering Vol. 49 (1985) pp. 299-318.

13) N. GAY, S. GRANGER, T. FRIOU : « Présentation d'une méthode de simulation du couplage fluidélastique en régime non linéaire » - Note HT-32/94/015.

14) E. HAIRER, S.P. NØRSETT, S. WANNER : « Solving ordinary differential equations I : Nonstiff problems » - Springer-Verlag, second revised edition, 1993

15) C. GARDNER : « Numerical Methods for Ordinary and Partial Differential Equations » - book draft, https://math.la.asu.edu/~gardner/522.html

16) BANK RE, COUGHRAN JR WM, FICHTNER W, GROSSE EH, ROSE DJ, SMITH RK « Transient Simulations Of Silicon Devices And Circuits » - IEEE Trans CAD 1985 ; CAD-4(4): 436–51

17) K. J.BATHE, M. M. I. BAIG : « On a composite implicit time integration procedure for nonlinear dynamics » - Computers and Structures, 83 (2005) 2513–2524

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

18) N. TARDIEU : « Implantation du schéma d'intégration temporelle TRBDF2 dans code_aster » - Note interne EDF R&D 6125-1722-2021-02622-FR

5 Description des versions du document

Version Aster	Auteur(s) Organisme(s)	Description des modifications
5	*G. JACQUART EDF-R&D/AMV*	Texte initial
6.4	*E. BOYERE,* R&D/AMA *A.C. LEGER,* EDF-R&D/TESE, *G. JACQUART DER/AMV*	
10.4	N.GREFFET, F.VOLDOIRE R&D/AMA	Ajout de la syntaxe PAS_MAXI pour le schéma ADAPT (fiche 14906). Ajout d'un schéma ADAPT basé sur le schéma d'Euler (fiche 16222).

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