

Chapitre 2 (suite)

Aspects algorithmiques de la méthode des éléments finis

4 Grandes étapes d'une résolution numérique par éléments finis

Maintenant que nous avons décrit le principe général de la méthode des éléments finis \mathbb{P}_1 -Lagrange, nous allons rentrer un peu plus dans les détails, et décrire plus précisément les algorithmes nécessaires à l'implémentation. Avant cela, commençons par donner le schéma général d'une simulation par éléments finis.

Étape 1

Description de la géométrie. On parle de "Conception Assistée par Ordinateur" (CAO) ou, en anglais de "Computer Aided Design" (CAD).

Étape 2

Génération du maillage. Cette étape peut être plus gourmande en calcul, ou en ressources humaines, que la résolution du système linéaire proprement dite.

Étape 3

Construction de la connectique: chargement du maillage et numérotation des inconnues.

Étape 4

Assemblage de la matrice du problème: calcul des intégrales $a(\varphi_k, \varphi_j)$.

Étape 5

Assemblage du second membre: calcul des intégrales $\ell(\varphi_j)$.

Étape 6

Prise en compte des conditions de bord: application d'un post-traitement si nécessaire, comme dans le cas des conditions de Dirichlet.

Étape 7

Résolution du système linéaire.

Étape 8

Post-traitement/visualisation des résultats (affichages de cartes de champs, calcul d'erreur, etc...)

Remarques Les étapes 1, 2 et 7 ci-dessus peuvent être sous-traitées par des logiciels/bibliothèques externes. Les étapes les plus coûteuses en temps calcul sont bien souvent les étapes 2 et 7. Mentionnons enfin que le cycle de résolution décrit ci-dessous est parfois imbriqué dans un processus itératif de plus haut niveau: optimisation, stratégie adaptative, second membres multiples, etc...

5 Structure type d'un fichier de maillage

La génération de maillage étant une étape qui précède directement la résolution par éléments finis, il s'agit d'une donnée d'entrée d'un code éléments finis (à moins qu'un tel code intègre la génération de maillage, exemple: FreeFEM++). A l'issue de sa génération, les données décrivant le maillage sont stockées dans un fichier selon un certain format. Il existe autant de formats de fichier de maillage que de logiciels de génération de maillage. Les différents formats de fichier de maillage suivent cependant bien souvent les mêmes principes élémentaires. Pour s'en faire une idée, nous donnons ici un format type de fichier de maillage.

#Nombre de noeuds

Ns

#Coordonnees des noeuds

x_1	y_1	z_1
x_2	y_2	z_2
\vdots	\vdots	\vdots
x_{Ns}	y_{Ns}	z_{Ns}

#Nombre de triangles

Nt

#Numeros des sommets de chaque triangle

$I_{1,1}$	$I_{1,2}$	$I_{1,3}$
$I_{2,1}$	$I_{2,2}$	$I_{2,3}$
\vdots	\vdots	\vdots
$I_{Nt,1}$	$I_{Nt,2}$	$I_{Nt,3}$

6 Assemblage des matrices

Dans toute la suite on va supposer que les noeuds sont numérotés de $1 \dots Ns$ de sorte que $\mathcal{S}_h = \{\mathbf{s}_j\}_{j=1 \dots Ns}$. Dans un souci de concision, on notera φ_j au lieu de $\varphi_{\mathbf{s}_j}$. La matrice $A = (A_{j,k})_{j,k=1 \dots Ns}$ du système (4) se décompose en une somme de matrices de rang 1. Si $\mathbf{e}_j \in \mathbb{R}^{Ns}$, $j = 1 \dots Ns$ désignent les vecteurs de la base canonique, on a en effet:

$$A = \sum_{j=1}^{Ns} \sum_{k=1}^{Ns} a(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T$$

Il va bien évidemment falloir assembler cette matrice i.e. calculer ses coefficients, et stocker ces derniers dans une structure de données qui représente (dans le programme) la matrice. Pour ce faire, on présente un premier algorithme, qui semble naturel en première approche. Mais, en pratique, *cet algorithme ne marche pas*.

Algorithme d'assemblage qui ne marche pas

```

A = 0
for j = 1...Ns
  for k = 1...Ns
    Aj,k = Aj,k + a(φk, φj)
  endfor
endfor

```

(7)

Cet algorithme ne marche pas en pratique car il est de complexité $\mathcal{O}(\text{Ns}^2)$. Ceci n'est pas raisonnable. Et de toute façon, nous allons voir comment obtenir le même résultat mais avec un algorithme beaucoup plus rapide de complexité linéaire.

Nous allons exploiter le caractère creux de la matrice des éléments finis. On dit qu'une matrice est creuse lorsqu'elle contient moins que $\mathcal{O}(\text{Ns})$ entrées non-nulle. Observons que, pour j, k donnés, si aucun triangle n'admet à la fois \mathbf{s}_j et \mathbf{s}_k comme sommet, alors $a(\varphi_k, \varphi_j) = 0$. La forme $a(\cdot, \cdot)$ peut donc se décomposer en sommes de contributions locales

$$a(u, v) = \sum_{\tau \in \mathcal{T}_h} a_\tau(u, v)$$

où $a_\tau(\cdot, \cdot)$ renvoie à la contribution de $a(\cdot, \cdot)$ localisée sur l'élément τ . Une telle décomposition est possible car l'expression de $a(\cdot, \cdot)$ prend la forme d'un nombre fini d'intégrales volumiques. Par exemple pour l'équation (6), on a

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla \bar{v} + u \bar{v} \, d\mathbf{x}$$

et alors $a_\tau(u, v) := \int_{\tau} \nabla u \cdot \nabla \bar{v} + u \bar{v} \, d\mathbf{x}$

Supposons que les éléments de la triangulation sont numérotées conformément au fichier de maillage décrit à la section 5, de sorte que $\mathcal{T}_h = \{\tau_q\}_{q=1 \dots \text{Nt}}$. La matrice du problème se décompose alors comme

$$\begin{aligned}
\mathbf{A} &= \sum_{j=1}^{\text{Ns}} \sum_{k=1}^{\text{Ns}} a(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\
&= \sum_{j=1}^{\text{Ns}} \sum_{k=1}^{\text{Ns}} \sum_{q=1}^{\text{Nt}} a_{\tau_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\
&= \sum_{q=1}^{\text{Nt}} \sum_{j=1}^{\text{Ns}} \sum_{k=1}^{\text{Ns}} a_{\tau_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T
\end{aligned}$$
(8)

Notons que, pour un τ_q fixé, on a $a_{\tau_q}(\varphi_k, \varphi_j) = 0$ sauf si $\tau_q \cap \text{supp}(\varphi_j) \neq \emptyset$ et $\tau_q \cap \text{supp}(\varphi_k) \neq \emptyset$, ce qui n'arrivera que si $j, k \in \{I_{q,1}, I_{q,2}, I_{q,3}\}$. Ainsi, à q fixé, la double somme sur ci-dessus $\sum_{j=1}^{N_s} \sum_{k=1}^{N_s}$ se réduit à une double somme sur l'ensemble $\{I_{q,1}, I_{q,2}, I_{q,3}\}$

$$\sum_{j=1}^{N_s} \sum_{k=1}^{N_s} a_{\tau_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T = \sum_{l=1}^3 \sum_{m=1}^3 a_{\tau_q}(\varphi_{I_{q,m}}, \varphi_{I_{q,l}}) \mathbf{e}_{I_{q,l}} \cdot \mathbf{e}_{I_{q,m}}^T.$$

On met ici en évidence le caractère creux de la matrice du problème. Cette matrice admet in fine la représentation suivante

$$\mathbf{A} = \sum_{q=1}^{N_t} \sum_{l=1}^3 \sum_{m=1}^3 a_{\tau_q}(\varphi_{I_{q,m}}, \varphi_{I_{q,l}}) \mathbf{e}_{I_{q,l}} \cdot \mathbf{e}_{I_{q,m}}^T$$

La somme ci-dessus contient $9 \times N_t$ termes, c'est-à-dire beaucoup moins que les $N_s \times N_s$ de l'algorithme (7). Ceci nous amène donc à l'algorithme suivant. Il s'agit de l'algorithme standard d'assemblage des matrices éléments finis.

Algorithme d'assemblage qui marche

```

A = 0
for q = 1 ... Nt
  for l = 1 ... 3
    for m = 1 ... 3
      AIq,l, Iq,m = AIq,l, Iq,m + aτq(φIq,m, φIq,l)
    endfor
  endfor
endfor

```

(9)

7 Assemblage des termes surfaciques

Dans certains cas la forme sesquilinéaire $a(\cdot, \cdot)$ contient des termes de bord. Pour fixer les idées, étant donné une partie du bord $\Gamma \subset \partial\Omega$ de mesure de surface non-nulle et $\lambda > 0$, considérons dans cette section le problème suivant,

$$\begin{cases} -\Delta u + u = f & \text{dans } \Omega \\ \partial_{\mathbf{n}} u = 0 & \text{sur } \Sigma = \partial\Omega \setminus \Gamma \\ \partial_{\mathbf{n}} u + \lambda u = 0 & \text{sur } \Gamma \end{cases} \quad (10)$$

Ce problème ne diffère de (6) que par les conditions aux limites. En recherchant une formulation variationnelle en suivant la méthodologie exposée au chapitre précédent, on aboutit à la forme sesquilinéaire suivante

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla \bar{v} + u \bar{v} \, d\mathbf{x} + b(u, v)$$

avec $b(u, v) := \lambda \int_{\Gamma} u \bar{v} \, d\sigma.$

Nous ferons deux hypothèses pratiques. Nous verrons plus tard durant ce cours comment s'en affranchir. On suppose premièrement que le maillage ait été généré de telle sorte que Γ soit une réunion d'arêtes du maillage

$$\begin{aligned}\bar{\Gamma} &= \bar{\gamma}_1 \cup \bar{\gamma}_2 \cup \dots \cup \bar{\gamma}_{Ng} \\ \text{avec } \gamma_j \cap \gamma_k &= \emptyset \text{ pour } j \neq k.\end{aligned}\tag{11}$$

On suppose par ailleurs que le générateur de maillage nous fournit le numéro des noeuds de chaque γ_j en incorporant une section supplémentaire dans le fichier de maillage qu'il produit, cette section prenant la forme suivante

```
#Nombre arete condition Robin
Ng
```

```
#Numeros des sommets de chaque arete
```

```
J1,1      J1,2
J2,1      J2,2
⋮         ⋮
JNg,1     JNg,2
```

Nous avons noté $b(\cdot, \cdot)$ la contribution surfacique dans la forme sesquilinéaire $a(\cdot, \cdot)$. Cette forme peut se décomposer en contributions élémentaires associées à chaque γ_j ,

$$b(u, v) = \sum_{j=1}^{Ng} b_{\gamma_j}(u, v)$$

Soit $B = (B_{j,k}) \in \mathbb{C}^{Ns \times Ns}$ la matrice associée à cette forme sesquilinéaire $B_{j,k} = b(\varphi_k, \varphi_j)$. Il faut bien noter que $b(\varphi_k, \varphi_j) = 0$ dès que \mathbf{s}_j ou \mathbf{s}_k n'appartient pas à Γ , ce qui est le cas de la plupart des noeuds du maillage, de sorte que la matrice B contient beaucoup de termes nulle, encore plus que A . Par ailleurs, on a aussi $b(\varphi_k, \varphi_j) = 0$ dès que \mathbf{s}_j et \mathbf{s}_k ne sont pas sommets d'une même arête de Γ . On peut alors dérouler un calcul semblable à (8), à savoir

$$\begin{aligned}B &= \sum_{j=1}^{Ns} \sum_{k=1}^{Ns} b(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T = \sum_{j=1}^{Ns} \sum_{k=1}^{Ns} \sum_{q=1}^{Ng} b_{\gamma_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\ &= \sum_{q=1}^{Ng} \sum_{j=1}^{Ns} \sum_{k=1}^{Ns} b_{\gamma_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\ &= \sum_{q=1}^{Ng} \sum_{l=1,2} \sum_{m=1,2} b_{\gamma_q}(\varphi_{J_{q,l}}, \varphi_{J_{q,m}}) \mathbf{e}_{J_{q,m}} \cdot \mathbf{e}_{J_{q,l}}^T\end{aligned}$$

Ici encore on a obtenu une somme qui ne compte que $2 \times Ng$ termes. Le cout d'assemblage correspondant est donc linéaire. L'algorithme d'assemblage correspondant s'écrit alors

Algorithme d'assemblage des termes de bord

```

B = 0
for q = 1...Ng
  for l = 1...2
    for m = 1...2
      BJq,l,Jq,m = BJq,l,Jq,m + bγq(φJq,m, φJq,l)
    endfor
  endfor
endfor

```

(12)

8 Assemblage du second membre

Dans le système (4) que l'on obtient à l'issue d'une discrétisation de Galerkin, outre la matrice A, il convient également d'assembler la matrice F. L'assemblage de ce terme suit le même principe de base que pour A, consistant à exploiter la localité des fonctions de forme. A noter aussi que, de même que $a(\cdot, \cdot)$, le second membre $\ell(\cdot)$ peut lui-même admettre à la fois des termes volumiques et surfaciques.

Plaçons-nous pour simplifier dans le cas où $\ell(\cdot)$ ne contient que des termes volumiques, et peut se décomposer en termes localisés dans chaque triangle du maillage comme

$$\ell(v) = \sum_{q=1}^{N_t} \ell_{\tau_q}(v)$$

La matrice colonne $F = (F_j) \in \mathbb{C}^{N_s}$, associé au second membre par la relation $F_j = \ell(\varphi_j)$, se décompose alors de la manière suivante

$$\begin{aligned}
F &= \sum_{j=1}^{N_s} \ell(\varphi_j) \mathbf{e}_j = \sum_{j=1}^{N_s} \sum_{q=1}^{N_t} \ell_{\tau_q}(\varphi_j) \mathbf{e}_j \\
&= \sum_{q=1}^{N_t} \sum_{j=1}^{N_s} \ell_{\tau_q}(\varphi_j) \mathbf{e}_j = \sum_{q=1}^{N_t} \sum_{l=1}^3 \ell_{\tau_q}(\varphi_{I_{q,l}}) \mathbf{e}_{I_{q,l}}
\end{aligned}$$

On en tire l'algorithme suivant.

Algorithme d'assemblage du second membre

```

F = 0
for q = 1...Nt
  for l = 1...3
    FIq,l = FIq,l + ℓτq(φIq,l)
  endfor
endfor

```

(13)