Méthode LAC – Local Average Contact

Résumé :

Ce document décrit la manière de traiter le contact entre solides (2D ou 3D) en grandes transformations par une méthode de type « mortar ». La formulation est très précise pour les pressions de contact obtenues, en particulier lorsque les maillages ne sont pas coïncidents ou que les surfaces sont courbes.

Cette formulation est disponible dans la commande DEFI_CONTACT sous le nom ALGO_CONT='LAC'.

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

Table des matières

1 Introduction....4
2 Formulation théorique de la méthode....5
2.1 Le problème de contact....5
2.1.1 Formulation forte....5
2.1.2 Formulation faible....6
2.2 Problèmes discrets....6
2.2.1 Inéquation variationnelle discrète....6
2.2.2 Formulation mixte discrète....7
2.3 Résultats mathématiques....8
2.4 Formulation matricielle....8
3 Implémentation de la méthode....10
3.1 Préparation du maillage....10
3.1.1 Principes de la découpe....10
3.1.2 Le cas 2D....10
3.1.3 Le cas 3D....11
3.1.4 Quelques remarques sur le découpage....12
3.2 Préparation du contact....12
3.2.1 Illustration de la technique sur le cas TRIA3....13
3.2.2 Illustration des difficultés sur le cas QUAD4....14
3.2.3 Le cas TRIA6....16
3.2.4 Le cas QUAD8....17
3.2.5 Le cas SEG2....18
3.2.6 Les cas SEG3 ou QUAD9....19
3.3 Résolution du problème de contact....19
3.3.1 Algorithme de résolution des non-linéarités liées au contact....20
3.3.2 Appariement....21
3.3.2.1 Algorithme de recherche de complexité linéaire....21
3.3.2.2 Outil de projection....21
3.3.2.3 Outil « jeu » moyen....22
3.3.2.4 Gestion des maillages quadratiques....24
3.3.3 Calcul des matrices de contact....25
3.3.3.1 Calcul des matrices élémentaires de contact....25
3.3.3.2 Choix de la normale....25
3.3.4 Assemblage intelligent....26
3.3.5 Post-traitement....28
4 Références....29

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

Table des illustrations

Figure 1: Solides en équilibre....5
Figure 2: Les trois espaces de la zone de contact (maître, esclave et LAC)....13
Figure 3: Numérotation locale du maillage de calcul, cas TRIA3 (obtenue par DECOUPE_LAC)....14
Figure 4: Attribution des degrés de liberté sur chaque nouvelle maille de contact esclave TRIA3....14
Figure 5: Numérotation locale du maillage de calcul, cas QUAD4 (obtenue par DECOUPE_LAC)....15
Figure 6: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD4 (type E5)....15
Figure 7: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD4 (type E1 E2 E3 E4)....16
Figure 8: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave TRIA6....17
Figure 9: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD8 (type E5)....17
Figure 10: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD8 (type E1 E2 E3 E4)....18
Figure 11: Numérotation locale du maillage de calcul, cas SEG2/TRIA3 (obtenue par DECOUPE_LAC)....18
Figure 12: Attribution des degrés de liberté sur les mailles de contact esclave SEG2/TRIA3....18
Figure 13: Numérotation locale du maillage de calcul, cas SEG2/QUAD4 (obtenue par DECOUPE_LAC)....18
Figure 14: Attribution des degrés de liberté sur les mailles de contact esclave SEG2/QUAD4....19
Figure 15: Attribution des degrés de liberté sur les mailles de contact esclave SEG3....19
Figure 16: Attribution des degrés de liberté sur les mailles de contact esclave QUAD9....19
Figure 17: Configuration 2D étudiée, en bleu les mailles maîtres M1 et M2 , en rouge les mailles esclaves E1 et E2....22
Figure 18: Mailles de contact obtenues par les deux premières opérations de l'appariement....23
Figure 19: Projection et intersection dans l'espace de référence de la maille esclave....23
Figure 20: Jeu au point de point de Gauss,....24
Figure 21: Différents choix de champ de normales....26
Figure 22: Maillage esclave, en noir (,..., , ,...,) les degrés de liberté de déplacement, en rouge (1, ..., 4) les degrés de liberté de Lagrange « informatiques »....27

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

1 Introduction

La méthode des éléments finis est actuellement massivement utilisée en mécanique numérique pour simuler les problèmes de contact unilatéral entre deux objets déformables. Ce sont des problèmes possédant une condition aux limites non linéaire, à savoir respecter une condition de non interpénétration pour le champ de déplacement, une condition de signe pour la pression normale sur la zone de contact et une condition de complémentarité entre ces deux champs. Dans le cas général, de tels problèmes amènent à la gestion de maillages incompatibles (i.e. dont les nœuds des zones de contact des objets considérés sont non coïncidents).

Les méthodes les plus utilisées sont basées sur des algorithmes de type nœud-segment en $2D$ et de type nœud-facette en $3D$ (dans code_aster, il s'agit des formulations DISCRETE et CONTINUE en mode INTEGRATION='NOEUD'). Ces méthodes sont connues pour échouer aux patchs tests dans le cas général des maillages incompatibles et pour mener à des analyses mathématiques sous optimales. D'autre part, le milieu académique a été très productif dans ce domaine ces quinze dernières années. Les méthodes les plus abouties sont les méthodes ***mortar*** adaptées au contac.

Ces méthodes consistent à utiliser une simple projection L^2 sur un espace joint, i.e., espace d'éléments finis définis entre les deux objets. Les différents choix d'espaces joints engendrent toute une famille de méthodes de type mortar. Cependant ces méthodes possèdent d'importants désavantages lorsque l'on considère leur implémentation de manière générique dans un code industriel.

La méthode que l'on propose ici donne des résultats numériques et mathématiques d'aussi bonne qualité que ceux obtenus par les méthodes mortar tout en gardant les aspects (localité, définition analytique des espaces joints, etc...) rendant l'implémentation dans *code_aster* plus aisée en préservant l'architecture actuelle du code

Pour ce faire, on va définir une condition de contact simple qui consiste à satisfaire les conditions de contact en moyenne sur chaque macro-élément d'un macro-maillage défini de manière idoine (qu'il faudra préparer grâce à l'option 'DECOUPE_LAC' de l'opérateur CREA_MAILLAGE).

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

2 Formulation théorique de la méthode

2.1 Le problème de contact

2.1.1 Formulation forte

Considérons le problème d'équilibre de deux solides en équilibre sur la figure 1 , notés par l'indice $l=1,2$.

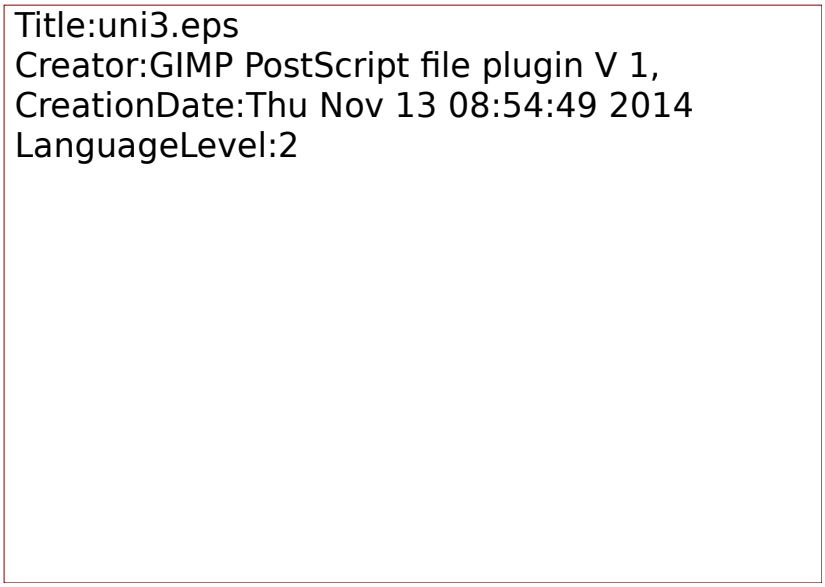


Figure 1: Solides en équilibre

On cherche le champ de déplacement $u=(u_1, u_2): \overline{\Omega^1} \times \overline{\Omega^2} \rightarrow \mathbb{R}^d$ ($d=2$ ou $d=3$) tel que (1)-(8) soient satisfaites pour $l=1,2$:

$$-\operatorname{div}\sigma(u_l)=f_l \text{ dans } \Omega^l \quad (1)$$

$$\sigma(u_l)=A_l\varepsilon(u_l) \text{ dans } \Omega^l \quad (2)$$

$$\sigma(u_l)n_l=F_l \text{ sur } \Gamma_N^l \quad (3)$$

$$u_l=0 \text{ sur } \Gamma_D^l \quad (4)$$

$$[u_N]\leqslant 0 \text{ sur } \Gamma_C \quad (5)$$

$$\sigma_N\leqslant 0 \text{ sur } \Gamma_C \quad (6)$$

$$\sigma_N[u_N]=0 \text{ sur } \Gamma_C \quad (7)$$

$$\sigma_\tau=0 \text{ sur } \Gamma_C \quad (8)$$

où $[v_N]=\sum_{l=1}^{2} v_l \cdot n_l$.

Les équations (1) à (4) correspondent au problème mécanique considéré (pour simplifier, on se place dans le cadre d'un problème d'élasticité linéaire) où A est un tenseur de quatrième ordre elliptique, symétrique et à composantes dans L^∞ provenant de la loi de comportement associée au solide Ω^l avec condition de Dirichlet sur le bord Γ_D^l et condition de Neumann sur le bord Γ_N^l . À ces équations, on ajoute les conditions de contact

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

sur Γ_C , à savoir une condition de non interpénétration pour le saut de déplacement $[u_N]$ (5), une condition de signe pour la pression normale σ_N la zone de contact (6) et une condition de complémentarité entre ces deux derniers champs (7). La dernière équation (8) vient du fait que l'on considère le contact sans frottement.

2.1.2 Formulation faible

A partir des équations (1) à (8), on peut introduire une formulation faible du problème de contact sous la forme de l'inéquation variationnelle suivante :

$$\text{Trouver } u \text{ tel que } \forall v \in K ,$$
$$\sum_{l=1}^{2} \int_{\Omega^l} A_l \varepsilon(u_l) : \varepsilon(v_l - u_l) d\Omega^l \geqslant \sum_{l=1}^{2} \int_{\Omega^l} f_l \cdot (v_l - u_l) d\Omega^l + \int_{\Gamma_N^l} F_l \cdot (v_l - u_l)_N d\Gamma^l \tag{9}$$

où $K \subset V$ est le cône convexe des déplacements admissibles (K contient la condition de non-interpénétration). On utilise donc les espaces suivants :

$$\begin{aligned} V_l &= \{ v_l \in (H^1(\Omega^l))^d : v = 0 \text{ sur } \Gamma_D^l \} \\ V &= V_1 \times V_2 \\ K &= \{ v \in V : [v_N] \leq 0 \text{ sur } \Gamma_C \} \end{aligned} \tag{10}$$

On notera par la suite :

$$a(u, v) = \sum_{l=1}^{2} \int_{\Omega^l} A_l \varepsilon(u_l) : \varepsilon(v_l) d\Omega^l \tag{11}$$

Et :

$$l(v) = \sum_{l=1}^{2} \int_{\Omega^l} f_l \cdot v_l d\Omega^l + \int_{\Gamma_N^l} F_l \cdot v_l d\Gamma^l \tag{12}$$

On introduit le problème mixte équivalent:

$$\begin{aligned} &\text{Trouver } u \in V \text{ et } \lambda \in M \text{ tels que:} \\ &a(u, v) - b(\lambda, v) = l(v) \quad \forall v \in V \\ &b(\mu - \lambda, u) \geq 0 \quad \forall \mu \in M \end{aligned} \tag{13}$$

Avec :

$$b(\mu, v) = \int_{\Gamma_C} \mu [v_N] d\Gamma \tag{14}$$

Et :

$$M = \left\{ \mu \in H^{-1/2}(\Gamma_C) : \int_{\Gamma_C} \mu \Psi d\Gamma \leq 0 \, \forall \Psi \in H^{-1/2}(\Gamma_C), \Psi \geq 0 \right\} \tag{15}$$

Remarque : *Les problèmes (9) et (13) sont bien posés. On a donc existence et unicité des solutions* (u, λ) *. De plus, les solutions* u *du p roblème (13) et* u *du p roblème (9) sont identiques.*

2.2 Problèmes discrets

2.2.1 Inéquation variationnelle discrète

On considère le problème de contact discret du point de vue inéquation variationnelle (discrétisation du problème) :

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

$$\begin{aligned} &\text{Trouver } u^h \in K^h \text{ tel que } \forall v^h \in K^h \\ &a(u^h, v^h - u^h) \geqslant l(v^h - u^h) \end{aligned} \tag{16}$$

où $K^h \subset V^h$ est le cône convexe des déplacements admissibles, a et l les formes bilinéaire et linéaire définies précédemment.

Pour définir une méthode, on doit choisir les espaces discrets V_l^h et l'ensemble K^h. Soit T_l^h une triangulation régulière de $\Omega^l \subset \mathbb{R}^d$, où $d=2,3$. On définit alors :

$$V_l^h = \{ v_l^h \in (C(\overline{\Omega^l}))^d : v_l^h|_T \in P_k(T), \forall T \in T_l^h, v_l^h = 0 \text{ sur } \Gamma_D^l \} \tag{17}$$

où $k=1,2$, $V^h = V_1^h \times V_2^h$.

On se propose d'utiliser la condition de contact suivante :

$$K^h = \left\{ v^h \in V^h : \int_{T^m} [v_N^h] d\Gamma \leq 0 \quad \forall T^m \in T^M \right\} \tag{18}$$

où T^M est un macro-maillage d'éléments T^m à définir selon le choix de V^h.

Cette condition de contact s'inspire des méthodes de type mortar tout en conservant un aspect local. En effet, la matrice de projection mortar n'est plus une matrice pleine car l'inverse de la matrice de base de l'espace $P^0(T^M)$ est diagonale. Cette propriété nous assure que les matrices de projection mortar (produit matriciel entre la matrice de base inverse de l'espace $P^0(T^M)$ et les matrices de couplage entre l'espace mortar et les espaces d'approximation) sont locales. Cependant, on sait que la simple condition de type :

$$\int_{T^h \cap \Gamma_C} [v_N^h] d\Gamma \leq 0 \tag{19}$$

n'est pas stable pour tous les éléments couramment utilisés dans les études de l'ingénierie (notamment en 3D, où seul l'hexaèdre à 27 nœuds est stable naturellement), et ne mène donc pas à des analyses mathématiques optimales. Cependant, en élargissant légèrement le support d'intégration, selon les cas, on arrive à démontrer des résultats de convergence optimale et de stabilité.

2.2.2 Formulation mixte discrète

On introduit le problème mixte équivalent :

$$\begin{aligned} &\text{Trouver } u^h \in V^h \text{ et } \lambda^h \in M^h \text{ tel que} \\ &a(u^h, v^h) - b(\lambda^h, v^h) = l(v^h) \quad \forall v^h \in V^h \\ &b(\mu^h - \lambda^h, u^h) \geq 0 \quad \forall \mu^h \in M^h \end{aligned} \tag{20}$$

avec :

$$M^h = \{ \mu^h \in X_1^h : \mu^h \leq 0 \text{ sur } \Gamma_C \} \tag{21}$$

Où :

$$X_1^h = \{ \mu^h \in L^2(\Gamma_C) : \mu^h|_{T^m} \in P^0(T^m), \forall T^m \in T^M \} \tag{22}$$

On obtient donc une méthode « mortar P^0 » qui sera automatiquement stabilisée par la définition du macro-maillage T^M.

Remarque : *La condition de contact dans K^h étant fixée, l'espace des multiplicateurs de Lagrange sera toujours un sous-espace de $P^0(T^m)$ quel que soit le type d'éléments finis utilisé dans l'espace d'approximation des déplacements V^h. On remarque donc que l'ordre de la condition de contact reste fixe*

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

pour tout espace d'approximation considéré (linéaire ou quadratique), seule la définition du macro-maillage varie pour assurer les bonnes propriétés mathématiques de la méthode.

2.3 Résultats mathématiques

Nous rappelons ici les principaux résultats mathématiques de la méthode LAC. L'hypothèse fondamentale de la méthode définit la manière de construire le *macro-maillage* est l'hypothèse du degré de liberté interne :

Hypothèse du degré de liberté interne :
Pour tout macro-élément $T^m \in T^M$, il existe un degré de liberté x_i de V_1^h tel que $supp(\varphi_i) \subset T^m$, où φ_i est la fonction de base associée à x_i.

Remarque : *on note qu'il existe une taille minimale pour les macro-mailles permettant à la fois de satisfaire l'hypothèse précédente et de garantir la localité de la méthode. On note aussi que le choix du maillage trace utilisé pour définir le macro-maillage est libre; ici, on a choisi le maillage 1.*

On peut maintenant rappeler les deux principaux résultats de convergence de la méthode LAC.

Théorème 1 : *Soient* u *et* u^h *les solutions des problèmes de contact continu et discret . On suppose que* $u \in (H^s(\Omega^1))^d \times (H^s(\Omega^2))^d$ *avec* $d=2,3$ *et* $3/2<s\leq 2$ *(*$3/2\leq s<5/2$ *si* $k=2$*). Si l'hypothèse du degré de liberté interne est satisfaite, alors il existe une constante* $C>0$ *indépendante de* h_1, h_2 *(où* h_1 *et* h_2 *sont les paramètres de discrétisation du solide* 1 *et du solide* 2 *respectivement) et* u*, telle que :*

$$\left\|u-u^h\right\|_{1,\Omega^1,\Omega^2} \leq C\left(h_1^{s-1}+h_2^{s-1}\right)\left\|u\right\|_{s,\Omega^1,\Omega^2} \tag{23}$$

Théorème 2 : *soient* (u,λ) *et* (u^h,λ^h) *les solutions des problèmes de contact continu et discret . On suppose que* $u \in (H^s(\Omega^1))^d \times (H^s(\Omega^2))^d$ *avec* $d=2,3$ *et* $3/2<s\leq 2$ *(*$3/2<s\leq 5/2$ *si* $k=2$*). Si l'hypothèse du DDL interne est satisfaite, alors il existe une constante* $C>0$ *indépendante de* h_1, h_2 *et* u*, telle que :*

$$\left\|\lambda-\lambda^h\right\|_{1/2,*,\Gamma_C} + \left\|u-u^h\right\|_{1,\Omega^1,\Omega^2} \leq C\left(h_1^{s-1}+h_2^{s-1}\right)\left\|u\right\|_{s,\Omega^1,\Omega^2} \tag{24}$$

où $\|.\|_{1/2,*,\Gamma_C}$ *est la norme duale de* $\|.\|_{1/2,\Gamma_C}$.

Remarque : *dans la suite de ce document on adoptera la convention suivante: on nommera surface esclave la surface sur laquelle on aura défini les macro-mailles, c'est à dire la surface portant les multiplicateurs de Lagrange de contact dans le cas de la formulation mixte . La surface en vis-à-vis avec cette dernière sera la surface maître.*

2.4 Formulation matricielle

On considère les matrices suivantes $C^E \in M_{e,k}(\mathbb{R})$, $C^M \in M_{m,k}(\mathbb{R})$ et $M^{LAC} \in M_k(\mathbb{R})$, où $M_{i,j}$ correspond à l'espace des matrices rectangulaires de taille $i \times j$ et $M_i = M_{i,i}$ correspond à l'espace des matrices carrés de taille i, définies par :

$$\begin{aligned} C_{ij}^E &= \int_{T_j} \left(\varphi_N^E\right)_i d\Gamma \\ C_{ij}^M &= \int_{T_j} \left(\varphi_N^M\right)_i d\Gamma \\ M_{ij}^{LAC} &= \delta_{ij} |T^i| \end{aligned} \tag{25}$$

où les T^i sont les supports des fonctions de base de $P^0(T^M)$, les φ_N^E sont les produits de la normale avec les fonctions de base associées à la surface esclave et les φ_N^M sont les produits de la normale avec les fonctions de base associées à la surface maître.

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

Soit U le champ de déplacement et Λ le multiplicateur de Lagrange, la formulation matricielle du problème est la suivante :

$$\begin{bmatrix} K_{NN} & K_{NM} & K_{NE} & 0 \\ K_{MN} & K_{MM} & K_{ME} & C^M \\ K_{EN} & K_{EM} & K_{EE} & C^E \end{bmatrix} \times \begin{bmatrix} U_N \\ U_M \\ U_E \\ \Lambda \end{bmatrix} = F \tag{26}$$

Avec les conditions suivantes :

$$\begin{aligned} M^{LAC^{-1}}\left({}^tC^E U_E + {}^tC^M U_M\right) \leqslant 0 \text{ dans } \mathbb{R}^k, \\ M^{LAC}\Lambda \leqslant 0 \text{ dans } \mathbb{R}^k, \\ M^{LAC^{-1}}\left({}^tC^E U_E + {}^tC^M U_M\right)\cdot M^{LAC}\Lambda = 0 \text{ dans } \mathbb{R}. \end{aligned} \tag{27}$$

En remarquant que M^{LAC} est diagonale définie positive, on a équivalence avec la formulation suivante :

$$\begin{bmatrix} K_{NN} & K_{NM} & K_{NE} & 0 \\ K_{MN} & K_{MM} & K_{ME} & C^M \\ K_{EN} & K_{EM} & K_{EE} & C^E \end{bmatrix} \times \begin{bmatrix} U_N \\ U_M \\ U_E \\ \Lambda \end{bmatrix} = F \tag{28}$$

Avec les conditions suivantes :

$$\begin{aligned} {}^tC^E U_E + {}^tC^M U_M \leqslant 0 \text{ dans } \mathbb{R}^k, \\ \Lambda \leqslant 0 \text{ dans } \mathbb{R}^k, \\ {}^tC^E U_E + {}^tC^M U_M \cdot M^{LAC}\Lambda = 0 \text{ dans } \mathbb{R}. \end{aligned} \tag{29}$$

On résout le problème (28)-(29) en utilisant une stratégie des contraintes actives (active set) couplée à un algorithme de Newton.

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

3 Implémentation de la méthode

Ce chapitre l'implémentation de la méthode « LAC » (Local Average Contact) dans *code_aster*. On discutera plus particulièrement des choix d'implémentation liés à la nouvelle condition de contact en évitant la création de nouveaux types de maille dans *code_aster* et en s'insérant dans le formalisme de gestion du contact déjà existant. Il convient donc pour cette première version d'implémentation de la méthode d'adopter la stratégie de contact maillé (on va définir des éléments tardifs de contact permettant de réaliser les calculs élémentaires des contributions de contact) et une approche de type Newton généralisé.
Les opérateurs impactés par la méthode sont :

- CREA_MAILLAGE;
- DEFI_CONTACT;
- STAT_NON_LINE.

La méthode proposée implique les évolutions suivantes :

- Une modification du maillage sur la zone de contact esclave (non invasive dans le volume) dans l'opérateur CREA_MAILLAGE;
- Une redéfinition des Lagrangiens de contact, implémentation de l'espace $P^0(T^M)$ dans l'opérateur DEFI_CONTACT;
- Un nouvel appariement de type « segment-segment » en 2D et de type « face-face » en 3D, nouvelle évaluation des statuts de contact et nouveau calcul élémentaire, impact dans l'opérateur STAT_NON_LINE et DYNA_NON_LINE.

3.1 Préparation du maillage

3.1.1 Principes de la découpe

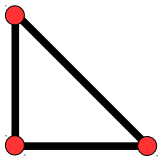
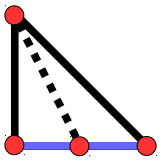
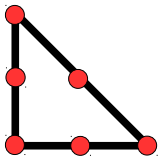
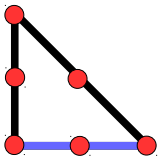
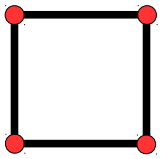
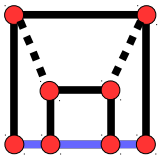
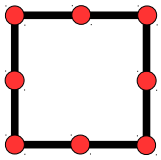
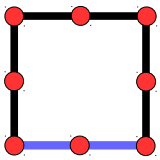
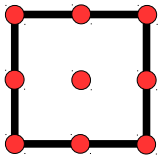
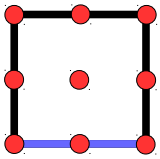
Le socle théorique nous impose de satisfaire l'hypothèse du degré de liberté interne (voir §2.3). Dans un souci de simplicité, de cohérence entre le 2D et le 3D, et pour ne pas « déraffiner » la condition de contact, on utilisera la méthode de raffinement local dans les cas 2D et 3D. Le mot-clef facteur DECOUPE_LAC de l'opérateur permet de préparer ce macro-maillage. Dans certaines situations, le mot-clef va découper les mailles de surfaces, dans d'autres, ce ne sera pas nécessaire. Quand on découpe un élément de surface on s'arrange donc pour ne pas *propager* cette découpe dans le volume. Cette stratégie permet de limiter la création de trop de mailles mais peut provoquer des mailles de mauvaise qualité.
D e même afin de pas modifier les choix initiaux de l'utilisateur, on proposera toujours un découpage ***conforme*** : un triangle sera découpé en triangles, un hexaèdre en hexaèdres, etc.

3.1.2 Le cas 2D

Dans le cas 2D, l'interface de contact esclave est « 1D ». Cette dernière est maillée soit en SEG2 soit en SEG3. En premier lieu, on remarque que le SEG3 associé à lui-même (macro-maille) satisfait l'hypothèse du DDL interne, il ne sera donc pas nécessaire de découper les mailles ayant des SEG3 en peau (TRIA6, QUAD8 et QUAD9). Dans le cas SEG2, il existe deux découpes selon le type de la maille de corps associée. Dans le cas TRIA3, on ajoute un nœud au centre de la maille SEG2 et on reconstruit les deux nouveaux éléments de peau et les deux nouveaux éléments de corps. Dans le cas QUAD4. Étant donné que l'on souhaite procéder à un découpage conforme, on ajoute deux nœuds au dans la maille SEG2, et deux nœuds dans la maille QUAD4. On peut ensuite reconstruire les trois nouveaux éléments de peau et les quatre nouveaux éléments de corps tout en renseignant les informations caractérisant la macro-maille. Les résultats de ces opérations sont présentés dans le tableau ci-dessous (la surface de contact est représentée en bleu).

Remarque : On portera une attention particulière à l'orientation des nouvelles mailles créées, à la conformité du maillage de calcul créé ainsi qu'à l'actualisation des groupes de maille. Sachant que l'on ne peut pas actualiser les groupes de nœuds. En effet, il n'est pas possible de déterminer si un nouveau nœud rentre dans un groupe de nœuds déjà défini ou non.

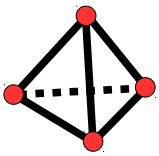
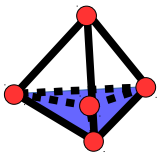
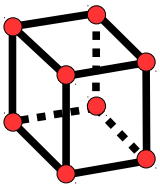
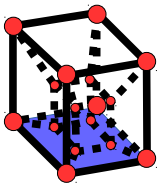
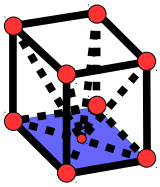
Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

	
TRIA3	TRIA3 après DECOUPE_LAC
	
TRIA6	TRIA6 après DECOUPE_LAC (pas de modifications)
	
QUAD4	QUAD4 après DECOUPE_LAC
	
QUAD8	QUAD8 après DECOUPE_LAC (pas de modifications)
	
QUAD9	QUAD9 après DECOUPE_LAC (pas de modifications)

3.1.3 Le cas 3D

Dans le cas 3D, l'interface de contact esclave est 2D. Cette dernière peut être maillée en TRIA3, TRIA6, QUAD4, QUAD8 et QUAD9. Comme dans le cas 2D, la maille QUAD9 associée à elle même satisfait l'hypothèse du DDL interne, on peut donc directement renseigner les structures de donnée contenant les informations relatives au macro-maillage sans raffiner localement. Dans les autres cas on utilisera deux découpes types, une pour les cas tétraédriques, et l'autre pour les cas hexaédriques. Ces deux découpes sont conformes dans le sens où l'on découpe un élément en générant le même type d'éléments. Les résultats de ces opérations est présenté dans le tableau ci-dessous.

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

	
TETRA4	TETRA4 après DECOUPE_LAC
	
	HEXA8 après DECOUPE_LAC (Version 1)
	
HEXA8	HEXA8 après DECOUPE_LAC (Version 2)
HEXA20	Deux versions (idem HEXA8)
HEXA27	HEXA27 après DECOUPE_LAC (pas de modifications)

3.1.4 Quelques remarques sur le découpage

La fonctionnalité DECOUPE_LAC de l'opérateur est capable de traiter:

- différents types de maille à l'intérieur d'une même zone de contact ;
- le multi-zone de contact.

Concernant la complexité des calculs sur le nouveau maillage on remarque que l'on rajoute un certain nombre de nœuds qui porteront des DDLs et potentiellement des Lagrangiens de contact. Soit M le nombre de mailles de la surface esclave de la zone de contact considérée, alors pour des maillages (de corps) en:

- TRIA3 on ajoute M nœuds et $2M$ mailles,
- QUAD4 on ajoute $4M$ nœuds et $5M$ mailles,
- TETRA4 on ajoute M nœuds et $4M$ mailles,
- TETRA10 on ajoute $5M$ nœuds et $4M$ mailles,
- HEXA8 (version 1) on ajoute $8M$ nœuds et $9M$ mailles,
- HEXA20 (version 1) on ajoute $28M$ nœuds et $9M$ mailles.

De plus l'aspect ratio et la configuration des mailles générées peuvent éventuellement parasiter le post-traitement de certaines données sur la surface esclave (éléments de peau et éléments de corps en contact).

3.2 Préparation du contact

La préparation des données du contact est assurée par l'opérateur DEFI_CONTACT. Cet opérateur doit prendre en compte la nouvelle définition des Lagrangiens de contact, i.e. l'espace $P^0(T^M)$. Il faut ajouter pour chaque macro-maille de la surface LAC un Lagrangien de contact (voir figure 2) sachant qu'il n'est pas possible pour ***code_aster*** d'affecter un degré de liberté à une autre entité physique qu'un nœud. On va présenter ici une

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

astuce permettant de définir l'espace $P^0(T^M)$ localement sur chaque maille de calcul sous réserve de disposer d'un assemblage intelligent du système matriciel.

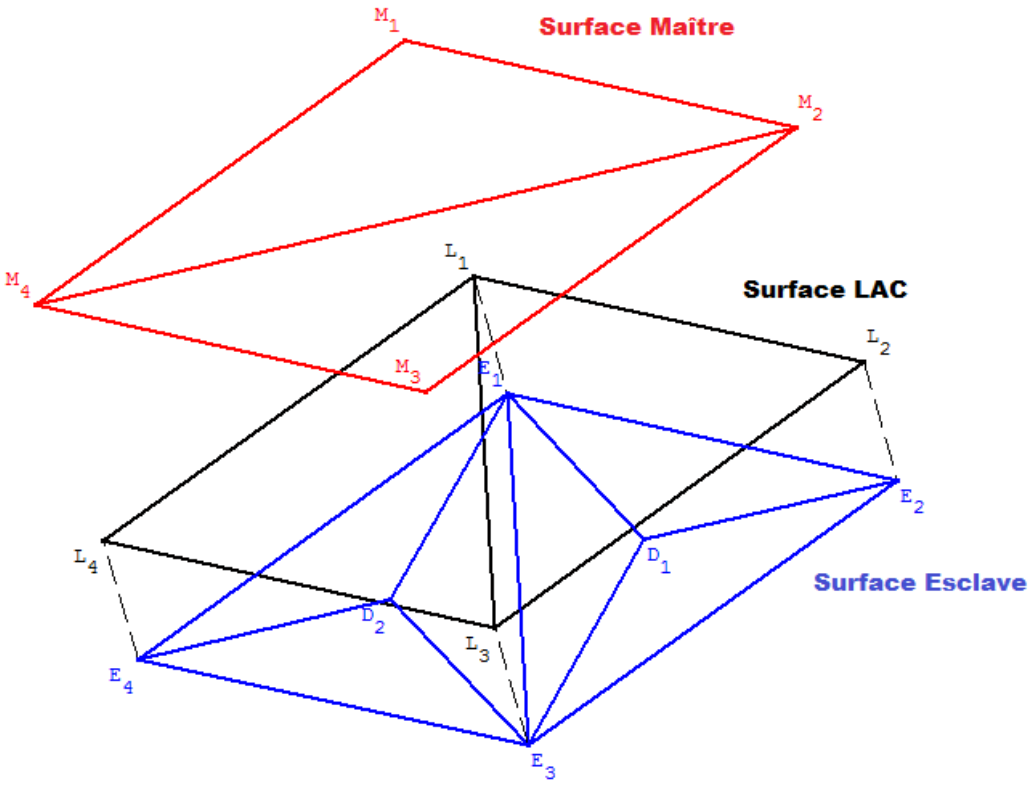


Figure 2: Les trois espaces de la zone de contact (maître, esclave et LAC)

3.2.1 Illustration de la technique sur le cas TRIA3

L'astuce réside dans le fait que l'on sait pour chaque maille de la zone de contact esclave du maillage donné par DECOUPE_LAC quel est le numéro local (dans la connectivité) du nœud interne de la macro-maille, voir figure 3.

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

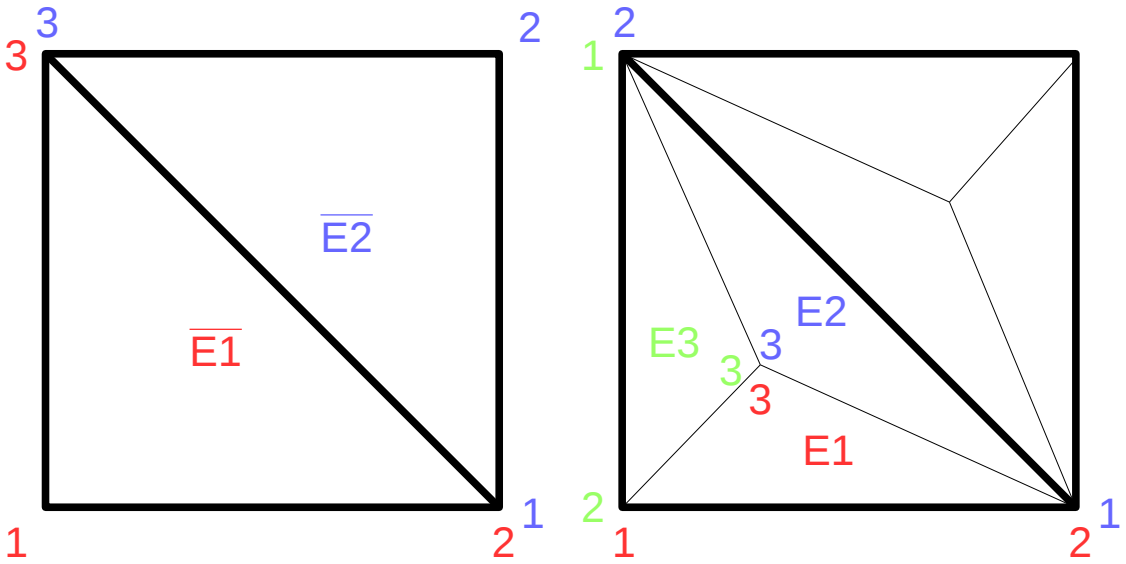


Figure 3: Numérotation locale du maillage de calcul, cas TRIA3 (obtenue par DECOUPE_LAC)

On sait donc sur quel nœud on doit ajouter le degré de liberté $LAGS_C$ (voir figure 4), le numéro local du nœud portant le degré de liberté $LAGS_C$ étant fixe on peut définir un seul nouveau type d'élément permettant de prendre en compte la localisation du $LAGS_C$.

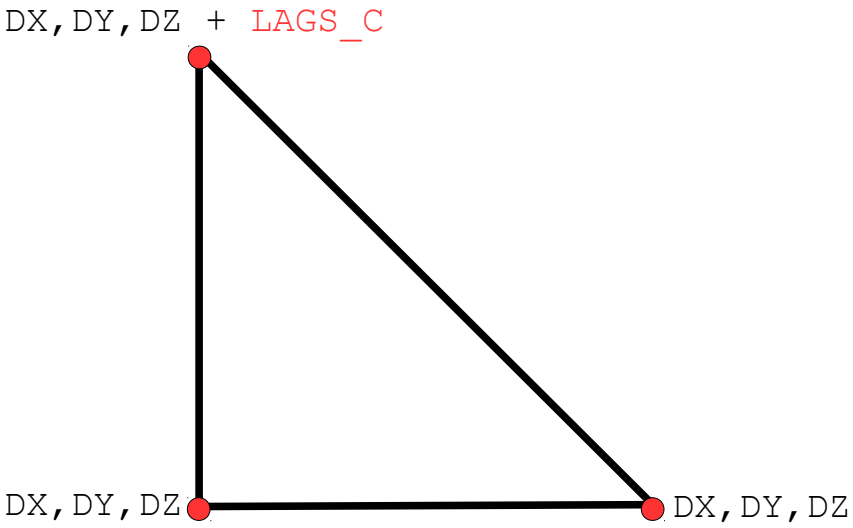


Figure 4: Attribution des degrés de liberté sur chaque nouvelle maille de contact esclave TRIA3

Chaque degré de liberté $LAGS_C$ est bien commun uniquement à trois éléments, les trois éléments composant la macro-maille. Étant donné que la fonction de base P^0 associée au multiplicateur de Lagrange de contact ne dépend pas des sommets géométriques de la macro-maille, on peut découper le calcul élémentaire sur la macro-maille en calcul élémentaire sur chaque maille la composant. Lors de l'assemblage, on somme les contributions de chaque maille sous-jacente à la macro-maille. Donc en attribuant à chaque degré de liberté $LAGS_C$ une fonction de base localement constante sur l'élément, on obtient bien que les degrés de liberté $LAGS_C$ correspondent à l'espace $P^0(T^M)$.

3.2.2 Illustration des difficultés sur le cas QUAD4

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

Le cas d'une interface de contact esclave maillée en **QUAD4** permet de mettre en évidence toutes les difficultés que cette astuce peut engendrer:

- Différentes localisations locales du nœud associé au degré de liberté $LAGS_C$;
- Démultiplication du degré de liberté $LAGS_C$, provenant du fait que l'on ne dispose pas d'un degré de liberté $LAGS_C$ commun à toute les mailles sous-jacentes à la macro-maille.

La deuxième difficulté vient du fait du découpage « conforme » en six HEXA8, voir figure 5.

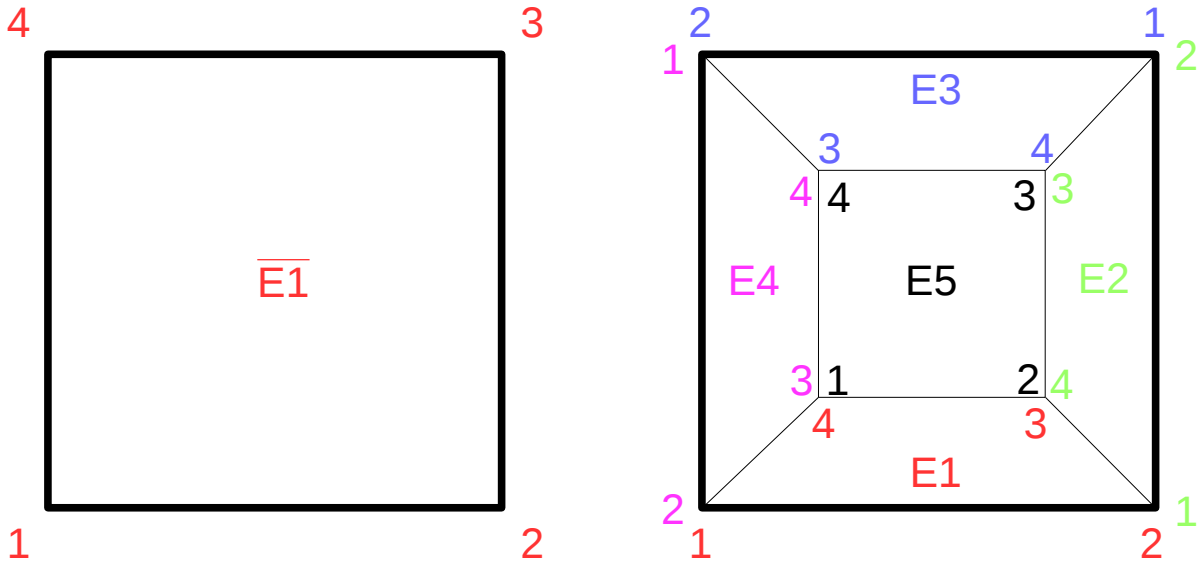


Figure 5: Numérotation locale du maillage de calcul, cas QUAD4 (obtenue par DECOUPE_LAC)

Ne disposant pas d'un degré de liberté $LAGS_C$ commun à toutes les mailles sous-jacentes à la macro-maille, on doit créer localement quatre $LAGS_C$ portés par chaque degré de liberté ajouté. On remarque alors la première difficulté. Il y a deux sortes de mailles sous-jacentes à la macro-maille, voir figures 6 et 7 :

- Quatre mailles portant deux degrés de liberté $LAGS_C$;
- Une maille portant les quatre degrés de liberté $LAGS_C$.

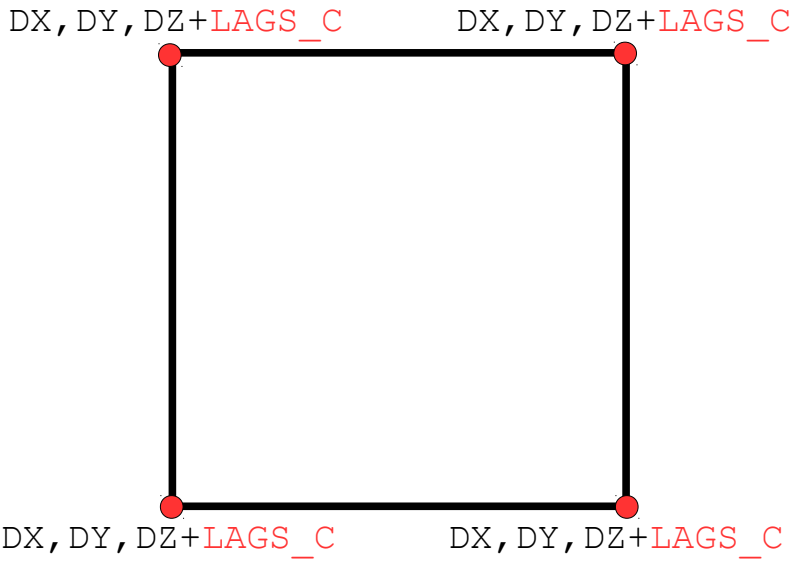


Figure 6: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD4 (type E5)

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

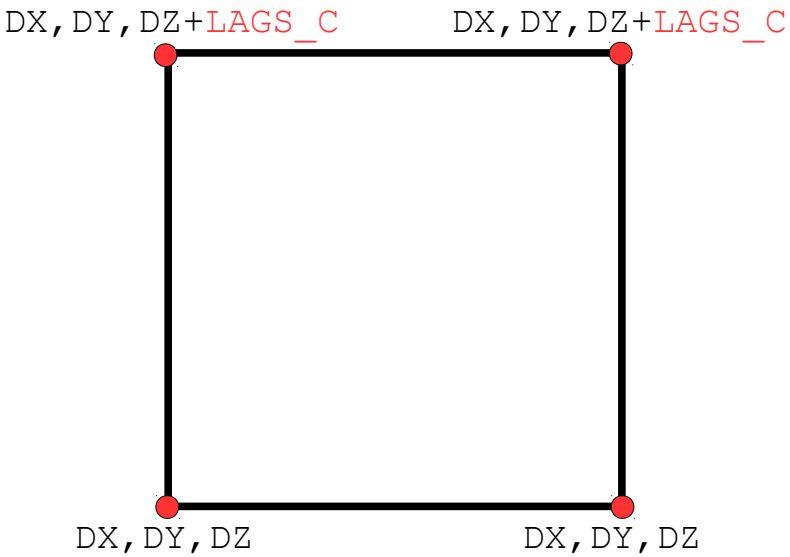


Figure 7: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD4 (type E1 E2 E3 E4)

Il faut, au cours de l'affectation des éléments de contact « tardifs » sur les mailles de contact esclaves dans DEFI_CONTACT, savoir si la maille est de type E5 ou E1, E2, E3, E4. On va donc boucler sur les macro-mailles puis sur les mailles sous-jacentes sachant qu'elles sont ordonnées de manière connue lors de la mise à jour du groupe de maille définissant la surface esclave de la zone de contact considérée (par exemple E1, E2, E3, E4 et E5).
Elle sera traitée dans STAT_NON_LINE par l'utilisation d'un assemblage intelligent. Lors de l'assemblage, on devra condenser les contributions dans un unique $LAGS_C$, afin de bien imposer la condition de contact en moyenne locale sur chaque macro-maille T^m.

3.2.3 Le cas TRIA6

Ce cas est similaire au cas TRIA3, voir figure 8.

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

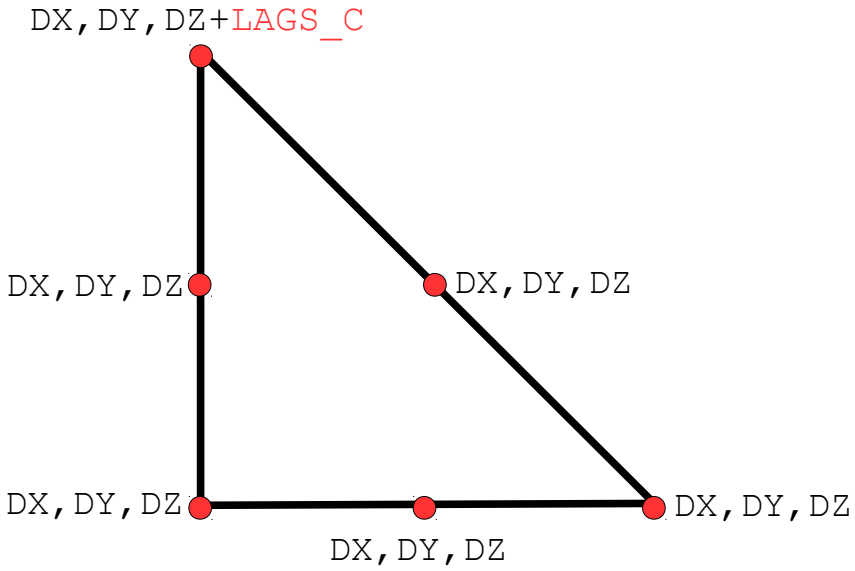


Figure 8: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave TRIA6

3.2.4 Le cas QUAD8

Ce cas est similaire au cas QUAD4, cependant on n'affecte pas de degré de liberté $LAGS_C$ aux nœuds milieux (voir figures 9 et 10).

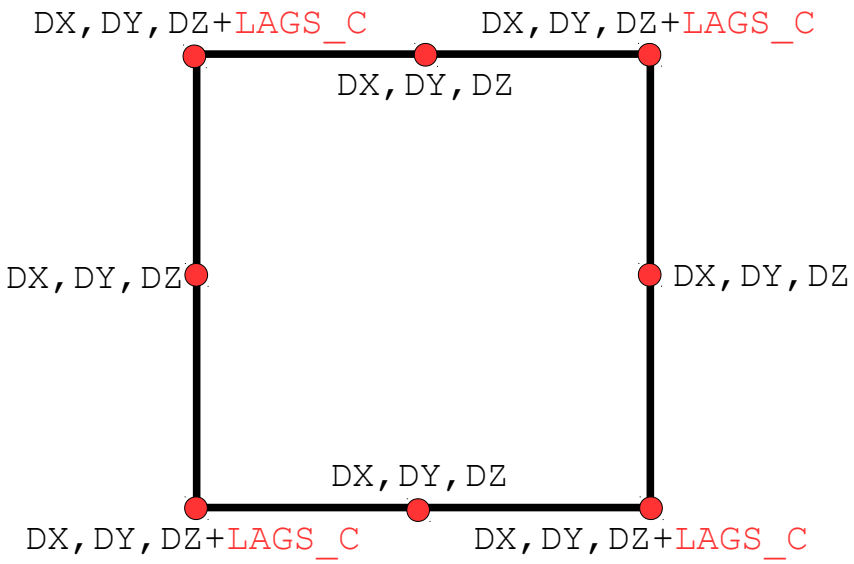


Figure 9: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD8 (type E5)

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

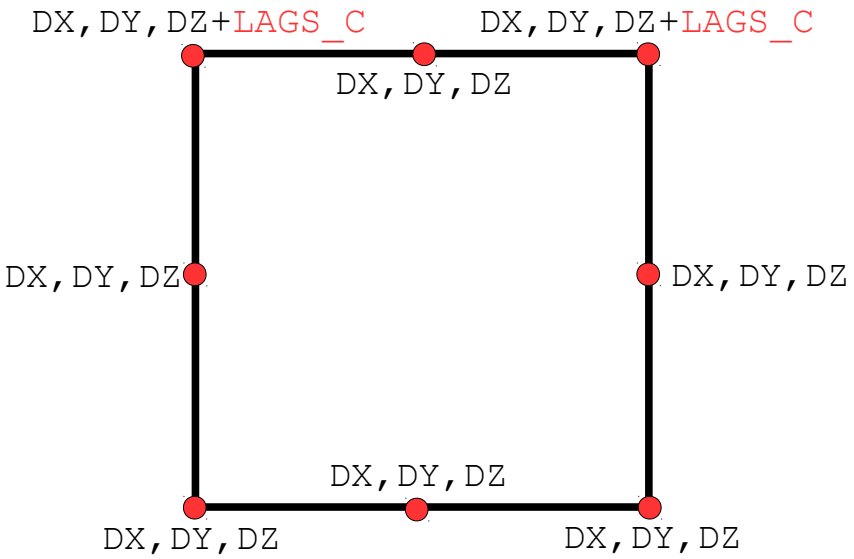


Figure 10: Attribution des degrés de liberté sur les nouvelles mailles de contact esclave QUAD8 (type E1 E2 E3 E4)

3.2.5 Le cas SEG2

On doit différencier ce cas en deux sous-cas, le premiers pour les SEG2 issus de TRIA3 et le second pour ceux issus de QUAD4. Ces cas sont similaires au cas QUAD4, en effet, comme l'on doit conserver l'orientation des mailles on ne peut pas ordonner librement les nœuds dans la connectivité du maillage issus de l'opération DECOUPE_LAC . Pour les SEG2/TRIA3, il existe deux types de mailles (voir figures 11 et 12),

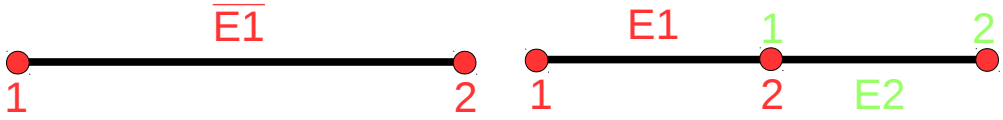


Figure 11: Numérotation locale du maillage de calcul, cas SEG2/TRIA3 (obtenue par DECOUPE_LAC)

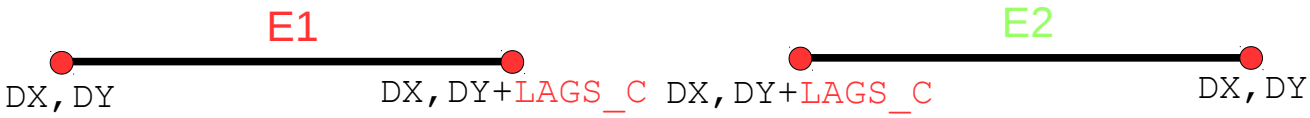


Figure 12: Attribution des degrés de liberté sur les mailles de contact esclave SEG2/TRIA3

Pour les SEG2/QUAD4, il existe trois types de mailles (voir figures 13 et 14).

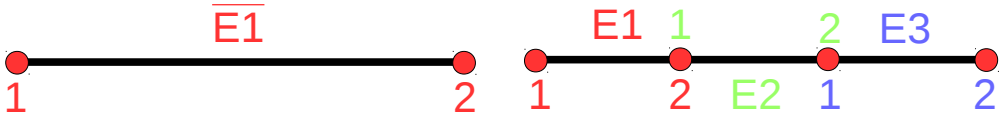


Figure 13: Numérotation locale du maillage de calcul, cas SEG2/QUAD4 (obtenue par DECOUPE_LAC)

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

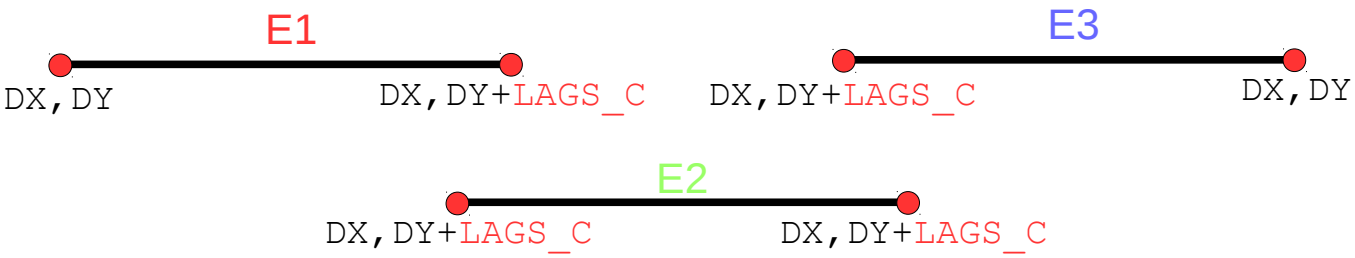


Figure 14: Attribution des degrés de liberté sur les mailles de contact esclave SEG2/QUAD4

3.2.6 Les cas SEG3 ou QUAD9

On a déjà remarqué que ces deux types de maille correspondent directement à des macro-mailles (les mailles et les macro-mailles sont confondues). Dans ce cas, on a donc un élément unique à affecter sur les mailles de contact SEG3 et QUAD9, voir figures 15 et 16.

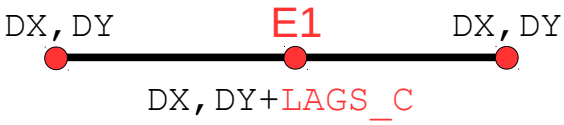


Figure 15: Attribution des degrés de liberté sur les mailles de contact esclave SEG3

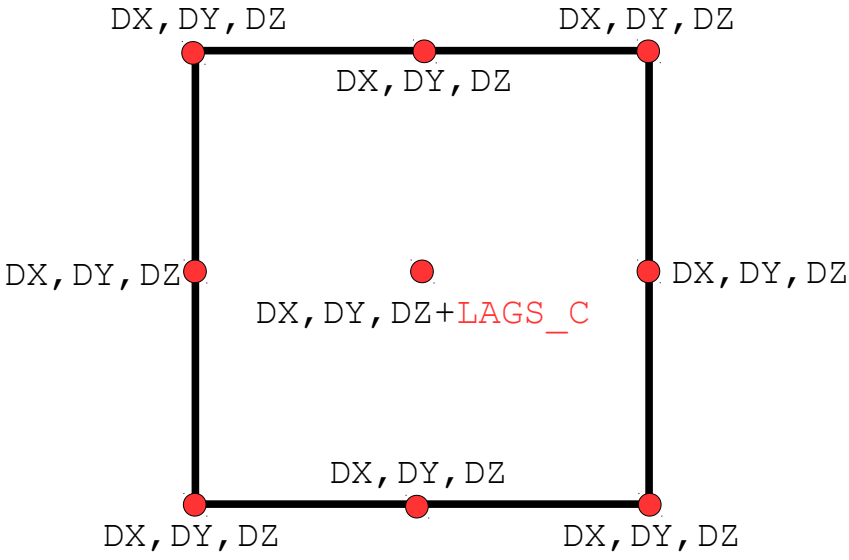


Figure 16: Attribution des degrés de liberté sur les mailles de contact esclave QUAD9

3.3 Résolution du problème de contact

Il y a trois impacts principaux :

- On doit avoir un appariement de type « segment-segment » en 2D et de type « face-face » en 3D ;
- On doit calculer les matrices de contact ;
- On doit développer un assemblage du système global intelligent (gestion de la deuxième difficulté liée à l'astuce dans DEFI_CONTACT, voir §3.2.2).

Les deux premiers points nécessitent un outil d'intersection de mailles efficace et robuste dans le cas de mailles droites (linéaire et quadratique) et dans le cas de mailles courbes (quadratiques).

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

3.3.1 Algorithme de résolution des non-linéarités liées au contact

On opte pour l'utilisation d'un algorithme de type Newton généralisé, gérant la non-linéarité géométrique, la stratégie de type contraintes actives pour la non-linéarité de contact et la non linéarité de comportement à l'intérieur de la même boucle de Newton. Ce cadre algorithmique est déjà présent dans *code_aster* (voir [R5.03.52]). Il sera donc suffisant de remplacer toutes les briques internes de l'algorithme pour implémenter la nouvelle méthode. On pose :

$$U=\begin{bmatrix} U_N \\ U_M \\ U_E \end{bmatrix} \; ; K=\begin{bmatrix} K_{NN} & K_{NM} & K_{NE} \\ K_{MN} & K_{MM} & K_{ME} \\ K_{EN} & K_{EM} & K_{EE} \end{bmatrix} \; ; C=\begin{bmatrix} 0 \\ C^M \\ C^E \end{bmatrix} \tag{30}$$

En prenant en compte le « jeu intégré » initial $\overline{G_0}$ (voir la définition au §), le problème est équivalent à :

$$\begin{cases} \begin{bmatrix} K & C \end{bmatrix} \times \begin{bmatrix} U \\ \Lambda \end{bmatrix} = F \\ {}^t CU \leqslant \overline{G_0} \text{ dans } \mathbb{R}^k \\ \Lambda \leqslant 0 \text{ dans } \mathbb{R}^k \\ {}^t CU \cdot \Lambda = 0 \text{ dans } \mathbb{R} \end{cases} \tag{31}$$

On introduit la fonction suivante :

$$A(U,\Lambda)=\Lambda+\max(0,\rho(CU-\overline{G_0})-\Lambda) \tag{32}$$

On peut alors montrer que est équivalent à :

$$\begin{bmatrix} K & C \end{bmatrix} \times \begin{bmatrix} U \\ \Lambda \end{bmatrix} = F \text{ et } A(U,\Lambda)=0 \tag{33}$$

La résolution de la non-linéarité de contact revient alors à la recherche des contraintes actives. Le contact est actif lorsque le statut du point est actif, ce que l'on notera $S\Lambda_i=1$. On obtient l'algorithme global suivant :

- Appariement initial : calcul de $\overline{G_0}$
- Boucle sur les Lagrange $i=1..dim(\Lambda)$
  - Si $\overline{G_{0i}} \leqslant 0$ alors le point n'est pas en contact, donc $\Lambda_i^0=0$
  - Sinon, le point est en contact, donc $\Lambda_i^0=1$
- Boucle de Newton $k=1$
  - Tant que { $S\Lambda^k \neq S\Lambda^{k+1}, \forall k$ et $\|r\|_2 \geqslant tole$ (résidu d'équilibre de Newton)
    - Calcul des matrices de rigidité et des seconds membres élémentaires
    - Calcul des matrices et des seconds membres de contact élémentaires
    - Boucle sur les Lagrange $i=1..dim(\Lambda)$
      - Si $S\Lambda_i^k=1$ alors $\begin{bmatrix} 0 & C_k^E & 0 \\ {}^t C_k^E & 0 & {}^t C_k^M \\ 0 & C_k^M & 0 \end{bmatrix}$

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

- Sinon $\begin{bmatrix} 0 & 0 & 0 \\ 0 & Id & 0 \\ 0 & 0 & 0 \end{bmatrix}$

◦ Assemblage du système global et résolution :

▪ $\begin{bmatrix} J_k^K & C_k & 0 \\ {}^tC_k & 0 & 0 \\ 0 & 0 & Id \end{bmatrix} \times \begin{bmatrix} \Delta U_k \\ \Delta \Lambda_k^a \\ \Delta \Lambda_k^{na} \end{bmatrix} = \begin{bmatrix} r_k \\ \overline{G_k} \\ -\Delta \Lambda_{k-1} \end{bmatrix}$.

◦ Mise à jour des inconnues :

▪ $U_{k+1}=U_k+\Delta U_k$ et $\Lambda_{k+1}=\Lambda_k+\Delta \Lambda_k$

◦ Appariement sur la géométrie actualisée calcul de $\overline{G_{k+1}}$

◦ Évaluation des nouveaux statuts :

▪ Boucle sur les Lagrange $i=1..dim(\Lambda)$

- Si $\Lambda_{k+1i}+\rho G_{k+1i} \leqslant 0$ alors le point est en contact, donc $S\Lambda_i^{k+1}=1$
- Sinon, le point n'est pas en contact, donc $S\Lambda_i^{k+1}=0$

◦ Calcul du résidu de Newton r_{k+1}

3.3.2 Appariement

Le but de cette phase est de gérer la non-linéarité géométrique du problème et celle issue de possibles grands déplacements. Pour chaque maille de contact esclave, on cherche les mailles maîtres susceptibles de communiquer avec la maille esclave. On doit aussi lors de cette opération calculer le « jeu » moyen entre les deux surfaces pour chaque macro-maille. Pour ce faire, on va utiliser quatre outils:

- Un outil de recherche de complexité linéaire utilisant l'information de voisinage au niveau des arêtes ;
- Un outil de projection de deux mailles sur le même plan de référence (espace paramétrique de référence de la maille esclave) ;
- Un outil d'intersection de mailles planes ;
- Un outil de calcul de distance moyenne entre un triangle et une surface (cas 3D) ou entre deux segments (cas 2D).

3.3.2.1 Algorithme de recherche de complexité linéaire

On adapte au cas du contact l'algorithme PANG de calcul de matrice de projection mortar pour la décomposition de domaine. Cet algorithme est de complexité linéaire dès lors que l'on a trouvé un couple de mailles de départ. Il utilise une recherche restreinte sur les mailles voisines maîtres des mailles maîtres intersectant une maille esclave. De plus il utilise les informations de voisinage au niveau des mailles esclaves pour mettre à jour la liste des couples de mailles de départ.
Cet algorithme permet un gain de temps important par rapport à un appariement de type force brute (i.e. double boucle d'appariement maître-esclave). Cependant il manque de robustesse. En effet, dans le cas de zone d'appariement ne possédant pas une seule composante connexe, il est mis en échec. On ne détecte alors une seule des zones d'appariement et l'algorithme de contact est dès lors mis en échec. On propose une version plus robuste mais plus coûteuse de l'algorithme initial en lui associant une recherche récursive de couple de mailles de départ par force brute.

3.3.2.2 Outil de projection

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

Cet outil est déjà disponible dans ***code_aster***. En effet, un des outils de l'appariement de la méthode CONTINUE (voir [R5.03.52]) effectue la projection orthogonale d'un point de l'espace réel dans l'espace paramétrique d'une maille donnée. On va donc pouvoir projeter n'importe quelle maille maître dans l'espace paramétrique de la maille esclave afin de réaliser le test d'intersection. On remarque deux points importants :

- On se limite au cas où la projection de la maille maître dans l'espace paramétrique esclave est ***convexe*** ;
- Étant donné que l'on effectue la projection dans l'espace paramétrique de la maille maître, on réalisera uniquement une intersection purement $1D$ ou $2D$.

De plus, une autre routine est disponible dans ***code_aster***: la projection sous une direction imposée. On note que cet algorithme de projection sera utile pour l'outil « jeu » moyen.

3.3.2.3 Outil « jeu » moyen

On introduit cet outil dans le cas 2D décrit dans la figure 17. Le maillage esclave est constitué de deux mailles ($E1$, $E2$) et donc une macro-maille associée à un degré de liberté $LAGS_C$; le maillage maître est lui aussi constitué de deux mailles ($M1$, $M2$).

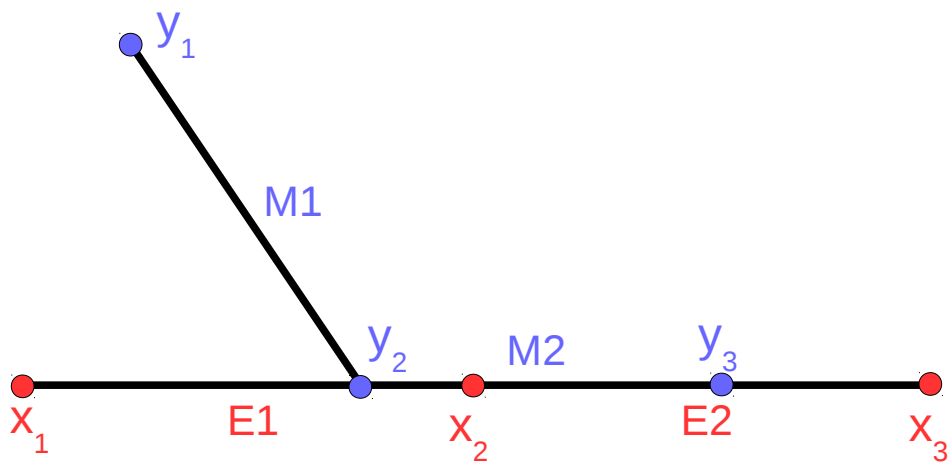


Figure 17: Configuration 2D étudiée, en bleu les mailles maîtres M1 et M2 , en rouge les mailles esclaves E1 et E2

En combinant les outils de projection et d'intersection, on obtient les mailles de contact décrites dans la figure 18. On va décrire l'outil de jeu moyen sur la première maille de contact.

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

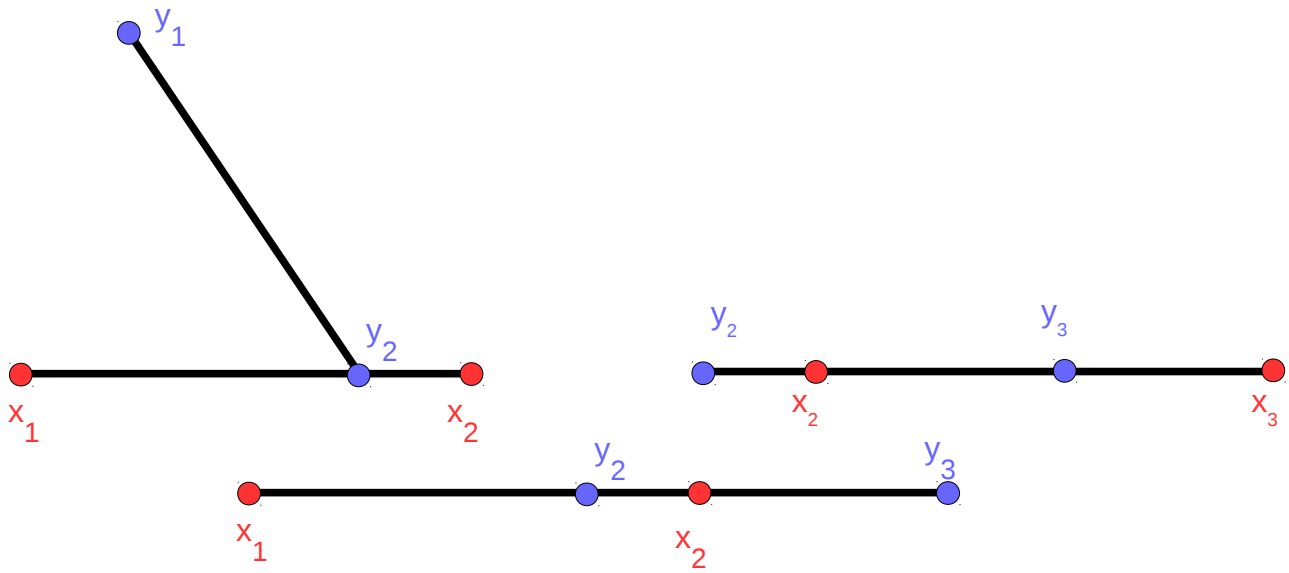


Figure 18: Mailles de contact obtenues par les deux premières opérations de l'appariement

Si l'on effectue les opérations projection sur la maille esclave et intersection, on obtient les projetés des points Y_1 et Y_2 dans l'espace de référence de la maille esclave $E1$, respectivement P_1 et P_2, et l'intersection (segment magenta). Le résultat de ces opérations est décrit dans la figure 19.

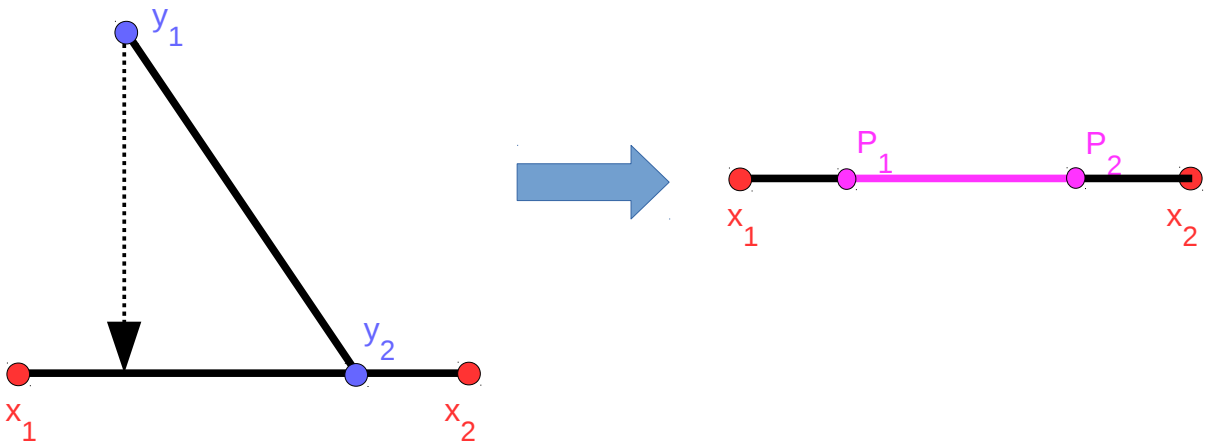


Figure 19: Projection et intersection dans l'espace de référence de la maille esclave $E1$

En considérant l'intersection comme un pseudo élément, on peut rapporter les points de Gauss issus d'une formule de quadrature dans l'espace de référence de la maille esclave. On considère ici par simplicité une formule de quadrature à un point de Gauss Pg ; son poids associé est la poids de l'intersection. On retourne alors dans l'espace réel pour pouvoir estimer la distance aux points de Gauss. Ces opérations sont décrites sur la figure 20.

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

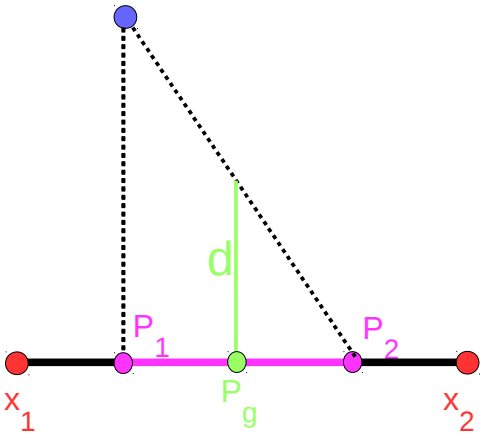


Figure 20: Jeu au point de point de Gauss, d

On ajoute la contribution suivante dans la composante du vecteur « jeu intégré », $\overline{G}$, correspondant au degré de liberté $LAGS_C$ de la maille de contact :

$$\overline{G}_{LAGS_C}=\overline{G}_{LAGS_C}+\sum_{i=1}^{N} d_i \omega_i \| (\partial_1 \sigma \times \partial_2 \sigma)(Pg_i) \|_2 \quad (34)$$

où d_i est le jeu signé au i-ème point de Gauss (Pg_i), ω_i est le poids associé au i-ème point de Gauss dans l'espace de référence de la maille maître de la maille de contact, σ est la transformation de passage élément de référence élément réel et N est le nombre de points de Gauss définis sur l'intersection.

Par itération sur les mailles de contact, on obtient le vecteur $\overline{G}$ dont la dimension correspond au nombre de degrés de liberté $LAGS_C$; dans notre exemple il n'y a une seule composante dans le vecteur « jeu » intégré.

Dans le cas tridimensionnel, un outil de triangulation de polygone convexe est nécessaire. En effet, il est plus aisé de calculer le $\overline{G}$ par morceaux sur un ensemble de triangles (définition plus simple des formules de quadrature que sur un polygone quelconque).

Pour finir l'opération il suffit de diviser chaque composante du vecteur « jeu » intégré par le poids d'intersection trouvé sur la macro-maille associée.

3.3.2.4 Gestion des maillages quadratiques

Dans le cas quadratique courbe, on ne peut pas approcher la géométrie de la maille maître dans l'espace paramétrique esclave. Pour simplifier l'algorithme, on va utiliser une simple linéarisation à la la place de la décomposition de la maille en éléments linéaires couramment utilisée dans la littérature des méthodes mortar pour le contact. La géométrie esclave est quant à elle approchée de manière exacte par son espace paramétrique. Contrairement à l'approche courante consistant à utiliser un plan auxiliaire d'intersection (définie à partir de la surface esclave), on a choisi de réaliser toute les opérations de projection et d'intersection dans l'espace paramétrique esclave. Ce choix permet à la fois l'utilisation d'un critère de tolérance d'appariement indépendant de la modélisation et de réaliser uniquement des intersections purement $2D$ ou $1D$. En utilisant cette approche et en contrôlant la qualité du maillage, on prévient l'apparition des cas pathologiques sévères qui aboutissent sur des erreurs fatales ou sur une non-convergence de l'algorithme de Newton généralisé. De plus, contrairement aux méthodes mortar classiques $P2/P2$ la méthode LAC n'est pas confrontée au problème de signe vis à vis des fonctions de base de l'espace des multiplicateurs de Lagrange. L'attention devra donc être principalement portée au niveau de la qualité du maillage, est plus particulièrement du maillage de la surface esclave. En effet, les choix algorithmiques mis en oeuvre dans cette première version de l'implémentation de la méthode ne prennent pas en compte le cas d'intersections non convexes. Si les mailles esclaves ne respectent pas cette contrainte, on obtient des intersections non pertinentes, on observe dès lors

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

des coefficients d'intersections aberrants (supérieur à un) ce qui aboutit à un échec de convergence de l'algorithme de résolution du problème.

3.3.3 Calcul des matrices de contact

Le calcul des matrices élémentaires de contact est dirigé par le champ de statuts $S\Lambda$. Si la contrainte associée au macro-élément T est active, alors on doit calculer les contributions suivantes :

$$\begin{aligned} C_i^{M_{elem}} &= \int_T \varphi_i^M \vec{n}_m \, d\Gamma = \sum_{M_c} \int_{M_m \cap M_e} \varphi_i^M \vec{n}_m \, d\Gamma \\ C_i^{E_{elem}} &= \int_T \varphi_i^E \vec{n}_e \, d\Gamma = \sum_{M_c} \int_{M_m \cap M_e} \varphi_i^E \vec{n}_e \, d\Gamma \end{aligned} \quad (35)$$

où φ^E et φ^M sont les fonctions de bases esclaves et maîtres, $\vec{n}_e$ et $\vec{n}_m$ les normales respectives, M_c l'ensemble des mailles de contact, et M_E l'ensemble des mailles escalves. Si la contrainte de contact associée au macro-élément T est inactive alors il n'y pas des conditions de contact à respecter sur ce macro-élément et l'on fixe à zéro le degré de liberté $LAGS_C$ associé. Étant donné que la macro-maille T n'existe pas en tant qu'élément, on calculera des matrices « sous-élémentaires » au niveau des mailles de contact M_c (association d'une maille de contact esclave et maître).

3.3.3.1 Calcul des matrices élémentaires de contact

On calcule en fait les matrices élémentaires de contact $C^{M_{elem}}$ correspondant aux intégrales sous la boucle sur les mailles de contact (voir équations 35). Pour chaque maille de contact, on doit calculer les matrices suivantes :

$$\begin{aligned} C_i^{M_{elem}} &= \int_{M_m \cap M_e} \varphi_i^M \vec{n}_m \, d\Gamma \\ C_i^{E_{elem}} &= \int_{M_m \cap M_e} \varphi_i^E \vec{n}_e \, d\Gamma \end{aligned} \quad (36)$$

où M_e est la maille esclave de la maille de contact et M_m est la maille maître de la maille de contact. Étant donné que la condition de contact choisie correspond à un jeu nul uniquement en moyenne, on doit procéder à une projection-intersection pour définir le support d'intégration. Pour ce faire, on réutilise les outils présentés dans la section précédente. En suivant le schéma de l'outil jeu intégré, on arrive à obtenir les coordonnées et poids d'une formule de quadrature dans l'espace paramétrique de la maille esclave de la maille de contact. En bouclant sur les degrés de liberté de déplacements maîtres, on peut alors calculer les coefficients de la matrice $C_i^{E_{elem}}$. Puis en utilisant la projection sur l'espace paramétrique de la maille maître de la maille de contact on obtient alors le support d'intégration dans l'espace référence de la maille maître, ce qui permet de calculer (en définissant un schéma d'intégration) les coefficients de la matrice $C_i^{M_{elem}}$.

3.3.3.2 Choix de la normale

La question du choix de la normale n'a pas encore été abordée. Ce choix n'étant pas unique, on se propose d'utiliser dans notre cas deux types de normale. Soit on évalue la normale à chaque point de Gauss maître et esclave, soit l'on utilise un champ de normales lissées (mot-clef `LISSAGE='OUI'`) calculé sur chaque configuration déformée (maître et esclave). Ces deux choix permettent d'éviter le problème de définition multiple de la normale au sommet des éléments lorsque l'on considère des géométries courbes. Dans le cas quadratique, ces deux choix sont relativement similaires, au contraire dans le cas des maillages linéaires, le champ de normale lissée permet de régulariser le champ des normales et de mieux approcher les géométries courbes, voir figure 21.

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

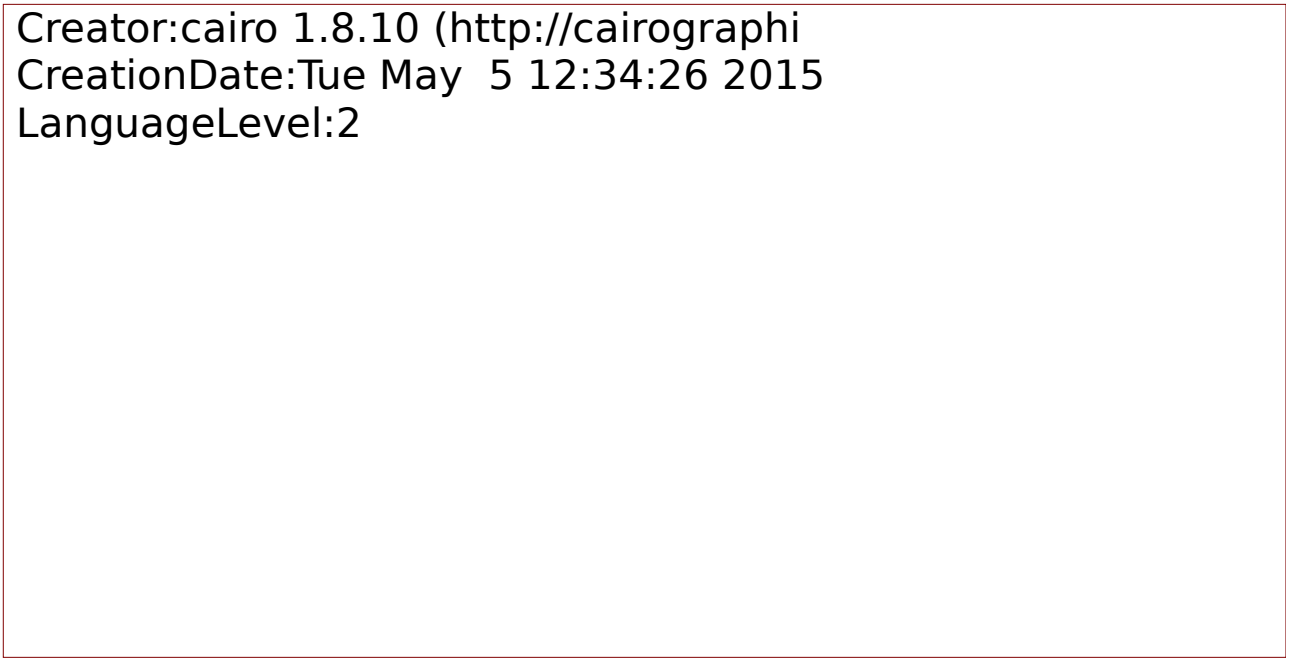


Figure 21: Différents choix de champ de normales

3.3.4 Assemblage intelligent

Nous présentons ici une piste pour gérer la deuxième difficulté liée à l'astuce utilisée dans DEFI_CONTACT. On se situe dans le cas HEXA8 ; on veut calculer la matrice C^E. On suppose que le maillage esclave est uniquement composé d'une macro-maille, il y a donc cinq mailles sur la surface esclave (pour simplifier nous ne parlerons que des contributions esclaves), huit degrés de liberté de déplacement, quatre Lagrangiens de contact « informatiques », mais un seul Lagrangien au sens mathématique. La figure 22 représente le maillage esclave considéré.

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

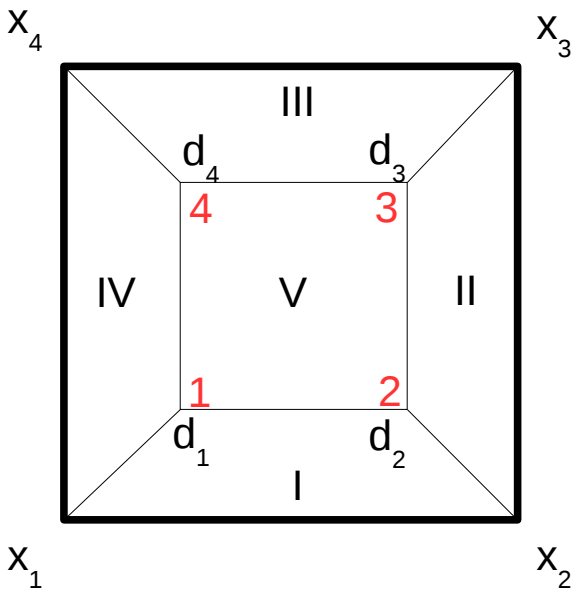


Figure 22: Maillage esclave, en noir (x_1 ,..., x_4 , d_1 ,..., d_4) les degrés de liberté de déplacement, en rouge (1, ..., 4) les degrés de liberté de Lagrange « informatiques ».

Les matrices élémentaires sont les suivantes :

- Élément I:

$$\begin{bmatrix} a_I & b_I & c_I & d_I \\ a_I & b_I & c_I & d_I \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ d_2 \\ d_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad (37)$$

- Élément II :

$$\begin{bmatrix} a_{II} & b_{II} & c_{II} & d_{II} \\ a_{II} & b_{II} & c_{II} & d_{II} \end{bmatrix} \times \begin{bmatrix} x_2 \\ x_3 \\ d_3 \\ d_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad (38)$$

- Élément III :

$$\begin{bmatrix} a_{III} & b_{III} & c_{III} & d_{III} \\ a_{III} & b_{III} & c_{III} & d_{II} \end{bmatrix} I \times \begin{bmatrix} x_3 \\ x_4 \\ d_4 \\ d_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \quad (39)$$

- Élément IV :

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

$$\begin{bmatrix} a_{IV} & b_{IV} & c_{IV} & d_{IV} \\ a_{IV} & b_{IV} & c_{IV} & d_{IV} \end{bmatrix} \times \begin{bmatrix} x_4 \\ x_1 \\ d_1 \\ d_4 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \quad (40)$$

- Élément V :

$$\begin{bmatrix} a_V & b_V & c_V & d_V \\ a_V & b_V & c_V & d_V \\ a_V & b_V & c_V & d_V \\ a_V & b_V & c_V & d_V \end{bmatrix} \times \begin{bmatrix} x_4 \\ x_1 \\ d_1 \\ d_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad (41)$$

Un assemblage classique nous donnerait la matrice globale C^E suivante :

$$\begin{bmatrix} a_I+b_{IV} & b_I & 0 & a_{IV} & a_V+c_I+c_{IV} & b_V+d_I & c_V & d_V+d_{IV} \\ a_I & a_{II}+b_I & b_{II} & 0 & a_V+c_I & b_V+d_I+d_{II} & c_V+c_{II} & d_V \\ 0 & a_{II} & b_{II}+a_{III} & b_{III} & a_V & b_V+d_{II} & c_V+c_{II}+d_{III} & d_V+c_{III} \\ b_{IV} & 0 & a_{III} & b_{III}+a_{IV} & a_V+c_{IV} & b_V & c_V+d_{III} & d_V+c_{III}+d_{IV} \end{bmatrix} \quad (42)$$

On remarque facilement que cette matrice ne correspond pas au couplage des déplacements avec les fonctions $P^0(T^M)$. On doit trouver une technique d'assemblage compatible avec l'assemblage utilisé dans ***code_aster*** (et qui ne doit pas être totalement incompatible avec le parallélisme) permettant d'obtenir la matrice de couplage suivante :

$${}^t\boldsymbol{DEPL} \times \begin{bmatrix} a_I+b_{IV} \\ b_I+a_{II} \\ b_{II}+a_{III} \\ b_{III}+a_{IV} \\ a_V+c_I+c_{IV} \\ b_V+d_I+d_I \\ c_V+d_{III}+c_{II} \\ d_V+c_{III}+d_{IV} \end{bmatrix} = LAG_C \quad (43)$$

On veut donc lors de l'assemblage élémentaire multiplier les contributions par $\frac{1}{2}$ dans le cas des éléments de type E1, E2, E3 et E4 (matrices élémentaires (37), (38), (39) et (40)) et par $\frac{1}{4}$ dans le cas des éléments de type E5 (matrice élémentaire (41)). Ensuite pour chaque macro-élément, on somme les lignes correspondantes de la matrice globale du système et on garde uniquement un Lagrangien dans le système correspondant à la condition de contact mathématique. Les trois Lagrangiens informatiques supplémentaires sont fixés égaux au multiplicateur de Lagrange physique.

3.3.5 Post-traitement

Toutes les variables de résolution du contact, statuts, multiplicateurs de Lagrange, « jeu », sont uniquement pertinentes au niveau de la macro-maille, le champ en sortie est nommé `CONT_ELEM`.

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

4 Références

[1] G. Bayada, M. Chambat, K. Lhalouani, and T. Sassi. Éléments finis avec joints pour des problèmes de contact avec frottement de Coulomb non local. (french) [on the mortar finite element method for contact problems with nonlocal Coulomb law]. C. R. Acad. Sci. Paris Sér. I Math., 325(12):1323–1328, 1997.

[2] F. Ben Belgacem and S. C. Brenner. Some non standard finite element estimates with applications to 3D Poisson and Signorini problems. Electronic Transactions on Numerical Analysis, 12:134–148, 2001.

[3] F. Ben Belgacem, P. Hild, and P. Laborde. Approximation of the unilateral contact problem by the mortar finite element method. C. R. Acad. Sci. Paris Sér. I Math., 324(1):123–127, 1997.

[4] F. Ben Belgacem, P. Hild, and P. Laborde. Extension of the mortar finite element method to a variational inequality modeling unilateral contact. Mathematical Models and Methods in Applied Sciences, 9(2):287–303, 1999.

[5] F. Ben Belgacem and Y. Renard. Hybrid finite element methods for the Signorini problem. Mathematics of Computation, 72(243):1117–1145, 2003.

[6] C. Bernardi and V. Girault. A local regularization operator for triangular and quadrilateral finite elements. SIAM Journal on Numerical Analysis, 35:1893–1916, 1998.

[7] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In J-L. Lions H. Brezis, editor, Collège de France Seminar, pages 13–51. Pitman, 1994.

[8] S. C. Brenner and R. Scott. The mathematical theory of finite element methods, volume 15. Springer, 2008.

[9] A. Chernov, M. Maischak, and E. P. Stephan. hp-mortar boundary element method for two-body contact problems with friction. Mathematical Methods in the Applied Sciences, 31(17):2029–2054, 2008.

[10] P. G. Ciarlet. The finite element method for elliptic problems. Elsevier, 1978.

[11] T. Cichosz and M. Bischoff. Consistent treatment of boundaries with mortar contact formulations using dual Lagrange multipliers. Computer Methods in Applied Mechanics and Engineering, 200(9):1317–1332, 2011.

[12] Z. Dost ́al, D. Horcak, and D. Stefanica. A scalable feti–dp algorithm with non-penetration mortar conditions on contact interface. Journal of computational and applied mathematics, 231(2):577–591, 2009.

[13] G. Drouet and P. Hild. An accurate Local Average Contact (LAC) method for nonmatching meshes in 2D and 3D. (hal-01143753), April 2015.

[14] G. Drouet and P. Hild. Optimal convergence for discrete variational inequalities modelling Signorini contact in 2D and 3D without additional assumptions on the unknown contact set. SIAM Journal on Numerical Analysis, to appear (preprint available in www.math.univ-toulouse.fr/∼phild/).

[15] A. Ern and J. L. Guermond. Theory and practice of finite elements. Springer-Verlag New York, Inc., 2004.

[16] P. Farah, A Popp, and W. A. Wall. Segment-based vs. element-based integration for mortar methods in computational contact mechanics. Comput. Mech., 55(1):209–228, 2015.

[17] G. Fichera. Problemi elastostatici con vincoli unilaterali: il problema di Signorini con ambigue condizioni al contorno. Atti Accad. Naz. Lincei Mem. Cl. Sci. Fis. Mat. Natur. Sez.I, 7(8):91–140, 1963/1964.

[18] S. Hartmann and E. Ramm. A mortar based contact formulation for non-linear dynamics using dual Lagrange multipliers. Finite Elements in Analysis and Design, 44(5):245–258, 2008.

[19] J. Haslinger, I. Hlavacek, and J. Necas. Numerical methods for unilateral problems in solid mechanics, volume 4 of Handbook of Numerical Analysis. North-Holland, Amsterdam, 1996.

[20] P. Hild. Problèmes de contact unilatéral et maillages éléments finis incompatibles. PhD thesis, Université Paul Sabatier, (www.math.univ-toulouse.fr/∼phild/), 1998.

[21] P. Hild. Numerical implementation of two nonconforming finite element methods for unilateral contact. Computer Methods in Applied Mechanics and Engineering, 184(1):99–123, 2000.

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

[22] N. Kikuchi and J. T. Oden. Contact problems in elasticity: a study of variational inequalities and finite element methods. SIAM, 1988.

[23] R. Krause. A nonsmooth multiscale method for solving frictional two-body contact problems in 2D and 3D with multigrid efficiency. SIAM Journal on Scientific Computing, 31(2):1399–1423, 2009.

[24] T. A. Laursen. Computational contact and impact mechanics: fundamentals of modeling interfacial phenomena in nonlinear finite element analysis. Springer, 2002.

[25] T. A. Laursen, M. A. Puso, and J. Sanders. Mortar contact formulations for deformable–deformable contact: past contributions and new extensions for enriched and embedded interface formulations. Computer methods in applied mechanics and engineering, 205:3–15, 2012.

[26] J. L. Lions and E. Magenes. Problemes aux limites non homogenes et applications, volume 1. Dunod, 1968.

[27] M. Moussaoui and K. Khodja. Régularité des solutions d'un problème mêlé Dirichlet–Signorini dans un domaine polygonal plan. Communications in partial differential equations, 17(5-6):805–826, 1992.

[28] A. Popp, B. I. Wohlmuth, M. W. Gee, and W. A. Wall. Dual quadratic mortar finite element methods for 3D finite deformation contact. SIAM Journal on Scientific Computing, 34(4):B421–B446, 2012.

[29] M. A. Puso and T. A. Laursen. A mortar segment-to-segment frictional contact method for large deformations. Computer Methods in Applied Mechanics and Engineering, 193(45-47):4891–4913, 2004.

[30] M. A. Puso, T. A. Laursen, and J. Solberg. A segment-to-segment mortar contact method for quadratic elements and large deformations. Computer Methods in Applied Mechanics and Engineering, 197(6):555–566, 2008.

[31] L.R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. Math. Comp., 54:483–493, 1990.

[32] A. Signorini. Questioni di elastostatica linearizzata e semilinearizzata. Rend. Math, 18:381–402, 1959.

[33] I. Temizer. A mixed formulation of mortarbased contact with friction. Computer Methods in Applied Mechanics and Engineering, 255:183–195, 2013.

[34] I. Temizer, P. Wriggers, and T.J.R. Hughes. Three-dimensional mortarbased frictional contact treatment in isogeometric analysis with NURBS. Computer Methods in Applied Mechanics and Engineering, 209/212:115–128, 2012.

[35] H. Triebel. Interpolation theory, function spaces, differential operators. North-Holland, 1978.

[36] M. Tur, F. J. Fuenmayor, and P. Wriggers. A mortar-based frictional contact formulation for large deformations using Lagrange multipliers. Computer Methods in Applied Mechanics and Engineering, 198(37):2860–2873, 2009.

[37] B. I. Wohlmuth. Variationally consistent discretization schemes and numerical algorithms for contact problems. Acta Numerica, 20:569–734, 2011.

[38] B. I. Wohlmuth, A. Popp, M. W. Gee, and W. A. Wall. An abstract framework for a priori estimates for contact problems in 3D with quadratic finite elements. Computational Mechanics, 49(6):735–747, 2012.

[39] P. Wriggers. Computational contact mechanics. Springer, 2006.

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