

Manuel de Référence
Fascicule R5.03 : Mécanique non linéaire
Document : R5.03.01

Algorithme non linéaire quasi-statique (opérateur STAT_NON_LINE)

Résumé :

L'opérateur `STAT_NON_LINE` [U4.51.03] du *Code_Aster* permet dans le cas d'une sollicitation quasi-statique d'intégrer divers types de non-linéarités provenant du comportement du matériau ou de grandes transformations géométriques. On décrit ici l'algorithme de résolution global employé. L'intégration des relations de comportement proprement dites est décrite dans d'autres documents, comme [R5.03.02] pour l'élasto-plasticité, auquel on pourra se reporter pour des exemples. Pour les calculs en grandes transformations géométriques, on pourra consulter par exemple le document [R5.03.20] sur l'élasticité non linéaire en grands déplacements, ou le document [R5.03.21] sur la thermoélastoplasticité à écrouissage isotrope.

Table des matières

1	Présentation.....	3
1.1	Généralités sur les relations de comportement.....	3
1.2	Position du problème quasi-statique non linéaire.....	4
2	Méthode de résolution	6
2.1	Principe de la méthode de NEWTON.....	6
2.2	Adaptation de la méthode de NEWTON au problème posé	7
2.2.1	Résolution sans conditions aux limites dualisées	7
2.2.2	Résolution avec des conditions aux limites dualisées	9
2.2.2.1	Phase de prédiction	9
2.2.2.2	Boucle sur les itérations de NEWTON.....	11
2.2.2.3	RIGI_MECA_TANG et FULL_MECA	16
2.2.3	Cas des chargements suiveurs.....	16
2.3	Recherche linéaire.....	17
2.3.1	Principe.....	17
2.3.2	Minimisation d'une fonctionnelle	18
2.3.3	Méthode de minimisation	18
2.3.4	Application à la minimisation de l'énergie	20
2.3.5	Détermination du pas d'avancement.....	21
2.4	Algorithme de STAT_NON_LINE	23
3	Pilotage	24
4	Grandes déformations	24
4.1	Objectif.....	24
4.2	Grandes déformations plastiques.....	25
4.3	Fonctionnalité PETIT_REAC.....	26
5	Bibliographie	28

1 Présentation

1.1 Généralités sur les relations de comportement

STAT_NON_LINE est l'opérateur du Code_Aster permettant d'effectuer des calculs mécaniques non linéaires lorsque les effets d'inertie sont négligés.

Le calcul ne porte a priori que sur les variables mécaniques (déplacements, contraintes, variables internes) en excluant tout couplage avec d'autres phénomènes physiques (thermique, ...). Par conséquent, les champs associés influant sur le comportement mécanique (champs thermiques, métallurgiques) sont calculés au préalable par d'autres opérateurs (THER_LINEAIRE [U4.33.01], THER_NON_LINE [U4.33.02]), voire par d'autres codes (SYRTHES, SATURNE, ...).

Cette hypothèse souffre une exception en ce qui concerne la modélisation thermo-hydro-mécanique (modélisation dite 'THM') pour laquelle STAT_NON_LINE traite l'ensemble du problème couplé des équations de diffusion de la thermique, de la pression du fluide et de l'équilibre mécanique [R7.01.10].

Dans STAT_NON_LINE, deux familles de comportements sont disponibles :

- celle qui correspond au mot-clé facteur COMP_ELAS (COMportement ELastique) conduit au travers de l'équation d'équilibre à un système non linéaire dépendant explicitement du champ de déplacements \mathbf{u} par rapport à la configuration de référence, et paramétré par l'instant de calcul (à travers entre autres l'évolution thermique). Pour plus de détails, on pourra se reporter, par exemple, au document [R5.03.20] concernant l'élasticité en grandes transformations (hyper-élasticité), ou le document [R5.03.21] sur la thermoélastoplasticité à écrouissage isotrope,
- l'autre famille, qui correspond au mot-clé facteur COMP_INCR (COMportement INCRémental), est associée à des relations de comportement exprimées par une équation différentielle implicite (par exemple l'élasto-plasticité, la visco-plasticité, l'hypo-élasticité). Dans ce cas, la relation de comportement est intégrée comme présentée par exemple en [R5.03.02] : en reliant un incrément de déplacement $\Delta\mathbf{u}$ calculé à partir d'un état mécanique donné (l'état mécanique étant représenté par un champ de déplacements \mathbf{u} , un champ de contraintes $\boldsymbol{\sigma}$ et un champ de variables internes $\boldsymbol{\alpha}$) au champ de contraintes à l'instant t du calcul. L'équation d'équilibre conduit donc à un système non linéaire en $\Delta\mathbf{u}$, mais qui est également paramétré par l'instant de calcul à travers les données du problème (variation du chargement mécanique et évolution thermique par exemple).

Dans les deux cas, on calcule la solution de proche en proche. Ce n'est théoriquement pas indispensable dans le cas élastique non linéaire, mais il se peut que la non linéarité de la solution recherchée soit trop forte pour l'algorithme de résolution utilisé, et qu'il soit indispensable, pour des raisons numériques, d'opérer pas à pas.

Il faut néanmoins avoir à l'esprit la différence fondamentale entre les deux approches. Le cas élastique suppose l'existence d'un état de référence par rapport auquel la déformation élastique est écrite : cet état correspond à un état sans déformation ni contrainte. C'est la valeur "absolue" du chargement qui crée la déformation. Le cas incrémental s'appuie sur l'état précédemment calculé et "oublie" toute référence aux états antérieurs hormis celle donnée par les variables internes. Dans ce cas, c'est la variation du chargement qui modifie l'état du système : en particulier, il faut une variation du champ de température pour créer des déformations thermiques.

1.2 Position du problème quasi-statique non linéaire

En conséquence du [§1.1], on voit qu'il est légitime de considérer que le problème non linéaire que l'on traite a comme inconnue un déplacement et est paramétré par le temps. Soit donc le problème non linéaire quasi-statique suivant :

$$\begin{cases} \mathbf{v}^T \mathbf{R}(\mathbf{u}, t) = \mathbf{v}^T \mathbf{L}(t) & \forall \mathbf{v} \text{ tel que } \mathbf{B}\mathbf{v} = 0 \\ \mathbf{B}\mathbf{u} = \mathbf{u}^d(t) \end{cases}$$

où :

- t représente la variable d'instant,
- \mathbf{u} le champ de déplacement pris à partir d'une configuration de référence,
- \mathbf{L} le chargement mécanique auquel est soumis la structure (pression, force imposée, ...),
- la relation $\mathbf{B}\mathbf{u} = \mathbf{u}^d(t)$ correspond aux conditions aux limites imposées en déplacements (déplacements imposés, liaisons entre degrés de liberté, ...) : \mathbf{B} est un opérateur linéaire de l'espace des champs de déplacements sur un espace de fonctions définies sur une partie du bord de la structure, \mathbf{u}^d est une fonction donnée sur cette partie,
- \mathbf{R} représente les forces internes du problème de mécanique quasi-statique non linéaire (dans le cas linéaire, on a $\mathbf{R}(\mathbf{u}, t) = \mathbf{K}\mathbf{u}$, où \mathbf{K} est la matrice de rigidité de la structure). La notation $\mathbf{R}(\mathbf{u}, t)$ est un raccourci que l'on utilisera dans la suite.

En fait, plus précisément, $\mathbf{R}(\mathbf{u}, t)$ est relié au champ de contraintes $\boldsymbol{\sigma}$ par l'opérateur du travail des déformations virtuelles \mathbf{Q}^T [§ 2.2.1] suivant la relation :

$$\mathbf{R}(\mathbf{u}, t) = \mathbf{Q}^T(\mathbf{u}) : \boldsymbol{\sigma} \quad \text{éq 1.2-1}$$

En petits déplacements, \mathbf{Q}^T est indépendant des déplacements ; pour les grands déplacements, se reporter à [R5.03.20]. Dans toute la suite, on se placera dans l'hypothèse des petits déplacements et petites déformations.

Le champ de contraintes $\boldsymbol{\sigma}_i$ à l'instant t_i s'écrit $\boldsymbol{\sigma}(\mathbf{u}_i, T_i, t_i, \mathbf{H}_{i-1})$, si l'on note T_i le champ de températures et \mathbf{H}_{i-1} l'histoire passée de la structure. Pour les comportements élastiques, l'histoire n'intervient pas : l'ensemble \mathbf{H}_{i-1} est donc vide. Pour les comportements incrémentaux, l'histoire est l'ensemble des états (champs de déplacements, contraintes et variables internes) à l'instant précédent : $\mathbf{H}_{i-1} = \{\mathbf{u}_{i-1}, \boldsymbol{\sigma}_{i-1}, \alpha_{i-1}, t_{i-1}\}$.

Dans le cas général, la dépendance de l'opérateur \mathbf{R} est, comme nous l'avons vu dans le [§1.1], implicite par rapport au temps : elle résulte de l'intégration de la relation de comportement dans le temps (pour les problèmes d'élasto-plasticité par exemple). La dépendance explicite par rapport au temps est très rare : elle apparaît notamment dans le cas de relations de comportement prenant en compte un phénomène d'écrouissage par le temps dit time-hardening.

La dualisation des conditions aux limites de Dirichlet $\mathbf{Bu} = \mathbf{u}^d(t)$ conduit au problème suivant [R3.03.01] :

$$\begin{cases} \mathbf{R}(\mathbf{u}, t) + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{L}(t) \\ \mathbf{Bu} = \mathbf{u}^d(t) \end{cases} \quad \text{éq 1.2-2}$$

Les inconnues sont à présent à tout instant t le couple $(\mathbf{u}, \boldsymbol{\lambda})$, où $\boldsymbol{\lambda}$ représente les "multiplicateurs de Lagrange" des conditions aux limites de Dirichlet [R3.03.01]. Le vecteur $\mathbf{B}^T \boldsymbol{\lambda}$ s'interprète comme l'opposé des réactions d'appui aux nœuds correspondants.

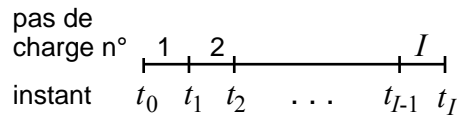
La formulation du problème quasi-statique consiste à exprimer l'équilibre de la structure (forces internes = forces externes) pour une suite d'instantes de calcul $\{t_i\}_{1 \leq i \leq I}$ qui paramètrent le chargement :

$$\begin{cases} \mathbf{R}(\mathbf{u}_i, t_i) + \mathbf{B}^T \boldsymbol{\lambda}_i = \mathbf{L}(t_i) \\ \mathbf{Bu}_i = \mathbf{u}^d(t_i) \end{cases} \quad \text{éq 1.2-3}$$

ce qui revient à annuler en $(\mathbf{u}_i, \boldsymbol{\lambda}_i, t_i)$ le vecteur $\mathbf{F}(\mathbf{u}, \boldsymbol{\lambda}, t)$ défini par :

$$\mathbf{F}(\mathbf{u}, \boldsymbol{\lambda}, t) = \begin{pmatrix} \mathbf{R}(\mathbf{u}, t) + \mathbf{B}^T \boldsymbol{\lambda} - \mathbf{L}(t) \\ \mathbf{Bu} - \mathbf{u}^d(t) \end{pmatrix}$$

L'état de la structure en t_0 est supposé connu. On effectue I incréments (ou pas) de charge définis comme suit :



Les inconnues sont calculées de façon incrémentale par l'algorithme de résolution global (même pour les comportements élastiques). À partir de $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$, solution satisfaisant l'équilibre en t_{i-1} , on détermine $\Delta \mathbf{u}_i$ et $\Delta \boldsymbol{\lambda}_i$, qui permettront d'obtenir la solution en t_i :

$$\begin{cases} \mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i \\ \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \Delta \boldsymbol{\lambda}_i \end{cases}$$

Les incréments $\Delta \mathbf{u}_i$ et $\Delta \boldsymbol{\lambda}_i$ sont d'abord estimés en linéarisant le problème par rapport au temps autour de $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1}, t_{i-1})$ (phase dite de prédiction ou d'Euler [bib 3]). Puis on utilise une méthode de NEWTON ou une de ses variantes pour résoudre l'équation [éq 1.2-3] de manière itérative (on calcule une suite $(\Delta \mathbf{u}_i^n, \Delta \boldsymbol{\lambda}_i^n)$).

Pour les relations de comportement incrémentales, on a besoin de connaître en t_{i-1} le champ de contraintes $\boldsymbol{\sigma}_{i-1}$ et le champ de variables internes $\boldsymbol{\alpha}_{i-1}$ (Cf. [R5.03.02] pour un exemple).

2 Méthode de résolution

2.1 Principe de la méthode de NEWTON

La méthode de NEWTON est une méthode classique de résolution des équations du type :

$$\mathbf{F}(\mathbf{x}) = 0$$

où \mathbf{F} est une fonction vectorielle (non linéaire) du vecteur \mathbf{x} .

Elle consiste à construire une suite de vecteurs $\{\mathbf{x}^n\}_n$ convergeant vers la solution \mathbf{x} . Pour trouver le nouvel itéré \mathbf{x}^{n+1} , on approche $\mathbf{F}(\mathbf{x}^{n+1})$ par un développement limité à l'ordre 1 autour de \mathbf{x}^n , et on exprime que $\mathbf{F}(\mathbf{x}^{n+1})$ doit être nul :

$$0 = \mathbf{F}(\mathbf{x}^{n+1}) \approx \mathbf{F}(\mathbf{x}^n) + \mathbf{F}'(\mathbf{x}^n)(\mathbf{x}^{n+1} - \mathbf{x}^n),$$

soit :

$$\mathbf{F}'(\mathbf{x}^n)(\mathbf{x}^{n+1} - \mathbf{x}^n) = -\mathbf{F}(\mathbf{x}^n) \quad \text{ou encore} \quad \mathbf{x}^{n+1} = \mathbf{x}^n - [\mathbf{F}'(\mathbf{x}^n)]^{-1} \mathbf{F}(\mathbf{x}^n)$$

Rappel :

$\mathbf{F}'(\mathbf{x})$ est l'application linéaire tangente associée à la fonction \mathbf{F} . La dérivée au point \mathbf{x} dans la direction \mathbf{h} est définie comme :

$$\mathbf{F}'(\mathbf{x}) \mathbf{h} = \lim_{\theta \rightarrow 0} \frac{\mathbf{F}(\mathbf{x} + \theta \mathbf{h}) - \mathbf{F}(\mathbf{x})}{\theta}.$$

La matrice de $\mathbf{F}'(\mathbf{x})$ dans les bases choisies pour les espaces vectoriels concernés s'appelle la matrice jacobienne de \mathbf{F} au point \mathbf{x} .

Lorsque \mathbf{F} est une fonction d'un espace vectoriel euclidien à valeurs réelles, $\mathbf{F}'(\mathbf{x})$ est une forme linéaire, et on peut montrer qu'il existe un vecteur (unique), noté $\nabla \mathbf{F}(\mathbf{x})$ et appelé le gradient de \mathbf{F} , tel que :

$$\mathbf{F}'(\mathbf{x}) \mathbf{h} = \mathbf{h}^T \nabla \mathbf{F}(\mathbf{x}) \quad (\text{produit scalaire de } \mathbf{h} \text{ et du gradient}).$$

Lorsque l'on est proche de la solution, la convergence de la méthode de NEWTON est quadratique (le nombre de 0 après la virgule dans l'erreur double à chaque itération : 0.19 - 0.036 - 0.0013 - 0.0000017 par exemple). Mais cette méthode (utilisant la vraie tangente) a plusieurs inconvénients :

- elle nécessite le calcul de la tangente à chaque itération, ce qui est coûteux (surtout dans le cas où on utilise un solveur direct),
- si l'incrément est grand, la tangente (dite cohérente ou consistante) peut conduire à des divergences de l'algorithme,
- elle peut ne pas être symétrique, ce qui oblige à utiliser des solveurs particuliers.

C'est pour cette raison que l'on peut utiliser d'autres matrices à la place de la matrice tangente : la matrice élastique, une matrice tangente obtenue antérieurement, la matrice tangente symétrisée, ... [§2.2.1].

2.2 Adaptation de la méthode de NEWTON au problème posé

2.2.1 Résolution sans conditions aux limites dualisées

Si l'on oublie dans un premier temps les conditions aux limites de Dirichlet, on doit résoudre un système de la forme :

$$\mathbf{R}(\mathbf{u}_i, t_i) = \mathbf{L}_i^{méca},$$

où $\mathbf{L}_i^{méca}$ désigne le chargement mécanique à l'instant t_i .

En utilisant les notations du [§ 2.1], cela revient à annuler la fonction vectorielle \mathbf{F} définie par :

$$\mathbf{F}(\mathbf{u}_i, t_i) = \mathbf{R}(\mathbf{u}_i, t_i) - \mathbf{L}_i^{méca}$$

Les forces nodales \mathbf{R} peuvent symboliquement être notées $\mathbf{Q}^T \boldsymbol{\sigma}$, où \mathbf{Q}^T est la matrice associée à l'opérateur divergence (expression du travail du champ de déformations virtuelles), avec $\boldsymbol{\sigma}(\mathbf{u}_i, T_i, t_i, \mathbf{H}_{i-1})$. On a :

$$(\mathbf{Q}^T \boldsymbol{\sigma})_k = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{w}_k) d\Omega$$

$$(\mathbf{L}_i)_k = \int_{\Omega} \mathbf{f}_i \cdot \mathbf{w}_k d\Omega + \int_{\Gamma} \mathbf{t}_i \cdot \mathbf{w}_k d\Gamma$$

où

- \mathbf{w}_k désigne la fonction de base associée au $k^{\text{ième}}$ degré de liberté de la structure,
- \mathbf{f}_i désigne les forces volumiques s'appliquant à l'instant t_i sur Ω ,
- \mathbf{t}_i désigne les forces surfaciques s'appliquant à l'instant t_i sur la frontière Γ de Ω .

[La matrice \mathbf{Q} dépend des déplacements \mathbf{u} en grands déplacements [R5.03.20].

L'application de la méthode de NEWTON conduit à résoudre une suite de problèmes linéaire du type (n est le numéro de l'itération de NEWTON, i celui de la variable d'instant) :

$$\mathbf{K}_i^n \delta \mathbf{u}_i^{n+1} = \mathbf{L}_i^{méca} - \mathbf{R}_i^n,$$

où l'on note $\delta \mathbf{u}_i^{n+1} = \mathbf{u}_i^{n+1} - \mathbf{u}_i^n$ l'incrément de déplacement entre deux itérations de NEWTON successives. La matrice \mathbf{K}_i^n est la matrice de rigidité tangente en \mathbf{u}_i^n et le vecteur \mathbf{R}_i^n représente les forces internes à la $n^{\text{ième}}$ itération de NEWTON du $i^{\text{ème}}$ pas de charge. La quantité $\mathbf{L}_i^{méca} - \mathbf{R}_i^n$ représente les forces non équilibrées, que l'on peut aussi appeler le "résidu".

La matrice \mathbf{K}_i^n est la matrice de l'application linéaire tangente de la fonction \mathbf{F} , elle vaut donc :

$$\mathbf{K}_i^n = \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i^n, t_i)} = \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i^n, t_i)} - \frac{\partial \mathbf{L}_i^{méca}}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i^n, t_i)}.$$

En l'absence de forces suiveuses [§2.2.3], le second terme est nul. La matrice \mathbf{K}_i^n est donc la dérivée au point \mathbf{u}_i^n des forces nodales (ou internes) par rapport aux déplacements :

$$\mathbf{K}_i^n = \left. \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right|_{(\mathbf{u}_i^n, t_i)}$$

Une petite erreur dans l'évaluation du résidu peut avoir des conséquences graves, car c'est son calcul exact qui garantit, si l'on converge, que ce sera vers la solution cherchée. Par contre, il n'est pas toujours nécessaire d'utiliser la vraie matrice tangente, dont le calcul et la factorisation sont coûteux. Par exemple, une variante de la méthode utilise la matrice élastique \mathbf{K}_0 .

La méthode utilisant la vraie matrice tangente \mathbf{K}_i^n (dite aussi matrice cohérente ou consistante) s'appelle la méthode de NEWTON ; les méthodes utilisant d'autres matrices (comme par exemple la matrice élastique \mathbf{K}_0) sont appelées méthodes de NEWTON modifiées. Le choix entre une matrice tangente (la dernière obtenue ou une matrice précédente) est effectué dans le *Code_Aster* par l'intermédiaire du mot-clé **MATRICE** : 'TANGENTE' ou 'ELASTIQUE' du mot-clé facteur **NEWTON**. De plus, il est possible d'utiliser une matrice de décharge, c'est-à-dire d'une matrice à variables internes constantes (l'évolution des non linéarités n'est donc pas prise en compte dans cette matrice), en dessous d'un certain pas de temps, pour certaines lois de comportement. On se reportera à la documentation [U4.51.03] pour l'utilisation de cette fonctionnalité.

La méthode de NEWTON à matrice tangente consistante peut s'illustrer simplement à l'aide du schéma de la figure [Figure 2.2.1-a].

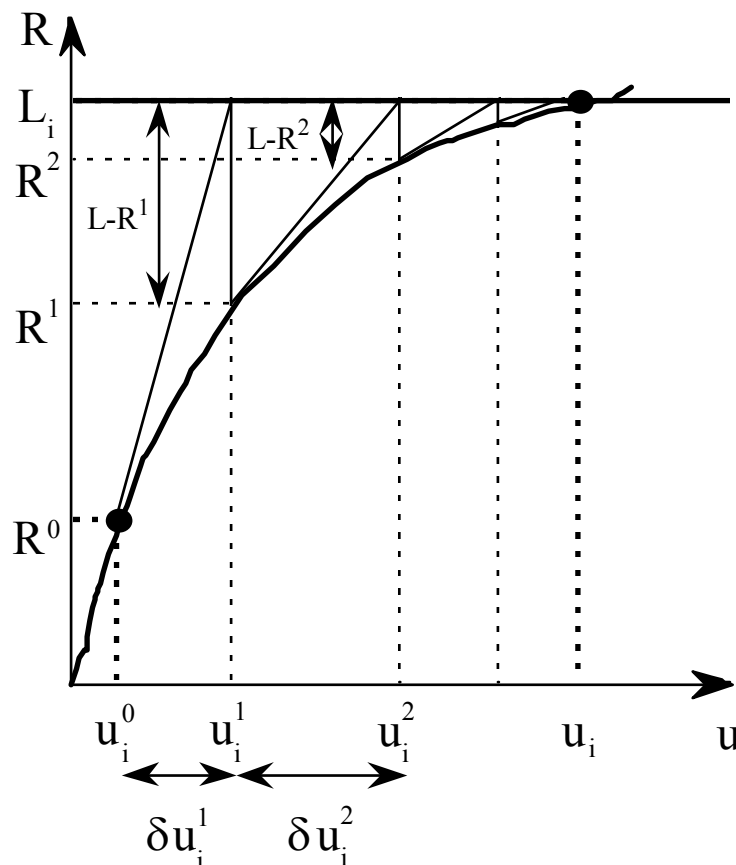


Figure 2.2.1-a

2.2.2 Résolution avec des conditions aux limites dualisées

Lorsque l'on prend en compte les conditions de déplacements imposés, le système à résoudre s'écrit :

$$\begin{cases} \mathbf{R}(\mathbf{u}_i, t_i) + \mathbf{B}^T \boldsymbol{\lambda}_i = \mathbf{L}_i^{méca} \\ \mathbf{B} \mathbf{u}_i = \mathbf{u}_i^d \end{cases}$$

On utilisera le symbole Δ pour noter les incréments depuis l'équilibre précédent (en t_{i-1}) des déplacements, des paramètres de Lagrange, du chargement et des déplacements imposés :

$$\begin{cases} \Delta \mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_{i-1} \\ \Delta \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_i - \boldsymbol{\lambda}_{i-1} \\ \Delta \mathbf{L}_i^{méca} = \mathbf{L}_i^{méca} - \mathbf{L}_{i-1}^{méca} \\ \Delta \mathbf{u}_i^d = \mathbf{u}_i^d - \mathbf{u}_{i-1}^d \end{cases}$$

Connaissant $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$, le couple $(\Delta \mathbf{u}_i, \Delta \boldsymbol{\lambda}_i)$ sera déterminé par la résolution du système :

$$\begin{cases} \mathbf{R}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_i, t_i) + \mathbf{B}^T (\boldsymbol{\lambda}_{i-1} + \Delta \boldsymbol{\lambda}_i) = \mathbf{L}_{i-1}^{méca} + \Delta \mathbf{L}_i^{méca} \\ \mathbf{B} (\mathbf{u}_{i-1} + \Delta \mathbf{u}_i) = \mathbf{u}_{i-1}^d + \Delta \mathbf{u}_i^d \end{cases} \quad \text{éq 2.2.2-1}$$

On va utiliser une méthode de NEWTON pour résoudre ce système. Cependant, l'expérience montre que la convergence de la méthode de NEWTON est fortement dépendante d'un choix judicieux de l'estimation initiale : "plus l'estimation initiale est proche de la solution, plus l'algorithme converge vite". Pour amorcer le processus itératif de la méthode, il est donc utile de déterminer un "bon" incrément initial $(\Delta \mathbf{u}_i^0, \Delta \boldsymbol{\lambda}_i^0)$. Pour cela, on linéarise par rapport au temps le problème continu : c'est ce qu'on appelle la phase de prédiction (ou d'initialisation). On enchaîne avec la boucle des itérations de NEWTON qui permet, à la convergence, d'obtenir les valeurs de $(\Delta \mathbf{u}_i, \Delta \boldsymbol{\lambda}_i)$, et donc celles de $(\mathbf{u}_i, \boldsymbol{\lambda}_i)$.

2.2.2.1 Phase de prédiction

On linéarise le système [éq 2.2.2-1] par rapport au temps autour de $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$; en tenant compte de l'équilibre obtenu à l'instant t_{i-1} , on obtient les équations permettant de calculer des valeurs prédictives $(\Delta \mathbf{u}_i^0, \Delta \boldsymbol{\lambda}_i^0)$:

$$\begin{cases} \left. \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right|_{\mathbf{u}_{i-1}} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_i^0 = \Delta \mathbf{L}_i^{méca} - \left. \frac{\partial \mathbf{R}}{\partial t} \right|_{t_{i-1}} \Delta t_i \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \Delta \mathbf{u}_i^d \end{cases}$$

$\left. \frac{\partial \mathbf{R}}{\partial t} \right|_{t_{i-1}}$ désigne la différentielle totale de $\mathbf{R} = \mathbf{Q}(\mathbf{u}) : \boldsymbol{\sigma}(t, T(t))$ par rapport à t . Cette notation

particulière a pour but d'attirer l'attention sur le fait que $\frac{\delta \boldsymbol{\sigma}}{\delta t} = \frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{\partial \boldsymbol{\sigma}}{\partial T} \frac{\partial T}{\partial t}$ comme on le voit dans la suite.

Donc, en utilisant [éq 1.2-1] et la dépendance de σ par rapport au temps et par rapport à la température :

$$\left\{ \begin{array}{l} \left(\mathbf{Q}^T : \frac{\partial \sigma}{\partial \mathbf{u}} \bigg|_{\mathbf{u}_{i-1}} + \frac{\partial \mathbf{Q}^T}{\partial \mathbf{u}} \bigg|_{\mathbf{u}_{i-1}} : \sigma \right) \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \lambda_i^0 = \Delta \mathbf{L}_i^{méca} - \mathbf{Q}^T : \frac{\partial \sigma}{\partial T} \bigg|_{t_{i-1}} \Delta T_i - \mathbf{Q}^T : \frac{\partial \sigma}{\partial t} \bigg|_{t_{i-1}} \Delta t_i \\ \mathbf{B} \Delta \mathbf{u}_i = \Delta \mathbf{u}_i^d \end{array} \right.$$

Si les contraintes dépendent d'autres variables, comme par exemple des proportions z des différentes phases en métallurgie i.e. $\sigma(t, T(t), z(t))$, on voit apparaître au second membre

le terme correspondant : $\mathbf{Q}^T : \frac{\partial \sigma}{\partial z} \bigg|_{t_{i-1}} \Delta z_i$. On suppose par la suite que les effets

métallurgiques [R4.04.02] sont intégrés au second membre représentatif des effets thermiques. De plus, on ne tient pas compte actuellement pour la phase de prédiction de la dépendance explicite des contraintes par rapport au temps. Enfin, la dépendance de la matrice \mathbf{Q}^T par rapport aux déplacements est négligée dans l'hypothèse des petits

déplacements : le terme $\frac{\partial \mathbf{Q}^T}{\partial \mathbf{u}} \bigg|_{\mathbf{u}_{i-1}}$, dit terme de rigidité géométrique, disparaît donc de

l'équation précédente.

Avec des notations évidentes, le système d'équations obtenu s'écrit également :

$$\left\{ \begin{array}{l} \mathbf{K}_{i-1} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \lambda_i^0 = \Delta \mathbf{L}_i^{méca} + \Delta \mathbf{L}_i^{ther} \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \Delta \mathbf{u}_i^d \end{array} \right. \quad \text{éq 2.2.2.1-1}$$

Notons que l'on peut remplacer formellement la matrice \mathbf{K}_{i-1} , dérivée de \mathbf{R} par rapport à \mathbf{u} en \mathbf{u}_{i-1} par la matrice élastique \mathbf{K}_0 .

Pour les développeurs, précisons que le calcul de la matrice tangente lors de la phase de prédiction est effectué via l'option de calcul RIGI_MECA_TANG [§ 2.2.2.3].

L'incrément effectif de chargement $\Delta \mathbf{L}_i$ apparaissant au second membre comprend deux termes :

- l'incrément de chargement mécanique $\Delta \mathbf{L}_i^{méca}$, composé des charges mortes (indépendantes de la géométrie, comme la pesanteur) et des charges suiveuses (dépendant de la géométrie, comme la pression suiveuse [R3.03.04]). En réalité, il existe des cas (le premier incrément de charge, par exemple) où $\mathbf{L}_{i-1}^{méca}$ est inconnu : de ce fait $\Delta \mathbf{L}_i^{méca}$ est plutôt calculé par la relation $\Delta \mathbf{L}_i^{méca} = \mathbf{L}_i^{méca} - \mathbf{Q}^T \sigma_{i-1} - \mathbf{B}^T \lambda_{i-1}$, ce qui revient exactement au même, dans le cas courant, en tenant compte de l'équilibre à l'incrément (i-1). On remarquera alors que l'expression fait intervenir les multiplicateurs de Lagrange à l'incrément (i-1), qui sont parfois inconnus (au premier incrément de charge, par exemple). Mais linéarité des conditions aux limites en déplacements imposés, qui se traduit par le fait que la matrice \mathbf{B} est constante, permet dans ce cas un abus de langage : on pose $\lambda_i = 0$ et $\Delta \lambda_i^0 = \lambda_{i-1}^{vrai} + \Delta \lambda_i^{0 \text{ vrai}}$ (l'exposant *vrai* correspond aux variables apparaissant effectivement dans le système d'équations ci-dessus) et on résout :

$$\left\{ \begin{array}{l} \mathbf{K}_{i-1} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \lambda_i^0 = \mathbf{L}_i^{méca} + \Delta \mathbf{L}_i^{ther} - \mathbf{Q}^T \sigma_{i-1} \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \mathbf{u}_i^d - \mathbf{B} \mathbf{u}_{i-1} \end{array} \right.,$$

- l'incrément de chargement dit "thermique" (et métallurgique [R4.04.02]) $\Delta \mathbf{L}_i^{ther}$, issu de la dérivation des forces nodales par rapport à la température et qui est une estimation de l'effet d'une variation de température. En effet, si l'on note K le module de compression hydrostatique et α le coefficient de dilatation thermique, l'incrément de chargement "thermique" s'écrit (puisque $\mathbf{R} = \mathbf{Q}^T \boldsymbol{\sigma}$ et $\boldsymbol{\sigma}^{ther} = -3K\alpha \Delta T_i \mathbf{I}_d$ où \mathbf{I}_d est la matrice identité) :

$$\Delta \mathbf{L}_i^{ther} = -\mathbf{Q}^T : \frac{\partial \boldsymbol{\sigma}}{\partial T} \bigg|_{t_{i-1}} \Delta T_i = \mathbf{Q}^T : 3K\alpha \Delta T_i \mathbf{I}_d,$$

Dans le cas élastique, ce sont les forces nodales associées à une dilatation thermique (ce n'est pas à proprement parler un chargement, cela s'assimile plutôt à l'effet d'une déformation initiale). Cette estimation est utilisée dans la phase de prédiction et dans le critère d'arrêt. Si les dilatations thermiques font sortir la structure du domaine élastique (plasticité par exemple), cette estimation sera corrigée lors des itérations de NEWTON.

Remarque :

Un cas particulier concerne l'utilisation d'une excitation de type `TYPE_CHARGE: 'DIDI'` signifiant Dirichlet différentiel, c'est-à-dire par rapport à l'état initial. Cela consiste, pour les conditions aux limites de type blocages, à imposer, non pas $\mathbf{B} \mathbf{u} = \mathbf{u}^d$, mais $\mathbf{B}(\mathbf{u} - \mathbf{u}_0) = \mathbf{u}^d$. Dans ce cas, le système à résoudre dans la phase de prédiction pour le nouvel incrément de charge est :

$$\begin{cases} \mathbf{K}_{i-1} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_i^0 = \mathbf{L}_i^{méca} + \Delta \mathbf{L}_i^{ther} - \mathbf{Q}^T \boldsymbol{\sigma}_{i-1} \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \mathbf{u}_i^d \end{cases}$$

2.2.2.2 Boucle sur les itérations de NEWTON

On doit trouver les valeurs $(\Delta \mathbf{u}_i, \Delta \boldsymbol{\lambda}_i)$ des incréments de déplacements et paramètres de Lagrange depuis les valeurs $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$ obtenues à l'équilibre précédent (instant t_{i-1}). On prend comme valeurs initiales $(\Delta \mathbf{u}_i^0, \Delta \boldsymbol{\lambda}_i^0)$ obtenues à l'issue de la phase de prédiction, avant de commencer les itérations de la méthode de NEWTON.

À chaque itération, on doit résoudre un système permettant de déterminer $(\delta \mathbf{u}_i^{n+1}, \delta \boldsymbol{\lambda}_i^{n+1})$, incréments des déplacements et des paramètres de Lagrange depuis le résultat $(\mathbf{u}_i^n, \boldsymbol{\lambda}_i^n)$ de l'itération précédente :

$$\begin{cases} \mathbf{K}_i^n \delta \mathbf{u}_i^{n+1} + \mathbf{B}^T \delta \boldsymbol{\lambda}_i^{n+1} = \mathbf{L}_i^{méca} - \mathbf{R}(\mathbf{u}_i^n) - \mathbf{B}^T \boldsymbol{\lambda}_i^n \\ \mathbf{B} \delta \mathbf{u}_i^{n+1} = 0 \end{cases} \quad \text{éq 2.2.2.2-1}$$

avec $\mathbf{R}(\mathbf{u}_i^n) = \mathbf{Q}^T \boldsymbol{\sigma}_i^n$, les contraintes $\boldsymbol{\sigma}_i^n$ étant calculées à partir des déplacements \mathbf{u}_i^n par l'intermédiaire de la relation de comportement du matériau [§1.1]. En fait, dans le cas des comportements incrémentaux, $\boldsymbol{\sigma}_i^n$ est calculé à partir de $(\boldsymbol{\sigma}_{i-1}, \alpha_{i-1})$ et de l'incrément de déformation $\boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^{n+1})$ induit par l'incrément de déplacement $\Delta \mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \delta \mathbf{u}_i^{n+1} - \mathbf{u}_{i-1}$.

Matrice "tangente"

Comme dans la phase de prédiction, on n'est pas obligé d'utiliser la vraie matrice tangente \mathbf{K}_i^n . En particulier, l'opérateur `STAT_NON_LINE` autorise l'utilisation de la matrice élastique \mathbf{K}_0 , ou la réactualisation de la matrice tangente tous les i_0 pas de temps (mot-clé `REAC_INCR`) ou toutes les n_0 itérations de NEWTON (mot-clé `REAC_ITER`). Ainsi, la matrice \mathbf{K}_i^n peut-être remplacée par une matrice \mathbf{K}_j , avec $j \leq i$, ou une matrice \mathbf{K}_i^m , avec $m \leq n$.

Attention : une matrice trop "raide" ne pose pas de problèmes de stabilité mais peut produire une convergence très lente ; une matrice trop "souple" peut conduire à divergence, il est conseillé dans ce cas de faire de la recherche linéaire [§ 2.3].

Il est difficile de donner une règle permettant de savoir quand on doit réactualiser la matrice tangente : cela dépend fortement du degré de non linéarité du problème et de la taille des incréments de charge.

En décharge, il est recommandé soit d'utiliser la matrice élastique pour la phase de prédiction et de résolution, soit d'utiliser la matrice élastique pour la phase de prédiction puis la matrice tangente pour la résolution.

Les figures ci-après illustrent les diverses possibilités de réactualisation de la matrice tangente : matrice élastique \mathbf{K}_0 utilisée à chaque itération [Figure 2.2.2.2-a], matrice tangente réactualisée à chaque itération et à chaque pas de temps [Figure 2.2.2.2-b], matrice tangente réactualisée tous les i_0 pas de temps ($i_0=1$ ici) [Figure 2.2.2.2-c], et matrice tangente réactualisée toutes les n_0 itérations de NEWTON ($n_0=2$ ici) [Figure 2.2.2.2-d].

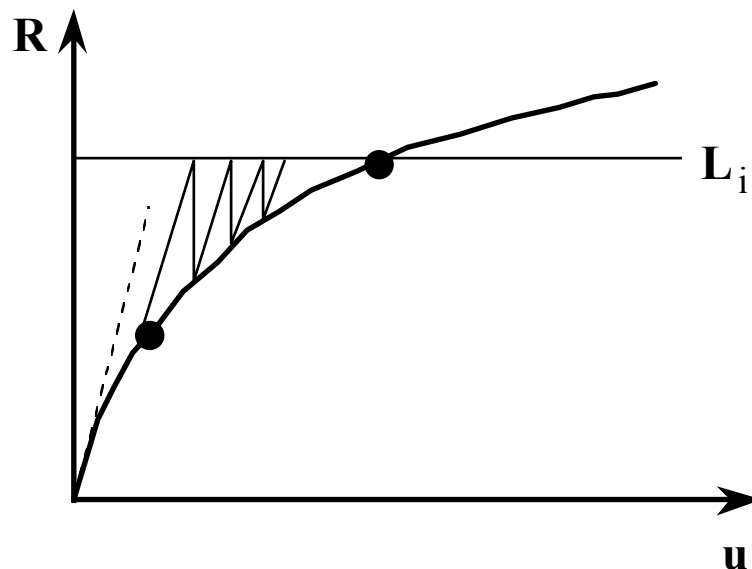


Figure 2.2.2.2-a : utilisation de la matrice élastique

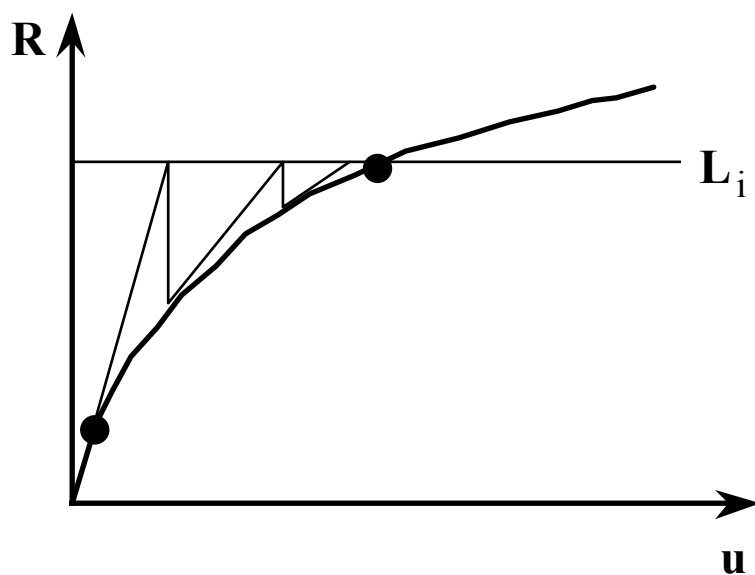


Figure 2.2.2.2-b : utilisation de la vraie matrice tangente réévaluée à chaque itération

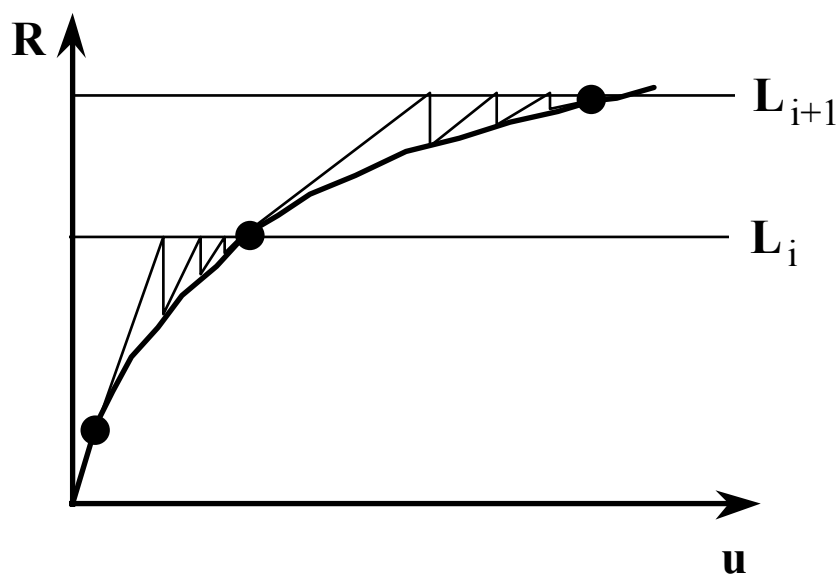


Figure 2.2.2.2-c : utilisation de la matrice tangente réévaluée à chaque pas de temps

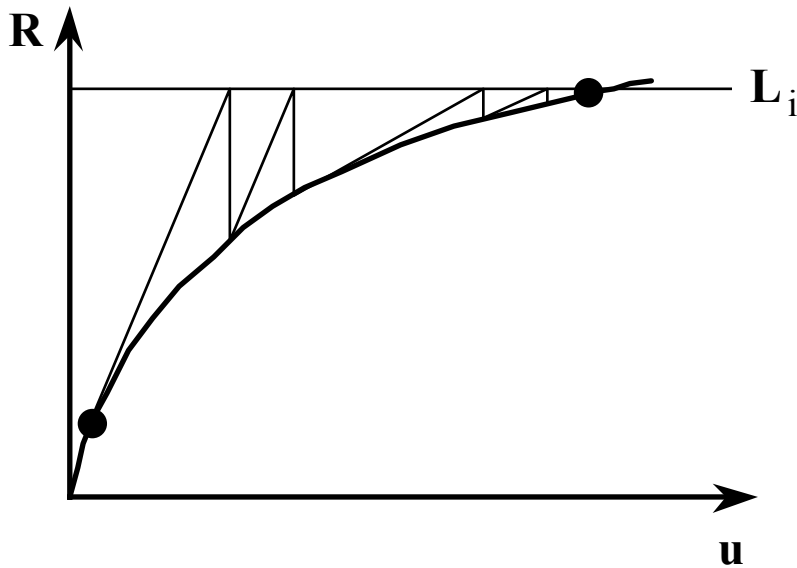


Figure 2.2.2-d : utilisation de la matrice tangente réévaluée toutes les 2 itérations de NEWTON

La méthode de NEWTON modifiée (utilisant une autre matrice que la matrice tangente consistante) converge moins vite que la méthode de NEWTON classique, mais est moins coûteuse. Elle est d'autant plus économique que le nombre de degrés de liberté du système est élevé. C'est pourquoi on peut conseiller, lorsque le système est de grande taille, de garder la même matrice tangente pendant quelques itérations. Enfin, n'oublions pas de signaler que dans certains cas, c'est le calcul avec la matrice élastique qui est le plus rapide en termes de temps de calcul, même si le nombre d'itérations effectuées est beaucoup plus important (ce sont des itérations bon marché puisque la matrice n'est calculée et factorisée qu'une seule fois) ; de plus, c'est la matrice élastique qu'il est recommandé d'utiliser lors des décharges.

Comme le montre l'équation [éq 2.2.2.2-1], il faut effectuer à chaque itération de NEWTON le calcul éventuel de la nouvelle matrice tangente \mathbf{K}_i^n et des "forces nodales" $\mathbf{R}_i^n + \mathbf{B}^T \boldsymbol{\lambda}_i^n$: pour les développeurs, précisons que ces opérations sont effectuées par l'option de calcul FULL_MECA (RAPH_MECA si la matrice tangente n'est pas recalculée).

La matrice tangente obtenue par l'option RIGI_MECA_TANG et la matrice tangente obtenue par l'option FULL_MECA sont en général calculées différemment [§ 2.2.2.3].

Actualisation des inconnues

L'actualisation des déplacements et de leurs incréments se fait comme suit, celle des paramètres de Lagrange est effectuée de manière identique :

$$\begin{cases} \mathbf{u}_i^{n+1} &= \mathbf{u}_i^n + \delta \mathbf{u}_i^{n+1} \\ \Delta \mathbf{u}_i^{n+1} &= \Delta \mathbf{u}_i^n + \delta \mathbf{u}_i^{n+1} \end{cases}$$

Critères de convergence

Il existe trois critères de convergence.

Le critère `RESI_GLOB_MAXI` consiste à vérifier que la norme infinie du résidu $\left\| \mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \boldsymbol{\lambda}_i^n - \mathbf{L}_i^{méca} \right\|_\infty$, est inférieure à la valeur spécifiée par l'utilisateur. Il n'est pas conseillé d'utiliser ce critère seul, car on ne peut pas facilement avoir une idée des ordres de grandeurs absolus admissibles.

Le critère de convergence choisi par défaut revient à vérifier que le résidu est suffisamment petit, comme précédemment, et ceci relativement à une quantité représentative du chargement (c'est le critère `RESI_GLOB_RELA`) :

$$\frac{\left\| \mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \boldsymbol{\lambda}_i^n - \mathbf{L}_i^{méca} \right\|_\infty}{\left\| \mathbf{L}_i^{méca} + \mathbf{L}_i^{ther} - \mathbf{B}^T \boldsymbol{\lambda}_i^n \right\|_\infty} \leq \eta, \quad \text{éq 2.2.2.2-2}$$

η étant la précision relative souhaitée donnée par l'utilisateur (ou la valeur par défaut de 10^{-6}) et $\left\| \right\|_\infty$ la norme du maximum.

On peut remarquer que, dans le cas d'utilisation de `RESI_GLOB_RELA`, le critère peut devenir singulier si le chargement extérieur $\mathbf{L}_i^{méca} + \mathbf{L}_i^{ther} - \mathbf{B}^T \boldsymbol{\lambda}_i^n$ devient nul. Ceci peut arriver en cas de décharge totale de la structure. Si un tel cas de figure se présente (i.e. chargement 10^{-6} fois plus petit que le plus petit chargement observé jusqu'au présent incrément), le code utilise alors le critère `RESI_GLOB_MAXI` avec comme valeur celle observée à la convergence de l'incrément précédent. Lorsque le chargement redevient non nul, on revient au critère initial.

Le troisième critère est le critère `RESI_REFE_RELA` : l'idée de ce critère est de construire une force nodale de référence, qui servira à estimer terme à terme, la nullité (approchée) du résidu :

$$\forall j \in \{ddl_s\} \quad \left| \left(\mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \boldsymbol{\lambda}_i^n - \mathbf{L}_i^{méca} \right)_j \right| \leq \varepsilon F_j^{ref}$$

Plus précisément, la force nodale de référence est construite à partir de la donnée d'une amplitude de contrainte de référence (en mécanique ; dans le cas de la THM, il est nécessaire de donner une référence à chaque phénomène physique entrant dans la contrainte généralisée) σ^{ref} . A partir de cette amplitude de contrainte de référence, on définit le tenseur σ_j^{test} : il est nul pour toutes ces composantes, sauf la j-ième qui vaut σ^{ref} . On définit alors, pour chaque nœud de chaque élément la force nodale suivante (le but étant de donner une idée de l'importance d'une composante en un point de Gauss de la contrainte sur la force nodale) :

$$\tilde{R}_i^e = \frac{1}{N} \sum_{\alpha=1}^N \sum_{j=1}^M \left| B_{i,j}^\alpha \sigma_j^{test} \right| \omega^\alpha$$

avec N le nombre de points de Gauss de l'élément, M le nombre de composantes du tenseur de contrainte ; l'exposant α servant à noter l'évaluation de quantité au point de Gauss α , ω sont les poids des points de Gauss.

La force nodale de référence est alors définie par :

$$F_i^{ref} = \min_{e \in \Gamma_i} \tilde{R}_i^e$$

où Γ_i est l'ensemble des éléments connectés au nœud i .

La convergence est décrétée réalisée lorsque tous les critères spécifiés par l'utilisateur sont vérifiés simultanément. Par défaut, on fait un test sur le résidu global relatif (`RESI_GLOB_RELA`) et le nombre maximum d'itérations de NEWTON (`ITER_GLOB_MAXI`).

2.2.2.3 RIGI_MECA_TANG et FULL_MECA

Il est important de souligner que la matrice tangente issue de l'option `RIGI_MECA_TANG` et la matrice tangente issue de l'option `FULL_MECA` ne sont en général pas identiques.

Supposons que l'on a atteint la convergence pour l'instant t_{i-1} et que l'on cherche maintenant à obtenir l'équilibre pour l'instant suivant t_i . La matrice issue de `RIGI_MECA_TANG` provient d'une linéarisation des équations d'équilibre par rapport au **temps** autour de $(\mathbf{u}_{i-1}, \lambda_{i-1})$ i.e. autour de l'équilibre à l'instant t_{i-1} . C'est donc la matrice tangente du système à convergence à l'instant t_{i-1} . Par contre, la matrice issue de `FULL_MECA` provient d'une linéarisation des équations d'équilibre par rapport au **déplacement** autour de $(\mathbf{u}_i^n, \lambda_i^n)$ i.e. autour de l'équilibre à l'instant t_i .

En outre, on peut interpréter les différences entre `RIGI_MECA_TANG` et `FULL_MECA` en d'autres termes. On peut ainsi montrer que la matrice issue de `RIGI_MECA_TANG` correspond à l'opérateur tangent du problème **continu** en temps, dit aussi problème en vitesse (et relie la vitesse de contrainte à la vitesse de déformation), alors que la matrice issue de `FULL_MECA` correspond à l'opérateur tangent du problème **discrétisé** en temps [bib1]. Le document [R5.03.02] donne l'expression dans chacun des deux cas pour la relation d'élasto-plasticité de VON MISES à écrouissage isotrope.

On rappelle que le traitement d'une relation de comportement [R5.03.02 § 5] consiste à :

- calculer les contraintes σ_i^n et les variables internes α_i^n à partir de l'état initial $(\sigma_{i-1}, \alpha_{i-1})$ et de l'incrément de déplacement $\Delta \mathbf{u}_i^n$,
- calculer les forces nodales $\mathbf{R}_i^n = \mathbf{Q}^T \sigma_i^n$ et les réactions d'appui $\mathbf{B}^T \lambda_i^n$,
- calculer (éventuellement) la matrice tangente (option `RIGI_MECA_TANG` pour la phase de prédiction, option `FULL_MECA` pour les itérations de NEWTON).

2.2.3 Cas des chargements suiveurs

Un chargement suiveur (en mécanique) est un chargement qui dépend de la géométrie de la structure, comme par exemple la pression qui s'exerce dans la direction opposée à la normale (ou les forces d'inertie dans un repère non galiléen). Ainsi, lorsque la structure se déforme avec l'évolution de la charge, le chargement, exprimé dans un repère absolu, est transformé. Les charges qui ne dépendent pas de la géométrie de la structure sont appelées des charges mortes ou fixes (par exemple, la pesanteur). Pour indiquer qu'une charge doit être traitée comme une charge suiveuse dans `STAT_NON_LINE`, on indique `TYPE_CHARGE : 'SUIV'` sous le mot-clé `EXCIT`.

Un chargement mécanique $\mathbf{L}^{méca}(t)$ comportant des charges suiveuses s'écrit $\mathbf{L}^{fixe}(t) + \mathbf{L}^{suiv}(\mathbf{u}, t)$ (l'exposant *fixe* désigne ici les charges mortes, et *suiv* les charges suiveuses). Le système d'équations à résoudre devient alors :

$$\begin{cases} \mathbf{R}(\mathbf{u}_i, t_i) + \mathbf{B}^T \lambda_i &= \mathbf{L}^{fixe}(t_i) + \mathbf{L}^{suiv}(\mathbf{u}_i, t_i) \\ \mathbf{B} \mathbf{u}_i &= \mathbf{u}_i^d \end{cases}$$

Les opérations de dérivation permettant d'écrire la phase de prédiction et les itérations de la méthode de NEWTON font donc intervenir les dérivées de \mathbf{L}^{suiv} par rapport aux déplacements \mathbf{u} .

La phase de prédiction devient :

$$\begin{cases} \left(\mathbf{K}_{i-1} - \frac{\partial \mathbf{L}^{suiv}}{\partial \mathbf{u}} \right)_{\mathbf{u}_{i-1}} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_i^0 = \Delta \mathbf{L}_i^{fixe} + \Delta \mathbf{L}_i^{suiv} + \Delta \mathbf{L}_i^{ther} \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \Delta \mathbf{u}_i^d \end{cases}$$

et les itérations de NEWTON consistent à résoudre le système :

$$\begin{cases} \left(\mathbf{K}_i^n - \frac{\partial \mathbf{L}^{suiv}}{\partial \mathbf{u}} \right)_{\mathbf{u}_i^n} \delta \mathbf{u}_i^{n+1} + \mathbf{B}^T \delta \boldsymbol{\lambda}_i^{n+1} = \mathbf{L}_i^{fixe} + \mathbf{L}^{suiv}(\mathbf{u}_i^n) - \mathbf{R}(\mathbf{u}_i^n) - \mathbf{B}^T \boldsymbol{\lambda}_i^n \\ \mathbf{B} \delta \mathbf{u}_i^{n+1} = 0 \end{cases}$$

Ainsi, au début de chaque pas de charge (prédiction) et à chaque itération de NEWTON, on doit calculer une matrice de "rigidité" $\left(-\partial \mathbf{L}^{suiv} / \partial \mathbf{u} \right)_{\mathbf{u}_i^n}$ et un vecteur $\mathbf{L}^{suiv}(\mathbf{u}_i^n)$ liés aux chargements suiveurs.

Les seules charges qui peuvent être traitées comme des charges suiveuses dans l'état actuel de l'opérateur STAT_NON_LINE sont :

- la pression pour les modélisations 3D, 3D_SI, D_PLAN, D_PLAN_SI, AXIS, AXIS_SI, C_PLAN, C_PLAN_SI [R3.03.04],
- le chargement de pesanteur pour les éléments CABLE_POULIE [R3.08.05], éléments à trois nœuds comportant une poulie et deux brins de câbles : la force de pesanteur s'exerçant sur l'élément dépend des longueurs respectives des deux brins,
- la force centrifuge en grands déplacements, qui pour une vitesse de rotation ω est donné par :

$$\int_{\Omega} \rho \omega \wedge [\omega \wedge \mathbf{OM}] d\Omega = \int_{\Omega} \rho \omega \wedge [\omega \wedge (\mathbf{OM}_0 + \mathbf{u})] d\Omega ;$$

- le chargement de pesanteur pour toutes les modélisations THM des milieux poreux non saturés [R7.01.10] : en effet, la masse volumique dépend des variables nodales \mathbf{u} , p et T pour tenir compte des relations de comportement des géomatériaux.

2.3 Recherche linéaire

La recherche linéaire exposé ici concerne la recherche linéaire en l'absence de pilotage. Pour la description de la recherche linéaire en présence de pilotage, on se reportera à la documentation [R5.03.80].

2.3.1 Principe

L'introduction de la recherche linéaire dans l'opérateur STAT_NON_LINE résulte d'un constat : la méthode de NEWTON avec matrice consistante ne converge pas dans tous les cas de figure, notamment lorsque l'on part trop loin de la solution. D'autre part, l'utilisation de matrices autres que la matrice tangente consistante peut, lorsqu'elles sont trop "souples", conduire à divergence. La recherche linéaire permet de se prémunir contre de telles divergences.

Elle consiste à considérer $(\delta \mathbf{u}_i^{n+1}, \delta \lambda_i^{n+1})$, non plus comme l'incrément des déplacements et des paramètres de Lagrange, mais comme une direction de recherche dans laquelle on va chercher à minimiser une fonctionnelle (l'énergie de la structure). On trouvera un pas d'avancement ρ dans cette direction, et l'actualisation des inconnues consistera à faire :

$$\begin{cases} \Delta \mathbf{u}_i^{n+1} = \Delta \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \Delta \lambda_i^{n+1} = \Delta \lambda_i^n + \rho \delta \lambda_i^{n+1} \end{cases}$$

En l'absence de recherche linéaire (par défaut) le scalaire ρ est bien sûr égal à 1.

2.3.2 Minimisation d'une fonctionnelle

Afin de mieux se convaincre du bien fondé de la recherche linéaire, on peut interpréter la méthode de NEWTON comme une méthode de minimisation d'une fonctionnelle (dans le cas où les matrices tangentes sont symétriques). Nous insistons sur le fait que les équations obtenues sont rigoureusement celles de la méthode de NEWTON exposée dans le [§2.2] et que seule la façon d'y parvenir est différente.

"Oublions" pour simplifier l'exposé la dualisation des conditions aux limites de Dirichlet et plaçons-nous dans l'hypothèse des petites déformations. On considère la fonctionnelle :

$$J : V \rightarrow \mathbb{R}$$

$$\mathbf{u} \rightarrow J(\mathbf{u}) = \int_{\Omega} w(\boldsymbol{\varepsilon}(\mathbf{u})) d\Omega - \int_{\Omega} \mathbf{f} \cdot \mathbf{u} d\Omega - \int_{\Gamma} \mathbf{t} \cdot \mathbf{u} d\Gamma$$

où la densité d'énergie libre w permet de relier le tenseur des contraintes $\boldsymbol{\sigma}$ au tenseur des déformations linéarisées $\boldsymbol{\varepsilon}$ par la relation $\boldsymbol{\sigma} = \frac{\partial w}{\partial \boldsymbol{\varepsilon}}$.

La fonctionnelle J étant convexe, trouver le minimum de J est équivalent à annuler son gradient, soit :

$$\nabla J(\mathbf{u}) \cdot \mathbf{v} = 0 \quad \forall \mathbf{v} \in V,$$

ce qui est exactement le Principe des Travaux Virtuels puisque :

$$\nabla J(\mathbf{u}) \cdot \mathbf{v} = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega - \int_{\Gamma} \mathbf{t} \cdot \mathbf{v} d\Gamma$$

Ainsi, résoudre les équations issues du Principe des Travaux Virtuels (base du problème formulé dans le [§1.2]) est équivalent à minimiser la fonctionnelle J qui représente l'énergie de la structure (énergie interne diminuée du travail des forces extérieures \mathbf{f} et \mathbf{t}).

2.3.3 Méthode de minimisation

La minimisation se fait de façon itérative, classiquement en deux temps à chaque itération :

- calcul d'une direction de recherche $\boldsymbol{\delta}$ le long de laquelle on va chercher l'itéré suivant,
- calcul du "meilleur" pas d'avancement ρ dans cette direction : $\mathbf{u}^{n+1} = \mathbf{u}^n + \rho \boldsymbol{\delta}$.

Dans un problème de minimisation, l'idée naturelle est d'avancer dans la direction opposée au gradient de la fonctionnelle, qui est localement la meilleure direction de descente puisque cette direction est normale aux lignes d'isovaleurs et dirigée dans le sens des valeurs décroissantes [Figure 2.3.3-a].

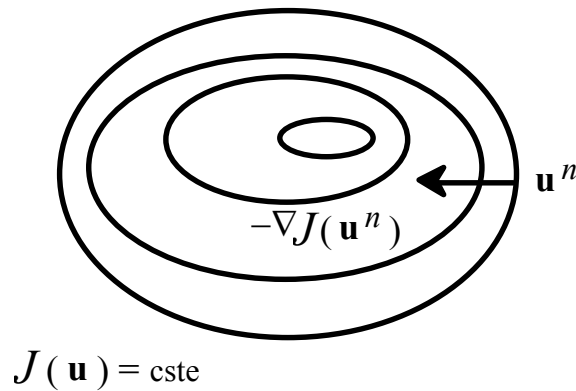


Figure 2.3.3-a

Cependant, il est possible d'améliorer le choix de la direction de descente en utilisant cette méthode de gradient dans une nouvelle métrique. C'est ce qui va nous permettre de retrouver les équations classiques de la méthode de NEWTON [§2.3.4].

Prenons l'exemple simple d'un problème à deux variables x et y pour lequel la fonctionnelle a la forme d'une ellipse dont le minimum est en $(\alpha/a, \beta/b)$:

$$J(x, y) = \frac{1}{2}ax^2 + \frac{1}{2}by^2 - \alpha x - \beta y$$

En choisissant comme direction de descente l'inverse du gradient de J , on passe d'un itéré au suivant (raisonnons sur x seulement) par :

$$x^{n+1} = x^n - (ax^n - \alpha),$$

qui ne pointe pas vers la solution puisque la normale en un point d'une ellipse ne passe en général pas par le centre de l'ellipse [Figure 2.3.3-b].

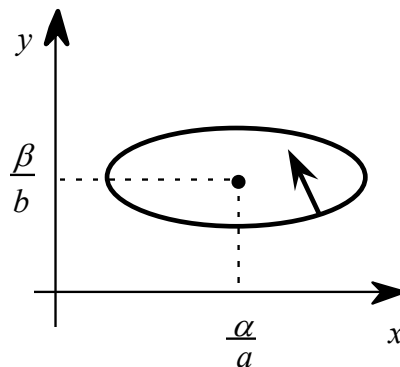


Figure 2.3.3-b

Par contre, si l'on effectue un changement de variables pour que les isovaleurs de J deviennent des cercles [Figure 2.3.3-c] :

$$\begin{cases} \bar{x} = \sqrt{a} x \\ \bar{y} = \sqrt{b} y \\ \bar{J}(\bar{x}, \bar{y}) = \frac{1}{2}(\bar{x}^2 + \bar{y}^2) - \frac{\alpha}{\sqrt{a}} \bar{x} - \frac{\beta}{\sqrt{b}} \bar{y} \end{cases}$$

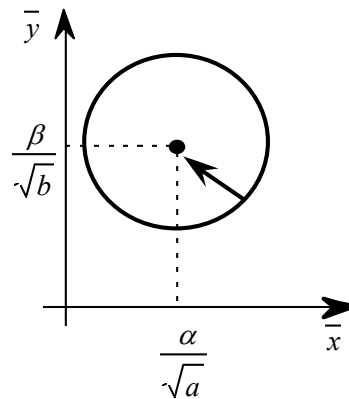


Figure 2.3.3-c

L'utilisation de la direction inverse du gradient de \bar{J} permet alors d'obtenir la solution en une itération :

$$\bar{x}^{n+1} = \bar{x}^n - (\bar{x}^n - \frac{\alpha}{\sqrt{a}}) = \frac{\alpha}{\sqrt{a}} \Rightarrow x^{n+1} = \frac{\alpha}{a}$$

Ainsi, l'utilisation de la méthode de gradient dans la nouvelle métrique permet une convergence immédiate. Dans un cas plus compliqué (fonctionnelle convexe mais différente d'une ellipse), la convergence n'est pas instantanée mais le changement de variables permet de réduire sensiblement le nombre d'itérations nécessaires.

2.3.4 Application à la minimisation de l'énergie

Pour simplifier, on va se placer dans le cas linéaire discrétisé où la fonctionnelle J vaut :

$$J(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{L}$$

où l'on note \mathbf{K} la matrice de rigidité de la structure, et \mathbf{L} le vecteur des chargements imposés.

Pour minimiser J , nous allons utiliser la même méthode de descente que précédemment en faisant au préalable un changement de variables tout à fait similaire. La matrice \mathbf{K} étant symétrique définie positive, ses valeurs propres sont réelles positives : on peut donc définir la "racine carrée" de \mathbf{K} que l'on notera $\sqrt{\mathbf{K}}$ (également symétrique). On pose $\bar{\mathbf{u}} = \sqrt{\mathbf{K}} \mathbf{u}$, la minimisation de J est alors équivalente à celle de :

$$\bar{J}(\bar{\mathbf{u}}) = \frac{1}{2} \bar{\mathbf{u}}^T \bar{\mathbf{u}} - \bar{\mathbf{u}}^T \sqrt{\mathbf{K}}^{-1} \mathbf{L}$$

En prenant comme direction de descente la direction inverse du gradient de \bar{J} , on obtient :

$$\bar{\mathbf{u}}^{n+1} = \bar{\mathbf{u}}^n - (\bar{\mathbf{u}}^n - \sqrt{\mathbf{K}}^{-1} \mathbf{L})$$

Soit, en revenant aux variables initiales :

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \mathbf{K}^{-1} (\mathbf{K} \mathbf{u}^n - \mathbf{L})$$

Ou encore :

$$\mathbf{K} (\mathbf{u}^{n+1} - \mathbf{u}^n) = \mathbf{L} - \mathbf{K} \mathbf{u}^n$$

On retrouve les équations de la méthode de NEWTON : la matrice \mathbf{K} est la Hessienne de la fonctionnelle J (matrice de la dérivée seconde) et le second membre est la différence du chargement et des forces nodales, appelé aussi résidu.

Ainsi la méthode de NEWTON peut-être interprétée comme résultant de la minimisation de l'énergie de la structure via une méthode de gradient appliquée après un changement de métrique.

2.3.5 Détermination du pas d'avancement

Revenons au problème réel, celui de la résolution de $\mathbf{R}(\mathbf{u}_i, t_i) = \mathbf{L}_i$. Ce problème peut être interprété comme la minimisation de :

$$W(\mathbf{u}_i) - \mathbf{u}_i^T \mathbf{L}_i,$$

où $W(\mathbf{u}_i)$ correspond à la discrétisation, sur la base des fonctions de forme, de l'énergie interne de la structure $\int_{\Omega} w(\boldsymbol{\varepsilon}(\mathbf{u})) d\Omega$.

On a calculé par la méthode de NEWTON un incrément de déplacement $\delta \mathbf{u}_i^{n+1}$ qui, dans le problème de minimisation, s'interprète comme une direction de recherche. On va calculer le pas d'avancement ρ dans cette direction permettant de minimiser la valeur de la fonctionnelle :

$$\text{Min}_{\rho \in \mathbb{R}} \left\{ W(\mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1}) - \mathbf{L}_i(\mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1}) \right\}$$

Pour trouver le minimum de cette fonction scalaire de ρ que l'on notera $f(\rho)$, on cherche le point où sa dérivée s'annule (cela revient à rendre orthogonaux le résidu final et la direction de recherche) :

$$f'(\rho) = [\delta \mathbf{u}_i^{n+1}]^T [\mathbf{Q}^T \boldsymbol{\sigma}(\mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1}) - \mathbf{L}_i] = 0$$

($f'(\rho)$ est la projection du résidu sur la direction de recherche)

Avec les notations du [§2.2.2] et en prenant en compte les réactions d'appui, l'équation scalaire à résoudre pour déterminer le pas d'avancement ρ , s'écrit :

$$[\delta \mathbf{u}_i^{n+1}]^T [\mathbf{Q}^T \boldsymbol{\sigma}(\mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1}) + \mathbf{B}^T (\boldsymbol{\lambda}_i^n + \rho \delta \boldsymbol{\lambda}_i^{n+1}) - \mathbf{L}_i^{méca}] = 0$$

Afin que la détermination de ρ ne soit pas trop coûteuse, on utilise une méthode de sécante dont le nombre maximum d'itérations est fixé par l'utilisateur. La méthode de sécante peut s'interpréter comme une méthode de NEWTON où la dérivée au point courant est approchée par la direction joignant le point courant et le point précédent :

$$\rho^{p+1} = \rho^p - \frac{\rho^p - \rho^{p-1}}{g^p - g^{p-1}} g^p = \frac{\rho^{p-1} g^p - \rho^p g^{p-1}}{g^p - g^{p-1}},$$

où l'on a noté $g^p = f'(\rho^p)$.

On part de $\rho^0 = 0$ et $\rho^1 = 1$. La méthode de sécante a un ordre de convergence de l'ordre de 1.6 [bib2]. Elle se représente schématiquement sur la figure [Figure 2.3.5-a].

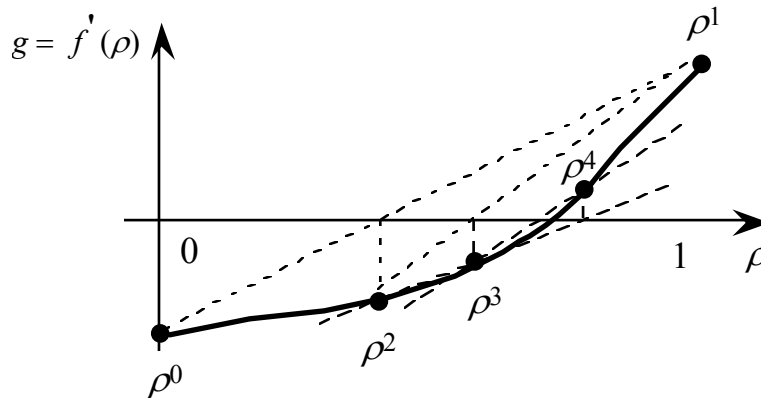


Figure 2.3.5-a

À la fin de la recherche linéaire, on actualise les déplacements et paramètres de Lagrange par :

$$\begin{cases} \Delta \mathbf{u}_i^{n+1} = \Delta \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \Delta \boldsymbol{\lambda}_i^{n+1} = \Delta \boldsymbol{\lambda}_i^n + \rho \delta \boldsymbol{\lambda}_i^{n+1} \end{cases}$$

Le test de convergence porte :

- sur le nombre maximum d'itérations de recherche linéaire indiqué par l'utilisateur sous le mot-clé `ITER_LINE_MAXI` du mot-clé facteur `NEWTON` (la valeur par défaut 0 inhibe la recherche linéaire, et ρ vaut alors 1),
- sur le critère `RESI_LINE_RELA` donné par $f(\rho) \leq \tau \cdot f(0)$, où τ vaut par défaut 0.1.

La recherche linéaire est en quelque sorte une "police d'assurance" permettant de se prémunir contre des divergences graves de la méthode de NEWTON. Lorsque la direction de recherche $(\delta \mathbf{u}_i^{n+1}, \delta \lambda_i^{n+1})$ est "mauvaise" (si la matrice tangente est trop souple, par exemple), l'algorithme de recherche linéaire aboutit à une faible valeur de ρ , ce qui évite d'aller "dans le décor". Il n'est pas nécessaire de faire beaucoup d'itérations dans la méthode de sécante (2 ou 3 suffisent pour éviter les catastrophes) car chacune coûte assez cher (il faut recalculer les forces internes) et on n'a pas l'ambition de trouver à chaque itération de NEWTON le ρ vraiment optimal.

2.4 Algorithme de `STAT_NON_LINE`

On utilisera comme précédemment l'indice i (comme "instant") pour noter le numéro d'un incrément de charge et l'exposant n (comme "newton") pour noter le numéro de l'itération de NEWTON en cours. L'algorithme utilisé dans l'opérateur `STAT_NON_LINE` peut alors s'écrire schématiquement de la façon suivante :

$(\mathbf{u}_0, \lambda_0)$ et σ_0 connus

Boucle sur des instants t_i (ou incréments de charge) : chargement $\mathbf{L}_i = \mathbf{L}(t_i)$

- $(\mathbf{u}_{i-1}, \lambda_{i-1})$ connus
- Prédiction : calcul de $\Delta \mathbf{u}_i^0$ et $\Delta \lambda_i^0$
- Boucle sur des itérations de NEWTON : calcul d'une suite $(\Delta \mathbf{u}_i^n, \Delta \lambda_i^n)$
 - $(\mathbf{u}_i^n, \lambda_i^n)$ et $(\Delta \mathbf{u}_i^n, \Delta \lambda_i^n)$ connus
 - Calcul des matrices et vecteurs associés aux charges suiveuses
 - Expression de la relation de comportement
 - calcul des contraintes σ_i^n et des variables internes α_i^n à partir des valeurs σ_{i-1} et α_{i-1} à l'équilibre précédent (t_{i-1}) et de l'incrément de déplacement $\Delta \mathbf{u}_i^n = \mathbf{u}_i^n - \mathbf{u}_{i-1}$ depuis cet équilibre
 - calcul des "forces nodales" : $\mathbf{Q}^T \sigma_i^n + \mathbf{B}^T \lambda_i^n$
 - calcul éventuel de la matrice de raideur tangente : $\mathbf{K}_i^n = \mathbf{K}(\mathbf{u}_i^n)$
 - Calcul de la direction de recherche $(\delta \mathbf{u}_i^{n+1}, \delta \lambda_i^{n+1})$ par résolution d'un système linéaire
 - Itérations de recherche linéaire : ρ
 - Actualisation des variables et de leurs incréments :

$$\begin{cases} \mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \lambda_i^{n+1} = \lambda_i^n + \rho \delta \lambda_i^{n+1} \end{cases} \quad \text{et} \quad \begin{cases} \Delta \mathbf{u}_i^{n+1} = \Delta \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \Delta \lambda_i^{n+1} = \Delta \lambda_i^n + \rho \delta \lambda_i^{n+1} \end{cases}$$
 - Test de convergence

- Archivage des résultats à l'instant t_i

$$\begin{cases} \mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i \\ \lambda_i = \lambda_{i-1} + \Delta \lambda_i \\ \sigma_i \\ \alpha_i \end{cases}$$

On remarque qu'il y a trois niveaux d'itérations imbriqués : une boucle extérieure sur les pas de temps, une boucle d'itérations (dites globales) de NEWTON et des sous-boucles éventuelles pour la recherche linéaire (si elle est demandée par l'utilisateur) et certaines relations de comportement nécessitant des itérations (dites internes), par exemple pour l'élasto-plasticité en contraintes planes.

3 Pilotage

On se reportera à la documentation [R5.03.80].

4 Grandes déformations

4.1 Objectif

Jusqu'à maintenant, nous avons fait l'hypothèse que déplacements et déformations restaient modérés de manière à respecter l'hypothèse des petites perturbations. Cette hypothèse devient caduque dans de nombreux cas et la méthode de résolution du problème doit alors intégrer l'évolution de la géométrie du problème, manipuler une cinématique particulière et utiliser une formulation adéquate de la loi de comportement.

En pratique, l'hypothèse des petites déformations peut être appliquée tant que le carré du module de la déformation reste inférieur à la précision des calculs envisagés. De même, l'hypothèse des petites rotations peut être appliquée tant que le produit entre le carré de l'angle de rotation et le module de la déformation reste inférieur à la précision des calculs envisagés.

Diverses alternatives existent au sein du *Code_Aster* ; notre objectif n'est pas ici d'en faire une présentation détaillée et nous renvoyons aux différents documents traitant spécifiquement chaque problématique :

- Relation de comportement élastique non linéaire en grands déplacements : [R5.03.20],
- Poutres en grands déplacements : [R5.03.40],
- Éléments de coques volumiques en non linéaire géométrique : [R5.07.05],
- Elasto(visco)plasticité, métallurgie et grandes déformations : [R4.04.03],
- Modélisation élastoplastique avec écrouissage isotrope en grandes déformations : [R5.03.21].

L'objectif est ici de présenter une fonctionnalité générale du code qui permet un traitement simple de problématiques grandes déformations : l'argument `PETIT_REAC` du mot clé `DEFORMATION` sous le mot clé facteur `COMP_INCR`.

Dans la suite, nous nous limiterons au cas des grandes déformations plastiques, qui permet de bien comprendre la problématique `PETIT_REAC`. Nous nous basons pour cela sur [bib4] et sur la documentation [R5.03.21].

4.2 Grandes déformations plastiques

Rappelons dans un premier temps les exigences d'une modélisation grandes déformations plastique en mécanique des milieux continus en terme d'équations d'équilibre, de description cinématique et de relation de comportement de manière à bien cerner les limitations de la modélisation PETIT_REAC.

Equations d'équilibre

Si l'on fait le choix d'écrire les équations d'équilibre sur la configuration actuelle Ω et d'utiliser le tenseur de contraintes de Cauchy σ , elles se résument à :

$$\int_{\Omega} \sigma : \nabla_{\Omega} \mathbf{v} \, d\Omega = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma} \mathbf{t} \cdot \mathbf{v} \, d\Gamma \quad \forall \mathbf{v} \in \mathbf{V}^{ad} \quad \text{éq 4.2-1}$$

L'équation précédente peut être écrite sur la configuration de référence ; c'est d'ailleurs ce qui est fait pour l'élasticité non linéaire en grandes déformations [R5.03.20]. Dans le cas de lois de comportement de nature incrémentale *i.e.* dont l'évolution est gouvernée par l'état actuel, cette écriture perd de son intérêt. On préfère alors l'écriture [éq 4.2-1], bien que la configuration actuelle Ω soit une inconnue du problème.

Description cinématique

Pour les lois de comportement plastique, on utilise communément comme mesure de la déformation le taux de déformation \mathbf{D} :

$$\mathbf{D} = \frac{1}{2} (\nabla_{\Omega} \dot{\mathbf{u}} + \nabla_{\Omega}^T \dot{\mathbf{u}})$$

En outre, on définit aussi le taux de rotation \mathbf{W} :

$$\mathbf{W} = \frac{1}{2} (\nabla_{\Omega} \dot{\mathbf{u}} - \nabla_{\Omega}^T \dot{\mathbf{u}})$$

Soulignons que dans les expressions précédentes, l'opérateur gradient est défini sur la configuration actuelle.

Relation de comportement élastoplastique

Avant d'écrire les relations de comportement, nous allons faire trois hypothèses simplificatrices :

- les déformations élastiques sont petites devant les déformations plastiques (ce qui se vérifie généralement bien dans le cas des métaux). Ceci permet de décomposer le taux de déformation de manière additive en une partie plastique et une partie élastique :

$$\mathbf{D} = \mathbf{D}^e + \mathbf{D}^p$$

- la déformation plastique est isochore. Ceci permet d'écrire que les tenseurs de contraintes de Kirchhoff τ (mesure de contrainte adéquat aux grandes déformations) et de Cauchy σ (intervenant dans les équations d'équilibre) sont identiques au premier ordre.
- le comportement est isotrope.

Sous ces hypothèses, les lois de comportement en grandes déformations peuvent s'écrire sous une forme proche des petites transformations. Ainsi pour la plasticité de Von-Mises à écrouissage isotrope obtient-on :

$$\left\{ \begin{array}{l} \sigma^j = \mathbf{A} : (\mathbf{D}^e - \mathbf{D}^p) \\ f(\sigma, p) = \sigma_{eq} - R(p) \\ \mathbf{D}^p = \frac{3}{2} \dot{p} \frac{\tilde{\sigma}}{\sigma_{eq}} \end{array} \right. \quad \text{éq 4.2-2}$$

où

- $\tilde{\sigma}$ est le déviateur de σ ,
- \mathbf{A} est le tenseur d'élasticité,
- σ^j désigne la dérivée de Jaumann de σ , qui tient compte de termes de transport définis à l'aide du taux de rotation \mathbf{W} et qui est donnée par :

$$\sigma^j = \dot{\sigma} - \mathbf{W} \cdot \sigma + \sigma \cdot \mathbf{W}$$

4.3 Fonctionnalité PETIT_REAC

Le principe de la modélisation PETIT_REAC consiste simplement à réactualiser la géométrie du problème au cours des itérations de Newton (et non à la fin de chaque pas de temps). Ceci signifie que toutes les quantités intervenant dans les équations du problème sont évaluées sur la configuration actuelle. Rien d'autre n'est modifié par rapport au cas petites perturbations.

Nous venons de voir quels ingrédients sont nécessaires à une intégration « propre » d'une loi de comportement élastoplastique en grandes déformations, moyennant trois hypothèses simplificatrices. Détaillons maintenant les différences qu'implique la résolution par PETIT_REAC.

Equations d'équilibre

Compte tenu de la réactualisation géométrique, les efforts intérieurs, premier terme de [éq 4.2-1], sont estimés de manière exacte, moyennant l'approximation par éléments finis. Le calcul des efforts extérieurs est lui indépendant de la résolution par PETIT_REAC : les charges mortes sont calculées sur la configuration de référence et les charges suiveuses sur la configuration actuelle. Les équations d'équilibre sont donc traitées de manière exacte.

Description cinématique

La description cinématique est la même que celle des petites perturbations. Ceci signifie qu'un incrément de déformation est calculé par :

$$\Delta \varepsilon = \frac{1}{2} (\nabla_{\Omega}(\Delta \mathbf{u}) + \nabla_{\Omega}^T(\Delta \mathbf{u}))$$

La déformation totale est ensuite la somme de chacun de ces incréments de déformation linéarisée, calculés sur des configurations différentes. Il est donc délicat de lui donner un sens physique et mieux vaut l'utiliser comme un indicateur du niveau de déformation atteint.

Relation de comportement élastoplastique

La différence entre la formulation [eq 4.2-2] et la formulation petites perturbations réside dans le remplacement de la dérivée de Jaumann par la simple dérivée en temps. L'inconvénient de cette dernière tient au fait qu'elle n'est pas incrémentalement objective *i.e.* pas invariante par rotation rigide de la structure (pour plus de détails, voir la documentation [R5.03.21]).

L'emploi de PETIT_REAC n'est donc pas approprié aux grandes rotations mais il l'est aux grandes déformations.

Intégration au sein du **Code_Aster**

En terme d'éléments finis, la résolution par `PETIT_REAC` implique à chaque pas de charge la résolution du même système non linéaire que précédemment :

$$\begin{cases} \mathbf{R}(\mathbf{u}_i, t_i) + \mathbf{B}^T \boldsymbol{\lambda}_i = \mathbf{L}(t_i) \\ \mathbf{B} \mathbf{u}_i = \mathbf{u}^d(t_i) \end{cases}$$

A la différence près que les forces internes sont formellement calculées par :

$$\mathbf{R}(\mathbf{u}_i, t_i) = \mathbf{Q}^T(\mathbf{u}_i) : \boldsymbol{\sigma}$$

où l'opérateur **Q** dépend des déplacements.

Dans ce cadre, le calcul de la matrice tangente mène formellement à :

$$\mathbf{K}_i'' = \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i'', t_i)} = \mathbf{Q}(\mathbf{u}) : \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i'', t_i)} + \frac{\partial \mathbf{Q}(\mathbf{u})}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i'', t_i)} : \boldsymbol{\sigma}$$

Le premier terme est la contribution du comportement, similaire à ce qui a été présenté en petites transformations, à la différence que cette contribution est ici évaluée en configuration actuelle. Le second terme est la contribution de la géométrie qui n'est pas présente en petites transformations. Dans le cadre de la résolution `PETIT_REAC`, ce terme n'est pas présent dans le calcul de la matrice tangente. On a donc :

$$\mathbf{K}_i'' = \mathbf{Q}(\mathbf{u}) : \frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{u}} \bigg|_{(\mathbf{u}_i'', t_i)}$$

On peut signaler que l'absence de la contribution géométrique dans la matrice tangente peut parfois rendre la convergence difficile. En outre, la présence de déformations plastiques importantes peut rendre le problème quasi-incompressible, du fait de la nature isochore de ces déformations. Des problèmes de blocage numérique peuvent alors apparaître qui peuvent être contournés par l'emploi d'éléments sous intégrés

5 Bibliographie

- [1] P. MIALON : Eléments d'analyse et de résolution numérique des relations de l'élastoplasticité. EDF - Bulletin de la Direction des Etudes et Recherches - Série C - N° 3 1986, p. 57 - 89.
- [2] J.F. MAITRE : Analyse numérique, cours photocopié de l'ENTPE.
- [3] J. SHI & M. A. CRISFIELD : "Combining arc-length control and line searches in path following", Communications in Numerical Methods in Engineering, Vol. 11, 793-803 (1995).
- [4] E. LORENTZ : Grandes déformations plastiques, modélisation dans Aster par PETIT_REAC, EDF - CR MMN 1536/07.