

Ecole Polytechnique  
Promotion 2008

PEREIRA Philippe Cesar Teixeira

# Schémas d'intégration adaptatifs pour le traitement du contact-frottement en grands glissements

NON CONFIDENTIEL

Département de Physique  
Option: Énergies du XXIème siècle  
Champs de l'option: PHY597 Énergies

Directeur de recherche : Franck CARRE  
Directeur de stage : Patrick MASSIN et Najib MAHJOUBI  
Dates du stage : 06 avril 2011 au 31 août 2011

Adresse de l'organisme :

EDF R&D Clamart - LaMSID  
1 Avenue Gén de Gaulle, 92140 Clamart

## Table des matières

Résumé.....	3
Présentation du Code_Aster.....	4
Formulation continue du problème de contact-frottement.....	4
Schémas d'intégration adaptatifs.....	10
Le fichier mmgaus.f.....	14
Organisation actuelle du fichier mmgaus.f.....	15
Modifications effectuées.....	16
Validation.....	27
Présentation du cas test retenu.....	27
Solution analytique du problème.....	28
Cas 1 : contraintes planes.....	28
Cas 2 : déformations planes.....	29
Rotation de la couronne intérieure.....	30
Simulations et conclusions.....	30
Bibliographie.....	38
Annexe A : fichier de commande SSNP152.....	39
Annexe B : fichier GMSH du maillage du cas test SSPN122.....	44
Annexe C : fiche descriptive du cas test SSNP152.....	46

## Résumé

Dans le cadre du traitement du contact en grands glissements, des techniques d'intégration adaptatives doivent être mises en œuvre, indispensables pour pouvoir traiter correctement le contact entre deux corps déformables. Une formulation de contact dite continue a été implantée dans Code\_Aster, code de thermo-mécanique développé à la R&D d'EDF et diffusé en libre, suite à une collaboration avec le Professeur H. Ben Dhia de l'École Centrale de Paris. Lorsque les points en vis-à-vis du contact se décalent les uns par rapport aux autres, des éléments finis dits tardifs sont créés à la volée et permettent de prendre en compte l'évolution du contact. Ces éléments finis composés des mailles des surfaces de contact en vis-à-vis changent au fur et à mesure de l'évolution du contact et sont recréés à chaque évolution des géométries des surfaces de contact. L'intégration des équations d'équilibre, de contact et de frottement pour ces éléments est délicate. On propose de faire évoluer les schémas d'intégration actuels à terme de façon à créer des sous-éléments pour l'intégration numérique issus de la projection des surfaces de contact l'une sur l'autre. Dans un premier temps on propose d'utiliser une intégration différenciée pour l'équation d'équilibre et les équation de contact et de frottement. Une maquette de travail en 2D a été réalisée, avec les validations associées de façon à montrer les améliorations apportées par cette méthode en terme de précision et de stabilité.

# Présentation du Code\_Aster

Code\_Aster est un logiciel libre de simulation numérique en mécanique des structures développé principalement par la R&D d'EDF. Aster (acronyme signifiant « Analyses des Structures et Thermomécanique pour des Études et des Recherches ») offre toute une gamme de méthodes d'analyse et de modélisation pour les plus divers phénomènes mécaniques : analyse sismiques, thermiques, acoustique, rupture, fatigue, endommagement, etc.

Le logiciel est codé en Fortran, même si de nombreux modules complémentaires sont réalisés en Python. Il comporte aujourd'hui environ 1,2 millions des lignes de code, de nombreux cas tests, ainsi qu'une énorme documentation de plus de 10000 pages.

La modélisation est faite à partir d'un fichier décrivant le maillage (créé avec un logiciel de maillage par éléments finis, comme Salomé, GMSH ou GIBI) et un fichier de commande spécifiant les caractéristiques du problème traité (lecture et modification du maillage de base, affectation de charges mécaniques et/ou thermiques, attribution des propriétés des matériaux, spécification du phénomène et des hypothèses de la modélisation, extraction et traitement statistique des données de sortie, etc.). Le logiciel auxiliaire ASTK permet à l'utilisateur d'organiser ces fichiers d'entrée et de sortie et aussi de définir d'autres paramètres de la simulation.

Parmi les plus divers phénomènes pris en compte par Aster lors des simulations, nous travaillerons sur le traitement du contact et du frottement, et plus précisément sur l'intégration des efforts de contact.

## **Formulation continue du problème de contact-frottement**

On considère deux solides  $B^i (i=1,2)$  déformables en contact frottant. Ces deux solides occupent dans leur configuration initiale l'adhérence de deux domaines  $\Omega_0^1$  et  $\Omega_0^2$  de  $\mathbb{R}^3$  et, dans leur configuration courante à l'instant  $t$ , l'adhérence de  $\Omega_t^1$  et  $\Omega_t^2$  (également compris dans  $\mathbb{R}^3$ ), respectivement. On suppose que, dans leur configuration initiale, ces deux solides sont dans un état naturel, c'est-à-dire sans contraintes résiduelles ou pré-déformations. Considérons une particule matérielle  $M^i$  du solide  $B^i$ , repérée par le vecteur  $\mathbf{p}^i$  dans la configuration initiale. On désigne par  $\varphi_t^i$  la transformation au temps  $t$  du solide  $B^i$ , fonction transportant la particule  $M^i$  vers la position  $\mathbf{x}_t^i$  dans la configuration courante. On a :

$$\begin{aligned}\varphi_t^i: \Omega_0^i &\rightarrow \Omega_t^i \\ \mathbf{p}^i &\rightarrow \mathbf{x}_t^i = \varphi_t^i(\mathbf{p}^i, t)\end{aligned}$$

Au cours de leur mouvement, les solides peuvent entrer en contact. La frontière

$\Gamma^i = \partial\Omega^i$  de chaque solide  $B^i$  est décomposée en parties  $\Gamma_u^i$ ,  $\Gamma_g^i$  et  $\Gamma_c^i$  dans la configuration initiale, dont les intersections sont vides deux à deux:

$$\Gamma_u^i \cap \Gamma_g^i = \emptyset$$

$$\Gamma_u^i \cap \Gamma_c^i = \emptyset$$

$$\Gamma_c^i \cap \Gamma_g^i = \emptyset$$

Ces frontières se déforment dans la configuration courante en  $\gamma_u^i$ ,  $\gamma_g^i$  et  $\gamma_c^i$ . Le solide  $B^i$  est encastré sur  $\Gamma_u^i$  et soumis à une densité nominale de forces surfaciques notée  $g^i$  sur la partie  $\Gamma_g^i$ . Les parties des surfaces  $\Gamma^i$  susceptibles d'entrer en contact lors de la déformation des deux solides sont notées  $\Gamma_c^i$ .

On définit le tenseur gradient de la transformation  $F_t^i$  par :

$$F_t^i = \frac{\partial \mathbf{x}_t^i}{\partial \mathbf{p}^i}$$

Si on désigne par  $\mathbf{u}^i$  le champ des déplacements du solide  $B^i$  :

$$\mathbf{x}_t^i = \mathbf{p}^i + \mathbf{u}^i$$

Alors le tenseur gradient de transformation se ré-écrit :

$$F_t^i = \nabla_{\mathbf{p}^i}(\mathbf{u}^i) + \mathbf{1}$$

La condition cinématique de non interpénétration lors du traitement du contact est associé à une opération d'appariement. On procède comme suit :

- On couple les points des surfaces de contact deux à deux.
- On impose entre les deux points appariés une condition de non-interpénétration suivant une direction donnée (généralement la normale à l'une des deux surfaces au point concerné).

On effectue l'appariement en recherchant, pour tout point  $\mathbf{x}^1$  de la frontière  $\gamma_c^1$ , le point  $\bar{\mathbf{x}}$  de  $\gamma_c^2$  le plus proche. Cela revient à résoudre le problème de minimisation suivant :

$$\begin{aligned} & \forall \mathbf{x}^1 \in \gamma_c^1 \text{ et } \forall t \geq 0 \text{ avec} \\ & \mathbf{x}^1 = \varphi_t^1(\mathbf{p}^1) \text{ et } \mathbf{p}^1 = \Phi^1(\boldsymbol{\zeta}^1) \text{ pour } \boldsymbol{\zeta}^1(\xi_1, \xi_2) \in \omega^1, \\ & \text{trouver } \bar{\boldsymbol{\zeta}}_t \in \omega^2 \text{ tel que:} \\ & \bar{\boldsymbol{\zeta}}_t = \text{ArgMin}_{\boldsymbol{\zeta}^2 \in \omega^2} \left\{ \frac{1}{2} \cdot \left\| \varphi^1(\Phi^1(\boldsymbol{\zeta}^1), t) - \varphi^2(\Phi^2(\boldsymbol{\zeta}^2), t) \right\|^2 \right\} \end{aligned}$$

On appellera surface esclave la surface de contact définie sur le solide 1 et surface maître la surface appariée définie sur le solide 2. La solution  $\bar{\boldsymbol{\zeta}}_t$  est la position dans l'espace de référence paramétrique de la projection M du nœud esclave P sur la maille maître. La solution de ce problème permet formellement de définir à chaque instant t une application d'appariement  $A_t$ , telle que :

$$A_t: \omega^1 \rightarrow \omega^2$$

$$\zeta^1 \rightarrow \bar{\zeta}_t(\bar{\zeta}, t)$$

L'application appariement  $A_t$  permet de définir le point  $\bar{p}$  de la surface  $\Gamma_c^2$ , apparié à  $p^1$  à l'instant  $t$  dans la configuration d'origine, et le point  $\bar{x}$  apparié à  $x^1$  à l'instant  $t$  dans la configuration actuelle. La distance  $d$  entre les points appariés est tout simplement :

$$d = (x^1 - \bar{x}) = \Phi_t^1(\zeta^1) - \Phi_t^2(\bar{\zeta}_t)$$

Soit  $(\hat{t}_1, \hat{t}_2, \hat{n})$  la base construite avec les deux vecteurs des vecteurs tangents et le vecteur normal sur la surface déformée. Le plan défini par les vecteurs  $(\hat{t}_1, \hat{t}_2)$  est tangent à la surface déformée  $\gamma_c^2$  et donc, en utilisant la définition de la normale  $\hat{n}$  à  $\gamma_c^2$ , on a :

$$[\Phi_t^1(\zeta^1) - \Phi_t^2(\bar{\zeta}_t)] = d_n \cdot \hat{n}$$

$d_n$  est une quantité algébrique signée, appelé **jeu** entre les deux solides au point  $x^1$ . On obtient :

$$x^1 = \Phi_t^2(\bar{\zeta}_t) + d_n \cdot \hat{n}$$

Les équations locales d'équilibre, complétées des conditions initiales et des conditions aux limites du problème considéré, s'écrivent sous la forme suivante :

$$\left\{ \begin{array}{ll} \text{div } \Pi^i + f^i = \rho_p^i \cdot \frac{\partial^2 \mathbf{u}^i}{\partial t^2} & \text{dans } \Omega_0^i \quad (a) \\ \Pi^i \cdot N^i = g^i & \text{sur } \Gamma_g^i \quad (b) \\ \mathbf{u}^i = \mathbf{0} & \text{sur } \Gamma_u^i \quad (c) \\ \Pi^i \cdot N^i = r^i & \text{sur } \Gamma_c^i \quad (d) \\ \mathbf{u}^i(p^i, 0) = \mathbf{u}_0^i(p^i) \text{ et } \frac{\partial \mathbf{u}^i(p^i, 0)}{\partial t} = \mathbf{v}_0^i(p^i) & \text{dans } \Omega_0^i \quad (e) \end{array} \right.$$

où  $\Pi^i$  désigne le premier *tenseur des contraintes de Piola-Kirchhoff* (non symétrique),  $\rho_p^i$  la masse volumique dans la configuration initiale,  $r^i$  la densité d'effort due aux interactions de contact frottant éventuelles entre les deux solides. L'exposant  $i$  prend la valeur 1 ou 2.

On démontre qu'il y a une équivalence entre les équations d'équilibre et le principe des travaux virtuels (PTV) ou formulation faible lagrangienne totale, qui s'écrit de la façon suivante :

$$\forall \mathbf{u}^{*i} \in CA^i$$

$$\int_{\Omega_0^i} \rho_p^i \cdot \frac{\partial^2 \mathbf{u}^i}{\partial t^2} \cdot \mathbf{u}^{*i} \cdot d\Omega + \int_{\Omega_0^i} \mathbf{F}^i \cdot \mathbf{S}^i : \nabla_p \mathbf{u}^{*i} \cdot d\Omega - \int_{\Gamma_c^i} \mathbf{R}^i(p^i, t) \cdot \mathbf{u}^{*i} \cdot d\Gamma_c^i = 0$$

où  $\mathbf{R}^i(p^i, t)$  est la densité des efforts de contact subie par le solide  $B^i$ ,  $\mathbf{u}^{*i}$  est le champ des déplacements virtuels et  $\mathbf{S}^i$  est le *second tenseur des contraintes de Piola-Kirchhoff*, symétrique et purement lagrangien, lié au premier tenseur  $\Pi^i$  par la relation :

$$\mathbf{S}^i = (\mathbf{F}^i)^{-1} \cdot \Pi^i$$

Nous écrivons le PTV de manière plus compacte:

$$\forall \mathbf{u}^{*i} \in \text{CA}^i$$

$$G_{\text{dyn}}^i(\mathbf{u}^i, \mathbf{u}^{*i}) + G_{\text{int}}^i(\mathbf{u}^i, \mathbf{u}^{*i}) = G_{\text{cf}}^i(\mathbf{R}^i, \mathbf{u}^{*i})$$

Avec  $G_{\text{dyn}}^i$  le travail des efforts d'inertie:

$$G_{\text{dyn}}^i(\mathbf{u}^i, \mathbf{u}^{*i}) = \int_{\Omega_0^i} \rho_p^i \cdot \frac{\partial^2 \mathbf{u}^i}{\partial t^2} \cdot \mathbf{u}^{*i} \cdot d\Omega$$

$G_{\text{int}}^i$  est le travail des efforts intérieurs :

$$G_{\text{int}}^i(\mathbf{u}^i, \mathbf{u}^{*i}) = \int_{\Omega_0^i} \mathbf{F}^i \cdot \mathbf{S}^i : \nabla_{\mathbf{p}^{*i}}(\mathbf{u}^{*i}) \cdot d\Omega$$

$G_{\text{cf}}^i$  est le travail des efforts de contact/frottement:

$$G_{\text{cf}}^i(\mathbf{R}^i, \mathbf{u}^{*i}) = \int_{\Gamma_c^i} \mathbf{R}^i(\mathbf{p}^i, t) \cdot \mathbf{u}^{*i} \cdot d\Gamma_c^i$$

Le principe de l'action et de la réaction s'écrit localement, sur les configurations initiales sous la forme suivante :

$$\mathbf{R}^1(\mathbf{p}^1, t) \cdot d\Gamma_c^1 + \mathbf{R}^2(\bar{\mathbf{p}}, t) \cdot d\Gamma_c^2 = 0$$

on peut donc prendre:

$$\mathbf{R}^1 \cdot d\Gamma_c^1 = -\mathbf{R}^2 \cdot d\Gamma_c^2 = \mathbf{R} \cdot d\Gamma_c$$

ce qui nous donne une nouvelle forme du terme de réaction du contact sous la forme forte:

$$G_{\text{cf}}(\mathbf{R}, \mathbf{u}^*) = \int_{\Gamma_c} \mathbf{R} \cdot (\mathbf{u}^{*1} - \mathbf{u}^{*2}) \cdot d\Gamma_c = \int_{\Gamma_c} \mathbf{R} \cdot \llbracket \mathbf{u}^* \rrbracket \cdot d\Gamma_c$$

En décomposant la réaction sur la composante normale (contact) et sur la composante tangentielle (frottement) :

$$\mathbf{R} = \mathbf{R}_\tau + \lambda \cdot \mathbf{n}$$

où  $\lambda$  est la pression de contact.

D'après la loi de Signorini pour le contact, la réaction est discontinue et prend deux valeurs (contact ou non) :

$$\mathbf{R} = \begin{cases} \lambda \cdot \mathbf{n} + \mu \cdot \lambda \cdot \mathbf{\Lambda} & \text{si } d_n = 0 \\ \mathbf{0} & \text{si } d_n < 0 \end{cases}$$

où  $d_n$  est le jeu algébrique et  $\mu$  est appelé coefficient de frottement de Coulomb. On écrit de manière plus compacte par :

$$\mathbf{R} = S_u \cdot (\lambda \cdot \mathbf{n} + \mu \cdot \lambda \cdot \mathbf{\Lambda}) \text{ avec } S_u = I_{\mathbb{R}}(\lambda - \rho_n \cdot d_n), \rho_n > 0$$

Le semi-multiplicateur de frottement est également discontinu et prend deux valeurs (glissant ou adhérent):

$$\Lambda = \begin{cases} \Lambda & \text{si } S_f = 1 \\ \frac{\mathbf{v}_\tau}{\|\mathbf{v}_\tau\|} & \text{si } S_f = 0 \end{cases} \text{ avec } S_f = I_{B(0,1)}(\Lambda + \rho_\tau \cdot \mathbf{v}_\tau), \rho_\tau > 0$$

où  $\mathbf{v}_\tau$  est la vitesse relative tangente.

Lors de l'écriture de la loi de Coulomb, on procède à une homogénéisation en introduisant le paramètre  $h_\tau > 0$  :

$$\Lambda = \begin{cases} \Lambda & \text{si } S_f = 1 \\ \frac{\Lambda + h_\tau \cdot \mathbf{v}_\tau}{\|\Lambda + h_\tau \cdot \mathbf{v}_\tau\|} & \text{si } S_f = 0 \end{cases}$$

ce qu'on écrit de manière plus compacte par :

$$\Lambda = S_f \cdot \Lambda + (1 - S_f) \cdot \frac{\Lambda + h_\tau \cdot \mathbf{v}_\tau}{\|\Lambda + h_\tau \cdot \mathbf{v}_\tau\|}$$

en réinjectant dans l'expression de  $\mathbf{R}$  :

$$\mathbf{R} = S_u \cdot \left( \lambda \cdot \mathbf{n} + \mu \cdot \lambda \cdot S_f \cdot \Lambda + \mu \cdot \lambda \cdot (1 - S_f) \cdot \frac{\Lambda + h_\tau \cdot \mathbf{v}_\tau}{\|\Lambda + h_\tau \cdot \mathbf{v}_\tau\|} \right)$$

Finalement, le terme de réaction de contact-frottement dans le PTV peut se décomposer en une composante pour le contact :

$$G_{\text{cont}}(\lambda_k, \mathbf{u}_k, \mathbf{u}^*) = - \int_{\Gamma_c} S_{uk} \cdot \lambda_k \cdot \llbracket \mathbf{u}^* \rrbracket_n \cdot d\Gamma_c$$

et une composante pour le frottement :

$$G_{\text{frot}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \mathbf{u}^*) = - \int_{\Gamma_c} S_{uk} \cdot \mu \cdot \lambda_k \cdot S_{fk} \cdot \Lambda_k \cdot \llbracket \mathbf{u}^* \rrbracket_\tau \cdot d\Gamma_c \\ - \int_{\Gamma_c} S_{uk} \cdot \mu \cdot \lambda_k \cdot (1 - S_{fk}) \cdot \frac{\Lambda_k + h_\tau \cdot \mathbf{v}_k}{\|\Lambda_k + h_\tau \cdot \mathbf{v}_k\|} \cdot \llbracket \mathbf{u}^* \rrbracket_\tau \cdot d\Gamma_c$$

La loi de Signorini pour le contact peut être écrite comme suit

$$\lambda - S_u \cdot (\lambda - h_n \cdot d_n) = 0, h_n \neq 0$$

on en obtient la forme faible de la loi de contact :

$$G_{\text{cont}}^{\text{faible}}(\lambda_k, \mathbf{u}_k, \lambda^*) = - \frac{1}{h_n} \cdot \int_{\Gamma_c} (\lambda_k - S_{uk} (\lambda_k - h_n \cdot d_{nk})) \cdot \lambda^* \cdot d\Gamma_c$$

La loi de Coulomb pour le frottement peut être écrite comme suit :

$$(1 - S_u) \cdot \Lambda + S_u \cdot \left[ (1 - S_f) \cdot \Lambda - S_f \cdot h_\tau \cdot \mathbf{v}_\tau - (1 - S_f) \cdot \frac{\Lambda + h_\tau \cdot \mathbf{v}_\tau}{\|\Lambda + h_\tau \cdot \mathbf{v}_\tau\|} \right] = \mathbf{0}, h_\tau \neq 0$$



on en obtient la forme faible de la loi de frottement :

$$\begin{aligned}
G_{\text{frot}}^{\text{faible}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \Lambda^*) &= \frac{1}{h_\tau} \cdot \int_{\Gamma_c} \mu \cdot \lambda_k \cdot S_{uk} \cdot \Lambda_k \cdot \Lambda^* \cdot d\Gamma_c - \\
&\frac{1}{h_\tau} \cdot \int_{\Gamma_c} \mu \cdot \lambda_k \cdot S_{uk} \cdot S_{fk} \cdot (\Lambda_k + h_\tau \cdot \mathbf{v}_{\tau k}) \cdot \Lambda^* \cdot d\Gamma_c - \\
&\frac{1}{h_\tau} \cdot \int_{\Gamma_c} \mu \cdot \lambda_k \cdot S_{uk} \cdot (1 - S_{fk}) \cdot \left( \frac{\Lambda_k + h_\tau \cdot \mathbf{v}_{\tau k}}{\|\Lambda_k + h_\tau \cdot \mathbf{v}_{\tau k}\|} \right) \cdot \Lambda^* \cdot d\Gamma_c + \\
&\int_{\Gamma_c} (1 - S_{uk}) \cdot \Lambda_k \cdot \Lambda^* \cdot d\Gamma_c
\end{aligned}$$

Il convient d'abord de remarquer que la loi de frottement de Coulomb introduit la notion de vitesse relative de glissement  $\mathbf{v}_{\tau k}$  entre les deux solides. En quasi-statique cette vitesse s'écrit sous forme incrémentale :

$$\mathbf{v}_{\tau k} = \frac{1}{\Delta t_k} \cdot \Delta \llbracket \mathbf{x}_k \rrbracket_\tau$$

où  $\Delta \llbracket \mathbf{x}_k \rrbracket_\tau$  représente la projection de l'incrément du saut de déplacement  $\mathbf{x}_k$ , à l'instant  $t_k$ , sur le plan tangent à la surface de contact au point de contact considéré.

Le système à résoudre (formulation lagrangien) est donc :

Trouver les champs  $(\mathbf{u}_k, \lambda_k, \Lambda_k) \in \text{CA} \times H \times \mathbf{H}$  tels que :

$$\forall (\mathbf{u}^*, \lambda^*, \Lambda^*) \in \text{CA} \times H \times \mathbf{H}$$

$$\begin{cases}
G_{\text{int}}(\mathbf{u}_k, \mathbf{u}^*) + G_{\text{cont}}(\lambda_k, \mathbf{u}_k, \mathbf{u}^*) + G_{\text{frot}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \mathbf{u}^*) = 0 & (a) \\
G_{\text{cont}}^{\text{faible}}(\lambda_k, \mathbf{u}_k, \lambda^*) = 0 & (b) \\
G_{\text{frot}}^{\text{faible}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \Lambda^*) = 0 & (c)
\end{cases}$$

avec les conditions limites habituelles.

où les espaces  $H$  et  $\mathbf{H}$  sont des espaces de Hilbert.

La formulation peut aussi être stabilisée par le rajout d'un terme de type pénalisation (dont le paramètre est déconnecté de celui d'augmentation) au niveau de  $G_{\text{cont}}(\lambda_k, \mathbf{u}_k, \mathbf{u}^*)$  et  $G_{\text{frot}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \mathbf{u}^*)$  apparaissant dans les équations d'équilibre. Elle donne alors lieu à une formulation lagrangienne stabilisée; le terme de stabilisation n'introduisant aucune perturbation de la solution. Si on ne garde que les termes de stabilisation, on retrouve une formulation par pénalisation classique, qui n'intervient qu'au niveau de l'équation d'équilibre : dans ce cas là on retrouve la dépendance des résultats obtenus au choix des coefficients de pénalisation strictement positifs.

Jusqu'à présent les équations ont été écrites sous forme d'intégrales continues. Pour évaluer numériquement ces intégrales, il convient de choisir un schéma de quadrature numérique idoine. La question se déplace donc vers le choix des points de collocation optimaux pour obtenir des efforts de contact précis, sachant que des termes discontinus doivent être pris en compte, du fait des statuts de contact ou de frottement.

## Schémas d'intégration adaptatifs

On propose ici une nouvelle façon d'intégrer les termes de contact pour la méthode continue. La façon actuelle dont Aster effectue les intégrations des efforts de contact le long des éléments n'est pas correcte car elle ne prend en compte que la contribution de l'élément de la maille maître avec lequel l'élément de la maille esclave est apparié. Dans certaines configurations entre les maillages maître et esclave, lorsqu'on travaille en grand glissement entre deux solides déformables, on peut s'apercevoir que certaines mailles maître en vis-à-vis de la surface esclave ne sont pas sollicitées. La figure ci-dessous illustre cette situation:

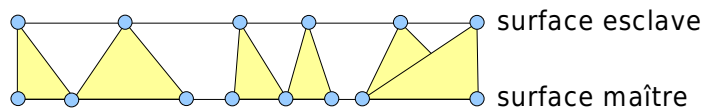


Figure 1: on voit que 2 mailles maître sont oubliées actuellement avec l'intégration aux nœuds des termes de contact. Les triangles jaunes montrent représentent les éléments auxquels les nœuds de la surface esclave sont associés.

Un autre exemple est illustré ci-dessous. Imaginons que l'aire élémentaire d'intégration sur la surface esclave recouvre plusieurs éléments de la surface maîtresse. Comme les fonctions-test ont pour support non-nul un élément fini, le fait d'intégrer numériquement, par une quadrature de Gauss, une fonction non polynomiale, mais identiquement nulle sur une partie de l'aire d'intégration, n'est clairement pas exact, car la quadrature est tout à fait inappropriée.

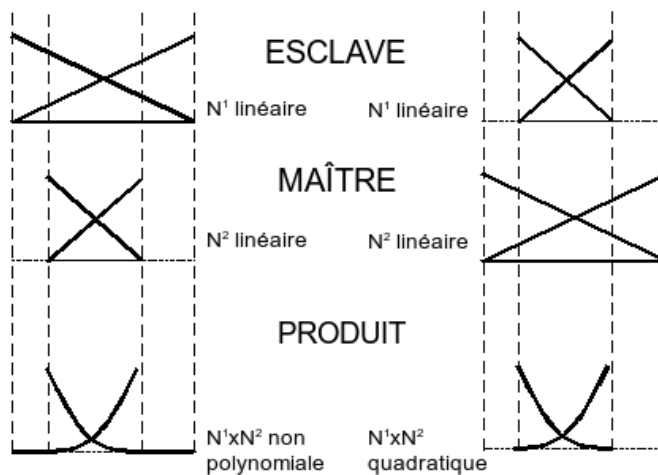


Figure 2: lors de la résolution des équations régissant le problème du contact, les intégrales du produit des fonctions de forme ne sont intégrables que si le support de la maille maître est plus large que celui de la maille esclave en vis-à-vis.

L'utilisation des différentes méthodes d'intégration numérique ont pour but de minimiser les problèmes d'incompatibilités entre les surfaces, mais en tous cas les calculs ainsi effectués sont soit faux soit approximatifs.

On propose un découpage adaptatif, utilisant la projection de la discrétisation de la surface maître sur la surface esclave. Lors de la phase d'appariement du contact, pour une maille esclave on va associer toutes les mailles maître où on détecte du contact. L'intégration des termes de contact se fera alors sur l'intersection d'une maille bleue et d'une maille rouge, avec un redécoupage en sous-domaines pour pouvoir faire les intégrations. L'absence de sous-intégration adaptative pour les éléments de contact, si la surface maître n'est pas rigide, conduit à des trous dans la prise en compte du contact lorsque les maillages maître-esclave ne sont pas compatibles.

L'adaptation par sous-découpage du schéma d'intégration n'est pas la seule difficulté à lever pour l'intégration des termes de contact frottement. La notion de statut des points de contact est aussi importante.

L'équation du contact est donnée ci-dessous:

$$G_{\text{cont}}^{\text{faible}}(\lambda_k, \mathbf{u}_k, \lambda^*) = -\frac{1}{h_n} \cdot \int_{\Gamma_c} ((1 - S_{uk})\lambda_k + S_{uk} \cdot h_n \cdot d_{nk}) \cdot \lambda^* \cdot d\Gamma_c = 0$$

Cette approche a l'avantage de pouvoir avoir immédiatement le statut au nœud du maillage et de distinguer la notion entre points de contrôle du contact autorisés par l'interpolation élément fini et point d'intégration du contact. Sur la partie en contact, l'intégrale se réduit à

$$G_{\text{cont}}^{\text{faible}}(\lambda_k, \mathbf{u}_k, \lambda^*) = -\int_{\Gamma_c} d_{nk} \cdot \lambda^* \cdot d\Gamma_c = 0$$

et sur la partie qui n'est pas, l'intégrale se réduit à

$$G_{\text{cont}}^{\text{faible}}(\lambda_k, \mathbf{u}_k, \lambda^*) = -\frac{1}{h_n} \cdot \int_{\Gamma_c} \lambda_k \cdot \lambda^* \cdot d\Gamma_c = 0$$

Pour des éléments compatibles géométriquement, et si le statut de contact est identique sur l'élément fini, les deux équation ci-dessus impliquent que soit les pression aux nœuds de calcul sont nulles, soit les sauts de déplacements aux nœuds de calcul sont nuls.

Nous pouvons raisonner sur un seul élément 2D, en se plaçant dans un premier temps dans le cas d'un élément de contact, avec une normale fixe unique (surface maître rigide par exemple), pour lequel les surfaces maîtres et esclaves utilisent les mêmes fonctions de forme et dont les nœuds maître et esclave sont en vis-à-vis, c'est-à-dire pour des maillages compatibles. On a:

$$\int_{\Gamma_c} ((1 - S_{u1})N_1 \lambda_1 + (1 - S_{u2})N_2 \lambda_2 + S_{u1} h_n N_1 d_{n1} + S_{u2} h_n N_2 d_{n2}) \cdot (N_1 \lambda_1^* + N_2 \lambda_2^*) \cdot d\Gamma_c = 0 \quad \forall (\lambda_1^*, \lambda_2^*)$$

Ce qui amène à résoudre un système de deux équations à deux inconnues en  $(\lambda_1, \lambda_2)$ , du type :

$$\begin{bmatrix} \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{1i} & \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{1i} \\ \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{2i} & \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{2i} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -h_n \sum_i (\omega_i S_{ui} N_{1i} [N_{1i} d_{n1} + N_{2i} d_{n2}]) \\ -h_n \sum_i (\omega_i S_{ui} N_{2i} [N_{1i} d_{n1} + N_{2i} d_{n2}]) \end{bmatrix}$$

où  $S_{ui}$  est le statut de contact au point d'intégration  $i$  et  $N_{ji}$  la valeur de la fonction de forme  $j$  au point d'intégration  $i$ . Si, au sein de l'élément, tous les points ont le même statut alors soit  $(d_{n1}, d_{n2}) = (0, 0)$  pour un statut de contact, soit  $(\lambda_1, \lambda_2) = (0, 0)$  pour un statut non contactant. Si certains points d'intégration ont des statuts différents, alors on peut exprimer

$(\lambda_1, \lambda_2)$  comme une fonction de  $(d_{n1}, d_{n2})$  par inversion du système précédent étant donné que :

$$\left[ \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{1i} \cdot \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{2i} \right] - \left[ \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{1i} \right]^2 \neq 0$$

Ce déterminant montre que la résolution du système précédent n'est possible que si au moins un point de contact (point d'intégration) n'est pas dans un état contactant et que les fonctions de forme ne s'annulent pas en ce point, ce qui exclut les schémas d'intégration nodale. Dans le cas contraire, si tous les points sont en contact, alors,  $(d_{n1}, d_{n2}) = (0, 0)$  et pour les intégrations nodales si le déterminant est nul soit  $(d_{n1}, d_{n2}) = (0, 0)$ , soit  $(d_{n1}, \lambda_2) = (0, 0)$ , soit  $(\lambda_1, d_{n2}) = (0, 0)$ .

Si au moins un point d'intégration n'est pas contactant, le système se résout de la manière suivante :

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \frac{-h_n}{det} \begin{bmatrix} \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{1i} N_{1i} - \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{2i} N_{2i} & \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{1i} N_{1i} - \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{2i} N_{2i} \\ \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{1i} \cdot \sum_i \omega_i S_{ui} N_{2i} N_{2i} - \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{1i} \cdot \sum_i \omega_i S_{ui} N_{1i} N_{1i} & \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{1i} N_{1i} - \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{2i} N_{2i} \end{bmatrix} \begin{bmatrix} d_{n1} \\ d_{n2} \end{bmatrix}$$

L'ensemble des solutions obtenues est résumé dans le tableau ci-dessous dans le cas où  $\exists i, S_{ui} = 0; \exists i' \neq i, S_{ui'} = 1$  :

Valeur du jeu	Valeur de la pression	Interprétation
$(d_{n1}, d_{n2}) = (0, 0)$	$(\lambda_1, \lambda_2) = (0, 0)$	Le contact au point d'intégration n'est pas vu par le système
$(d_{n1}, d_{n2}) \neq (0, 0)$	$(\lambda_1, \lambda_2) \neq (0, 0)$	Du fait qu'il y a au moins un point d'intégration en contact, les pressions de contact nodales sont non nulles, même si aux nœuds de l'élément on ne détecte pas de contact car $(d_{n1}, d_{n2}) \neq (0, 0)$ .

La résolution de ce système qui n'assure pas aux nœuds de lien logique entre les valeurs de  $(d_{n1}, d_{n2})$  et  $(\lambda_1, \lambda_2)$  conduit généralement à des oscillations sur les pressions de contact : elle est donc à éviter, sauf si une intégration aux nœuds est utilisée. Dans ce cas là, le système se réduit à :

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \frac{-h_n}{det} \begin{bmatrix} \sum_i \omega_i (1 - S_{ui}) N_{2i} N_{2i} \cdot \sum_i \omega_i S_{ui} N_{1i} N_{1i} & 0 \\ 0 & \sum_i \omega_i (1 - S_{ui}) N_{1i} N_{1i} \cdot \sum_i \omega_i S_{ui} N_{2i} N_{2i} \end{bmatrix} \begin{bmatrix} d_{n1} \\ d_{n2} \end{bmatrix}$$

ce qui conduit à :

Statut du contact	Valeur du jeu	Valeur de la pression
$(S_{u1}, S_{u2}) = (0, 0)$ nœuds sommet de l'élément non contactant		$(\lambda_1, \lambda_2) = (0, 0)$
$(S_{u1}, S_{u2}) = (1, 0)$	$d_{n1} = 0$	$\lambda_2 = 0$
$(S_{u1}, S_{u2}) = (0, 1)$	$d_{n2} = 0$	$\lambda_1 = 0$

Si l'on prend en compte non plus le statut au point de gauss, mais le statut aux nœuds inconnus du système, le système précédent devient :

$$\begin{bmatrix} (1 - S_{u1}) \sum_i \omega_i N_{1i} N_{1i} & (1 - S_{u2}) \sum_i \omega_i N_{2i} N_{2i} \\ (1 - S_{u1}) \sum_i \omega_i N_{1i} N_{2i} & (1 - S_{u2}) \sum_i \omega_i N_{2i} N_{1i} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -h_n S_{u1} d_{n1} \sum_i \omega_i N_{1i} N_{1i} - h_n S_{u2} d_{n2} \sum_i \omega_i N_{1i} N_{2i} \\ -h_n S_{u1} d_{n1} \sum_i \omega_i N_{2i} N_{1i} - h_n S_{u2} d_{n2} \sum_i \omega_i N_{2i} N_{2i} \end{bmatrix}$$

Ce système est à déterminant nul. Il donne une solution cohérente entre les statuts de contact, les valeurs de déplacements et les valeurs de pression.

Statut du contact	Valeur du jeu	Valeur de la pression
$(S_{u1}, S_{u2})=(0,0)$ nœuds sommet de l'élément non contactant		$(\lambda_1, \lambda_2)=(0,0)$
$(S_{u1}, S_{u2})=(1,0)$	$d_{n1}=0$	$\lambda_2=0$
$(S_{u1}, S_{u2})=(0,1)$	$d_{n2}=0$	$\lambda_1=0$
$(S_{u1}, S_{u2})=(1,1)$ nœuds sommet de l'élément non contactant	$(d_{n1}, d_{n2})=(0,0)$	

Nous allons donc utiliser une intégration nodale pour les formulations faibles de contact et de frottement, et une intégration aux points de gauss des termes de contact et de frottement dans l'équation d'équilibre. Physiquement cela s'interprète de la manière suivante : la formulation faible permet de savoir si un nœud va porter une contribution de contact (saut nul et pression à déterminer) ou non (pression nulle et saut non nul). L'équation d'équilibre permet de déterminer la valeur de la pression dans le premier cas et la valeur du saut dans le second : les différentes contributions aux points de gauss se cumulent pour déterminer la contribution aux nœuds de calcul du système. Ce point particulier est développé dans le cadre ce stage.

$$\begin{cases} G_{\text{int}}(\mathbf{u}_k, \mathbf{u}^*) + G_{\text{cont}}(\lambda_k, \mathbf{u}_k, \mathbf{u}^*) + G_{\text{frot}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \mathbf{u}^*) = 0 & \text{quadrature choisie par l'utilisateur} \\ G_{\text{cont}}^{\text{faible}}(\lambda_k, \mathbf{u}_k, \lambda^*) = 0 & \text{intégration nodale} \\ G_{\text{frot}}^{\text{faible}}(\Lambda_k, \lambda_k, \mathbf{u}_k, \Lambda^*) = 0 & \text{intégration nodale} \end{cases}$$

Les travaux réalisés sont intégrés dans la version NEW10 d'Aster par Najib Mahjoubi, et un des objectif de ce stage est le raccordement entre ces travaux et la version en développement du code (NEW11).

En NEW10, deux TE sont impliqués dans la constructions des matrices de rigidité (RIGI\_CONT et RIGI\_FROT) et des seconds membres (CHAR\_MECA\_CONT et CHAR\_MECA\_FROT): respectivement te0364.f et te0365.f. Au moment du calculs de ces termes, la récupération des données du contact se fait dans un CHAM\_ELEM de contact. Les calculs préparatifs des calculs (calcul du jeu, coordonnées actualisées, etc) sont regroupés dans mmtpe.f alors que précédemment ces calculs apparaissent clairement dans les TE.

## Le fichier mmgauus . f

Lors de la définition des paramètres de contact dans le fichier de commande d'Aster, l'utilisateur peut choisir la quadrature qui sera utilisée pour l'intégration numérique des efforts sur la surface de contact.

$$\int_D f(x) dx = \sum_i \omega_i f(x_i), x_i \in D$$

```
CONT=DEFI_CONTACT([...]
  ZONE=(
    _F([...]
      INTEGRATION = // choix de la quadrature //,
      [...]),),);
```

Le cas test SSNP152 nous a montré que ce choix peut nous donner des résultats très différents et souvent écartés de la solution analytique. Actuellement (version STA10) on a les choix suivants:

- NOEUD

C'est la méthode par défaut d'Aster pour le traitement du contact. Les points d'intégration sont les nœuds de l'élément et on leur attribue les mêmes poids normalisés.

- GAUSS

C'est une méthode de quadrature exacte pour un polynôme de degré  $2n-1$  avec  $n$  points pris sur le domaine d'intégration. Les nœuds sont déterminés comme les  $n$  racines du  $n$ -ème polynôme orthogonal associé à la formule de quadrature, qui dans le cas le plus classique est le polynôme de Legendre (quadrature de Gauss-Legendre). Le polynôme de Legendre d'ordre  $n$  et le poids du  $i$ -ème point de Gauss sont données par les formules:

$$P_n(x) = \frac{1}{2^n} \sum_{k=0}^n \binom{n}{k} (x-1)^{n-k} (x+1)^k$$
$$\omega_i = \frac{-2}{(n+1)P_n'(x_i)P_{n+1}(x_i)}$$

- SIMPSON
- SIMPSON1
- SIMPSON2

La quadrature de Simpson consiste à intégrer le polynôme interpolateur d'ordre 2 passant par des points équirépartis sur le domaine (3 points pour les éléments linéiques, 6 pour les triangulaires ou 9 pour les quadrangulaires). Les options SIMPSON1 et SIMPSON2 subdivisent ce domaine en respectivement 2 et 4 sous-domaines suivant chaque dimension (2D ou 3D) et calculent les intégrales des polynômes quadratiques

interpolant chaque subdivision.

- NCOTES
- NCOTES1
- NCOTES2

La quadrature de Newton-Cotes est une généralisation de la quadrature de Simpson: au lieu d'utiliser des polynômes interpolateurs d'ordre 2, on utilise des polynômes d'ordre quelconque, toujours en prenant des points équirépartis sur le domaine d'intégration. Les options NCOTES, NCOTES1 et NCOTES2 utilisent des polynômes interpolateurs d'ordre 3, 4 et 9 respectivement.

Le nombre de points d'intégration de chaque quadrature dépend de l'élément du maillage sur lequel on intègre.

Expression du nombre de points d'intégration				
Quadrature Élément	Noeud	Gauss (pour un schéma d'ordre n)	Simpson (avec n subdivisions)	Newton-Cotes (pour un polynôme interpolateur d'ordre n)
SEx	x	n	2n+1	n+1
TRx	x	$fpg_{TR}(n)$	$\frac{(2n+1)(2n+2)}{2}$	$\frac{(n+1)(n+2)}{2}$
QUx	x	$n^2$	$(2n+1)^2$	$(n+1)^2$

n	1	2	3	4	5	6	7	8	9	10
$fpg_{TR}(n)$	1	3	4	6	7	12	13	16	19	25

### **Organisation actuelle du fichier mmgaus . f**

Lors des simulations du cas test SSNP152, nous avons effectué des modifications dans les méthodes d'intégration afin d'obtenir des résultats plus proches de la solution analytique. Dans un premier temps, on a modifié les fichiers du code afin de permettre l'utilisation des quadratures de Gauss d'ordre supérieur. En effet, les quadratures d'ordre 4, 6 et 7 était déjà implémentées, mais restreintes aux formulations basées sur XFEM (FORMULATION == 'XFEM'), pouvant être utilisées avec les options FPG4, FPG6 et FPG7.

Les coordonnées et les poids des points d'intégration sont stockés dans le fichier mmgaus . f. Cette sous-routine d'Aster prend en entrée les arguments suivants:

- ALIAS : une chaîne de caractères indiquant le nom des éléments
- TYPI : un entier indiquant le type de quadrature
- NORD : un entier indiquant le numéro du point d'intégration

et fournit en sortie

- XPG : la première coordonnée paramétrique du point d'intégration
- YPG : la seconde coordonnée paramétrique du point d'intégration (nulle

pour des éléments linéiques)

- HPG : le poids associé à ce point

L'algorithme a été implémenté de la façon suivante:

- On regarde le type d'élément (ALIAS: 'QU4'; 'QU8' ou 'QU9'; 'TR3'; 'TR6' ou 'TR7'; 'SE2'; 'SE3');
- Ensuite, le type d'intégration choisi (TYPI: 1 - NOEUD; 2 - GAUSS; 3 - SIMPSON; 4 - SIMPSON1; 5 - SIMPSON2; 6 - NCOTES; 7 - NCOTES1; 8 NCOTES2; 14 - FPG4; 16 - FPG6; 17 - FPG7);
- pour chaque numéro du point (NORD), on donne les valeurs explicites de XPG, YPG et HPG;

Une analyse détaillée du fichier `mmgaus.f` (version NEW11) nous a montré pourtant:

- quelques quadratures n'ont pas été implémentées (SIMPSON1, SIMPSON2 et NCOTES2 pour les éléments triangulaires) ou correspondaient en effet à une autre quadrature (les points et les poids de NCOTES et NCOTES1 pour les éléments triangulaires sont ceux de la quadrature de Gauss d'ordre 3 et 4 respectivement);
- le fichier présente des blocs répétés (les poids et les points des quadratures de Newton-Cotes pour les éléments linéiques ou quadrangulaires sont les mêmes quelque soit l'ordre de l'élément, linéaire (SE2, QU4) ou quadratique (SE3, QU8 ou QU9);
- dû aux symétries des quadratures, les poids et les coordonnées sont souvent redondantes, de telle sorte que le stockage des valeurs explicites pour tous les points n'est pas nécessaire (par exemple, les coordonnées des points d'intégration des éléments quadrangulaires sont les mêmes que celles des éléments linéiques, répartis suivant une dimension de plus).

Afin d'améliorer les performance du code et de permettre aux utilisateurs d'Aster d'obtenir des résultats plus précis, on a effectué quelques modifications dans les méthodes d'intégration.

## **Modifications effectuées**

Le but à la base était de regarder l'influence du nombre de points et de la méthode d'intégration choisis sur les résultats obtenus lors des simulations. Vu que les quadratures utilisant une grande quantité de points d'intégration demandent un coût de calcul (temps et mémoire) plus important, il serait intéressant pour l'utilisateur d'avoir la possibilité de pouvoir contrôler ces paramètres en fonction du problème traité.

Les remarques faites dans le paragraphe précédent ont été aussi prises en compte dans les développements. Finalement, on a fait les modifications suivantes:

- au lieu de regarder d'abord le type d'élément, on commence par le type d'intégration choisi. Cela nous aidera dans l'organisation de l'algorithme;
- l'entier représentant la quadrature choisie porte aussi un paramètre



d'intégration supplémentaire:

`TYPI = 10 * parint + type`

Ce paramètre d'intégration `parint` est soit l'ordre de la quadrature de Gauss (de 1 à 10, 3 par défaut) soit le nombre de subdivisions du domaine d'intégration pour Simpson (de 1 à 4, 1 par défaut) soit l'ordre de la quadrature de Newton-Cotes (de 3 à 10, 3 par défaut). Par exemple, si `TYPE` vaut 52, on utilise la méthode de Gauss d'ordre 5. A chaque fois qu'on appelle la sous-routine `mmgaus` on prend la division entière de `TYPE` par 10 et le reste de la division par 10 pour obtenir ces paramètres. On évite ainsi de rajouter un argument de plus à la méthode.

- au lieu d'écrire dans le fichier les valeurs des poids et coordonnées pour chaque point, on essaiera soit de les calculer (les coordonnées des quadratures de Simpson et Newton-Cotes sont équiréparties sur le domaine, on peut donc écrire une fonction pour obtenir leurs valeurs à partir de `NORD`) soit de les chercher dans un tableau qui prendrait en compte les redondances (les coordonnées des points de Gauss sont symétriques par rapport à l'origine, donc pour une quadrature de ce type d'ordre  $n$ , on n'a besoin de stocker que  $\lfloor \frac{n}{2} \rfloor$  valeurs différentes).

Le paramètre d'intégration supplémentaire sera indiqué par l'utilisateur comme un argument facultatif lors de la définition des propriétés du contact.

```
CONT=DEFI_CONTACT( [ ... ]
                  ZONE=(
                    _F([ ... ]
                      INTEGRATION = // GAUSS | SIMPSON | NCOTES //,
                      ORDRE_INT   = // 1-10 | 1-4   | 3-10 //,
                      [ ... ]), ), );
```

Afin de ne pas imposer des modifications dans les fichiers de commandes déjà existants, les options `SIMPSON1`, `SIMPSON2`, `NCOTES1` et `NCOTES2` demeurent valables, avec les équivalences suivantes:

<code>INTEGRATION = SIMPSON1,</code>	<code>=</code>	<code>INTEGRATION = SIMPSON,</code> <code>ORDRE_INT   = 2,</code>
<code>INTEGRATION = SIMPSON2,</code>	<code>=</code>	<code>INTEGRATION = SIMPSON,</code> <code>ORDRE_INT   = 4,</code>
<code>INTEGRATION = NCOTES1,</code>	<code>=</code>	<code>INTEGRATION = NCOTES,</code> <code>ORDRE_INT   = 5,</code>
<code>INTEGRATION = NCOTES2,</code>	<code>=</code>	<code>INTEGRATION = NCOTES,</code> <code>ORDRE_INT   = 8,</code>

Lorsqu'on choisit la méthode d'intégration par nœud (`TYPI = 1`), on procède comme dans le code déjà implémenté: on regarde le type d'élément et ensuite on donne les coordonnées et poids pour chaque point explicitement. Une partie du code est donnée ci-dessous:

```
IF (TYPI .EQ. 1) THEN
  IF (ALIAS(1:3) .EQ. 'SE2') THEN
```

```

IF (NORD .EQ. 1) THEN
  XPG = -1.D0
ELSE IF (NORD .EQ. 2) THEN
  XPG = 1.D0
ELSE
  CALL ASSERT(.FALSE.)
END IF
YPG = 0.D0
HPG = 1.D0
[...]
```

Pour les quadratures de Gauss, les coordonnées et poids sont stockés dans des tableaux définis au début du fichier mmgaus.f. Pour les éléments linéiques et quadrangulaires on a un tableau à trois dimensions appelé FPG:

```

REAL*8      FPG(10,5,2)
[...]
```

FPG(1,1,1) = 0.D0
FPG(1,1,2) = 2.D0
FPG(2,1,1) = 0.577350269189626D0
FPG(2,1,2) = 1.D0
FPG(3,1,1) = 0.D0
FPG(3,1,2) = 0.888888888888889D0
FPG(3,2,1) = 0.774596669241483D0
FPG(3,2,2) = 0.555555555555556D0
[...]

Le premier indice indique l'ordre de la quadrature, le deuxième la numérotation des points avec des poids distincts et le troisième vaut 1 pour la coordonnée et 2 pour le poids. On voit par exemple que pour la quadrature de Gauss d'ordre 3 pour les segments on travaille avec 3 points:

$$(\xi_1 = -\sqrt{\frac{3}{5}}, \omega_1 = \frac{5}{9}); (\xi_2 = 0, \omega_2 = \frac{8}{9}); (\xi_3 = \sqrt{\frac{3}{5}}, \omega_3 = \frac{5}{9})$$

```

ELSE IF (mod(TYPI,10) .EQ. 2) THEN
  PARAM = TYPI/10

  IF (ALIAS(1:2) .EQ. 'SE') THEN
    H = (PARAM+1)/2
    I = mod(NORD,H)+1

    XPG = sign(FPG(PARAM,I,1),REAL(2*(NORD/(H+1))-1))
    YPG = 0.D0
    HPG = FPG(PARAM,I,2)
    [...]
```

Lorsque l'utilisateur choisit la méthode de Gauss, TYPI vaut  $10n+2$ , où  $n$  est l'ordre de la quadrature. D'abord on calcule le reste de la division de TYPI par 10, ensuite on stocke en PARAM l'ordre choisi en faisant la division entière par 10. Comme mentionné ci-dessus, la symétrie de la quadrature nous donne un nombre  $\lceil \frac{n}{2} \rceil$  de poids distincts (valeur calculée en H). On obtient l'indice I du NORD-ème point en faisant le reste de la division de NORD par H (ajouté d'une unité, car en FORTRAN la numérotation des tableaux commence par 1). Finalement, pour les points pour lesquels la valeur de NORD est plus petite que H+1 on attribuera des valeurs négatives aux coordonnées. Le tableau ci-dessous présente un exemple de calcul pour PARAM égal à 5 ( $H=(PARAM+1)/2 = 3$ ):

	NORD	1	2	3	4	5
	$I = \text{mod}(\text{NORD}, H) + 1$	2	3	1	2	3
	$\text{REAL}(2 * (\text{NORD} / (H + 1)) - 1)$	-1,0	-1,0	-1,0	1,0	1,0
	$\text{XPG} = \text{sign}(\text{FPG}(\text{PARAM}, I, 1), \text{REAL}(2 * (\text{NORD} / (H + 1)) - 1))$	-0,538	-0,906	0,000	0,538	0,906
	$\text{HPG} = \text{FPG}(\text{PARAM}, I, 2)$	0,479	0,237	0,569	0,479	0,237

Le même procédé est utilisé pour le calcul des points de Gauss pour les éléments quadrangulaires, mais avec deux différences:

- on prend la division entière et le reste de la division de  $\text{NORD} - 1$  par  $\text{PARAM}$  de façon à obtenir les deux coordonnées (XPG et YPG);
- le poids de chaque point est obtenu par le produit des poids suivant chaque direction.

```

ELSE IF (ALIAS(1:2) .EQ. 'QU') THEN
  H = (PARAM+1)/2
  J = mod(NORD-1, PARAM)+1
  I = mod(J, H)+1

  XPG = sign(FPG(PARAM, I, 1), REAL(2*(J/(H+1))-1))
  HPG = FPG(PARAM, I, 2)

  J = (NORD-1)/PARAM+1
  I = mod(J, H)+1

  YPG = sign(FPG(PARAM, I, 1), REAL(2*(J/(H+1))-1))
  HPG = HPG*FPG(PARAM, I, 2)
[...]
```

Pour les éléments triangulaires, les poids et coordonnées des points de Gauss sont calculées depuis les données du tableau FPGT:

```

REAL*8      FPGT(10, 6, 3)
[...]
```

FPGT(7, 1, 3) = -0.074785022233841D0

FPGT(7, 2, 1) = 0.479308067841920D0  
FPGT(7, 2, 3) = 0.087807628716604D0

FPGT(7, 3, 1) = 0.869739794195568D0  
FPGT(7, 3, 3) = 0.026673617804419D0

FPGT(7, 4, 1) = 0.048690315425316D0  
FPGT(7, 4, 2) = 0.312865496004874D0  
FPGT(7, 4, 3) = 0.038556880445129D0  
[...]

Comme pour les éléments linéiques, les deux premiers indices du tableau représentent l'ordre de la quadrature et la numérotation des points avec des poids distincts. Le troisième indice vaut 1 ou 2 pour les coordonnées et 3 pour le poids.

Les points de Gauss sur un domaine triangulaire présentent une particularité, due à leur symétrie: les points avec les mêmes poids possèdent les mêmes coordonnées barycentriques  $(a, b, c)$  permutées. On a donc trois types de points:

- le barycentre  $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ . On l'appellera point du type A;
- des points appartenant au segment reliant le barycentre à un des sommets. Dans ce cas, deux de leurs coordonnées barycentriques sont égales (si un point ayant pour coordonnées  $(a, a, c)$  est un point de Gauss, les points  $(a, c, a)$  et  $(c, a, a)$  le sont aussi avec les mêmes poids). On les appellera points du type B;
- des points quelconque (dans ce cas, si le point  $(a, b, c)$  est un point de Gauss, les 6 permutations de ces coordonnées donnent des points avec le même poids). On les appellera points du type C.

Le tableau FPGT a été organisé de la façon suivante:

- si les deux coordonnées du point sont nulles, il est le barycentre (type A);
- si la deuxième coordonnée est nulle, il y a un groupe de trois points du type B avec ce poids et la première coordonnée est celle différente des deux autres;
- si les coordonnées ne sont pas nulles, il y a un groupe de six points du type C avec ce poids ayant comme coordonnées leurs permutations.

Ainsi, on voit par exemple que les points de Gauss d'un schéma d'ordre 7 pour un élément triangulaire sont:

- son barycentre A;  
XPG=0,333; YPG=0,333; HPG=-0,075
- deux groupes B  
XPG=0,479; YPG=0,5 - 0,479/2; HPG= 0,088  
XPG=0,5 - 0,479/2; YPG=0,479; HPG= 0,088  
XPG=0,5 - 0,479/2; YPG=0,5 - 0,479/2; HPG= 0,088  
  
XPG=0,870; YPG=0,5 - 0,870/2; HPG= 0,027  
XPG=0,5 - 0,870/2; YPG=0,870; HPG= 0,027  
XPG=0,5 - 0,870/2; YPG=0,5 - 0,870/2; HPG= 0,027
- un groupe de C  
XPG=0,049; YPG=0,313; HPG= 0,039  
XPG=0,313; YPG=0,049; HPG= 0,039  
XPG=0,049; YPG=1-0,049-0,313; HPG= 0,039  
XPG=1-0,049-0,313; YPG=0,049; HPG= 0,039  
XPG=0,313; YPG=1-0,049-0,313; HPG= 0,039  
XPG=1-0,049-0,313; YPG=0,313; HPG= 0,039

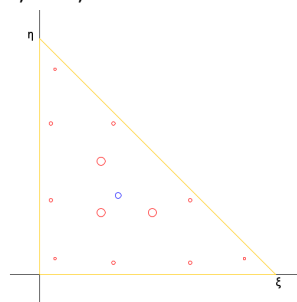


Figure 3: Position et poids des points de Gauss d'ordre 7. Les cercles rouges représentent des valeurs positives et les bleus négatives

On a au total treize points de Gauss.  
La lecture du tableau est faite par le code suivant:

```
ELSE IF (ALIAS(1:2) .EQ. 'TR') THEN
  H=0
  DO WHILE (NORD .GT. 0)
    H=H+1
```

```

IF (FPGT(PARAM,H,1) .EQ. 0) THEN
  NORD = NORD - 1
  I = 0
ELSE IF (FPGT(PARAM,H,2) .EQ. 0) THEN
  NORD = NORD - 3
  I = 1
ELSE
  NORD = NORD - 6
  I = 2
END IF
END DO
NORD = 1-NORD

HPG = FPGT(PARAM,H,3)

IF (I .EQ. 0) THEN
  XPG = 0.3333333333333333D0
  YPG = 0.3333333333333333D0
ELSE IF (I .EQ. 1) THEN
  IF(NORD .EQ. 1) THEN
    XPG = FPGT(PARAM,H,1)
    YPG = 0.5D0-FPGT(PARAM,H,1)/2.D0
  ELSE IF(NORD .EQ. 2) THEN
    XPG = 0.5D0-FPGT(PARAM,H,1)/2.D0
    YPG = FPGT(PARAM,H,1)
  ELSE IF(NORD .EQ. 3) THEN
    XPG = 0.5D0-FPGT(PARAM,H,1)/2.D0
    YPG = XPG
  ELSE
    CALL ASSERT(.FALSE.)
  END IF
ELSE IF (I .EQ. 2) THEN
  IF(NORD .EQ. 1) THEN
    XPG = FPGT(PARAM,H,1)
    YPG = FPGT(PARAM,H,2)
  ELSE IF(NORD .EQ. 2) THEN
    XPG = FPGT(PARAM,H,2)
    YPG = FPGT(PARAM,H,1)
  ELSE IF(NORD .EQ. 3) THEN
    XPG = FPGT(PARAM,H,1)
    YPG = 1.D0-FPGT(PARAM,H,1)-FPGT(PARAM,H,2)
  ELSE IF(NORD .EQ. 4) THEN
    XPG = 1.D0-FPGT(PARAM,H,1)-FPGT(PARAM,H,2)
    YPG = FPGT(PARAM,H,1)
  ELSE IF(NORD .EQ. 5) THEN
    XPG = FPGT(PARAM,H,2)
    YPG = 1.D0-FPGT(PARAM,H,1)-FPGT(PARAM,H,2)
  ELSE IF(NORD .EQ. 6) THEN
    XPG = 1.D0-FPGT(PARAM,H,1)-FPGT(PARAM,H,2)
    YPG = FPGT(PARAM,H,2)
  ELSE
    CALL ASSERT(.FALSE.)
  END IF
ELSE
  CALL ASSERT(.FALSE.)
END IF

```

Dans un premier temps, il faut qu'on sache à quel groupe le NORD-ème point appartient. Dans la boucle, on soustrait une unité de NORD s'il y a un groupe du type A dans le tableau, 3 s'il en y un du type B ou 6 s'il en y a un du type C jusqu'à ce que la valeur de NORD ne soit plus positive. La variable I nous dit à quel type de groupe le point appartient (0-A;1-B;2-C) et la variable H la position du tableau où se trouvent ses coordonnées et son poids. Ensuite, la valeur 1-NORD nous donne l'information sur la permutation des coordonnées correspondant au point donné. Le tableau ci-dessous illustre un exemple de calculs pour PARAM égal à 7 (pour la boucle, ce tableau figure les valeurs de NORD):

NORD	1	2	3	4	5	6	7	8	9	10	11	12	13
H=0 DO WHILE (NORD .GT. 0) H=H+1 IF (FPGT(PARAM,H,1) .EQ. 0) THEN NORD = NORD - 1 I = 0 ELSE IF (FPGT(PARAM,H,2) .EQ. 0) THEN NORD = NORD - 3 I = 1 ELSE NORD = NORD - 6 I = 2 END IF END DO	0	1	2	3	4	5	6	7	8	9	10	11	12
		-2	-1	0	1	2	3	4	5	6	7	8	9
					-2	-1	0	1	2	3	4	5	6
								-5	-4	-3	-2	-1	0
H	1	2	2	2	3	3	3	4	4	4	4	4	4
I	0	1	1	1	1	1	1	2	2	2	2	2	2
NORD = 1-NORD	1	3	2	1	3	2	1	6	5	4	3	2	1

Les quadratures de Simpson et Newton-Cotes utilisent des points équirépartis sur le domaine d'intégration, leurs coordonnées peuvent donc être calculées directement à partir de NORD, comme illustré ci-dessous:

```

ELSE IF (mod(TYPI,10) .EQ. 3) THEN
  PARAM = TYPI/10
  IF (ALIAS(1:2) .EQ. 'SE') THEN
    NORD = NORD-1
    XPG = -1.D0 + REAL(NORD)/PARAM
    YPG = 0.D0

    IF ((NORD .EQ. 0) .OR. (NORD .EQ. 2*PARAM)) THEN
      HPG = 1.D0
    ELSE
      IF (mod(NORD,2) .EQ. 0) THEN
        HPG = 2.D0
      ELSE
        HPG = 4.D0
      END IF
    END IF
  END IF
  HPG = HPG/(PARAM*3.D0)

```

Cette partie du code calcule les coordonnées et poids pour la quadrature de Simpson pour les éléments linéiques. Les points sont espacés d'une distance  $1/\text{PARAM}$  entre eux, donc le NORD-ème point est à la position  $-1+\text{NORD}/\text{PARAM}$  suivant l'axe  $\xi$ . Dans le cas d'une seule subdivision du domaine, on a trois points aux coordonnées  $-1, 0$  et  $1$  avec les poids  $\frac{1}{3}, \frac{4}{3}$  et  $\frac{1}{3}$  respectivement. Lorsqu'il y a plus d'une subdivision, les points se resserrent et les poids se répètent. Lorsque deux points de sous-domaines différents se trouvent à la même position, leurs poids sont sommés. Par exemple, dans le cas avec 2 subdivisions, on a cinq points aux coordonnées  $-1, -\frac{1}{2}, 0, \frac{1}{2}$  et  $1$  avec les poids  $\frac{1}{6}, \frac{4}{6}, \frac{2}{6}, \frac{4}{6}$  et  $\frac{1}{6}$  respectivement. Dans le code, on voit que si le point est à l'extrémité du domaine il reçoit un poids de 1, sinon un poids de 2 si NORD-1 est pair ou 4 s'il est impair, ces poids étant normalisés à la fin en fonction du nombre de subdivisions. Pour les éléments triangulaires et quadrangulaires on a implémenté des algorithmes semblables tout en exploitant les symétries de la quadrature.

La pondération des points de la quadrature de Newton-Cotes pour les éléments linéiques et quadratiques ressemble à celle de la méthode de Gauss, de sorte qu'on ne détaillera pas son implémentation. Ces poids sont stockés dans un tableau appelé PNC.

```

REAL*8      PNC(8,6)
[...]
PNC(1,1)   = 0.25D0
PNC(1,2)   = 0.75D0

PNC(2,1)   = 0.155555555555556D0
PNC(2,2)   = 0.711111111111111D0
PNC(2,3)   = 0.266666666666667D0

PNC(3,1)   = 0.131944444444444D0
PNC(3,2)   = 0.520833333333333D0
PNC(3,3)   = 0.347222222222222D0
[...]
```

Pour les éléments triangulaires, on a un autre tableau appelé PNCT:

```

REAL*8      PNCT(6,10)
[...]
PNCT(1,1)  = 0.016666666666667D0
PNCT(1,2)  = 0.0375D0
PNCT(1,3)  = 0.225D0

PNCT(2,1)  = 0.D0
PNCT(2,2)  = 0.0444444444444445D0
PNCT(2,3)  = -0.011111111111111D0
PNCT(2,4)  = 0.088888888888889D0
[...]
```

La partie du code effectuant la lecture de ce tableau est affiché ci-dessous:

```

ELSE IF (ALIAS(1:2) .EQ. 'TR') THEN
  H=0
  DO WHILE (NORD .GT. 0)
    H=H+1
    NORD=NORD-H
  END DO
  I = -NORD
  J = H-1+NORD

  XPG = REAL(I)/PARAM
  YPG = REAL(J)/PARAM

  H=min(I, J, PARAM-I-J)
  I=PARAM-max(I, J, PARAM-I-J)

  J = PARAM/2
  IF (I .LE. J) THEN
    HPG=PNCT(PARAM-2, ((I+1)/2)*(I/2+1)+1+H)
  ELSE
    HPG=PNCT(PARAM-2, ((I+1)/2)*(I/2+1)+1+H
&          -(I-(PARAM/2))*(I-((PARAM-1)/2)))
  END IF
```

Le calcul des variables I et J a été fait de façon à obtenir la numérotation donnée dans la figure ci-dessous:

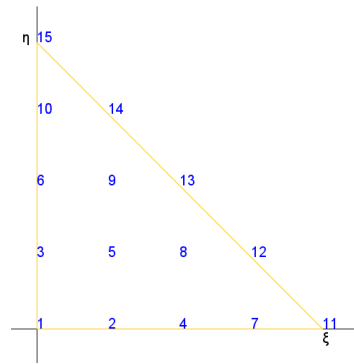


Figure 4: Numérotation des points de la quadrature de Newton-Cotes d'ordre 4. On remarque que cette numérotation est la même utilisée pour la quadrature de Simpson (avec 2 subdivisions).

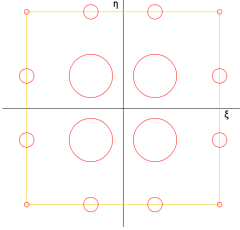
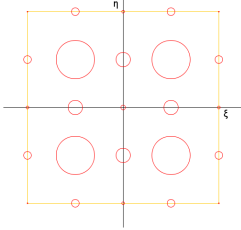
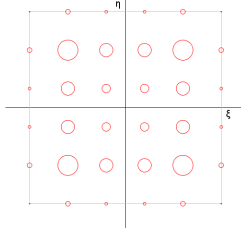
Ensuite, la pondération est faite par le calcul des deux coordonnées barycentriques non-normalisées (c.a.d. la somme des coordonnées ne donne pas 1) qui sont stockées dans les variables H et I.

Le tableau ci-dessous illustre les poids et les position des points d'intégrations pour les quadratures de Gauss, Newton-Cotes et Simpson. Les graphiques ont été créés avec un logiciel en Java obtenu à partir de la traduction du fichier mmgau.s.f en ce langage associé à un module graphique. Pour les éléments surfaciques, les cercles rouges représentent des valeurs positives et les bleus des valeurs négatives.

éléments	quadratures	Gauss (3) Simpson (1) Newton-Cotes (3)	Gauss (4) Simpson (2) Newton-Cotes (4)	Gauss (5) Simpson (3) Newton-Cotes (5)
SE	Gauss			
	Simpson			



	Newton-Cotes			
TR	Gauss			
	Simpson			
	Newton-Cotes			
QU	Gauss			
	Simpson			

	<p>Newton-Cotes</p>			
--	---------------------	-----------------------------------------------------------------------------------	------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

# Validation

## Présentation du cas test retenu

Le cas test développé sert à évaluer la qualité des résultats de Code\_Aster en ce qui concerne le traitement du contact entre deux structures où on a une compatibilité entre les mailles à l'instant initial et différentes positions avec incompatibilité au cours du temps. Il permet de valider le traitement du contact avec la méthode continue en prenant en compte des grandes rotations.

On considère une structure composée de deux couronnes concentriques. La couronne externe est soumise à une pression uniforme sur son bord libre alors qu'on impose une rotation finie à la couronne intérieure. Leur rigidité, représentée par les modules d'Young joue un rôle important dans l'évaluation de la valeur des déformations et des fluctuations de la pression de contact au cours de la rotation. On cherche aussi à savoir quels sont les effets de l'utilisation d'un maillage d'ordre supérieur (ici quadratique) sur la qualité de la solution.

La figure ci-dessous illustre le problème étudié: la structure est composée de deux couronnes circulaires concentriques. Le rayon interne  $R_2$  de la couronne extérieure est égal au rayon externe de la couronne intérieure.

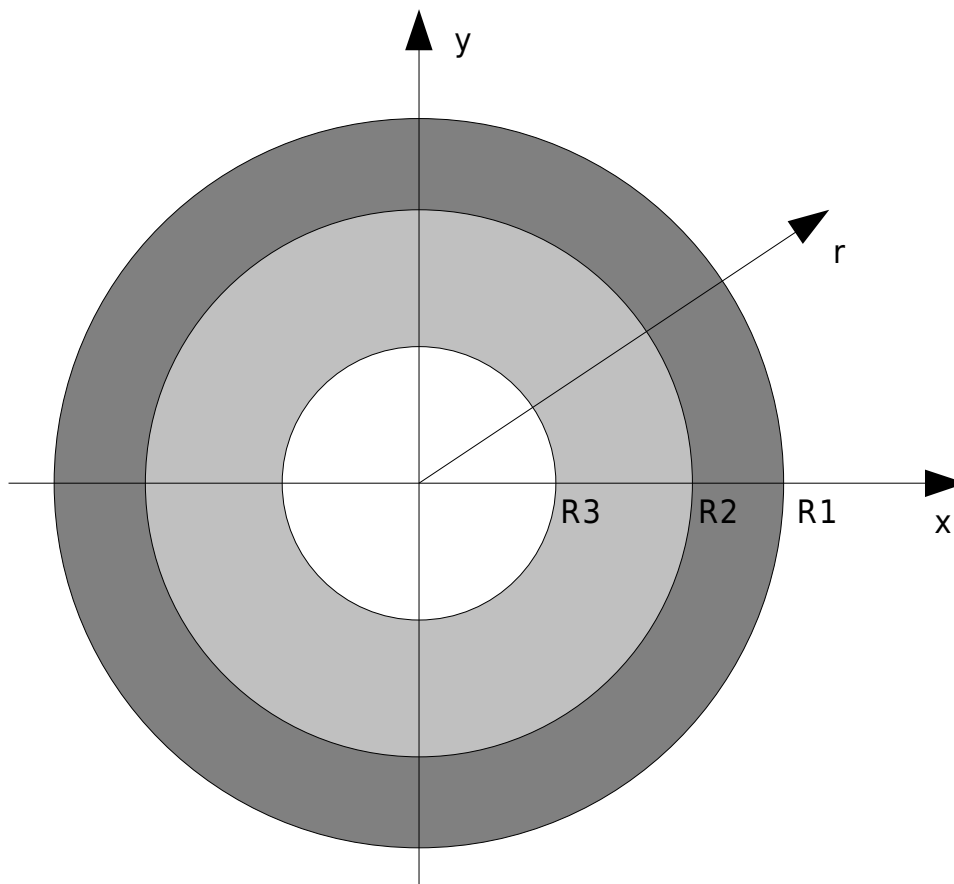


Figure 5: Géométrie du problème

Le module d'Young et le coefficient de Poisson du matériau de la couronne extérieure sont donnés par  $E_1$  et  $\nu_1$  (respectivement  $E_2$  et  $\nu_2$  pour la couronne intérieure).

## **Solution analytique du problème**

Nous développons ici une solution analytique pour le problème présenté ci-dessus. Cette solution sera développée sur l'hypothèse de petites déformations et en considérant que les matériaux des couronnes sont isotropes, régis par une loi élasticité linéaire et que tout se déroule en régime élastique et sans variation de température.

### **Cas 1 : contraintes planes**

Nous supposons d'abord qu'il n'y a pas de contraintes suivant la direction perpendiculaire au plan des couronnes ( $\underline{\sigma}_i \cdot \underline{e}_z = \underline{0}$ ). Les symétries du problème nous amènent à un champ de contraintes qui peut être écrit sous la forme:

$$\underline{\sigma}_i(r) = \sigma_{rr;i} \underline{e}_r \otimes \underline{e}_r + \sigma_{\theta\theta;i} \underline{e}_\theta \otimes \underline{e}_\theta$$

où l'indice  $i$  vaut 1 pour la couronne extérieure et 2 pour la couronne intérieure. En l'absence de forces volumiques et en considérant le problème quasi-statique (on néglige les effets de l'accélération), on a :

$$\text{div } \underline{\sigma} = \underline{0}$$

d'où l'on obtient :

$$\sigma_{rr;i}(r) = A_i - \frac{B_i}{r^2}$$

$$\sigma_{\theta\theta;i}(r) = A_i + \frac{B_i}{r^2}$$

$$\underline{\sigma}_i(r) = A_i \underline{1} - \frac{B_i}{r^2} (\underline{e}_r \otimes \underline{e}_r - \underline{e}_\theta \otimes \underline{e}_\theta)$$

Il ne nous reste plus qu'à calculer les valeurs de  $A_i$  et  $B_i$  pour chacune des couronnes. On notera  $\lambda_n$  la pression de contact entre les deux couronnes telle que :

$$\underline{\sigma}_1(R_2) \cdot (-\underline{e}_r) = \lambda_n \underline{e}_r$$

$$\underline{\sigma}_2(R_2) \cdot \underline{e}_r = -\lambda_n \underline{e}_r$$

avec les conditions aux limites :

$$\underline{\sigma}_1(R_1) \cdot \underline{e}_r = -p \cdot \underline{e}_r$$

$$\underline{\sigma}_2(R_3) \cdot (-\underline{e}_r) = \underline{0}$$

on a :

$$A_1 = \frac{-\rho R_1^2 + \lambda_n R_2^2}{R_1^2 - R_2^2}; B_1 = (-\rho + \lambda_n) \frac{R_1^2 R_2^2}{R_1^2 - R_2^2}$$

$$A_2 = -\lambda_n \frac{R_2^2}{R_2^2 - R_3^2}; B_2 = -\lambda_n \frac{R_2^2 R_3^2}{R_2^2 - R_3^2}$$

par ailleurs la loi de comportement des matériaux permet d'écrire :

$$\underline{\underline{\epsilon}}_i = \frac{1+\nu_i}{E_i} \underline{\underline{\sigma}}_i - \frac{\nu_i}{E_i} Tr(\underline{\underline{\sigma}}_i) \underline{\underline{1}} = \frac{1}{E_i} [A_i(1-\nu_i) \underline{\underline{1}} - B_i(1+\nu_i) \frac{1}{r^2} (\underline{\underline{e}}_r \otimes \underline{\underline{e}}_r - \underline{\underline{e}}_\theta \otimes \underline{\underline{e}}_\theta)]$$

$$= \frac{\partial u_{r;i}}{\partial r} \underline{\underline{e}}_r \otimes \underline{\underline{e}}_r + \frac{1}{2} \left( \frac{1}{r} \frac{\partial u_{r;i}}{\partial \theta} + \frac{\partial u_{\theta;i}}{\partial r} - \frac{u_{\theta;i}}{r} \right) (\underline{\underline{e}}_r \otimes \underline{\underline{e}}_\theta + \underline{\underline{e}}_\theta \otimes \underline{\underline{e}}_r) + \left( \frac{1}{r} \frac{\partial u_{\theta;i}}{\partial \theta} + \frac{u_{r;i}}{r} \right) \underline{\underline{e}}_\theta \otimes \underline{\underline{e}}_\theta$$

ce qui permet d'obtenir :

$$\underline{\underline{u}}_i = \frac{1}{E_i} [A_i(1-\nu_i)r + B_i(1+\nu_i) \frac{1}{r}] \underline{\underline{e}}_r = f_{CP}(r) \underline{\underline{e}}_r$$

Pour obtenir la valeur de  $\lambda_n$ , on impose la continuité du vecteur déplacement en  $r=R_2$  :

$$\underline{\underline{u}}_1(R_2) = \underline{\underline{u}}_2(R_2)$$

$$\frac{1}{E_1} [A_1(1-\nu_1)R_2 + B_1(1+\nu_1) \frac{1}{R_2}] \underline{\underline{e}}_r = \frac{1}{E_2} [A_2(1-\nu_2)R_2 + B_2(1+\nu_2) \frac{1}{R_2}] \underline{\underline{e}}_r$$

$$\frac{1}{E_1} \left[ \frac{-\rho R_1^2 + \lambda_n R_2^2}{R_1^2 - R_2^2} (1-\nu_1)R_2 + (-\rho + \lambda_n) \frac{R_1^2 R_2^2}{R_1^2 - R_2^2} (1+\nu_1) \frac{1}{R_2} \right] \underline{\underline{e}}_r$$

$$= \frac{1}{E_2} \left[ -\lambda_n \frac{R_2^2}{R_2^2 - R_3^2} (1-\nu_2)R_2 - \lambda_n \frac{R_2^2 R_3^2}{R_2^2 - R_3^2} (1+\nu_2) \frac{1}{R_2} \right] \underline{\underline{e}}_r$$

soit encore :

$$\lambda_n = \frac{2\rho R_1^2}{R_1^2(1+\nu_1) + R_2^2(1-\nu_1) + \frac{E_1}{E_2} \frac{R_1^2 - R_2^2}{R_2^2 - R_3^2} (R_2^2(1-\nu_2) + R_3^2(1+\nu_2))}$$

## Cas 2 : déformations planes

Une fois que l'on a calculé la valeur des déplacements et de la pression de contact pour le cas en contraintes planes, on peut aisément calculer ces valeurs en déformations planes en remplaçant les valeur du module d'Young et le coefficient de Poisson:

$$E_{DP} = \frac{E_{CP}}{1 - \nu_{CP}^2}; \nu_{DP} = \frac{\nu_{CP}}{1 - \nu_{CP}^2}$$

On obtient ainsi les valeurs des déplacements:

$$\underline{u}_i = \frac{1 + \nu_i}{E_i} \left[ A_i (1 - 2 \nu_i) r + \frac{B_i}{r} \right] \underline{e}_r = f_{DP}(r) \underline{e}_r$$

et de la pression de contact:

$$\lambda_n = \frac{2 p R_1^2 (1 - \nu_1)}{R_1^2 + R_2^2 (1 - 2 \nu_1) + \frac{E_1}{E_2} \frac{1 + \nu_2}{1 + \nu_1} \frac{R_1^2 - R_2^2}{R_2^2 - R_3^2} (R_2^2 (1 - 2 \nu_2) + R_3^2)}$$

On remarque que les valeurs de  $A_i$  et  $B_i$  demeurent inchangées.

## Rotation de la couronne intérieure

Outre le déplacement radial, on impose sur la surface interne de la couronne intérieure ( $r=R_3$ ) un déplacement entraînant la rotation de celle-ci. L'expression du déplacement doit prendre en compte la contraction de la couronne intérieure due à l'application de la pression, son nouveau rayon interne étant  $R_3 + f(R_3)$  :

$$\xi_x(X, Y, i) = (R_3 + f(R_3)) \cos(\arctan(\frac{Y}{X}) - \omega(i)) - X$$

$$\xi_y(X, Y, i) = (R_3 + f(R_3)) \sin(\arctan(\frac{Y}{X}) - \omega(i)) - Y$$

éq 1.6

$$\omega(i) = \frac{2 \pi}{N_t N_e} i, i \in \{n \in \mathbb{Z}, 0 \leq n \leq N_t\}$$

où  $N_t$  est le nombre de pas de temps et  $N_e$  le nombre d'éléments le long de la surface de contact. En effet, on impose une rotation d'un angle  $\frac{2 \pi}{N_e}$ , ce qui veut dire qu'au dernier instant on aura une configuration semblable à la configuration initiale mais où la couronne intérieure aura été décalée d'un élément.

## Simulations et conclusions

Afin de valider ce cas test, on a proposé sept simulations dans lesquelles on essaie de mettre en évidence certains aspects du problème traité. La fiche descriptive de ce cas test est disponible en annexe de ce rapport.

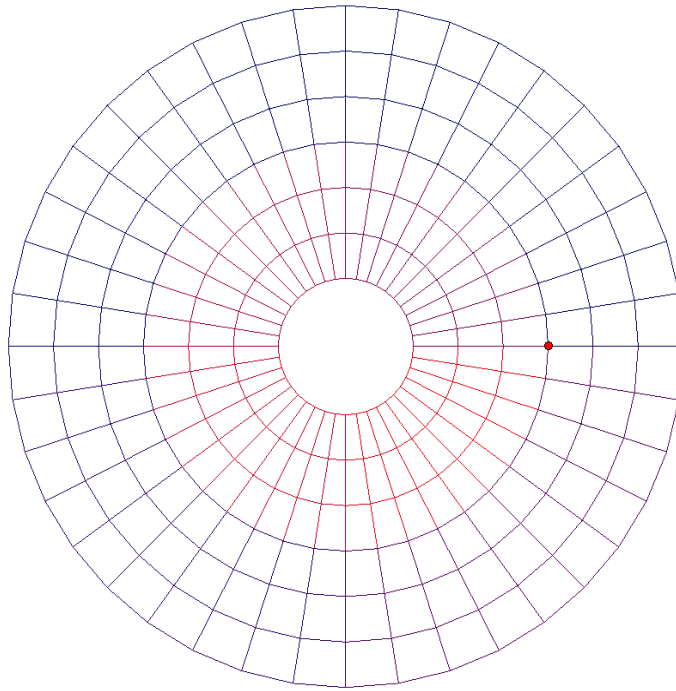


Figure 6: Le maillage des modélisations. Le point sur la figure indique le nœud dont les valeurs de la pression de contact seront évaluées.

Les deux premières modélisations (nommées A et B) ont été élaborées afin de valider la solution analytique développée et de bien savoir quel est son domaine de validité en fonction de la charge appliquée.

Dans un premier temps on n'impose pas la rotation de la couronne intérieure. Un chargement de pression augmentant exponentiellement avec le temps est appliqué au bord de la couronne extérieure et on teste la valeur de la pression de contact à l'interface des couronnes. L'idée est de savoir jusqu'à quelle valeur de pression les résultats obtenus peuvent être considérés comme valables par rapport à la solution analytique une fois que celle-ci a été construite sur l'hypothèse des petites déformations. La simulation A traite la problème en contraintes planes alors que la simulation B le fait en déformations planes.

On pose les valeurs ci-dessous pour les simulations:

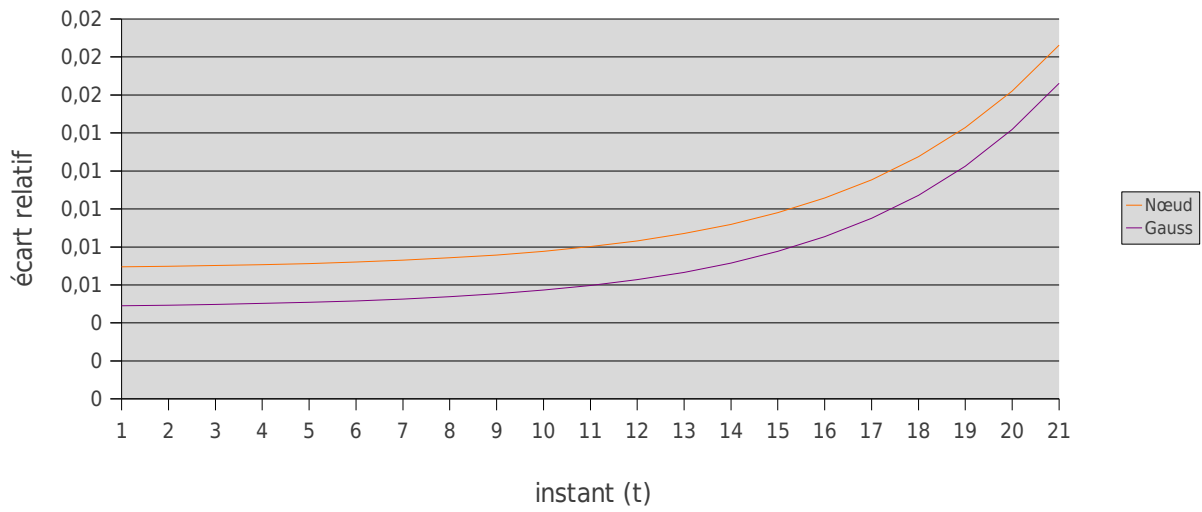
$$E_1=1,0GPa; E_2=1,0GPa; \nu_1=0,2; \nu_2=0,2; R_1=1,0; R_2=0,6; R_3=0,2$$

La valeur de la pression appliquée au bord à chaque pas de temps est donnée par la formule:

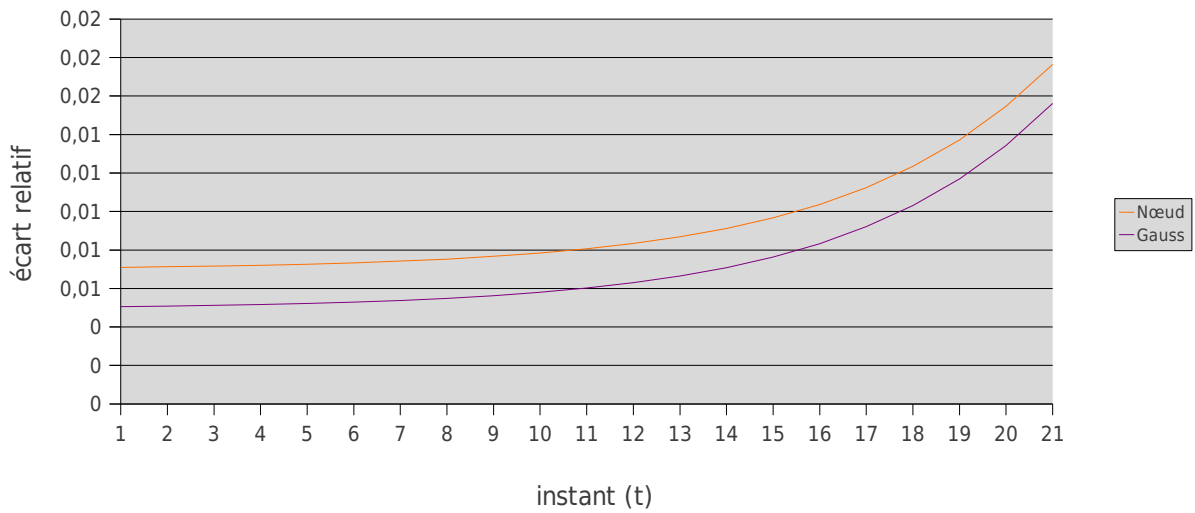
$$p(t)=p_0 10^{\frac{t}{10}-1,1}, t \in \{n \in \mathbb{Z}, 1 \leq n \leq 21\}, p_0=1,0MPa$$

pour  $t = 1$ , la pression vaut 0,1MPa et pour  $t = 21$  elle vaut 10MPa. On remarque que les simulations ont été faites en négligeant les effets de l'accélération (commande STAT\_NON\_LINE) ce qui implique que les pas de temps possèdent des unités arbitraires. Le graphique ci-dessous indique l'écart relatif de la pression de contact calculée entre les deux couronnes par rapport à la valeur obtenue analytiquement pour deux méthodes d'intégration différentes (NOEUD et GAUSS).

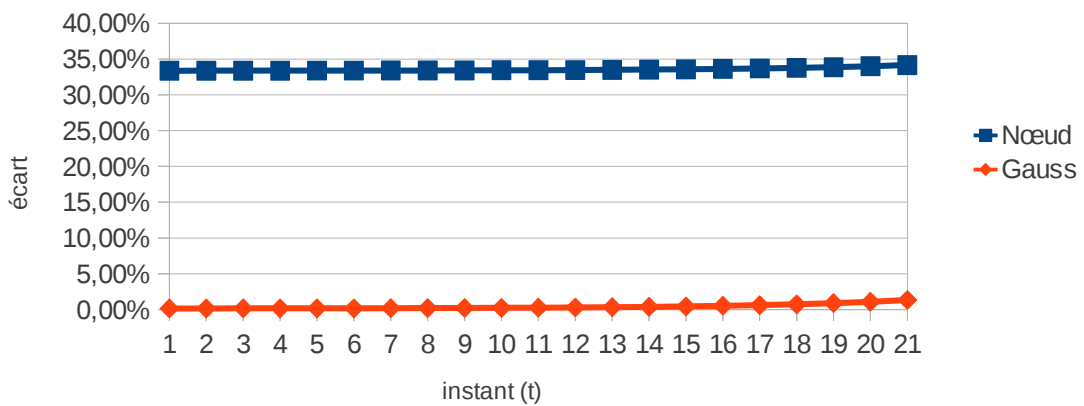
Écart relatif de la pression de contact calculée (contraintes planes)



Écart relatif de la pression de contact calculée (déformation planes)



Ecart relatif de la pression de contact calculée (contraintes planes, éléments quadratiques)

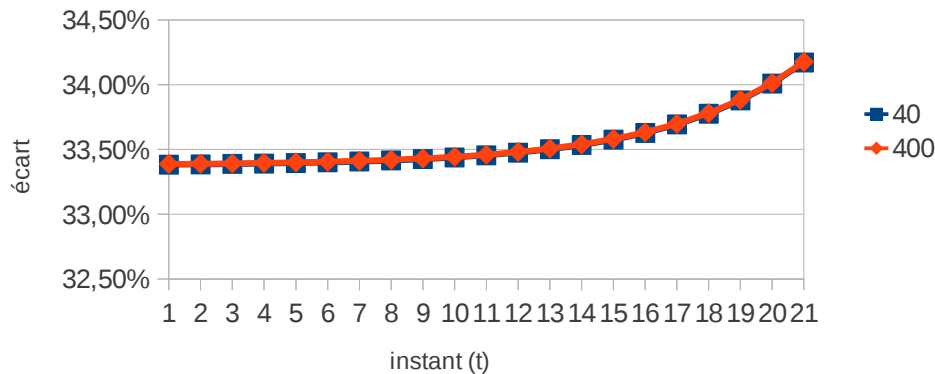


On voit qu'avec des éléments quadratiques, les résultats calculés avec la méthode d'intégration aux nœuds sont très écartés par rapport à la solution analytique.



Le schéma d'intégration aux nœuds utilisés dans Code\_Aster pour les éléments quadratiques de contact est un schéma sous-intégré (il permet d'intégrer exactement un polynôme de degré 2 alors que l'on cherche à estimer un ordre 4). Le raffinement ne permet plus d'obtenir une solution exacte. Les résultats sont faux. Il faut donc avoir la possibilité d'utiliser d'autres schémas d'intégration. Le graphique ci-dessous nous montre que, même qu'on raffine plus le maillage, les résultats demeurent écartés. Les chiffres dans la légende indiquent le nombre d'éléments le long de la surface de contact.

Ecart relatif de la pression de contact calculée  
(contraintes planes, intégration aux nœud)



Pour les modélisations C et D on travaille avec la solution en contraintes planes, on garde une valeur constante pour la pression appliquée et on impose la rotation de la couronne intérieure. On considère qu'il n'y a pas de frottement entre les surfaces. La modélisation C diffère de la D par l'emploi des matériaux avec les modules de Young et coefficients de Poisson différents.

Simulation	$E_1$	$\nu_1$	$E_2$	$\nu_2$
C	1,0GPa	0,2	1,0GPa	0,2
D	1,0GPa	0,3	0,1GPa	0,2

Lorsqu'on a une compatibilité entre les éléments du maillage à l'interface des couronnes, la valeur calculée pour la pression de contact est bien celle qu'on obtiendrait avec de la modélisation sans rotation. Au fur et à mesure qu'on impose la rotation, les éléments de la surface se retrouvent dans une configuration qui ressemble à celle de deux polygones, l'un inscrit dans l'autre, comme illustré sur la figure ci-dessous. Afin de respecter les conditions d'impénétrabilité les deux surfaces doivent trouver une configuration d'accommodation : la surface de contact de la couronne extérieure doit être plus large et celle de la couronne intérieure plus petite pour qu'elles puissent s'insérer l'une dans l'autre. Le rapport entre le taux d'élargissement et de rétraction des couronnes dépend essentiellement des propriétés des matériaux (par exemple, si le matériau de la couronne extérieure est beaucoup plus rigide, celle-ci ne changera guère, alors que la couronne intérieure doit se déformer afin de respecter la condition d'impénétrabilité). Ces déformations entraîneront une fluctuation de la valeur de la pression de contact calculée.

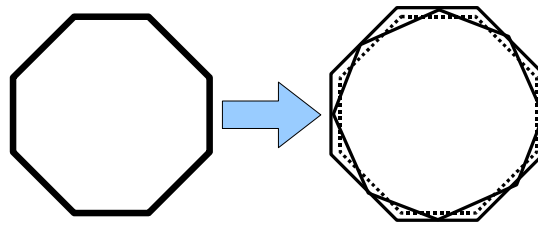


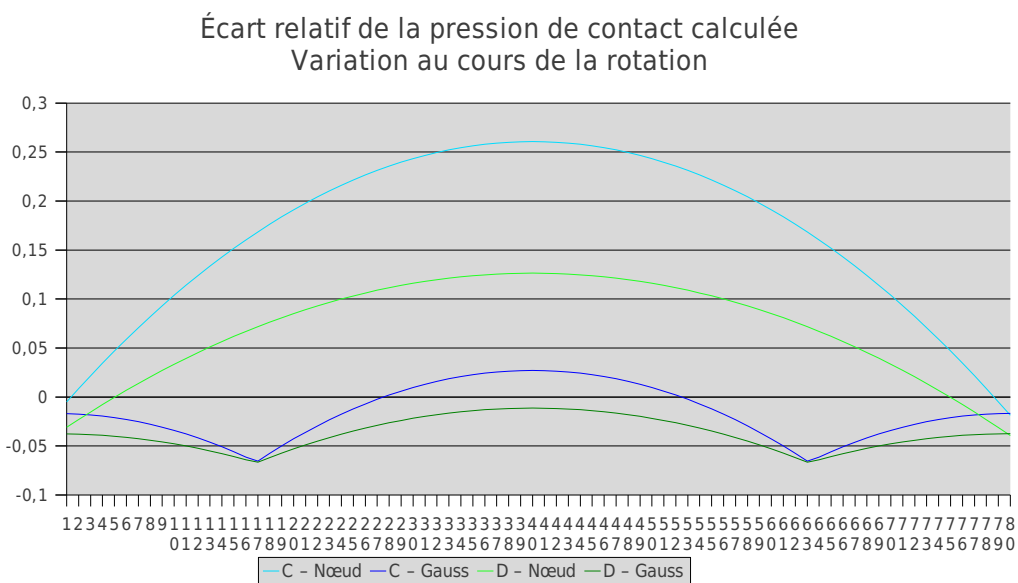
Figure 7: l'incompatibilité des mailles lors de la rotation

Cette incompatibilité entre les deux surfaces est inhérente au modèle : même si le maillage est très discrétisé, on ne peut pas mailler un cercle parfait avec des éléments finis linéaires. Pour autant on peut (et il en est de notre intérêt) réduire les effets issus de cette incompatibilité en utilisant soit des éléments d'ordre supérieur soit un maillage plus fin.

Comme nous l'avons mentionné ci-dessus, la rigidité de la couronne induit directement la valeur des contraintes nécessaires aux déformations pendant la rotation. Lorsque les déplacements deviennent trop importants du fait de la faible rigidité des couronnes, la valeur calculée s'écarte de la valeur analytique, développée sur l'hypothèse de petits déformations.

D'un autre côté, lorsque la rigidité des couronnes est trop importante, la fluctuation de la pression de contact augmente sensiblement. Ceci vient du fait que lorsqu'on impose un déplacement sur une structure, les contraintes auxquelles celle-ci est soumise peuvent être élevées. Pour une structure très rigide, un petit déplacement est possible seulement avec des contraintes considérables.

Cette variation au cours de la rotation est illustrée sur le graphique ci-dessous. Aux instants de temps 0 et 80 tous les éléments en vis-à-vis sont en compatibilité. Comme pour les simulations A et B, on a calculé les valeurs avec deux méthodes d'intégration différentes (NOEUD et GAUSS).



On remarque qu'on a obtenu des résultats plus proches de la solution analytique pour la simulation D (couronne intérieure moins rigide par rapport à la simulation C) et que les points de minimum sur l'erreur des courbes obtenues avec la méthode

d'intégration de Gauss sont calculés lorsque les nœuds de la surface intérieure sont face aux points de Gauss des éléments de la surface extérieure (aux instants 17 et 63).

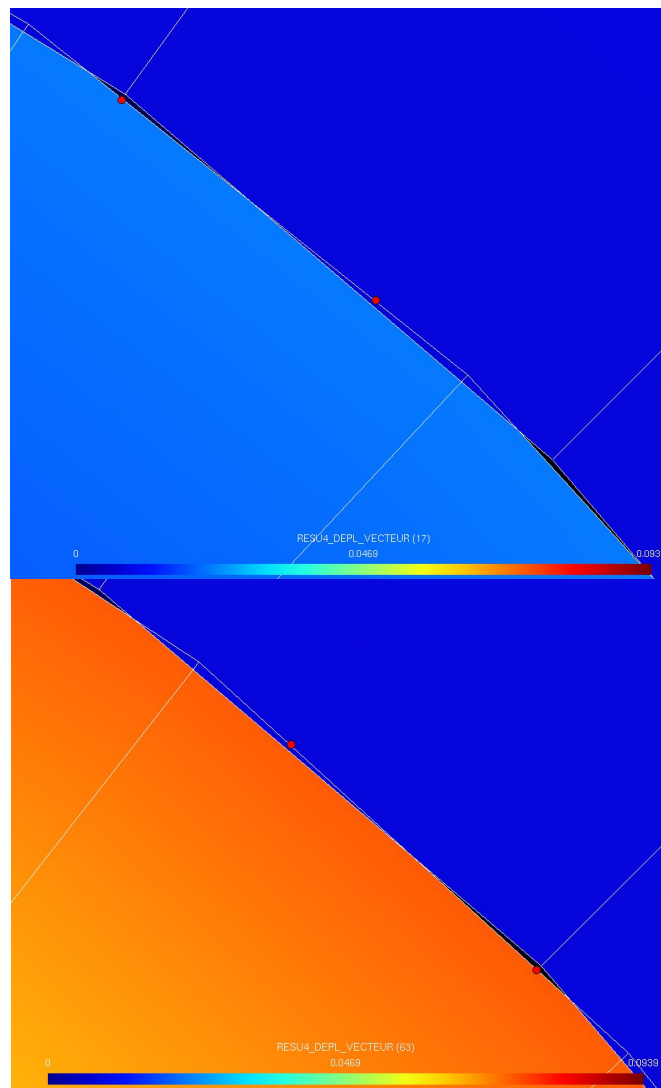
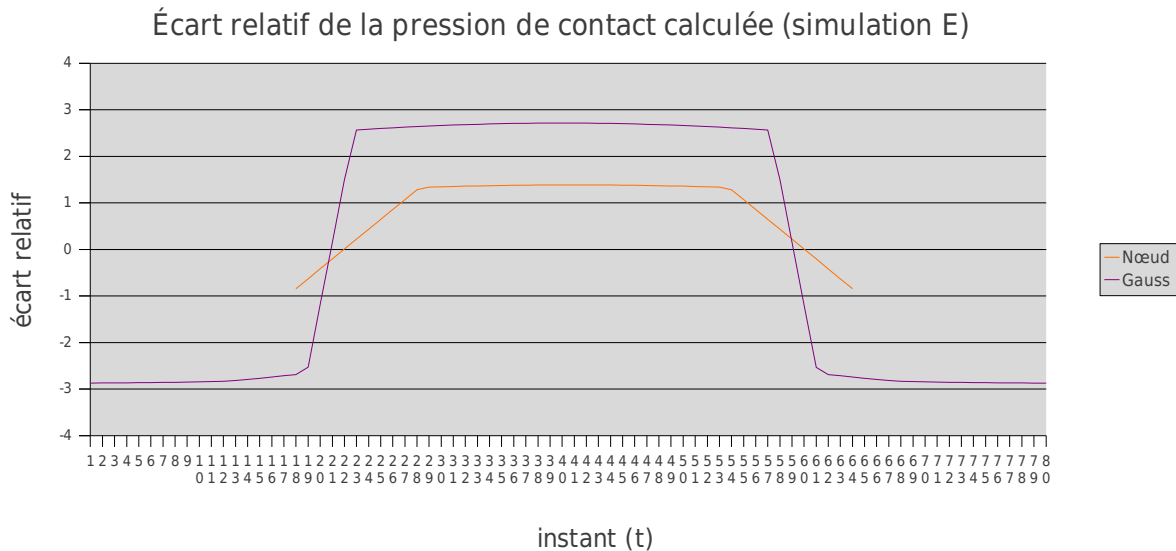
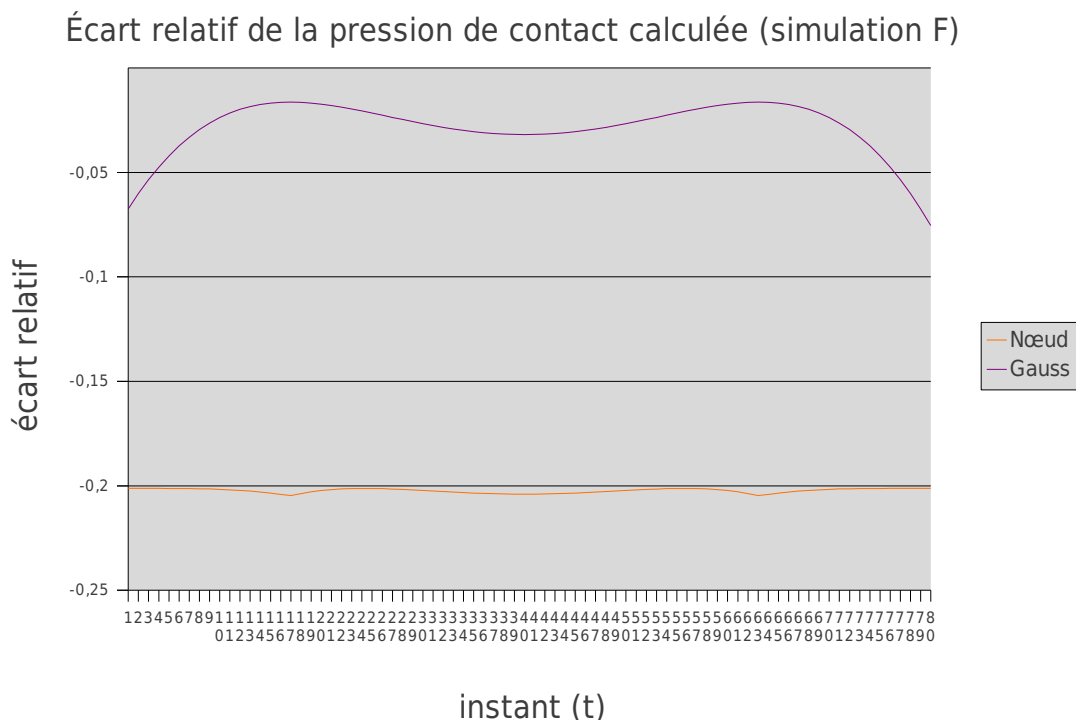


Figure 8: Simulation C, quadrature de Gauss, instants 17 (en haut) et 63 (en bas). Les points en rouge sont les points de Gauss de l'élément de la surface de la couronne intérieure, calculés avec les coordonnées en pixels de la figure. Dans ces configurations on calcule la plus petite valeur pour la pression de contact

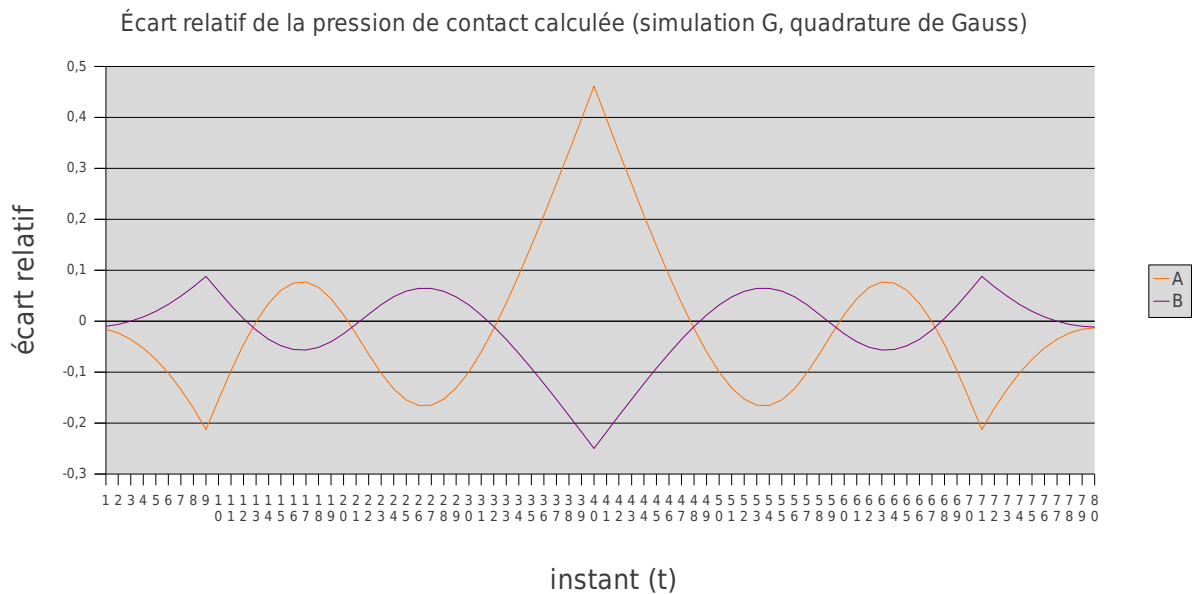
Pour les modélisations E, F et G, toujours en contraintes planes et à pression constante, on travaille avec des maillages quadratiques soit pour la couronne intérieure (E), soit pour la couronne extérieure (F), soit pour l'ensemble (G).



Dans la simulation E, du fait que la surface intérieure est composée d'éléments quadratiques (même nombre d'éléments mais 50% de nœuds de plus le long de la surface de contact) certains nœuds se détachent de la surface extérieure au cours de la rotation. Si on utilise la méthode d'intégration avec nœuds, on calculera une valeur nulle pour la pression de contact pendant quelques instants. Si on utilise la méthode de Gauss, la valeur de la pression aux nœuds de l'élément qui sont en contact compense la valeur nulle aux nœud qui ne le sont pas, pour finalement obtenir des résultats totalement écartés de la solution analytique.



Dans la simulation F, on remarque que la méthode par nœud commence à montrer son insuffisance : le produit de leur fonctions de forme des éléments d'ordre 1 et 2 donne des polynômes d'ordre 3, alors que l'intégration nodale ne permet que d'intégrer un polynôme d'ordre 2, ce schéma étant dans ce cas sous-intégré, ce qui explique l'écart observé.



*Figure 9: l'élément SE3, présent le long de la surface de contact des couronnes dans la simulation G. Sur le graphique ci-dessus, les valeurs « A » sont calculées aux les sommets (les nœuds N1 et N2) et les valeurs « B » aux nœuds N3.*

Lorsqu'on travaille avec des éléments quadratiques et qu'on choisit la quadrature de Gauss, on augmente de 2 à 3 le nombre des points de Gauss. En regardant les courbes de la simulation G, on observe le même phénomène déjà mentionné pour la simulation D: les points d'extremum de ces courbes (aux instants 9, 40 et 71) sont calculés lorsque les nœuds couronne intérieure sont face aux points de Gauss des éléments de la surface de contact de la couronne extérieure.

Pour l'ensemble de tests E, F et G on vérifie que la fluctuation de la pression de contact est plus petite, mais que la méthode d'intégration par défaut d'Aster pour le contact (intégration par nœud) donne des résultats différents des autres quadratures (Gauss, Simpson et Newton-Cotes).

## Bibliographie

- 1 ABBAS M. : Eléments de contact dérivés d'une formulation hybride continue. Manuel de référence de Code\_Aster. 2009.
- 2 MASSIN P. : Modélisation et simulation numérique de structures avec interfaces. Habilitation a diriger des recherches, Université Paris 6. 2010.
- 3 LE TALLEC P. : Modélisation et calcul des milieux continus. Éditions de l'Ecole Polytechnique. 2009.
- 4 ABRAMOWITZ M., STEGUN I. : Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables 10<sup>ème</sup> édition. Editeur : National Bureau of Standards. 1972.
- 5 DUNAVANT D. A. : High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle. International Journal for numerical methods in Engineering, Vol. 21, pag. 1129-1148. 1985.
- 6 SILVESTER P. : Symmetric Quadrature Formulae for Simplexes. Mathematics of Computation, vol. 24, No. 109. Editeur : American Mathematical Society. 1970.

# Annexe A : fichier de commande SSNP152

Pour chaque modélisation proposée pour le cas test SSNP152 on a un fichier de commande spécifique. On présente ci-dessous celui de la simulation A:

```
# MODIF DATE 26/05/2011 AUTEUR P. PEREIRA
# CAS TEST : SSNP152
# TITRE INCLUSION DE DEUX COURONNES
# CONFIGURATION MANAGEMENT OF EDF VERSION
# =====
# COPYRIGHT (C) 1991 - 2009 EDF R&D WWW.CODE-ASTER.ORG
# THIS PROGRAM IS FREE SOFTWARE; YOU CAN REDISTRIBUTE IT AND/OR MODIFY
# IT UNDER THE TERMS OF THE GNU GENERAL PUBLIC LICENSE AS PUBLISHED BY
# THE FREE SOFTWARE FOUNDATION; EITHER VERSION 2 OF THE LICENSE, OR
# (AT YOUR OPTION) ANY LATER VERSION.
#
# THIS PROGRAM IS DISTRIBUTED IN THE HOPE THAT IT WILL BE USEFUL, BUT
# WITHOUT ANY WARRANTY; WITHOUT EVEN THE IMPLIED WARRANTY OF
# MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SEE THE GNU
# GENERAL PUBLIC LICENSE FOR MORE DETAILS.
#
# YOU SHOULD HAVE RECEIVED A COPY OF THE GNU GENERAL PUBLIC LICENSE
# ALONG WITH THIS PROGRAM; IF NOT, WRITE TO EDF R&D CODE_ASTER,
# 1 AVENUE DU GENERAL DE GAULLE, 92141 CLAMART CEDEX, FRANCE.
# =====
# ELEMENTS LINEAIRES, GRANDES ROTATIONS, CONTRAINTES PLANES,
# COURONNE EXTERIEURE MAITRE, SANS ROTATION, CHARGEMENT PROGRESSIF EXPONENTIEL
# p=1.0E6, E1=1.0E9, E2=1.0E9, nu1=0.2, nu2=0.2, R1 = 1.0, R2 = 0.6, R3 = 0.2

DEBUT(CODE=_F(NOM='SSNP152',NIV_PUB_WEB='INTERNET'))

PRE_GMSH();

MA=LIRE_MALLAGE(VERI_MAIL=_F(VERIF='OUI'),);

#+-----+
#| noms des groupes de noeuds et mailles |
#+-----+

MA=DEFI_GROUP(reuse=MA,MAILLAGE=MA,CREA_GROUP_MA=( _F(GROUP_MA='GM100',NOM='C1'),
                                                    _F(GROUP_MA='GM101',NOM='C2'),
                                                    _F(GROUP_MA='GM200',NOM='S1R1'),
                                                    _F(GROUP_MA='GM300',NOM='S1R2'),
                                                    _F(GROUP_MA='GM400',NOM='S2R2'),
                                                    _F(GROUP_MA='GM500',NOM='S2R3')),
CREA_GROUP_NO=( _F(GROUP_MA='S1R2',NOM='CONT_N_1'),
                _F(GROUP_MA='S2R2',NOM='CONT_N_2'),),);

MA=MODI_MALLAGE(reuse=MA,MAILLAGE=MA,ORIE_PEAU_2D=(
                                                    _F(GROUP_MA = 'S1R1',),
                                                    _F(GROUP_MA = 'S1R2',),
                                                    _F(GROUP_MA = 'S2R2',),
                                                    _F(GROUP_MA = 'S2R3',),),);

#+-----+
#| propriétés des matériaux |
#+-----+

E1=1.0E9
E2=1.0E9
nu1=0.2
nu2=0.2

MAT1=DEFI_MATERIAU(ELAS = _F(E=E1,NU = nu1,));
MAT2=DEFI_MATERIAU(ELAS = _F(E=E2,NU = nu2,));

CHMAT=AFFE_MATERIAU(MAILLAGE=MA,AFFE=( _F(GROUP_MA = 'C1',MATER = MAT1,),
                                         _F(GROUP_MA = 'C2',MATER = MAT2,),),);

#+-----+
#| modèle |
#+-----+

M0=AFFE_MODELE(MAILLAGE=MA,AFFE=_F(TOUT = 'OUI',PHENOMENE = 'MECANIQUE',MODELISATION = 'C_PLAN'));

#+-----+
#| conditions aux limites |
#+-----+

# pression externe

p=1.E6;

# géométrie

R1 = 1.0;
R2 = 0.6;
R3 = 0.2;
```

```

# calcul analytique de la pression de contact
pres_cont = (2*p*R1*R1)/((R2*R2*(1-nu1)+R1*R1*(1+nu1))+((E1/E2)*((R1*R1-R2*R2)/(R2*R2-R3*R3))*(R2*R2*(1-nu2)+R3*R3*(1+nu2))));

#
# calcul des déplacements auxquels les bords des couronnes seront soumis
#

# couronne externe
a1=(pres_cont*R2*R2-p*R1*R1)/(R1*R1-R2*R2);
b1=(pres_cont-p)*(R1*R1*R2*R2)/(R1*R1-R2*R2);

eps_r_1 = (1/E1)*(a1*(1-nu1)*R1+b1*(1+nu1)/R1);

ux1 = FORMULE(NOM_PARA=('X','Y',), VALE='eps_r_1*cos(atan2(Y,X))');
uy1 = FORMULE(NOM_PARA=('X','Y',), VALE='eps_r_1*sin(atan2(Y,X))');

CH1=AFFE_CHAR_MECA_F(MODELE=M0, DDL_IMP0=( _F(GROUP_MA = 'S1R1', DX=ux1,DY=uy1,)),);

# couronne interne
a2=-pres_cont*R2*R2/(R2*R2-R3*R3);
b2=-pres_cont*(R2*R2*R3*R3)/(R2*R2-R3*R3);

eps_r_2 = (1/E2)*(a2*(1-nu2)*R3+b2*(1+nu2)/R3);

ux2 = FORMULE(NOM_PARA=('X','Y',), VALE='eps_r_2*cos(atan2(Y,X))');
uy2 = FORMULE(NOM_PARA=('X','Y',), VALE='eps_r_2*sin(atan2(Y,X))');

CH2=AFFE_CHAR_MECA_F(MODELE=M0, DDL_IMP0=( _F(GROUP_MA='S2R3',DX=ux2,DY=uy2,)),);

#+-----+
#|          contact          |
#+-----+

CONT_ND = DEFI_CONTACT(MODELE = M0,
    FORMULATION = 'CONTINUE',
    REAC_GEO = 'AUTOMATIQUE',
    ITER_GEO_MAXI = 50,
    LISSAGE = 'OUI',
    ZONE = (
        _F(GROUP_MA_ESCL = 'S2R2',
            GROUP_MA_MAIT = 'S1R2',
            APPARIEMENT = 'MAIT_ESCL',
            NORMALE = 'MAIT',
            INTEGRATION = 'NOEUD',
            TOLE_PROJ_EXT = 0.0001,
            ALGO_CONT = 'STANDARD',
            COEF_CONT = E1,
            CONTACT_INIT = 'OUI',
            GLISSIERE = 'NON',),),);

#+-----+
#|          calcul          |
#+-----+

from math import pow

T=DEFI_LIST_REEL(DEBUT=0.1,INTERVALLE = _F(JUSQU_A = 2.1,NOMBRE=20,));
EXP=FORMULE(NOM_PARA='INST',VALE='pow(10,INST-1.1)',);
RP_EXP=CALC_FONC_INTERP(FONCTION = EXP,
    LIST_PARA = T,
    PROL_DROITE='EXCLU',
    PROL_GAUCHE='EXCLU',);

INSTS=DEFI_LIST_REEL(DEBUT = 0.0,
    INTERVALLE = _F(JUSQU_A = 2.1,
        NOMBRE = 21,));

RESU_ND=STAT_NON_LINE(MODELE=M0,
    CHAM_MATER=CHMAT,
    EXCIT = (_F(CHARGE = CH1, FONC_MULT = RP_EXP),
        _F(CHARGE = CH2, FONC_MULT = RP_EXP),),),
    CONTACT = CONT_ND,
    COMP_INCR = (_F(RELATION = 'ELAS',
        DEFORMATION = 'GROT_GDEP',),),),
    NEWTON = _F(REAC_INCR=1,REAC_ITER=1),
    INCREMENT = _F(LIST_INST=INSTS,),
    CONVERGENCE = (_F(ITER_GLOB_MAXI = 20,
        RESI_GLOB_MAXI = 1E-6),),),
    SOLVEUR = _F(METHODE = 'MUMPS',
        PCENT_PIVOT = 250,),),
    ARCHIVAGE = _F(CHAM_EXCLU = 'VARI_ELGA',),),
    INFO=1,);

#+-----+
#| impression des résultats |
#+-----+

# déplacements - fichier .pos de GMSH

IMPR_RESU(MODELE = M0,
    UNITE = 37,
    FORMAT = 'GMSH',
    RESU = _F(RESULTAT = RESU_G3,
        NOM_CHAM = 'DEPL',
        TYPE_CHAM = 'VECT_2D',
        NOM_CMP = ('DX','DY'),),);

```



```

# pression de contact - fichier .pos de GMSH
IMPR_RESU(MODELE = MO,
          UNITE = 38,
          FORMAT = 'GMSH',
          RESU = _F(RESULTAT = RESU_G3_,
                   NOM_CHAM = 'DEPL',
                   NOM_CMP = 'LAGS_C',
                   TYPE_CHAM = 'SCALAIRE',
                   ),);

# resultats
# NDL - noeud / valeurs locales
RESU_NDL = POST_RELEVE_T(ACTION=_F(
                                INTITULE = 'ND',
                                NOEUD = 'N10',
                                RESULTAT = RESU_NDL_,
                                NOM_CHAM = 'DEPL',
                                NOM_CMP = 'LAGS_C',
                                OPERATION = 'EXTRACTION',
                                ),);

# NDG et NDT - noeud / valeurs globales - triées
RESU_NDG = POST_RELEVE_T(ACTION=_F(
                                INTITULE = 'ND',
                                GROUP_NO = 'CONT_N_2',
                                RESULTAT = RESU_NDL_,
                                NOM_CHAM = 'DEPL',
                                NOM_CMP = 'LAGS_C',
                                OPERATION = 'MOYENNE',
                                ),);

RESU_NDT = CALC_TABLE(TABLE = RESU_NDG,
                      ACTION = _F(OPERATION = 'TRI',
                                   NOM_PARA = ('QUANTITE',),
                                   ),);

# resultats - tableau unité 40
IMPR_TABLE(TABLE = RESU_NDL,
           TITRE = 'Noeud - N10',
           UNITE = 40,
           NOM_PARA = ('NUME_ORDRE', 'LAGS_C',),
           );

IMPR_TABLE(TABLE = RESU_NDT,
           TITRE = 'Noeud - Valeurs globales',
           UNITE = 40,
           FILTRE = (_F(NOM_PARA = 'QUANTITE',
                       VALE_K = ('MAXIMUM', 'MINIMUM',),),),
           NOM_PARA = ('QUANTITE', 'NUME_ORDRE', 'LAGS_C',),
           );

#+-----+
#|          tests de validation          |
#+-----+

# ND

TEST_TABLE(TABLE = RESU_NDL,
           NOM_PARA = 'LAGS_C',
           PRECISION= 2.5E-2,
           FILTRE = _F(NOM_PARA = 'INST',
                       VALE = 0.1, ),
           VALE = -pres_cont*EXP(0.1),
           REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE = RESU_NDL,
           NOM_PARA = 'LAGS_C',
           PRECISION= 2.5E-2,
           FILTRE = _F(NOM_PARA = 'INST',
                       VALE = 0.2, ),
           VALE = -pres_cont*EXP(0.2),
           REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE = RESU_NDL,
           NOM_PARA = 'LAGS_C',
           PRECISION= 2.5E-2,
           FILTRE = _F(NOM_PARA = 'INST',
                       VALE = 0.3, ),
           VALE = -pres_cont*EXP(0.3),
           REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE = RESU_NDL,
           NOM_PARA = 'LAGS_C',
           PRECISION= 2.5E-2,
           FILTRE = _F(NOM_PARA = 'INST',
                       VALE = 0.4, ),
           VALE = -pres_cont*EXP(0.4),
           REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE = RESU_NDL,
           NOM_PARA = 'LAGS_C',
           PRECISION= 2.5E-2,
           FILTRE = _F(NOM_PARA = 'INST',
                       VALE = 0.5, ),
           VALE = -pres_cont*EXP(0.5),
           REFERENCE= 'ANALYTIQUE',);

```

```

VALE      = -pres_cont*EXP(0.5),
REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 0.6, ),
  VALE     = -pres_cont*EXP(0.6),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 0.7, ),
  VALE     = -pres_cont*EXP(0.7),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 0.8, ),
  VALE     = -pres_cont*EXP(0.8),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 0.9, ),
  VALE     = -pres_cont*EXP(0.9),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.0, ),
  VALE     = -pres_cont*EXP(1.0),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.1, ),
  VALE     = -pres_cont*EXP(1.1),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION=2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.2, ),
  VALE     = -pres_cont*EXP(1.2),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.3, ),
  VALE     = -pres_cont*EXP(1.3),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.4, ),
  VALE     = -pres_cont*EXP(1.4),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.5, ),
  VALE     = -pres_cont*EXP(1.5),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.6, ),
  VALE     = -pres_cont*EXP(1.6),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',
                VALE      = 1.7, ),
  VALE     = -pres_cont*EXP(1.7),
  REFERENCE= 'ANALYTIQUE',);

TEST_TABLE(TABLE
  = RESU_NDL,
  NOM_PARA = 'LAGS_C',
  PRECISION= 2.5E-2,
  FILTRE   = _F(NOM_PARA = 'INST',

```

```

        VALE      = -pres_cont*EXP(1.8),      VALE      = 1.8, ),
        REFERENCE= 'ANALYTIQUE', );
TEST_TABLE(TABLE
    = RESU_NDL,
    NOM_PARA = 'LAGS_C',
    PRECISION= 2.5E-2,
    FILTRE   = _F(NOM_PARA = 'INST',
        VALE      = -pres_cont*EXP(1.9),      VALE      = 1.9, ),
        REFERENCE= 'ANALYTIQUE', );
TEST_TABLE(TABLE
    = RESU_NDL,
    NOM_PARA = 'LAGS_C',
    PRECISION= 2.5E-2,
    FILTRE   = _F(NOM_PARA = 'INST',
        VALE      = -pres_cont*EXP(2.0),      VALE      = 2.0, ),
        REFERENCE= 'ANALYTIQUE', );
TEST_TABLE(TABLE
    = RESU_NDL,
    NOM_PARA = 'LAGS_C',
    PRECISION= 2.5E-2,
    FILTRE   = _F(NOM_PARA = 'INST',
        VALE      = -pres_cont*EXP(2.1),      VALE      = 2.1, ),
        REFERENCE= 'ANALYTIQUE', );

FIN();

```

# Annexe B : fichier GMSH du maillage du cas test SSPN122

```
// dimension

R1 = 1.0;
R2 = 0.6;
R3 = 0.2;

n_ext = 11;
n_ext_cont = 11;
n_int_cont = 11;
n_int = 11;

radial = 4;

// points

Point(1) = { 0.0, 0.0, 0.0, 0.0};

Point(2) = { R1, 0.0, 0.0, 0.0};
Point(3) = { 0.0, R1, 0.0, 0.0};
Point(4) = { -R1, 0.0, 0.0, 0.0};
Point(5) = { 0.0, -R1, 0.0, 0.0};

Point(6) = { R2, 0.0, 0.0, 0.0};
Point(7) = { 0, R2, 0.0, 0.0};
Point(8) = { -R2, 0.0, 0.0, 0.0};
Point(9) = { 0, -R2, 0.0, 0.0};

Point(10) = { R2, 0.0, 0.0, 0.0};
Point(11) = { 0, R2, 0.0, 0.0};
Point(12) = { -R2, 0.0, 0.0, 0.0};
Point(13) = { 0, -R2, 0.0, 0.0};

Point(14) = { R3, 0.0, 0.0, 0.0};
Point(15) = { 0.0, R3, 0.0, 0.0};
Point(16) = { -R3, 0.0, 0.0, 0.0};
Point(17) = { 0.0, -R3, 0.0, 0.0};

// lignes - couronne extérieure

Circle(1) = {2,1,3};
Line(2) = {3,7};
Circle(3) = {7,1,6};

Circle(5) = {3,1,4};
Line(6) = {4,8};
Circle(7) = {8,1,7};

Circle(9) = {4,1,5};
Line(10) = {5,9};
Circle(11) = {9,1,8};

Circle(13) = {5,1,2};
Line(14) = {2,6};
Circle(15) = {6,1,9};

Line Loop(1) = {1,2,3,-14};
Line Loop(2) = {5,6,7,-2};
Line Loop(3) = {9,10,11,-6};
Line Loop(4) = {13,14,15,-10};

// lignes - couronne intérieure

Circle(17) = {10,1,11};
Line(18) = {11,15};
Circle(19) = {15,1,14};

Circle(21) = {11,1,12};
Line(22) = {12,16};
Circle(23) = {16,1,15};

Circle(25) = {12,1,13};
Line(26) = {13,17};
Circle(27) = {17,1,16};

Circle(29) = {13,1,10};
Line(30) = {10,14};
Circle(31) = {14,1,17};

Line Loop(5) = {17,18,19,-30};
Line Loop(6) = {21,22,23,-18};
Line Loop(7) = {25,26,27,-22};
Line Loop(8) = {29,30,31,-26};

// surface

Plane Surface(1) = {1};
Plane Surface(2) = {2};
Plane Surface(3) = {3};
Plane Surface(4) = {4};

Plane Surface(5) = {5};
Plane Surface(6) = {6};
Plane Surface(7) = {7};
Plane Surface(8) = {8};
```

```

// transfinite
Transfinite Line{ 1, 5, 9,13} = n_ext;
Transfinite Line{ 3, 7,11,15} = n_ext_cont;
Transfinite Line{17,21,25,29} = n_int_cont;
Transfinite Line{19,23,27,31} = n_int;
Transfinite Line{ 2, 6,10,14,18,22,26,30} = radial;

Transfinite Surface{1} = { 2, 3, 7, 6};
Transfinite Surface{2} = { 3, 4, 8, 7};
Transfinite Surface{3} = { 4, 5, 9, 8};
Transfinite Surface{4} = { 5, 2, 6, 9};
Transfinite Surface{5} = {10,11,15,14};
Transfinite Surface{6} = {11,12,16,15};
Transfinite Surface{7} = {12,13,17,16};
Transfinite Surface{8} = {13,10,14,17};

Recombine Surface{1,2,3,4,5,6,7,8};

// physical
Mesh.ElementOrder = 1;

Physical Surface(100) = {1,2,3,4};
Physical Surface(101) = {5,6,7,8};

Physical Line(200) = { 1, 5, 9,13};
Physical Line(300) = { 3, 7,11,15};
Physical Line(400) = {17,21,25,29};
Physical Line(500) = {19,23,27,31};

```

## **Annexe C : fiche descriptive du cas test SSNP152**