Calcul des contraintes aux nœuds par lissage local

Résumé :

On présente une méthode locale de calcul de contraintes aux nœuds à partir des contraintes aux points de GAUSS. Elle est utilisée dans les options SIGM_ELNO et SIEF_ELNO.

Cette méthode se résume à calculer les contraintes aux sommets d'un élément en multipliant les contraintes aux points de GAUSS par une matrice de lissage, constante pour chaque type d'élément.

Pour les éléments isoparamétriques de degré 2, les contraintes aux nœuds milieux sont obtenues par moyennage des valeurs des contraintes aux 2 sommets de l'arête.

Cette méthode de lissage présente deux avantages :

- les contraintes nodales obtenues ont un ordre de précision de plus que par le calcul direct aux nœuds,
- la méthode est peu coûteuse en temps CPU.

Cette méthode a été généralisée :

- aux calculs des déformations (option EPSI_ELNO) et des variables internes (option VARI_ELNO) aux nœuds en mécanique,
- au calcul des flux (option FLUX_ELNO) aux nœuds en thermique.

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

Table des Matières

1 Préliminaires....3
2 Méthode locale de minimisation par moindres carrés....4
3 Méthodes de lissage local (réf [bib2] et [bib3])....5
3.1 Lissage local continu....5
3.2 Lissage local discret....6
4 Application de la méthode au calcul des contraintes aux nœuds pour différents éléments....8
4.1 Matrices de passage carrées....8
4.1.1 Matrices de passage carrées pour les éléments 2D....8
4.1.1.1 Triangles....8
4.1.1.2 Quadrangles....8
4.1.2 Matrices de passage carrées pour les éléments 3D....9
4.1.2.1 Tétraèdres....9
4.1.2.2 Pentaèdres....9
4.1.2.3 Hexaèdres....9
4.2 Matrices de passage rectangulaires....10
5 Autres options de calcul utilisant la même méthode....11
6 Autres méthodes de lissage de contraintes....11
7 Bibliographie....11
8 Description des versions du document....11

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

1 Préliminaires

Cette méthode s'appuie sur la constatation [bib1] qu'il existe des points où le calcul des contraintes, à partir des déplacements dans une formulation primale en déplacements, est plus précis.

Dans le cas d'éléments finis isoparamétriques d'ordre 2 (SEG3 en 1D, QUAD8 et QUAD9 en 2D, HEXA20 en 3D), on montre que les points de GAUSS de la formule de quadrature à 2^n points (n : dimension de l'espace) sont tels que l'on peut espérer, sans que cela soit formellement démontré, pour le calcul de $\boldsymbol{\sigma}$ un même ordre de précision que pour le calcul du champ de déplacement $\mathbf{u}$.

L'idée de la méthode est de calculer pour chaque élément les contraintes $\hat{\boldsymbol{\sigma}}$ aux nœuds à partir de $\boldsymbol{\sigma}^{\mathbf{k}}$ aux points de GAUSS, ces dernières étant calculées sur chaque élément par la formule :

$$\boldsymbol{\sigma}^{\mathbf{k}} = \mathbf{D}\,\mathbf{B}^{\mathbf{k}}\,\mathbf{u} = \mathbf{D}\sum_{i=1}^{NNO}\mathbf{B}_{\mathbf{i}}^{\mathbf{k}}\,\mathbf{U}_{\mathbf{i}}$$

où :

- $\mathbf{D}$ est la matrice d'élasticité,
- $\mathbf{B}^{\mathbf{k}}$ est la matrice reliant les déformations aux déplacements au point de GAUSS k,
- $\mathbf{U}_{\mathbf{i}}$ sont les déplacements nodaux (NNO noeuds)

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

2 Méthode locale de minimisation par moindres carrés

D'une façon générale, on souhaite approximer, au sens des moindres carrés, la répartition spatiale des contraintes par $\boldsymbol{\sigma}(x)$ une fonction polynômiale :

$$\hat{\boldsymbol{\sigma}}(x)=\sum_{i=0,\dots,p}\mathbf{a_i}P^i(x)$$

Le problème revient à trouver les coefficients $\mathbf{a_i}$ qui minimisent la fonctionnelle :

$$\chi=\int\int(\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}})^2dx\,dy$$

Les valeurs de la fonction $\boldsymbol{\sigma}$ sont connues ici seulement aux points de Gauss : $\boldsymbol{\sigma}^{\mathbf{k}}=\boldsymbol{\sigma}(x_k)$

Le minimum sera atteint si et seulement si :

$$\frac{\partial\chi}{\partial\mathbf{a_i}}=0\qquad\forall i=0,\dots,p$$

Dans le cadre de la méthode des éléments finis en déplacement, on choisit la fonction de lissage suivante :

$$\hat{\boldsymbol{\sigma}}(x)=\sum_{i=1}^{n}N_i(x)\,\hat{\boldsymbol{\sigma}}_{\mathbf{i}}$$

où :

N_i est la fonction de forme associée au nœud i sur l'élément fini considéré,
$\hat{\boldsymbol{\sigma}}_{\mathbf{i}}$ est la valeur de la contrainte au nœud i cherchée,
n le nombre de nœuds retenus pour le lissage.

On doit donc résoudre le système :

$$\frac{\partial\chi}{\partial\hat{\boldsymbol{\sigma}}_{\mathbf{i}}}=0\qquad\forall i=1,\dots,n \qquad \text{éq 2-1}$$

On peut choisir entre deux méthodes de lissage local : lissage continu ou lissage discret.

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

3 Méthodes de lissage local (réf [bib2] et [bib3])

3.1 Lissage local continu

Ce type de lissage conduit à résoudre le système [éq 2-1] avec la fonctionnelle définie sur l'élément fini courant :

$$\chi = \int_e (\sigma - \hat{\sigma})^2 = \int_e (\sigma - \sum_{i=1}^{n} N_i \hat{\sigma}_i)^2$$

La minimisation conduit à $M^e \hat{\sigma} = F^e$

avec :

$$M_{ij}^e = \int_e N_i N_j \, dx \, dy = \sum_{k=1}^{npg} \bar{N}_i(\xi_k) \, N_j(\xi_k) (det \, \mathbf{J})_k \, \omega_k$$

$$F_i^e = \int_e N_i \sigma \, dx \, dy = \sum_{k=1}^{npg} \bar{N}_i(\xi_k) \sigma_k (det \, \mathbf{J})_k \omega_k$$

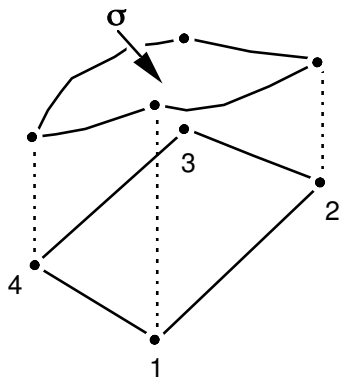
où ξ_k sont les points de GAUSS dans l'élément de référence

$(det \, \mathbf{J})_k$ le jacobien de la transformation géométrique entre l'élément de référence et l'élément courant au point ξ_k.

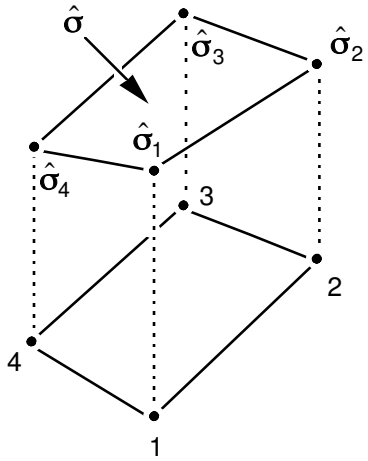
ω_k : le poids associé au point ξ_k

ξ_k : la contrainte au point ξ_k

$\bar{N}_i(\xi_k)$: la valeur de la fonction de forme dans l'élément de référence au point ξ_k



calcul direct des contraintes



contraintes lissées

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

Remarques :

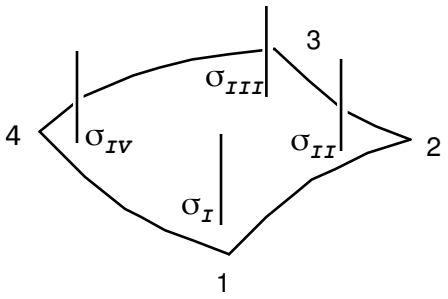
- *Si les espaces d'interpolation de $\boldsymbol{\sigma}$ et de $\hat{\boldsymbol{\sigma}}$ sont les mêmes, on a $\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}$. Dans la pratique, on retient pour espace de $\hat{\boldsymbol{\sigma}}$ un espace plus petit que celui où est défini $\boldsymbol{\sigma}$ par l'élément fini.*
- *On voit le lien entre l'approximation aux points de GAUSS de $\boldsymbol{\sigma}$ où $\boldsymbol{\sigma}$ converge donc mieux et ce procédé de lissage dont la justification est au contraire continue.*
- *La manière dont $\boldsymbol{\sigma}$ est calculé aux points de GAUSS n'intervient pas. La généralisation aux problèmes non linéaires est donc évidente, bien qu'elle ne puisse relever de la même justification.*

Cette méthode n'est pourtant pas retenue car elle nécessite une résolution de système linéaire pour chaque calcul de $\hat{\boldsymbol{\sigma}}$.

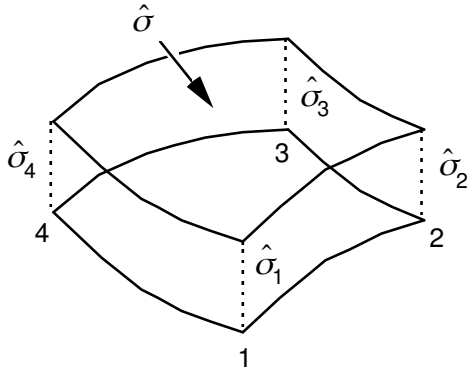
3.2 Lissage local discret

Dans ce cas, la fonctionnelle χ est remplacée par la sommation :

$$\tilde{\chi}=\sum_{k=1}^{npg}(\boldsymbol{\sigma}(\xi_k)-\hat{\boldsymbol{\sigma}}(\xi_k))^2=\sum_{k=1}^{npg}\left(\boldsymbol{\sigma}(\xi_k)-\sum_{i=1}^{n}\hat{\boldsymbol{\sigma}}_i N_i(\xi_k)\right)^2$$



contraintes aux points de GAUSS



contraintes lissées

Le système à résoudre s'écrit là encore : $\frac{\partial\tilde{\chi}}{\partial\hat{\boldsymbol{\sigma}}_{\mathbf{i}}}=0$ soit :

$$\sum_{k=1}^{npg}\sum_{j=1}^{n}N_i(\xi_k)N_j(\xi_k)\hat{\boldsymbol{\sigma}}_{\mathbf{j}}=\sum_{k=1}^{npg}N_i(\xi_k)\boldsymbol{\sigma}(\xi_k)\quad\forall i=1,\ldots,n$$

soit sous forme matricielle : $\mathbf{M}\{\hat{\boldsymbol{\sigma}}_{\mathbf{noeud}}\}=\mathbf{P}\{\boldsymbol{\sigma}_{\mathbf{GAUSS}}\}$

Les matrices $\mathbf{M}$ (carrée $n\times n$) et $\mathbf{P}$ (rectangulaire $n\times npg$) sont alors indépendantes de l'élément courant e.
Elles peuvent donc être calculées une fois pour toutes sur l'élément de référence.

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

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

Remarques :

- *Cette méthode est plus économique que la précédente et donne des résultats comparables [bib2],*
- *Là encore, la manière donc* σ^k *est calculée en chaque point de GAUSS est indifférente (du moment que le nombre de points de GAUSS utilisé pour le calcul de* σ *et celui de* $\hat{\sigma}$ *est le même). On pourra donc utiliser cette méthode en non linéaire*.

4 Application de la méthode au calcul des contraintes aux nœuds pour différents éléments

Le lissage local adopté dans *Code_Aster* est le lissage local discret [§2.2], qui permet d'éviter le calcul d'intégrales sur l'élément.

Sur tous les éléments de milieu continu 2D et 3D, on a choisi un espace de lissage s'appuyant sur les fonctions de forme relatives aux sommets de l'élément.

La méthode permet donc d'obtenir les contraintes aux sommets. Dans le cas des éléments d'ordre 2, on calcule les contraintes aux nœuds milieux en prenant la valeur moyenne arithmétique des deux sommets "encadrant" le nœud milieu considéré.

On donne ci-après les matrices de passage permettant de calculer les contraintes aux nœuds sommets à partir des contraintes aux points de GAUSS. Ces matrices peuvent être carrées ou rectangulaires. En effet, les matrices de passage $\mathbf{M}^{-1}\mathbf{P}$ sont calculées une fois pour toutes à l'initialisation de chaque type d'élément fini (dans AFFE_MODELE). Deux types de matrices existent :

- des matrices $\mathbf{M}^{-1}\mathbf{P}$ carrées, qui sont à utiliser lorsque le nombre de points de GAUSS utilisé pour le calcul des contraintes aux points de GAUSS σ^k est identique au nombre de nœuds sommets,
- des matrices $\mathbf{M}^{-1}\mathbf{P}$ rectangulaires, qui sont à utiliser lorsque le nombre de points de GAUSS de σ^k est différent (en général supérieur) au nombre de nœuds sommets.

4.1 Matrices de passage carrées

Ces matrices sont utilisées dans les éléments pour lesquels le nombre de points de GAUSS du calcul de SIEF_ELGA/SIGM_ELGA est égal au nombre de sommets. L'option calcule en premier les contraintes en un nombre de points de GAUSS égal au nombre de sommets. Puis les matrices $\mathbf{M}^{-1}\mathbf{P}$ (données après) sont utilisées pour calculer les contraintes aux nœuds. Il s'agit des éléments :

- en 2D : QUAD4, TRIA6, QUAD8 sous-intégré,
- en 3D : TETRA4, PENTA6, HEXA8, PYRAM5 et HEXA20 sous-intégré.

4.1.1 Matrices de passage carrées pour les éléments 2D

4.1.1.1 Triangles

$$\mathbf{M}^{-1}\mathbf{P} = \frac{1}{3}\begin{bmatrix} 5 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & 5 \end{bmatrix}$$

4.1.1.2 Quadrangles

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

$$\mathbf{M^{-1}P}=\begin{bmatrix} 1+\frac{\sqrt{3}}{2} & -\frac{1}{2} & 1-\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1+\frac{\sqrt{3}}{2} & -\frac{1}{2} & 1-\frac{\sqrt{3}}{2} \\ 1-\frac{\sqrt{3}}{2} & -\frac{1}{2} & 1+\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1-\frac{\sqrt{3}}{2} & -\frac{1}{2} & 1+\frac{\sqrt{3}}{2} \end{bmatrix}$$

4.1.2 Matrices de passage carrées pour les éléments 3D

4.1.2.1 Tétraèdres

$$\mathbf{M^{-1}P}=\frac{1}{a-b}\begin{bmatrix} a & a & a-1 & a \\ a & a-1 & a & a \\ a-1 & a & a & a \\ a & a & a & a-1 \end{bmatrix}$$

avec $a=\frac{5-\sqrt{5}}{20}$ $b=\frac{5+3\sqrt{5}}{20}$

4.1.2.2 Pentaèdres

$$\mathbf{M^{-1}P}=\begin{bmatrix} \alpha & -\alpha & \alpha & 1-\alpha & \alpha-1 & 1-\alpha \\ \alpha & \alpha & -\alpha & 1-\alpha & 1-\alpha & \alpha-1 \\ -\alpha & \alpha & \alpha & \alpha-1 & 1-\alpha & 1-\alpha \\ 1-\alpha & \alpha-1 & 1-\alpha & \alpha & -\alpha & \alpha \\ 1-\alpha & 1-\alpha & \alpha-1 & \alpha & \alpha & -\alpha \\ \alpha-1 & 1-\alpha & 1-\alpha & -\alpha & \alpha & \alpha \end{bmatrix}$$

$$\alpha=\frac{\sqrt{3}+1}{2}$$

4.1.2.3 Hexaèdres

$$\mathbf{M^{-1}P}=\begin{bmatrix} a & b & b & c & b & c & c & d \\ b & c & c & d & a & b & b & c \\ c & d & b & c & b & c & a & b \\ b & c & a & b & c & d & b & c \\ b & a & c & b & c & b & d & c \\ c & b & d & c & b & a & c & b \\ d & c & c & b & c & b & b & a \\ c & b & b & a & d & c & c & b \end{bmatrix}$$

$a=\frac{5+3\sqrt{3}}{4}$
$b=\frac{1+\sqrt{3}}{4}$
$c=\frac{\sqrt{3}-1}{4}$
$d=\frac{5-3\sqrt{3}}{4}$

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

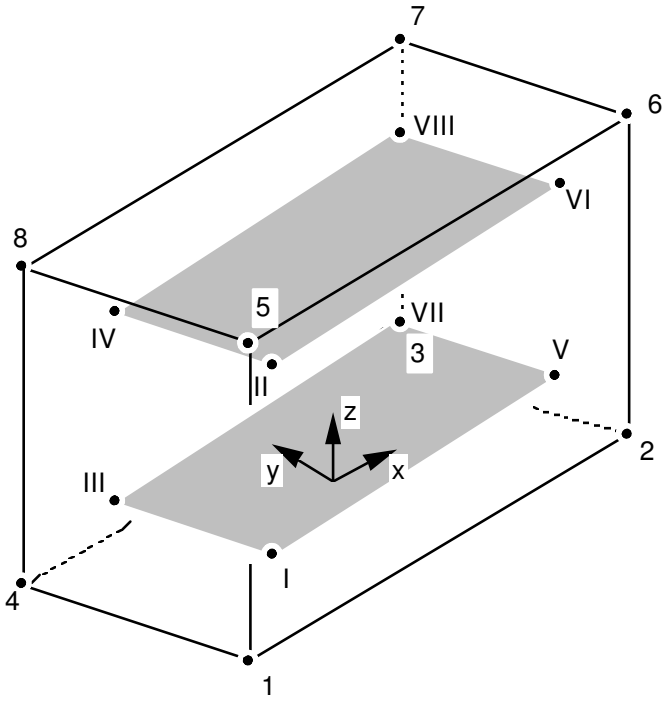


Figure 4.1.2.3-a : Numérotation des points de GAUSS sur l'hexaèdre à 8 nœuds

4.2 Matrices de passage $\mathbf{M^{-1}P}$ rectangulaires

En non linéaire pour certains types d'éléments (TRIA3, QUAD8 et QUAD9 en 2D, TETRA10, PENTA15 et HEXA20 en 3D), les contraintes et les variables internes aux points de GAUSS sont calculées sur une famille de points de GAUSS plus riche (9 points pour les quadrangles, 15 points pour les tétraèdres, 21 points pour les pentaèdres, 27 points pour les hexaèdres).

Le lissage local discret est alors effectué à partir de ces champs et le transport aux nœuds fait intervenir des matrices différentes des précédentes. Elles ne sont plus carrées, car de dimension (nombre de sommets, nombre de points de GAUSS). Les matrices de passage $\mathbf{M^{-1}P}$ ne sont pas calculées explicitement, en particulier $\mathbf{M}$ est inversée par *Code_Aster*.

Dans le cas particulier du triangle à 3 nœuds, les champs sont supposés constants par élément (un seul point de GAUSS) et :

$$\mathbf{M^{-1}P}=\begin{bmatrix}1\\1\\1\end{bmatrix}$$

Par exemple, le calcul effectué par l'option SIGM_ELNO est alors le suivant :

Si les contraintes ont été calculées sur une famille possédant un nombre de points de GAUSS supérieur au nombre de sommets (pour les éléments signalés ci-dessus). $\mathbf{M^{-1}P}$ est alors rectangulaire, et $\hat{\boldsymbol{\sigma}}_\mathbf{i}=\sum_{i=1}^{\text{nb sommets}}\sum_{k=1}^{\text{nb pts Gauss}}(\mathbf{M^{-1}P})_{ik}\boldsymbol{\sigma}^\mathbf{k}$

Sinon, si le nombre de points de GAUSS est égal au nombre de sommets, $\mathbf{M^{-1}P}$ est alors carrée. On calcule $\hat{\boldsymbol{\sigma}}_\mathbf{i}=(\mathbf{M^{-1}P})_{ik}\boldsymbol{\sigma}^\mathbf{k}$ [§4.1].

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

5 Autres options de calcul utilisant la même méthode

La méthode décrite précédemment est utilisée dans *Code_Aster* pour calculer les déformations, les variables internes et les flux aux nœuds.

Les champs produits sont des cham_elem aux nœuds.

6 Autres méthodes de lissage de contraintes

Il existe deux autres méthodes de lissage, portant seulement sur les contraintes, utilisées par les estimateurs de Zhu-Zienkiewicz version 1 et 2 [R4.10.01 §3].

Les champs de contraintes aux nœuds produits sont alors des cham_no.

Les options de calcul correspondantes sont accessibles par la commande CALC_ERREUR [U4.81.06].

7 Bibliographie

1) BARLOW J. - Optimal stress locations in finite element models - International Journal for Numerical Methods in Engineering Vol.10 p 243 - 251 (1976).

2) HINTON E., CAMPBELL JJ. - Local and global smoothing of discontinuous finite element functions using a least squares method - International Journal for Numerical Methods in Engineering Vol.8 p 461 - 480 (1974).

3) HINTON E., SCOTT F.C., RICKETTS R.E. - Local least squares stress smoothing for parabolic isoparametric elements - Int. J. for Num. Meth. in Eng. Vol 9 p 235 - 256 (1975)

8 Description des versions du document

Version Aster	Auteur(s) Organisme(s)	Description des modifications
10.2	*X. DESROCHES (EDF/IMA/T62)*	Petites corrections

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