# Accélération de Convergence

## 1 Accélération de convergence

### 1.1 Construction d'un sous-espace sécant

Le principe consiste à réduire l'opérateur tangent (inconnu et on ne souhaite pas le calculer) à un sous espace (Line Search Multidimension)

A chaque itération est associé un couple $\left\{\overrightarrow{U_i}, \overrightarrow{R_i}\right\}$ vérifiant l'équation suivante :

$$\overline{F_{ext}\left(\overrightarrow{U_i}\right)} - \overline{F_{\text{int}}\left(\overrightarrow{U_i}\right)} = \overrightarrow{R_i}$$

Il faut définir un produit scalaire auquel on associe la norme Euclidienne (le choix de la norme n'est pas unique) :

- $\vec{u} \cdot \vec{v} = \sum_{i=1}^{q} \left(u_i v_i\right)$
- $\left\|\vec{v}\right\| = \vec{v} \cdot \vec{v}$

**Remarques** :

1- La norme Euclidienne est choisie parce que la minimisation (dérivation) fait apparaitre un système linéaire (facile à résoudre).

2- On minimise $\vec{R} \cdot \vec{R}$ mais on pourrait envisager de minimiser le travail associé à l'incrément de déplacement : $\overrightarrow{\Delta U} \cdot \vec{R}$

En sélectionnant $\overrightarrow{R_n}$ l'un des résidus (au choix), après $m+1$ itérations, on peut construire $m$ vecteurs $\left(\overrightarrow{R_{i \neq n}} - \overrightarrow{R_n}\right)$ constituant un espace vectoriel (Ils ne sont pas nécessairement libres). Une base orthonormée $\left\{\overrightarrow{b_i}\right\}$constituée de $p$ vecteurs ( $p \leq m$ ) est fabriquée à l'aide du procédé d'orthogonalisation de Gram-Schmidt :

$$\begin{cases} \overrightarrow{b_1} = \beta_1\left(\overrightarrow{R_1} - \overrightarrow{R_n}\right) \\ \vdots \\ \overrightarrow{b_{i \neq n}} = \beta_i\left[\left(\overrightarrow{R_i} - \overrightarrow{R_n}\right) - \sum_{k=1}^{i-1}\left(\left(p_{ik} - p_{nk}\right)\overrightarrow{b_k}\right)\right] \end{cases} \quad \text{Avec : } \begin{cases} \beta_i = \left\|\left(\overrightarrow{R_i} - \overrightarrow{R_n}\right) - \sum_{k=1}^{i-1}\left(\left(p_{ik} - p_{nk}\right)\overrightarrow{b_k}\right)\right\| \\ p_{jk} = \overrightarrow{R_j} \cdot \overrightarrow{b_k} \end{cases}$$

Il est à présent possible d'écrire n'importe quel vecteur $\vec{R}$ dans le sous espace affine associé par la relation suivante :

$$\vec{R} = \overrightarrow{R_n} + \sum_{i=1}^{p}\left(\lambda_i \overrightarrow{b_i}\right)$$

## 1.2 Minimisation du résidu

L'opération suivante consiste à déterminer les $\lambda_k$ tel que la norme $\left\|\overrightarrow{R_{acc}}\right\|$ soit minimale.

$$\forall k \in [1;\, p] \quad \frac{\partial\left(\overrightarrow{R_{acc}} \cdot \overrightarrow{R_{acc}}\right)}{\partial \lambda_k} = 2\frac{\partial \overrightarrow{R_{acc}}}{\partial \lambda_k} \cdot \overrightarrow{R_{acc}} = 0$$

$$\Rightarrow \forall k \in [1;\, p] \quad 2\frac{\partial \overrightarrow{R_{acc}}}{\partial \lambda_k} \cdot \overrightarrow{R_{acc}} = 0$$

Calcul du terme $\dfrac{\partial \overrightarrow{R_{acc}}}{\partial \lambda_k}$

$$\frac{\partial \overrightarrow{R_{acc}}}{\partial \lambda_k} = \frac{\partial\left(\overrightarrow{R_n} + \sum_{i=1}^{p}\left(\lambda_i \overrightarrow{b_i}\right)\right)}{\partial \lambda_k} = \frac{\partial\left(\sum_{i=1}^{p}\left(\lambda_i \overrightarrow{b_i}\right)\right)}{\partial \lambda_k} = \sum_{i=1}^{p}\left(\frac{\partial\left(\lambda_i \overrightarrow{b_i}\right)}{\partial \lambda_k}\right) = \sum_{i=1}^{p}\left(\frac{\partial \lambda_i}{\partial \lambda_k}\overrightarrow{b_i}\right) = \sum_{i=1}^{p}\left(\delta_{ik}\overrightarrow{b_i}\right) = \overrightarrow{b_k}$$

Ce qui donne, en l'introduisant dans la relation précédente et en développant :

$$\forall k \in [1;\, p] \quad \vec{R} \cdot \frac{\partial \overrightarrow{R_{acc}}}{\partial \lambda_k} = 0$$

$$\Rightarrow \left(\overrightarrow{R_n} + \sum_{i=1}^{p}\left(\lambda_i \overrightarrow{b_i}\right)\right) \cdot \overrightarrow{b_k} = 0$$

$$\Rightarrow \overrightarrow{R_n} \cdot \overrightarrow{b_k} + \left[\sum_{i=1}^{p}\left(\lambda_i \overrightarrow{b_i}\right)\right] \cdot \overrightarrow{b_k} = 0$$

$$\Rightarrow \overrightarrow{R_n} \cdot \overrightarrow{b_k} + \left[\sum_{i=1}^{p}\left(\lambda_i\left(\overrightarrow{b_k} \cdot \overrightarrow{b_i}\right)\right)\right] = 0$$

$$\Rightarrow \overrightarrow{R_n} \cdot \overrightarrow{b_k} + \left[\sum_{i=1}^{p}\left(\lambda_i \delta_{ki}\right)\right] = 0$$

$$\Rightarrow \overrightarrow{R_n} \cdot \overrightarrow{b_k} + \lambda_k = 0$$

$$\Rightarrow \lambda_k = -\overrightarrow{R_n} \cdot \overrightarrow{b_k}$$

Précédemment on a déjà calculé $\overrightarrow{R_n} \cdot \overrightarrow{b_k}$ lors de l'orthogonalisation :

$$p_{nk} = \overrightarrow{R_n} \cdot \overrightarrow{b_k}$$

$$\Rightarrow \lambda_k = -p_{nk}$$

Les $\lambda_k$ minimisant $\left\|\vec{R}\right\|$ sont déjà déterminés durant la fabrication de la base orthonormée. Ainsi, le nouveau résidu projeté (corrigé) vaut :

$$\overrightarrow{R_{acc}} = \overrightarrow{R_n} - \sum_{i=1}^{p}\left(p_{ni}\overrightarrow{b_i}\right)$$

Le nouvel incrément de déplacement $\overrightarrow{\Delta U_{acc}}$ sera calculé à l'aide de l'opérateur $\overline{\overline{K}}$ (raideur élastique par défaut dans Cast3M mais on pourrait choisir autre chose) et sera ajouté au déplacement $\overrightarrow{U_n}$ associé au résidu $\overrightarrow{R_n}$ de référence :

$$
\begin{aligned}
&\overrightarrow{\Delta U_{acc}} = \overline{\overline{K^{-1}}} \cdot \overrightarrow{R_{acc}} \\
&\Rightarrow \overrightarrow{U_{i+1}} = \overrightarrow{U_n} + \overrightarrow{\Delta U_{acc}}
\end{aligned}
$$

Le comportement peut à nouveau être évalue par $\overrightarrow{F_{ext}\left(\overrightarrow{U_{i+1}}\right)} - \overrightarrow{F_{\text{int}}\left(\overrightarrow{U_{i+1}}\right)} = \overrightarrow{R_{i+1}}$ et l'opération se poursuit à chaque itération.