Algorithme non linéaire quasi-statique
(STAT_NON_LINE)

Résumé :

L'opérateur STAT_NON_LINE [U4.51.03] de *Code_Aster* permet dans le cas d'une sollicitation quasi-statique d'intégrer divers types de non-linéarités provenant du comportement du matériau, de grandes transformations géométriques ou des conditions de contact/frottement. On décrit ici l'algorithme de résolution global employé.

L'intégration des relations de comportement proprement dite est décrite dans les documents [R5.03....] et [R7.01....] , (par exemple [R5.03.02] pour l'élasto-plasticité), auxquels on pourra se reporter pour plus de détails.

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 les documents [R5.03.21], [R5.03.22] sur la thermoélastoplasticité à écrouissage isotrope.

Pour le contact frottement, il existe trois documents : [R5.03.50] sur la formulation discrète du contact/frottement, [R5.03.52] pour la formulation hybride par des éléments de contact/frottement, et [R5.03.53] sur le contact en grands glissements avec la méthode XFEM.

Pour tout ce qui concerne le pilotage, il faut se référer au document [R5.03.80].

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

1 Présentation....
1.1 Généralités....4
1.2 Types de non linéarités....4
1.2.1 Comportements non linéaires....4
1.2.2 Grandes transformations....4
1.2.3 Contact unilatéral et frottement....5
1.2.4 Conditions limites et chargements....5
1.3 Position du problème quasi-statique non linéaire....6
1.3.1 Problème général....6
1.3.2 Conditions limites de Dirichlet....7
2 Méthode de Newton....
2.1 Principe de la méthode....9
2.2 Adaptation de la méthode de Newton au problème posé....9
2.3 Phase de prédiction d'Euler....11
2.3.1 Linéarisation....11
2.3.2 Matrice tangente de prédiction....12
2.3.3 Variantes de la prédiction....12
2.3.4 Vecteur second membre des variables de commande....12
2.3.5 Vecteur second membre du chargement mécanique....13
2.3.6 Système linéaire....14
2.4 Phase de correction de Newton....14
2.4.1 Linéarisation....15
2.4.2 Système linéaire....15
2.4.3 Différence des matrices en prédiction et correction....15
2.5 Critères de convergence....16
2.6 Utilisation d'une matrice évolutive tangente-sécante....18
2.7 Méthode de Newton-Krylov....19
2.7.1 Principe général....19
2.7.2 Mise en œuvre....20
3 Conditions limites et chargements....
3.1 Classification....21
3.2 Chargements suiveurs....21
3.3 Conditions limites dualisées – Cas linéaire....22
3.3.1 Phase de prédiction d'Euler....23
3.3.2 Phase de correction de Newton....24
3.3.3 Adaptation des critères de convergence....25
3.4 Conditions limites dualisées – Cas non-linéaire....25
3.4.1 Problème d'équilibre sous conditions d'égalités non linéaires....26
3.4.2 Linéarisation du problème....27

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

3.4.3 Adaptation du critère de convergence....28
4 Recherche linéaire....
4.1 Principe....29
4.2 Minimisation d'une fonctionnelle....29
4.3 Méthode de minimisation....29
4.4 Application à la minimisation de l'énergie....31
4.5 Détermination du pas d'avancement....32
4.6 Calcul du coefficient de recherche linéaire....33
4.6.1 Méthode sécante (METHODE='CORDE')....33
4.6.2 Méthode mixte (METHODE='MIXTE')....34
4.6.3 Cas particulier : la méthode de Newton-Krylov....34
5 Pilotage....
6 Bibliographie....
7 Historique des versions du document....

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

1 Présentation

1.1 Généralités

STAT_NON_LINE est l'opérateur de *Code_Aster* permettant d'effectuer des calculs mécaniques non linéaires lorsque les effets d'inertie sont négligés (si l'on veut prendre en compte les effets d'inertie, il faut utiliser DYNA_NON_LINE, voir [R5.05.05]).

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, hydriques, 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 (par exemple CODE_SATURNE pour la mécanique des fluides, ...).

Il y a 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(des) fluide(s) et de l'équilibre mécanique [R7.01.10].

Il faut noter que lorsque l'on parle d'instant de calcul dans ce document, on fait quasiment toujours référence à un pseudo-temps, qui n'a pas de signification physique et qui ne sert qu'à paramétrer l'algorithme incrémental. Toutefois, l'instant garde une signification physique en visco-plasticité et quand les variables de commande en dépendent.

1.2 Types de non linéarités

1.2.1 Comportements non linéaires

Les relations de comportement non linéaires sont décrites dans les documents [R5.03....], pour les comportement généraux, et [R7.01....] pour les géo-matériaux. Le choix du type du comportement se fait par le mot-clé facteur COMPORTEMENT . Ceci conduit au travers de l'équation d'équilibre à un système non linéaire qui peut être de deux formes :

- Le système dépend 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).
- Le système dépend implicitement du champ de déplacements $\mathbf{u}$ par une ***équation différentielle implicite*** (par exemple l'élasto-plasticité, la visco-plasticité, l'hypo-élasticité, etc.). 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 σ et un champ de variables internes α) 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).

1.2.2 Grandes transformations

Dans les cas où l'hypothèse des petites perturbations (déplacements et déformations modérés) n'est pas vérifiée, 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 de *Code_Aster* ; notre objectif n'est pas ici d'en faire une présentation détaillée et nous renvoyons aux différents documents traitant spécifiquement chaque problématique. Il convient

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

de distinguer les formulations qui opèrent sur des éléments isoparamétriques massifs (2D ou 3D) et les formulations servant aux éléments de structure (poutres, plaques et coques). Pour les cas des éléments isoparamétriques massifs, on trouve trois grands types de formulation de la cinématique pour le cas des grandes déformations:

- La cinématique DEFORMATION='PETIT_REAC' permet de traiter une loi de comportement quelconque en grandes déformations. La loi est écrite en petites déformations et la prise en compte des grandes déformations se fait uniquement en réactualisant la géométrie. Cette formulation n'est pas incrémentalement objective. De plus, on ne peut l'utiliser que si le comportement est isotrope, si les déformations élastiques sont faibles devant les déformations plastiques, si les rotations restent faibles (inférieures à $10°$) et si on adopte une discrétisation en temps suffisamment fine. De plus, l'absence de la contribution géométrique dans la matrice tangente peut parfois rendre la convergence difficile (voir [R5.03.21 pour plus de détails).
- La cinématique DEFORMATION='SIMO_MIEHE' permet de traiter une loi de comportement élastoplastique avec écrouissage isotrope en grandes déformations, la loi de rupture ductile dite « Rousselier » ou l'élasto(visco)plasticité avec changement de phases pour la métallurgie (voir [R5.03.21], [R5.03.06] et [R4.04.03]). Cette formulation est incrémentalement objective, sans limitation sur le niveau des transformations appliquées mais elle n'est disponible que pour les trois lois de comportement citées et ne permet de traiter que les cas où le comportement est isotrope.
- La cinématique DEFORMATION='GDEF_LOG' permet aussi de traiter n'importe quelle loi de comportement hypo-élastoplastique. Elle est incrémentalement objective, sans limitation sur le niveau des transformations appliquées et permet de traiter le cas des comportements anisotropes (voir [R5.03.24]).

Pour traiter les grandes déformations élastiques, il faut employer un autre formalisme, appelé via DEFORMATION = 'GROT_GDEP' , qui est utilisable pour les relations de comportement hyper-élastiques non linéaire en grands déplacements (voir [R5.03.20] et [R5.03.22] pour le cas des petites déformations) ou pour la loi de comportement hyperélastique (voir [R5.03.19]).

Enfin, pour les éléments de structure (poutre, plaques ou coques), il existe des formulations spécifiques. On peut citer:

- Les poutres en grands déplacements (voir [R5.03.40]) ou les poutres multifibres en grands déplacements (voir [R3.08.09]). Mot-clef DEFORMATION='GROT_GDEP'.
- Les éléments de coques volumiques en non linéaire géométrique (voir [R3.07.05]).

Il n'existe pas d'éléments de structures (poutre, plaque ou coque) utilisables en grandes déformations dans *Code_Aster*.

1.2.3 Contact unilatéral et frottement

Pour le contact et le frottement, on se référera à trois documents : [R5.03.51] sur le contact discret avec frottement, [R5.03.52] pour la formulation hybride par des éléments de contact/frottement et [R5.03.53] sur le contact en grands glissements avec la méthode XFEM.

1.2.4 Conditions limites et chargements

Il est possible de définir des conditions limites non-linéaires, c'est-à-dire dépendant du déplacement de la structure. Il y a deux grandes catégories de conditions limites non-linéaires :

- Pour les chargements de Neumann, il s'agit essentiellement de la pression : celle-ci devant rester normale à la surface, le fait que la structure subisse de grandes transformations implique que le chargement *dépend* du déplacements. On consultera à ce propos les documents [R3.03.07], [R3.03.04] et le §3.2 ;
- Pour les chargements de Dirichlet, Code_Aster ne permet pour l'instant que le chargement de type LIAISON_SOLIDE (voir [U4.44.01]) qui rigidifie de façon exacte une partie de la structure. On consultera à ce propos [R3.02.02] ;

Dans les deux cas, il est nécessaire de spécifier *explicitement* que ce chargement est non-linéaire (on parle de chargement ***suiveur***) grâce au mot-clef TYPE_CHARGE dans le mot-clef-facteur EXCIT définissant les chargements (voir [U4.01.03]).

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

Remarque : en toute généralité, on peut assimiler le contact-frottement à une condition limite mixte (en déplacement et en pression) qui est non-linéaire.

1.3 Position du problème quasi-statique non linéaire

1.3.1 Problème général

En conséquence du paragraphe 1.1, on voit qu'il est légitime de considérer que le problème non linéaire a comme inconnue un déplacement et qu'il est paramétré par le temps. Considérons donc le problème non linéaire quasi-statique par l'expression classique du principe des travaux virtuels.

$$\mathbf{v}^T.\mathbf{L}^{\text{int}}(\mathbf{u},t)=\mathbf{v}^T.\mathbf{L}^{\text{ext}}(t) \qquad \forall\,\mathbf{v}=0 \tag{1}$$

où :

- t représente la variable d'instant ;
- $\mathbf{u}$ est le champ de déplacement pris à partir d'une configuration de référence ;
- $\mathbf{v}$ est le champ de déplacement virtuel cinématiquement admissible ;
- $\mathbf{L}^{\text{ext}}$ est le chargement mécanique extérieur auquel est soumis la structure (pression, force imposée,...) ;
- $\mathbf{L}^{\text{int}}$ représente les forces internes du problème de mécanique quasi-statique non linéaire. Dans le cas linéaire, on a $\mathbf{L}^{\text{int}}(\mathbf{u},t)=\mathbf{K}.\mathbf{u}$, où $\mathbf{K}$ est la matrice de rigidité de la structure.

Pour l'instant, on ne tient pas compte des conditions limites de Dirichlet. En fait, plus précisément, $\mathbf{L}^{\text{int}}(\mathbf{u},t)$ est relié au champ de contraintes $\boldsymbol{\sigma}$ par l'opérateur du travail des déformations virtuelles $\mathbf{Q}^T$ suivant la relation suivante:

$$\mathbf{L}^{\text{int}}(\mathbf{u},t)=\mathbf{Q}^T(\mathbf{u}).\boldsymbol{\sigma} \tag{2}$$

En petits déplacements, $\mathbf{Q}^T$ est indépendant des déplacements ; pour les grands déplacements, ce n'est plus le cas. On se donne une discrétisation de l'intervalle de temps à calculer:

$$t\rightarrow\left[t_0,\cdots,t_i,\cdots,t_n\right] \tag{3}$$

Le champ de contraintes $\boldsymbol{\sigma}_i$ à l'instant t_i s'écrit $\boldsymbol{\sigma}(\mathbf{u}_i,\beta_i,t_i,\mathbf{H}_{i-1})$, si l'on note β_i les champs de variables de commande 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 : $\mathbf{H}_{i-1}=\left\{\mathbf{u}_{i-1},\boldsymbol{\sigma}_{i-1},\boldsymbol{\alpha}_{i-1},t_{i-1}\right\}$.

Dans le cas général, la dépendance de l'opérateur $\mathbf{L}^{\text{int}}$ 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 apparaît notamment dans le cas de relations de comportement prenant en compte un phénomène d'écrouissage par le temps (time-hardening) ou dans le cas du vieillissement.

La formulation du problème quasi-statique consiste à exprimer l'équilibre de la structure (les forces internes sont égales aux forces externes) pour une suite d'instants de calcul $\left(t_i\right)_{1\leq i\leq I}$ qui paramètrent le chargement, on notera les quantités à l'instant t_i par l'indice i (par exemple $\mathbf{L}^{\text{int}}(\mathbf{u_i},t_i)=\mathbf{L}_i^{\text{int}}$) :

$$\mathbf{L}_i^{\text{int}}=\mathbf{L}_i^{\text{ext}} \tag{4}$$

Ce qui revient à annuler en $(\mathbf{u}_i,t_i)$ le vecteur $\mathbf{R}_i(\mathbf{u}_i,t_i)$ dit vecteur résidu d'équilibre, défini par :

$$\mathbf{R}_i(\mathbf{u}_i,t_i)=\mathbf{L}_i^{\text{int}}-\mathbf{L}_i^{\text{ext}} \tag{5}$$

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

L'état de la structure en t_0 est supposé connu. On effectue I incréments (ou pas) de charge. Les inconnues sont calculées de façon incrémentale par l'algorithme de résolution global (même pour les comportements élastiques). À partir de $\mathbf{u}_{i-1}$, solution satisfaisant l'équilibre en t_{i-1}, on détermine $\Delta\mathbf{u}_i$ qui permettront d'obtenir la solution en t_i :

$$\begin{cases} t_i = t_{i-1} + \Delta t_i \\ \mathbf{u}_i = \mathbf{u}_{i-1} + \Delta\mathbf{u}_i \end{cases} \tag{6}$$

L'incrément $\Delta\mathbf{u}_i$ est d'abord estimé en linéarisant le problème par rapport au temps autour de $(\mathbf{u}_{i-1}, t_{i-1})$ (phase dite de prédiction ou d'Euler). Puis on utilise une méthode de Newton ou une de ses variantes pour résoudre l'équation (4) de manière itérative (on calcule une suite $\Delta\mathbf{u}_i^n$ où l'exposant n est le numéro de l'itération). En plus de ces variables, 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 $\boldsymbol{\alpha}_{i-1}$ (voir [R5.03.02] pour un exemple).

1.3.2 Conditions limites de Dirichlet

Nous avons imposé que le champ des déplacements virtuels soit *cinématiquement* admissible, c'est-à-dire qu'il respecte les conditions aux limites en déplacements (ou conditions de Dirichlet). De manière générale, les conditions de Dirichlet ***linéaires*** s'écrivent :

$$\sum_{j=1}^{r} \alpha_j u_j = \beta(t) \tag{7}$$

Avec u_j la liste des r degrés de liberté, α_j les coefficients et β le second membre pouvant dépendre du temps.

Dans Code_Aster, il y a deux manières de prendre en compte ces conditions :

- Par une méthode dite « cinématique » (définition par AFFE_CHAR_CINE) qui utilise une méthode de pseudo élimination dans le système linéaire résultant (voir [U4.44.03]). Dans ce cas, on ne peut définir que des conditions limites simples telles que $r=1$;
- Par une méthode de dualisation (définition par AFFE_CHAR_MECA) qui utilise la technique des doubles Lagrange (voir [R3.01.01]).

Il est possible d'éliminer des conditions limites dualisées par l'usage d'une méthode spéciale que l'on décrit dans [R3.03.05], mais uniquement en utilisant le solveur PETSc.

Nous décrivons maintenant la manière de prendre en compte les conditions limites ***dualisées*** dans la résolution d'un problème non-linéaire. Dans un premier temps, nous supposerons que les conditions limites (7) peuvent s'écrire sous la forme discrète suivante :

$$\mathbf{B}.\mathbf{u} = \mathbf{u}^d(t) \tag{8}$$

$\mathbf{B}$ 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. La dualisation des conditions aux limites de Dirichlet $\mathbf{B}.\mathbf{u} = \mathbf{u}^d(t)$ conduit à modifier (4) de la manière suivante :

$$\begin{cases} \mathbf{L}^{\text{int}}(\mathbf{u}, t) + \mathbf{B}^T.\boldsymbol{\lambda} = \mathbf{L}^{\text{ext}}(t) \\ \mathbf{B}.\mathbf{u} = \mathbf{u}^d(t) \end{cases} \tag{9}$$

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. Le vecteur $\mathbf{B}^{\mathrm{T}}.\lambda$ s'interprète alors comme l'opposé des réactions d'appui aux nœuds correspondants.
En notant les quantités à l'instant t_i par l'indice i :

$$\begin{cases} \mathbf{L}_i^{\text{int}} + \mathbf{B}^T.\boldsymbol{\lambda}_i = \mathbf{L}_i^{\text{ext}} \\ \mathbf{B}.\mathbf{u}_i = \mathbf{u}_i^d \end{cases} \tag{10}$$

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

Le vecteur résidu d'équilibre $\mathbf{R}_i(\mathbf{u}_i, \boldsymbol{\lambda}_i, t_i)$ vaut maintenant :

$$\mathbf{R}_i(\mathbf{u}_i, \boldsymbol{\lambda}_i, t_i) = \begin{pmatrix} \mathbf{L}_i^{\text{int}} + \mathbf{B}^{\mathbf{T}}.\boldsymbol{\lambda}_i - \mathbf{L}_i^{\text{ext}} \\ \mathbf{B}.\mathbf{u}_i - \mathbf{u}_i^d \end{pmatrix} \quad (11)$$

Les inconnues sont toujours calculées de façon incrémentale par l'algorithme de résolution global. 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 :

$$\begin{cases} t_i = t_{i-1} + \Delta t_i \\ \mathbf{u}_i = \mathbf{u}_{i-1} + \Delta\mathbf{u}_i \\ \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \Delta\boldsymbol{\lambda}_i \end{cases} \quad (12)$$

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

2 Méthode de Newton

2.1 Principe de la méthode

La méthode de Newton est une méthode classique de résolution des équations du type recherche de zéro. Considérons une fonction vectorielle $\mathbf{F}$ non-linéaire du vecteur $\mathbf{x}$. On cherche le zéro de cette fonction, c'est à dire:

$$\mathbf{F}(\mathbf{x})=0 \tag{13}$$

La méthode de Newton consiste à construire une suite de vecteurs $\mathbf{x}^n$ 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 un autour de $\mathbf{x}^n$ et on exprime que $\mathbf{F}(\mathbf{x}^{n+1})$ doit être nul :

$$0=\mathbf{F}(\mathbf{x}^{n+1})\approx\mathbf{F}(\mathbf{x}^n)+\mathbf{F}'(\mathbf{x}^n)(\mathbf{x}^{n+1}-\mathbf{x}^n) \tag{14}$$

Soit encore:

$$\mathbf{F}'(\mathbf{x}^n)(\mathbf{x}^{n+1}-\mathbf{x}^n)=-\mathbf{F}(\mathbf{x}^n) \tag{15}$$

Finalement:

$$\mathbf{x}^{n+1}=\mathbf{x}^n-\left[\mathbf{F}'(\mathbf{x}^n)\right]^{-1}.\mathbf{F}(\mathbf{x}^n) \tag{16}$$

$\mathbf{F}'(\mathbf{x})$ 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 la dérivée directionnelle suivante:

$$\mathbf{F}'(\mathbf{x}).\mathbf{h}=\lim_{\theta\to 0}\frac{\mathbf{F}(\mathbf{x}+\theta.\mathbf{h})-\mathbf{F}(\mathbf{x})}{\theta} \tag{17}$$

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 :

$$\mathbf{F}'(\mathbf{x}).\mathbf{h}=\mathbf{h}^T.\nabla\mathbf{F}(\mathbf{x}) \tag{18}$$

C'est-à-dire le produit scalaire de $\mathbf{h}$ et du gradient de $\mathbf{F}$.

Lorsque l'on est proche de la solution, la convergence de la méthode de Newton est quadratique c'est-à-dire que le nombre de zéros 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 d'autant plus coûteux que la taille du problème est grande (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 Adaptation de la méthode de Newton au problème posé

Dans un premier temps, on ne prend pas en compte les conditions aux limites de Dirichlet. On doit résoudre un système (non-linéaire car dépendant de $\mathbf{u}_i$) de la forme :

$$\mathbf{L}_i^{\text{int}}(\mathbf{u}_i)=\mathbf{L}_i^{\text{méca}}(\mathbf{u}_i) \tag{19}$$

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

où $\mathbf{L}_i^{\text{méca}}$ désignera désormais, à l'instant t_i, la partie *mécanique* du chargement extérieur total $\mathbf{L}_i^{\text{ext}}$ afin de la distinguer du chargement thermique. Notons que le chargement mécanique $\mathbf{L}_i^{\text{méca}}(\mathbf{u}_i)$ peut dépendre des déplacements $\mathbf{u}_i$ dans le cas des forces dites « suiveuses » comme la pression ou la force centrifuge (voir [§3.2]). En utilisant les notations du [§2.1], cela revient à annuler la fonction vectorielle $\mathbf{R}$ définie par :

$$\mathbf{R}(\mathbf{u}_i, t_i) = \mathbf{L}_i^{\text{int}} - \mathbf{L}_i^{\text{méca}} \tag{20}$$

Les forces internes $\mathbf{L}_i^{\text{int}}$ peuvent ***symboliquement*** être notées $\mathbf{Q}_i^T . \boldsymbol{\sigma}_i$, où $\mathbf{Q}_i^T$ est la matrice associée à l'opérateur divergence (expression du travail du champ de déformations virtuelles). Les forces internes s'expriment alors :

$$\mathbf{L}_i^{\text{int}} = \mathbf{Q}_i^T . \boldsymbol{\sigma}_i = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}_i) : \boldsymbol{\sigma}(\mathbf{u}_i) . d\Omega \tag{21}$$

Et les forces du chargement mécanique:

$$\mathbf{L}_i^{\text{méca}} = \int_\Omega \mathbf{f}_i . \mathbf{w}_i . d\Omega + \int_\Gamma \mathbf{t}_i . \mathbf{w}_i . d\Gamma \tag{22}$$

Où:

- $\mathbf{w}_i$ désigne le champ des déplacements virtuels;
- $\mathbf{f}_i$ 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 .

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) :

$$\mathbf{K}_i^n . \Delta \mathbf{u}_i^{n+1} = \mathbf{L}_i^{\text{méca},n} - \mathbf{L}_i^{\text{int},n} \tag{23}$$

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 $\mathbf{L}_i^{\text{int},n}$ représente les forces internes à la $n^{\text{ième}}$ itération de Newton du $i^{\text{ème}}$ pas de charge. La quantité $\mathbf{R}_i^n = \left(\mathbf{L}_i^{\text{méca},n} - \mathbf{L}_i^{\text{int},n}\right)$ représente les forces non équilibrées, que l'on appelle le « résidu d'équilibre ». La matrice $\mathbf{K}_i^n$ est la matrice de l'application linéaire tangente de la fonction $\mathbf{R}_i^n$, elle vaut donc :

$$\mathbf{K}_i^n = \left.\frac{\partial \mathbf{R}_i^n}{\partial \mathbf{u}}\right|_{(\mathbf{u}_i^n, t_i)} = \left.\frac{\partial \mathbf{L}_i^{\text{int},n}}{\partial \mathbf{u}}\right|_{(\mathbf{u}_i^n, t_i)} - \left.\frac{\partial \mathbf{L}_i^{\text{méca},n}}{\partial \mathbf{u}}\right|_{(\mathbf{u}_i^n, t_i)} \tag{24}$$

En l'absence de forces suiveuses [§3.2], le second terme est nul. Il ne reste donc de la matrice $\mathbf{K}_i^n$ que la dérivée au point $\mathbf{u}_i^n$ des forces internes par rapport aux déplacements :

$$\mathbf{K}_i^n = \left.\frac{\partial \mathbf{L}_i^{\text{int},n}}{\partial \mathbf{u}}\right|_{(\mathbf{u}_i^n, t_i)} \tag{25}$$

Une petite erreur dans l'évaluation des forces internes peut avoir des conséquences graves, car c'est leur 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 $\mathbf{K}_{\text{élas}}$. 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 $\mathbf{K}_{\text{élas}}$) sont appelées méthodes de Newton modifiées ou méthodes de quasi-Newton. Le choix entre une matrice tangente (la dernière obtenue ou une matrice précédente) et une matrice élastique est effectué dans *Code_Aster* par l'intermédiaire du mot-clé `MATRICE='TANGENTE'` ou `MATRICE='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.

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

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

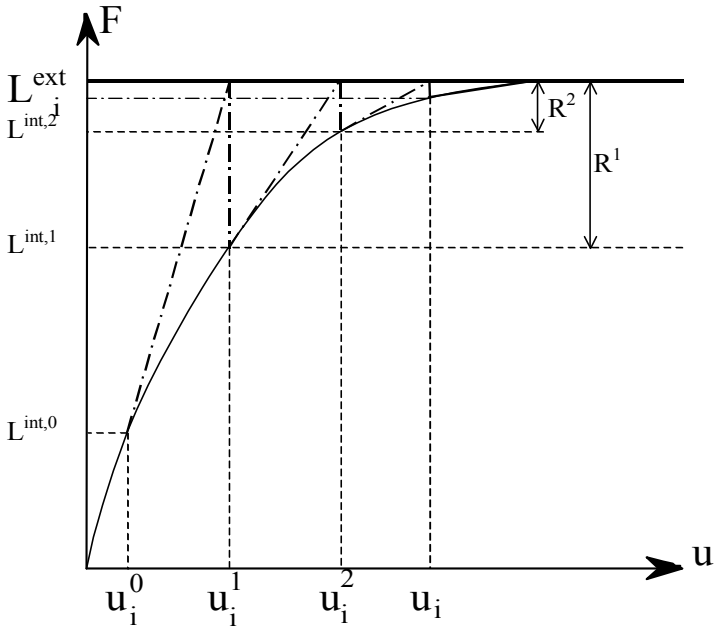


Figure 2.2.1-a

On enchaîne avec la boucle des itérations de Newton qui permet, à la convergence, d'obtenir les valeurs de $\Delta \mathbf{u}_i$, et donc celles de $\mathbf{u}_i$ par application de l'équation (6):

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i \tag{26}$$

2.3 Phase de prédiction d'Euler

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 $\Delta \mathbf{u}_i^0$. Pour cela, on linéarise par rapport au temps le problème continu : c'est ce qu'on appelle la phase de *prédiction*.

2.3.1 Linéarisation

On va donc linéariser le système (19) par rapport au temps autour de $\mathbf{u}_{i-1}$. On commence par linéariser les forces internes $\mathbf{L}_i^{\text{int}}$:

$$\mathbf{L}_i^{\text{int}} \approx \mathbf{L}_{i-1}^{\text{int}} + \left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} . \Delta \mathbf{u}_i^0 + \left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}} \tag{27}$$

On suppose que le chargement mécanique ne dépend pas du temps (les charges suiveuses sont exclues), donc :

$$\begin{cases} \mathbf{L}_i^{\text{méca}} = \mathbf{L}_{i-1}^{\text{méca}} + \Delta \mathbf{L}_i^{\text{méca}} \\ \mathbf{u}_i^d = \mathbf{u}_{i-1}^d + \Delta \mathbf{u}_i^d \end{cases} \tag{28}$$

En ré-injectant 27et 28 dans la première équation du système 26, on obtient pour l'équation d'équilibre:

$$\mathbf{L}_{i-1}^{\text{int}} + \left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} . \Delta \mathbf{u}_i^0 + \left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}} . \Delta t_i = \mathbf{L}_{i-1}^{\text{méca}} + \Delta \mathbf{L}_i^{\text{méca}} \tag{29}$$

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

On a équilibre à l'instant t_{i-1} , c'est-à-dire:

$$\mathbf{L}_{i-1}^{\text{int}} = \mathbf{L}_{i-1}^{\text{méca}} \tag{30}$$

Et il reste donc :

$$\left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} . \Delta \mathbf{u}_i^0 + \left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}} . \Delta t_i = \Delta \mathbf{L}_i^{\text{méca}} \tag{31}$$

On obtient le système d'équations permettant de calculer des valeurs prédictives $\Delta \mathbf{u}_i^0$:

$$\left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} . \Delta \mathbf{u}_i^0 = \Delta \mathbf{L}_i^{\text{méca}} - \left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}} . \Delta t_i \tag{32}$$

2.3.2 Matrice tangente de prédiction

La quantité $\left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}}$ désigne la dérivée partielle à temps constant de $\mathbf{L}_{i-1}^{\text{int}}$, elle peut se développer :

$$\mathbf{K}_{i-1} = \left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} = \left(\mathbf{Q}_{i-1}^T . \left.\frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} + \left.\frac{\partial \mathbf{Q}^T}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} . \boldsymbol{\sigma}_{i-1} \right) \tag{33}$$

La matrice $\mathbf{K}_{i-1}$ est appelée matrice tangente de prédiction. La dépendance de la matrice $\mathbf{Q}$ par rapport aux déplacements est négligée dans l'hypothèse des petits déplacements : le terme $\left.\frac{\partial \mathbf{Q}^T}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}}$, dit terme de rigidité géométrique, disparaît donc de 33. Ce terme est pris en compte pour les grandes transformations (voir §1.2.2). 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`.

2.3.3 Variantes de la prédiction

Il existe d'autres options de prédiction disponibles dans `STAT_NON_LINE`.

- On peut utiliser une matrice élastique $\mathbf{K}_{\text{élas}}$ à la place de la matrice tangente en vitesse $\mathbf{K}_{i-1}$, c'est l'option `PREDICTION='ELASTIQUE'` (option `RIGI_MECA`).
- On peut utiliser une matrice sécante $\mathbf{K}_{\text{sécante}}$ à la place de la matrice tangente en vitesse $\mathbf{K}_{i-1}$, c'est l'option `PREDICTION='SECANTE'` (option `RIGI_MECA_ELAS`). La matrice sécante est une matrice élastique dont le module de Young est utilisé en appliquant l'endommagement (voir par exemple [R5.03.18] pour plus de détails)
- On peut utiliser un incrément de déplacement précédemment calculé à la place de l'estimation, c'est l'option `PREDICTION='DEPL_CALCULE'`. Dans ce cas on ne fait aucune inversion de système et le $\Delta \mathbf{u}_i^0$ est directement donné. Voir documentation [U4.51.03] pour son usage.
- On peut utiliser un incrément de déplacement extrapolé par rapport au pas précédent. On calcule l'estimation de l'incrément de déplacement à partir de l'incrément total obtenu comme solution au pas de temps précédent (pondéré par le rapport des pas de temps). C'est l'option `PREDICTION='EXTRAPOL'`.

Dans ces deux derniers cas, afin d'assurer que le déplacement initial est cinématiquement admissible, on projette l'estimation sur l'ensemble des champs cinématiquement admissibles (i.e. satisfaisant les conditions aux limites de Dirichlet) selon la norme donnée par la matrice ***élastique*** , qui doit donc être calculée.

2.3.4 Vecteur second membre des variables de commande

Une variable de commande $\beta(t)$ est une quantité scalaire, fonction du temps et de l'espace[1], donnée a priori par l'utilisateur via le mot-clef `AFFE_VARC` dans l'opérateur `AFFE_MATERIAU`. C'est un *paramètre* du problème

1 Pour être plus précis, une variable de commande est affectée à une maille. Il ne s'agit donc pas d'une fonction de la géométrie (et donc des déplacements), mais de la topologie du maillage.

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

et non une *inconnue*. La quantité $\left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}}$ désigne la différentielle partielle, par rapport à t et à $\mathbf{u}$ constant, de $\mathbf{L}^{\text{int}}=\mathbf{Q}.\boldsymbol{\sigma}(t,\beta(t))$. Cette notation particulière a pour but d'attirer l'attention sur le fait que pour n_{varc} variables de commandes, on écrit la différentielle totale:

$$\frac{\delta \boldsymbol{\sigma}}{\delta t}=\frac{\partial \boldsymbol{\sigma}}{\partial t}+\sum_{j=1,n_{varc}}\frac{\delta \boldsymbol{\sigma}}{\delta \beta^j}.\frac{\delta \beta^j}{\delta t} \tag{34}$$

Si on prend comme exemple la variable de commande décrivant la température θ :

$$\frac{\delta \boldsymbol{\sigma}}{\delta t}=\frac{\partial \boldsymbol{\sigma}}{\partial t}+\frac{\delta \boldsymbol{\sigma}}{\delta \theta}.\frac{\delta \theta}{\delta t} \tag{35}$$

On suppose que la température varie linéairement entre les deux instants:

$$\frac{\delta \theta}{\delta t}=\frac{\Delta \theta_i}{\Delta t_i} \tag{36}$$

La dépendance de $\boldsymbol{\sigma}$ par rapport au temps et par rapport à la température permet d'écrire :

$$\left.\frac{\delta \mathbf{L}^{int}}{\delta t}\right|_{t_{i-1}}=\frac{\delta}{\delta t}\left(\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}\right)=\mathbf{Q}_{i-1}^T.\left(\left.\frac{\partial \boldsymbol{\sigma}}{\partial t}\right|_{t_{i-1}}+\left.\frac{\partial \boldsymbol{\sigma}}{\partial \theta}\right|_{t_{i-1}}.\frac{\Delta \theta_i}{\Delta t_i}\right) \tag{37}$$

Car $\mathbf{Q}$ ne dépend pas du temps et donc $\frac{\delta}{\delta t}\left(\mathbf{Q}^T\right)=0$. Le vecteur $\Delta \mathbf{L}_i^{\text{varc}}$, dont l'expression est donnée par 38, est l'incrément de chargement de température (attention au changement de signe !) que l'on a généralisé à toutes les *variables de commandes* : température, irradiation, phases métallurgiques (voir *[R4.04.02]),*…

$$\Delta \mathbf{L}_i^{varc}=-\left.\frac{\delta \mathbf{L}^{int}}{\delta t}\right|_{t_{i-1}}.\Delta t_i=-\mathbf{Q}_{i-1}^T.\left(\left.\frac{\partial \boldsymbol{\sigma}}{\partial t}\right|_{t_{i-1}}.\Delta t_i+\sum_{j=1,n_{varc}}\left(\left.\frac{\partial \boldsymbol{\sigma}}{\partial \beta^j}\right|_{t_{i-1}}.\Delta \beta_i^j\right)\right) \tag{38}$$

On ne tient pas compte actuellement de la dépendance explicite des contraintes par rapport au temps et donc le premier terme de 38 vaut zéro. Et donc finalement :

$$\Delta \mathbf{L}_i^{varc}=-\mathbf{Q}_{i-1}^T.\left(\sum_{j=1,n_{varc}}\left(\left.\frac{\partial \boldsymbol{\sigma}}{\partial \beta^j}\right|_{t_{i-1}}.\Delta \beta_i^j\right)\right) \tag{39}$$

L'incrément de chargement *des variables de commande* $\Delta \mathbf{L}_i^{\text{varc}}$, issu de la dérivation des forces internes par rapport aux variables de commande est une estimation de l'effet d'une variation des variables de commandes.

Dans le cas de la température, si l'on note K le module de compression hydrostatique et α le coefficient de dilatation thermique, la contrainte thermique s'écrit:

$$\boldsymbol{\sigma}_i^{\text{ther}}=-3.K.\alpha.\Delta\theta_i.\mathbf{I}+\boldsymbol{\sigma}_{i-1}^{\text{ther}} \quad \text{si } \Delta\theta_i=\theta_i-\theta_{i-1} \tag{40}$$

Où $\mathbf{I}$ est la matrice identité. Et donc, en appliquant 39:

$$\Delta \mathbf{L}_i^{ther}=-\left(\mathbf{Q}_{i-1}^T.\left.\frac{\partial \boldsymbol{\sigma}}{\partial \theta}\right|_{t_{i-1}}\right).\Delta\theta_i=3.K.\alpha.\Delta\theta_i.\left(\mathbf{Q}_{i-1}^T.\mathbf{I}\right) \tag{41}$$

Dans le cas élastique, ce sont les forces internes 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.

2.3.5 Vecteur second membre du chargement mécanique

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

L'incrément de chargement *mécanique* $\Delta \mathbf{L}_i^{\text{méca}}$ est composé des charges mortes (indépendantes de la géométrie, comme la pesanteur) et des charges suiveuses. En réalité, il existe des cas (le premier incrément de charge, par exemple) où $\mathbf{L}_{i-1}^{\text{méca}}$ est inconnu. On rappelle que l'incrément de chargement (28) s'écrit :

$$\mathbf{L}_i^{\text{méca}} = \mathbf{L}_{i-1}^{\text{méca}} + \Delta \mathbf{L}_i^{\text{méca}} \tag{42}$$

On a équilibre à l'instant t_{i-1} , donc en appliquant 30:

$$\Delta \mathbf{L}_i^{\text{méca}} = \mathbf{L}_i^{\text{méca}} - \mathbf{L}_{i-1}^{\text{int}} \tag{43}$$

L'expression des forces internes au pas de temps précédent $\mathbf{L}_{i-1}^{\text{int}}$ implique soit de sauvegarder ce vecteur du calcul précédent s'il existe (reprise d'un calcul antérieur), soit d'intégrer la loi de comportement à partir de l'état initial donné par l'utilisateur (ce qui peut être coûteux). Par souci de simplicité et d'efficacité, on choisit de ne pas réintégrer la loi de comportement et on exprime simplement les forces internes comme les forces nodales en prenant les contraintes connues à cet instant, soit :

$$\mathbf{L}_{i-1}^{int} = \mathbf{Q}_{i-1}^{T} . \boldsymbol{\sigma}_{i-1} \tag{44}$$

D'où la nouvelle expression:

$$\Delta \mathbf{L}_i^{\text{méca}} = \mathbf{L}_i^{\text{méca}} - \mathbf{Q}_{i-1}^{T} . \boldsymbol{\sigma}_{i-1} \tag{45}$$

Le calcul direct à partir de (44) demande à l'utilisateur de prendre garde à la cohérence entre le champ des contraintes, les champs de déplacements et de variables internes (DEPL , SIGM et VARI dans ETAT_INIT). vis-à-vis de l'intégration de la loi de comportement dans le cas d'une reprise de calcul.

2.3.6 Système linéaire

En ré-injectant l'expression de $\left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}}$ (équation 33) , de $\left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}}$ (équation 38) et $\Delta \mathbf{L}_i^{\text{méca}}$ (équation 45) dans 32, le système d'équations en prédiction s'écrit :

$$\mathbf{K}_{i-1} . \Delta \mathbf{u}_i^0 = \mathbf{L}_i^{\text{méca}} - \mathbf{Q}_{i-1}^{T} . \boldsymbol{\sigma}_{i-1} + \Delta \mathbf{L}_i^{\text{varc}} \tag{46}$$

2.4 Phase de correction de Newton

A l'issue de la phase de prédiction, nous nous retrouvons avec une estimation de l'incrément des déplacements $\Delta \mathbf{u}_i^0$. Si cette estimation est exacte (modulo l'application des critères de convergence décrits au §2.5), alors on obtient la solution ***convergée*** pour le pas de temps t_i :

$$\mathbf{u}_i^{\text{convergé}} = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i^0 \tag{47}$$

Mais si ce n'est pas le cas, on doit trouver la valeur $\Delta \mathbf{u}_i$ de l'incrément de déplacement à partir de la valeur $\mathbf{u}_{i-1}$ obtenues à l'équilibre précédent (instant t_{i-1}):

$$\mathbf{u}_i^{\text{convergé}} = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i \tag{48}$$

Si la phase de prédiction a convergé, on a donc trivialement:

$$\Delta \mathbf{u}_i = \Delta \mathbf{u}_i^0 \tag{49}$$

Sinon, on prend comme valeur initiale $\Delta \mathbf{u}_i^0$ obtenue à l'issue de la phase de prédiction, avant de corriger par les itérations $\delta \mathbf{u}_i^n$ de la méthode de Newton. Avec un nombre n_{CV} suffisant d'itérations de Newton (toujours avec l'arbitrage du critère de convergence):

$$\mathbf{u}_i^{\text{convergé}} = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i^0 + \sum_{j=1}^{n=n_{CV}} \delta \mathbf{u}_i^j \tag{50}$$

Tant qu'on a pas convergé (si le nombre d'itérations de Newton n'est pas suffisant), on note:

$$\Delta \mathbf{u}_i^n = \Delta \mathbf{u}_i^0 + \sum_{j=1}^{n<n_{CV}} \delta \mathbf{u}_i^j \tag{51}$$

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

Le déplacement total, pour le pas de temps i et l'itération de Newton n s'écrira donc:

$$\mathbf{u}_i^n = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i^n \tag{52}$$

À chaque itération, on doit résoudre un système permettant de déterminer $\delta \mathbf{u}_i^n$, les incréments des déplacements depuis le résultat $\mathbf{u}_i^{n-1}$ de l'itération précédente:

$$\mathbf{u}_i^{n-1} = \mathbf{u}_{i-1} + \mathbf{u}_i^0 + \sum_{j=1}^{n-1} \delta \mathbf{u}_i^j \tag{53}$$

On note aussi:

$$\Delta \mathbf{u}_i^{n-1} = \Delta \mathbf{u}_i^0 + \sum_{j=1}^{n-1} \delta \mathbf{u}_i^j \tag{54}$$

Soit encore:

$$\mathbf{u}_i^n = \mathbf{u}_i^{n-1} + \delta \mathbf{u}_i^n = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i^{n-1} + \delta \mathbf{u}_i^n \tag{55}$$

2.4.1 Linéarisation

On doit linéariser le système (19) par rapport aux inconnues en $\mathbf{u}_i^n$ à t_i constant. On commence par linéariser les forces internes $\mathbf{L}_i^{\text{int},n}$:

$$\mathbf{L}_i^{\text{int},n} \approx \mathbf{L}_i^{\text{int},n-1} + \left. \frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}} \right|_{\mathbf{u}_i^{n-1}} . \delta \mathbf{u}_i^n \tag{56}$$

On suppose que le chargement mécanique ne dépend pas du temps (les charges suiveuses sont exclues).

2.4.2 Système linéaire

En ré-injectant (56) dans l' équation (19), on obtient pour l'équation d'équilibre:

$$\mathbf{L}_i^{\text{int},n-1} + \left. \frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}} \right|_{\mathbf{u}_i^{n-1}} . \delta \mathbf{u}_i^n = \mathbf{L}_i^{\text{méca}} \tag{57}$$

La quantité $\left. \frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}} \right|_{\mathbf{u}_i^{n-1}}$ est appelée matrice tangente cohérente et elle est notée $\mathbf{K}_i^{n-1}$:

$$\mathbf{K}_i^{n-1} = \left. \frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}} \right|_{\mathbf{u}_i^{n-1}} \tag{58}$$

Le système à résoudre s'écrit finalement:

$$\mathbf{K}_i^{n-1} . \delta \mathbf{u}_i^n = \mathbf{L}_i^{\text{méca}} - \mathbf{L}_i^{\text{int},n-1} \tag{59}$$

Le vecteur des forces internes $\mathbf{L}_i^{\text{int},n-1}$ est calculé à partir des contraintes $\boldsymbol{\sigma}_i^{n-1}$. Celles-ci étant calculées à partir des déplacements $\mathbf{u}_i^{n-1}$ par l'intermédiaire de la relation de comportement du matériau [§1.1]. En fait, dans le cas des comportements ***incrémentaux***, $\boldsymbol{\sigma}_i^{n-1}$ est calculé à partir de $(\boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1})$ et de l'incrément de déformation $\boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^{n-1})$ induit par l'incrément de déplacement depuis le début du processus itératif (y compris la phase de prédiction) ou par le gradient de la transformation $\mathbf{F}$ dans le cas des grandes transformations.

2.4.3 Différence des matrices en prédiction et correction

Il est important de souligner que la matrice tangente issue de l'option `RIGI_MECA_TANG` (phase de prédiction) et la matrice tangente issue de l'option `FULL_MECA` (phase de correction) 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

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

équations d'équilibre par rapport au ***temps*** autour de $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$ *i.e.* autour de l'équilibre à l'instant t_{i-1}. C'est donc la matrice tangente du système convergé 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 $\mathbf{u}_i^n$ *i.e.* autour de l'équilibre à l'instant t_i.

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. 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 ou cinématique linéaire.

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

- Calculer les contraintes $\boldsymbol{\sigma}_i^n$ et les variables internes $\boldsymbol{\alpha}_i^n$ à partir de l'état initial $(\boldsymbol{\sigma}_{i-1}, \boldsymbol{\alpha}_{i-1})$ et de l'incrément de déformation $\boldsymbol{\varepsilon}(\Delta \mathbf{u}_i^{n-1})$ induit par l'incrément de déplacement depuis le début du processus itératif (y compris la phase de prédiction).
- Calculer les forces internes $\mathbf{L}_i^{\text{int},n} = \mathbf{Q}_i^T . \boldsymbol{\sigma}_i^n$.
- Calculer (éventuellement) la matrice tangente (option RIGI_MECA_TANG pour la phase de prédiction, option FULL_MECA pour les itérations de Newton).

2.5 Critères de convergence

À la fin de la prédiction et à chaque itération de Newton, on doit estimer si le processus itératif a convergé (l'équilibre de la structure est atteint). On se place au pas de temps courant t_i et à l'itération de Newton n (étant entendu que la valeur $n=0$ correspond à la prédiction). Il existe quatre critères de convergence.
Le critère RESI_GLOB_MAXI consiste à vérifier que la norme infinie[2] du résidu est inférieure à la valeur γ spécifiée par l'utilisateur.

$$\left\| \mathbf{Q}^T . \boldsymbol{\sigma}_i^n - \mathbf{L}_i^{\text{méca}} \right\|_\infty \leq \gamma \tag{60}$$

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 RESI_GLOB_RELA (choisi par défaut) revient à vérifier que le résidu est suffisamment petit, comme précédemment, et ceci relativement à une quantité représentative du chargement.

$$\frac{\left\| \mathbf{Q}^T . \boldsymbol{\sigma}_i^n - \mathbf{L}_i^{\text{méca}} \right\|_\infty}{\left\| \mathbf{L}_i^{\text{méca}} + \mathbf{L}_i^{\text{varc}} \right\|_\infty} \leq \eta \tag{61}$$

η étant la précision relative souhaitée donnée par l'utilisateur sous le mot-clef RESI_GLOB_RELA (ou la valeur par défaut de 10^{-6}).
On peut remarquer que, dans le cas d'utilisation de RESI_GLOB_RELA, le critère peut devenir singulier si le chargement extérieur $\mathbf{L}_i^{\text{méca}} + \mathbf{L}_i^{\text{varc}}$ 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 :

$$\forall\, j \in \{\text{ddl}\} \qquad \left| \left(\mathbf{Q}^T . \boldsymbol{\sigma}_i^n - \mathbf{L}_i^{\text{méca}} \right)_j \right| \leq \varepsilon \, . \mathbf{F}_j^{\text{ref}} \tag{62}$$

2 La norme infinie correspond simplement à la composante maximale de la valeur absolue du vecteur

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

Plus précisément, la force nodale de référence $\mathbf{F}_j^{\text{ref}}$ est construite à partir de la donnée d'une amplitude de référence A^{ref} qui peut être :

- Une contrainte ;
- Une pression, une température dans le cas de la THM ;
- Une force généralisée dans le cas des poutres ou des coques ;
- autres …

La liste est accessible dans la documentation d'utilisation de la commande STAT_NON_LINE [U4.51.03], description de l'opérande RESI_REFE_RELA.

Si on prend comme exemple une contrainte, l'amplitude de référence σ^{ref} étant donnée par l'utilisateur via le mot-clef SIGM_REFE . A partir de cette amplitude de contrainte de référence, on définit le tenseur $\boldsymbol{\sigma}^{\text{test}}$: 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 $\tilde{R}_i^e$ (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) :

$$\tilde{R}_i^e = \frac{1}{N} . \sum_{\alpha=1}^{N} \sum_{j=1}^{M} | B_{i,j}^{\alpha} . \sigma_j^{\text{test}} | . \omega^{\alpha} \tag{63}$$

Avec N le nombre de points de Gauss de l'élément, M le nombre de composantes du tenseur des contraintes ; l'exposant α servant à noter l'évaluation de quantité au point de Gauss. Par exemple ω^{α} sont les poids des points de Gauss.

Remarque : Pour certains éléments, comme les barres, les grilles ou les membranes, cette définition aboutit à des résidus de référence nuls sur certains axes. Pour y remédier, on détermine les forces de référence nodales via un calcul d'ordre de grandeur basé sur la taille de l'élément.

La force nodale de référence est alors définie par :

$$\mathbf{F}_i^{\text{ref}} = \min_{e \in \Gamma_i} \tilde{R}_i^e \tag{64}$$

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

Le quatrième critère est le critère RESI_COMP_RELA : l'idée de ce critère est de séparer les différentes contributions du résidu composantes par composantes (au sens DX, DY, DZ, PRE1, PRE2, TEMP). Chacun des vecteurs obtenus sera ensuite normé par la force interne correspondant à ce résidu. Ce choix de critère de convergence n'a de sens que pour les problèmes fortement évolutifs, typiquement les problèmes THM. Qui plus est, ce choix sera efficace dés lors que l'on aura des problèmes à forts contrastes. Ce sont en effet les zones à «fort gradient » qui vont contrôler la convergence. On définit $\mathbf{F}^n(c)$ la partie du résidu $\mathbf{Q}^T . \boldsymbol{\sigma}_i^n - \mathbf{L}_i^{\text{méca}}$ correspondant à la composante c et $\mathbf{L}^{\text{int},0}(c)$ le vecteur des forces internes en prédiction correspondant à ce même composant c . Le critère RESI_COMP_RELA revient alors à vérifier que ce résidu est suffisamment petit, c'est-à-dire :

$$\max_{c=1,\ldots,nbcmp} \left(\frac{\max_{d=1,\ldots,nbddl} |\mathbf{F}^n(c,d)|}{\max_{d=1,\ldots,nbddl} |\mathbf{L}^{\text{int},0}(c,d)|} \right) < \varepsilon \tag{65}$$

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).

Pour les résidus RESI_GLOB_RELA et RESI_GLOB_MAXI , toutes les composantes du champ de déplacement sont utilisés dans l'évaluation de la norme $\|.\|_{\infty}$, sauf dans deux cas où un traitement particulier est fait au niveau du choix des composantes :

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

- Pour les chargements de type AFFE_CHAR_CINE , le degré de liberté concerné est ignoré dans l'évaluation de la norme du résidu car la procédure d'élimination des inconnues ne permet pas d'accéder aux réactions d'appuis ;
- Pour le contact continu, les composantes LAGR_C et LAGR_F1/LAGR_F2 sont ignorés dans l'évaluation de la norme car la loi de Signorini-Coulomb est déjà vérifié dans l'algorithme (voir [R5.03.52]) et que ces termes sont dimensionnellement incohérents avec ceux relatifs aux déplacements ; Par contre, pour le cas du contact dans XFEM, ces composantes sont conservées car elles servent à vérifier la condition LBB ;

2.6 Utilisation d'une matrice évolutive tangente-sécante

La méthode décrite dans ce paragraphe s'applique exclusivement aux problèmes fortement non-linéaires, où une méthode de Newton classique échoue pour tout type de choix de matrice, pour la phase de prédiction ou de correction. Typiquement, la méthode de Newton habituelle est mise en défaut pour les problèmes mal-posés provenant de l'utilisation des lois de comportement adoucissantes (voir par exemple R5.04.01) .

Dans ces situations, une non-convergence se manifeste lorsque l'algorithme ne parvient pas à choisir entre plusieurs solutions admissibles, dans un incrément de (pseudo)-temps donné. Ce défaut de convergence au niveau global se traduit le plus souvent au niveau local (i.e. au point d'intégration) par une alternance répétée entre un état non-dissipatif (élastique) et un état dissipatif (plasticité, endommagement, …) au cours des itérations de Newton consécutives.

Une des stratégies consiste à utiliser la notion de pilotage (voir R5.03.80) pour pallier les insuffisances de Newton. L'autre stratégie consiste à modifier la matrice tangente. C'est cette dernière stratégie que l'on détaille ici.

En suivant l'état de chaque point d'intégration d'une itération à l'autre, on peut repérer les points responsables d'une non-convergence globale. Une fois ces points repérés, on décide de modifier la matrice en espérant que cette modification permettra une convergence globale.

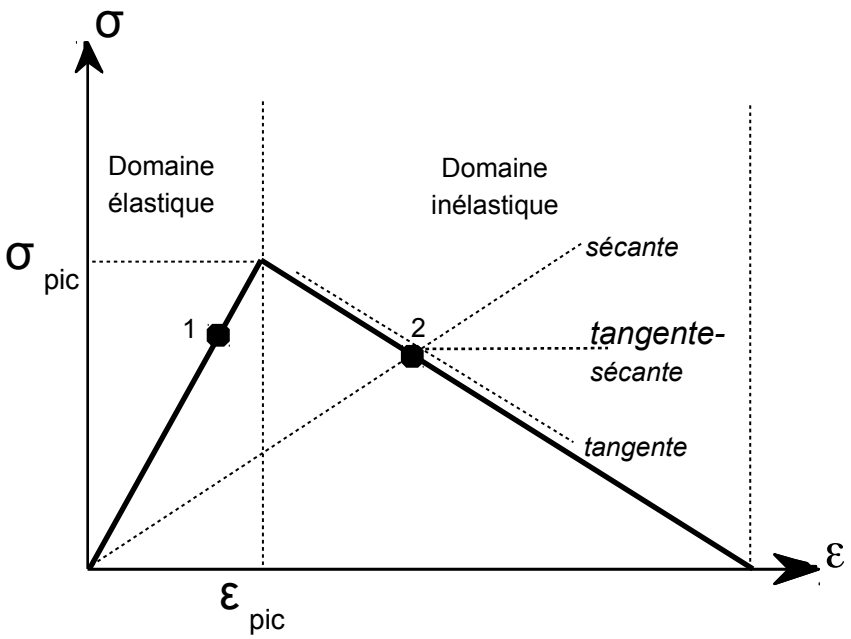


Figure 2-2.6-a: Loi de comportement adoucissante, utilisation d'une matrice mixte tangente-sécante

L'approche baptisée tangente-sécante, activée sous le mot-clef COMPORTEMENT avec TANGENTE_SECANTE='OUI', se justifie par le raisonnement suivant : si la méthode de Newton utilisant la matrice tangente cohérente ne converge pas, c'est souvent parce qu'un certain nombre (variable) de points

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

d'intégration change d'état (non-dissipatif/dissipatif) entre deux itérations globales de Newton. Au niveau local (voir Figure 2-2.6-a), cela veut dire qu'on continue à alterner entre le domaine 1 ($\varepsilon<\varepsilon_{pic}$) et le domaine 2 ($\varepsilon>\varepsilon_{pic}$). A cause du changement de pente entre 1 et 2, l'utilisation d'une matrice sécante ou tangente cohérente conduit à une approximation très pauvre, d'où l'intérêt de la modifier. Le choix que l'on présente consiste à construire la matrice tangente comme une combinaison linéaire entre la matrice tangente cohérente et la matrice sécante, les deux étant déterminées par les lois de comportement. Actuellement, l'approche n'est disponible que pour la loi de comportement adoucissante ENDO_ISOT_BETON.

Pour gérer l'amorçage de l'option tangente-sécante, on introduit une variable interne supplémentaire par rapport aux variables internes existantes, $\boldsymbol{\alpha}=(\alpha_1,\ldots\alpha_n,\alpha_{n+1})$ (avec n le nombre de variables internes du modèle utilisé). Cette variable traduit l'alternance éventuelle entre l'état élastique et l'état adoucissant d'un point de Gauss. On l'initialise à la première itération de Newton de chaque nouveau pas de temps, puis on la fait évoluer pour savoir, en chaque point de Gauss, le nombre d'alternances successives entre les deux états. En ayant cette information, on peut estimer qu'à partir d'un certain seuil (par exemple $S_0=3$ alternances), l'algorithme de Newton ne pourra plus converger pour l'incrément de temps courant et qu'il est nécessaire de modifier la matrice tangente. Pour modifier la matrice, on s'appuie directement sur la manière dont la matrice tangente cohérente dans ENDO_ISOT_BETON est construite (voir [R7.01.04]). Il s'agit de faire la somme entre la partie dissipative et la partie non dissipative.

$$\mathbf{K}_{\text{tg}}=\mathbf{K}_{\text{sc}}+\mathbf{K}_{\text{end}} \tag{66}$$

où $\mathbf{K}_{\text{tg}}$ est la matrice tangente, $\mathbf{K}_{\text{sc}}$ la matrice sécante (partie non-dissipative) et $\mathbf{K}_{\text{end}}$ la correction endommageante (partie dissipative). Pour la matrice modifiée $\mathbf{K}_{\text{t-s}}$, on remplace l'expression $\mathbf{K}_{\text{tg}}$ dans (66) par:

$$\mathbf{K}_{\text{t-s}}=\mathbf{K}_{\text{sc}}+\eta.\mathbf{K}_{\text{end}} \tag{67}$$

où η est une fonction de α_{n+1} avec des valeurs entre 0 et 1. La fonction η retenue dans la suite est la suivante:

$$\eta=\frac{1}{A^{(\alpha_{n+1}-S_0)}} \tag{68}$$

où A est une constante et S_0 le seuil sur la valeur du nombre d'alternances successives subies à partir duquel la matrice tangente est modifiée. Pour $\alpha_{n+1}=S_0$ la matrice reste tangente cohérente et pour $\alpha_{n+1}\gg S_0$, elle devient sécante. Des valeurs jugées satisfaisantes pour ces paramètres sont $A=1{,}5$ et $S_0=3$ (valeurs par défaut). Le choix sur l'évolution de la valeur de α_{n+1} est primordial pour la performance de l'algorithme. On choisit une augmentation de α_{n+1} de $G=1{,}0$, $\alpha_{n+1}\rightarrow\alpha_{n+1}+G$ pour chaque nouvelle alternance entre un état élastique et un état endommageant, puis une diminution de α_{n+1} de P , $\alpha_{n+1}\rightarrow\alpha_{n+1}-P$, lorsque l'état reste endommageant deux fois de suite. L'objectif de l'utilisation de P est de permettre le retour à la matrice tangente cohérente lorsqu'un point de Gauss reste endommageant sur plusieurs itérations, puisque la matrice tangente cohérente rend la convergence quadratique, à condition qu'on soit près de la solution. La valeur du taux de diminution P par rapport au taux d'augmentation G , est cruciale pour le comportement de l'algorithme évolutif. L'idée globale consiste à augmenter α_{n+1} , lorsqu'on est loin de la solution pour avoir un opérateur plus proche du sécant que du tangent cohérent, puis une fois « proche » de la solution, de basculer en matrice tangente cohérente (qui est la meilleure au sens de Newton).Le rapport P/G (mot-clef TAUX_RETOUR – 0.05 par défaut) représente le troisième paramètre de l'algorithme, en plus de A (mot-clef AMPLITUDE) et S_0 (mot-clef SEUIL).

2.7 Méthode de Newton-Krylov

2.7.1 Principe général

La méthode de Newton-Krylov fait partie des méthodes de Newton inexactes. Elle est uniquement utilisable quand le solveur du système linéaire (59) est itératif (par opposition à un solveur direct). Cette approche ne modifie pas le choix de la matrice tangente du système comme les méthodes précédentes. Elle joue sur le critère de convergence du solveur itératif utilisé pour le système linéarisé. En adaptant au mieux le critère de

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

convergence de la méthode itérative à la convergence de Newton, on évite de faire des itérations inutiles (dans le solveur linéaire) et on gagne ainsi en temps de calcul.

2.7.2 Mise en œuvre

Le principe général des méthodes de Newton inexactes est de remplacer la condition que l'incrément de solution $\delta \mathbf{u}_i^n$ soit la solution exacte du système (59) par une condition plus faible. On demande que $\delta \mathbf{u}_i^n$ vérifie la condition :

$$\left\| \mathbf{K}_i^{n-1} . \delta \mathbf{u}_i^n - \left(\mathbf{L}_i^{\text{méca}} - \mathbf{L}_i^{\text{int},n-1} \right) \right\| \leqslant \eta_n \left\| \mathbf{L}_i^{\text{méca}} - \mathbf{L}_i^{\text{int},n-1} \right\| \quad (69)$$

où η_n est appelé le *forcing term*.

On peut montrer que la méthode proposée est convergente et que quand la suite η_n tend vers 0, la convergence est super-linéaire (voir [8] p.96). Plus cette valeur est petite, plus la solution sera proche de celle obtenue par une résolution exacte, mais moins on gagnera de temps à la résolution du système linéaire. Il faut donc trouver un bon compromis entre résoudre rapidement les systèmes linéaires et ne pas trop dégrader la convergence des itérations de Newton.
En examinant la condition (69), on constate qu'elle est identique au critère de convergence relatif des solveurs itératifs utilisés pour résoudre le système linéarisé de Newton. Pour vérifier cette condition, il s'agit donc d'utiliser le *forcing term* comme critère de convergence d'un solveur linéaire itératif.

Comme on l'a vu précédemment, il faut aussi s'assurer que la suite η_n tende vers 0 pour conserver la convergence super-linéaire de la méthode de Newton. Pour ce faire, on va asservir η_n à la décroissance du résidu de Newton par la loi d'évolution (*cf.* [8] p.105) :

$$\eta_{n+1}^{Res} = \gamma \frac{\left\| \mathbf{R}_i^n \right\|^2}{\left\| \mathbf{R}_i^{n-1} \right\|^2} \quad (70)$$

où la constante est choisie telle que $\gamma = 0.1$.

Cette simple formule n'est pas suffisante dans la pratique car il faut garantir une décroissance adéquate de η_n . Pour cela, on détermine complètement η_n par l'expression suivante :

$$\eta_{n+1} = \begin{cases} \eta_0 & n = -1 \\ max\left(min\left(0.4\,\eta_n, \eta_{n+1}^{Res}\right), \eta_{min} \right) & n \geqslant 0, \quad (1-\gamma)\eta_n^2 \leqslant 0.2 \\ min\left(0.4\,\eta_n, max\left(\eta_{n+1}^{Res}, (1-\gamma)\eta_n^2\right)\right) & n \geqslant 0, \quad (1-\gamma)\eta_n^2 > 0.2 \end{cases} \quad (71)$$

où la constante vaut $\eta_0 = 0.9$ et correspond au critère de convergence utilisé pour la première résolution linéaire.
η_{min} , quant à elle, est la valeur du critère de convergence du solveur linéaire itératif fournie par l'utilisateur (mot-clé RESI_RELA).

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

3 Conditions limites et chargements

3.1 Classification

Dans les paragraphes suivants, nous allons décrire l'ensemble des chargements possibles dans l'opérateur et leur classification. En premier lieu, il convient de distinguer trois grandes catégories de chargements :

- Les chargements de Dirichlet (conditions limites) s'appliquent sur les inconnues nodales au sens large : ce peut être un déplacement, une température ou une pression (pour la THM). Ils peuvent être appliqués par élimination (AFFE_CHAR_CINE) ou par dualisation (AFFE_CHAR_MECA). Quand les conditions limites sont traitées par élimination, leur prise en compte se fait au moment de la résolution du système linéaire, de manière interne et ne modifie donc aucunement les équations déjà décrites précdemment ;
- Les chargements de Neumann s'appliquent en général aux flux. Ce sont des conditions en efforts, en général, définis sur les mailles (sauf pour le chargement FORCE_NODALE).

Il y a une autre distinction à réaliser entre les chargements ***suiveurs*** et ceux qui ne le sont pas. Un chargement est suiveur lorsqu'il dépend de la géométrie de la structure. Typiquement, la pression s'exerce dans la direction opposée à la normale. Donc, lorsque la structure se déforme avec l'évolution de la charge, le chargement, exprimé dans un repère absolu, est transformé. Il existe également des chargements de Dirichlet suiveurs. Ce sont des conditions limites qui dépendent de la géométrie parce qu'ils sont non-linéaires. Les charges qui ne dépendent pas de la géométrie de la structure sont appelées des charges mortes ou fixes (par exemple, la pesanteur).

3.2 Chargements suiveurs

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 $\mathbf{L}_i^{\text{méca}}(\mathbf{u}_i)$ comportant des charges suiveuses se décompose donc en deux parties:

$$\mathbf{L}_i^{\text{méca}}(\mathbf{u}_i)=\mathbf{L}_i^{\text{fixe}}+\mathbf{L}_i^{\text{suiv}}(\mathbf{u}_i) \tag{72}$$

L'exposant $^{\text{fixe}}$ désigne ici les charges mortes, et $^{\text{suiv}}$ les charges suiveuses. Le système d'équations à résoudre devient alors :

$$\mathbf{L}_i^{\text{int}}=\mathbf{L}_i^{\text{fixe}}+\mathbf{L}_i^{\text{suiv}}(\mathbf{u}_i) \tag{73}$$

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 $\mathbf{L}_i^{\text{suiv}}(\mathbf{u}_i)$ par rapport aux déplacements $(\mathbf{u}_i)$.La phase de prédiction devient :

$$\left(\mathbf{K}_{i-1}-\left.\frac{\partial \mathbf{L}^{\text{suiv}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_i^{n-1}}\right).\Delta \mathbf{u}_i^0=\Delta \mathbf{L}_i^{\text{fixe}}+\Delta \mathbf{L}_i^{\text{suiv}}+\Delta \mathbf{L}_i^{\text{varc}} \tag{74}$$

Et les itérations de Newton consistent à résoudre le système :

$$\left(\mathbf{K}_i^{n-1}-\left.\frac{\partial \mathbf{L}^{\text{suiv}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_i^{n-1}}\right).\delta \mathbf{u}_i^n=\mathbf{L}^{\text{fixe}}+\mathbf{L}_i^{\text{suiv},n-1}(\mathbf{u}_i)-\mathbf{L}_i^{\text{int},n-1} \tag{75}$$

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

Les seules charges de Neumann 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] et COQUE_3D [R3.03.07];
- 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;

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

- La force centrifuge en grands déplacements, qui pour une vitesse de rotation $\boldsymbol{\omega}$ est donnée par : $\int_\Omega \rho.\boldsymbol{\omega}\wedge[\boldsymbol{\omega}\wedge\mathbf{OM}].d\Omega = \int_\Omega \rho.\boldsymbol{\omega}\wedge[\boldsymbol{\omega}\wedge(\mathbf{OM_0}+\mathbf{u})].d\Omega$. Disponible pour les modélisations 3D et AXIS_FOURIER;
- 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 pour tenir compte des relations de comportement des géomatériaux.

3.3 Conditions limites dualisées – Cas linéaire

Nous allons maintenant prendre en compte les conditions limites dualisées dans l'écriture du système linéaire à résoudre. Dans un premier temps, nous allons seulement considérer des relations ***linéaires***. On suppose l'existence d'un potentiel $J(v)$ tel que la minimisation de ce dernier permet d'exprimer l'équilibre mécanique (au sens faible) de la structure. Si $J(u)$ est ce minimum alors on a :

$$\begin{cases}\text{Trouver } u\in V \text{ tel que :}\\ J(u)=\min_{v\in V} J(v)\equiv\{L^{\text{int}}(u,v)-L^{\text{ext}}(u,v)\}\\ \text{Avec : } V=\{v\in R^n \text{ tel que } Bv=0\}\end{cases} \tag{76}$$

- $L^{\text{int}}(u,v)$ désigne la somme des forces internes discrétisées. Elle dépend de la solution recherchée via les lois de comportement, les cinématiques de grandes transformations, etc.
- $L^{\text{ext}}(u,v)$ désigne la somme des forces externes discrétisées. Elle peut dépendre de la solution comme dans le cas des forces suiveuses, du contact, etc.
- V désigne l'ensemble des solutions tests admissibles ;
- U désigne l'ensemble des solutions recherchées admissibles ;
- 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 ;

Remarque : en mécanique, on peut considérer la présence de forces dissipatives liées au caractère visqueux de la loi de comportement ou aux forces de frottement. Dans ces cas le potentiel est remplacé par l'existence d'un pseudo-potentiel de dissipation. On ne rentre pas dans les détails mais on montre qu'on peut se ramener dans le cadre d'un problème d'optimisation grâce à la transformation de Legendre-Fenchel. Par souci de simplicité, on garde le cadre général de l'équation précédente car la suite des développements n'est pas impactée par l'existence d'un pseudo-potentiel.

Sous forme discrète, la dualisation des conditions aux limites de Dirichlet $\mathbf{B}.\mathbf{u}=\mathbf{u}^d(t)$ conduit au problème suivant [R3.03.01] :

$$\begin{cases}\mathbf{L}^{\text{int}}(\mathbf{u},t)+\mathbf{B}^T.\boldsymbol{\lambda}=\mathbf{L}^{\text{ext}}(t)\\ \mathbf{B}.\mathbf{u}=\mathbf{u}^d(t)\end{cases} \tag{77}$$

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 $\mathbf{B}^{\mathbf{T}}.\lambda$ s'interprète comme l'opposé des réactions d'appui aux nœuds correspondants. En paramétrant par l'instant t_i :

$$\begin{cases}\mathbf{L}_i^{\text{int}}+\mathbf{B}^T.\boldsymbol{\lambda}_i=\mathbf{L}_i^{\text{ext}}\\ \mathbf{B}.\mathbf{u}_i=\mathbf{u}_i^d\end{cases} \tag{78}$$

Trouver l'équilibre consiste donc à annuler en $(\mathbf{u}_i,\boldsymbol{\lambda}_i,t_i)$ le vecteur résidu d'équilibre qui sera défini par :

$$\mathbf{R}_i(\mathbf{u}_i,\boldsymbol{\lambda}_i,t_i)=\begin{pmatrix}\mathbf{L}_i^{\text{int}}+\mathbf{B}^{\mathbf{T}}.\boldsymbol{\lambda}_i-\mathbf{L}_i^{\text{ext}}\\ \mathbf{B}.\mathbf{u}_i-\mathbf{u}_i^d\end{pmatrix} \tag{79}$$

Les inconnues sont calculées de façon incrémentale par l'algorithme de résolution global y compris les multiplicateurs de Lagrange des conditions limites dualisées. 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 :

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

$$\begin{cases} t_i = t_{i-1} + \Delta t_i \\ \mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i \\ \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} + \Delta \boldsymbol{\lambda}_i \end{cases} \quad (80)$$

3.3.1 Phase de prédiction d'Euler

Pour la prédiction, il faut toujours linéariser le système (77) par rapport au temps mais, dans ce cas, autour des deux inconnues (déplacements et multiplicateurs de Lagrange) $(\mathbf{u}_{i-1}, \boldsymbol{\lambda}_{i-1})$. On note d'abord que la linéarisation des réactions d'appui $\mathbf{B}^T.\boldsymbol{\lambda}_i$ est immédiate car on considère des conditions limites linéaires et donc la matrice $\mathbf{B}$ est constante (elle ne dépend pas des déplacements ou du temps). Comme $\boldsymbol{\lambda}_i = \Delta\boldsymbol{\lambda}_i^0 + \boldsymbol{\lambda}_{i-1}$, il vient immédiatement :

$$\mathbf{B}^T.\boldsymbol{\lambda}_i = \mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0 + \mathbf{B}^T.\boldsymbol{\lambda}_{i-1} \quad (81)$$

On suppose que le chargement mécanique ne dépend pas du temps (les charges suiveuses sont exclues) et que les conditions limites de Dirichlet sont aussi linéaires, donc :

$$\begin{cases} \mathbf{L}_i^{\text{méca}} = \mathbf{L}_{i-1}^{\text{méca}} + \Delta\,\mathbf{L}_i^{\text{méca}} \\ \mathbf{u}_i^d = \mathbf{u}_{i-1}^d + \Delta\,\mathbf{u}_i^d \end{cases} \quad (82)$$

On obtient pour l'équation d'équilibre :

$$\mathbf{L}_{i-1}^{\text{int}} + \left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} .\Delta\,\mathbf{u}_i^0 + \left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}} .\Delta t_i + \mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0 + \mathbf{B}^T.\boldsymbol{\lambda}_{i-1} = \mathbf{L}_{i-1}^{\text{méca}} + \Delta\,\mathbf{L}_i^{\text{méca}} \quad (83)$$

On a équilibre à l'instant t_{i-1}, c'est-à-dire :

$$\mathbf{L}_{i-1}^{\text{int}} + \mathbf{B}^T.\boldsymbol{\lambda}_{i-1} = \mathbf{L}_{i-1}^{\text{méca}} \quad (84)$$

Et il reste donc :

$$\left.\frac{\partial \mathbf{L}^{\text{int}}}{\partial \mathbf{u}}\right|_{\mathbf{u}_{i-1}} .\Delta\,\mathbf{u}_i^0 + \left.\frac{\delta \mathbf{L}^{\text{int}}}{\delta t}\right|_{t_{i-1}} .\Delta t_i + \mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0 = \Delta\,\mathbf{L}_i^{\text{méca}} \quad (85)$$

Si on regarde maintenant la deuxième équation du système (77), on obtient pour les conditions limites de Dirichlet:

$$\mathbf{B}.(\mathbf{u}_{i-1} + \Delta\,\mathbf{u}_i^0) = \mathbf{u}_{i-1}^d + \Delta\,\mathbf{u}_i^d \quad (86)$$

On a équilibre à l'instant t_{i-1}, c'est-à-dire:

$$\mathbf{B}.\mathbf{u}_{i-1} = \mathbf{u}_{i-1}^d \quad (87)$$

Il reste finalement:

$$\mathbf{B}.\Delta\,\mathbf{u}_i^0 = \Delta\,\mathbf{u}_i^d \quad (88)$$

On obtient le système d'équations permettant de calculer des valeurs prédictives $(\Delta\,\mathbf{u}_i^0, \Delta\boldsymbol{\lambda}_i^0)$:

$$\begin{cases} \mathbf{K}_{i-1}.\Delta\,\mathbf{u}_i^0 + \mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0 = \mathbf{L}_i^{\text{méca}} - \mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1} - \mathbf{B}^T.\boldsymbol{\lambda}_{i-1} + \Delta\,\mathbf{L}_i^{\text{varc}} \\ \mathbf{B}.\Delta\,\mathbf{u}_i^0 = \Delta\,\mathbf{u}_i^d \end{cases} \quad (89)$$

On remarquera alors que cette expression fait désormais intervenir les multiplicateurs de Lagrange à l'instant *précédent*, qui sont parfois inconnus (au premier incrément de charge, par exemple). Ce qui veut dire qu'avec cette nouvelle expression, on a déplacé le problème de la connaissance des forces internes à l'instant t_{i-1} vers l'ignorance des multiplicateurs de Lagrange à ce même instant ! Mais comme les conditions limites sont linéaires, le problème est simplifié. Considérons que la solution de (89) en ce qui concerne les multiplicateurs de Lagrange $\hat{\boldsymbol{\lambda}}_i$ s'écrive sous forme incrémentale:

$$\hat{\boldsymbol{\lambda}}_i = \hat{\boldsymbol{\lambda}}_{i-1} + \Delta\,\hat{\boldsymbol{\lambda}}_i \quad (90)$$

Cette solution résout la première équation du système:

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

$$\mathbf{K}_{i-1}.\Delta\mathbf{u}_i^0+\mathbf{B}^T.\Delta\hat{\boldsymbol{\lambda}}_i^0=\mathbf{L}_i^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}-\mathbf{B}^T.\hat{\boldsymbol{\lambda}}_{i-1}+\Delta\mathbf{L}_i^{\text{varc}} \quad (91)$$

L'idée est de rechercher ce $\hat{\boldsymbol{\lambda}}_i$. Comme l'opérateur $\mathbf{B}^T$ est constant, en appliquant 90, on a:

$$\mathbf{K}_{i-1}.\Delta\mathbf{u}_i^0+\mathbf{B}^T.\hat{\boldsymbol{\lambda}}_i=\mathbf{L}_i^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}+\Delta\mathbf{L}_i^{\text{varc}} \quad (92)$$

On suppose que les conditions limites sont vérifiées, donc:

$$\mathbf{B}.\Delta\mathbf{u}_i^0=\Delta\mathbf{u}_i^d \quad (93)$$

Les déplacements imposés s'écrivent aussi sous forme incrémentale:

$$\mathbf{u}_i^d=\mathbf{u}_{i-1}^d+\Delta\mathbf{u}_i^d \quad (94)$$

La matrice $\mathbf{B}$ est constante, nous avions donc à l'incrément précédent (le problème a été résolu):

$$\mathbf{B}.\mathbf{u}_{i-1}=\mathbf{u}_{i-1}^d \quad (95)$$

En utilisant 94 et 95 dans 93 :

$$\mathbf{B}.\Delta\mathbf{u}_i^0=\Delta\mathbf{u}_i^d=\mathbf{u}_i^d-\mathbf{u}_{i-1}^d=\mathbf{u}_i^d-\mathbf{B}.\mathbf{u}_{i-1} \quad (96)$$

A l'équilibre, on a donc $\hat{\boldsymbol{\lambda}}_i$ qui satisfait aussi les conditions limites que l'on ré-écrit:

$$\mathbf{B}.\Delta\mathbf{u}_i^0=\mathbf{u}_i^d-\mathbf{B}.\mathbf{u}_{i-1} \quad (97)$$

Le vecteur des multiplicateurs de Lagrange $\hat{\boldsymbol{\lambda}}_i$ peut donc être trouvé lors de la phase de prédiction en modifiant l'équation d'imposition des conditions limites par l'expression 97. Par analogie avec l'incrément des déplacements trouvés en prédiction $\Delta\mathbf{u}_i^0$ on notera $\Delta\boldsymbol{\lambda}_i^0=\hat{\boldsymbol{\lambda}}_i$:

$$\begin{cases}\mathbf{K}_{i-1}.\Delta\mathbf{u}_i^0+\mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0=\mathbf{L}_i^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}+\Delta\mathbf{L}_i^{\text{varc}}\\ \mathbf{B}.\Delta\mathbf{u}_i^0=\mathbf{u}_i^d-\mathbf{B}.\mathbf{u}_{i-1}\end{cases} \quad (98)$$

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 $\mathbf{B}.(\mathbf{u}-\mathbf{u}_{\text{didi}})=\mathbf{u}^d$ Dans ce cas, le système à résoudre dans la phase de prédiction pour le nouvel incrément de charge est :

$$\begin{cases}\mathbf{K}_{i-1}.\Delta\mathbf{u}_i^0+\mathbf{B}^T.\Delta\boldsymbol{\lambda}_i^0=\mathbf{L}_i^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}+\Delta\mathbf{L}_i^{\text{varc}}\\ \mathbf{B}.\Delta\mathbf{u}_i^0=\mathbf{u}_i^d-\mathbf{B}.\mathbf{u}_{i-1}+\mathbf{B}.\mathbf{u}_{\text{didi}}\end{cases} \quad (99)$$

3.3.2 Phase de correction de Newton

On doit linéariser le système 26 par rapport aux inconnues en $(\mathbf{u}_i^n,\boldsymbol{\lambda}_i^n)$ à t_i constant. On commence par linéariser les forces internes $\mathbf{L}_i^{\text{int},n}$:

$$\mathbf{L}_i^{\text{int},n}\approx\mathbf{L}_i^{\text{int},n-1}+\left.\frac{\partial\mathbf{L}^{\text{int}}}{\partial\mathbf{u}}\right|_{\mathbf{u}_i^{n-1}}.\delta\mathbf{u}_i^n \quad (100)$$

La linéarisation des réactions d'appui $\mathbf{B}^T.\boldsymbol{\lambda}_i^n$ est immédiate (conditions linéaires) :

$$\mathbf{B}^T.\boldsymbol{\lambda}_i^n=\mathbf{B}^T.\boldsymbol{\lambda}_i^{n-1}+\mathbf{B}^T.\delta\boldsymbol{\lambda}_i^n \quad (101)$$

On suppose que le chargement mécanique ne dépend pas du temps (les charges suiveuses sont exclues) et que les conditions limites de Dirichlet sont aussi linéaires. On obtient pour l'équation d'équilibre :

$$\mathbf{L}_i^{\text{int},n-1}+\left.\frac{\partial\mathbf{L}^{\text{int}}}{\partial\mathbf{u}}\right|_{\mathbf{u}_i^{n-1}}.\delta\mathbf{u}_i^n+\mathbf{B}^T.\boldsymbol{\lambda}_i^{n-1}+\mathbf{B}^T.\delta\boldsymbol{\lambda}_i^n=\mathbf{L}_i^{\text{méca}} \quad (102)$$

Pour les conditions limites, la linéarisation du système (77) nous donne de manière analogue à 97:

$$\mathbf{B}.\delta\mathbf{u}_i^n=\mathbf{u}_i^d-\mathbf{B}.\mathbf{u}_i^{n-1} \quad (103)$$

Le système à résoudre s'écrit finalement:

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

$$\begin{cases} \mathbf{K}_i^{n-1}.\delta\mathbf{u}_i^n+\mathbf{B}^T.\delta\boldsymbol{\lambda}_i^n=\mathbf{L}_i^{\text{méca}}-\mathbf{L}_i^{\text{int},n-1}-\mathbf{B}^T.\boldsymbol{\lambda}_i^{n-1} \\ \mathbf{B}.\delta\mathbf{u}_i^n=\mathbf{u}_i^d-\mathbf{B}.\mathbf{u}_i^{n-1} \end{cases} \quad (104)$$

3.3.3 Adaptation des critères de convergence

On explicite ici l'adaptation de RESI_GLOB_MAXI et RESI_GLOB_RELA aux cas des conditions limites dualisées. Le critère RESI_GLOB_MAXI consiste à vérifier que la norme infinie du résidu est inférieure à la valeur γ spécifiée par l'utilisateur. Il est nécessaire d'introduire la contribution des conditions limites dualisées :

$$\|\mathbf{Q}^T.\boldsymbol{\sigma}_i^n+\mathbf{B}^T.\lambda_i^n+-\mathbf{L}_i^{\text{méca}}\|_\infty\leq\gamma \quad (105)$$

•Le critère RESI_GLOB_RELA (choisi par défaut) revient à vérifier que le résidu est suffisamment petit, comme précédemment, et ceci relativement à une quantité représentative du chargement.

$$\bullet\ \frac{\|\mathbf{Q}^T.\boldsymbol{\sigma}_i^n+\mathbf{B}^T.\lambda_i^n-\mathbf{L}_i^{\text{méca}}\|_\infty}{\|\mathbf{L}_i^{\text{méca}}+\mathbf{L}_i^{\text{varc}}+\mathbf{B}^T.\lambda_i^n\|_\infty}\leq\eta \quad \bullet(106)$$

3.4 Conditions limites dualisées – Cas non-linéaire

Considérons maintenant le cas de conditions limites dualisées non-linéaires. Par exemple, LIAISON_SOLIDE en grandes transformations. On suppose l'existence d'un potentiel $J(v)$ tel que la minimisation de ce dernier permet d'exprimer l'équilibre mécanique (au sens faible) de la structure. Si $J(u)$ est ce minimum alors on a :

$$\begin{array}{l} \text{Trouver } u\in V \text{ tel que :} \\ \left\{ \begin{array}{l} J(u)=\underset{v\in V}{min}\,J(v)\equiv\{L^{\text{int}}(u,v)-L^{\text{ext}}(u,v)\} \\ \text{Avec : } V=\{v\in R^n \text{ tel que } Bv=0 \text{ et } C(v)=0\} \end{array} \right. \end{array} \quad (107)$$

- $L^{\text{int}}(u,v)$ désigne la somme des forces internes discrétisées. Elle dépend de la solution recherchée via les lois de comportement, les cinématiques de grandes transformations, etc.
- $L^{\text{ext}}(u,v)$ désigne la somme des forces externes discrétisées. Elle peut dépendre de la solution comme dans le cas des forces suiveuses, du contact, etc.
- V désigne l'ensemble des solutions tests admissibles ;
- U désigne l'ensemble des solutions recherchées admissibles ;
- 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 ;
- $C(v)$ est un opérateur *non-linéaire* de l'espace des champs de déplacements sur un espace de fonctions définies sur une partie du bord de la structure ;

Comme conditions limites non-linéaires, on considère l'exemple illustré sur la figure (4.3-1), on veut que le solide reste rigide, c'est-à-dire que la distance entre le nœud M est lié aux autres nœuds M_i reste constante. Soit :

$$C(u_{M_i}-u_M)=\sqrt{\left(x_{\vec{e}_x M_i}-x_{\vec{e}_x M}\right)^2+\left(x_{\vec{e}_y M_i}-x_{\vec{e}_y M}\right)^2+\left(x_{\vec{e}_z M_i}-x_{\vec{e}_z M}\right)^2} \quad (108)$$

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

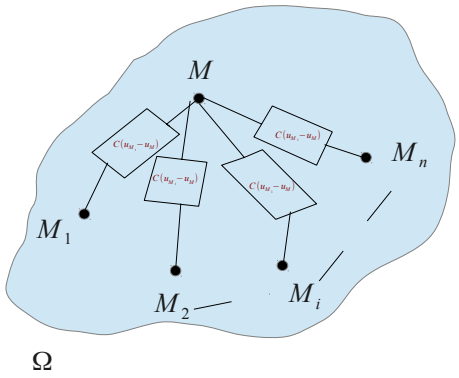


Figure 3.4-1: Exemple de relations non-linéaires

Cette relation est non-linéaire car elle dépend des déplacements des différents nœuds M_i .

3.4.1 Problème d'équilibre sous conditions d'égalités non linéaires

On suppose l'existence d'une égalité non-linéaire discrète $C(u)$ de $\mathbb{R}^n$ dans $\mathbb{R}$ dépendant de la solution $u \in R^n$ tel que l'on ait le problème suivant :

$$\begin{aligned} &\text{Trouver } u \in V \text{ tel que :} \\ &\begin{cases} J(u) = \min_{v \in V} J(v) \equiv \{ L^{\text{int}}(u,v) - L^{\text{ext}}(u,v) \} \\ \text{Avec : } V = \{ v \in R^n \text{ tel que } Bv = 0 \text{ et } C(v) = 0 \} \end{cases} \end{aligned} \tag{109}$$

On suppose que $C(.)$ est une fonction différentiable[3] de classe au moins un. Le problème précédent (107) et le problème (109) sont tels que l'on cherche le minimum dans une région de solutions admissibles $V \subset R^n$. La difficulté réside dans la construction à priori de l'ensemble des solutions admissibles, ce qui n'est pas direct dans le cas du problème (109) à cause du caractère inconnu (dépendant de la solution) de cet espace admissible. On va utiliser la technique de dualisation par les multiplicateurs de Lagrange, on écrit le Lagrangien augmenté de ce système :

$$L(v, \lambda_l^*, \lambda_{nl}^*) = J(v) + \lambda_l^* Bv + \lambda_{nl}^* C(v) \tag{110}$$

D'où le problème suivant équivalent au problème (109) :

$$\begin{aligned} &\text{Trouver les extrémums locaux } \boldsymbol{u} \in \mathbb{R}^n, \boldsymbol{\lambda_l} \in \mathbb{R} \text{ et } \boldsymbol{\lambda_{nl}} \in \mathbb{R} \\ &\text{de } L(v, \lambda_l^*, \lambda_{nl}^*) \\ &\forall (v, \lambda_l^*, \lambda_{nl}^*) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \end{aligned} \tag{111}$$

Pour cela, il suffit d'établir les conditions de stationnarité du lagrangien, soit:

$$\begin{cases} \partial_v L(v, \lambda_l^*, \lambda_{nl}^*)\big|_{u, \lambda_l, \lambda_{nl}} \delta v = 0 \\ \partial_{\lambda_l^*} L(v, \lambda_l^*, \lambda_{nl}^*)\big|_{u, \lambda_l, \lambda_{nl}} \delta \lambda_l^* = 0 \\ \partial_{\lambda_{nl}^*} L(v, \lambda_l^*, \lambda_{nl}^*)\big|_{u, \lambda_l, \lambda_{nl}} \delta \lambda_{nl}^* = 0 \end{cases} \quad \forall (\delta v, \delta \lambda_l^*, \delta \lambda_{nl}^*) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \tag{112}$$

3 Si $C(.)$ est non-différentiable alors le formalisme général ne change pas à l'exception de la définition correcte du jacobien généralisé de la fonction non différentiable (voir [R5.03.52] par exemple).

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

D'où le système non-linéaire suivant à résoudre:

$$\begin{cases} L^{\text{int}}(u,\delta v)-L^{\text{ext}}(u,\delta v)+\mathbf{B}^T\lambda_l\delta v+\nabla\mathbf{C}^T\lambda_{nl}\delta v=0 \\ \delta\lambda_l^*\mathbf{B}u=0 \\ \delta\lambda_{nl}^*\mathbf{C}(u)=0 \end{cases} \quad \forall(\delta v,\delta\lambda_l^*,\delta\lambda_{nl}^*)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R} \tag{113}$$

Qu'on écrira sous forme plus compacte :

$$S(u,\lambda)=0 \tag{114}$$

3.4.2 Linéarisation du problème

Pour résoudre ce problème on utilise également la méthode de Newton. Mais il faut linéariser le système (113). On se place au pas de temps t_i (par simplification des écritures, on omet de préciser la dépendance des quantités au pas de temps). On approxime les quantités à l'itération de Newton n, par rapport à ces mêmes quantités à l'itération $n-1$:

$$S(u^n,\lambda^n)\approx S(u^{n-1},\lambda^{n-1})+\langle\partial_u S(u^{n-1},\lambda^{n-1}),\Delta u\rangle+\langle\partial_\lambda S(u^{n-1},\lambda^{n-1}),\Delta\lambda\rangle\approx 0 \tag{115}$$

On pose :

$$\langle K(u^{n-1}),\Delta u\rangle=\langle\partial_u S(u^{n-1},\lambda^{n-1}),\Delta u\rangle+\langle\partial_\lambda S(u^{n-1},\lambda^{n-1}),\Delta\lambda\rangle \tag{116}$$

En développant tous les termes :

$$\begin{aligned}\langle K(u^{n-1}),\Delta u\rangle=&\delta v\left\{\partial_u L^{\text{int},n-1}\Delta u-\partial_u L^{\text{ext},n-1}\Delta u+\partial^2_{u^2}\mathbf{C}^{n-1}\lambda_{nl}^{n-1}\Delta u\right\}\\ &\delta v\left\{\mathbf{B}^T\Delta\lambda_l^{n-1}\right\}+\delta\lambda_l\left\{\mathbf{B}\Delta u\right\}\\ &\delta v\left\{\partial_u\mathbf{C}^{n-1}\Delta\lambda_{nl}^{n-1}\right\}+\delta\lambda_{nl}\left\{\partial_u\mathbf{C}^{n-1}\Delta u\right\}\end{aligned} \tag{117}$$

On a finalement le système suivant :

$$\langle K(u^{n-1}),\Delta u\rangle=\begin{Bmatrix}\delta v & \delta\lambda_l & \delta\lambda_{nl}\end{Bmatrix}\begin{bmatrix}K^{\text{int},n-1}+\lambda_{nl}^{n-1}\nabla\nabla\mathbf{C}^{n-1} & \mathbf{B}^T & \nabla^T\mathbf{C}^{n-1}\\ \mathbf{B} & 0 & 0\\ \nabla\mathbf{C}^{n-1} & 0 & 0\end{bmatrix}\begin{Bmatrix}\Delta u\\ \Delta\lambda_l\\ \Delta\lambda_{nl}\end{Bmatrix} \tag{118}$$

Avec :

- $K^{\text{int},n-1}=\partial_u L^{\text{int},n-1}$ la matrice tangente classique (voir 2.3.2 et 2.4.3) ;
- $\nabla\mathbf{C}^{n-1}=\partial_u\mathbf{C}^{n-1}$ la matrice jacobienne de la relation non-linéaire ;
- $\nabla\nabla\mathbf{C}^{n-1}\lambda_{nl}^{n-1}=\partial^2_{u^2}\mathbf{C}^{n-1}\lambda_{nl}^{n-1}$ la matrice Hessienne de la relation non linéaire

On retrouve le problème approché suivant, trouver les incréments $\Delta u\in\mathbb{R}^n$, $\Delta\lambda_l\in\mathbb{R}$ et $\Delta\lambda_{nl}\in\mathbb{R}$ du système en prédiction (linéarisation autour du pas de temps i). Au §2.3.5, nous avons trouvé l'expression de l'incrément de chargement *mécanique* $\Delta\mathbf{L}_i^{\text{méca}}$:

$$\Delta\mathbf{L}_i^{\text{méca}}=\mathbf{L}_i^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1} \tag{119}$$

Pour les conditions limites linéaires, le paragraphe § 3.3.1 en a aussi donné l'expression.

$$\begin{bmatrix}K_{i-1}^{\text{int}}+\lambda_{nl,i-1}\nabla\mathbf{C}_{i-1} & \mathbf{B}^T & \nabla^T\mathbf{C}_{i-1}\\ \mathbf{B} & 0 & 0\\ \nabla\mathbf{C}_{i-1} & 0 & 0\end{bmatrix}\begin{Bmatrix}\Delta u\\ \Delta\lambda_l\\ \Delta\lambda_{nl}\end{Bmatrix}=\begin{Bmatrix}\mathbf{L}_i^{\text{méca}}-\mathbf{Q}_{i-1}^T.\boldsymbol{\sigma}_{i-1}-\mathbf{B}^T.\boldsymbol{\lambda}_{i-1}+\Delta\mathbf{L}_i^{\text{varc}}\\ \mathbf{B}u_i\\ \mathbf{C}_i u_i\end{Bmatrix} \tag{120}$$

Il convient de noter la différence entre $\mathbf{B}$, matrice des relations linéaires et $\mathbf{C}_i$ matrice des relations non-linéaires. En effet, la matrice $\mathbf{B}$ est constante pendant tout le calcul, tandis que la matrice $\mathbf{C}_i$ change à chaque itération de Newton. C'est pour ça que cette dernière est indicée par i, contrairement à la matrice $\mathbf{B}$. De manière équivalente, pendant les itérations de Newton n, on doit résoudre le système suivant:

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

$$\begin{bmatrix} K^{\text{int,n-1}}+\lambda_{nl}^{n-1}\nabla\nabla\mathbf{C}^{n-1} & \mathbf{B}^T & \nabla^T\mathbf{C}^{n-1} \\ \mathbf{B} & 0 & 0 \\ \nabla\mathbf{C}^{n-1} & 0 & 0 \end{bmatrix}\begin{Bmatrix} \delta u^n \\ \delta\lambda_l^n \\ \delta\lambda_{nl}^n \end{Bmatrix}=\begin{Bmatrix} \mathbf{L}^{\text{méca},n-1}-\mathbf{L}^{\text{int},n-1}-\mathbf{B}^T.\boldsymbol{\lambda}^{n-1} \\ \mathbf{u}^d-\mathbf{B}.\mathbf{u}^{n-1} \\ \mathbf{C}^{n-1}u^{n-1} \end{Bmatrix} \quad (121)$$

Remarques :

- Comme dans le cas standard, il est possible de faire du quasi-Newton en ignorant la contribution de $\nabla\nabla\mathbf{C}$ à la matrice tangente. Mais c'est un choix du développeur car l'utilisateur n'a pas accès au choix de réactualisation de ce terme contrairement au cas général ;
- Le problème devenant non-linéaire en termes de conditions limites, il est nécessaire de basculer avec une matrice réactualisée toutes les itérations de Newton (`REAC_ITER=1`). Un message d'alarme vous prévient si vous ne le faites pas ;

3.4.3 Adaptation du critère de convergence

Il est important de bien définir le critère d'arrêt à la fin de chaque itération de Newton aussi bien après la prédiction que pendant la correction. On rappelle (voir §2.5) que dans *Code_Aster, on peut le définir de plusieurs manières.* On explicite ici l'adaptation de `RESI_GLOB_MAXI` et `RESI_GLOB_RELA` aux cas non-linéaires.

Le critère `RESI_GLOB_MAXI` consiste à vérifier que la norme infinie du résidu est inférieure à la valeur γ spécifiée par l'utilisateur. Il est nécessaire d'introduire la contribution des conditions limites non-linéaires :

$$\left\|\mathbf{Q}^T.\boldsymbol{\sigma}_i^n+\mathbf{B}^T.\lambda_i^n+\nabla\mathbf{C}_i^{n,T}.\lambda_{nl,i}^n-\mathbf{L}_i^{\text{méca}}\right\|_\infty\leq\gamma \quad (122)$$

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

$$\frac{\left\|\mathbf{Q}^T.\boldsymbol{\sigma}_i^n+\mathbf{B}^T.\lambda_i^n+\nabla\mathbf{C}_i^{n,T}.\lambda_{nl,i}^n-\mathbf{L}_i^{\text{méca}}\right\|_\infty}{\left\|\mathbf{L}_i^{\text{méca}}+\mathbf{L}_i^{\text{varc}}+\mathbf{B}^T.\lambda_i^n+\nabla\mathbf{C}_i^{n,T}.\lambda_{nl,i}^n\right\|_\infty}\leq\eta \quad (123)$$

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

4 Recherche linéaire

La recherche linéaire exposé ici concerne la recherche linéaire en l'absence de pilotage.

4.1 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 $(\delta \mathbf{u}_i^n, \delta \boldsymbol{\lambda}_i^n)$, non plus comme l'incrément des déplacements et des multiplicateurs 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 :

$$\begin{cases} \mathbf{u}_i^n = \mathbf{u}_i^{n-1} + \rho . \delta \mathbf{u}_i^n \\ \boldsymbol{\lambda}_i^n = \boldsymbol{\lambda}_i^{n-1} + \rho . \delta \boldsymbol{\lambda}_i^n \end{cases} \quad (124)$$

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

4.2 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] 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 :

$$J : V \to \mathbb{R}$$
$$\mathbf{u} \to J(\mathbf{u}) = \int_\Omega \Phi(\boldsymbol{\varepsilon}(\mathbf{u})) . d\Omega - \int_\Omega \mathbf{f} . \mathbf{u} . d\Omega - \int_\Gamma \mathbf{t} . \mathbf{u} . d\Gamma \quad (125)$$

où la densité d'énergie libre Φ permet de relier le tenseur des contraintes $\boldsymbol{\sigma}$ au tenseur des déformations linéarisées $\boldsymbol{\varepsilon}$ par la relation $\boldsymbol{\varepsilon} = \frac{\partial \Phi}{\partial \boldsymbol{\sigma}}$ dans le cas de l'hyperélasticité (on généralise cette situation aux autres non linéarités dans la suite du document). La fonctionnelle J étant convexe, trouver le minimum de J est équivalent à annuler son gradient, soit :

$$\nabla J(\mathbf{u}) . \mathbf{v} = 0 \quad \forall \mathbf{v} \in V \quad (126)$$

Ce qui est exactement le Principe des Travaux Virtuels puisque :

$$\nabla J(\mathbf{u}) . \mathbf{v} = \int_\Omega \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) . d\Omega - \int_\Omega \mathbf{f} . \mathbf{v} . d\Omega - \int_\Gamma \mathbf{t} . \mathbf{v} . d\Gamma \quad (127)$$

Ainsi, résoudre les équations issues du Principe des Travaux Virtuels (base du problème formulé dans le [§ 1.3]) 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 $\mathbf{t}$).

4.3 Méthode de minimisation

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

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

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

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 2-4.3-1

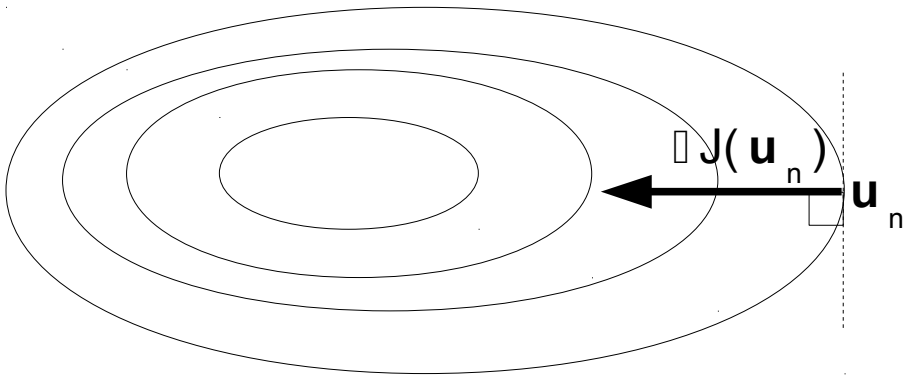


Figure 2-4.3-1

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. 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 $\left(\frac{\alpha}{a},\frac{\beta}{b}\right)$:

$$J(x,y)=\frac{1}{2}.ax^2+\frac{1}{2}.by^2-\alpha x-\beta y \qquad (128)$$

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

$$x^{n+1}=x^{n}-ax^{n}+\alpha \qquad (129)$$

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

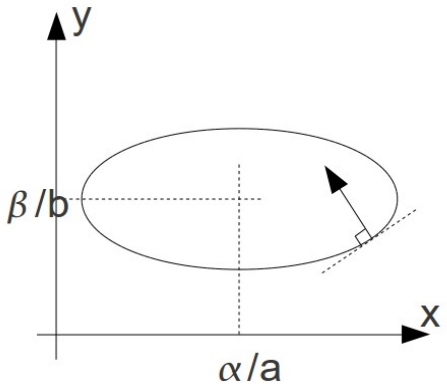


Figure 2 - 4.3-2

Par contre, si l'on effectue un changement de variables pour que les isovaleurs de J deviennent des cercles :

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

$$\begin{cases} \bar{x}=\sqrt{a}.x \\ \bar{y}=\sqrt{b}.y \\ \bar{J}(\bar{x},\bar{y})=\frac{1}{2}.(\bar{x}^2+\bar{y}^2)-\frac{\alpha}{\sqrt{a}}.\bar{x}-\frac{\beta}{\sqrt{b}}.\bar{y} \end{cases} \quad (130)$$

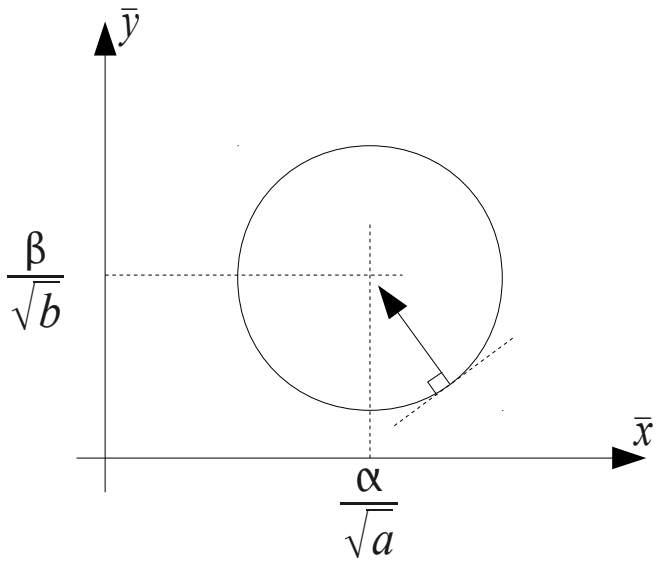


Figure 2-4.3-3

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

$$\bar{x}^{n+1}=\bar{x}^n-\left(\bar{x}^n-\frac{\alpha}{\sqrt{a}}\right)=\frac{\alpha}{\sqrt{a}}\Rightarrow x^{n+1}=\frac{\alpha}{a} \quad (131)$$

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.

4.4 Application à la minimisation de l'énergie

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

$$J(\mathbf{u})=\frac{1}{2}.\mathbf{u}^T.\mathbf{K}.\mathbf{u}-\mathbf{u}^T.\mathbf{L} \quad (132)$$

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 :

$$\bar{J}(\bar{\mathbf{u}})=\frac{1}{2}.\bar{\mathbf{u}}^T.\bar{\mathbf{u}}-\bar{\mathbf{u}}^T.\left(\sqrt{\mathbf{K}}\right)^{-T}.\mathbf{L} \quad (133)$$

En utilisant une décomposition par la diagonale:

$$\mathbf{K}=\mathbf{P}.\mathbf{D}.\mathbf{P}^{-1} \quad (134)$$

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

Avec $\mathbf{D}$ diagonale, donc:

$$\sqrt{\mathbf{K}}=\mathbf{P}.\sqrt{\mathbf{D}}.\mathbf{P}^{-1} \tag{135}$$

Ce qui conduit à :

$$\sqrt{\mathbf{K}^T}.\sqrt{\mathbf{K}}=\mathbf{P}.\sqrt{\mathbf{D}^T}.\sqrt{\mathbf{D}}.\mathbf{P}^{-1}=\mathbf{K} \tag{136}$$

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

$$\bar{\mathbf{u}}^{n+1}=\bar{\mathbf{u}}^{n}-\left(\bar{\mathbf{u}}^{n}-\sqrt{\mathbf{K}^{-1}}.\mathbf{L}\right) \tag{137}$$

Soit, en revenant aux variables initiales :

$$\mathbf{u}^{n+1}=\mathbf{u}^{n}-\mathbf{K}^{-1}.\left(\mathbf{K}.\mathbf{u}^{n}-\mathbf{L}\right) \tag{138}$$

Ou encore :

$$\mathbf{K}.\left(\mathbf{u}^{n+1}-\mathbf{u}^{n}\right)=\mathbf{L}-\mathbf{K}.\mathbf{u}^{n} \tag{139}$$

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 internes, appelé aussi résidu d'équilibre. 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.

4.5 Détermination du pas d'avancement

Revenons au problème réel, celui de la résolution de $\mathbf{L}_i^{\text{int}}\left(\mathbf{u}_i\right)=\mathbf{L}_i^{\text{ext}}$. Ce problème peut être interprété comme la minimisation de la fonctionnelle suivante :

$$J=W\left(\mathbf{u}_i\right)-\mathbf{u}_i^T.\mathbf{L}_i^{\text{ext}} \tag{140}$$

où $W\left(\mathbf{u_i}\right)$ correspond à la discrétisation, sur la base des fonctions de forme, de l'énergie interne de la structure:

$$W=\int_\Omega \Phi\left(\boldsymbol{\varepsilon}\left(\mathbf{u}\right)\right).d\,\Omega \tag{141}$$

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

$$\underset{\rho\in\mathbb{R}}{\text{Min}}\left\{W\left(\mathbf{u}_i^{n-1}+\rho.\delta\,\mathbf{u}_i^{n}\right)-\mathbf{L}_i^{\text{ext}}\left(\mathbf{u}_i^{n-1}+\rho.\delta\,\mathbf{u}_i^{n}\right)\right\} \tag{142}$$

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

$$f'\left(\rho\right)=\left[\delta\,\mathbf{u}_i^{n}\right]^T.\left[\mathbf{Q}^T.\boldsymbol{\sigma}\left(\mathbf{u}_i^{n-1}+\rho.\delta\,\mathbf{u}_i^{n}\right)-\mathbf{L}_i^{\text{ext}}\right]=0 \tag{143}$$

$f'\left(\rho\right)$ est la projection du résidu sur la direction de recherche. Avec les notations du [§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 :

$$f'\left(\rho\right)=\left[\delta\,\mathbf{u}_i^{n}\right]^T.\left[\mathbf{Q}^T.\boldsymbol{\sigma}\left(\mathbf{u}_i^{n-1}+\rho.\delta\,\mathbf{u}_i^{n}\right)+\mathbf{B}^T.\left(\boldsymbol{\lambda}_i^{n-1}+\rho.\delta\,\boldsymbol{\lambda}_i^{n}\right)-\mathbf{L}_i^{\text{ext}}\right]=0 \tag{144}$$

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

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

$$\begin{cases} \mathbf{u}_i^n = \mathbf{u}_i^{n-1} + \rho . \delta \mathbf{u}_i^n \\ \boldsymbol{\lambda}_i^n = \boldsymbol{\lambda}_i^{n-1} + \rho . \delta \boldsymbol{\lambda}_i^n \end{cases} \quad (145)$$

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 $f(\rho) \leq \tau . f(0)$, où τ vaut par défaut 0.1.

La recherche linéaire est en quelque sorte une "assurance" permettant de se prémunir contre des divergences graves de la méthode de Newton. Lorsque la direction de recherche 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 de trop s'éloigner de la solution. Il n'est pas nécessaire de faire beaucoup d'itérations dans la méthode de sécante (deux ou trois 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 la valeur de ρ vraiment optimale.

4.6 Calcul du coefficient de recherche linéaire

Il existe deux méthodes pour calculer le ρ optimal dans la recherche linéaire.

4.6.1 Méthode sécante (METHODE='CORDE')

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 :

$$\rho^{p+1} = \rho^p - \frac{\rho^p - \rho^{p-1}}{g^p - g^{p-1}} \cdot g^p = \frac{\rho^{p-1} \cdot g^p - \rho^p \cdot g^{p-1}}{g^p - g^{p-1}} \quad (146)$$

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. Elle se représente schématiquement sur la Figure 2-4.6.1-1 .

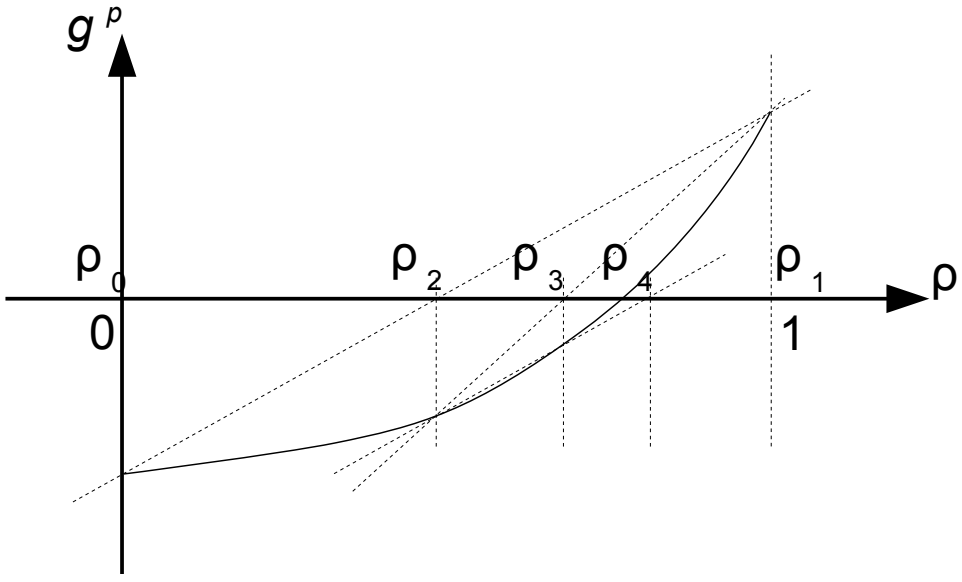


Figure 2-4.6.1-1

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

4.6.2 Méthode mixte (METHODE='MIXTE')

Cette méthode mélange plusieurs techniques de résolution pour être plus robuste. Elle consiste essentiellement en l'application d'une méthode de la sécante (voir paragraphe précédente) entre deux bornes pré-déterminées. Il s'agit en fait de l'application de la méthode de sécante avec des bornes variables. Voici l'algorithme utilisé :

1. On suppose que $f'(0)>0$. Si ce n'est pas le cas, on change le sens de la direction de descente (en examinant des ρ négatifs, ce qui revient à définir f' comme étant égale à $-f'$
2. On cherche un $\rho_{\max}$ positif tel que $f'(\rho_{\max})<0$. La méthode est simplement itérative en faisant $\rho_{n+1}=3.\rho_n$ avec $\rho_0=1$ (étape de bracketage ou encadrement)
3. On a ainsi les deux nouvelles bornes entre lesquelles la fonction change de signe. Si on suppose que la fonction f' est continue, il existe donc une solution entre ces bornes.
4. On applique la méthode de la sécante sur cet intervalle : on part de $\rho^0=0$ et $\rho^1=\rho_{max}$.

4.6.3 Cas particulier : la méthode de Newton-Krylov

Il a été précisé plus haut que la recherche linéaire est réalisée simultanément sur les inconnues $\mathbf{u}$ et $\boldsymbol{\lambda}$ comme le montre la formule (Error: Reference source not found) d'actualisation des variables. Or la fonctionnelle à minimiser ne présente pas de minimum en fonction des inconnues $(\mathbf{u},\boldsymbol{\lambda})$, il s'agit en effet d'un Lagrangien qui présente un point selle, c'est-à-dire un minimum en $\mathbf{u}$ et un maximum en $\boldsymbol{\lambda}$ (voir [R3.03.01]). Cette manière de faire n'est donc pas licite dans le cas général.
Cela dit, on peut montrer que dans le cas où le système en prédiction est résolu « exactement » (tout du moins à une précision numériquement satisfaisante), cette approche est licite. C'est généralement le cas dans l'usage habituel de *Code_Aster*.
Ce n'est par contre pas le cas dans le cadre de l'usage de la méthode de Newton-Krylov, où les systèmes linéaires sont justement résolus de manière volontairement inexacte. Dans cette situation, pour contourner le problème, seules les inconnues $\mathbf{u}$ sont affectées par la recherche linéaire et la formule de mise à jour des variables devient donc :

$$\begin{cases}\mathbf{u}_i^n=\mathbf{u}_i^{n-1}+\rho.\delta\mathbf{u}_i^n\\ \boldsymbol{\lambda}_i^n=\boldsymbol{\lambda}_i^{n-1}+\delta\boldsymbol{\lambda}_i^n\end{cases} \quad (147)$$

Dans la mesure où la fonctionnelle à minimiser dispose bien d'un minimum en $\mathbf{u}$, la procédure de recherche linéaire est licite.

5 Pilotage

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

6 Bibliographie

1) A. Ibrahimbegovic **–** « Nonlinear Solid Mechanics: Theoretical Formulations and Finite Element Solution Methods » – Springer – 2009

2) M.A. Crisfield – « Non-Linear Finite Element Analysis of Solids and Structures : Essentials » – Wiley Professional Software – 1996

3) M.A. Crisfield – « Non-Linear Finite Element Analysis of Solids and Structures : Advanced Topics » – Wiley Professional Software – 1997

4) J.C. Simo, T.J.R. Hughes – « Computational inelasticity » – Springer – 2000

5) « Éléments d'analyse et de résolution numérique des relations de l'élasto-plasticité » – EDF – Bulletin de la Direction des Études et Recherches – Série C – N° 3 1986 – p. 57 - 89.

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

6) J.F. Maître – « Analyse numérique » – cours polycopié de l'ENTPE.

7) 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).

8) C.T. Kelley - « Iterative Methods for Solving Linear and Nonlinear Equations », Vol. 16 in Frontiers in Applied Mathematics,Siam, Philadelphia, 1995.

7 Historique des versions du document

Version Aster	Auteur(s) ou contributeur(s), organisme	Description des modifications
5	N. Tardieu, I. Vautier, E. Lorentz EDF R&D MMN	
7.4	P.Badel, J.Laverne,N. Tardieu EDF R&D AMA	
8.5	M.Abbas EDF R&D AMA	Mise à jour des notations, ajout des variables de commande.
9.2	M.Abbas EDF R&D AMA	Mise à jour du § 2.2.
9.4	M.Abbas, D.Markovic EDF R&D AMA	Ajout de la matrice tangente-sécante et de la recherche linéaire mixte.
10.1	M.Abbas EDF R&D AMA	Mise à jour du paragraphe sur les grandes déformations.
11.1	N. Tardieu EDF R&D AMA	Ajout de la méthode de Newton-Krylov.
11.3	M. David EDF R&D AMA	Précisions concernant le critère RESI_REFE_RELA

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