

Chapter 8

Algorithmique dans les 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 d'un code de simulation par éléments finis. Nous 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...).

Les étapes 1, 2 et 7 ci-dessus sont bien souvent 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...

8.1 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++¹). 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 sommets
nv
```

```
#Coordonnees des sommets
```

```
x1  y1  z1
x2  y2  z2
⋮   ⋮   ⋮
xnv ynv znv
```

```
#Nombre elements
ne
```

```
#Indices des sommets de chaque element
```

```
I1,0  I1,1  ...  I1,d
I2,0  I2,1  ...  I2,d
⋮      ⋮      ⋮
Ine,0 Ine,1  ...  Ine,d
```

Dans le format de fichier type ci-dessus, d fait référence à la dimension de l'espace ambiant. Dans le cas où nous traitons un problème bi-dimensionnel on a $d = 2$ et les éléments sont des triangles qui admettent donc $d + 1 = 3$ sommets. Si nous traitons un problème tri-dimensionnel, alors $d = 3$ et les éléments sont des tétraèdres qui admettent donc $d + 1 = 4$ sommets.

8.2 Assemblage des matrices

Dans la suite, nous supposons que les sommets sont numérotés consécutivement selon $1 \dots nv$ de sorte que $\mathcal{S}_h = \{\mathbf{s}_j\}_{j=1 \dots nv}$. Dans un souci de concision, nous noterons φ_j au lieu de $\varphi_{\mathbf{s}_j}$. La matrice $A = (A_{j,k})_{j,k=1 \dots nv}$ du système (7.4) se décompose en matrices de rang 1. Si $\mathbf{e}_j \in \mathbb{R}^{nv}$, $j = 1 \dots nv$ désignent les vecteurs de la base canonique, nous avons en effet:

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

¹<https://freefem.org/>

Il est bien clair qu'il nous faut assembler cette matrice c'est-à-dire calculer ses coefficients et les stocker dans une structure de donnée qui a vocation à modéliser cette matrice i.e. à la représenter dans le programme. Pour faire cela, nous proposons ci-dessous un algorithme qui semble naturel à première vue. Cependant, en pratique, *cet algorithme est naïf et ne fonctionne pas*.

Algorithm 1 Algorithme d'assemblage qui **ne fonctionne pas**

```

A = 0
for j = 1 ... nv do
  for k = 1 ... nv do
    Aj,k = Aj,k + a(φk, φj)
  end for
end for

```

Cet algorithme ne fonctionne pas en pratique car il est de complexité $\mathcal{O}(nv^2)$. 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 lorsque son ratio (termes non-nuls)/(termes nuls) est très petit. 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 dans le cas de (7.14), on a

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla \bar{v} + \mu^2 u \bar{v} d\mathbf{x}$$

et alors $a_\tau(u, v) := \int_{\tau} \nabla u \cdot \nabla \bar{v} + \mu^2 u \bar{v} d\mathbf{x}$

Supposons que les éléments de la triangulation soient numérotés selon le format décrit dans la section, avec $\mathcal{T}_h = \{\tau_q\}_{q=1 \dots ne}$. Notons ici encore $\mathbf{e}_j, j = 1 \dots nv$ les vecteurs de la base canonique de \mathbb{C}^{nv} . La matrice du problème se décompose alors de la manière suivante

$$\begin{aligned}
A &= \sum_{j=1}^{nv} \sum_{k=1}^{nv} a(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\
&= \sum_{j=1}^{nv} \sum_{k=1}^{nv} \sum_{q=1}^{ne} a_{\tau_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\
&= \sum_{q=1}^{ne} \sum_{j=1}^{nv} \sum_{k=1}^{nv} a_{\tau_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T
\end{aligned} \tag{8.1}$$

Notez bien que pour un élément τ_q fixé, nous avons $a_{\tau_q}(\varphi_k, \varphi_j) = 0$ sauf si $\tau_q \cap \text{supp}(\varphi_j) \neq \emptyset$ et $\tau_q \cap \text{supp}(\varphi_k) \neq \emptyset$, ce qui se produit seulement si $j, k \in \{I_{q,0}, \dots, I_{q,d}\}$. Par conséquent,

pour un q fixé, la double somme $\sum_{j=1}^{\text{nv}} \sum_{k=1}^{\text{nv}}$ dans le calcul ci-dessus se réduit à une double somme sur l'ensemble $\{I_{q,0}, \dots, I_{q,d}\}$

$$\sum_{j=1}^{\text{nv}} \sum_{k=1}^{\text{nv}} a_{\tau_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T = \sum_{l=0}^d \sum_{m=0}^d a_{\tau_q}(\varphi_{I_{q,m}}, \varphi_{I_{q,l}}) \mathbf{e}_{I_{q,l}} \cdot \mathbf{e}_{I_{q,m}}^T.$$

Ici nous exploitons à fond la parcimonie (le caractère creux) de la matrice du problème. A la fin, la matrice admet la représentation suivante

$$A = \sum_{q=1}^{\text{ne}} \sum_{l=0}^d \sum_{m=0}^d a_{\tau_q}(\varphi_{I_{q,m}}, \varphi_{I_{q,l}}) \mathbf{e}_{I_{q,l}} \cdot \mathbf{e}_{I_{q,m}}^T$$

La somme ci-dessus fait intervenir $(d+1) \times (d+1) \times \text{ne}$ termes, ce qui est beaucoup moins que les $\text{nv} \times \text{nv}$ itérations de l'algorithme (1). Ceci nous amène donc naturellement à l'algorithme alternatif suivant. C'est l'algorithme standard pour l'assemblage des matrices éléments finis.

Algorithm 2 Algorithme d'assemblage qui **fonctionne**

```

A = 0
for q = 1 ... ne do
  for l = 0 ... d do
    for m = 0 ... d do
      j = Iq,l
      k = Iq,m
      Aj,k = Aj,k + aτq(φk, φj)
    end for
  end for
end for

```

8.3 Assemblage des termes de surface

Dans certains cas, la forme sesquilinéaire $a(\cdot, \cdot)$ fait intervenir des termes de bord. Pour fixer les idées, étant donnée une portion du bord $\Gamma \subset \partial\Omega$ de mesure de surface non-nulle et $\lambda > 0$, considérons icile problème suivant

$$\begin{cases} -\Delta u + \mu^2 u = f & \text{in } \Omega \\ \partial_{\mathbf{n}} u = 0 & \text{sur } \Sigma = \partial\Omega \setminus \Gamma \\ \partial_{\mathbf{n}} u + \lambda u = 0 & \text{sur } \Gamma. \end{cases} \quad (8.2)$$

Ce problème diffère de (7.13) uniquement à travers les conditions de bord. En suivant la méthodologie que nous avons présenté précédemment dans le cours, on obtient la formulation variationnelle suivante pour ce problème aux limites

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

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

Nous formulons maintenant deux hypothèses pratiques. Premièrement nous supposons que le maillage a été généré de telle sorte que Γ soit exactement une réunion de faces du (bord du) maillage

$$\begin{aligned}\bar{\Gamma} &= \bar{\gamma}_1 \cup \bar{\gamma}_2 \cup \dots \cup \bar{\gamma}_{N_g} \\ \text{with } \gamma_j \cap \gamma_k &= \emptyset \text{ for } j \neq k.\end{aligned}\tag{8.3}$$

Ici Γ est une surface de dimension $(d-1)$, recouverte par une triangulation de dimension $(d-1)$, qui n'est cependant pas contenue dans l'espace \mathbb{R}^{d-1} . Dans le cas où $d = 3$ par exemple i.e. nous considérons un problème tri-dimensionnel, alors Γ est une portion de surface bi-dimensionnelle $2D$ recouverte par un maillage en triangle.

Nous supposons par ailleurs que le générateur de maillage nous fournit en sortie les indices des sommets de chaque face γ_j en incorporant une section supplémentaire au fichier de maillage. Cette section pourrait prendre la forme suivante.

#Nombre de faces avec condition de Robin

nb

#Indices des sommets de chaque face

J _{1,0}	J _{1,1}	...	J _{1,d-1}
J _{2,0}	J _{2,1}	...	J _{2,d-1}
⋮	⋮		⋮
J _{nb,0}	J _{nb,1}	...	J _{nb,d-1}

Nous avons noté $b(\cdot, \cdot)$ le terme de bord dans la forme sesquilinéaire $a(\cdot, \cdot)$. Il peut se décomposer en contributions élémentaires attachées à chaque γ_j ,

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

Soit $B = (B_{j,k}) \in \mathbb{C}^{nv \times nv}$ la matrice associée à la forme sesquilinéaire de ce terme de bord $B_{j,k} = b(\varphi_k, \varphi_j)$. Observons que $b(\varphi_k, \varphi_j) = 0$ dès lors que \mathbf{s}_j ou bien \mathbf{s}_k n'appartient pas à Γ , ce qui se produit pour presque tous les sommets du maillage, si bien que la matrice B contient encore plus de zéros que A . De plus on a $b(\varphi_k, \varphi_j) = 0$ quand \mathbf{s}_j et \mathbf{s}_k n'appartiennent pas à la même face de Γ . Nous pouvons alors utiliser un algorithme similaire à (8.1) pour l'assemblage de B . On a

$$\begin{aligned}B &= \sum_{j=1}^{nv} \sum_{k=1}^{nv} b(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T = \sum_{j=1}^{nv} \sum_{k=1}^{nv} \sum_{q=1}^{nb} b_{\gamma_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T \\ &= \sum_{q=1}^{nb} \sum_{j=1}^{nv} \sum_{k=1}^{nv} b_{\gamma_q}(\varphi_k, \varphi_j) \mathbf{e}_j \cdot \mathbf{e}_k^T = \sum_{q=1}^{nb} \sum_{l=0}^{d-1} \sum_{m=0}^{d-1} b_{\gamma_q}(\varphi_{J_{q,l}}, \varphi_{J_{q,m}}) \mathbf{e}_{J_{q,m}} \cdot \mathbf{e}_{J_{q,l}}^T\end{aligned}$$

A nouveau nous avons obtenu une somme ne faisant intervenir que $d \times d \times nb$ termes. Le cout d'assemblage est donc linéaire. L'algorithme correspondant s'écrit comme suit.

Algorithm 3 Assemblage des termes de bord

```

B = 0
for q = 1 ... nb do
  for l = 0, ..., d - 1 do
    for m = 0, ..., d - 1 do
      j = Jq,l
      k = Jq,m
      Bj,k = Bj,k + bγq(φk, φj)
    end for
  end for
end for

```

8.4 Assemblage des termes source

Pour traiter le système (7.4) obtenu après discrétisation de Galerkin, outre la matrice A , nous avons également besoin de la matrice F . L'assemblage de ce terme suit les mêmes principes que pour A qui consistent à prendre en compte la localité des fonctions de forme. De la même manière que pour $a(\cdot, \cdot)$, le second membre $\ell(\cdot)$ peut lui aussi admettre des termes de volume et de surface mais, pour simplifier le propos, supposons que seuls des termes de volume interviennent. Alors $\ell(\cdot)$ peut être décomposé sur chaque élément du maillage de la manière suivante

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

La matrice colonne $F = (F_j) \in \mathbb{C}^{nv}$, associée au terme source au travers de la relation $F_j = \ell(\varphi_j)$, se décompose alors de la manière suivante

$$\begin{aligned}
F &= \sum_{j=1}^{nv} \ell(\varphi_j) \mathbf{e}_j = \sum_{j=1}^{nv} \sum_{q=1}^{ne} \ell_{\tau_q}(\varphi_j) \mathbf{e}_j \\
&= \sum_{q=1}^{ne} \sum_{j=1}^{nv} \ell_{\tau_q}(\varphi_j) \mathbf{e}_j = \sum_{q=1}^{ne} \sum_{l=0}^d \ell_{\tau_q}(\varphi_{I_{q,l}}) \mathbf{e}_{I_{q,l}}
\end{aligned}$$

Nous en déduisons l'algorithme suivant.

Algorithm 4 Assemblage du second membre

```

F = 0
for q = 1 ... ne do
  for l = 0, ..., d do
    j = Iq,l
    Fj = Fj + ℓτq(φj)
  end for
end for

```

8.5 Prise en compte des conditions de Dirichlet

Les conditions aux limites de Neumann ne requièrent aucun traitement particulier. A contrario, les conditions au bord de Dirichlet réclament, elles, un traitement spécifique. Pour fixer les idées considérons une portion du bord du domaine $\Gamma \subset \partial\Omega$ de mesure de surface non nulle. Supposons que l'on souhaite résoudre le problème

$$\begin{cases} -\Delta u = f & \text{dans } \Omega, \\ \partial_{\mathbf{n}} u = 0 & \text{sur } \Sigma = \partial\Omega \setminus \Gamma, \\ u = 0 & \text{sur } \Gamma. \end{cases} \quad (8.4)$$

Une formulation variationnelle possible s'écrit alors

$$\begin{cases} \text{Trouver } u \in H_{0,\Gamma}^1(\Omega) \text{ tel que} \\ a(u, v) = \ell(v) \quad \forall v \in H_{0,\Gamma}^1(\Omega). \end{cases} \quad (8.5)$$

$$\text{avec } a(u, v) = \int_{\Omega} \nabla u \cdot \nabla \bar{v} d\mathbf{x} \quad \text{et} \quad \ell(v) = \int_{\Omega} f \bar{v} d\mathbf{x}.$$

Supposons que le maillage ait été généré de sorte que la partie du bord Γ associée à la condition de Dirichlet soit exactement formé d'une réunion de faces du (bord du) maillage $\Gamma = \cup_{j=1}^{nb} \bar{\gamma}_j$ comme en (8.3). Soit par ailleurs $\mathcal{S}_h(\Gamma) := \{\mathbf{s} \in \mathcal{S}_h \mid \mathbf{s} \in \bar{\Gamma}\}$ l'ensemble des sommets du maillage localisés sur Γ . L'espace variationnel discret "naturellement" associé à (8.5) est alors

$$\begin{aligned} V_h^\Gamma &:= V_h \cap H_{0,\Gamma}^1(\Omega) = \text{vect}\{\varphi_{\mathbf{s}} \mid \mathbf{s} \in \mathcal{S}_h \setminus \mathcal{S}_h(\Gamma)\} \\ &= \{v \in \mathcal{C}^0(\bar{\Omega}), v|_{\tau} \in \mathbb{P}_1(\tau) \forall \tau \in \mathcal{T}_h \text{ et } v|_{\Gamma} = 0\} \\ \mathfrak{I}_\Gamma &:= \{j \in \{1 \dots nv\} \mid \mathbf{s}_j \in \mathcal{S}_h \setminus \mathcal{S}_h(\Gamma)\} \end{aligned}$$

Une formulation variationnelle discrète de (8.5) peut être obtenue simplement en remplaçant $H_{0,\Gamma}^1(\Omega)$ par V_h^Γ , ce qui conduit à la formulation suivante

$$\begin{cases} \text{Find } u_h \in V_h^\Gamma \text{ such that} \\ a(u_h, v_h) = \ell(v_h) \quad \forall v_h \in V_h^\Gamma. \end{cases} \quad (8.6)$$

Ré-écrivons cette identité variationnelle sous forme matricielle. Nous avons $\varphi_j \in V_h^\Gamma \iff j \in \mathfrak{I}_\Gamma$ et, en notant $U = (u_h(\mathbf{s}_1), \dots, u_h(\mathbf{s}_{nv}))^\top$, on a $u_h \in V_h^\Gamma \iff \mathbf{e}_j^\top U = 0 \quad \forall j \in \{1, \dots, nv\} \setminus \mathfrak{I}_\Gamma$. Par conséquent (8.6) peut être ré-écrit de façon équivalente comme

$$\begin{aligned} \mathbf{e}_j^\top U &= 0 & \forall j \in \{1, \dots, nv\} \setminus \mathfrak{I}_\Gamma, \\ \mathbf{e}_j^\top AU &= \mathbf{e}_j^\top F & \forall j \in \mathfrak{I}_\Gamma. \end{aligned} \quad (8.7)$$

où A est la matrice que nous aurions obtenue dans le cas d'une condition au bord de Neumann i.e. $A = (A_{j,k})_{j,k=1,\dots,nv}$ avec $A_{j,k} = a(\varphi_k, \varphi_j)$. A présent introduisons une matrice diagonale $D_\Gamma := \sum_{j \in \mathfrak{I}_\Gamma} \mathbf{e}_j \cdot \mathbf{e}_j^\top \in \mathbb{R}^{nv \times nv}$ qui agit comme un projecteur $D_\Gamma \cdot D_\Gamma = D_\Gamma$. Alors la première équation de (8.7) se ré-écrit $(\text{Id} - D_\Gamma)U = 0$ et la seconde équation de (8.7) se ré-écrit $D_\Gamma AU = D_\Gamma F$. Puisque $D_\Gamma U = U$, ces deux équations sont équivalente à la suivante

$$\begin{aligned} (D_\Gamma A D_\Gamma + \text{Id} - D_\Gamma)U &= D_\Gamma F \\ \text{avec } D_\Gamma &:= \sum_{j \in \mathfrak{I}_\Gamma} \mathbf{e}_j \cdot \mathbf{e}_j^\top \end{aligned} \quad (8.8)$$

Le procédé qui fait passer de la matrice A à la matrice $\dot{A} := D_\Gamma A D_\Gamma + \text{Id} - D_\Gamma$ est appelé *pseudo-élimination*.

En pratique, réaliser un produit matrice-matrice entre matrices creuses peut être réalisé avec en complexité $\mathcal{O}(nv)$ (linéaire), si bien que le produit $D_\Gamma A D_\Gamma$ peut être évalué de manière rapide. Mentionnons cependant que la matrice pseudo-éliminée $D_\Gamma A D_\Gamma + \text{Id} - D_\Gamma$ peut aussi bien être obtenu en partant de la matrice A dans laquelle on met à zéro les lignes et les colonnes associées aux sommets s_j vérifiant $j \notin \mathfrak{I}_\Gamma$ (ce qui correspond à la transformation $A \rightarrow D_\Gamma A D_\Gamma$), et ensuite en ajoutant des 1 sur la diagonale pour ces indices (ce qui correspond à la transformation $D_\Gamma A D_\Gamma \rightarrow D_\Gamma A D_\Gamma + \text{Id} - D_\Gamma$). Ci-dessous nous donnons une implémentation possible qui consiste à intégrer directement la pseudo-élimination dans l'algorithme d'assemblage ce qui permet de s'affranchir du produit matrice-matrice.

Algorithm 5 Assemblage avec pseudo-élimination

```

 $\dot{A} = 0$ 
for  $q = 1 \dots n_e$  do
  for  $l = 0, \dots, d$  do
    for  $m = 0, \dots, d$  do
       $j = I_{q,l}$ 
       $k = I_{q,m}$ 
      interior =  $(j \in \mathfrak{I}_\Gamma \text{ and } k \in \mathfrak{I}_\Gamma)$ 
      if interior then
         $\dot{A}_{j,k} = \dot{A}_{j,k} + a_{\tau_q}(\varphi_k, \varphi_j)$ 
      else if  $j == k$  then
         $\dot{A}_{j,k} = \dot{A}_{j,k} + 1$ 
      end if
    end for
  end for
end for

```

8.6 Calcul des interactions élémentaires

Nous avons réduit l'assemblage de la matrice et du second membre au calcul de termes de la forme $a_\tau(\varphi_k, \varphi_j)$ localisés sur un seul élément τ . Nous allons à présent examiner comment ces termes peuvent être calculés. Dans ce qui suit, nous allons prendre explicitement avantage du fait que nous discrétisons par une méthode d'éléments finis \mathbb{P}_1 -Lagrange (et pas par des éléments \mathbb{P}_k -Lagrange d'ordre plus élevé).

Dans ce paragraphe on considère un d -simplexe $\tau \in \mathcal{T}_h$ non-dégénéré dont les sommets seront notés $s_0^\tau, \dots, s_d^\tau$. Par conséquent, si $\tau = \tau_q$, nous aurons $s_k^\tau = s_{I_{q,k}}, k = 0, \dots, d$. Nous adopterons une notation circulaire en ce qui concerne les indices i.e. par convention $s_{k+(d+1)}^\tau = s_k^\tau$ pour tout entier $j \in \mathbb{Z}$.

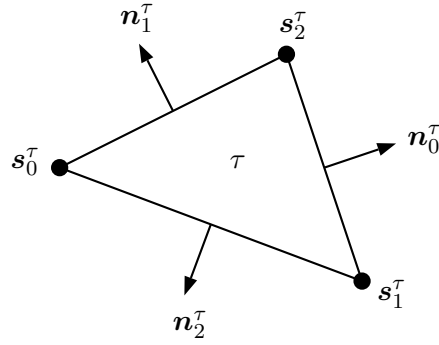
Fonctions de forme locale

Nous commençons par obtenir une expression explicite des fonctions de formes locales à l'élément τ . Nous noterons $\varphi_j^\tau(\mathbf{x})$ l'unique fonction de $\mathbb{P}_1(\tau)$ qui satisfait $\varphi_j^\tau(\mathbf{s}_j^\tau) = 1$ ainsi que $\varphi_j^\tau(\mathbf{s}_k^\tau) = 0$ for $j \neq k \pmod{d+1}$. D'après le lemme 7.7, il y a existence et unicité d'une telle fonction. Pour chaque j , nous considérons un vecteur \mathbf{n}_j^τ qui est normal à la face opposée au sommet \mathbf{s}_j^τ i.e.

$$(\mathbf{s}_k^\tau - \mathbf{s}_l^\tau) \cdot \mathbf{n}_j^\tau = 0 \quad (8.9)$$

pour tout k, l tels que $k \neq j, l \neq j$ (modulo $d+1$).

Cette situation est représentée sur la figure ci-dessous. Bien entendu le vecteur \mathbf{n}_j^τ peut être choisi comme le vecteur unitaire sortant normal à la face, mais ce n'est pas une obligation et ce vecteur n'a en particulier pas besoin d'être de norme 1.



Considérons \mathbb{R}^d pour $d = 1, 2, 3$, que nous regarderons comme plongé dans l'espace \mathbb{R}^3 , l'opération " \times " renvoie alors au produit vectoriel de \mathbb{R}^3 . Voici une manière de choisir le vecteur \mathbf{n}_j^τ en pratique (bien que ce ne soit pas le seul choix possible).

- **d=1:** $\mathbf{n}_j^\tau = \mathbf{s}_{j+1}^\tau - \mathbf{s}_j^\tau$.
- **d=2:** $\mathbf{n}_j^\tau = \mathbf{e}_3 \times (\mathbf{s}_{j+1}^\tau - \mathbf{s}_{j+2}^\tau)$.
- **d=3:** $\mathbf{n}_j^\tau = (\mathbf{s}_{j+1}^\tau - \mathbf{s}_{j+2}^\tau) \times (\mathbf{s}_{j+1}^\tau - \mathbf{s}_{j+3}^\tau)$

Un point important ici est que le vecteur \mathbf{n}_j^τ peut être évalué directement à partir de la seule connaissance des coordonnées des sommets de τ que nous avons à notre disposition grâce au fichier de maillage. On peut alors vérifier de manière directe que la fonction de forme locale attachée à \mathbf{s}_j^τ admet l'expression suivante

$$\varphi_j^\tau(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{s}_{j+1}^\tau) \cdot \mathbf{n}_j^\tau}{(\mathbf{s}_j^\tau - \mathbf{s}_{j+1}^\tau) \cdot \mathbf{n}_j^\tau} \quad (8.10)$$

Cette expression ne dépend pas du choix de \mathbf{n}_j^τ à condition qu'il respecte les conditions d'orthogonalité (8.9). Le gradient de la fonction de forme locale peut être facilement calculée à partir de cette expression: on a $\nabla \varphi_j^\tau = \mathbf{n}_j^\tau / ((\mathbf{s}_j^\tau - \mathbf{s}_{j+1}^\tau) \cdot \mathbf{n}_j^\tau)$ qui est un vecteur constant de \mathbb{R}^d .

Exemple Considérons le cas où τ est le triangle de référence de sommets $\mathbf{s}_0^\tau = (0, 0)$, $\mathbf{s}_1^\tau = (1, 0)$ et $\mathbf{s}_2^\tau = (0, 1)$. Dans ce cas nous avons

$$\begin{aligned}\varphi_0^\tau(\mathbf{x}) &= 1 - x_1 - x_2, & \nabla\varphi_0^\tau &= -\mathbf{e}_1 - \mathbf{e}_2, \\ \varphi_1^\tau(\mathbf{x}) &= x_1, & \nabla\varphi_1^\tau &= \mathbf{e}_1, \\ \varphi_2^\tau(\mathbf{x}) &= x_2, & \nabla\varphi_2^\tau &= \mathbf{e}_2.\end{aligned}$$

Matrice de rigidité élémentaire

Examinons comment calculer les termes de la forme $\int_\tau \nabla\varphi_j^\tau \cdot \nabla\varphi_k^\tau d\mathbf{x}$. Observons simplement que, d'après (8.10), chaque $\nabla\varphi_j^\tau \in \mathbb{R}^d$ est un vecteur constant. Nous pouvons donc écrire directement

$$\int_\tau \nabla\varphi_j^\tau \cdot \nabla\varphi_k^\tau d\mathbf{x} = \nabla\varphi_j^\tau \cdot \nabla\varphi_k^\tau \int_\tau d\mathbf{x} = \frac{\mathbf{n}_j^\tau \cdot \mathbf{n}_k^\tau |\tau|}{[(\mathbf{s}_j^\tau - \mathbf{s}_{j+1}^\tau) \cdot \mathbf{n}_j^\tau][(\mathbf{s}_k^\tau - \mathbf{s}_{k+1}^\tau) \cdot \mathbf{n}_k^\tau]}.$$

Cette expression peut toute entière être évaluée à partir de la seule connaissances des coordonnées des sommets \mathbf{s}_j^τ , que l'on connaît grâce au fichier de maillage. Notons que cette expression fait intervenir le volume de l'élément $|\tau|$ qui peut être calculé à partir des coordonnées des sommets avec les formules suivantes

- **d=1:** $|\tau| = |\mathbf{s}_0^\tau - \mathbf{s}_1^\tau|$.
- **d=2:** $|\tau| = |(\mathbf{s}_0^\tau - \mathbf{s}_1^\tau) \times (\mathbf{s}_1^\tau - \mathbf{s}_2^\tau)|/2$.
- **d=3:** $|\tau| = |[(\mathbf{s}_0^\tau - \mathbf{s}_1^\tau) \times (\mathbf{s}_1^\tau - \mathbf{s}_2^\tau)] \cdot (\mathbf{s}_2^\tau - \mathbf{s}_3^\tau)|/6$.

Exemple Dans le cas où τ est le triangle de référence i.e. $\mathbf{s}_0^\tau = (0, 0)$, $\mathbf{s}_1^\tau = (1, 0)$ et $\mathbf{s}_2^\tau = (0, 1)$, la matrice de rigidité élémentaire $\mathbf{K}^\tau = (\mathbf{K}_{j,k}^\tau)_{j,k=0,1,2} \in \mathbb{R}^{3 \times 3}$ définie par $\mathbf{K}_{j,k}^\tau = \int_\tau \nabla\varphi_j^\tau \cdot \nabla\varphi_k^\tau d\mathbf{x}$ amdet l'expression explicite suivante

$$\mathbf{K}^\tau = \frac{1}{2} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Matrice de masse élémentaire

Pour le calcul effectif des termes de la forme $\int_\tau \varphi_j^\tau(\mathbf{x}) \varphi_k^\tau(\mathbf{x}) d\mathbf{x}$, il est possible d'avoir recours à un résultat connu sous le nom de "formule magique".

Lemma 8.1 (admis).

Soit $\tau \subset \mathbb{R}^d$ un simplexe non-dégénéré n -dimensionnel de sommets $\mathbf{s}_0^\tau, \dots, \mathbf{s}_n^\tau$, et soient $\varphi_0^\tau, \dots, \varphi_n^\tau$ les fonctions de forme locales correspondantes. Quels que soient les entiers $\alpha_0, \dots, \alpha_n \geq 0$ on a

$$\int_\tau \varphi_0^\tau(\mathbf{x})^{\alpha_0} \dots \varphi_n^\tau(\mathbf{x})^{\alpha_n} d\mathbf{x} = \frac{n! \alpha_0! \dots \alpha_n! |\tau|}{(n + \alpha_0 + \dots + \alpha_n)!}.$$

Exemple Comme application directe, considérons un élément bi-dimensionnel i.e. un triangle $\tau \subset \mathbb{R}^2$, quel qu'il soit. Alors la matrice de masse élémentaire $M^\tau = (M_{j,k}^\tau)_{j,k=0,1,2} \in \mathbb{R}^{3 \times 3}$ définie par $M_{j,k}^\tau = \int_\tau \varphi_j^\tau \varphi_k^\tau d\mathbf{x}$ dépend uniquement de $|\tau|$ et est donnée par

$$M^\tau = \frac{|\tau|}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

8.7 Quadrature sur un simplexe dans le cas général

On peut avoir à calculer des intégrales plus compliquées que les intégrales élémentaires considérées dans les paragraphes précédents. Étant donné un n -simplexe $\tau \subset \mathbb{R}^d$, on se pose alors la question du calcul de l'intégrale $\int_\tau g(\mathbf{x}) d\mathbf{x} = ?$ sans rien savoir sur la fonction $g(\mathbf{x})$ a priori, si ce n'est qu'il s'agit d'une fonction régulière. Pour réaliser un tel calcul, on a recours à une règle de quadrature qui permet d'évaluer l'intégrale non pas exactement, mais de manière approchée. Une règle de quadrature prend la forme

$$\int_\tau g(\mathbf{x}) d\mathbf{x} \simeq \sum_{q=1}^Q g(\mathbf{x}_q) \omega_q.$$

où les $\mathbf{x}_q \in \mathbb{R}^2$ (resp. $\omega_q \in \mathbb{R}$) sont les points (resp. poids) de la règle de quadrature qui dépendent du domaine d'intégration i.e. le triangle τ . En utilisant une règle de quadrature on commet une erreur, mais cette erreur est d'autant plus petite que la triangulation est fine.

On donne ci-dessous quelques exemples de formules de quadrature pour les trois cas de figure τ = tétraèdre, triangle ou segment. On utilise là encore la notation circulaire pour les indices des sommets des éléments. Ces formules sont intrinsèques (i.e. indépendantes d'un quelconque système de coordonnées) et elle s'appliquent même lorsque $n < d$.

Quadrature sur un tétraèdre

Dans ce cas τ est un 3-simplexe i.e. un tétraèdre qui admet donc quatre sommets $\mathbf{s}_0^\tau, \mathbf{s}_1^\tau, \mathbf{s}_2^\tau, \mathbf{s}_3^\tau$. Les règles de quadrature sur un tétraèdre interviennent dans le calcul des termes de volume d'un problème tri-dimensionnel ($n = 3$ et $d = 3$).

$$\int_\tau g(\mathbf{x}) d\mathbf{x} \simeq g\left(\frac{\mathbf{s}_0^\tau + \mathbf{s}_1^\tau + \mathbf{s}_2^\tau + \mathbf{s}_3^\tau}{4}\right) |\tau| \quad (8.11a)$$

$$\int_\tau g(\mathbf{x}) d\mathbf{x} \simeq \sum_{j=0,\dots,3} g(\alpha \mathbf{s}_j^\tau + \alpha \mathbf{s}_{j+1}^\tau + \alpha \mathbf{s}_{j+2}^\tau + (1-3\alpha) \mathbf{s}_{j+3}^\tau) \frac{|\tau|}{4} \quad (8.11b)$$

$$\text{avec } \alpha = \frac{5 - \sqrt{5}}{20}.$$

Quadrature sur un triangle

Dans ce cas τ est un 2-simplexe i.e. un triangle qui admet donc trois sommets $\mathbf{s}_0^\tau, \mathbf{s}_1^\tau, \mathbf{s}_2^\tau$. Les règles de quadrature sur un triangle peuvent intervenir par exemple quand on cherche à

calculer des termes de volume dans un problème bi-dimensionnel ($n = 2$ et $d = 2$), ou bien quand on cherche à calculer des termes de bord dans un problème tri-dimensionnel ($n = 2$ et $d = 3$).

$$\int_{\tau} g(\mathbf{x}) d\mathbf{x} \simeq g\left(\frac{\mathbf{s}_0^{\tau} + \mathbf{s}_1^{\tau} + \mathbf{s}_2^{\tau}}{3}\right) |\tau| \quad (8.12a)$$

$$\int_{\tau} g(\mathbf{x}) d\mathbf{x} \simeq \left(g\left(\frac{\mathbf{s}_0^{\tau} + \mathbf{s}_1^{\tau}}{2}\right) + g\left(\frac{\mathbf{s}_1^{\tau} + \mathbf{s}_2^{\tau}}{2}\right) + g\left(\frac{\mathbf{s}_2^{\tau} + \mathbf{s}_0^{\tau}}{2}\right) \right) \frac{|\tau|}{3} \quad (8.12b)$$

$$\int_{\tau} g(\mathbf{x}) d\mathbf{x} \simeq \left(27 g\left(\frac{\mathbf{s}_0^{\tau} + \mathbf{s}_1^{\tau} + \mathbf{s}_2^{\tau}}{3}\right) + \sum_{j=0,1,2} 3 g(\mathbf{s}_j^{\tau}) + 8 g\left(\frac{\mathbf{s}_{j+1}^{\tau} + \mathbf{s}_{j+2}^{\tau}}{2}\right) \right) \frac{|\tau|}{60} \quad (8.12c)$$

Quadrature sur une arête

Dans ce cas τ est un 1-simplexe i.e. un segment qui admet donc deux sommets \mathbf{s}_0^{τ} et \mathbf{s}_1^{τ} . Les règles de quadrature sur un segment peuvent intervenir quand on cherche à calculer des termes de volume dans un problème uni-dimensionnel ($n = 1$ et $d = 1$), ou bien quand on cherche à calculer des termes de bord dans un problème bi-dimensionnel ($n = 1$ et $d = 2$).

$$\int_{\tau} g(\mathbf{x}) d\sigma(\mathbf{x}) \simeq g\left(\frac{\mathbf{s}_0^{\tau} + \mathbf{s}_1^{\tau}}{2}\right) |\tau| \quad (8.13a)$$

$$\int_{\tau} g(\mathbf{x}) d\sigma(\mathbf{x}) \simeq \left(g(\mathbf{s}_0^{\tau}) + 4g\left(\frac{\mathbf{s}_0^{\tau} + \mathbf{s}_1^{\tau}}{2}\right) + g(\mathbf{s}_1^{\tau}) \right) \frac{|\tau|}{6}. \quad (8.13b)$$

8.8 Validation d'un code élément finis

Dans le cas d'un code élément finis \mathbb{P}_1 -Lagrange, il y a certains tests assez simple qu'il est possible de réaliser afin de mettre en évidence la présence éventuelle de bugs. Si le code passe ces tests, cela ne signifie pas nécessairement qu'il est dépourvu de bug. Par contre, un échec lors de ces tests implique forcément la présence d'un bug, même si un tel bug peut éventuellement être localisé dans la mise en oeuvre du test lui-même.

Test sur la rigidité

Commençons par observer que, si $u \in \mathbb{P}_1(\mathbb{R}^d)$, alors on a exactement $u(\mathbf{x}) = \sum_{j=1}^{nv} u(\mathbf{s}_j) \varphi_j(\mathbf{x})$. Considérons deux fonctions $u, v \in \mathbb{P}_1(\mathbb{R}^2)$ définies par $u(\mathbf{x}) = \boldsymbol{\alpha}_u \cdot \mathbf{x} + \beta_u$ et $v(\mathbf{x}) = \boldsymbol{\alpha}_v \cdot \mathbf{x} + \beta_v$. Notons $\mathbf{U} := (u(\mathbf{s}_j))_{j=1 \dots nv}$ et $\mathbf{V} := (v(\mathbf{s}_j))_{j=1 \dots nv}$, et pour la matrice de rigidité posons $\mathbf{K} = (\mathbf{K}_{j,k})_{j,k=1 \dots nv}$ avec $\mathbf{K}_{j,k} = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_k d\mathbf{x}$. Alors on a

$$\begin{aligned} \boldsymbol{\alpha}_u \cdot \boldsymbol{\alpha}_v |\Omega| &= \int_{\Omega} \nabla u \cdot \nabla v d\mathbf{x} = \int_{\Omega} \nabla \left(\sum_{k=1}^{nv} u(\mathbf{s}_k) \varphi_k \right) \cdot \nabla \left(\sum_{j=1}^{nv} v(\mathbf{s}_j) \varphi_j \right) d\mathbf{x} \\ &= \sum_{k=1}^{nv} \sum_{j=1}^{nv} v(\mathbf{s}_j) u(\mathbf{s}_k) \int_{\Omega} \nabla \varphi_k \cdot \nabla \varphi_j d\mathbf{x} \\ &= \sum_{k=1}^{nv} \sum_{j=1}^{nv} \mathbf{K}_{j,k} v(\mathbf{s}_j) u(\mathbf{s}_k) = \mathbf{V}^{\top} \cdot \mathbf{K} \cdot \mathbf{U} \end{aligned}$$

Par exemple, si on prend $\alpha_u = \mathbf{e}_1$ et $\alpha_v = \mathbf{e}_2$ et β_u, β_v arbitrairement, on doit avoir $\mathbf{V}^\top \cdot \mathbf{K} \cdot \mathbf{U} = 0$. Si on a $\alpha_u = \alpha_v = \mathbf{e}_1$ alors $\mathbf{V}^\top \cdot \mathbf{K} \cdot \mathbf{U} = |\Omega|$. Enfin, avec le même type de calcul, on voit que

$$\mathbf{K} \cdot \mathbf{U} = 0 \quad \text{si } \mathbf{U}^\top = (1, \dots, 1).$$

Test sur la masse

Notons $\mathbf{M} = (M_{j,k})_{j,k=1\dots nv}$ la matrice de masse définie par $M_{j,k} := \int_{\Omega} \varphi_j \varphi_k d\mathbf{x}$. Alors avec un calcul analogue à ce qui précède on voit que

$$\mathbf{U}^\top \cdot \mathbf{M} \cdot \mathbf{U} = |\Omega| \quad \text{pour } \mathbf{U}^\top = (1, \dots, 1).$$