Éléments de coques volumiques en non linéaire géométrique

Résumé :

Nous présentons dans ce document la formulation théorique et l'implantation numérique d'un élément fini de coque volumique pour des analyses en non linéaire géométrique. Cette approche doit permettre de prendre en compte de grands déplacements et de grandes rotations de structures minces, dont le rapport épaisseur sur longueur caractéristique est inférieur à $1/10$. On veillera à ce que ces rotations restent inférieures à 2π.

Cette formulation est basée sur une approche de milieu continu 3D, dégénérée par l'introduction de la cinématique de coque en contraintes planes dans la forme faible de l'équilibre. La mesure des déformations que nous retenons est celle de Green-Lagrange, énergétiquement conjuguée aux contraintes de Piola-Kirchhoff de deuxième espèce. La formulation de l'équilibre est donc lagrangienne totale.

Le problème entièrement non linéaire géométrique est examiné en premier. Le cas du flambement linéaire est traité comme un cas limite de la première approche.

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

Table des matières

1Introduction....4
2Formulation....5
2.1Géométrie des éléments de coque volumique....5
2.2Cinématique des coques volumiques....6
2.3Loi de comportement....9
2.3.1Prise en compte du cisaillement transverse....10
3Principe du travail virtuel....10
3.1Travail virtuel interne....10
3.1.1Forme incrémentale du travail virtuel interne....10
3.1.2Passage du repère global au repère local....11
3.1.3Relation déformation-déplacement....13
3.1.4Calcul des contraintes de Cauchy....16
4Discrétisation numérique de la formulation variationnelle issue du principe du travail virtuel....19
4.1Éléments finis....19
4.2Discrétisation du champ de déplacement....20
4.3Discrétisation du gradient du déplacement....22
4.3.1Gradient du déplacement total....22
4.3.2Gradient du déplacement virtuel....24
4.3.3Gradient du déplacement itératif....26
4.3.4Gradient de la variation itérative du déplacement virtuel....27
4.4Discrétisation de la formulation variationnelle issue du principe du travail virtuel....28
4.4.1Vecteur des forces internes....28
4.4.2Matrice tangente de rigidité....29
4.4.3Schémas d'intégration....30
5Rigidité autour de la transformée de la normale....34
5.1Singularité de la matrice tangente de rigidité....34
5.2Principe du travail virtuel pour les termes associés à la rotation autour de la normale....34
5.3Remarque....36
5.4Cas limite de l'analyse géométriquement linéaire....36
5.5Détermination du coefficient k....37
6Flambement linéaire....39
7Implantation des éléments de coque dans Code_Aster....42
7.1Description....42
7.2Utilisation....42
7.3Calcul en " élasticité " non linéaire géométrique....42
7.4Implantation....42
7.4.1Modification du TE0414....43

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

7.4.2Ajout d'une routine VDGNLR....43
7.5Calcul en flambement linéaire....43
8Conclusion....44
9Bibliographie....45
10Description des versions du document....45

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

1 Introduction

Les grandes transformations de coque sont caractérisées par de grands déplacements de la surface moyenne et de grandes rotations des fibres initialement normales à cette surface. La transformation est donc exactement représentée, du moins dans le problème continu. La dérivation des objets éléments finis associés au système linéarisé d'équations issu du principe du travail virtuel est effectuée sans aucune hypothèse simplificatrice sur les déplacements ou les rotations. De plus, un nouveau schéma d'intégration numérique sélective est présenté afin de résoudre le problème de blocage en membrane et en cisaillement transverse.

Les degrés de liberté de rotation retenus sont les composantes du vecteur de rotation itérative spatiale. Entre deux itérations, c'est le vecteur de la rotation infinitésimale superposée à la configuration déformée. Ce choix conduit à une matrice tangente de rigidité qui n'est pas symétrique. Ceci est dû au caractère non vectoriel des grandes rotations qui appartiennent en réalité à la variété différentielle $SO(3)$. Les rotations doivent rester inférieures à 2π du fait du choix de mise à jour des grandes rotations implanté dans *Code_Aster*, pour lequel il y a non bijection entre le vecteur de rotation totale et la matrice orthogonale de rotation.

Une différence importante par rapport à l'analyse linéaire est à signaler. Les objets éléments finis sont directement construits dans le repère global ; les déplacements et les rotations nodaux sont mesurés dans le repère global.

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

2 Formulation

Dans ce chapitre, nous présentons les différentes équations gouvernant le problème de déformation de la coque dans le cadre d'une théorie de grandes transformations.

2.1 Géométrie des éléments de coque volumique

La coque volumique est représentée par le volume Ω (ensemble des points $Q\left(\xi_3 \neq 0\right)$) construit autour de la surface moyenne ω (ensemble des points $P\left(\xi_3 = 0\right)$). En tout point Q de Ω , on construit un repère orthonormé local $\left[t_1\left(\xi_1,\xi_2,\xi_3\right):t_2\left(\xi_1,\xi_2,\xi_3\right):n\left(\xi_1,\xi_2\right)\right]$. Le vecteur $n\left(\xi_1,\xi_2\right)$ représente la normale à la surface ω .

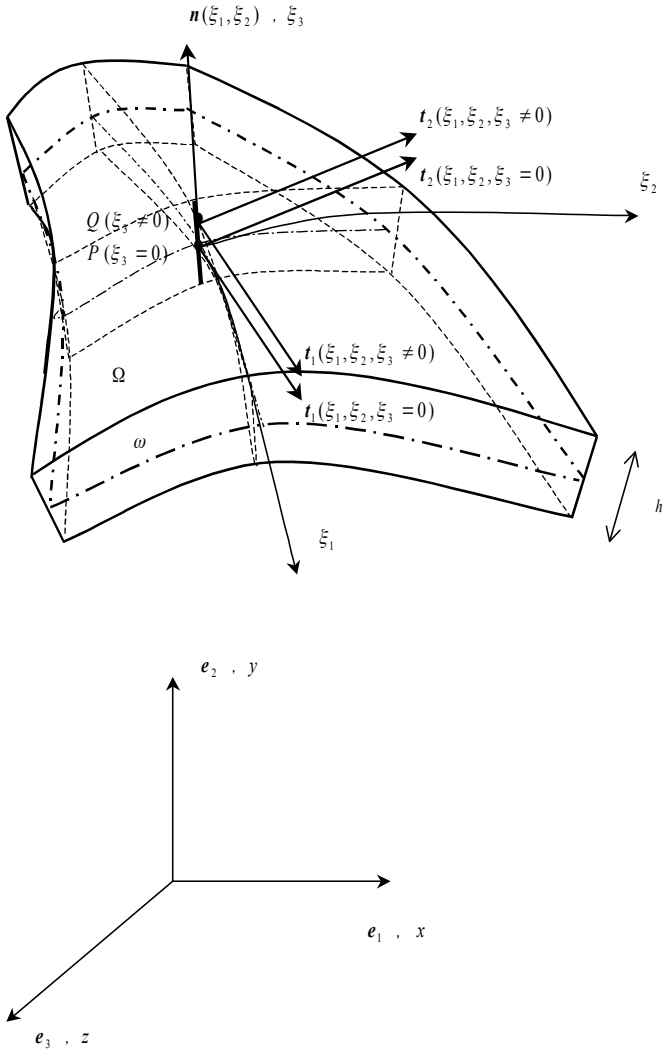


Figure 2.1-a : Coque volumique. Repères locaux sur la configuration de référence

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

Dans la configuration initiale, la position d'un point quelconque Q normal à la surface moyenne peut être exprimée, en fonction de la position du centre gravité P de la fibre normale, de la manière suivante :

$$x_Q(\xi_1,\xi_2,\xi_3)=\xi_P(\xi_1,\xi_2)+\xi_3\frac{h}{2}n(\xi_1,\xi_2)$$

2.2 Cinématique des coques volumiques

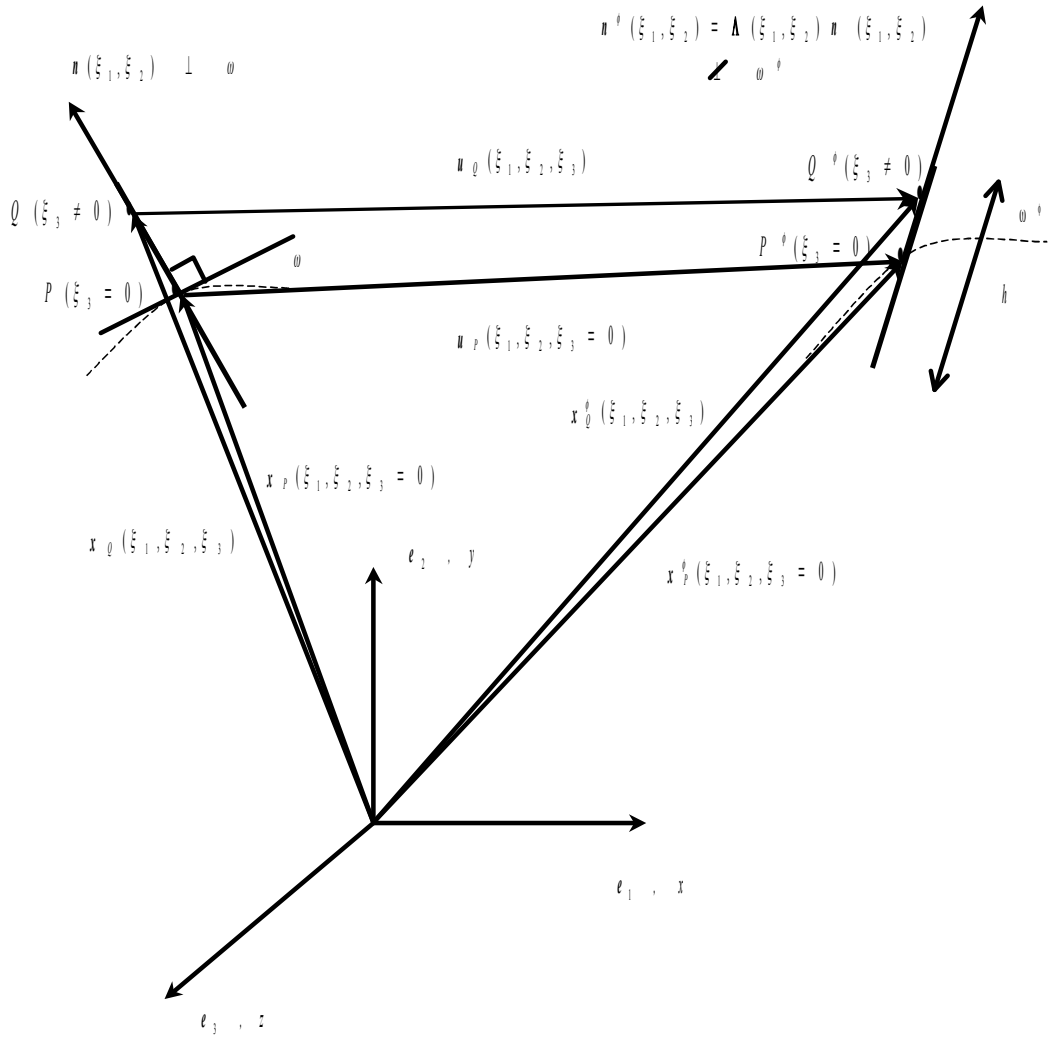


Figure 2.2-a : Coque volumique.
Grandes transformations d'une fibre initialement normale à la surface moyenne

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

Dans la configuration déformée, la position du point Q peut aussi être exprimée en fonction de la position du point P :

$$x_Q^{\varphi}(\xi_1,\xi_2,\xi_3)=x_P^{\varphi}(\xi_1,\xi_2)+\xi_3\frac{h}{2}\boldsymbol{n}^{\boldsymbol{\varphi}}(\xi_1,\xi_2)$$

où $\boldsymbol{n}^{\varphi}$ est le vecteur unitaire obtenu par grande rotation de la normale $\boldsymbol{n}$.

Le vecteur $\boldsymbol{n}^{\varphi}$ n'est pas nécessairement normal à la surface moyenne déformée, du fait de la déformation de cisaillement transverse. Il est relié au vecteur normal initial par la relation :

$$\boldsymbol{n}^{\boldsymbol{\varphi}}=\Lambda(\xi_1,\xi_2)\boldsymbol{n}$$

Λ est l'opérateur orthogonal de la grande rotation autour du vecteur Θ , d'angle θ , subie par la fibre qui était initialement normale à la surface moyenne dont l'expression est donnée par :

$$\Lambda=\exp[\Theta\times]=\cos\theta[I]+\frac{\sin\theta}{\theta}[\Theta\times]+\frac{1-\cos\theta}{\theta^2}[\Theta\otimes\Theta]$$

où $[\Theta\times]$ est l'opérateur antisymétrique du vecteur de rotation totale Θ dont l'expression matricielle est :

$$[\Theta\times]=\begin{bmatrix}0 & -\Theta_z & \Theta_y\\ \Theta_z & 0 & -\Theta_x\\ -\Theta_y & \Theta_x & 0\end{bmatrix}$$

et $[\Theta\otimes\Theta]$ est l'opérateur symétrique donné par $[\Theta\otimes\Theta]=\Theta\Theta^T$.

Plus de détails sur les grandes rotations et leur traitement numérique peuvent être trouvés dans [bib1] ou [R5.03.40]. On peut aussi écrire :

$$t_1^{\varphi}=\Lambda(\xi_1,\xi_2)t_1$$

$$t_2^{\varphi}=\Lambda(\xi_1,\xi_2)t_2$$

On peut exprimer la variation virtuelle de l'opérateur de grande rotation sous la forme :

$$\delta\Lambda=[\delta w\times]\Lambda$$

où $[\delta w\times]$ est l'opérateur antisymétrique du vecteur de rotation virtuelle spatiale δw qui est aussi la partie rotation des fonctions tests :

$$[\delta\mathbf{w}\times]\mathbf{b}=\delta\mathbf{w}\wedge\mathbf{b}\;\forall\, b\in\mathbb{R}^3$$

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

Son expression matricielle est :

$$[\delta w\times]=\begin{bmatrix}0 & -\delta w_z & \delta w_y\\ \delta w_z & 0 & -\delta w_x\\ -\delta w_y & \delta w_x & 0\end{bmatrix}$$

On peut aussi exprimer la variation itérative de l'opérateur de grande rotation sous la forme :

$$\Delta\Lambda=[\Delta w\times]\Lambda$$

où Δw est le vecteur de rotation itérative spatiale, qui est aussi la partie rotation de la solution du système d'équations linéarisées.

Ce vecteur peut être relié au vecteur de rotation itérative totale. On a ainsi les relations :

$$\Delta w=T(\Theta)\Delta\Theta \quad \text{et} \quad \delta w=T(\Theta)\delta\Theta$$

où $T(\Theta)$ est l'opérateur différentiel de rotation, dont l'expression en fonction du vecteur de rotation totale est donnée par :

$$T(\Theta)=\frac{\sin\theta}{\theta}[I]-\frac{1-\cos\theta}{\theta^2}[\Theta\times]+\frac{\theta-\sin\theta}{\theta^3}[\Theta\otimes\Theta]$$

Cette matrice possède les mêmes valeurs et vecteurs propres que la matrice Λ et vérifie la relation :

$$T(\Theta)=\Lambda(\Theta)T^T(\Theta)$$

Par ailleurs, la variation itérative de la matrice de rotation virtuelle peut être mise sous la forme :

$$\Delta\delta\Lambda=[\delta w\times][\Delta w\times]\Lambda$$

Le déplacement total du point Q sur la fibre peut être relié au déplacement du centre de gravité P :

$$u_Q(\xi_1,\xi_2,\xi_3)=u_P(\xi_1,\xi_2)+\xi_3\frac{h}{2}\left(n^{\varphi}(\xi_1,\xi_2)-n(\xi_1,\xi_2)\right)$$

Afin d'aboutir à un système d'équations linéarisées, obtenu à partir de la forme faible de l'équilibre, nous avons besoin de calculer différentes variations différentielles de ce déplacement total. Le déplacement virtuel a pour expression :

$$\delta u_Q(\xi_1,\xi_2,\xi_3)=\delta u_P(\xi_1,\xi_2)+\xi_3\frac{h}{2}\delta w(\xi_1,\xi_2)\wedge n^{\varphi}(\xi_1,\xi_2);\ \delta n=0$$

Le déplacement itératif a pour expression :

$$\Delta u_Q(\xi_1,\xi_2,x_3)=\Delta u_P(\xi_1,\xi_2)+x_3\frac{h}{2}\Delta w(\xi_1,\xi_2)\wedge n^{\varphi}(\xi_1,\xi_2);\ \Delta n=0$$

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

La variation itérative du déplacement virtuel a pour expression :

$$\Delta\delta u_Q(\xi_1,\xi_2,\xi_3)=\xi_3\frac{h}{2}\delta w(\xi_1,\xi_2)\wedge\left(\Delta w(\xi_1,\xi_2)\wedge n^{\varphi}(\xi_1,\xi_2)\right)$$

Remarque : La formulation proposée reste limitée à des rotations inférieures à 2π . Cette limite est due au choix particulier de mise à jour des grandes rotations implantées dans *Code_Aster*. Ceci est dû à la non bijection entre le vecteur de rotation totale et la matrice orthogonale de rotation.

2.3 Loi de comportement

Nous considérons une loi de comportement hyper-élastique linéaire : les contraintes locales de Piola-Kirchhoff de deuxième espèce sont proportionnelles aux déformations locales de Green-Lagrange :

$$\tilde{S}=D\tilde{E}$$

Ci-après, le symbole ~ désigne les quantités exprimées dans le repère orthonormé $\left[t_1(\xi_1,\xi_2,\xi_3):t_2(\xi_1,\xi_2,\xi_3):n(\xi_1,\xi_2)\right]$.

La matrice de comportement élastique linéaire en contraintes planes s'écrit comme suit :

$$D=\begin{bmatrix}\frac{E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 & 0 & 0\\ & \frac{E}{1-\nu^2} & 0 & 0 & 0\\ & & \frac{E}{2(1+\nu)} & 0 & 0\\ & sym & & \frac{Ek}{2(1+\nu)} & 0\\ & & & & \frac{Ek}{2(1+\nu)}\end{bmatrix}$$

E étant le module d'Young, ν le coefficient de Poisson et k le coefficient de correction de cisaillement transverse.

Dans le repère local, l'état de contrainte Piola-Kirchhoff de deuxième espèce est plan $\left(\tilde{S}_{nn}=0\right)$ et peut être caractérisé par un vecteur à 5 composantes :

$$\tilde{S}=\begin{pmatrix}\tilde{S}_{t_1t_1}\\ \tilde{S}_{t_2t_2}\\ \tilde{S}_{t_1t_2}\\ \tilde{S}_{t_1n}\\ \tilde{S}_{t_2n}\end{pmatrix}$$

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

Le vecteur des déformations de Green-Lagrange s'exprime lui aussi dans le repère local par un vecteur à 5 composantes :

$$\tilde{E}=\begin{pmatrix}\tilde{E}_{t_1t_1}\\ \tilde{E}_{t_2t_2}\\ \tilde{\gamma}_{t_1t_2}\\ \\ \tilde{\gamma}_{t_1n}\\ \tilde{\gamma}_{t_2n}\end{pmatrix}$$

Ici, nous avons ignoré le terme $\tilde{E}_{nn}$ qui est normal à la surface moyenne et qui n'est pas forcément nul. Ceci est une conséquence de l'hypothèse des contraintes planes.

2.3.1 Prise en compte du cisaillement transverse

La correction de la contrainte de cisaillement transverse est effectuée par extension des équivalences énergétiques déterminées dans le cas des petites déformations et des petits déplacements [R3.07.03].

3 Principe du travail virtuel

Le principe des travaux virtuels est la formulation faible de l'équilibre statique des forces internes et des forces externes :

$$\delta\pi_{\text{int}}-\delta\pi_{ext}=0$$

La non-linéarité des équations d'équilibre nous conduit à résoudre le système ci-dessus de façon itérative par une méthode de Newton. Nous procédons ainsi à la linéarisation exacte du principe des travaux virtuels à chaque itération, qui conduit à l'égalité :

$$\Delta\delta\pi_{\text{int}}-\Delta\delta\pi_{ext}=\delta\pi_{ext}-\delta\pi_{\text{int}}$$

3.1 Travail virtuel interne

Le travail virtuel des forces internes peut être écrit sur la configuration initiale sous la forme :

$$\delta\pi_{\text{int}}=\int_{\Omega}\left(\delta\tilde{\mathbf{E}}.\tilde{\mathbf{S}}\right)d\Omega$$

où $\tilde{E}$ et $\tilde{S}$ sont les vecteurs de déformation de Green-Lagrange et de contrainte Piola-Kirchhoff de deuxième espèce respectivement, exprimés dans le repère local. En effet, comme l'état de contrainte est plan pour le Piola-Kirchhoff de deuxième espèce, nous utilisons la formulation du principe du travail virtuel dans le repère local. Cependant, pour limiter les passages du repère local au repère global et vice-versa, les vecteurs de déformations et de contraintes locales ne sont pas calculés explicitement dans le repère local mais ils sont obtenus par la rotation de leur représentation dans le repère global.

3.1.1 Forme incrémentale du travail virtuel interne

La variation itérative du travail du travail virtuel interne s'écrit :

$$\Delta\delta\pi_{\text{int}}=\int_{\Omega}\left(\delta\tilde{E}.\Delta\tilde{S}+\Delta\delta\tilde{E}.\tilde{S}\right)d\Omega$$

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

Dans cette égalité, la variation itérative du vecteur de contraintes locales de Piola-Kirchhoff de deuxième espèce se calcule par la forme discrète itérative de la relation de comportement :

$$\Delta \tilde{S} = D \Delta \tilde{E}$$

3.1.2 Passage du repère global au repère local

Sous forme tensorielle on passe du tenseur des contraintes globales au tenseur des contraintes locales 3×3 (voir [bib4] p. 111 pour les contraintes de Cauchy, les mêmes relations s'appliquant aux contraintes de Piola-Kirchhoff de deuxième espèce) en utilisant :

$$[\tilde{S}] = P[S]P^T$$

et du tenseur des contraintes locales au tenseur des contraintes globales par l'inversion de la relation précédente :

$$[S] = P^T[\tilde{S}]P$$

Dans les deux expressions précédentes, la matrice de passage du repère local au repère global est une matrice orthogonale $P^{-1} = P^T$, et son expression explicite en fonction des vecteurs unitaires du repère orthonormé local est :

$$\mathbf{P}(\xi_1, \xi_2, \xi_3) = \begin{bmatrix} \mathbf{t_1^T}(\xi_1, \xi_2, \xi_3) \\ \mathbf{t_2^T}(\xi_1, \xi_2, \xi_3) \\ \mathbf{n^T}(\xi_1, \xi_2, \xi_3) \end{bmatrix}$$

Dans le cadre de la notation conventionnelle, on pourra noter :

$$\begin{aligned} t_1(\xi_1, \xi_2, \xi_3) &= \Lambda_0 e_1 \\ t_2(\xi_1, \xi_2, \xi_3) &= \Lambda_0 e_2 \\ t_3(\xi_1, \xi_2, \xi_3) &= n(\xi_1, \xi_2) = \Lambda_0 e_3 \end{aligned}$$

avec la matrice orthogonale de passage (rotation initiale) :

$$\Lambda_0(\xi_1, \xi_2, \xi_3) = \left[t_1(\xi_1, \xi_2, \xi_3) : t_2(\xi_1, \xi_2, \xi_3) : t_3(\xi_1, \xi_2, \xi_3) \right]$$

On remarquera que :

$$\Lambda_0 = P^T$$

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

Les deux relations de rotation des contraintes sont aussi valables pour les tenseurs des déformations de Green-Lagrange. Néanmoins, une écriture qui relie les vecteurs de déformation locale et globale est nécessaire. Cette relation permet de passer du vecteur 6×1 des déformations globales au vecteur 6×1 des déformations locales :

$$\overset{6\times1}{\tilde{E}} = \overset{6\times6}{\bar{H}}\ \overset{6\times1}{E}$$

avec l'expression de la matrice de transformation des vecteurs 6×1 de déformation (voir [bib2] p. 258) :

$$\overset{6\times6}{\overline{H}} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & m_1 n_1 & n_1 l_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & m_2 n_2 & n_2 l_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & m_3 n_3 & n_3 l_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & l_1 m_2 + l_2 m_1 & m_1 n_2 + m_2 n_1 & n_1 l_2 + n_2 l_1 \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & l_2 m_3 + l_3 m_2 & m_2 n_3 + m_3 n_2 & n_2 l_3 + n_3 l_2 \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & l_3 m_1 + l_1 m_3 & m_3 n_1 + m_1 n_3 & n_3 l_1 + n_1 l_3 \end{bmatrix}$$

et les composantes des vecteurs unitaires du repère local :

$$\begin{array}{lll} l_1 = t_1 . e_1 & m_1 = t_1 . e_2 & n_1 = t_1 . e_3 \\ l_2 = t_2 . e_1 & m_2 = t_2 . e_2 & n_2 = t_2 . e_3 \\ l_3 = t_3 . e_1 & m_3 = t_3 . e_2 & n_3 = t_3 . e_3 \end{array}$$

Ces expressions sont générales pour les repères curvilignes. Dans le repère global cartésien $\left[e_1 : e_2 : e_3\right]$, ces composantes sont :

$$\begin{array}{lll} l_1 = t_1(1) & m_1 = t_1(2) & n_1 = t_1(3) \\ l_2 = t_2(1) & m_2 = t_2(2) & n_2 = t_2(3) \\ l_3 = t_3(1) & m_3 = t_3(2) & n_3 = t_3(3) \end{array}$$

Nous avons, en réalité, besoin d'une écriture qui relie le vecteur de déformation locale 5×1 et le vecteur de déformation globale 6×1 :

$$\overset{5\times1}{\tilde{E}} = \overset{5\times6}{H}\ \overset{6\times1}{E}$$

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

Pour cela, on oublie la troisième ligne de l'expression de $^{6\times6}_{H}$ (ligne associée à S_{nn}) :

$$
{}^{5\times6}H=\begin{bmatrix}
(t_1(1))^2 & (t_1(2))^2 & (t_1(3))^2 \\
(t_2(1))^2 & (t_2(2))^2 & (t_2(3))^2 \\
2t_1(1)t_2(1) & 2t_1(2)t_2(2) & 2t_1(3)t_2(3) \\
2t_2(1)t_3(1) & 2t_2(2)t_3(2) & 2t_2(3)t_3(3) \\
2t_3(1)t_1(1) & 2t_3(2)t_1(2) & 2t_3(3)t_1(3)
\end{bmatrix}
$$

$$
\begin{bmatrix}
t_1(1)t_1(2) & t_1(2)t_1(3) & t_1(3)t_1(1) \\
t_2(1)t_2(2) & t_2(2)t_2(3) & t_2(3)t_2(1) \\
t_1(1)t_2(2)+t_2(1)t_1(2) & t_1(2)t_2(3)+t_2(2)t_1(3) & t_1(3)t_2(1)+t_2(3)t_1(1) \\
t_2(1)t_3(2)+t_3(1)t_2(2) & t_2(2)t_3(3)+t_3(2)t_2(3) & t_2(3)t_3(1)+t_3(3)t_2(1) \\
t_3(1)t_1(2)+t_1(1)t_3(2) & t_3(2)t_1(3)+t_1(2)t_3(3) & t_3(3)t_1(1)+t_1(3)t_3(1)
\end{bmatrix}
$$

Les mêmes relations précédentes peuvent être appliquées pour le passage des vecteurs de déformation globale à la déformation locale.

3.1.3 Relation déformation-déplacement

Le tenseur 3×3 des déformations globales de Green-Lagrange est défini par (voir par exemple [bib2]) :

$$[E]=\frac{1}{2}\left(\nabla u+\nabla u^T+\nabla u^T\nabla u\right)$$

avec le tenseur du gradient des déplacements :

$$\nabla u=\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix}\langle u\,v\,w\rangle=\begin{bmatrix}\frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x}\\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y}\\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z}\end{bmatrix}$$

Le tenseur de déformation de Green-Lagrange peut aussi s'écrire :

$$[E]=\frac{1}{2}\left(F^TF-I\right)$$

avec F le tenseur gradient des déformations 3×3 qui n'est pas symétrique :

$$F=\nabla x^{\varphi}=I+\nabla u$$

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

et I le tenseur identité :

$$I=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

Le vecteur 6×1 des déformations globales de Green-Lagrange est ordonné comme suit (voir [bib4] p 117) :

$$E=\begin{pmatrix}E_{xx}\\E_{yy}\\E_{zz}\\\gamma_{xy}\\\gamma_{xz}\\\gamma_{yz}\end{pmatrix}=\begin{pmatrix}u_{,x}\\v_{,y}\\w_{,z}\\u_{,y}+v_{,x}\\u_{,z}+w_{,x}\\v_{,z}+w_{,y}\end{pmatrix}+\begin{pmatrix}\frac{1}{2}\left(u_{,x^2}+v_{,x^2}+w_{,x^2}\right)\\\frac{1}{2}\left(u_{,y^2}+v_{,y^2}+w_{,y^2}\right)\\\frac{1}{2}\left(u_{,z^2}+v_{,z^2}+w_{,z^2}\right)\\u_{,x}u_{,y}+v_{,x}v_{,y}+w_{,x}w_{,y}\\u_{,x}u_{,z}+v_{,x}v_{,z}+w_{,x}w_{,z}\\u_{,y}u_{,z}+v_{,y}v_{,z}+w_{,y}w_{,z}\end{pmatrix}$$

On le calcule comme suit :

$$E=\left[Q+\frac{1}{2}A\left(\frac{\partial u}{\partial x}\right)\right]\frac{\partial u}{\partial x}$$

avec :

$$Q=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&1\\0&1&0&1&0&0&0&0&0\\0&0&1&0&0&0&1&0&0\\0&0&0&0&0&1&0&1&0\end{bmatrix}$$

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

et le vecteur du gradient des déplacements :

$$\frac{\partial u}{\partial x}=\begin{pmatrix} u_{,x} \\ u_{,y} \\ u_{,z} \\ v_{,x} \\ v_{,y} \\ v_{,z} \\ w_{,x} \\ w_{,y} \\ w_{,z} \end{pmatrix}$$

et le tenseur A dépendant du gradient des déplacements :

$$A\left(\frac{\partial u}{\partial x}\right)=\begin{bmatrix} u_{,x} & 0 & 0 & v_{,x} & 0 & 0 & w_{,x} & 0 & 0 \\ 0 & u_{,y} & 0 & 0 & v_{,y} & 0 & 0 & w_{,y} & 0 \\ 0 & 0 & u_{,z} & 0 & 0 & v_{,z} & 0 & 0 & w_{,z} \\ u_{,y} & u_{,x} & 0 & v_{,y} & v_{,x} & 0 & w_{,y} & w_{,x} & 0 \\ u_{,z} & 0 & u_{,x} & v_{,z} & 0 & v_{,x} & w_{,z} & 0 & w_{,x} \\ 0 & u_{,z} & u_{,y} & 0 & v_{,z} & v_{,y} & 0 & w_{,z} & w_{,y} \end{bmatrix}$$

La variation virtuelle, notée δ, des déformations de Green-Lagrange est obtenue par un calcul différentiel :

$$\delta E=\left[Q+A\left(\frac{\partial u}{\partial x}\right)\right]\frac{\partial \delta u}{\partial x}$$

Dans cette expression et dans celle qui suit, nous avons tenu compte de la propriété suivante (voir [bib4] p 141) :

$$\frac{1}{2}A\left(\frac{\partial \delta u}{\partial x}\right)\frac{\partial u}{\partial x}=\frac{1}{2}A\left(\frac{\partial u}{\partial x}\right)\frac{\partial \delta u}{\partial x}$$

La variation itérative Δ est elle aussi obtenue par un calcul différentiel :

$$\Delta E=\left[Q+A\left(\frac{\partial u}{\partial x}\right)\right]\frac{\partial \Delta u}{\partial x}$$

La variation itérative de la déformation virtuelle de Green-Lagrange se met ainsi sous la forme :

$$\Delta \delta E=\underbrace{A\left(\frac{\partial \Delta u}{\partial x}\right)\frac{\partial \delta u}{\partial x}}_{\text{terme classique}}+\underbrace{\left[Q+A\left(\frac{\partial u}{\partial x}\right)\right]\frac{\partial \Delta \delta u}{\partial x}}_{\text{terme non classique}}$$

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

Alors que le premier terme de cette expression est classique pour les milieux continus 3D, le deuxième, qui traduit la prise en compte des grandes rotations, l'est moins.

3.1.4 Calcul des contraintes de Cauchy

3.1.4.1 Cas général

Le tenseur 3×3 des contraintes globales de Piola-Kirchhoff de deuxième espèce est relié au tenseur 3×3 des contraintes globales de Cauchy par la relation :

$$[S]=\det(F)\,F^{-1}[\sigma]\,F^{-T}$$

Ainsi, connaissant l'état des contraintes de Piola-Kirchhoff de deuxième espèce, on peut calculer l'état des contraintes de Cauchy par la relation :

$$[\sigma]=\frac{1}{\det(F)}F[S]F^{T}$$

Il est à noter que l'état de contraintes de Cauchy n'est pas plan, en général, contrairement à l'état de contraintes de Piola-Kirchhoff de deuxième espèce. Par ailleurs, le choix d'un repère local dans lequel représenter ce tenseur n'est pas du tout évident. On montrera cependant, au paragraphe suivant, que dans le cadre des petites déformations, il existe un repère local, facilement identifiable, dans lequel l'état de contraintes de Cauchy est lui-aussi plan.

Dans le cas de lois tout à fait générales, une attention particulière devra porter sur les schémas d'intégration numérique permettant de calculer les valeurs de substitution du gradient F aux points d'intégration numérique normale.

3.1.4.2 Approximation en petites déformations

On rappelle [bib4] que le gradient F peut être écrit grâce à la décomposition polaire sous deux formes :

$$F=\mathbf{RU}=\mathbf{VR}$$

où $R=R^{-T}$ est un tenseur orthogonal, et où U et V sont des matrices d'élongation symétriques définies positives.

Dans le domaine non linéaire géométrique, nous pouvons introduire une simplification importante dans la décomposition polaire du gradient des déformations si les déformations restent petites. Cette simplification n'est pas introduite dans le calcul non linéaire mais en post-traitement des contraintes.

L'élongation au point Q étant mineure devant la grande rotation de la section :

$$U\approx V\approx I$$

On peut alors écrire :

$$F\approx R=\Lambda$$

où Λ est le tenseur de grande rotation qui transforme la normale n en n^{φ} :

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

$$\Lambda\, n = n^{\varphi}$$

La simplification traduit le fait que sur une section, la transformation est réduite à une grande rotation. Avec cette approximation du gradient des déformations, on peut écrire :

$$F \approx R = \Lambda$$

et donc, en exploitant l'orthogonalité de Λ on obtient :

$$F^{-1} \approx \Lambda^{T}$$

et :

$$\det(F) \approx 1 \ .$$

Ces simplifications conduisent à la relation finale :

$$[\sigma] \approx \Lambda [S] \Lambda^{T}$$

Cette relation traduit le fait que les contraintes de Cauchy sont tout simplement obtenues par la grande rotation des contraintes de Piola-Kirchhoff de deuxième espèce.

On peut maintenant réécrire la propriété de contraintes planes du tenseur de Piola-Kirchhoff de deuxième espèce $n.[S]\, n = 0$ sous la nouvelle forme :

$$n.\Lambda^{T}[s]\Lambda\, n = 0$$

qui conduit par ailleurs à la propriété :

$$n^{\varphi}.[\sigma]\, n^{\varphi} = 0$$

Soit encore :

$$\tilde{\sigma}_{n^{\varphi} n^{\varphi}} = 0$$

Les contraintes de Cauchy $\left[\sigma\left(\xi_1, \xi_2, \xi_3\right)\right]$ sont aussi planes dans le repère local $\left[t_1^{\varphi}\left(\xi_1, \xi_2, \xi_3\right) : t_2^{\varphi}\left(\xi_1, \xi_2, \xi_3\right) : n^{\varphi}\left(\xi_1, \xi_2\right)\right]$ obtenu par grande rotation du repère local sur la configuration initiale :

$$\left[t_1^{\varphi} : t_2^{\varphi} : n^{\varphi}\right] = \Lambda \left[t_1 : t_2 : n\right]$$

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

Dans ce repère, nous pouvons écrire toutes les composantes du tenseur $[s]$ comme suit :

$$\begin{bmatrix} \tilde{\sigma}_{t_1^\varphi t_1^\varphi} & \tilde{\sigma}_{t_1^\varphi t_2^\varphi} & \tilde{\sigma}_{t_1^\varphi n^\varphi} \\ \tilde{\sigma}_{t_2^\varphi t_1^\varphi} & \tilde{\sigma}_{t_2^\varphi t_1^\varphi} & \tilde{\sigma}_{t_2^\varphi n^\varphi} \\ \tilde{\sigma}_{t_1^\varphi n^\varphi} & \tilde{\sigma}_{t_1^\varphi n^\varphi} & 0 \end{bmatrix} = \begin{bmatrix} t_1^\varphi.[\sigma]t_1^\varphi & t_1^\varphi.[\sigma]t_2^\varphi & t_1^\varphi.[\sigma]n^\varphi \\ t_2^\varphi.[\sigma]t_1^\varphi & t_2^\varphi.[\sigma]t_2^\varphi & t_2^\varphi.[\sigma]n^\varphi \\ n^\varphi.[\sigma]t_1^\varphi & n^\varphi.[\sigma]t_2^\varphi & n^\varphi.[\sigma]n^\varphi \end{bmatrix}$$

En reprenant la relation $[\sigma] \approx \Lambda[S]\Lambda^T$, on obtient :

$$\begin{bmatrix} t_1^\varphi.[\sigma]t_1^\varphi & t_1^\varphi.[\sigma]t_2^\varphi & t_1^\varphi.[\sigma]n^\varphi \\ t_2^\varphi.[\sigma]t_1^\varphi & t_2^\varphi.[\sigma]t_2^\varphi & t_2^\varphi.[\sigma]n^\varphi \\ n^\varphi.[\sigma]t_1^\varphi & n^\varphi.[\sigma]t_2^\varphi & n^\varphi.[\sigma]n^\varphi \end{bmatrix} = \begin{bmatrix} t_1.[S]t_1 & t_1.[S]t_2 & t_1.[S]n \\ t_2.[S]t_1 & t_2.[S]t_2 & t_2.[S]n \\ n.[S]t_1 & n.[S]t_2 & n.[S]n \end{bmatrix}$$

d'où le résultat final :

$$\begin{bmatrix} \tilde{\sigma}_{t_1^\varphi t_1^\varphi} & \tilde{\sigma}_{t_1^\varphi t_2^\varphi} & \tilde{\sigma}_{t_1^\varphi n^\varphi} \\ \tilde{\sigma}_{t_2^\varphi t_1^\varphi} & \tilde{\sigma}_{t_2^\varphi t_1^\varphi} & \tilde{\sigma}_{t_2^\varphi n^\varphi} \\ \tilde{\sigma}_{t_1^\varphi n^\varphi} & \tilde{\sigma}_{t_1^\varphi n^\varphi} & 0 \end{bmatrix} = \begin{bmatrix} \tilde{S}_{t_1 t_1} & \tilde{S}_{t_1 t_2} & \tilde{S}_{t_1 n} \\ \tilde{S}_{t_2 t_2} & \tilde{S}_{t_2 t_2} & \tilde{S}_{t_2 n} \\ \tilde{S}_{t_1 n} & \tilde{S}_{t_2 n} & 0 \end{bmatrix}$$

Pour autant que la déformation reste petite, les composantes du tenseur des contraintes de Cauchy dans le repère local attaché à la configuration déformée sont identiques aux composantes du tenseur des contraintes de Piola-Kirchhoff de deuxième espèce dans le repère local attaché à la configuration initiale.

Nous prenons le parti dans la suite, de ne considérer que les contraintes de Piola-Kirchhoff de deuxième espèce. Nous devons noter que dans le cadre d'une loi constitutive plus générale, on pourra passer d'une mesure de contrainte à une autre comme indiqué au paragraphe précédent.

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

4 Discrétisation numérique de la formulation variationnelle issue du principe du travail virtuel

4.1 Éléments finis

Les trois figures ci-dessous résument les choix éléments finis concernant les coques volumiques [R3.07.04]. Les éléments finis choisis sont des quadrangles ou des triangles isoparamétriques. Le quadrangle est représenté ci-dessous. On choisit parmi les éléments à fonctions d'interpolation quadratiques, l'élément hétérosis dont les déplacements sont approchés par les fonctions d'interpolation de l'élément Sérendip et les rotations par les fonctions de l'élément de Lagrange. Toutes les justifications quant à ces choix sont données dans [R3.07.04].

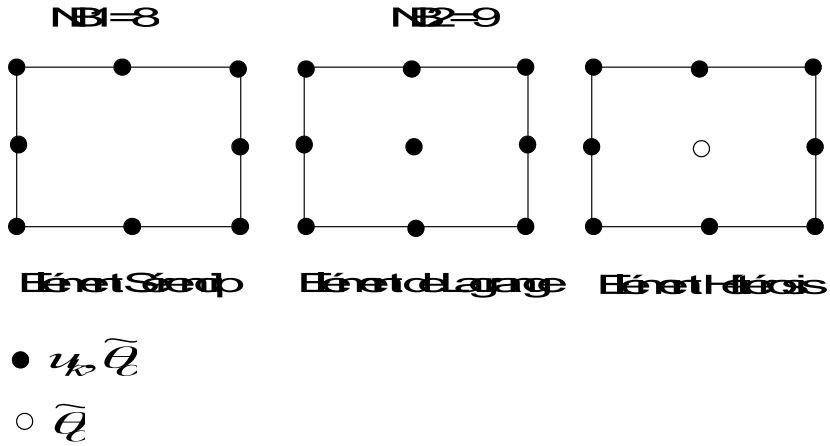


Figure 4.1-a : Familles d'éléments finis pour le quadrangle isoparamétrique

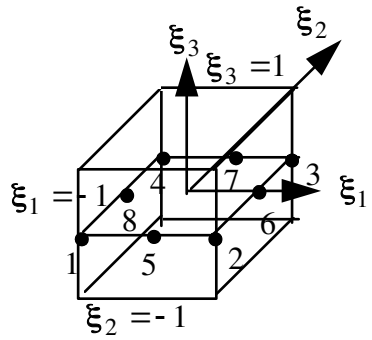


Figure 4.1-b : Élément volumique de référence

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

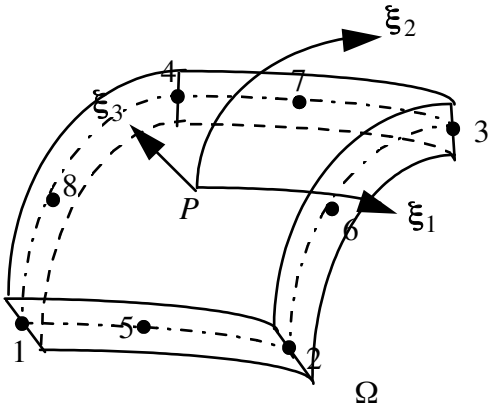


Figure 41-c : Élément volumique réel

4.2 Discrétisation du champ de déplacement

Dans le but d'éviter le calcul explicite des courbures, qui devient extrêmement lourd dans le cas des grandes rotations, nous choisissons d'interpoler la normale à la surface moyenne initiale au lieu d'interpoler les rotations :

$$n\left(\xi_1,\xi_2\right)=\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right) n_I$$

où $N_I^{(2)}\left(\xi_1,\xi_2\right)$ désigne la fonction d'interpolation au nœud I parmi les $NB2$ nœuds de Lagrange.

Les mêmes interpolations sont adoptées pour la transformée de la normale initiale :

$$n^{\varphi}\left(\xi_1,\xi_2\right)=\sum_{I=1}^{NB2} N_1^{(2)}\left(\xi_1,\xi_2\right) n_I^{\varphi}$$

L'interpolation de la position initiale d'un point sur la surface moyenne de la coque (point P) est donnée par :

$$x\left(\xi_1,\xi_2\right)=\sum_{I=1}^{NB1} N_I^{(1)}\left(\xi_1,\xi_2\right)\begin{pmatrix} x \\ y \\ z \end{pmatrix}_I$$

où $N_I^{(1)}\left(\xi_1,\xi_2\right)$ désigne la fonction d'interpolation au nœud I parmi les $NB1=NB2-1$ nœuds de Serendip.

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

L'interpolation de la position initiale d'un point quelconque de la coque (point Q) peut alors être écrite sous la forme :

$$x\left(\xi_1,\xi_2,\xi_3\right)=\sum_{I=1}^{NB1} N_I^{(1)}\left(\xi_1,\xi_2\right)\begin{pmatrix}x\\y\\z\end{pmatrix}_I+\xi_3\frac{h}{2}\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right)\begin{pmatrix}n_x\\n_y\\n_z\end{pmatrix}$$

Les mêmes interpolations sont adoptées pour la position déformée d'un point quelconque de la fibre :

$$x^{\varphi}\left(\xi_1,\xi_2,\xi_3\right)=\sum_{I=1}^{NB1} N_I^{(1)}\left(\xi_1,\xi_2\right)\begin{pmatrix}x^{\varphi}\\y^{\varphi}\\z^{\varphi}\end{pmatrix}_I+\xi_3\frac{h}{2}\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right)\begin{pmatrix}n_x^{\varphi}\\n_y^{\varphi}\\n_z^{\varphi}\end{pmatrix}$$

Les interpolations pour les positions initiale et déformée étant les mêmes, nous pouvons les adopter pour le déplacement réel d'un point quelconque de la coque :

$$u\left(\xi_1,\xi_2,\xi_3\right)=\sum_{I=1}^{NB1} N_I^{(1)}\left(\xi_1,\xi_2\right)\begin{pmatrix}u\\v\\w\end{pmatrix}_I+\xi_3\frac{h}{2}\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right)\left(\begin{pmatrix}n_x^{\varphi}\\n_y^{\varphi}\\n_z^{\varphi}\end{pmatrix}_I-\begin{pmatrix}n_x\\n_y\\n_z\end{pmatrix}_I\right)$$

Ainsi, l'interpolation du déplacement virtuel devient :

$$\delta u\left(\xi_1,\xi_2,\xi_3\right)=\sum_{I=1}^{NB1} N_I^{(1)}\left(\xi_1,\xi_2\right)\begin{pmatrix}\delta u\\\delta v\\\delta w\end{pmatrix}_I-\xi_3\frac{h}{2}\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right)\begin{bmatrix}0&-n_z^{\varphi}&n_y^{\varphi}\\n_z^{\varphi}&0&-n_x^{\varphi}\\-n_y^{\varphi}&n_x^{\varphi}&0\end{bmatrix}_I\begin{pmatrix}\delta w_x\\\delta w_y\\\delta w_z\end{pmatrix}_I$$

De même, l'interpolation du déplacement itératif devient :

$$\Delta u\left(\xi_1,\xi_2,\xi_3\right)=\sum_{I=1}^{NB1} N_I^{(1)}\left(\xi_1,\xi_2\right)\begin{pmatrix}\Delta u\\\Delta v\\\Delta w\end{pmatrix}_I-\xi_3\frac{h}{2}\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right)\begin{bmatrix}0&-n_z^{\varphi}&n_y^{\varphi}\\n_z^{\varphi}&0&-n_x^{\varphi}\\-n_y^{\varphi}&n_x^{\varphi}&0\end{bmatrix}_I\begin{pmatrix}\Delta w_x\\\Delta w_y\\\Delta w_z\end{pmatrix}_I$$

De plus, l'interpolation de la variation itérative du déplacement virtuel est :

$$\Delta\,\delta u\left(\xi_1,\xi_2,\xi_3\right)=\xi_3\frac{h}{2}\sum_{I=1}^{NB2} N_I^{(2)}\left(\xi_1,\xi_2\right)\left(\delta w\wedge\left(\Delta w\wedge n^{\varphi}\right)\right)_I$$

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

4.3 Discrétisation du gradient du déplacement

4.3.1 Gradient du déplacement total

Le vecteur du gradient du déplacement réel peut être relié au gradient isoparamétrique du déplacement réel par la relation suivante :

$$\frac{\partial u}{\partial x}=\tilde{J}^{-1}\frac{\partial u}{\partial \xi}$$

Le gradient isoparamétrique du déplacement est organisé comme suit :

$$\frac{\partial u}{\partial \xi}=\begin{Bmatrix} u,_{\xi_1} \\ u,_{\xi_2} \\ u,_{\xi_3} \\ v,_{\xi_1} \\ v,_{\xi_2} \\ v,_{\xi_3} \\ w,_{\xi_1} \\ w,_{\xi_2} \\ w,_{\xi_3} \end{Bmatrix}$$

La matrice jacobienne généralisée 9×9 $\tilde{J}^{-1}$ peut être exprimée en fonction de la matrice jacobienne de la transformation isoparamétrique 3×3 comme suit :

$$\tilde{J}^{-1}=\begin{bmatrix} J^{-1} & 0 & 0 \\ 0 & J^{-1} & 0 \\ 0 & 0 & J^{-1} \end{bmatrix}$$

Le gradient isoparamétrique du déplacement réel peut être calculé comme suit :

$$\frac{\partial u}{\partial \xi}=\left[\frac{\partial N}{\partial \xi}\right]_1 p^e$$

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

avec la première matrice des dérivées des fonctions de forme :

$$\left[\begin{array}{ccc} N^{(1)}_{I,\xi_1} & 0 & 0 \\ N^{(1)}_{I,\xi_2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & N^{(1)}_{I,\xi_1} & 0 \\ 0 & N^{(1)}_{I,\xi_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & N^{(1)}_{I,\xi_1} \\ 0 & 0 & N^{(1)}_{I,x_2} \\ 0 & 0 & 0 \end{array}\right] \frac{h}{2} \left[\begin{array}{ccc} \xi_3 N^{(2)}_{I,\xi_1} & 0 & 0 \\ \xi_3 N^{(2)}_{I,\xi_2} & 0 & 0 \\ N^{(2)}_{I} & 0 & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_1} & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} & 0 \\ 0 & N^{(2)}_{I} & 0 \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_1} \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_2} \\ 0 & 0 & N^{(2)}_{I} \end{array}\right] \ldots I=1, NB1$$

$$\frac{h}{2} \left[\begin{array}{ccc} \xi_3 N^{(2)}_{NB2,\xi_1} & 0 & 0 \\ \xi_3 N^{(2)}_{NB2,\xi_2} & 0 & 0 \\ N^{(2)}_{NB2} & 0 & 0 \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_1} & 0 \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_2} & 0 \\ 0 & N^{(2)}_{NB2} & 0 \\ 0 & 0 & \xi_3 N^{(2)}_{NB2,x_1} \\ 0 & 0 & \xi_3 N^{(2)}_{NB2,\xi_2} \\ 0 & 0 & N^{(2)}_{NB2} \end{array}\right]$$

$$\left[\frac{\partial N}{\partial \xi}\right]_1 = ¿[¿][¿][]¿$$

et le vecteur de « déplacement réel nodal généralisé » :

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

$$p^e = \begin{matrix} \vdots \\ \begin{pmatrix} u \\ v \\ w \\ n_x^{\varphi} - n_x \\ n_y^{\varphi} - n_y \\ n_z^{\varphi} - n_z \end{pmatrix}_I \\ \vdots \\ I = 1, NB\,1 \\ \vdots \\ \begin{pmatrix} n_x^{\varphi} - n_x \\ n_y^{\varphi} - n_y \\ n_z^{\varphi} - n_z \end{pmatrix}_{NB\,2} \end{matrix}$$

Finalement, on pourra écrire le gradient de déplacement réel sous la forme :

$$\frac{\partial u}{\partial x} = \tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_1 p^e$$

4.3.2 Gradient du déplacement virtuel

En procédant de façon analogue au gradient du déplacement réel, on peut relier les deux gradients de déplacement virtuel :

$$\frac{\partial \delta u}{\partial x} = \tilde{J}^{-1} \frac{\partial \delta u}{\partial \xi}$$

Le gradient isoparamétrique du déplacement virtuel peut être calculé comme suit :

$$\frac{\partial \delta u}{\partial \xi} = \left[\frac{\partial N}{\partial \xi} \right]_2 \delta u^e$$

avec la deuxième matrice des dérivées des fonctions de forme :

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

$$\cdots \begin{vmatrix} N^{(1)}_{I,\xi_1} & 0 & 0 \\ N^{(1)}_{I,\xi_2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & N^{(1)}_{I,\xi_1} & 0 \\ 0 & N^{(1)}_{I,\xi_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & N^{(1)}_{I,\xi_1} \\ 0 & 0 & N^{(1)}_{I,\xi_2} \\ 0 & 0 & 0 \end{vmatrix} \frac{h}{2} \begin{vmatrix} 0 & \xi_3 N^{(2)}_{I,\xi_1} n_z^{\varphi} & -\xi_3 N^{(2)}_{I,\xi_1} n_y^{\varphi} \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} n_z^{\varphi} & -\xi_3 N^{(2)}_{I,\xi_2} n_y^{\varphi} \\ 0 & N^{(2)}_1 n_z^{\varphi} & -N^{(2)}_1 n_y^{\varphi} \\ -\xi_3 N^{(2)}_{I,\xi_1} n_z^{\varphi} & 0 & \xi_3 N^{(2)}_{I,\xi_1} n_x^{\varphi} \\ -\xi_3 N^{(2)}_{I,\xi_2} n_z^{\varphi} & 0 & \xi_3 N^{(2)}_{I,\xi_2} n_x^{\varphi} \\ -N^{(2)}_1 n_z^{\varphi} & 0 & N^{(2)}_1 n_x^{\varphi} \\ \xi_3 N^{(2)}_{I,\xi_1} n_y^{\varphi} & -\xi_3 N^{(2)}_{I,\xi_1} n_x^{\varphi} & 0 \\ \xi_3 N^{(2)}_{I,\xi_2} n_y^{\varphi} & -\xi_3 N^{(2)}_{I,\xi_2} n_x^{\varphi} & 0 \\ N^{(2)}_1 n_y^{\varphi} & -N^{(2)}_1 n_x^{\varphi} & 0 \end{vmatrix} \cdots I=1,NB1$$

$$\left| \frac{h}{2} \begin{vmatrix} 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n_z^{\varphi} & -\xi_3 N^{(2)}_{NB2,\xi_1} n_y^{\varphi} \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n_z^{\varphi} & -\xi_3 N^{(2)}_{NB2,\xi_2} n_y^{\varphi} \\ 0 & N^{(2)}_{NB2} n_z^{\varphi} & -N^{(2)}_{NB2} n_y^{\varphi} \\ -\xi_3 N^{(2)}_{NB2,\xi_1} n_z^{\varphi} & 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n_x^{\varphi} \\ -\xi_3 N^{(2)}_{NB2,\xi_2} n_z^{\varphi} & 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n_x^{\varphi} \\ -N^{(2)}_{NB2} n_z^{\varphi} & 0 & N^{(2)}_{NB2} n_x^{\varphi} \\ \xi_3 N^{(2)}_{NB2,\xi_1} n_y^{\varphi} & -\xi_3 N^{(2)}_{NB2,\xi_1} n_x^{\varphi} & 0 \\ \xi_3 N^{(2)}_{NB2,\xi_2} n_y^{\varphi} & -\xi_3 N^{(2)}_{NB2,\xi_2} n_x^{\varphi} & 0 \\ N^{(2)}_{NB2} n_y^{\varphi} & -N^{(2)}_{NB2} n_x^{\varphi} & 0 \end{vmatrix} \right|$$

$$\left[\frac{\partial N}{\partial \xi} \right]_2 = ¿[¿][¿][]$$

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

et le vecteur des variables nodales virtuelles :

$$\delta u^e = \begin{matrix} \vdots \\ \begin{pmatrix} \delta u \\ \delta v \\ \delta w \\ \delta w_x \\ \delta w_y \\ \delta w_z \end{pmatrix}_I \\ \vdots \\ I = 1, NB1 \\ \vdots \\ \begin{pmatrix} \delta w_x \\ \delta w_y \\ \delta w_z \end{pmatrix}_{NB2} \end{matrix}$$

Finalement, on pourra écrire le gradient de déplacement virtuel sous la forme :

$$\frac{\partial \delta u}{\partial x} = \tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_2 \delta u^e$$

4.3.3 Gradient du déplacement itératif

La démarche ici est similaire au calcul virtuel. Il suffit de remplacer δ par Δ :

$$\frac{\partial \Delta u}{\partial x} = \tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_2 \Delta u^e$$

avec le vecteur des variables nodales itératives :

$$\Delta u^e = \begin{matrix} \vdots \\ \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \\ \Delta w_x \\ \Delta w_y \\ \Delta w_z \end{pmatrix}_I \\ \vdots \\ I = 1, NB1 \\ \vdots \\ \begin{pmatrix} \Delta w_x \\ \Delta w_y \\ \Delta w_z \end{pmatrix}_{NB2} \end{matrix}$$

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

4.3.4 Gradient de la variation itérative du déplacement virtuel

$$\frac{\partial \Delta \delta u}{\partial x} = \tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_3 q^e$$

avec :

$$\left[\frac{\partial N}{\partial \xi} \right]_3 = \left[\cdots \frac{h}{2} \begin{bmatrix} x_3 N^{(2)}_{I,\xi_1} & 0 & 0 \\ x_3 N^{(2)}_{I,\xi_2} & 0 & 0 \\ N^{(2)}_I & 0 & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_1} & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} & 0 \\ 0 & N^{(2)}_I & 0 \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_1} \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_2} \\ 0 & 0 & N^{(2)}_I \end{bmatrix} \cdots I = 1, NB2 \right]$$

et le vecteur de la variation itérative “ déplacement virtuel nodal ” généralisé :

$$q^e = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \left(\delta w \wedge \left(\Delta w \wedge n^{\varphi} \right) \right)_I \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} \quad I = 1, NB2$$

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

4.4 Discrétisation de la formulation variationnelle issue du principe du travail virtuel

Nous reprenons la variation itérative (entre deux itérations) du travail virtuel interne :

$$\Delta \delta \pi_{\text{int}} = \int_{\Omega} \left(\delta \tilde{E} . \Delta \tilde{S} + \Delta \delta \tilde{E} . \tilde{S} \right) d\Omega$$

et la variation itérative du vecteur de contraintes locales de Piola-Kirchhoff de deuxième :

$$\Delta \tilde{S} = D \Delta \tilde{E}$$

Alors, la forme linéarisée du principe des travaux virtuels du §3 peut être écrite pour l'élément fini décrit ci-dessus sous la forme matricielle suivante :

$$\delta u^e . K^{e_T} \Delta u^e = \delta u^e . \left(\lambda f^e - r^e \right)$$

où δu^e est le vecteur nodal des fonctions tests. On en déduit le système d'équations :

$$K^{e_T} \Delta u^e = \lambda f^e - r^e$$

où :

K^{e_T} est la matrice tangente de rigidité

Δu^e est le vecteur élémentaire de la solution du système linéarisé d'équations (vecteur nodal entre deux itérations)

λ est le niveau de charge externe

f^e est le vecteur nodal des forces externes (associé à $\lambda = 1$)

r^e est le vecteur nodal des forces internes

4.4.1 Vecteur des forces internes

Il s'agit là d'un vecteur $(6 \times Nbl + 3) \times 1$ entièrement exprimé dans le repère global et qui doit être évalué à chaque itération par la relation :

$$r^e = \int_{\Omega_\xi} \tilde{B}_2^T \tilde{S} \det J \, d\xi_1 d\xi_2 d\xi_3$$

avec le vecteur des contraintes locales Piola-Kirchhoff de deuxième espèce :

$$\tilde{S} = D \tilde{E}$$

On rappelle que le symbole ~ désigne un objet exprimé dans le repère local.

Les déformations locales de Green-Lagrange sont remises à jour à chaque itération :

$$\tilde{E} = \tilde{B}_1 p^e$$

où l'opérateur des déformations totales (premier opérateur des déformations) s'écrit :

$$\tilde{B}_1 = H \left[Q + \frac{1}{2} A \left(\frac{\partial u}{\partial x} \right) \right] \tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_1$$

avec le gradient du déplacement réel :

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

$$\frac{\partial u}{\partial x}=\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_1 p^e$$

L'opérateur des déformations virtuelles (deuxième opérateur des déformations) :

$$\tilde{B}_2=H\left(Q+A\left(\frac{\partial u}{\partial x}\right)\right)\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2$$

est mis en évidence par les relations :

$$\delta \tilde{E}=\tilde{B}_2 \delta u^e$$

$$\Delta \tilde{E}=\tilde{B}_2 \Delta u^e$$

4.4.2 Matrice tangente de rigidité

La matrice tangente de rigidité qui est de taille $(6\times NB1+3)\times(6\times NB1+3)$ s'exprime aussi entièrement dans le repère global. On doit pouvoir l'évaluer à chaque itération si on veut que la convergence de la méthode de Newton soit quadratique. D'une façon classique en non linéaire géométrique, elle prend la forme :

$$K_T^e=K_m^e+K_g^e$$

où la première partie représente la partie matérielle :

$$K_m^e=\int_{\Omega_\xi}\tilde{B}_2^T D\,\tilde{B}_2 \det J\, d\xi_1 d\xi_2 d\xi_3$$

et la deuxième partie représente la partie géométrique, elle même composée de deux parties :

$$K_g^e=K_g^{e\,\text{classique}}+K_g^{e\,\text{non classique}}$$

avec la partie classique de la partie géométrique (voir [bib4] p. 141) :

$$K_g^{e\,\text{classique}}=\int_{\Omega_\xi}\left[\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2\right]^T \bar{S}\left[\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2\right]\det J\, d\xi_1 d\xi_2 d\xi_3$$

où $\bar{S}$ le tenseur généralisé des contraintes exprimées dans le repère global s'écrit :

$$\overset{9\times 9}{\bar{S}}=\begin{bmatrix}\overset{3\times 3}{[S]} & 0 & 0\\ 0 & [S] & 0\\ 0 & 0 & [S]\end{bmatrix}$$

La partie non classique de la partie géométrique représente des termes découplés de rotation non symétriques qui ont pour forme :

$$\overset{3\times 3}{K_{g\,\text{non classique}}^e}(I,I)=[z_I\times][n_I^j\times]$$

où n_I^{φ} est la transformée de la normale initiale au nœud I et z_I un vecteur 3×1 au nœud $I=1,NB2$ du vecteur nodal $3\times(NB2\times 1)\,Z_I$

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

$$Z_I = \begin{pmatrix} \vdots \\ z_I \\ \vdots \\ I = 1, NB2 \end{pmatrix}$$

Le vecteur nodal Z_I est similaire à un vecteur de force interne et son expression est :

$$Z_I = \int_{\Omega_\xi} \tilde{B}_3^T \tilde{S} \det J \, d\xi_1 d\xi_2 d\xi_3$$

avec l'opérateur de la variation itérative des déformations virtuelles (troisième opérateur des déformations) :

$$\tilde{B}_3 = H\left[Q + A\left(\frac{\partial u}{\partial x} \right) \right] \tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_3$$

qui est mis en évidence par la relation :

$$\int_{\Omega_\xi} \tilde{B}_3^T \tilde{S} \det J \, d\xi_1 \, d\xi_2 \, d\xi_3 = \int_\Omega \Delta \, \delta \, \tilde{E}_{\text{non classique}} . \tilde{S} \, d\Omega$$

4.4.3 Schémas d'intégration

L'intégration des termes de rigidité dans l'épaisseur de la coque est identique à la méthode utilisée en analyse linéaire géométrique [R3.07.04] pour des comportements non linéaires. L'épaisseur initiale est divisée en N couches d'épaisseurs identiques. Il y a trois points d'intégration par couche. Les points d'intégration sont situés en peau supérieure de couche, au milieu de la couche et en peau inférieure de couche. Une couche dans l'épaisseur de la coque se révèle suffisante dans la plupart des cas.

Afin de pouvoir palier au problème de blocage en membrane des coques courbes et de résoudre le problème de blocage en cisaillement transverse, il est nécessaire de modifier le schéma d'intégration sur la surface moyenne. Si la technique est complètement connue en analyse linéaire, elle l'est moins en analyse non linéaire géométrique.

La procédure se présente comme une généralisation de la séparation des effets de membrane, de flexion et de cisaillement transverse au cas où l'on utilise les déformations de Green-Lagrange :

$$\tilde{E} = \begin{pmatrix} \tilde{E}_m \\ \tilde{E}_S \end{pmatrix}$$

où $\tilde{E}_m = \begin{pmatrix} \tilde{E}_{t_1 t_1} \\ \tilde{E}_{t_2 t_2} \\ \tilde{\gamma}_{t_1 t_2} \end{pmatrix}$ représente la déformation de membrane-flexion et $\tilde{E}_S = \begin{pmatrix} \tilde{\gamma}_{t_1 n} \\ \tilde{\gamma}_{t_2 n} \end{pmatrix}$ la déformation de cisaillement transverse.

Lors de l'évaluation numérique des déformations aux points d'intégration numérique normale de Gauss (9 points pour le quadrilatère et 7 points pour le triangle), on utilise la relation $\tilde{E} = \tilde{B}_1 p^e$. La modification est introduite au niveau du premier opérateur des déformations :

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

$$\tilde{B}_1 = \begin{bmatrix} \tilde{B}_{mf_1^{\text{substitution}}} \\ \tilde{B}_{s_1^{\text{substitution}}} \end{bmatrix}$$

$\tilde{B}_{\mathbf{mf}_1^{\text{substitution}}}$ et $\tilde{B}_{s_1^{\text{substitution}}}$ sont les premiers opérateurs des déformations de substitution de membrane-flexion et de cisaillement transverse, respectivement.

Lors du calcul du vecteur nodal des forces internes et de la partie matérielle de la matrice tangente de rigidité, la modification est introduite de façon analogue au niveau du deuxième opérateur des déformations :

$$\tilde{B}_2 = \begin{bmatrix} \tilde{B}_{mf_2^{\text{su}}} \\ \tilde{B}_{s_2^{\text{su}}} \end{bmatrix}$$

4.4.3.1 Opérateurs de déformations de substitution

Dans ce qui suit les points d'intégration numérique normale et réduite de Gauss, sur la surface moyenne, sont au nombre de $\mathrm{NPGSN}=9$ et $\mathrm{NPGSR}=4$, respectivement, pour l'élément quadrilatéral, et $\mathrm{NPGSN}=7$ et $\mathrm{NPGSR}=3$, respectivement, pour l'élément triangulaire.

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

Partie membrane-flexion

Au point $INTSN$ parmi les $NPGSN$ points d'intégration numérique normale de Gauss de la surface moyenne, on calculera :

$$\tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{substitution}}(\mathrm{INTSN}) = \tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{normal\ complet}}(INTSN) - \tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{normal\ incomplet}}(\mathrm{INTSN}) + \sum_{\mathrm{INTSR}=1,\,\mathrm{NPGSR}} N_I(\mathrm{INTSN})\,\tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{réduit\ incomplet}}(\mathrm{INTSR})$$

où $INTSR$ est un point parmi les $NPGSR$ points d'intégration numérique réduite de Gauss de la surface moyenne.

Dans l'expression ci-dessus, $\tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{normal\ complet}}$ représente les trois premières lignes de $\tilde{B}_1$ calculé aux points d'intégration numérique normale en considérant la matrice complète $\left[\frac{\partial N}{\partial \xi}\right]_1$. L'opérateur $\tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{normal\ incomplet}}$ représente les trois premières lignes de $\tilde{B}_1$ calculé aux points d'intégration numérique normale en considérant une matrice $\left[\frac{\partial N}{\partial \xi}\right]_1$ incomplète où les colonnes de rotation sont annulées :

$$\left[\frac{\partial N}{\partial \xi}\right]_1^{inc} = \left[\cdots \begin{bmatrix} N_{I,\xi_1}^{(1)} & 0 & 0 & 0 & 0 & 0 \\ N_{I,\xi_2}^{(1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_{I,\xi_1}^{(1)} & 0 & 0 & 0 & 0 \\ 0 & N_{I,\xi_2}^{(1)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N_{I,\xi_1}^{(1)} & 0 & 0 & 0 \\ 0 & 0 & N_{I,\xi_2}^{(1)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdots I=1,NB1 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right]$$

Les $\tilde{\mathbf{B}}_{\mathrm{mf1}}^{\mathrm{réduit\ incomplet}}(INSTR)$ représentent les trois premières lignes de $\tilde{B}_1$ calculées aux points d'intégration numérique réduite avec la matrice $\left[\frac{\partial N}{\partial \xi}\right]_1^{inc}$ incomplète définie ci-dessus. Elles sont donc stockées pour être extrapolées à chaque point d'intégration numérique normale.

Partie cisaillement transverse

Pour la partie cisaillement transverse, on calculera :

$$\tilde{\mathbf{B}}_{S1}^{\mathrm{substitution}}(INTSN) = \sum_{INTSR=1,\,NPGSR} N_I(INTSN)\,\tilde{\mathbf{B}}_{S1}^{\mathrm{réduit\ complet}}(INTSR)$$

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

où $\tilde{\mathbf{B}}_{S1}^{\substack{\text{réduit}\\ \text{complet}}}(INTSR)$ représente les deux dernières lignes de $\tilde{B}_1$ calculées aux points d'intégration numérique réduite avec une matrice $\left[\frac{\partial N}{\partial \xi}\right]_1$ complète. Elles sont aussi stockées pour être extrapolées à chaque point d'intégration numérique normale.

4.4.3.2 Substitution de la partie géométrique de la matrice tangente de rigidité

La partie non classique de la matrice tangente de rigidité $K_g^{e\ \text{non classique}}$ est numériquement intégrée aux points d'intégration normale de Gauss. Aucune opération de substitution n'est nécessaire. Pour la partie classique de la matrice tangente de rigidité, nous utilisons la substitution :

$$\mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \text{substitution}} = \mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{normale}\\ \text{complet}\\ \text{membrane}\\ \text{flexion}}} - \mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{normale}\\ \text{incomplet}\\ \text{membrane}\\ \text{flexion}}} + \mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{réduit}\\ \text{incomplet}\\ \text{membrane}\\ \text{flexion}}} + \mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{réduit}\\ \text{complet}\\ \text{cisaillement}\\ \text{transverse}}}$$

où :

$\mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{normale}\\ \text{complet}\\ \text{membrane}\\ \text{flexion}}}$ est numériquement intégrée sur les points d'intégration normale avec une matrice $\left[\frac{\partial N}{\partial \xi}\right]_2$ complète, et les contraintes locales de membrane flexion seulement ;

$\mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{normale}\\ \text{incomplet}\\ \text{membrane}\\ \text{flexion}}}$ est numériquement intégrée sur les points d'intégration normale avec une matrice $\left[\frac{\partial N}{\partial \xi}\right]_2$ incomplète, et les contraintes locales de membrane flexion seulement ;

$\mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{réduit}\\ \text{incomplet}\\ \text{membrane}\\ \text{flexion}}}$ est numériquement sommée sur les points d'intégration réduite avec une matrice $\left[\frac{\partial N}{\partial \xi}\right]_2$ incomplète, et les contraintes locales intégrées de membrane flexion seulement ;

$\mathbf{K}_{\mathbf{g}\ \text{classique}}^{\mathbf{e}\ \substack{\text{normale}\\ \text{complet}\\ \text{cisaillement}\\ \text{transverse}}}$ est numériquement sommée sur les points d'intégration réduite avec une matrice $\left[\frac{\partial N}{\partial \xi}\right]_2$ complète, et les contraintes locales intégrées de cisaillement transverse seulement ;

Pour pouvoir calculer les deux dernières matrices tangentes dans l'équation précédente, nous procédons à l'intégration numérique des contraintes locales sur les $NPGSN$ points d'intégration normale :

$$\tilde{S}(INTSR) = \int_{\Omega_\xi} \sum_{INTSN=1,NPGSN} N_I(INTSR)\,\tilde{S}(INTSN)\det J\, d\xi_1\, d\xi_2\, d\xi_3$$

Cette équation contient les termes de poids des points de Gauss.

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

5 Rigidité autour de la transformée de la normale

5.1 Singularité de la matrice tangente de rigidité

Bien que les objets éléments finis de la coque sont exprimés directement dans le repère global $\left[e_1:e_2:e_3\right]$ (les degrés de liberté sont des déplacements et des rotations dans le repère global), la matrice tangente de rigidité présente une singularité par rapport à la composante de la rotation autour de la transformée de la normale en chaque nœud :

$$\left(\underset{\delta w \Delta w}{K_T^e} \; n^{\varphi}=0 \right)_{I=1,NB2}$$

Les contributions $\left(\Delta w \cdot n^{\varphi}\right)_{I=1,NB2}$ sont nulles.

Dans l'équation précédente, cette matrice représente la rigidité de rotation dans le repère global. Sa structure est pleine :

$$\underset{\delta w \Delta w}{\left[K_T^e\right]_I}=\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{12} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_I$$

c'est une matrice non symétrique.

Cette singularité est une conséquence directe de la cinématique de coque. Elle est due au produit vectoriel apparaissant dans les déplacements linéarisés (virtuel et incrémental). Ainsi le déplacement entre deux itérations est donné par :

$$\Delta u_Q\left(\xi_1,\xi_2,\xi_3\right)=\Delta u_p\left(\xi_1,\xi_2\right)+\xi_3\frac{h}{2}\Delta w\left(\xi_1,\xi_2\right)\wedge n^{\varphi}\left(\xi_1,\xi_2\right)$$

On remarque que la contribution $\xi_3\frac{h}{2}\Delta w\left(\xi_1,\xi_2\right)\wedge n^{\varphi}\left(\xi_1,\xi_2\right)$ est perpendiculaire à n^{φ}. On interprète cette singularité de la façon suivante : la rotation d'une fibre initialement normale à la surface moyenne ne conduit pas à une élongation de celle-ci, et par conséquent n'induit pas de déformation.

5.2 Principe du travail virtuel pour les termes associés à la rotation autour de la normale

Nous proposons de définir la rotation totale autour de la transformée de la normale à la coque comme la projection du vecteur de rotation totale sur la transformée de la normale :

$$\theta_{n^{\varphi}}=\Theta . n^{\varphi}$$

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

On rappelle que le vecteur de rotation Θ est un invariant de la matrice de rotation $\Lambda = \exp[\Theta\times]$

$$\Lambda\,\Theta = \Theta$$

Le vecteur de rotation est un vecteur propre de la matrice de rotation associé à la valeur propre identitaire. De ce fait, la première relation se réécrit :

$$\begin{aligned}\theta_{n^\varphi} &= (\Lambda\,\Theta).(\Lambda\, n)\\ &= \Theta . n\\ &= \theta_n\end{aligned}$$

Cette relation traduit un résultat important :

La projection du vecteur de rotation totale sur la transformée de la normale est égale à la projection du vecteur de rotation totale sur la normale initiale

Sous forme discrète, on définit une énergie de déformation associée à cette rotation :

$$\pi_{n^\varphi} = \frac{1}{2} k \sum_{I=1,\,NB2} \left(\theta^2_{n^\varphi}\right)_I$$

où k est une rigidité de torsion dont la détermination de la valeur sera discutée plus loin. On suppose que cette rigidité reste constante et ne subit ni variation virtuelle ni variation incrémentale.

On suppose l'existence du potentiel :

$$\pi_{n^j} = \frac{1}{2} k \sum_{I=1,\,NB2} \left((\Theta . n^\varphi)(\Theta . n^\varphi)\right)_I$$

que l'on peut réécrire sous une forme plus élégante :

$$\pi_{n^\varphi} = \frac{1}{2} k \sum_{I=1,\,NB2} \left(\Theta[n^\varphi \otimes n^\varphi]\Theta\right)_I$$

En exploitant la propriété d'orthogonalité $\Lambda^{-1} = \Lambda^T$ de la matrice de rotation :

$$n^\varphi \otimes n^\varphi = n^\varphi {n^\varphi}^T = (\Lambda\, n)(\Lambda\, n)^T = \mathbf{n n^T} = n \otimes n$$

Cette propriété sera exploitée dans la double linéarisation de l'énergie potentielle.

On réécrit le potentiel sous la forme :

$$\pi_{n^\varphi} = \frac{1}{2} k \sum_{I=1,\,NB2} \left(\Theta[n \otimes n]\Theta\right)_I$$

La première linéarisation de π_{n^φ}, permet d'obtenir la variation virtuelle :

$$\begin{aligned}\delta\pi_{n^\varphi} &= \frac{1}{2} k \sum_{I=1,\,NB2} \left(\delta\Theta[n \otimes n]\Theta + \Theta[n \otimes n]\delta\Theta\right)_I\\ &= k \sum_{I=1,\,NB2} \left(\delta\Theta[n \otimes n]\Theta\right)_I\end{aligned}$$

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

Il est nécessaire d'exprimer cette forme-là en fonction des fonction tests de rotations retenues dans la forme variationnelle.

$$\delta\Theta = T^{-1}(\Theta)\delta w$$

avec l'expression de la matrice inverse de l'opérateur différentiel de rotation :

$$T^{-1}(\Theta) = \frac{\frac{\theta}{2}}{\tan\frac{\theta}{2}}[I] - \frac{1}{2}[\Theta\times] + \frac{1}{\theta^2}\left[1 - \frac{\frac{\theta}{2}}{\tan\frac{\theta}{2}}\right][\Theta\otimes\Theta]$$

D'où la forme finale du travail virtuel qui permet de déduire le vecteur des forces intérieures :

$$\delta\pi_{n^\varphi} = k\sum_{I=1,NB2}(\delta w\, T^{-T}(\Theta)[n\otimes n]\Theta)_I$$

On procède à la deuxième linéarisation de $\delta\pi_{n^\varphi}$:

$$\Delta\delta\pi_{n^\varphi} = k\sum_{I=1,NB2}\left(\delta w.\left(T^{-T}(\Theta)[n\otimes n]\Delta\Theta + \Delta T^{-T}(\Theta)[n\otimes n]\Theta\right)\right)_I$$

avec le choix particulier des degrés de liberté de rotation $\Delta\delta w = 0$, et du fait que la normale initiale ne " bouge pas " pendant les itérations $\Delta n = 0$.

L'expression de l'opérateur tangent qui donne naissance aux termes correspondant aux degrés de liberté de rotation autour de la transformée de la normale de matrice tangente est la suivante :

$$\Delta\delta\pi_{n^\varphi} = k\sum_{I=1,NB2}\left(\delta w.\left(T^{-T}(\Theta)[n\otimes n]T^{-1}(\Theta)\right)\Delta w\right)_I + k\sum_{I=1,NB2}\left(\delta w.\left(\Delta T^{-T}(\Theta)[n\otimes n]\Theta\right)\right)_I$$

Dans cette relation, le dernier terme est un terme différentiel dû à la relation non linéaire entre les paramètres de rotation. Sa linéarisation est lourde à effectuer et sa contribution sera négligée dans l'expression de l'opérateur tangent.

Avec la propriété : $n^\varphi\otimes n^\varphi = n\otimes n$, nous donnons l'expression finale :

$$\Delta\delta\pi_{n^\varphi} \equiv k\sum_{I=1,NB2}\left(\delta w.\left(T^{-T}(\Theta)[n^\varphi\otimes n^\varphi]T^{-1}(\Theta)\right)\Delta w\right)_I$$

On note la contribution de l'opérateur $T^{-1}(\Theta)\neq[I]$ dans la matrice tangente de rigidité.

5.3 Remarque

L'énergie potentielle apportée par les termes de rotation autour de la transformée de la normale est non nulle même pour un mouvement de rotation rigide. Cette énergie ne correspond pas à une déformation. C'est pour cette raison qu'elle doit être non significative. La valeur par défaut du `COEF_RIGI_DRZ` doit garantir cela.

5.4 Cas limite de l'analyse géométriquement linéaire

Dans le cas de l'analyse géométriquement linéaire, la configuration initiale et la configuration déformée sont confondues ce qui nous conduit à confondre la normale initiale n avec sa transformée n^φ :

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

$$n^{\varphi} \approx n$$

Les rotations deviennent petites dans ce cas et l'opérateur de grande rotation devient:

$$\Lambda = \exp[\Theta\times] \approx [I] + [\Theta\times]$$

L'opérateur différentiel des rotations devient :

$$T(\Theta) \approx [I]$$

et les paramètres de rotations deviennent confondus :

$$\Delta w \approx \Delta\Theta \quad \text{et} \quad \delta w \approx \delta\Theta$$

Toutes ces approximations introduites dans le travail virtuel conduisent à sa simplification :

$$\delta\pi_{n^{\varphi}} \approx k \sum_{k=1,NB2} (\delta\Theta[n\otimes n]\Theta)_I$$

Les mêmes approximations introduites dans la partie différentielle du travail virtuel conduisent aussi à sa simplification :

$$\Delta\delta\pi_{n^{\varphi}} \equiv k \sum_{k=1,NB2} (\delta\Theta[n\otimes n]\Delta\Theta)_I$$

Les deux dernières équations sont celles de l'analyse géométriquement linéaire. Elles montrent que les contributions dans le vecteur des forces intérieures et dans la matrice tangente de rigidité recouvrent bien le cas limite de [R3.07.04].

5.5 Détermination du coefficient k

Le coefficient k est calculé à chaque itération de chaque pas de temps. A chaque itération de Newton de chaque pas de temps, on construit aux $NB2$ nœuds la matrice de passage

$$\bar{\Lambda}_I = \left[t_1^{\varphi} : t_2^{\varphi} : n^{\varphi}\right]_I ; I = 1, NB2$$

qui permet de passer du vecteur δw_I , vecteur de rotation itérative au nœud exprimé dans le repère global $\left[e_1 : e_2 : e_3\right]$ au vecteur $\delta\tilde{w}_I$ exprimé dans repère local $\left[t_1^{\varphi} : t_2^{\varphi} : n^{\varphi}\right]_I$:

$$\delta\tilde{w}_I = \bar{\Lambda}_I \delta w_I ; I = 1, NB2$$

On peut construire en chaque nœud, la matrice $\left[K_T^e\right]_I$ de taille 3×3

$$[K_T^e]_I = \bar{\Lambda}_I{}^T [K_T^e]_I \bar{\Lambda}_I$$

Cette matrice représente la rigidité de rotation dans le repère local. Sa structure, non symétrique, est :

$$[K_T^e]_I = \begin{bmatrix} k_{t_1^{\varphi} t_1^{\varphi}} & k_{t_1^{\varphi} t_2^{\varphi}} & 0 \\ k_{t_2^{\varphi} t_1^{\varphi}} & k_{t_2^{\varphi} t_2^{\varphi}} & 0 \\ k_{n^{\varphi} t_1^{\varphi}} & k_{n^{\varphi} t_2^{\varphi}} & 0 \end{bmatrix}_I$$

Le coefficient k est alors calculé suivant la relation :

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

$$k = COEF_RIGI_DRZ \times KMIN$$

où $COEF_RIGI_DRZ$ est un coefficient introduit comme une caractéristique mécanique de coque par l'utilisateur. En analyse linéaire classique des coques ou des plaques, ce coefficient est choisi entre 0.001 et 0.000001 . Par défaut il vaut 0.00001 . Dans le cas de grandes rotations calculées avec des grands pas de charge, on conseille d'utiliser la valeur 0.001 .

$KMIN$ est le minimum des termes non nuls de rotation sur la diagonale de $\tilde{K}_T^e$.

$$KMIN = \underset{I=1,NB2}{MIN} \left(k_{t_1^{\varphi} t_1^{\varphi}}, k_{t_2^{\varphi} t_2^{\varphi}} \right)_I$$

Remarque :

Il serait sans doute plus rigoureux de calculer k à la première itération du premier pas de temps et de stocker cette valeur comme une information invariante pendant les itérations et les pas de temps suivants. L'expérience montre que cette façon de procéder n'est souvent pas optimale dans la mesure où les valeurs des coefficients des matrices de rigidité peuvent évoluer de façon importante au cours d'un calcul en grands déplacements. La valeur de k déterminée initialement peut alors devenir trop petite et la matrice de rigidité finir par être singulière.

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

6 Flambement linéaire

Le flambement linéaire se présente comme un cas particulier du problème non linéaire géométrique. Il est basé sur l'hypothèse d'une dépendance linéaire des champs de déplacements, de déformations et de contraintes par rapport au niveau de charge, et ce, avant que la charge critique ne soit atteinte.

Dans une formulation lagrangienne totale, on rappelle que l'équilibre linéarisé peut s'écrire sous la forme variationnelle :

$$\Delta \delta \pi_{\text{int}} - \Delta \delta \pi_{ext} = \delta \pi_{ext} - \delta \pi_{\text{int}}$$

soit sous forme matricielle après discrétisation :

$$\delta u \cdot K_T \Delta u = \delta u \cdot (\lambda f - r)$$

où la dépendance de la matrice tangente de rigidité K_T est non linéaire par rapport au vecteur de déplacement nodal généralisé $p = \underset{e=1,Nel}{U} p^e$.

Si nous supposons la dépendance linéaire du déplacement par rapport au niveau de charge, on peut écrire :

$$u = \lambda u_0$$

où u_0 est la solution obtenue suite à une analyse linéaire pour $\lambda = 1$ par :

$$K_0 u_0 = f$$

où K_0 est la matrice tangente de rigidité initiale. On peut alors développer la matrice tangente de rigidité d'une manière linéaire par rapport au niveau de charge :

$$\underset{e=1,Nel}{U} K_T^e = K_0^e + \lambda \left(K_u^e + K_\sigma^e \right) + \ldots .$$

où K_u^e est la matrice des déplacements initiaux dépendant de p_0^e , traditionnellement négligée dans *Code_Aster,* et K_σ^e la matrice des contraintes initiales dépendant du tenseur global des contraintes de Piola-Kirchhoff de deuxième espèce $\left[S_0\right]$ et du vecteur local $\tilde{S}_0$. Ces contraintes sont volontairement confondues avec les contraintes de Cauchy. Elles sont obtenues par un post-traitement de l'analyse linéaire.

Pour la partie rotation de p^e , l'hypothèse de linéarité des déformations en fonction du niveau de charge se traduit par l'égalité de :

$$n_I^\varphi = n_I$$

qui nous conduit à confondre les normales initiales n_I avec leurs transformées n_I^φ .

La matrice des contraintes initiales K_σ^e représente la partie constante en λ de la partie géométrique de la matrice tangente de rigidité. Elle est évaluée au point p_0^e et $\lambda = 1$ avec une transformée de la normale remplacée par la normale initiale :

$$K_\sigma^e = K_\sigma^{e\,\text{classique}} + K_\sigma^{e\,\text{non classique}}$$

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

avec la partie classique de la partie géométrique (voir [bib4] tome 1 p. 141) :

$$K_{\sigma}^{e_{classique}} = \int_{\Omega_\zeta} \left[\tilde{J}^{-1} \left[\frac{\partial N}{\partial \sigma} \right]_2 \right]^T \bar{S} \left[\tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_2 \right] \det J \, d\xi_1 \, d\xi_2 \, d\xi_3$$

où la deuxième matrice des dérivées des fonctions de forme devient :

$$\cdots \begin{bmatrix} N^{(1)}_{I,\xi_1} & 0 & 0 \\ N^{(1)}_{I,\xi_2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & N^{(1)}_{I,\xi_1} & 0 \\ 0 & N^{(1)}_{I,\xi_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & N^{(1)}_{I,\xi_1} \\ 0 & 0 & N^{(1)}_{I,\xi_2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{h}{2} \end{bmatrix} \begin{bmatrix} 0 & \xi_3 N^{(2)}_{I,\xi_1} n_z & -\xi_3 N^{(2)}_{I,\xi_1} n_y \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} n_z & -\xi_3 N^{(2)}_{I,\xi_2} n_y \\ 0 & N^{(2)}_{I} n_z & -N^{(2)}_{I} n_y \\ -\xi_3 N^{(2)}_{I,\xi_1} n_z & 0 & \xi_3 N^{(2)}_{I,\xi_1} n_x \\ -\xi_3 N^{(2)}_{I,\xi_2} n_z & 0 & \xi_3 N^{(2)}_{I,\xi_2} n_x \\ -N^{(2)}_{I} n_z & 0 & N^{(2)}_{I} n_x \\ \xi_3 N^{(2)}_{I,\xi_1} n_y & -\xi_3 N^{(2)}_{I,\xi_1} n_x & 0 \\ \xi_3 N^{(2)}_{I,\xi_2} n_y & -\xi_3 N^{(2)}_{I,\xi_2} n_x & 0 \\ N^{(2)}_{I} n_y & -N^{(2)}_{I} n_x & 0 \end{bmatrix} \cdots I = 1, NB1$$

$$\left| \frac{h}{2} \right| \begin{bmatrix} 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n_z & -\xi_3 N^{(2)}_{NB2,\xi_1} n_y \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n_z & -\xi_3 N^{(2)}_{NB2,\xi_2} n_y \\ 0 & N^{(2)}_{NB2} n_z & -N^{(2)}_{1} n_y \\ -\xi_3 N^{(2)}_{NB2,\xi_1} n_z & 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n_x \\ -\xi_3 N^{(2)}_{NB2,\xi_2} n_z & 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n_x \\ -N^{(2)}_{NB2} n_z & 0 & N^{(2)}_{NB2} n_x \\ \xi_3 N^{(2)}_{NB2,\xi_1} n_y & -\xi_3 N^{(2)}_{NB2,\xi_1} n_x & 0 \\ \xi_3 N^{(2)}_{NB2,\xi_2} n_y & -\xi_3 N^{(2)}_{NB2,\xi_2} n_x & 0 \\ N^{(2)}_{NB2} n_y & -N^{(2)}_{NB2} n_x & 0 \end{bmatrix}$$

$$\left[\frac{\partial N}{\partial \xi} \right]_2 = ¿[¿][¿][]$$

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

et le tenseur généralisé des contraintes globales :

$$\overset{9\times9}{\bar{S}}=\begin{bmatrix} \overset{3\times3}{[S]} & 0 & 0 \\ 0 & [S] & 0 \\ 0 & 0 & [S] \end{bmatrix}$$

La partie non classique de la partie géométrique représente les termes découplés de rotation, non symétriques. Puisque l'algorithme actuel de résolution du problème aux valeurs propres [R5.06.01] $\left[K_0+\lambda\left(K_u+K_\sigma\right)\right]F=0$ (λ étant le niveau de charge critique) ne considère que des matrices symétriques, on rend symétrique, en divisant par deux la somme avec sa transposée, la matrice :

$$K_\sigma^{e\,\text{non classique}}(I,I)=\frac{1}{2}\left(\left[z_I\times\right]\left[n_I\times\right]\right)$$

où n_I est la normale au nœud I et z_I un vecteur 3×3 au nœud $I=1,NB2$ du vecteur nodal $3\times(3\times NB2)Z_I$:

$$Z_I=\begin{Bmatrix} . \\ . \\ . \\ z_I \\ . \\ . \\ . \end{Bmatrix}_{I=1,NB2}$$

Le vecteur nodal Z_I est similaire à un vecteur de force interne et son expression est :

$$Z_I=\int_{\Omega_z}\tilde{B}_3^T\,\tilde{S}\det J\,d\xi_1\,d\xi_2\,d\xi_3$$

avec l'opérateur de la variation itérative des déformations virtuelles (troisième opérateur des déformations) :

$$\tilde{B}_3=\mathbf{HQ}\;\tilde{\boldsymbol{J}}^{-1}\left[\frac{\partial N}{\partial\xi}\right]_3$$

qui est mis en évidence par la relation :

$$\int_{\Omega_z}\tilde{B}_3^T\,\tilde{S}\det J\,d\xi_1\,d\xi_2\,d\xi_3=\int_\Omega\Delta\,\delta\,\tilde{E}_{\text{non classique}}\cdot\tilde{S}\,d\Omega$$

Remarque :

Pour l'intégration numérique dans l'épaisseur des différents termes de rigidité, nous retenons un schéma de Gauss à deux points tout comme en élasticité pour les coques linéaires géométriques [R3.07.04].

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

7 Implantation des éléments de coque dans *Code_Aster*

7.1 Description

Ces éléments (de noms MEC3QU9H et MEC3TR7H) s'appuient sur des mailles QUAD9 et TRIA7 qui sont de géométrie courbes [R3.07.04].

7.2 Utilisation

Ces éléments s'utilisent de la façon suivante :

AFFE_MODELE (MODELISATION : 'COQUE_3D' ...) pour le triangle et le quadrangle.

On fait appel à la routine INI080 pour les calculs standard d'intégration numérique.

```
AFFE_CARA_ELEM ( COQUE:(EPAISSEUR:'EP'
                  ANGL_REP : ( ' ' ' ' )
                 COEF_RIGI_DRZ : 'CTOR')
```

pour introduire les caractéristiques de coque.

ELAS : (E :young NU : ν ALPHA : α .. RHO : ρ ..)
Pour un comportement thermo-élastique isotrope homogène dans l'épaisseur on utilise le mot-clé ELAS dans DEFI_MATERIAU où l'on définit les coefficients E , module d'Young, ν , coefficient de Poisson, α , coefficient de dilatation thermique et ρ la masse volumique.

AFFE_CHAR_MECA (DDL_IMPO : (
DX :.. DY :.. DZ :.. DRX :.. DRY :.. DRZ :.. degré de liberté de plaque dans le repère global.

FORCE_COQUE : (FX :.. FY :.. FZ :.. MX :.. MY :.. MZ :..)
Il s'agit des efforts surfaciques sur des éléments de plaque. Ces efforts peuvent être donnés dans le repère global ou dans le repère utilisateur défini par ANGL_REP.

FORCE_NODALE : (FX :.. FY :.. FZ :.. MX :.. MY :.. MZ :..)
Il s'agit des efforts de coque dans le repère global.

7.3 Calcul en " élasticité " non linéaire géométrique

Le calcul impose les instructions utilisateur suivantes :

```
COMPORTEMENT : (RELATION     : 'elas'
                COQUE_NCOU : 1 (ou plus)
                DEFORMATION    : 'GROT_GDEP')
```

L'intégration numérique dans l'épaisseur est basée sur une approche multi-couches avec 3 points d'intégration par couche. Il s'agit de l'approche actuellement utilisée en non linéaire matériel [R3.07.04]. Les options de post-traitement SIGM_ELNO des contraintes et VARI_ELNO des variables internes (ici nulles) sont par défaut activées à la convergence de chaque pas archivé.

7.4 Implantation

Les options FULL_MECA, RIGI_MECA_TANG, et RAPH_MECA sont déjà actives dans les catalogues élémentaires mec3qu9h.cata et mec3tr7h.cata pour la non-linéarité matérielle. Elles dirigent le calcul vers /fort/te0414.f, ensuite vers /fort/vdxnlr.f pour calculer et stocker, entre autres,

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

la matrice tangente de rigidité symétrique dans l'adresse correspondante au mode MMATUUR PMATUUR.

Pour l'analyse non linéaire géométrique, le calcul de la matrice tangente de rigidité est dirigé vers la nouvelle routine VDGNLR. Cette matrice n'est pas symétrique et doit être stockée dans l'adresse correspondant au mode MMATUNS PMATUNS.

On définit les deux modes locaux symétrique et non symétrique à la fois, en sortie des catalogues élémentaires. La matrice tangente de rigidité non symétrique en non linéaire géométrique est stockée à l'adresse réservée à une matrice non symétrique. Par contre, s'il s'agit de non linéarité matérielle en petites déformations, toute la matrice tangente de rigidité est stockée à l'adresse correspondant au mode non symétrique. La partie triangulaire inférieure est dupliquée à partir de la partie triangulaire supérieure. Le calcul non linéaire matériel en petites déformations se déroule donc aussi en non symétrique.

7.4.1 Modification du TE0414

Le calcul est dirigé vers /fort/vdgnlr.f quand le type de comportement COMPORTEMENT est vérifié, c'est-à-dire quand le problème est non linéaire géométrique.

7.4.2 Ajout d'une routine VDGNLR

Selon l'option, la routine /fort/vdgnlr.f a pour rôle de :

Options : FULL_MECA et RAPH_MECA :

Calculer les 6 composantes de l'état des contraintes locales de Cauchy (confondu avec l'état des contraintes de Piola-Kirchhoff de deuxième espèce) aux points d'intégration numérique normale et le vecteur nodal des forces internes. Ils sont stockés dans les modes locaux ECONTPG PCONTPR et MVECTUR PVECTUR respectivement. En remarque supplémentaire, on notera que SIEF_ELGA/SIGM_ELGA après la résolution est de type PK2 tandis que SIEF_ELNO/SIGM_ELNO en post-traitement est de type Cauchy.

Options : FULL_MECA et RIGI_MECA_TANG :

Calculer et stocker la matrice tangente de rigidité non symétrique dans le mode MMATUNS PMATUNS.

7.5 Calcul en flambement linéaire

L'option RIGI_MECA_GE, inactive jusqu'à présent, est activée dans les catalogues élémentaires mec3qu9h.cata et mec3tr7h.cata.

Le nouveau TE0402 est dédié au calcul de la matrice de rigidité géométrique due aux contraintes initiales pour le flambement d'Euler. On récupère les états plans des contraintes locales de Cauchy (composante S_{nn} nulle) aux points d'intégration numérique normale de Gauss. Ces états de contraintes doivent être obtenus par post-traitement avec l'option de calcul SIEF_ELGA suite à une analyse linéaire (mode ECONTPG PCONTRR).

En analyse de flambement d'Euler, les contraintes $\left[\sigma\right]$ de Cauchy peuvent être confondues avec les contraintes $\left[S\right]$ de Piola-Kirchhoff de deuxième espèce. C'est pour cela que nous garderons la notation $\left[S\right]$.

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

La matrice de rigidité des contraintes initiales peut être décomposée en une partie classique symétrique et une partie non classique non symétrique. La première est calculée en fonction du tenseur global des contraintes 3×3 , contrairement à la deuxième qui, elle, est calculée en fonction du vecteur des contraintes locales 5×1 .

Puisque l'algorithme actuel de résolution du problème aux valeurs propres [R5.06.01] ne considère que des matrices symétriques, nous forçons la symétrie de la partie non classique de la matrice géométrique avant de stocker la partie triangulaire supérieure de toute la matrice dans le mode MMATUUR PMATUUR.

8 Conclusion

La formulation que nous venons de décrire s'applique aux calculs de structures minces courbes en grands déplacements, dont le rapport épaisseur sur longueur caractéristique est inférieur à $1/10$. Elle vient en complément direct des éléments finis décrits dans la précédente documentation de référence [R3.07.04] et utilisés dans le cadre des petits déplacements et déformations. Ils s'appuient sur les mêmes mailles quadrangle et triangle.

Leur formulation repose sur une approche de milieu continu 3D dans laquelle on introduit une cinématique de coque de type Hencky-Mindlin-Naghdi, en contraintes planes, dans la formulation faible de l'équilibre. La mesure des déformations retenue est celle de Green-Lagrange, énergétiquement conjuguée aux contraintes de Piola-Kirchhoff de deuxième espèce. La formulation de l'équilibre est donc lagrangienne totale. La distorsion transverse est traitée de la même manière qu'en [R3.07.04]. Les rotations doivent rester inférieures à 2π du fait du choix de mise à jour des grandes rotations implanté dans *Code_Aster* pour lequel il y a non bijection entre le vecteur de rotation totale et la matrice orthogonale de rotation.

Du fait de la singularité de la matrice tangente de rigidité par rapport à la composante de la rotation autour de la transformée de la normale, on définit une énergie de déformation fictive associée à cette rotation. A cette rotation, on associe une rigidité de torsion constante. Les efforts intérieurs associés à cette énergie potentielle sont pris en compte. Cette énergie potentielle, non nulle, ne correspond pas à une déformation physique. Il faut donc qu'elle reste négligeable, ce que l'utilisateur peut contrôler en imposant une valeur du COEF_RIGI_DRZ valant de 10^{-3} à 10^{-5} .

Pour le post-traitement des contraintes, on se limite au cadre des petites déformations. On a alors pu établir l'identité entre le tenseur des contraintes de Piola-Kirchhoff observé dans la géométrie initiale et le tenseur des contraintes de Cauchy dans la géométrie déformée. En outre, l'état des contraintes étant plan pour le tenseur de Piola-Kirchhoff, on retrouve cette propriété pour l'état de contraintes de Cauchy. Il est à noter que dans des contextes plus généraux, cette propriété n'est pas conservée.

Le flambement linéaire est traité comme un cas particulier du problème non linéaire géométrique. Il repose sur l'hypothèse d'une dépendance linéaire des champs de déplacements, de déformations et de contraintes par rapport au niveau de charge, avant la charge critique. Il en résulte que l'on peut développer la matrice tangente de rigidité linéairement par rapport au niveau de charge. On retrouve alors la partie géométrique des matrices du non linéaire géométrique général obtenue en identifiant la déformée de la normale à la surface moyenne et la normale initiale, ce qui est cohérent avec la linéarité des déformations en fonction du niveau de charge.

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

9 Bibliographie

1) Al Mikdad M., " Statique et Dynamique des Poutres en Grandes Rotations et Résolution des Problèmes d'Instabilité Non Linéaire ", thèse de doctorat, Université de Technologie de Compiègne, 1998.

2) Bathe K.J., " Finite Element Proceedings in Engineering Analysis ", Prentice Hall, New Jersey, 1982.

3) Cardonna A., " An Integrated Approach to Mechanism Analysis ", thèse de doctorat, Université de Liège, 1989.

4) Crisfield M.A., " Non-linear Finite Element Analysis of Solids and Structures ", Volume 1 : Essentials, John Wiley, Chichester, 1994.

5) Crisfield M.A., " Non-linear Finite Element Analysis of Solids and Structures ", Volume 2 : Advanced topics, John Wiley, Chichester, 1994.

6) Jetteur Ph., " Cinématique Non Linéaire des Coques ", rapport SAMTECH, contrat PP/GC-134/96, 1998.

7) Simo J.C., " The (symmetric) Hessian for Geometrically Nonlinear Models in Solid Mechanics : Intrinsic Definition and Geometric Interpretation", Comp. Methods Appl. Mech. 96 : 189-200, 1992.

8) Vautier I., " Mise en œuvre de STAT_NON_LINE ", manuel de développement *Code_Aster* [D9.05.01].

9) Massin P., " Eléments de plaque DKT, DST, DKQ, DSQ et Q4g ", Manuel de Référence du *Code_Aster* [R3.07.03].

10) Massin P., Laulusa A., Al Mikdad M., Bui D., Voldoire F., " Modélisation Numérique des Coques Volumiques ", Manuel de Référence du *Code_Aster* [R3.07.04].

11) Lorentz E., " Relation de Comportement Elastique Non Linéaire ", Manuel de Référence du *Code_Aster* [R5.03.20].

12) Jacquart G., " Méthode de Ritz en dynamique linéaire et non linéaire ", Manuel de Référence du *Code_Aster* [R5.06.01].

13) Aufaure M., " Modélisation Statique et Dynamique des Poutres en Grandes Rotations ", Manuel de Référence du *Code_Aster* [R5.03.40].

10 Description des versions du document

Version Aster	Auteur(s) Organisme(s)	Description des modifications
5	P.Massin, M. AL MIKDAD EDF-R&D/MMN	Texte initial
7.4	X.Desroches	Mise à jour : modifications mineures

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

Annexe 1: Organigramme du calcul en flambement linéaire

Repères locaux aux $NB2$ **nœuds** $\left[t_1 : t_2 : n\right]_I$

Boucle sur les points d'intégration numérique normale de Gauss

- récupération du vecteur des contraintes locales $\tilde{S}=\begin{pmatrix}\tilde{S}_{t_1t_1}\\ \tilde{S}_{t_2t_2}\\ \tilde{\tau}_{t_1t_2}\\ \\ \tilde{\tau}_{t_1n}\\ \tilde{\tau}_{t_2n}\end{pmatrix}=\begin{pmatrix}\tilde{S}_{t_1t_1}\\ \tilde{S}_{t_2t_2}\\ \sqrt{2}\,\tilde{S}_{t_1t_2}\\ \\ \sqrt{2}\,\tilde{S}_{t_1n}\\ \sqrt{2}\,\tilde{S}_{t_2n}\end{pmatrix}$

  à partir des 6 composantes tenseurs stockées dans le mode `PCONTRR` $\begin{pmatrix}\tilde{S}_{t_1t_1}\\ \tilde{S}_{t_2t_2}\\ 0\\ \tilde{S}_{t_1t_2}\\ \tilde{S}_{t_1n}\\ \tilde{S}_{t_2n}\end{pmatrix}$

- formation du tenseur symétrique 3×3 des contraintes locales $\left[\tilde{S}\right]$
- construction de la matrice de transformation $\mathbf{P}\left(\xi_{1,}\xi_{2,}\xi_{3}\right)=\begin{bmatrix}\mathbf{t_1^T}\left(\xi_{1,}\xi_{2,}\xi_{3}\right)\\ \mathbf{t_2^T}\left(\xi_{1,}\xi_{2,}\xi_{3}\right)\\ \mathbf{t_3^T}\left(\xi_{1,}\xi_{2,}\xi_{3}\right)\end{bmatrix}$ où $\mathbf{t_3}\left(\xi_{1,}\xi_{2,}\xi_{3}\right)=n_1\left(\xi_2\right)$
- calcul du tenseur symétrique 3×3 des contraintes globales $\left[S\right]=P^T\left[\tilde{S}\right]P$
- pour le terme non classique, calcul de $\mathbf{HQ}=\begin{bmatrix}\overset{3\times9}{\left[HSFM\right]}\\ \overset{2\times9}{\left[HSS\right]}\end{bmatrix}$

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

HQ=

$$\begin{bmatrix} (t_1(1))^2 & (t_1(2))^2 & (t_1(3))^2 & t_1(1)t_1(2) & t_1(2)t_1(3) & t_1(3)t_1(1) \\ (t_2(1))^2 & (t_2(2))^2 & (t_2(3))^2 & t_2(1)t_2(2) & t_2(2)t_2(3) & t_2(3)t_2(1) \\ (t_3(1))^2 & (t_3(2))^2 & (t_3(3))^2 & t_3(1)t_3(2) & t_3(2)t_3(3) & t_3(3)t_3(1) \\ 2t_2(1)t_3(1) & 2t_2(2)t_3(2) & 2t_2(3)t_3(3) & t_2(1)t_3(2)+t_3(1)t_2(2) & t_2(2)t_3(3)+t_3(2)t_2(3) & t_2(3)t_3(1)+t_3(3)t_3(1 \\ 2t_3(1)t_1(1) & 2t_3(2)t_1(2) & 2t_3(3)t_1(3) & t_3(1)t_1(2)+t_1(1)t_3(2) & t_3(2)t_1(3)+t_1(2)t_3(3) & t_3(3)t_1(1)+t_1(3)t_3(1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- calcul de la matrice jacobienne inverse J^{-1} et du déterminant $\det J$
- calcul de $\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_3$ avec :

$$\tilde{J}^{-1}=\begin{bmatrix} J^{-1} & 0 & 0 \\ 0 & J^{-1} & 0 \\ 0 & 0 & J^{-1} \end{bmatrix};\left[\frac{\partial N}{\partial \xi}\right]_3=\begin{bmatrix} \cdots\frac{h}{2}\begin{bmatrix} \xi_3 N^{(2)}_{I,\xi_1} & 0 & 0 \\ \xi_3 N^{(2)}_{I,\xi_2} & 0 & 0 \\ N^{(2)}_I & 0 & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_1} & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} & 0 \\ 0 & N^{(2)}_I & 0 \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_1} \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_2} \\ 0 & 0 & N^{(2)}_I \end{bmatrix}\cdots I=1,NB2 \end{bmatrix}$$

- calcul du troisième opérateur des déformations $\tilde{B}_3=\boldsymbol{HQ}\,\tilde{\boldsymbol{J}}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_3$
- calcul et intégration numérique $Z_I=\int_{\Omega_\zeta}\tilde{B}_3^T\,\tilde{S}\det J\,d\xi_1\,d\xi_2\,d\xi_3$
- calcul du tenseur généralisé des contraintes **globales**

$$\overset{9\times 9}{\tilde{S}}=\begin{bmatrix} \overset{3\times 3}{[S]} & 0 & 0 \\ 0 & [S] & 0 \\ 0 & 0 & [S] \end{bmatrix}$$

- calcul de $\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2$ avec :

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

$$\cdots \begin{bmatrix} \begin{bmatrix} N^{(1)}_{I,\xi_1} & 0 & 0 \\ N^{(1)}_{I,\xi_2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & N^{(1)}_{I,\xi_1} & 0 \\ 0 & N^{(1)}_{I,\xi_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & N^{(1)}_{I,\xi_1} \\ 0 & 0 & N^{(1)}_{I,\xi_2} \\ 0 & 0 & 0 \end{bmatrix} \frac{h}{2} \begin{bmatrix} 0 & \xi_3 N^{(2)}_{I,\xi_1} n_z & -\xi_3 N^{(2)}_{I,\xi_1} n_y \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} n_z & -\xi_3 N^{(2)}_{I,\xi_2} n_y \\ 0 & N^{(2)}_1 n_z & -N^{(2)}_1 n_y \\ -x_3 N^{(2)}_{I,\xi_1} n_z & 0 & \xi_3 N^{(2)}_{I,\xi_1} n_x \\ -\xi_3 N^{(2)}_{I,\xi_2} n_z & 0 & \xi_3 N^{(2)}_{I,\xi_2} n_x \\ -N^{(2)}_1 n_z & 0 & N^{(2)}_1 n_x \\ \xi_3 N^{(2)}_{I,\xi_1} n_y & -\xi_3 N^{(2)}_{I,\xi_1} n_x & 0 \\ \xi_3 N^{(2)}_{I,\xi_2} n_y & -\xi_3 N^{(2)}_{I,\xi_2} n_x & 0 \\ N^{(2)}_1 n_y & -N^{(2)}_1 n_x & 0 \end{bmatrix} \end{bmatrix} \cdots I=1, NB1$$

$$\left| \frac{h}{2} \begin{bmatrix} 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n_z & -\xi_3 N^{(2)}_{NB2,\xi_1} n_y \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n_z & -\xi_3 N^{(2)}_{NB2,\xi_2} n_y \\ 0 & N^{(2)}_{NB2} n_z & -N^{(2)}_1 n_y \\ -\xi_3 N^{(2)}_{NB2,\xi_1} n_z & 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n_x \\ -\xi_3 N^{(2)}_{NB2,\xi_2} n_z & 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n_x \\ -N^{(2)}_{NB2} n_z & 0 & N^{(2)}_{NB2} n_x \\ \xi_3 N^{(2)}_{NB2,\xi_1} n_y & -\xi_3 N^{(2)}_{NB2,\xi_1} n_x & 0 \\ \xi_3 N^{(2)}_{NB2,\xi_2} n_y & -\xi_3 N^{(2)}_{NB2,\xi_2} n_x & 0 \\ N^{(2)}_{NB2} n_y & -N^{(2)}_{NB2} n_x & 0 \end{bmatrix} \right|$$

$$\left[\frac{\partial N}{\partial \xi} \right]_2 = ¿[¿][¿][\,]$$

- calcul et intégration numérique de la matrice de rigidité géométrique classique

$$K^{e}_{\sigma\,\text{classique}} = \int_{\Omega_\zeta} \left[\tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_2 \right]^T \bar{S} \left[\tilde{J}^{-1} \left[\frac{\partial N}{\partial \xi} \right]_2 \right] \det J \, d\xi_1 \, d\xi_2 \, d\xi_3$$

Fin boucle sur les points d'intégration

Boucle sur tous les nœuds de Lagrange avec distinction du super nœud

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

- calcul de $\left[z_I\times\right]$ sachant que $Z_I=\left\{\begin{array}{c}\cdot\\\cdot\\\cdot\\z_I\\\cdot\\\cdot\\\cdot\\I=1,NB2\end{array}\right\}$
- calcul du vecteur normale n_I et de sa matrice anti-symétrique $\left[n_I\times\right]$
- calcul de la matrice de rigidité géométrique non classique

$$K^{e}_{\sigma\,\text{non classique}}{}^{3\times3}(I,I)=\left[z_I\times\right]\left[n_I\times\right]I=1,NB2$$

- rajout de $K^{e}_{\sigma\,\text{non classique}}{}^{3\times3}(I,I)$ avec distinction du super nœud

Fin boucle sur les nœuds

Stockage de la partie triangulaire supérieure de K^e_σ

FIN

Annexe 2: Organigramme du calcul non linéaire géométrique

Repères locaux aux NB2 nœuds $\left[t_1:t_2:n\right]_I$

Début Boucle JN sur les NB2 nœuds

IF JN $\leq$ NB1

- récupération du vecteur de déplacement total déjà mis à jour par MAJOUR :

$$u_I(ii)=ZR(IDEPLP+IDEPLM-1+6\times(JN-1)+ii);ii=1,3;(JN=1,NB1)$$

- récupération du vecteur de rotation totale déjà mis à jour par MAJOUR

$$\theta_I(ii)=ZR(IDEPLP-1+6\times(JN-1)+ii+3);ii=1,3;(JN=1,NB1)$$

ELSE JN

- récupération du vecteur de rotation totale

$$\theta_I(ii)=ZR(IDEPLP-1+6\times NB1+ii);ii=1,3;(JN=NB2)$$

END IF JN

- calcul de la matrice de rotation $\Lambda_I=\exp\left[\theta_I\times\right]$
- transformée de la normale $n_I^{\varphi}=\Lambda_I n_I$

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

Fin Boucle sur les $NB2$ nœuds

calcul de $p^e = \begin{matrix} \vdots \\ \begin{pmatrix} u \\ v \\ w \\ n_x^{\varphi} - n_x \\ n_y^{\varphi} - n_y \\ n_z^{\varphi} - n_z \end{pmatrix}_I \\ \vdots \\ I = 1, NB1 \\ \vdots \\ \begin{pmatrix} n_x^{\varphi} - n_x \\ n_y^{\varphi} - n_y \\ n_z^{\varphi} - n_z \end{pmatrix}_{NB2} \end{matrix}$

Début Boucle INTSR sur les points d'intégration réduite normale de Gauss

- construction d'une partie des opérateurs $\tilde{B}_1, \tilde{B}_2$
  aux J = 1, INTSR points d'intégrations pour pouvoir les extrapoler

Fin Boucle INTSR sur les points d'intégration réduite normale de Gauss

Début Boucle INTSN sur les points d'intégration numérique normale de Gauss

- construction de la matrice de transformation :

$$\mathbf{P}(\xi_1, \xi_2, \xi_3) = \begin{bmatrix} \mathbf{t_1^T}(\xi_1, \xi_2, \xi_3) \\ \mathbf{t_2^T}(\xi_1, \xi_2, \xi_3) \\ \mathbf{t_3^T}(\xi_1, \xi_2, \xi_3) \end{bmatrix}$$

  où $t_3(\xi_1, \xi_2, \xi_3) = n(\xi_1, \xi_2)$

- calcul de la matrice jacobienne inverse J^{-1} et du déterminant $\det J$
- calcul de $\tilde{J}^{-1} = \begin{bmatrix} J^{-1} & 0 & 0 \\ 0 & J^{-1} & 0 \\ 0 & 0 & J^{-1} \end{bmatrix}$
- calcul de la deuxième matrice des dérivées des fonctions de forme $\left[\frac{\partial N}{\partial \xi}\right]_1$

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

$$\cdots \begin{bmatrix} \begin{bmatrix} N^{(1)}_{I,\xi_1} & 0 & 0 \\ N^{(1)}_{I,\xi_2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & N^{(1)}_{I,\xi_1} & 0 \\ 0 & N^{(1)}_{I,\xi_2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & N^{(1)}_{I,\xi_1} \\ 0 & 0 & N^{(1)}_{I,\xi_2} \\ 0 & 0 & 0 \end{bmatrix} \frac{h}{2} \begin{bmatrix} \xi_3 N^{(2)}_{I,\xi_1} & 0 & 0 \\ \xi_3 N^{(2)}_{I,\xi_2} & 0 & 0 \\ N^{(2)}_I & 0 & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_1} & 0 \\ 0 & \xi_3 N^{(2)}_{I,\xi_2} & 0 \\ 0 & N^{(2)}_I & 0 \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_1} \\ 0 & 0 & \xi_3 N^{(2)}_{I,\xi_2} \\ 0 & 0 & N^{(2)}_I \end{bmatrix} \end{bmatrix} \ldots I = 1, NB1$$

$$\left| \frac{h}{2} \begin{bmatrix} \xi_3 N^{(2)}_{NB2,\xi_1} & 0 & 0 \\ \xi_3 N^{(2)}_{NB2,\xi_2} & 0 & 0 \\ N^{(2)}_{NB2} & 0 & 0 \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_1} & 0 \\ 0 & \xi_3 N^{(2)}_{NB2,\xi_2} & 0 \\ 0 & N^{(2)}_{NB2} & 0 \\ 0 & 0 & \xi_3 N^{(2)}_{NB2,\xi_1} \\ 0 & 0 & \xi_3 N^{(2)}_{NB2,\xi_2} \\ 0 & 0 & N^{(2)}_{NB2} \end{bmatrix} \right.$$

$$\left[\frac{\partial N}{\partial \xi}\right]_1 = ¿[¿][¿][\,]$$

- calcul de $\frac{\partial u}{\partial x} = \tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_1 p^e$
- calcul de $A\left(\frac{\partial u}{\partial x}\right) = \begin{bmatrix} u_{,x} & 0 & 0 & v_{,x} & 0 & 0 & w_{,x} & 0 & 0 \\ 0 & u_{,y} & 0 & 0 & v_{,y} & 0 & 0 & w_{,y} & 0 \\ 0 & 0 & u_{,z} & 0 & 0 & v_{,z} & 0 & 0 & w_{,z} \\ u_{,y} & u_{,x} & 0 & v_{,y} & v_{,x} & 0 & w_{,y} & w_{,x} & 0 \\ u_{,z} & 0 & u_{,x} & v_{,z} & 0 & v_{,x} & w_{,z} & 0 & w_{,x} \\ 0 & u_{,z} & u_{,y} & 0 & v_{,z} & v_{,y} & 0 & w_{,z} & w_{,y} \end{bmatrix}$$
  - calcul de $H\left[Q + \frac{1}{2} A\left(\frac{\partial u}{\partial x}\right)\right]$

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

$$H=\begin{bmatrix}
(t_1(1))^2 & (t_1(2))^2 & (t_1(3))^2 & t_1(1)t_1(2) & t_1(2)t_1(3) & t_1(3)t_1(1) \\
(t_2(1))^2 & (t_2(2))^2 & (t_2(3))^2 & t_2(1)t_2(2) & t_2(2)t_2(3) & t_2(3)t_2(1) \\
(t_3(1))^2 & (t_3(2))^2 & (t_3(3))^2 & t_3(1)t_3(2) & t_3(2)t_3(3) & t_3(3)t_3(1) \\
2t_2(1)t_3(1) & 2t_2(2)t_3(2) & 2t_2(3)t_3(3) & t_2(1)t_3(2)+t_3(1)t_2(2) & t_2(2)t_3(3)+t_3(2)t_2(3) & t_2(3)t_3(1)+t_3(3)t_3(1) \\
2t_3(1)t_1(1) & 2t_3(2)t_1(2) & 2t_3(3)t_1(3) & t_3(1)t_1(2)+t_1(1)t_3(2) & t_3(2)t_1(3)+t_1(2)t_3(3) & t_3(3)t_1(1)+t_1(3)t_3(1)
\end{bmatrix}$$

- calcul du premier opérateur des déformations

$$\tilde{B}_1=H\left[Q+\frac{1}{2}A\left(\frac{\partial u}{\partial x}\right)\right]\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_1$$

- calcul du vecteur des déformations locales $\tilde{E}=\tilde{B}_1 p^e$
- calcul de la deuxième matrice des dérivées des fonctions de forme $\left[\frac{\partial N}{\partial \xi}\right]_2$

$$\cdots\begin{bmatrix}
N^{(1)}_{I,\xi_1} & 0 & 0 \\
N^{(1)}_{I,\xi_2} & 0 & 0 \\
0 & 0 & 0 \\
0 & N^{(1)}_{I,\xi_1} & 0 \\
0 & N^{(1)}_{I,\xi_2} & 0 \\
0 & 0 & 0 \\
0 & 0 & N^{(1)}_{I,\xi_1} \\
0 & 0 & N^{(1)}_{I,\xi_2} \\
0 & 0 & 0
\end{bmatrix}\frac{h}{2}\begin{bmatrix}
0 & \xi_3 N^{(2)}_{I,\xi_1} n^{\varphi}_z & -\xi_3 N^{(2)}_{I,\xi_1} n^{\varphi}_y \\
0 & \xi_3 N^{(2)}_{I,\xi_2} n^{\varphi}_z & -\xi_3 N^{(2)}_{I,\xi_2} n^{\varphi}_y \\
0 & N^{(2)}_1 n^{\varphi}_z & -N^{(2)}_1 n^{\varphi}_y \\
-\xi_3 N^{(2)}_{I,\xi_1} n^{\varphi}_z & 0 & \xi_3 N^{(2)}_{I,\xi_1} n^{\varphi}_x \\
-\xi_3 N^{(2)}_{I,\xi_2} n^{\varphi}_z & 0 & \xi_3 N^{(2)}_{I,\xi_2} n^{\varphi}_x \\
-N^{(2)}_1 n^{\varphi}_z & 0 & N^{(2)}_1 n^{\varphi}_x \\
\xi_3 N^{(2)}_{I,\xi_1} n^{\varphi}_y & -\xi_3 N^{(2)}_{I,\xi_1} n^{\varphi}_x & 0 \\
\xi_3 N^{(2)}_{I,\xi_2} n^{\varphi}_y & -\xi_3 N^{(2)}_{I,\xi_2} n^{\varphi}_x & 0 \\
N^{(2)}_1 n^{\varphi}_y & -N^{(2)}_1 n^{\varphi}_x & 0
\end{bmatrix} LI=1,NB1$$

$$|\frac{h}{2}\begin{bmatrix}
0 & \xi_3 N^{(2)}_{NB2,\xi_1} n^{\varphi}_z & -\xi_3 N^{(2)}_{NB2,\xi_1} n^{\varphi}_y \\
0 & \xi_3 N^{(2)}_{NB2,\xi_2} n^{\varphi}_z & -\xi_3 N^{(2)}_{NB2,\xi_2} n^{\varphi}_y \\
0 & N^{(2)}_{NB2} n^{\varphi}_z & -N^{(2)}_{NB2} n^{\varphi}_y \\
-\xi_3 N^{(2)}_{NB2,\xi_1} n^{\varphi}_z & 0 & \xi_3 N^{(2)}_{NB2,\xi_1} n^{\varphi}_x \\
-\xi_3 N^{(2)}_{NB2,\xi_2} n^{\varphi}_z & 0 & \xi_3 N^{(2)}_{NB2,\xi_2} n^{\varphi}_x \\
-N^{(2)}_{NB2} n^{\varphi}_z & 0 & N^{(2)}_{NB2} n^{\varphi}_x \\
\xi_3 N^{(2)}_{NB2,\xi_1} n^{\varphi}_y & -\xi_3 N^{(2)}_{NB2,\xi_1} n^{\varphi}_x & 0 \\
\xi_3 N^{(2)}_{NB2,\xi_2} n^{\varphi}_y & -\xi_3 N^{(2)}_{NB2,\xi_2} n^{\varphi}_x & 0 \\
N^{(2)}_{NB2} n^{\varphi}_y & -N^{(2)}_{NB2} n^{\varphi}_x & 0
\end{bmatrix}$$

$$\left[\frac{\partial N}{\partial \xi}\right]_2=¿[¿][¿][\,]$$

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

- calcul du deuxième opérateur des déformations

$$\tilde{B}_2 = H\left(Q + A\left(\frac{\partial u}{\partial x}\right)\right)\tilde{J}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2$$

- calcul et intégration numérique $r^e = \int_{\Omega_\zeta} \tilde{\boldsymbol{B}}_2^T \tilde{\boldsymbol{S}} \det \boldsymbol{J}\, d\xi_1 d\xi_2 d\xi_3$
- calcul de la matrice de comportement $\boldsymbol{D}$
- calcul et intégration numérique $\boldsymbol{K}_{\boldsymbol{m}}^{\boldsymbol{e}} = \int_{\Omega_\zeta} \tilde{\boldsymbol{B}}_2^{\boldsymbol{T}} \boldsymbol{D}\, \tilde{\boldsymbol{B}}_2 \det \boldsymbol{J}\, d\xi_1 d\xi_2 d\xi_3$
- construction du tenseur symétrique 3×3 des contraintes locales $[\tilde{\boldsymbol{S}}]$
- calcul du tenseur symétrique 3×3 des contraintes globales $[\boldsymbol{S}] = \boldsymbol{P}^{\boldsymbol{T}}[\tilde{\boldsymbol{S}}]\boldsymbol{P}$
- calcul de $\tilde{\boldsymbol{B}}_3^T = \boldsymbol{H}[\boldsymbol{S}]\tilde{\boldsymbol{J}}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_3$

calcul et intégration numérique $\boldsymbol{Z}_{\boldsymbol{I}} = \int_{\Omega_z} \tilde{B}_3^T \tilde{S} \det \boldsymbol{J}\, d\xi_1 d\xi_2 d\xi_3$

- calcul du tenseur généralisé des contraintes **globales** $\bar{S}^{9\times 9} = \begin{bmatrix} [S]^{3\times 3} & 0 & 0 \\ 0 & [S] & 0 \\ 0 & 0 & [S] \end{bmatrix}$
- calcul et intégration numérique de la rigidité classique

$$\boldsymbol{K}_{\boldsymbol{g}}^{\boldsymbol{e}\,\mathbf{classique}} = \int_{\Omega_z} \left[\tilde{\boldsymbol{J}}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2\right]^T \bar{\boldsymbol{S}} \left[\tilde{\boldsymbol{J}}^{-1}\left[\frac{\partial N}{\partial \xi}\right]_2\right] \det \boldsymbol{J}\, d\xi_1 d\xi_2 d\xi_3$$

Fin boucle INTSN sur les points d'intégration

Début Boucle JN sur les $NB2$ **nœuds**

- calcul de $\left[z_I \times\right]$ sachant que $Z_I = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ z_I \\ \cdot \\ \cdot \\ \cdot \\ I = 1, NB2 \end{pmatrix}$
- calcul de $\left[n_I \times\right]$

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

- calcul de la matrice non symétrique $K^{e}_{\text{gnon classique}}{}^{3\times 3}(I,I)=[z_I\times][n_I\times]$

IF JN $\leq$ NB1

- rajout de $K^{e}_{\text{gnon classique}}{}^{3\times 3}(I,I)$ avec distinction de l'extra-nœud

ELSE JN

- affectation de $K^{e}_{\text{gnon classique}}{}^{3\times 3}(I,I)$ avec distinction de l'extra-nœud

END IF JN

Stockage de toute la matrice non symétrique $K^{e}{}_{T}$

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