
Intégration numérique de lois de comportement élastoplastique

Gilbert Touzot — Jaouad Dabounou

Département de génie mécanique
Université de Technologie de Compiègne
BP 649
60206 Compiègne

RESUMÉ. Cet article présente de nouveaux algorithmes d'intégration de lois de comportement élastoplastique en vue d'une utilisation dans un code de calcul des structures. Dans toutes les méthodes présentées le pas d'intégration est ajusté dynamiquement de manière à obtenir la précision souhaitée à moindre coût. Une technique originale d'ajustement du pas et de prolongement de la solution dans le cas d'une méthode de type Runge-Kutta d'ordre quatre est proposée dans ce travail. Les schémas présentés n'exigent aucune forme de correction des contraintes pour éviter un dépassement du critère de plasticité. Afin d'améliorer la convergence des algorithmes de résolution des équations d'équilibre, on propose une expression du module tangent consistant avec l'algorithme d'intégration. Plusieurs exemples permettent de comparer les méthodes proposées avec les méthodes existantes.

ABSTRACT. This paper presents some new integration algorithms for integration of elastoplastic constitutive laws. These algorithms are designed for use in finite element codes. In all methods presented the error is controlled by adjusting the size of each substep automatically. An original technique of adjusting the stepsize and extending the solution in case of using a fourth order Runge-Kutta method is proposed. Neither of the algorithms requires any form of stress correction to avoid drift from the yield surface. To preserve the quadratic convergence of the iterative Newton-Raphson schemes, an expression of tangent moduli consistent with the integration algorithm is developed. The proposed methods are compared with other integration algorithms and are tested with several examples.

MOTS-CLÉS : éléments finis, contrôle de pas, Runge-Kutta, rigidité consistante.

KEY WORDS : finite elements, stepsize control, Runge-Kutta, consistent modulus.

1. Introduction

Les algorithmes de résolution utilisés dans la modélisation du comportement non-linéaire des solides sont en général constitués d'une première boucle sur les pas de chargement, qui peut être vue comme une prédiction de la configuration

d'équilibre, à l'intérieur de laquelle une deuxième boucle d'itérations améliore le résidu d'équilibre. Dans cette dernière, on est amené à utiliser des lois de comportement qui relient le taux de contraintes au taux de déformations. En élastoplasticité ces lois sont le plus souvent des relations différentielles de la forme:

$$\dot{\sigma} = \mathbf{D}^{ep}(\sigma, \kappa) \dot{\varepsilon} \quad [1]$$

où σ est le tenseur des contraintes, ε le tenseur des déformations, κ un paramètre d'érouissage et \mathbf{D}^{ep} la matrice tangente élastoplastique. Les dérivations sont effectuées par rapport à un paramètre de chargement noté t (pseudo-temps), définissant l'évolution du chargement ou des déplacements imposés à la structure.

A chaque itération on doit calculer les contraintes à la fin du pas de chargement Δt , correspondant à un accroissement de déformation estimé sur le pas. Cela peut être réalisé en intégrant l'équation [1] sur le pas Δt tout entier ce qui s'avère souvent très coûteux du fait qu'on est amené alors à utiliser des pas de chargement très petits. Pour atteindre une précision d'intégration suffisante on peut également subdiviser le pas de chargement en sous-pas sur lesquels on intègre l'équation [1]. On peut ainsi utiliser des pas de chargement relativement grands, et donc diminuer le coût des calculs tout en conservant une bonne précision. La subdivision du pas de chargement peut être faite de différentes manières, mais pour respecter les non-linéarités rencontrées au cours de l'intégration, il est souhaitable qu'elle s'adapte à ces non-linéarités pour conduire à une erreur constante au cours du chargement.

Dans ce travail on présente quelques algorithmes d'intégration basés sur des méthodes d'Euler, de Runge-Kutta, ou d'extrapolation. On propose aussi un algorithme, qui constitue une généralisation originale de l'une des méthodes de Runge-Kutta, qui permet d'ajuster la taille des sous-pas et de prolonger la solution.

Le caractère auto-adaptatif des schémas présentés permet d'éviter tout dépassement du critère de plasticité; [SLO 87] et [POL 89]. Les corrections des contraintes utilisées habituellement dans les méthodes à pas constant sont alors inutiles.

L'utilisation de la matrice tangente élastoplastique \mathbf{D}^{ep} , dite module tangent continu, introduit des erreurs supplémentaires dans le processus de Newton-Raphson utilisé pour résoudre les équations d'équilibre [SIM 85], nous proposons donc un module tangent *consistant* avec l'algorithme d'intégration, c'est à dire obtenu par linéarisation de l'algorithme utilisé pour l'intégration des équations différentielles élastoplastiques. Plusieurs exemples sont traités pour comparer la performance des différentes méthodes et valider l'algorithme proposé.

2. Formulation des relations élastoplastiques

En utilisant les notations standards de l'élastoplasticité, le tenseur taux de déformations se décompose en deux parties, élastique et plastique, de la manière suivante:

$$\dot{\varepsilon} = \dot{\varepsilon}^e + \dot{\varepsilon}^p \quad [2]$$

où la partie élastique est liée au tenseur des contraintes par la loi de Hooke

$$\dot{\varepsilon}^e = \mathbf{D}^{e-1} \dot{\sigma} \quad [3]$$

et où la déformation plastique se produit seulement lorsque le critère de plasticité s'annule

$$f(\sigma, R, X) = 0 \quad [4]$$

La loi de normalité s'écrit:

$$\begin{aligned} \dot{\varepsilon}^p &= \dot{\lambda} \frac{\partial F}{\partial \sigma} \\ - \dot{\alpha} &= \dot{\lambda} \frac{\partial F}{\partial X} \\ - \dot{p} &= \dot{\lambda} \frac{\partial F}{\partial R} \end{aligned} \quad [5]$$

où $F(\sigma, R, X)$ est le potentiel d'écoulement, p et α sont respectivement les variables duales de R et de X , et $\dot{\lambda}$ est le multiplicateur plastique qui est toujours positif. Dans le cas d'une loi associée on a

$$F(\sigma, R, X) = f(\sigma, R, X) \quad [6]$$

Pendant l'écoulement on a

$$\begin{aligned} f(\sigma, R, X) &= 0 \\ \dot{f}(\sigma, R, X) &= 0 \end{aligned} \quad [7]$$

Les relations précédentes donnent alors

$$\dot{\lambda} = \frac{\frac{\partial f^T}{\partial \sigma} \mathbf{D}^e \dot{\varepsilon}}{\frac{\partial f^T}{\partial \sigma} \mathbf{D}^e \frac{\partial F}{\partial \sigma} + \frac{\partial f}{\partial R} \frac{\partial r}{\partial p} \frac{\partial F}{\partial R} + \frac{\partial f^T}{\partial X} \frac{\partial \chi}{\partial \alpha} \frac{\partial F}{\partial X}} \quad [8]$$

où r et χ sont des données expérimentales vérifiant

$$R = r(p) \quad \text{et} \quad X = \chi(\alpha)$$

En utilisant l'expression [8] de $\dot{\lambda}$ dans [5], et en tenant compte des relations [2], [3] et [5] on obtient:

$$\begin{aligned} \dot{\sigma} &= [\mathbf{D}^e - \mathbf{D}^p(\sigma, p, \alpha)] \dot{\epsilon} \\ \dot{\alpha} &= \mathbf{A}(\sigma, p, \alpha) \dot{\epsilon} \\ \dot{p} &= \mathbf{P}(\sigma, p, \alpha) \dot{\epsilon} \end{aligned} \tag{9}$$

où

$$\begin{aligned} \mathbf{D}^p &= \frac{\mathbf{D}^e \frac{\partial F}{\partial \sigma} \frac{\partial f^T}{\partial \sigma} \mathbf{D}^e}{\frac{\partial f^T}{\partial \sigma} \mathbf{D}^e \frac{\partial F}{\partial \sigma} + \frac{\partial f}{\partial R} \frac{\partial r}{\partial p} \frac{\partial F}{\partial R} + \frac{\partial f^T}{\partial X} \frac{\partial \chi}{\partial \alpha} \frac{\partial F}{\partial X}} \\ \mathbf{A} &= - \frac{\frac{\partial F}{\partial X} \frac{\partial f^T}{\partial \sigma} \mathbf{D}^e}{\frac{\partial f^T}{\partial \sigma} \mathbf{D}^e \frac{\partial F}{\partial \sigma} + \frac{\partial f}{\partial R} \frac{\partial r}{\partial p} \frac{\partial F}{\partial R} + \frac{\partial f^T}{\partial X} \frac{\partial \chi}{\partial \alpha} \frac{\partial F}{\partial X}} \\ \mathbf{P} &= - \frac{\frac{\partial F}{\partial R} \frac{\partial f^T}{\partial \sigma} \mathbf{D}^e}{\frac{\partial f^T}{\partial \sigma} \mathbf{D}^e \frac{\partial F}{\partial \sigma} + \frac{\partial f}{\partial R} \frac{\partial r}{\partial p} \frac{\partial F}{\partial R} + \frac{\partial f^T}{\partial X} \frac{\partial \chi}{\partial \alpha} \frac{\partial F}{\partial X}} \end{aligned}$$

Pour chaque itération d'équilibre on doit calculer les contraintes à la fin du pas de chargement qui correspondent à un accroissement de déformation donné, les contraintes au début du pas étant connues. On est donc amené à intégrer le système [9] d'équations différentielles ordinaires. Dans la partie suivante, on présente quelques unes des méthodes les plus utilisées à cette fin.

3. Présentation de quelques méthodes d'intégration d'équations différentielles ordinaires

On présente ici quelques méthodes d'intégration d'un système d'équations différentielles ordinaires

$$\begin{aligned} \dot{y} &= f(y, t) \quad t \in [T, T+\Delta T] \\ y(T) &= y_0 \quad \text{connue} \end{aligned} \tag{10}$$

Pour toutes ces méthodes, l'intégration se fait sur des sous-pas h_0, h_1, \dots de taille variable qui s'adapte automatiquement aux non-linéarités rencontrées. On se propose donc de résoudre l'équation [10] sur un sous-pas h et on suppose que y_n , estimation de la solution à l'instant t_n , est connue, l'estimation de la solution à l'instant $t_{n+1} = t_n+h$ étant notée y_{n+1} .

3.1. Méthode d'Euler semi-implicite

Elle est définie pour θ choisi dans $[0, 1]$ par:

$$y_{n+1} = y_n + h f_{n+\theta} \quad [11]$$

où

$$f_{n+\theta} = (1 - \theta) f(y_n, t_n) + \theta f(y_{n+1}, t_{n+1})$$

Cette méthode est illustrée par la figure 1. Si $\theta = 0$, il s'agit de la méthode d'Euler explicite. Lorsque θ est non nul cette méthode nécessite des itérations de Newton-Raphson. La méthode semi-implicite ($0 < \theta < 1$) présente les meilleures conditions de stabilité pour $\theta = 0.5$. Dans ce cas elle est d'ordre deux et on peut estimer la taille du prochain sous-pas h_p par la relation suivante [ELM 89]:

$$h_p = h \min \left(S \sqrt[3]{\frac{\varepsilon}{\text{ereul}}}, 2 \right) \quad [12]$$

$$S = 0.9$$

où ε est la précision requise et ereul , estimation de l'erreur de troncature, est donnée par:

$$\text{ereul} = y(t_{n+1}) - y_{n+1} = -\frac{h^3}{12} \ddot{f}(t_n, y_n) \quad [13]$$

Cette méthode est notée (euler).

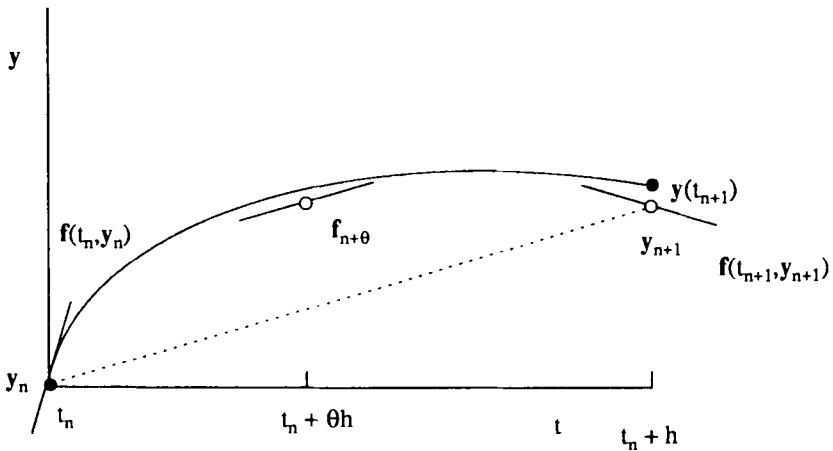


Figure 1. Méthode d'Euler semi-implicite

3.2. Méthodes de Runge-Kutta

3.2.1. Méthode de Runge-Kutta classique

On estime f en t_n , $t_n + \frac{h}{2}$, et $t_{n+1} = t_n + h$ où h est la taille du sous-pas d'intégration (voir figure 2)

$$\begin{aligned}
 k_1 &= f(t_n, y_n) \\
 k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_1\right) \\
 k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2} k_2\right) \\
 k_4 &= f(t_n + h, y_n + h k_3)
 \end{aligned}
 \tag{14}$$

et

$$y_{n+1} = y_n + \frac{h}{6} (k_1 + 2(k_2 + k_3) + k_4)
 \tag{15}$$

On a

$$y_{n+1} = y(t_{n+1}) + O(h^5)$$

où on a supposé $y_n = y(t_n)$. Cette méthode d'ordre quatre est notée (rk4c).

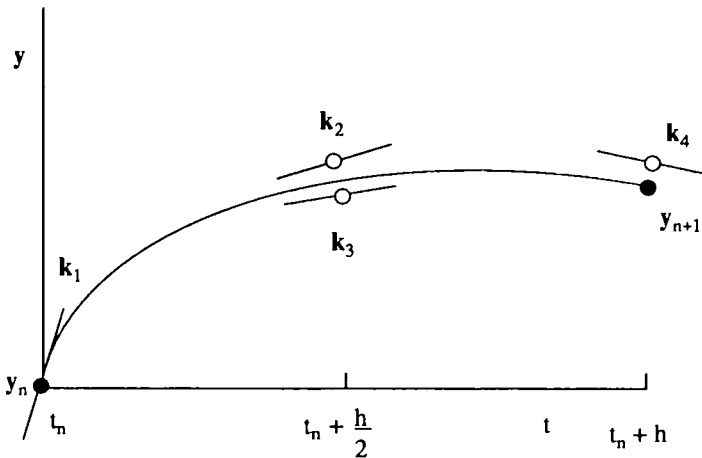


Figure 2. Pour (rk4c), f est estimée quatre fois: une fois au début du pas, deux fois au milieu et une fois à la fin et y_{n+1} est calculée à partir de ces estimations

3.2.2. Une autre méthode de Runge-Kutta d'ordre quatre

On estime f en t_n , $t_n + \frac{h}{3}$, $t_n + 2\frac{h}{3}$ et $t_{n+1} = t_n + h$

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(t_n, \mathbf{y}_n) \\ \mathbf{k}_2 &= \mathbf{f}\left(t_n + \frac{h}{3}, \mathbf{y}_n + \frac{h}{3} \mathbf{k}_1\right) \\ \mathbf{k}_3 &= \mathbf{f}\left(t_n + 2\frac{h}{3}, \mathbf{y}_n - \frac{h}{3} \mathbf{k}_1 + h \mathbf{k}_2\right) \\ \mathbf{k}_4 &= \mathbf{f}(t_n + h, \mathbf{y}_n + h(\mathbf{k}_1 - \mathbf{k}_2 + \mathbf{k}_3)) \end{aligned} \quad [16]$$

et

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{8} (\mathbf{k}_1 + 3(\mathbf{k}_2 + \mathbf{k}_3) + \mathbf{k}_4) \quad [17]$$

On a

$$\mathbf{y}_{n+1} = \mathbf{y}(t_{n+1}) + O(h^5)$$

Cette méthode d'ordre quatre est notée (rk4nc) pour la distinguer de la méthode de Runge-Kutta classique. Les méthodes de Runge-Kutta présentées sont explicites et stables; [CRO 84] et [BUT 87].

3.2.3. Contrôle de pas dans les méthodes de Runge-Kutta.

Pour contrôler la taille des sous-pas dans les méthodes de Runge-Kutta d'ordre quatre, on utilise très souvent la technique du pas double. Chaque sous-pas est considéré deux fois, tout d'abord comme un sous-pas de taille $2h$, et puis comme deux demi-sous-pas de taille h ; (figure 3). Soit \mathbf{y}_1 la solution obtenue pour un sous-pas de longueur $2h$, on a

$$\mathbf{y}(t_n + 2h) = \mathbf{y}_1 + (2h)^5 \varphi + O(h^6) \quad [18]$$

et soit \mathbf{y}_2 la solution obtenue en deux sous-pas chacun de longueur h , on a

$$\mathbf{y}(t_n + 2h) = \mathbf{y}_2 + 2h^5 \varphi + O(h^6) \quad [19]$$

où le facteur φ est proportionnel à $\mathbf{y}^{(5)}$ supposée constante sur tout le sous-pas de longueur $2h$. La différence:

$$\Delta = \mathbf{y}_2 - \mathbf{y}_1 = 30 h^5 \varphi + O(h^6) \quad [20]$$

est une bonne estimation de l'erreur de troncature. Si on effectue un pas h et qu'on obtient par la technique décrite plus haut une estimation de l'erreur Δ , le pas h_0 qui aurait donné la précision souhaitée ε est calculé d'après [20] par la relation:

$$h_0 = S h \left(\frac{\epsilon}{|\Delta|_{\infty}} \right)^{-0.20} \quad [21]$$

$$S = 0.9$$

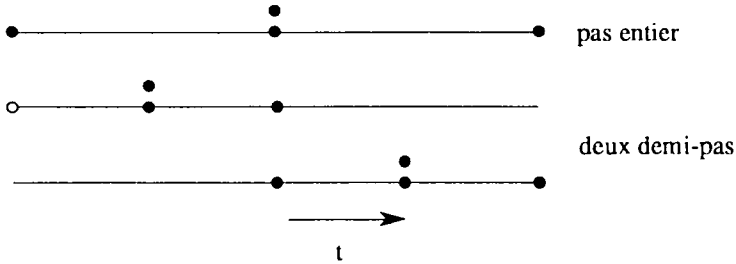


Figure 3. Le contrôle de la taille des pas dans (rk4c): Les points où f est évaluée sont désignés par des cercles pleins, le cercle creux veut dire que l'on réutilise la valeur de f calculée au début du pas entier (juste au-dessus). Le nombre des évaluations de f est donc 11 par pas. La comparaison des solutions données par le pas entier et les deux demi-pas permet de prédire le pas prochain ou de réduire le pas en cours.

3.2.4. Méthode de Runge-Kutta généralisée

Le contrôle de la taille des sous-pas utilisé dans les méthodes de Runge-Kutta est assez coûteux; il nécessite 11 estimations du second membre par sous-pas (figure 3). Nous proposons donc, en collaboration avec O. DÉBORDES*, une généralisation originale de l'une des méthodes de Runge-Kutta présentées. Le contrôle de pas peut alors être effectué sans estimations supplémentaires du second membre. Nous approchons f sur $[t_n, t_{n+1}]$ par une fonction \bar{f} , ce qui permet d'approcher y par \bar{y} , la primitive de \bar{f} , et d'avoir ainsi une estimation de l'erreur de troncature pour le contrôle de la taille des sous-pas.

Soient k_1, k_2, k_3 et k_4 les estimations de f données par [16], l'approximation \bar{f} de f sur $[t_n, t_{n+1}]$, est le polynôme d'interpolation de Lagrange défini par:

$$\bar{f}(t_n) = k_1$$

$$\bar{f}\left(t_n + \frac{h}{3}\right) = \frac{1}{12}(k_1 + 9k_2 + 3k_3 - k_4)$$

$$\bar{f}\left(t_n + 2\frac{h}{3}\right) = \frac{1}{6}(-k_1 + 3k_2 + 3k_3 + k_4)$$

$$\bar{f}(t_n + h) = \frac{1}{4}(k_1 - 3k_2 + 3k_3 + 3k_4)$$
[22]

* IMT, Marseilles, France

On montre (Annexe) que \bar{f} vérifie sur $[t_n, t_{n+1}]$ la relation suivante:

$$\bar{f}(t) = f(t, y(t)) + O(h^3) \quad [23]$$

Et si on pose $\theta = \frac{t - t_n}{h}$, alors \bar{f} s'écrit en fonction de θ

$$\bar{f}(t) = \bar{p}(\theta) = \bar{a}_2 \theta^2 + \bar{a}_1 \theta + \bar{a}_0 \quad [24]$$

où

$$\bar{a}_2 = \frac{3}{2} (2k_1 - 3k_2 + k_4)$$

$$\bar{a}_1 = -\frac{3}{4} (5(k_1 - k_2) - k_3 + k_4) \quad [25]$$

$$\bar{a}_0 = k_1$$

Posons

$$\bar{q}(\theta) = y_n + \int_0^\theta h \bar{p}(\xi) d\xi = y_n + \int_{t_n}^t \bar{f}(\tau) d\tau$$

on a

$$\bar{q}(\theta) = y_n + h\theta \left(\frac{1}{3} \bar{a}_2 \theta^2 + \frac{1}{2} \bar{a}_1 \theta + \bar{a}_0 \right) \quad [26]$$

ce qui donne pour $\theta = 1$; (voir Annexe):

$$\bar{q}(1) = y_n + \frac{h}{8} (k_1 + 3(k_2 + k_3) + k_4) = y(t_n + h) + O(h^5) \quad [27]$$

où on retrouve l'estimation de la solution donnée par (rk4nc), (relation [17]). Grâce à la relation [27], une estimation de l'erreur de l'approximation de f par \bar{f} en t_n+h est obtenue par:

$$\bar{e}(1) = \bar{p}(1) - f(t_n+h, \bar{q}(1)) = \bar{p}(1) - f(t_n+h, y(t_n+h)) + O(h^5) \quad [28]$$

Description de l'algorithme proposé

On peut utiliser la méthode présentée de la façon suivante:

1. On se donne h , éventuellement fonction d'informations du pas précédent
2. On calcule les estimations de f comme dans [16]
3. On calcule $\bar{p}(1)$ et $\bar{q}(1)$
4. On calcule $\bar{e}(1)$, estimation de l'erreur sur f en t_n+h , comme dans [28]

L'erreur sur y est alors donnée par:

$$\text{err} = \frac{h \bar{e}(1)}{4} \tag{29}$$

5. Soit ϵ la précision requise, si

$$|\text{err}|_{\infty} > \epsilon \tag{30}$$

réduire la taille du sous-pas en utilisant la formule suivante

$$h_0 = S h \left\| \frac{\text{err}}{\epsilon} \right\|_{\infty}^{-0.25} \tag{31}$$

$$S = 0.9$$

et aller en 1 avec $h = h_0$

6. Mettre à jour la solution

$$t_{n+1} = t_n + h, \quad y_{n+1} = \bar{q}(1) \tag{32}$$

7. Aller au sous-pas suivant dont la taille est obtenue d'après la relation

$$h_1 = S h \left\| \frac{\text{err}}{\epsilon} \right\|_{\infty}^{-0.25} \tag{33}$$

On remarque que le coût de l'estimation de l'erreur dans le schéma proposé est nul. En effet, l'estimation de f dans la relation [30] est utilisée au début du prochain sous-pas. Par ailleurs, il est possible de prolonger la solution à $t_n + \theta_n h$, $\theta_n \geq 1$, dans le cas où cette solution reste admissible sur $[t_n, t_n + \theta_n h]$ au sens de \bar{e} , on a alors

$$t_{n+1} = t_n + \theta_n h, \quad y_{n+1} = \bar{q}(\theta_n) \tag{34}$$

où

$$\theta_n = \min (2, S \left\| \frac{\text{err}}{\epsilon} \right\|_{\infty}^{-0.25}) \tag{35}$$

Exemple

On va montrer sur un exemple simple le caractère *auto-adaptatif* de la méthode présentée. On se propose d'intégrer l'équation différentielle suivante:

$$\dot{y} = f(t, y) = \begin{cases} \pi \cos(\pi t) & \text{si } 0 \leq t \leq 0.25 \text{ ou } 0.75 \leq t \leq 1 \\ 0 & \text{sinon} \end{cases}$$

$$t \in [0, 1]$$

$$y(0) = 0$$

La solution analytique de cette équation est donnée par:

$$y(t) = \begin{cases} \sin(\pi t) & \text{si } 0 \leq t \leq 0.25 \text{ ou } 0.75 \leq t \leq 1 \\ 0.75 & \text{sinon} \end{cases}$$

La figure 4 montre comment varie le pas au voisinage des discontinuités de pente. On voit aussi que la solution donnée par (rkg) et la solution analytique coïncident sur l'intervalle d'intégration.

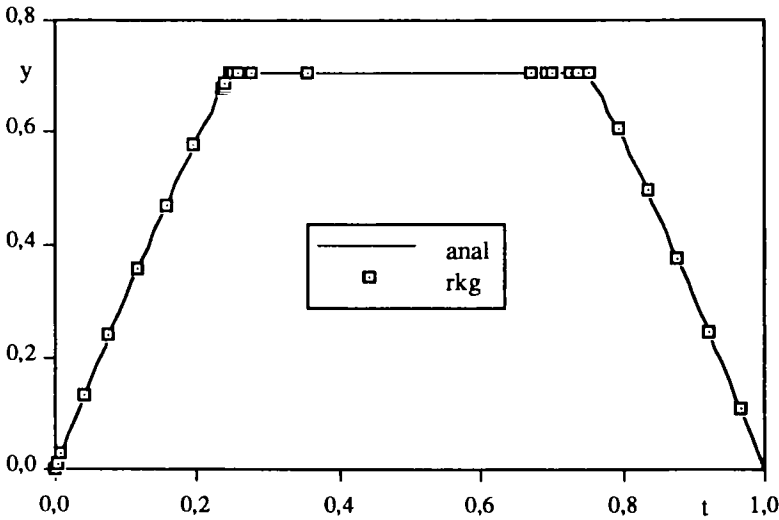


Figure 4. Auto-adaptation des pas en fonction des discontinuités de pente

3.2.5. Méthodes de Runge-Kutta emboîtées [DOR 80]

Pour contrôler la taille des sous-pas, on utilise souvent deux méthodes de Runge-Kutta, l'une d'ordre p qui donne la solution y_{n+1} , et l'autre d'ordre $p+1$ qui donne \hat{y}_{n+1} . On pose

$$\Delta = \hat{y}_{n+1} - y_{n+1} \quad [36]$$

alors Δ est une bonne approximation de l'erreur de troncature faite sur y_{n+1} , et le contrôle de la taille des sous-pas se fait alors, pour $p = 4$, par la formule:

$$h_0 = S h \left\| \frac{\Delta}{\varepsilon} \right\|_{\infty}^{-0.20} \quad [37]$$

où h et Δ correspondent au pas précédent et h_0 est la taille du sous-pas prédite pour satisfaire la précision ε .

Une méthode de ce type est souvent donnée par le tableau 1, et les relations

$$\begin{aligned}
 k_i &= f(t_n + c_i h, y_n + h \sum_j a_{ij} k_j) \\
 y_{n+1} &= y_n + h \sum_j b_j k_j \\
 \widehat{y}_{n+1} &= y_n + h \sum_j \widehat{b}_j k_j
 \end{aligned}$$

c_i	a_{ij}
y_{n+1}	b_j
\widehat{y}_{n+1}	\widehat{b}_j

Tableau 1. Méthodes de Runge-Kutta emboîtées

Le tableau 2 présente une des meilleures méthodes de cette classe (notée dopri5).

0						
<u>1</u>	<u>1</u>					
5	5					
<u>3</u>	<u>3</u>	9				
10	40	40				
<u>4</u>	<u>44</u>	<u>56</u>				
5	45	15				
<u>8</u>	<u>19372</u>	<u>25360</u>	<u>64448</u>	<u>212</u>		
9	6561	2187	6561	729		
1	<u>9017</u>	<u>355</u>	<u>46732</u>	<u>49</u>	<u>5103</u>	
1	3168	33	5247	176	18656	
1	<u>35</u>	0	<u>500</u>	<u>125</u>	<u>2187</u>	<u>11</u>
1	384		1113	192	6784	84
y_{n+1}	<u>35</u>	0	<u>500</u>	<u>125</u>	<u>2187</u>	<u>11</u>
\widehat{y}_{n+1}	384		1113	192	6784	84
y_{n+1}	<u>35</u>	0	<u>500</u>	<u>125</u>	<u>2187</u>	<u>11</u>
\widehat{y}_{n+1}	<u>5179</u>	0	<u>7571</u>	<u>393</u>	<u>92097</u>	<u>184</u>
	57600		16695	640	339200	2100

Tableau 2. Méthodes de Runge-Kutta emboîtées (de Dormand et Prince)

2.3. Méthode du point milieu modifiée

Pour intégrer le système d'équations

$$\begin{aligned}
 \dot{y} &= f(y, t), \quad t \in [t_n, t_{n+1}] \\
 y(t_n) &= y_n \quad \text{connue}
 \end{aligned}
 \tag{38}$$

cette méthode utilise m sous-pas de taille $h_m = \frac{h}{m}$. Elle s'exprime comme suit:

$$z_0 = y_n$$

$$\begin{aligned}
 z_1 &= z_0 + h_m f(t_n, z_0) \\
 &\vdots \\
 &\vdots \\
 z_{i+1} &= z_{i-1} + 2h_m f(t_n + ih, z_i) \quad i = 1, 2, \dots, m-1 \\
 \\
 y_{n+1}^m &= \frac{1}{2}(z_m + z_{m-1} + h_m f(t_n + h, z_m)) \quad [39]
 \end{aligned}$$

L'intérêt de la méthode vient du fait que l'erreur de troncature de [39] est exprimée par une série en h_m qui ne contient que des puissances paires

$$y_{n+1}^m - y(t_n + h) = \sum_{i=1}^{\infty} \alpha_i h_m^{2i} \quad [40]$$

Cette propriété est exploitée dans la partie suivante.

2.4. Méthodes d'extrapolation

Trois idées sont à la base des méthodes d'extrapolation. On considère la solution [39] obtenue par la méthode du point milieu modifiée comme une fonction de h_m ; pour $h_m = 0$ (m infini) on aurait obtenu la solution exacte à l'instant $t_n + h$, donc pour approcher cette valeur, on calcule les solutions y_{n+1}^m obtenues pour différents entiers m , et on utilise une extrapolation de Richardson vers $h_m = 0$, (figure 5). Enfin, la propriété [40] permet d'utiliser h_m^2 au lieu de h_m dans cette extrapolation [STO 80].

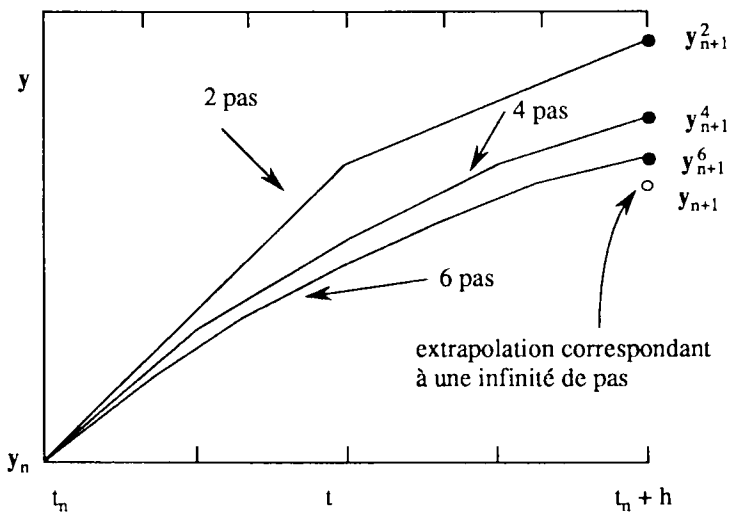


Figure 5. Méthode d'extrapolation

2.4.1. *Méthode de Bulirsch-Stoer*

Cette méthode (notée (bs)), utilise la technique décrite ci-dessus en choisissant m dans la suite

$$m = 2,6,4,8,12,16,24,32,48,64,96 \quad (n_j = 2 n_{j-1}) \quad [41]$$

A priori on ne sait pas jusqu'où on va aller dans la suite [41]; après chaque essai correspondant à une des valeurs précédentes de m on extrapole vers l'infini, ce qui fournit une solution extrapolée et une estimation de l'erreur commise. Si l'erreur est plus grande que l'erreur admissible, on refait les mêmes calculs avec la valeur suivante de m dans la suite [41]. Dans le cas contraire on passe au pas suivant et on recommence avec m = 2. Si aucun entier de la suite [41] ne donne la solution avec la précision demandée, on diminue la taille du pas et on recommence à partir de m = 2. Le choix de la suite [41] permet de réutiliser, pour chaque m dans cette suite, les estimations du second membre calculées auparavant (voir [39]).

2.4.2. *Méthode d'extrapolation*

Dans cette méthode (notée (odex)), on fait varier la taille du pas ainsi que l'ordre d'extrapolation, soit h la taille du pas k l'ordre d'extrapolation, forcément pair, on évalue y_{n+1}^m pour $m = 2,4,6,\dots,k$ puis on extrapole pour m infini. Si l'erreur obtenue ne satisfait pas la précision demandée, on réduit le pas, on augmente éventuellement l'ordre k et on recommence. Sinon on passe au pas suivant avec une taille de pas et un ordre prédits.

3. **Application aux équations de l'élastoplasticité**

Afin de comparer les différents algorithmes présentés on effectue des tests d'une part au niveau d'un point d'intégration et d'autre part au niveau d'une structure dont la discrétisation est constituée de plusieurs éléments finis.

3.1. *Calcul au niveau d'un point d'intégration*

3.1.1. *Matériau de Von-Mises en plasticité parfaite*

Le critère retenu est celui de Von-Mises en plasticité parfaite ($E = 20000$ MPa, $\nu = 0.3$, $R = 40$ MPa)

$$f(\sigma) = \sqrt{\frac{3}{2} s:s} - R \leq 0$$

où s est le déviateur de σ . La relation de comportement associée à un taux de déformation imposé $\dot{\epsilon}$ constant sur $[0, T]$ est

$$\forall t \in [0, T] \quad \dot{\sigma} = \frac{E}{1+\nu} (\dot{\epsilon} - \frac{s(t):\dot{\epsilon}}{s(t):s(t)} s(t)) + \frac{E}{3(1-2\nu)} \text{tr}(\dot{\epsilon})$$

$\dot{\epsilon}$ étant le déviateur de $\dot{\epsilon}$. L'état des contraintes initial est un cisaillement pur:

$$\sigma(0) = \sigma_0 = (\sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}) = (0, 0, 0, \frac{40}{\sqrt{3}}, 0, 0)$$

Le taux de déformation est un déviateur

$$\dot{\epsilon} = (\dot{\epsilon}_x, \dot{\epsilon}_y, \dot{\epsilon}_z, \dot{\epsilon}_{xy}, \dot{\epsilon}_{yz}, \dot{\epsilon}_{xz}) = (1, -\frac{1}{2}, -\frac{1}{2}, 0, 0, 0)$$

Ce problème admet une solution analytique $\sigma_{\text{exact}}(t)$, (voir [DRO 85]), et la mesure de l'erreur relative d'intégration par rapport à cette solution est donnée par:

$$\text{Erreur} = \sup_{t \in [0, T]} \frac{|\sigma_{\text{exact}}(t) - \sigma_{\text{calculé}}(t)|_{\infty}}{|\sigma_{\text{exact}}(t)|_{\infty}}$$

On utilise la méthode de Runge-Kutta généralisée pour calculer les contraintes à la fin du pas. Comme dans [BON 91], la non-linéarité du problème est mise en évidence par la figure 6; on calcule les contraintes pour trois précisions requises $\epsilon = 0.1, 0.01$ et 0.001 , l'erreur sur σ_x varie avec ϵ et est visible sur la figure 6 pour $\epsilon = 0.1$, par contre σ_{xy} coïncide pratiquement avec la solution analytique pour les trois précisions.

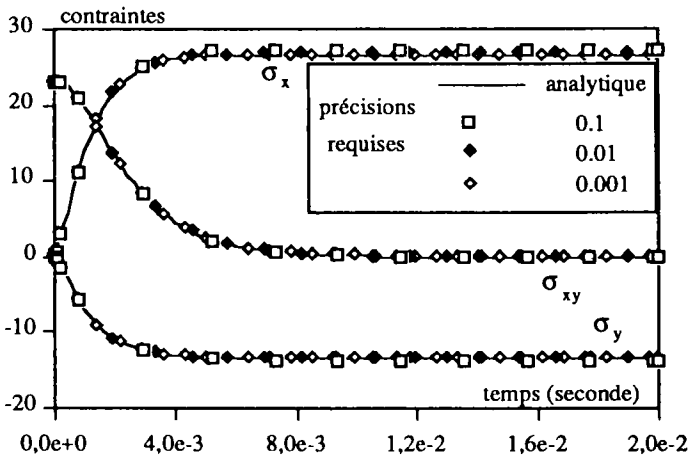


Figure 6. Evolution des contraintes

On voit sur la figure 7 qu'au début la taille des sous-pas augmente, puis au croisement des courbes σ_x et σ_{xy} où la norme $\|\cdot\|_{\infty}$ n'est pas différentiable, l'erreur atteint un maximum (figure 8). La taille des sous-pas diminue donc pour s'adapter à cette singularité.

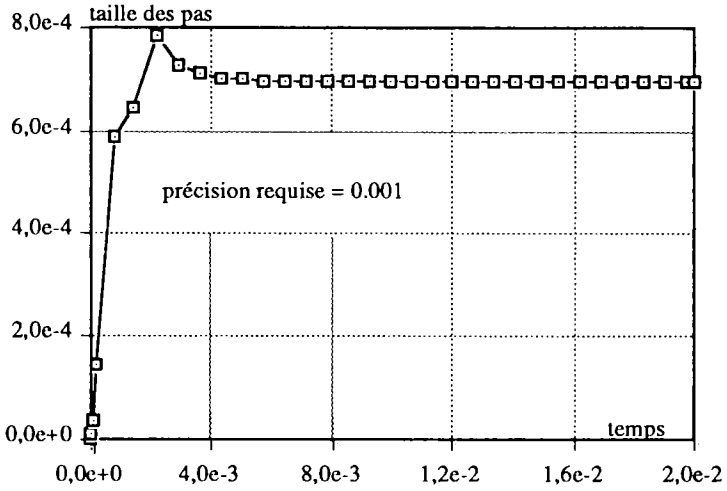


Figure 7. Evolution de la taille des pas

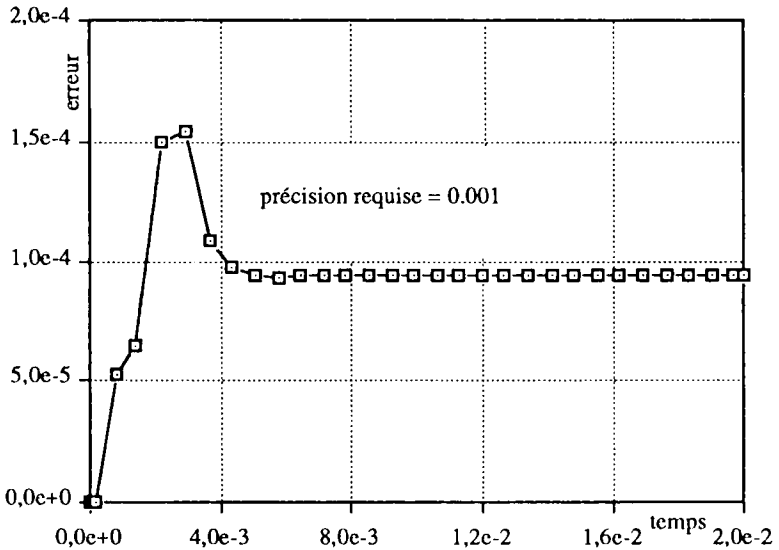


Figure 8. Evolution de l'erreur

La figure 9 montre que si la précision requise est correctement choisie, le critère de plasticité reste vérifié au cours de l'intégration. Toute correction des contraintes est donc inutile.

Sur la figure 10 on présente l'erreur effective en fonction du nombre d'évaluations du second membre de [43], ce qui est une bonne estimation du coût de

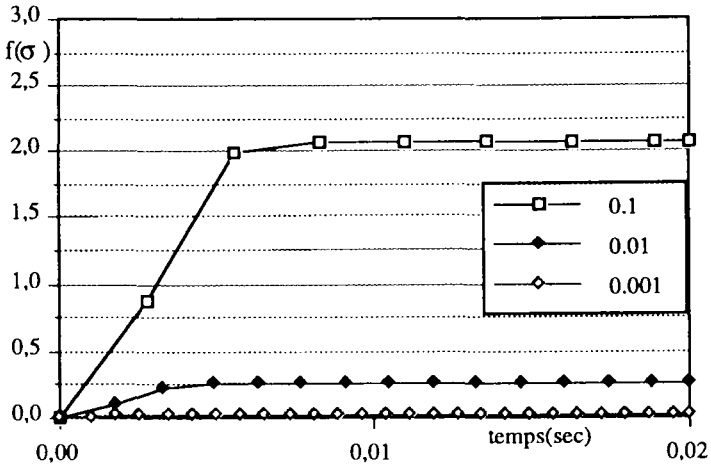


Figure 9. Evolution du critère

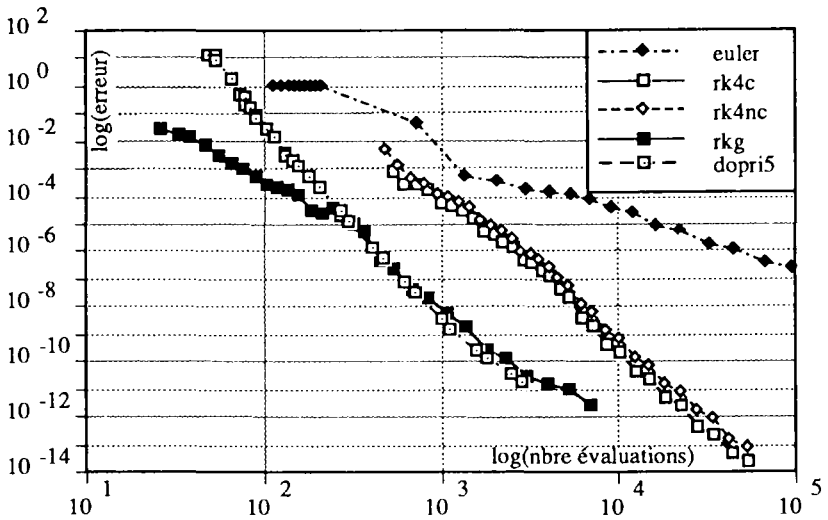


Figure 10. Erreur obtenue en fonction du nombre d'évaluations de la dérivée

la méthode. Les courbes correspondent aux méthodes présentées de type Euler et Runge-Kutta, et on voit que la méthode de Runge-Kutta généralisée donne les meilleurs résultats surtout pour des précisions faibles (inférieures à 10^{-5}) qui nous intéressent dans la pratique.

La figure 11 présente la précision obtenue par chacune des méthodes en fonction de la précision demandée. Elle montre que c'est avec (rkg) que précision demandée et précision obtenue coïncident le mieux.

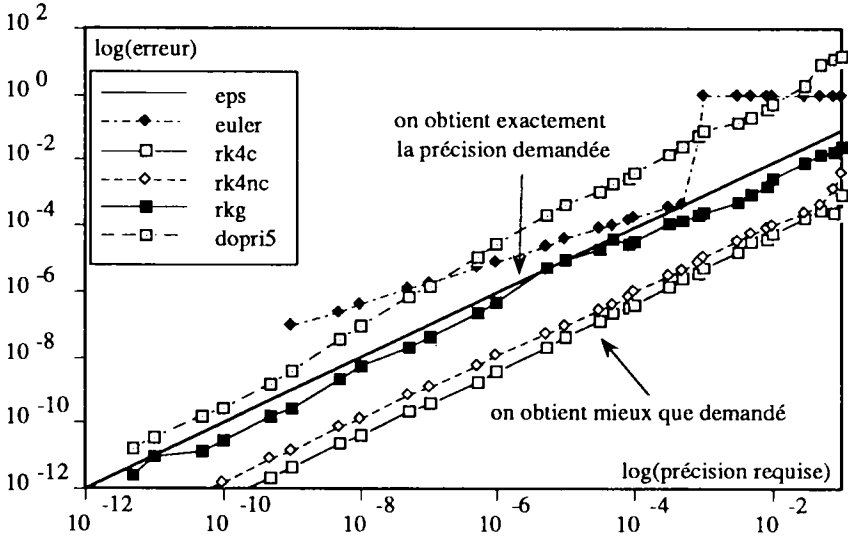


Figure 11. Erreur en fonction de la précision requise

Les figures 12 et 13 mettent en évidence les risques d'instabilité qui accompagnent l'utilisation des méthodes basées sur l'extrapolation de Richardson, et permettent de comparer ces méthodes à (rkg).

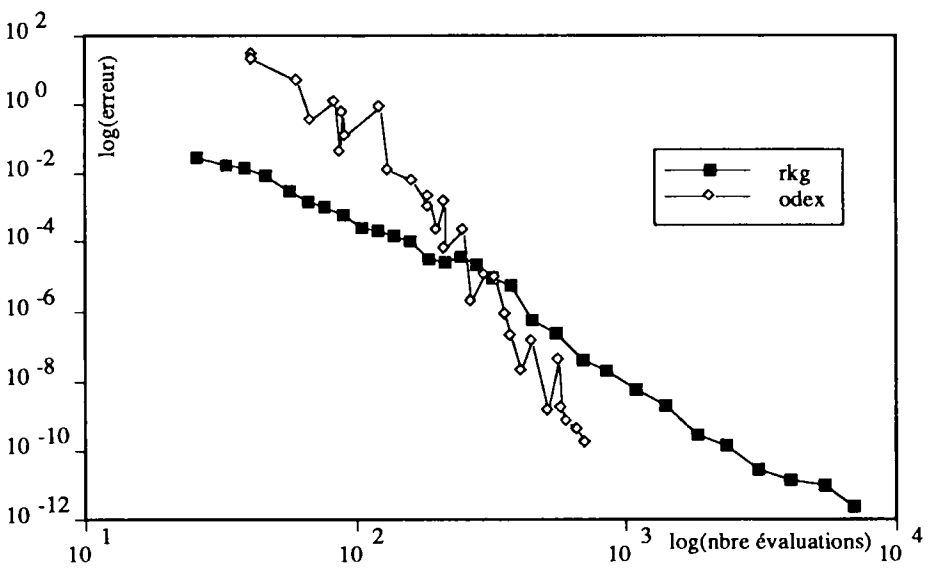


Figure 12. Erreur obtenue en fonction du nombre d'évaluations de la dérivée

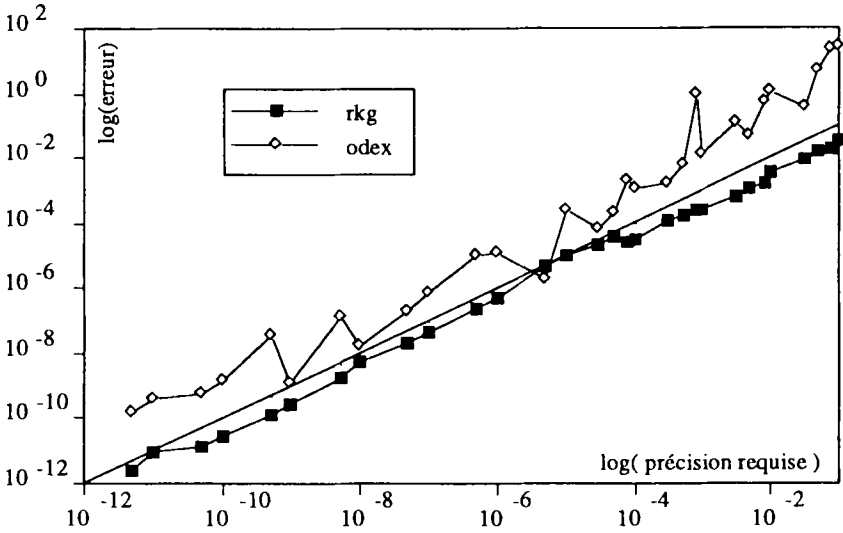


Figure 13. Erreur obtenue en fonction de la précision requise

3.1.1. Modèle de Nova

Dans ce deuxième exemple on considère le modèle de Nova (Version 82), cas associé. Ce modèle est utilisé en mécanique des sols. L'expression de la loi de comportement et les caractéristiques du sol étudié sont comme dans [BAG 93]. On exerce sur un point de la matière une compression uniforme et on se propose de calculer les contraintes correspondantes. Ce problème admet une solution analytique [BAG. 93], et l'erreur commise en utilisant l'une des méthodes présentée sera calculée par rapport à cette solution. La figure 14 présente les solutions analytique et celles obtenues par différentes méthodes pour un même temps CPU = 0.5 s, et on voit que la solution donnée par (rkg) coïncide pratiquement avec la solution analytique. En étudiant les courbes des figures 15 et 16 comme dans l'exemple précédent, on voit que la méthode de Runge-Kutta généralisée donne les meilleurs résultats. Et on remarque là encore les phénomènes d'instabilité liés à l'utilisation des méthodes d'extrapolation. On signale que les autres méthodes de type Runge-Kutta qui ne sont pas sur ces figures donnent des résultats semblables à ceux donnés par (rk4nc). Il en est de même pour (odex) par rapport à (bs).

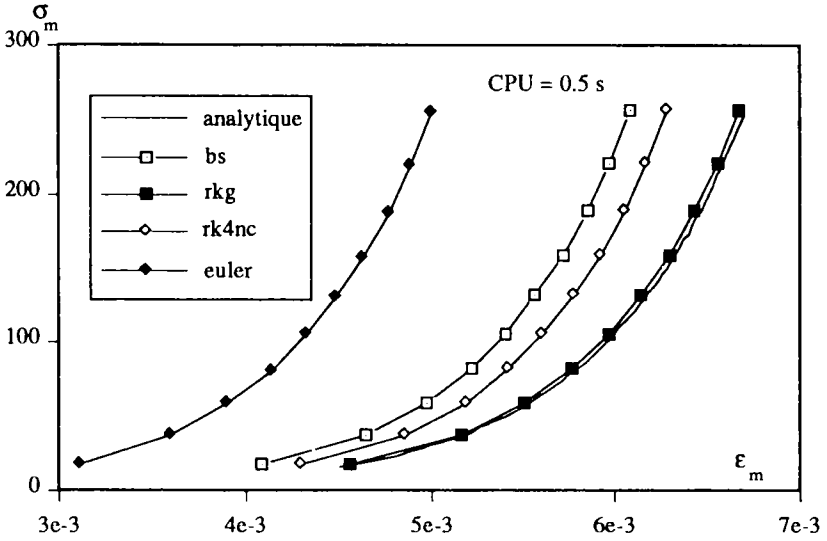


Figure 14. Contraintes en fonction des déformations

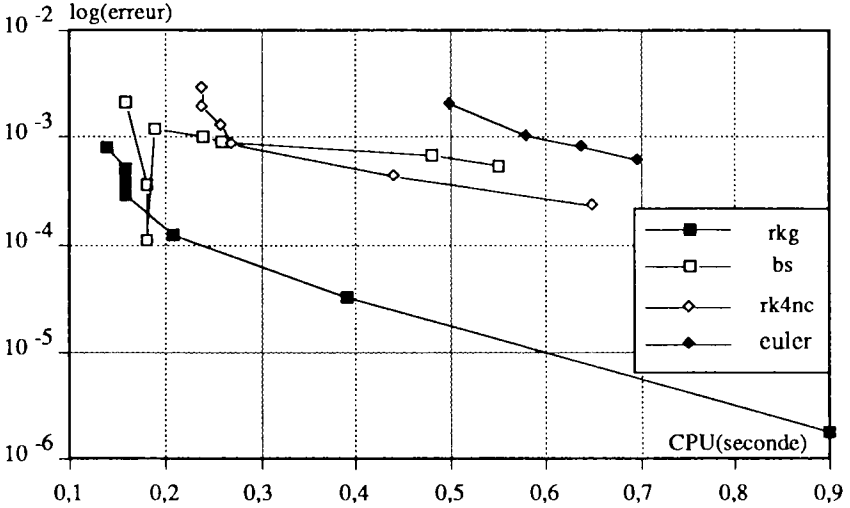


Figure 15. Erreur obtenue en fonction du temps CPU

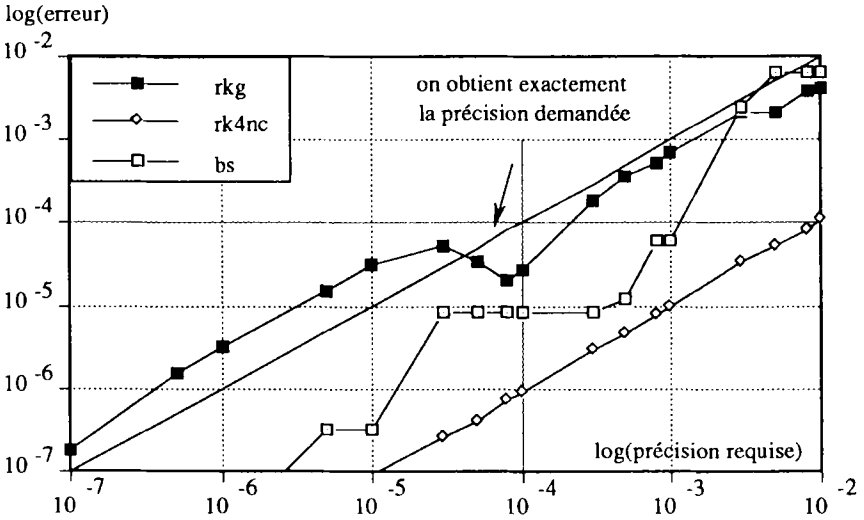


Figure 16. Erreur obtenue en fonction de la précision requise

3.2. Calcul au niveau d'une structure

On considère un matériau élastoplastique isotrope (Critère de Von-Mises) à écrouissage isotrope linéaire. Les lois de comportement qui relient vitesses de contraintes et de déformations déviatoires s'écrivent :

$$\dot{\mathbf{s}} = G(\dot{\boldsymbol{\varepsilon}} - \dot{\lambda}\mathbf{s}) \quad [42]$$

$$\dot{p} = \dot{\lambda} \sqrt{\frac{2}{3}} \phi \quad \text{où} \quad \phi = \sqrt{\mathbf{s}^T [\mathbf{P}] \mathbf{s}}$$

$$\dot{\lambda} = \frac{G \mathbf{s}^T [\mathbf{P}] \dot{\boldsymbol{\varepsilon}}}{(G + \frac{2}{3}H)\phi^2}$$

avec

$$\mathbf{s} = \langle s_x \ s_y \ s_z \ s_{xy} \ s_{yz} \ s_{xz} \rangle, \quad \text{et} \quad \boldsymbol{\varepsilon} = \langle \varepsilon_x \ \varepsilon_y \ \varepsilon_z \ \varepsilon_{xy} \ \varepsilon_{yz} \ \varepsilon_{xz} \rangle$$

et

$$[\mathbf{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Le critère de Von-Mises s'écrit :

$$f = \frac{1}{2} \mathbf{s}^T [\mathbf{P}] \mathbf{s} - \frac{1}{3} k^2(p) \leq 0 \tag{43}$$

où

$$k(p) = \sigma_c + H p \tag{44}$$

est la limite élastique courante.

A un instant t_n on suppose connues \mathbf{e}_n , \mathbf{s}_n et p_n et on se donne $\mathbf{e}_{n+1} = \mathbf{e}_n + \Delta \mathbf{e}_n$. On doit calculer \mathbf{s}_{n+1} et p_{n+1} qui correspondent à \mathbf{e}_{n+1} et à l'instant $t_{n+1} = t_n + \Delta t$, avec $\Delta t = 1$. On doit donc intégrer le système d'équations différentielles

$$\begin{aligned} \dot{\mathbf{s}} &= G(\Delta \mathbf{e}_n - \lambda \mathbf{s}) \\ \dot{p} &= \lambda \sqrt{\frac{2}{3}} \phi \end{aligned} \tag{45}$$

où

$$\lambda = \frac{G \mathbf{s}^T [\mathbf{P}] \Delta \mathbf{e}_n}{(G + \frac{2}{3}H)\phi^2} \tag{46}$$

On a supposé que le taux de déformation est constant sur le pas de chargement, donc son déviateur l'est aussi et est égal à $\Delta \mathbf{e}_n$. Si on note $\Delta \mathbf{s} = G(\Delta \mathbf{e}_n - \lambda \mathbf{s})$ et $\Delta p = \lambda \sqrt{\frac{2}{3}} \phi$, l'utilisation de la méthode de Runge-Kuta généralisée conduit à calculer $\Delta \mathbf{s}_i$ et Δp_i , $i=1,4$ (voir [16]), et la solution à l'instant t_{n+1} est donnée par:

$$\begin{aligned} \mathbf{s}_{n+1} &= \mathbf{s}_n + \frac{1}{8} (\Delta \mathbf{s}_1 + 3(\Delta \mathbf{s}_2 + \Delta \mathbf{s}_3) + \Delta \mathbf{s}_4) \\ p_{n+1} &= p_n + \frac{1}{8} (\Delta p_1 + 3(\Delta p_2 + \Delta p_3) + \Delta p_4) \end{aligned} \tag{47}$$

3.2.1. Calcul du module tangent consistant

L'utilisation du module tangent continu (matrice \mathbf{D}^∞ qui relie $\dot{\sigma}$ à $\dot{\epsilon}$) ralentit la convergence du schéma de Newton-Raphson employé pour trouver la configuration d'équilibre. Nous proposons donc d'utiliser un module tangent consistant avec l'algorithme d'intégration; c'est à dire obtenu en dérivant l'expression [47] de \mathbf{s}_{n+1} par rapport à $\Delta \mathbf{e}_n$. On est donc amené à calculer $\frac{d(\Delta \mathbf{s}_i)}{d(\Delta \mathbf{e}_n)}$, $i=1,4$. D'après [46] on a :

$$\frac{\partial \lambda}{\partial (\Delta \mathbf{e}_n)} = \frac{G}{(G + \frac{2}{3}H)\phi^2} (\mathbf{s}^T [\mathbf{P}] + \frac{\partial \mathbf{s}^T}{\partial (\Delta \mathbf{e}_n)} [\mathbf{P}] \Delta \mathbf{e}_n) - \frac{2\lambda}{\phi^2} \mathbf{s}^T [\mathbf{P}] \frac{\partial \mathbf{s}}{\partial (\Delta \mathbf{e}_n)}$$

et

$$\frac{\partial(\Delta s)}{\partial(\Delta \mathbf{e}_n)} = G ([I] - \frac{\partial \lambda}{\partial(\Delta \mathbf{e}_n)} \mathbf{s} - \lambda \frac{\partial \mathbf{s}}{\partial(\Delta \mathbf{e}_n)})$$

Enfin on a:

$$\frac{\partial s_{n+1}}{\partial(\Delta \mathbf{e}_n)} = \frac{1}{8} \left(\frac{\partial(\Delta s_1)}{\partial(\Delta \mathbf{e}_n)} + 3 \left(\frac{\partial(\Delta s_2)}{\partial(\Delta \mathbf{e}_n)} + \frac{\partial(\Delta s_3)}{\partial(\Delta \mathbf{e}_n)} \right) + \frac{\partial(\Delta s_4)}{\partial(\Delta \mathbf{e}_n)} \right)$$

On obtient:

$$\sigma_{n+1} = s_{n+1} + \frac{1}{3} [U] (\sigma_n + K \Delta \mathbf{e}_n)$$

et

$$\frac{\partial \sigma_{n+1}}{\partial(\Delta \mathbf{e}_n)} = \frac{\partial s_{n+1}}{\partial(\Delta \mathbf{e}_n)} - \frac{1}{3} \left(\frac{\partial s_{n+1}}{\partial(\Delta \mathbf{e}_n)} - K [I] \right) [U]$$

où

$$K = \frac{E}{(1 - 2\nu)}$$

et

$$[U] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Remarque: Pour diminuer le coût de la méthode on peut utiliser certaines propriétés liées au cas particulier étudié, par exemple on a

$$\phi = \sqrt{\mathbf{s}^T [P] \mathbf{s}} = \sqrt{\frac{2}{3}} k(p)$$

où on a supposé que les contraintes restent sur la surface d'écoulement. On peut aussi utiliser des relations du type:

$$s_2^T [P] \Delta \mathbf{e}_n = s_1^T [P] \Delta \mathbf{e}_n + \frac{G}{3} (\Delta \mathbf{e}_n^T [P] \Delta \mathbf{e}_n + \lambda s_1^T [P] \Delta \mathbf{e}_n)$$

avec

$$s_2 = s_1 + \frac{1}{3} \Delta s_1 = s_1 + \frac{G}{3} (\Delta \mathbf{e}_n + \lambda s_1)$$

Nous avons appliqué l'algorithme présenté ci-dessus et basé sur la méthode de Runge-Kutta généralisée (à sous-pas variables) à un problème proposé par le GRECO-CNRS "Grandes Déformations et Endommagement". Ce problème consiste à étudier le comportement en grandes transformations d'une poutre courte ou épaisse ou de faible élanement. La figure 17 présente le maillage utilisé; 48 éléments à 8 noeuds et 4 points d'intégration, les caractéristiques mécaniques du matériau, le chargement (déplacement imposé) et les conditions aux limites. On adopte l'hypothèse des déformations planes et la dérivée objective de Jaumann. L'intérêt du test est qu'il présente les principales difficultés des grandes transformations: grands déplacements, grandes déformations et grandes rotations. Dans le tableau suivant, on présente les résultats obtenus par l'algorithme proposé ainsi que ceux donnés par la méthode du retour radial [SIM 85]. Des résultats obtenus par d'autres laboratoires sont aussi mentionnés dans ce tableau.

$\bar{\epsilon}^P$ la déformation plastique équivalente

Point de Gauss	Code	σ_x	σ_y	σ_{xy}	$\bar{\epsilon}^P$
1	Référence	73.63	- 1.127	2.095	0.2484
	INSA	73.66	- 1.126	2.099	0.2487
	CEMEF	73.64	- 1.128	2.096	0.2485
	Retour rad.	73.59	- 1.120	2.114	0.2480
	rkg	73.60	- 1.131	2.086	0.2481
2	Référence	58.18	2.663	14.82	0.1451
	INSA	58.18	2.643	14.82	0.1451
	CEMEF	58.18	2.665	14.83	0.1451
	Retour rad.	58.13	2.532	14.78	0.1453
	rkg	58.20	2.656	14.81	0.1452
3	Référence	- 75.71	- 11.68	- 13.95	0.2049
	INSA	- 75.75	- 11.70	- 13.97	0.2051
	CEMEF	- 75.71	- 11.68	- 13.96	0.2049
	Retour rad.	- 75.68	- 11.64	- 13.95	0.2050
	rkg	- 75.67	- 11.63	- 13.94	0.2049
4	Référence	0.901	- 0.366	- 0.100	0.0000
	INSA	0.889	- 0.364	- 0.099	0.0000
	CEMEF	0.898	- 0.366	- 0.100	0.0000
	Retour rad.	0.900	- 0.368	- 0.101	0.0000
	rkg	0.924	- 0.369	- 0.101	0.0000
5	Référence	35.46	61.15	3.610	0.0545
	INSA	35.52	61.09	3.650	0.0545
	CEMEF	35.47	61.10	3.602	0.0545
	Retour rad.	35.50	61.38	3.835	0.0544
	rkg	35.38	61.49	3.695	0.0543

Tableau 3. Résultats numériques du test sur une structure

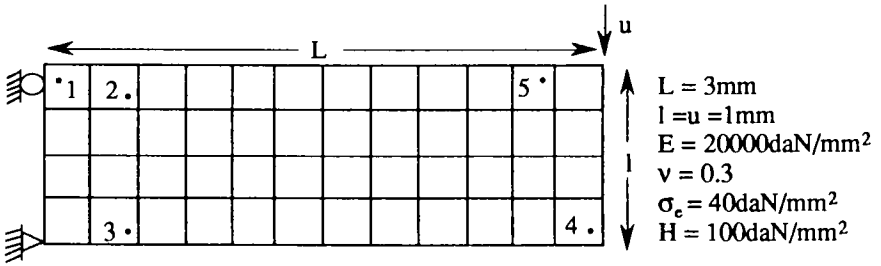


Figure 17. Maillage et données mécaniques

On désigne par (Référence) les valeurs calculées avec un très grand nombre de pas de chargement (500 pas). On obtient ces mêmes valeurs en utilisant l'une ou l'autre des méthodes présentées.

On signale que 50 pas de chargement de même taille ont été utilisés dans le cas de l'algorithme proposé pour obtenir la convergence du schéma de Newton-Raphson, et que cela s'est fait en près de 1mn 25s de temps CPU sur une station VAX 4000.90. La convergence a été obtenue avec le même nombre de pas mais en 1mn 40s dans le cas du retour radial. Ceci devant 160 [BRA 86] et 450 [PIN 90] pas de chargement dans d'autres laboratoires.

Pour les autres méthodes présentées dans ce papier, la convergence a nécessité encore 50 pas de chargement et en temps CPU près de 1mn 50s pour les méthodes de type Runge-Kutta avec module tangent constant (paragraphe 3.2.1), près de 2mn 15s si ce module est calculé en perturbant chaque composante de l'accroissement de déformation [ELM 89] et près de 3mn pour les méthodes d'extrapolation; dans ce cas il a été difficile de trouver une expression du module tangent constant, on l'a donc calculé par perturbation.

On a aussi utilisé un chemin de chargement particulier (figure 19) où le pas de chargement est calculé; [CRI 91] par la relation

$$\Delta t_{n+1} = \Delta t_n \left(\frac{I_{opt}}{I_n} \right)^2$$

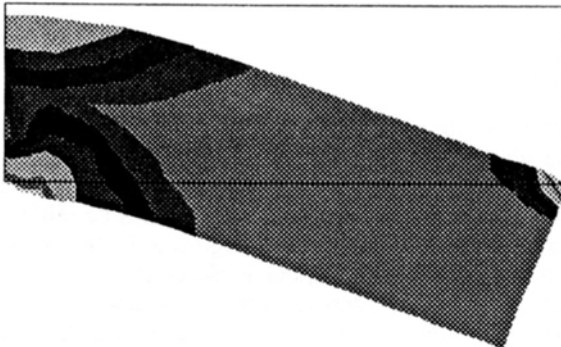


Figure 18. Poutre déformée

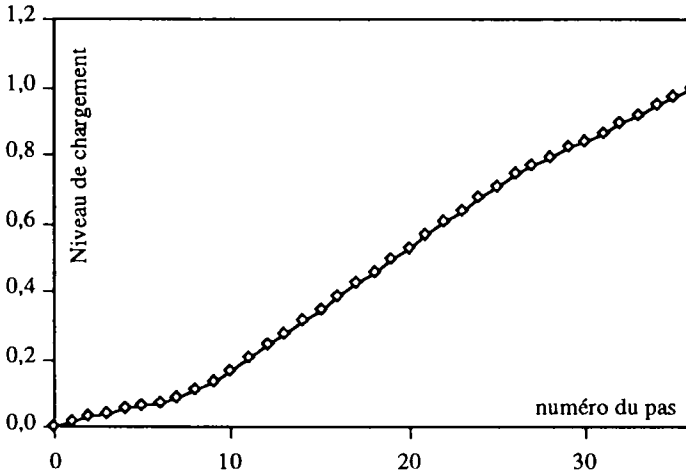


Figure 19. Evolution du chargement

Δ_n étant le pas de chargement à l'instant t_n , I_n le nombre d'itérations nécessaire pour l'accomplir, la donnée I_{opt} est le nombre d'itérations optimal et Δ_{n+1} le pas suivant. Dans ce cas, avec 36 pas et près de 1mn en temps CPU, l'algorithme proposé donne des résultats très voisins de ceux obtenus plus haut (tableau 4).

Point de Gauss	Code	σ_x	σ_y	σ_{xy}	$\bar{\epsilon}^p$
1	rkg(36pas)	73.67	- 1.126	2.089	0.2487
2	rkg(36pas)	58.17	2.640	14.83	0.1452
3	rkg(36pas)	- 75.62	- 11.65	- 14.00	0.2047
4	rkg(36pas)	0.869	- 0.357	- 0.096	0.0000
5	rkg(36pas)	35.50	60.85	3.553	0.0544

Tableau 4. Résultats numériques du test sur une structure

4. Conclusion

L'algorithme proposé, basé sur la méthode de Runge-Kutta généralisée, peut concurrencer la plupart des algorithmes utilisés pour l'intégration des équations différentielles élastoplastiques et notamment la méthode du retour radial.

La subdivision des pas de chargement en sous-pas de taille variable selon les difficultés rencontrées lors de l'intégration augmente l'efficacité des algorithmes et permet d'éviter tout dépassement du critère de plasticité.

Les exemples étudiés ici ont surtout permis de valider l'algorithme proposé, car le champ de son application est beaucoup plus large et son efficacité apparaît encore plus importante dans des problèmes où le calcul du second membre est très coûteux.

Son utilisation pour résoudre des équations du second ordre est très intéressante et dans ce cas on peut facilement voir que le coût de l'estimation de l'erreur est pratiquement nul même si on prolonge la solution.

Dans cet algorithme on utilise une méthode d'ordre quatre et l'estimation d'erreur utilisée est seulement d'ordre trois. A priori on peut dire que c'est le prix de la diminution du coût de l'estimation d'erreur, mais on remarque que cela évite souvent la prédiction de pas trop petits, source des erreurs d'arrondi.

Pour des lois de comportement simples on peut calculer un module tangent consistant qui permet une convergence rapide du processus de Newton-Raphson en éliminant les erreurs supplémentaires introduites par l'utilisation du module tangent continu. Si la loi de comportement est compliquée on peut toujours obtenir un module tangent consistant D_c^{ep} en perturbant chaque composante de l'accroissement de déformation et en calculant les contraintes correspondantes. On a ainsi

$$D_c^{ep} = \frac{\partial \sigma_{n+1}}{\partial (\Delta \epsilon_n)}$$

Il est clair que le coût de cette technique peut être important, mais on peut le réduire en utilisant des méthodes d'intégration efficaces telles que celles présentées dans cet article.

5. Bibliographie

- [BAG 93] Baghdadi A. H., Modélisation en grandes déformations des solides et des sols-application aux fondations, Thèse de Doctorat, Div. Modèles Numériques en Mécanique, Université de Technologie de Compiègne, 1993.
- [BON 91] Bonnelli S., Debordes O. et Degoutte G., Intégration numérique de lois de comportement élastoplastiques. Multimécanismes pour les Géomatériaux, 3ème Symposium International sur la plasticité et ses applications, Grenoble. Août 1991.
- [BRA 86] Braudel H. J., Abouaf M. and Chenot J.L., An implicit incrementally objective formulation for the solution of Elastoplastic problems at finite strain by the F.E.M., *Computers and Structures*, vol 24, 6, pp 825-843, 1986.
- [BUT 87] Butcher J.C., *The numerical analysis of ordinary differential equations*. Wiley.1987.
- [CRI 91] Crisfield M.A., *Non-linear Finite Element Analysis of Solids and Structures*. Volume 1. Willey. 1991.
- [CRO 84] Crouzeix M. et Mignot A.L., *Analyse numérique des équations différentielles*. Masson.1984.
- [DOR 80] Dormand J. R. and Prince P.J., A family of embedded Runge-Kutta formulae, *J. Comp. Math.*,6,19-20,1980.
- [DRO 85] Drobysz S., *Intégration numérique de lois de comportement et construction automatique de problèmes tests en grandes transformations élastoplastiques*, Thèse, Université de Nantes 1985.
- [ELM 89] El Mouatassim M., *Modélisation en grandes transformations des solides massifs par éléments finis*, Thèse de Doctorat, Div. Modèles Numériques en Mécanique, Université de Technologie de Compiègne, Novembre 1989.
- [GEA 71] Gear C.W., *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs N. J., Printice Hall. 1971.
- [PIN 90] Pinto Y., Contact et frottement en Grandes Déformations Plastiques, Application au serre-flan en emboutissage, Thèse de Doctorat, Université de Province (Aix-Marseille I), Decembre 1990.

[POL 89] Polat M.U. and Dokainsh M.A., An automatic subincrementation scheme for accurate integration of elasto-plastic constitutive relations. *Computers & Structures*, vol 31, 3, pp 339-347, 1989.

[PRE 86] Press W.H., Flannery B. P., Teukolsky S. A. and Vetterling W. T., *Numerical Recipes, The Art of Scientific Computing*. Combridge University Press. 1986.

[SIM 85] Simo J. C. and Taylor R. L., Consistant tangent Operators for rate-independent Elastoplasticity. *Comp. Meth. Appl. Mech. Eng.*,48,101-118.1984.

[SIM 86] Simo J.C. and Taylor R. L., A return Mapping Algorithm for plane stress Elastoplasticity. *inter. j. num. methods. eng.*, 22, 649-670, 1986.

[SLO 87] Sloan S.W., Substepping schemas for the numerical integration of elastoplastic stress-strain relations. *inter. j. num. methods. eng.*, 24, 893-911. 1987.

[STO 80] Stoer J. and Bulirsch R., *Introduction to Numerical Analysis*. New York. Springer Verlag.1980.

Annexe

Le polynôme d'interpolation de Lagrange qui définit \bar{f} s'écrit en fonction de $\theta = \frac{t - t_n}{h}$

$$\bar{f}(t) = \bar{p}(\theta) = p_{10}(\theta)\bar{f}(t_n) + p_{20}(\theta)\bar{f}(t_n + \frac{h}{3}) + p_{30}(\theta)\bar{f}(t_n + \frac{2h}{3}) + p_{40}(\theta)\bar{f}(t_{n+1}) \quad [A.1]$$

avec

$$\begin{aligned} p_{10} &= -\frac{9}{2}\theta^3 + 9\theta^2 - \frac{11}{2}\theta + 1 \\ p_{20} &= \frac{27}{2}\theta^3 - \frac{45}{2}\theta^2 + 9\theta \\ p_{30} &= -\frac{27}{2}\theta^3 + 18\theta^2 - \frac{9}{2}\theta \\ p_{40} &= \frac{9}{2}\theta^3 - \frac{9}{2}\theta^2 + \theta \end{aligned} \quad [A.2]$$

En utilisant les relations [22] et [A.2] dans [A.1] et en développant les calculs on arrive à l'expression de \bar{p} donnée par [24].

D'autre part, d'après [26] on a

$$\bar{q}(1) = y_n + h \left(\frac{1}{3}\bar{a}_2 + \frac{1}{2}\bar{a}_1 + \bar{a}_0 \right) \quad [A.3]$$

et en substituons l'expression [25] des \bar{a}_i dans [A.3], on aboutit à la relation [27].

Reste à démontrer la relation [23], pour cela on commence par montrer que cette relation est vérifiée pour $t = t_n, t_n + \frac{h}{3}, t_n + 2\frac{h}{3}$, et $t_n + h$. Pour simplifier les notations on pose:

$$\mathbf{f} = \mathbf{f}(t_n, \mathbf{y}(t_n)), \quad \dot{\mathbf{f}}_t = \frac{\partial \mathbf{f}}{\partial t}(t_n, \mathbf{y}(t_n)),$$

$$\ddot{\mathbf{f}}_{t^2} = \frac{\partial^2 \mathbf{f}}{\partial t^2}(t_n, \mathbf{y}(t_n)), \quad \ddot{\mathbf{f}}_{t\mathbf{y}} = \frac{\partial^2 \mathbf{f}}{\partial t \partial \mathbf{y}}(t_n, \mathbf{y}(t_n)), \dots \quad [\text{A.4}]$$

On voit facilement que l'on a :

$$\mathbf{f}(t_n + \varepsilon h, \mathbf{y}(t_n + \varepsilon h)) = \mathbf{f} + (\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y)\varepsilon h +$$

$$+ (\ddot{\mathbf{f}}_{t^2} + 2\mathbf{f}\ddot{\mathbf{f}}_{t\mathbf{y}} + \dot{\mathbf{f}}_t^2 + (\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y)\dot{\mathbf{f}}_y) \frac{(\varepsilon h)^2}{2} + O(h^3) \quad [\text{A.5}]$$

Par ailleurs, en utilisant les expressions [16] des \mathbf{k}_i , on a

$$\mathbf{k}_1 = \mathbf{f}$$

$$\mathbf{k}_2 = \mathbf{f} + (\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y) \frac{h}{3} + (\ddot{\mathbf{f}}_{t^2} + 2\mathbf{f}\ddot{\mathbf{f}}_{t\mathbf{y}} + \dot{\mathbf{f}}_t^2 + \dot{\mathbf{f}}_y^2) \frac{h^2}{18} + O(h^3)$$

$$\mathbf{k}_3 = \mathbf{f} + (\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y) \frac{2h}{3} + \left(\frac{3}{2}(\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y)\dot{\mathbf{f}}_y + 2\mathbf{f}\ddot{\mathbf{f}}_{t\mathbf{y}} + \dot{\mathbf{f}}_t^2 + \dot{\mathbf{f}}_y^2 \right) \frac{4h^2}{18} + O(h^3)$$

$$\mathbf{k}_4 = \mathbf{f} + (\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y)h + (2(\dot{\mathbf{f}}_t + \mathbf{f}\dot{\mathbf{f}}_y)\dot{\mathbf{f}}_y + 3\dot{\mathbf{f}}_t^2 + 6\mathbf{f}\ddot{\mathbf{f}}_{t\mathbf{y}} + 3\dot{\mathbf{f}}_y^2) \frac{3h^2}{18} + O(h^3)$$

Ces expressions des \mathbf{k}_i , ensemble avec les relations [22] et [A.5], impliquent alors

$$\bar{\mathbf{f}}(t_n) = \mathbf{f}$$

$$\bar{\mathbf{f}}\left(t_n + \frac{h}{3}\right) = \mathbf{f}\left(t_n + \frac{h}{3}, \mathbf{y}\left(t_n + \frac{h}{3}\right)\right) + O(h^3)$$

$$\bar{\mathbf{f}}\left(t_n + \frac{2h}{3}\right) = \mathbf{f}\left(t_n + \frac{2h}{3}, \mathbf{y}\left(t_n + \frac{2h}{3}\right)\right) + O(h^3)$$

$$\bar{\mathbf{f}}(t_n + h) = \mathbf{f}(t_n + h, \mathbf{y}(t_n + h)) + O(h^3)$$

Par substitution de ces relations dans [A.1], on a

$$\bar{\mathbf{f}}(t) = \bar{\mathbf{p}}(\theta) = p_{10}(\theta)\mathbf{f}(t_n, \mathbf{y}(t_n)) + p_{20}(\theta)\mathbf{f}\left(t_n + \frac{h}{3}, \mathbf{y}\left(t_n + \frac{h}{3}\right)\right)$$

$$+ p_{30}(\theta)\mathbf{f}\left(t_n + \frac{2h}{3}, \mathbf{y}\left(t_n + \frac{2h}{3}\right)\right) + p_{40}(\theta)\mathbf{f}(t_{n+1}, \mathbf{y}(t_{n+1})) + O(h^3) \quad [\text{A.8}]$$

Or, un résultat de l'interpolation de Lagrange d'ordre quatre montre que l'on a

$$\begin{aligned} \mathbf{f}(t, \mathbf{y}(t)) = & p_{10}(\theta) \mathbf{f}(t_n, \mathbf{y}(t_n)) + p_{20}(\theta) \mathbf{f}\left(t_n + \frac{h}{3}, \mathbf{y}\left(t_n + \frac{h}{3}\right)\right) \\ & + p_{30}(\theta) \mathbf{f}\left(t_n + \frac{2h}{3}, \mathbf{y}\left(t_n + \frac{2h}{3}\right)\right) + p_{40}(\theta) \mathbf{f}(t_{n+1}, \mathbf{y}(t_{n+1})) + O(h^5) \quad [\text{A.9}] \end{aligned}$$

et en utilisant ces deux dernières relations, on obtient [23].