Approximation des EDPs

ENSMM, semestre vert

(Notes du cours et des travaux dirigés du printemps)

Enseignant : Nathaël Alibaud[1]

Version de juin 2017

1. Enseignant du semestre d'automne : Mohamed Rachid Laydi

Table des matières

Préface

Ce polycopié concerne l'approximation des équations aux dérivées partielles (EDPs). Il est constitué des notes du cours et des travaux dirigés du semestre vert de printemps de l'Ecole Nationale Supérieure de Mécanique et des Microtechniques (ENSMM). On se concentre uniquement sur la méthode des éléments finis (MEF), qui est la méthode d'approximation la plus importante pour la mécanique. Par souci de simplicité, on considère seulement quelques exemples représentatifs de problèmes elliptiques linéaires (donc stationnaires). La partie théorique de la MEF est écrite dans un formalisme mathématique simplifié, accessible à des étudiants ingénieurs de deuxième année. Les prérequis sont les notions élémentaires sur l'algèbre linéaire, les polynômes, ainsi que le calcul différentiel et intégral. L'objectif du cours est la mise en oeuvre de la méthode, avec pour finalité l'écriture d'un code entier par les étudiants. Le langage de programmation choisi est celui de matlab, pour sa simplicité d'utilisation et son intérêt dans le monde industriel.

Le contenu de ce polycopié est organisé de la manière suivante : le premier chapitre est constitué des notes du cours d'amphithéâtre, avec une première partie sur les problèmes 1-d et une autre sur les problèmes 2-d ; le deuxième et dernier chapitre présente les travaux dirigés corrigés, sur un problème de poutre 1-d et un autre de barrage 2-d. Ces problèmes sont inspirés de ceux que les étudiants rencontrent en calcul des structures. Les travaux dirigés sont divisés en deux catégories, selon qu'ils soient concernés par les aspects théoriques ou l'implémentation sur machine. Les énoncés des travaux dirigés sur machine sont intitulés « TP », bien qu'ils soient enseignés dans le cadre de TD avec plus d'une vingtaine d'étudiants. Une introduction à matlab est donnée lors de la première séance sur machine.

Commentaires techniques

Les fonctions d'interpolation seront calculées directement par ordinateur, à l'aide de la matrice de passage entre la base canonique et la base d'interpolation. Ces calculs seront faits pour chaque élément, sans utiliser de changement de variable pour se ramener à un élément de référence. L'intérêt est surtout pédagogique. Celà permet d'éviter des calculs techniques qui pourraient compliquer l'exposé théorique et la programmation. L'inconvénient est que le programme devra inverser plusieurs petites matrices (bien que celà ne change pas la complexité asymptotique). D'une manière générale, nous ne réfléchirons pas aux problèmes de mémoire et de coût des calculs. Par exemple, nous ne discuterons pas des différentes méthodes de résolution de systèmes linéaires que nous pourrions utiliser pour améliorer l'efficacité du programme. Enfin, en ce qui concerne les maillages, les étudiants devront écrire un code en 1-d, puis en 2-d en faisant appel à la fonction `delaunay.m` de matlab.

Commentaires bibliographiques

Pour en savoir plus sur la MEF, voir [1, 2, 3, 4]. L'ouvrage [3] est le support officiel du cours (donné à l'ENSMM). Il a été écrit par M. R. Laydi qui est un spécialiste de la MEF et enseigne l'approximation des EDPs durant le semestre d'automne. Pour le semestre de printemps, les étudiants sont invités à lire avant tout le présent manuscrit, qui est une synthèse de tout ce qu'ils étudieront en cours et en travaux dirigés.

Logiciels

Mentionnons enfin que depuis quelques années, de plus en plus de logiciels sur la MEF sont développés. Ils permettent de traiter un grand nombre de problèmes. Selon les besoins, il est aussi possible de les combiner avec nos propres codes, par exemple en s'en servant pour générer des maillages, etc. Voici quelques références utiles : pour le calcul scientifique en général, *Octave* est une bonne alternative à matlab, car son langage est quasiment identique et il est libre ; pour la MEF, *FreeFem* est également libre et trés complet. Pour plus d'information, consultez les sites

`https://www.gnu.org/software/octave/` et `http://www.freefem.org`.

Numérotation des équations

Les références de chaque sous-section sont indépendantes, chacune étant soit une partie entière du cours soit un TD entier. Ainsi, il se peut qu'il y ait des équations de plusieurs sous-sections portant le même numéro ; à chaque fois, le lecteur doit se référer à celle de la sous-section concernée.

Chapitre 1

Cours

1.1 Un problème elliptique modèle 1-d

(Tournez la page.)

Quelques exemples d'applications

de la méthode

des éléments finis

Première partie : dimension 1

I) Un problème aux limites modèle

Etant donné un domaine $\Omega =]0,1[$, on considère le problème de trouver une fonction

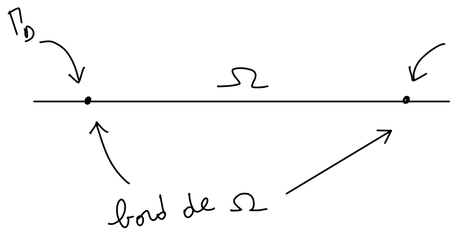
$$u : \overline{\Omega} = [0,1] \to \mathbb{R}$$

telle que

$$(\mathcal{P}) \begin{cases} -u''(x) = f(x) & \forall x \in \Omega, & \text{(équation)} \\ u(0) = \alpha, & & \text{(Dirichlet)} \\ u'(1) = \beta, & & \text{(Neuman)} \end{cases}$$

où $\alpha, \beta \in \mathbb{R}$ et $f : \Omega \to \mathbb{R}$ sont donnés.

On notera $\Gamma_D = \{0\}$ la partie du bord où on a une condition de Dirichlet et $\Gamma_N = \{1\}$ celle où on a Neuman :



II) Formulation variationnelle

On voudrait reformuler le problème à l'aide de fonctions test (travaux virtuels, théorie des distributions).

1) Calcul formel

<u>Idée générale</u> : faire des intégrations par partie (IPP) jusqu'à ce que l'on ait autant de dérivées sur la solution que sur la fonction test.

On considère des fonctions test $\phi : \overline{\Omega} \to \mathbb{R}$

telles que $\boxed{\phi = 0 \text{ sur } \Gamma_D.}$ Ce choix simplifiera

les IPPs, mais en conséquence on devra garder la

condition « $u = \alpha$ sur Γ_D » dans la définition qui suivra.

On multiplie l'équation par ϕ puis on intègre sur Ω :

$$-\int_\Omega u''(x)\,\phi(x)\,dx = \int_\Omega f(x)\,\phi(x)\,dx.$$

On fait une IPP :

$$\begin{aligned}
-\int_\Omega u''(x)\,\phi(x)\,dx &= -\int_0^1 u''(x)\,\phi(x)\,dx \\
&= \int_0^1 u'(x)\,\phi'(x)\,dx \\
&\quad + \underbrace{\cancel{u'(0)\,\phi(0)}}_{=0 \text{ car } \phi(0)=0} - \overbrace{u'(1)}^{=\beta \text{ par Neuman}}\phi(1) \\
&= \int_\Omega u'(x)\,\phi'(x)\,dx - \beta\,\phi(1)
\end{aligned}$$

D'où

$$\boxed{\underbrace{\int_\Omega u'(x)\,\phi'(x)\,dx}_{\text{noté } a(u,\phi)} = \underbrace{\int_\Omega f(x)\,\phi(x)\,dx + \beta\,\phi(1)}_{\text{noté } \ell(\phi)}.}$$

2) Définition et principaux résultats

On introduit l'ensemble

$$V = \{ \text{fonctions } u : \overline{\Omega} \longrightarrow \mathbb{R} \}$$

muni de sa structure naturelle d'espace vectoriel.

Définition

On dit que u est solution variationnelle de $(\mathcal{P})$ si

$$\boxed{\begin{cases} u \in V, \\ a(u, \phi) = \ell(\phi) \quad \forall \phi \in V \text{ nulle sur } \Gamma_D, \\ u = \alpha \quad \text{sur } \Gamma_D. \end{cases}}$$

Remarques

Les formes $(u, \phi) \longmapsto a(u, \phi)$ et $\phi \longmapsto \ell(\phi)$ définies précédemment sont respectivement bilinéaire et linéaire.

Théorème (admis) Il existe toujours une unique solution variationnelle à $(\mathcal{P})$ (qui coïncide avec la solution classique de $(\mathcal{P})$ lorsque cette dernière existe aussi).

III) Approximation par éléments finis (EFs)

Considérons maintenant l'approximation de $(\mathcal{P})$ par la méthode des éléments finis. Pour simplifier, on se concentre sur les éléments finis les plus faciles : ceux de Lagrange $\mathbb{P}_1$.

1) Méthode de Galerkin

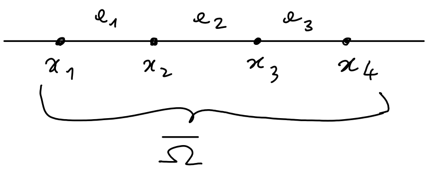
C'est une méthode générale d'approximation dont les EFs sont un cas particulier. Elle consiste à « approcher l'espace de fonctions V dans lequel on cherche la solution ».

Considérons donc un sous espace vectoriel $V_h \subset V$ de dimension finie et cherchons une solution approchée u_h telle que

$$\boxed{(\mathcal{P}_h)\begin{cases} u_h \in V_h, \\ a(u_h, \phi_h) = \ell(\phi_h) \quad \forall \phi_h \in V_h \text{ nulle sur } \Gamma_D, \\ u_h = \alpha \text{ sur } \Gamma_D. \end{cases}}$$

2) Choix de V_h (Lagrange $\mathbb{P}_1$)

On découpe $\overline{\Omega}$ en plusieurs intervalles, appellés « mailles » ou « éléments » :



(Les x_i s'appellent les « nœuds » du maillage.)

On notera

$$\mathcal{T}_h = \{e_1, e_2, e_3\}$$

l'ensemble des mailles. On choisit alors

$$\boxed{V_h = \left\{ u_h \in C(\overline{\Omega}) \;\middle|\; u_h\big|_e \in \mathbb{P}_1 \quad \forall e \in \mathcal{T}_h \right\}}$$

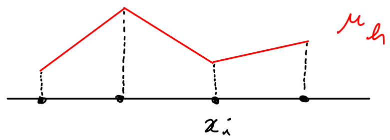
où :

$$\begin{cases} C(\overline{\Omega}) = \{\text{fonctions continues de } \overline{\Omega} \text{ dans } \mathbb{R}\}, \\ \mathbb{P}_n = \{\text{polynômes de degré} \leq n\} = \text{vect}\{1, x, \ldots, x^n\}, \\ u_h\big|_e \text{ est la restriction de } u_h \text{ à la maille } e. \end{cases}$$

(Rappel : $u_h|_e$ est donc la fonction $x \in e \mapsto u_h(x)$.)

L'espace V_h (respectivement $\mathcal{P}_1$) s'appelle « l'espace d'approximation ou d'interpolation global (resp. local) ».

On aura donc des fonctions de la forme suivante :



Les différentes étapes pour résoudre $(\mathcal{P}_h)$ sont les suivantes :

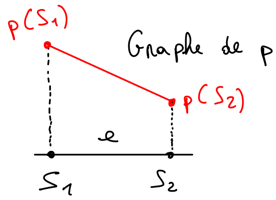
* on choisit les inconnues du problème, appelées aussi « degrés de liberté (ddls) » ;
* on calcule la base associée à ces ddls ;
* on écrit le problème sous forme matricielle ;
* on résoud le système obtenu par ordinateur.

3) Choix des ddls

$\boxed{\text{Sur une maille}}$ Notons $e = [S_1, S_2]$ et $p = u_h|_e$.

(C'est une droite !)

On a $p \in \mathrm{vect}\{1, x\}$.



Donc p est entièrement déterminé par la donnée de deux réels, par exemple $\underbrace{p(S_1) \text{ et } p(S_2)}_{\text{Ce sont les ddls locaux}}$.

$\boxed{\text{Sur le domaine}}$

Par suite, u_h est entièrement déterminé par la donnée des réels $\underbrace{u_h(x_1), \dots, u_h(x_4)}_{\text{Ce sont les ddls globaux}}$.

4) Base d'interpolation

Notons plus généralement $\mathcal{P}_e$ l'espace d'interpolation local.

Les ddls locaux sont donc des « fonctions coordonnées » qui à $p \in \mathcal{P}_e$ associent « ses coordonnées $p(S_1)$ et $p(S_2)$ ».

On obtient des formes linéaires notées L_k^e.

Ici, on a deux ddls locaux :

$$\boxed{\begin{aligned} L_k^e : \ \mathcal{P}_e &\longrightarrow \mathbb{R} \\ p &\longmapsto p(S_k) \end{aligned}}$$

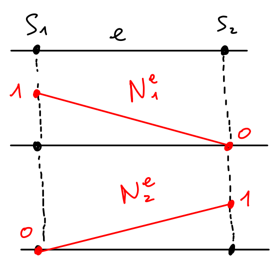
(pour $k = 1, 2$).

Base locale

Notons $\{N_1^e, N_2^e\}$ la base de $\mathcal{P}_e$ associée à ces fonctions coordonnées. Elle est donnée par la formule suivante :

$$\boxed{L_k^e(N_m^e) = \delta_{km} = \begin{cases} 1 & \text{si } k = m \\ 0 & \text{si } k \neq m \end{cases} \qquad \forall k, m \in \{1, 2\}.}$$

Calculons N_1^e. On a



$$\begin{cases} N_1^e \in \mathcal{P}_e = \mathcal{P}_1 = \text{vect}\{1, x\}, \\ L_1^e(N_1^e) = 1 = N_1^e(S_1), \\ L_2^e(N_1^e) = 0 = N_1^e(S_2). \end{cases}$$

Donc $\boxed{N_1^e(x) = \dfrac{x - S_2}{S_1 - S_2}.}$ De même $\boxed{N_2^e(x) = \dfrac{x - S_1}{S_2 - S_1}.}$

Exercice

Montrez que $\{N_1^e, N_2^e\}$ est bien une base de $\mathcal{P}_e$ et

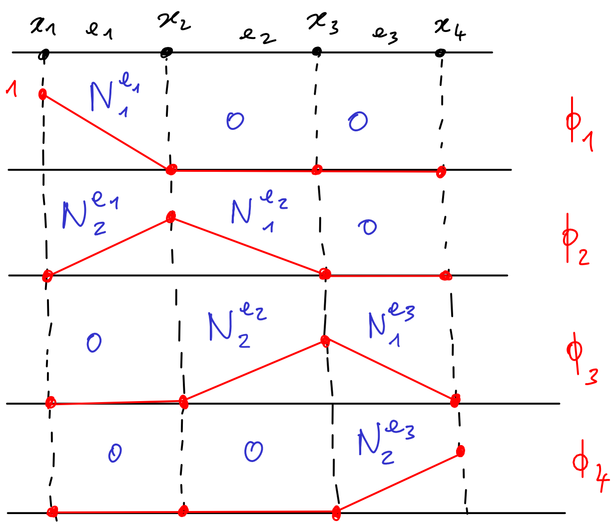
que pour tout $p \in \mathcal{P}_e$, on a

$$\boxed{p = \sum_{k=1}^{2} L_k^e(p)\, N_k^e .}$$

Base globale

On construit ensuite une base de V_h en recollant les N_k^e

comme ci-dessous :



Exercice

1) Montrez que $\{\phi_1, \dots, \phi_4\}$ est bien une base de V_h.

2) Montrez que

$$\boxed{L_i(\phi_j) = \delta_{ij} \qquad \forall i,j \in \{1, \dots, 4\}}$$

où on note

$$\boxed{\begin{aligned} L_i : V_h &\longrightarrow \mathbb{R} \\ u_h &\longmapsto u_h(x_i) \end{aligned}}$$

les ddls globaux.

3) Montrez que pour tout $u_h \in V_h$, on a

$$\boxed{u_h = \sum_{i=1}^{4} L_i(u_h)\, \phi_i \,.}$$

5) Ecriture matricielle

Etant donné $u_h \in V_h$, on note

$$\boxed{U_h = \begin{pmatrix} L_1(u_h) \\ \vdots \\ L_4(u_h) \end{pmatrix} \in \mathbb{R}^4}$$

ses coordonnées sur la base des ϕ_i.

Théorème

u_h est solution du problème approché $(\mathcal{P}_h)$

si et seulement si

$$\boxed{AU_h = F}$$

où

$$\boxed{\begin{array}{l} A_{ij} = \begin{cases} a(\phi_j, \phi_i) & \text{si } i \notin I_D, \\ \delta_{ij} & \text{si } i \in I_D, \end{cases} \\ \\ F_i = \begin{cases} \ell(\phi_i) & \text{si } i \notin I_D, \\ \alpha & \text{si } i \in I_D, \end{cases} \end{array}}$$

avec

$$\boxed{I_D = \left\{ i \;\middle|\; \phi_i \not\equiv 0 \text{ sur } \Gamma_D \right\}.}$$

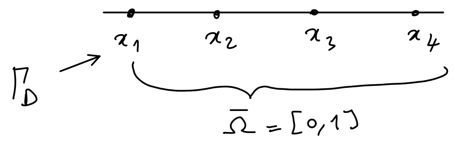
(n'est pas identiquement nulle)

Conclusion

Il reste à calculer les coefficients de A et de F, puis à résoudre le système $AU_h = F$ (voir les TDs et TPs).

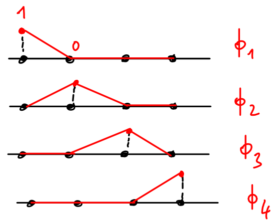
Preuve (du théorème)

On rappelle qu'on a choisi le maillage suivant :



ddls : $L_i(u_h) = u_h(x_i)$

base :



On a donc

$$\left[u_h = \alpha \text{ sur } \Gamma_D \right] \iff u_h(0) = \alpha$$

$$\iff u_h(x_1) = \alpha$$

$$\iff L_1(u_h) = \alpha$$

$$\iff \boxed{L_i(u_h) = \alpha \quad \forall i \in I_D = \{1\}.} \quad (*)$$

De plus, notons

$$V_h^0 = \{ \phi_h \in V_h \mid \phi_h = 0 \text{ sur } \Gamma_D \}.$$

Exercice

C'est un sous espace vectoriel de V_h ayant pour base

la famille $\{\phi_2, \phi_3, \phi_4\} = \{\phi_i \mid i \notin I_D\}$.

On en déduit que :

$$\left[a(u_h, \phi_h) = \ell(\phi_h) \quad \forall \phi_h \in V_h \text{ nulle sur } \Gamma_D\right]$$

$$\Updownarrow$$

$$\left[a(u_h, \phi_h) = \ell(\phi_h) \quad \forall \phi_h \in V_h^0\right]$$

$$\Updownarrow \longleftarrow \begin{cases} \text{car :} \\ a \text{ est bilinéaire,} \\ \ell \text{ est linéaire,} \\ \text{et } \{\phi_i \mid i \notin I_D\} \text{ engendre } V_h^0 \end{cases}$$

$$\left[a(\underbrace{u_h}_{\sum_{j=1}^{4} L_j(u_h)\phi_j}, \phi_i) = \ell(\phi_i) \quad \forall i \notin I_D\right]$$

$$\Updownarrow \leftarrow \text{bilinéarité de } a$$

$$(**) \quad \boxed{\sum_{j=1}^{4} a(\phi_j, \phi_i)\, L_j(u_h) = \ell(\phi_i) \quad \forall i \notin I_D \quad .}$$

Maintenant, il est facile de voir que

$$[(*) \text{ et } (**)] \Longrightarrow AU_h = F. \qquad \blacksquare$$

(fin de preuve)

Remarque

Pour approcher notre problème modèle, nous avons choisi les EFs de Lagrange $\mathbb{P}_1$, mais il est possible de prendre d'autres EFs. Un inventaire des principaux EFs utilisés en dimension 1 est donné en appendice.

Annexe : Notion d'éléments finis

Définition

On dit que $(e, \mathcal{P}_e, \Sigma_e)$ est un élément fini si

i) e est un ensemble non vide,

ii) $\mathcal{P}_e$ est un espace vectoriel de fonctions sur e de dimension finie d,

iii) $\Sigma_e = \{L_1^e, \dots, L_d^e\}$ est une famille constituée de d formes linéaires distinctes de $\mathcal{P}_e$ dans $\mathbb{R}$,

iv) (Unisolvance)

$$(\forall p \in \mathcal{P}_e) \quad \left[L_1^e(p) = \dots = L_d^e(p) = 0 \implies p = 0 \right].$$

Remarque

On demande toujours à avoir $\dim \mathcal{P}_e = \operatorname{Card} \Sigma_e$.

(Ce qui est naturel car les ddls $\{L_1^e, \dots, L_d^e\}$ formeront un système de coordonnées sur $\mathcal{P}_e$.)

A) Base d'interpolation

Proposition

Soit $(e, \mathcal{P}_e, \Sigma_e)$ un EFs. Alors :

i) Pour tout $m = 1, \dots, d$, il existe un unique $N_m^e \in \mathcal{P}_e$ solution du système d'équations :

$$\boxed{L_k^e(N_m^e) = \delta_{km} \qquad \forall k = 1, \dots, d.}$$

ii) La famille $\{N_1^e, \dots, N_d^e\}$ ainsi obtenue est une base de $\mathcal{P}_e$ telle que

$$\boxed{p = \sum_{k=1}^{d} L_k^e(p)\, N_k^e \qquad \forall p \in \mathcal{P}_e.}$$

Preuve (similaire à celle de l'exercice 1 du TD no. 1)

Posons $T : p \in \mathcal{P}_e \longmapsto \begin{pmatrix} L_1^e(p) \\ \vdots \\ L_d^e(p) \end{pmatrix} \in \mathbb{R}^d$.

Alors T est linéaire car les L_k^e le sont.

Elle est de plus injective, car si $T(p) = T(q)$ alors $T(p-q) = 0$, ce qui veut dire que

$$L_1^e(p-q) = \dots = L_d^e(p-q) = 0$$

et on en déduit que $p - q = 0$ par unisolvance.

Mais, on sait de plus que

$$\dim \mathcal{P}_e = d = \dim \mathbb{R}^d$$

et le théorème de la dimension implique que T est bijective. Il suffit alors de prendre

$$N_k^e = T^{-1}\left(\begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}\right) \leftarrow k^{\text{ème}} \text{ place}$$

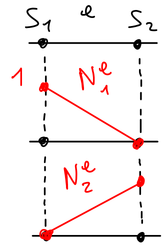
pour conclure la preuve. ∎

B) Principaux éléments finis en dimension 1

Dans la suite, $e = [S_1, S_2]$, $\mathcal{P}_e$ est un espace de polynômes, p un polynôme général de $\mathcal{P}_e$, et on montre seulement l'unisolvance.

a) Lagrange $\mathcal{P}_1$

$\mathcal{P}_e = \mathcal{P}_1$ (espace d'interpolation)



(base)

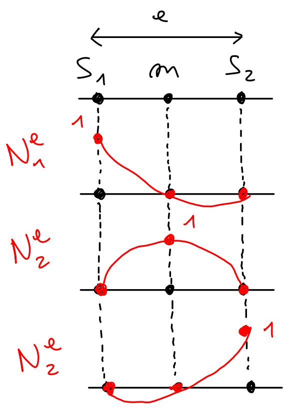
$$\begin{cases} L_1^e(p) = p(S_1) \\ L_2^e(p) = p(S_2) \end{cases}$$

(ddls)

(Voir le cours précédent pour la preuve de l'unisolvance et le calcul des N_k^e.)

b) Lagrange $\mathcal{P}_2$

$\mathcal{P}_e = \mathcal{P}_2 = \text{Vect}\{1, x, x^2\}$



$m = \dfrac{S_1 + S_2}{2}$

$$\begin{cases} L_1^e(p) = p(S_1) \\ L_2^e(p) = p(m) \\ L_3^e(p) = p(S_2) \end{cases}$$

$\boxed{\text{Base : } L_k^e(N_m^e) = \delta_{km}}$

(⚠ Ce sont des paraboles !)

Preuve de l'unisolvance

Si $p \in \mathcal{P}_2$ est tel que $L_1^e(p) = L_2^e(p) = L_3^e(p) = 0$,

alors $p(S_1) = p(m) = p(S_2) = 0$. Ce polynôme de degré

au plus 2 a donc 3 racines distinctes, ce qui implique

que c'est nécessairement le polynôme nul. $\blacksquare$

Calcul de la base

Calculons N_1^e. On a

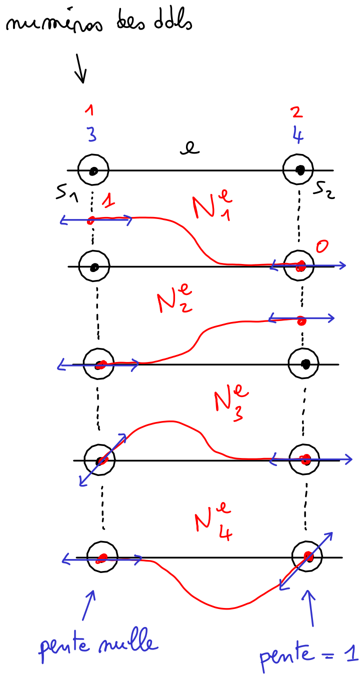
$$\begin{cases} N_1^e \in \mathcal{P}_2, \\ L_1^e(N_1^e) = 1 = N_1^e(S_1), \\ L_2^e(N_1^e) = 0 = N_1^e(m), \\ L_3^e(N_1^e) = 0 = N_1^e(S_2). \end{cases}$$

racines

Donc $$\boxed{N_1^e(x) = \frac{(x-m)(x-S_2)}{(S_1-m)(S_1-S_2)}.}$$

De même $$\boxed{N_2^e(x) = \frac{(x-S_1)(x-S_2)}{(m-S_1)(m-S_2)}, \quad \text{etc.}}$$

c) Hermite $\mathcal{P}_3$



$$\mathcal{P}_e = \mathcal{P}_3 = \text{vect}\{1, x, x^2, x^3\}$$

$$\left.\begin{aligned} L_1^e(p) &= p(S_1) \\ L_2^e(p) &= p(S_2) \end{aligned}\right\} \text{ ddls « valeurs aux noeuds »}$$

$$\left.\begin{aligned} L_3^e(p) &= p'(S_1) \\ L_4^e(p) &= p'(S_2) \end{aligned}\right\} \text{ ddls « dérivées aux noeuds »}$$

$$\boxed{\text{Base : } L_k^e(N_m^e) = \delta_{km}.}$$

Preuve de l'unisolvance

Si $p \in \mathcal{P}_3$ est tel que $L_1^e(p) = \dots = L_4^e(p) = 0$, alors

$$p(S_1) = p'(S_1) = 0 = p(S_2) = p'(S_2).$$

Donc S_1 et S_2 sont deux racines doubles distinctes et nécessairement p est le polynôme nul (car il est de degré au plus 3). ∎

Calcul de la base

Le calcul direct, comme précédemment, est un peu plus long. Utilisons une autre méthode, qui aura l'avantage d'être générale et implémentable par ordinateur.

Notons $\{\mathbb{1}, X, X^2, X^3\}$ la base canonique de $\mathcal{P}_3$.

Ces notations signifient que ce sont des fonctions :

$$\mathbb{1} : x \mapsto 1, \quad X : x \mapsto x, \quad \text{etc}.$$

Ecrivons cette base sur la « base d'interpolation » $\{N_1^e, \dots, N_4^e\}$. Prenons par exemple X^2. On a besoin de calculer ses coordonnées qui seront données par les $L_k^e(X^2)$. On a

$$\begin{cases} L_1^e(X^2) = X^2(S_1) = S_1^2 \\ \text{et de même } L_2^e(X^2) = S_2^2. \end{cases}$$

Pour calculer les autres, on remarque que

$$(X^2)'(x) = 2x \quad \left(\text{puisque } X^2(x) = x^2\right).$$

On a donc

$$\begin{cases} L_3^e(X^2) = (X^2)'(S_1) = 2S_1 \\ \text{et} \quad L_4^e(X^2) = 2S_2. \end{cases}$$

Maintenant, on applique la formule de la proposition de la section A. On obtient :

$$\begin{aligned} X^2 &= \sum_{k=1}^{4} L_k^e(X^2)\, N_k^e \\ &= S_1^2 N_1^e + S_2^2 N_2^e + 2S_1 N_3^e + 2S_2 N_4^e. \end{aligned}$$

Cela veut dire que pour tout réel x,

$$x^2 = S_1^2 N_1^e(x) + S_2^2 N_2^e(x) + 2S_1 N_3^e(x) + 2S_2 N_4^e(x).$$

On montre de même que

$$1 = N_1^e(x) + N_2^e(x),$$

$$x = S_1 N_1^e(x) + S_2 N_2^e(x) + N_3^e(x) + N_4^e(x),$$

$$x^3 = S_1^3 N_1^e(x) + S_2^3 N_2^e(x) + 3S_1^2 N_3^e(x) + 3S_2^2 N_4^e(x).$$

D'où, pour tout $x \in \mathbb{R}$, on a

$$\boxed{\begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 \\ s_1 & s_2 & 1 & 1 \\ s_1^2 & s_2^2 & 2s_1 & 2s_2 \\ s_1^3 & s_2^3 & 2s_1^2 & 2s_2^2 \end{pmatrix}}_{\text{notée } M} \begin{pmatrix} N_1^e(x) \\ \vdots \\ N_4^e(x) \end{pmatrix}.}$$

On remarque que M est la matrice de passage entre la base canonique et la base d'interpolation. Elle est donc inversible et on obtient la formule :

$$\boxed{\begin{pmatrix} N_1^e(x) \\ \vdots \\ N_4^e(x) \end{pmatrix} = M^{-1} \begin{pmatrix} 1 \\ x \\ x^2 \\ x^3 \end{pmatrix} \qquad \forall x \in \mathbb{R}.}$$

<u>Remarque</u> : Il reste à calculer M^{-1}, ce que l'on fera directement dans le programme en TPs pour simplifier.

Remarque (Calcul des dérivées)

Puisque M ne dépends pas de x, on peut dériver et obtenir de plus que

$$\boxed{\begin{pmatrix} (N_1^e)'(x) \\ \vdots \\ (N_4^e)'(x) \end{pmatrix} = M^{-1} \begin{pmatrix} 0 \\ 1 \\ 2x \\ 3x^2 \end{pmatrix} \qquad \forall x \in \mathbb{R}.}$$

Remarque (élément de référence)

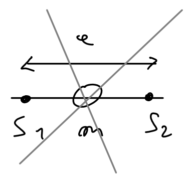
En général, on effectue les calculs précédents sur un élément de référence, par exemple $\hat{e} = [-1,1]$, auquel on se ramène par changement de variable. L'avantage est que les calculs des fonctions d'interpolation ne se font qu'une seule fois. Dans ce cours, on a choisi de faire les calculs directement sur chaque élément, car cela simplifiera les programmes des TPs. Pour avoir des détails sur les éléments de références, voir la bibliographie.

d) Un contre exemple

⚠ On ne peut pas choisir n'importe quoi comme ddls. Il est nécessaire que le triplet soit unisolvant pour obtenir un système de coordonnées sur l'espace d'interpolation.

Voici un contre exemple. Prenons $\mathcal{P}_e = \mathcal{P}_2$ et

$$\begin{cases} L_1^e(p) = p(S_1) \\ L_2^e(p) = p'(m) \\ L_3^e(p) = p(S_2). \end{cases}$$



Ce n'est pas un EF !

En effet, ce triplet n'est pas unisolvant car le polynôme

$$p(x) = (x - S_1)(x - S_2) \in \mathcal{P}_2$$

satisfait $L_1^e(p) = L_2^e(p) = L_3^e(p) = 0$, bien que ce ne soit pas le polynôme nul.

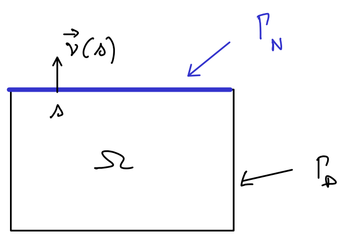
1.2 Un problème elliptique modèle 2-d

(Tournez la page.)

$\boxed{\text{Deuxième partie : dimension 2}}$

I) Un problème aux limites modèle

Soient $\Omega =]0,1[\times]0,1[$ un domaine de $\mathbb{R}^2$, $\overline{\Omega} = [0,1] \times [0,1]$ sa fermeture et $\partial\Omega = \overline{\Omega} \setminus \Omega$ son bord. Pour tout $s \in \partial\Omega$, on note $\vec{\nu}(s)$ le vecteur d'origine s qui est normal au bord, de norme euclidienne $\| \vec{\nu}(s) \| = 1$, et dirigé vers l'extérieur de Ω ; voir le dessin qui suit.



On se donne aussi une partition $\{\Gamma_D, \Gamma_N\}$ du bord comme ci-dessus ; c'est-à-dire telle que

$$\begin{cases} \Gamma_D \cup \Gamma_N = \partial\Omega, \\ \Gamma_D \cap \Gamma_N = \emptyset. \end{cases}$$

On considère alors le problème de trouver

$$u : \overline{\Omega} \longrightarrow \mathbb{R}$$

telle que

(Equation aux Dérivées Partielles)

$$\boxed{(\mathcal{P}) \begin{cases} -\Delta u = f & \text{dans } \Omega, \quad \text{(EDP)} \\ u = \alpha & \text{sur } \Gamma_D, \quad \text{(Dirichlet)} \\ \dfrac{\partial u}{\partial \nu} + b u = \beta & \text{sur } \Gamma_N, \quad \text{(Neuman)} \end{cases}}$$

où les fonctions

$$\begin{cases} f : \Omega \longrightarrow \mathbb{R}, \\ \alpha : \Gamma_D \longrightarrow \mathbb{R}, \\ \text{et } b, \beta : \Gamma_N \longrightarrow \mathbb{R} \end{cases}$$

sont supposées connues.

Notations :

$$\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}, \qquad \text{(Laplacien)}$$

$$\vec{\nabla} u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2} \right), \qquad \text{(Gradient)}$$

$$\frac{\partial u}{\partial \nu} = \vec{\nabla} u \cdot \vec{\nu}. \qquad \text{(Dérivée normale)}$$

produit scalaire

II) Formulation variationnelle

1) Formule de Green-Gauss (IPP)

Théorème (admis)

Soient $\phi : \overline{\Omega} \longrightarrow \mathbb{R}$ et $\vec{F} : \overline{\Omega} \longrightarrow \mathbb{R}^2$ « suffisamment régulières ».

Alors

$$\int_\Omega \phi \, \mathrm{div}(\vec{F}) = - \int_\Omega \vec{\nabla}\phi \cdot \vec{F} + \int_{\partial\Omega} \phi \, \vec{F} \cdot \vec{\nu}.$$

Notation

$\vec{F} = (F_i)_{1 \leq i \leq 2}$ et $\mathrm{div}(\vec{F}) = \dfrac{\partial F_1}{\partial x_1} + \dfrac{\partial F_2}{\partial x_2}$. (Divergence)

Notations des intégrales

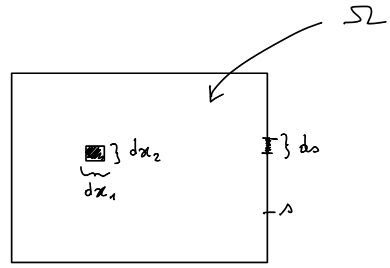
Pour tout $g : \overline{\Omega} \longrightarrow \mathbb{R}$, on note

$$\int_\Omega g = \int_\Omega g(x_1, x_2)\, dx_1\, dx_2$$

et $$\int_{\partial\Omega} g = \int_{\partial\Omega} g(s)\, ds,$$

⚠ c'est l'intégrale non orientée le long d'une courbe que l'on précisera plus tard.

où (x_1, x_2) est la variable générale de Ω, $dx_1\, dx_2$ l'élément d'aire, s la variable du bord $\partial\Omega$ et ds l'élément de longueur ; voir le dessin qui suit.



2) Calcul formel

Reformulons le problème à l'aide de fonctions test. Supposons donc que u est solution de (P) et considérons une fonction $\phi : \overline{\Omega} \longrightarrow \mathbb{R}$ arbitraire telle que $\boxed{\phi = 0 \text{ sur } \Gamma_D}$.

Alors :

$$\underbrace{-\int_\Omega \phi\,\Delta u}_{\parallel} = \int_\Omega f\phi$$

IPP avec $\vec{F} = \vec{\nabla u}$ car $\Delta u = \mathrm{div}(\vec{\nabla u})$

$$\int_\Omega \vec{\nabla u}\cdot\vec{\nabla\phi} - \underbrace{\int_{\partial\Omega} \phi\,\vec{\nabla u}\cdot\vec{\nu}}_{\parallel}$$

$$\begin{pmatrix}\text{car } \phi = 0\\ \text{sur } \Gamma_D\end{pmatrix} \cancel{\int_{\Gamma_D}} + \int_{\Gamma_N}$$

$$\parallel$$

$$\int_{\Gamma_N} \phi \frac{\partial u}{\partial \nu} \underset{(\text{Neumann})}{=} -\int_{\Gamma_N} b u\phi + \int_{\Gamma_N} \beta\phi.$$

D'où

$$\boxed{\begin{array}{c} \overbrace{\int_\Omega \vec{\nabla u}\cdot\vec{\nabla\phi} + \int_{\Gamma_N} b u \phi}^{\substack{a(u,\phi)\\ \parallel \text{ noté}}} \\ \parallel \\ \underbrace{\int_\Omega f\phi + \int_{\Gamma_N} \beta\phi.}_{\substack{\parallel \text{ noté}\\ \ell(\phi)}} \end{array}}$$

3) Définition et principaux résultats

Notons $V = \{ \text{fonctions de } \bar{\Omega} \text{ dans } \mathbb{R} \}$.

Définition

On dit que u est solution variationnelle de $(\mathcal{P})$ si

$$\begin{cases} u \in V, \\ a(u,\phi) = \ell(\phi) \quad \forall \phi \in V \text{ nulle sur } \Gamma_D, \\ u = \alpha \quad \text{sur } \Gamma_D. \end{cases}$$

(Elle coïncide encore avec la solution classique lorsque cette dernière existe.)

Théorème (admis)

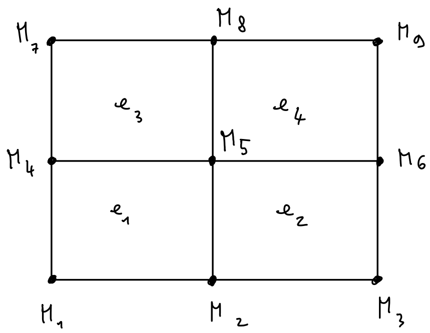
Il existe toujours une unique solution variationnelle à $(\mathcal{P})$.

III) Approximation

1) Choix des éléments finis

On choisit le maillage avec les EFs comme sur le dessin qui suit, appelés « éléments finis de Lagrange Q_1 ».

(Global)



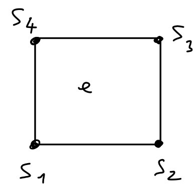
Espace d'interpolation

$$V_h = \left\{ u_h \in C(\bar{\Omega}) \text{ tel que } u_h|_e \in \mathcal{P}_e \quad \forall e = e_1, \dots, e_4 \right\}$$

ddls

$$\begin{cases} L_1(u_h) = u_h(M_1) \\ \vdots \\ L_9(u_h) = u_h(M_9) \end{cases}$$

(Local)



Espace d'interpolation

$$\mathcal{P}_e = \mathrm{Vect}\{1, x_1, x_2, x_1 x_2\} \underset{\text{noté}}{=} Q_1$$

ddls

$$\begin{cases} L_1^e(p) = p(S_1) \\ \vdots \\ L_4^e(p) = p(S_4) \end{cases}$$

Notation

On dit que $p \in \mathrm{Vect}\{1, x_1, x_2, x_1 x_2\}$ si $p : \mathbb{R}^2 \to \mathbb{R}$ est un polynôme de la forme $p(x_1, x_2) = a + b\, x_1 + c\, x_2 + d\, x_1 x_2$, pour certaines constantes $a, b, c, d \in \mathbb{R}$.

Exercice

Montrez que le triplet $(e, \mathcal{P}_e, \{L_1^e, \ldots, L_4^e\})$ est un élément fini.

Remarque

Notons $\mathcal{T}_h = \{e_1, \ldots, e_4\}$ l'ensemble des mailles. Pour que $\{L_1, \ldots, L_9\}$ forment un système de coordonnées de V_h, on choisit en général $\mathcal{T}_h$ telle que

(i) $\overline{\Omega} = \bigcup_{e \in \mathcal{T}_h} e$

(ii) et

$$e \cap \tilde{e} = \begin{cases} \emptyset, \\ \text{ou 1 sommet commun à } e \text{ et } \tilde{e}, \\ \text{ou 1 arête commune \_\_\_\_\_\_\_\_}, \end{cases}$$

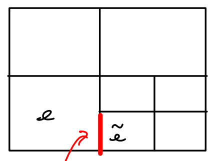
pour tout $e, \tilde{e} \in \mathcal{T}_h$ tels que $e \neq \tilde{e}$.

On dit alors que $\mathcal{T}_h$ est une « triangulation » de $\overline{\Omega}$.

Exemple : notre précédent maillage est bien une triangulation.

(et on peut en déduire rigoureusement que les L_i forment un système de coordonnées sur V_h (ce que l'on admet))

Contre - exemple



Ce n'est pas une arête commune

2) Bases d'interpolation

Soient $\{\phi_1, \dots, \phi_9\}$ et $\{N_1^e, \dots, N_4^e\}$ les bases respectives de V_h et $\mathcal{P}_e$ associées aux ddls précédents par les formules

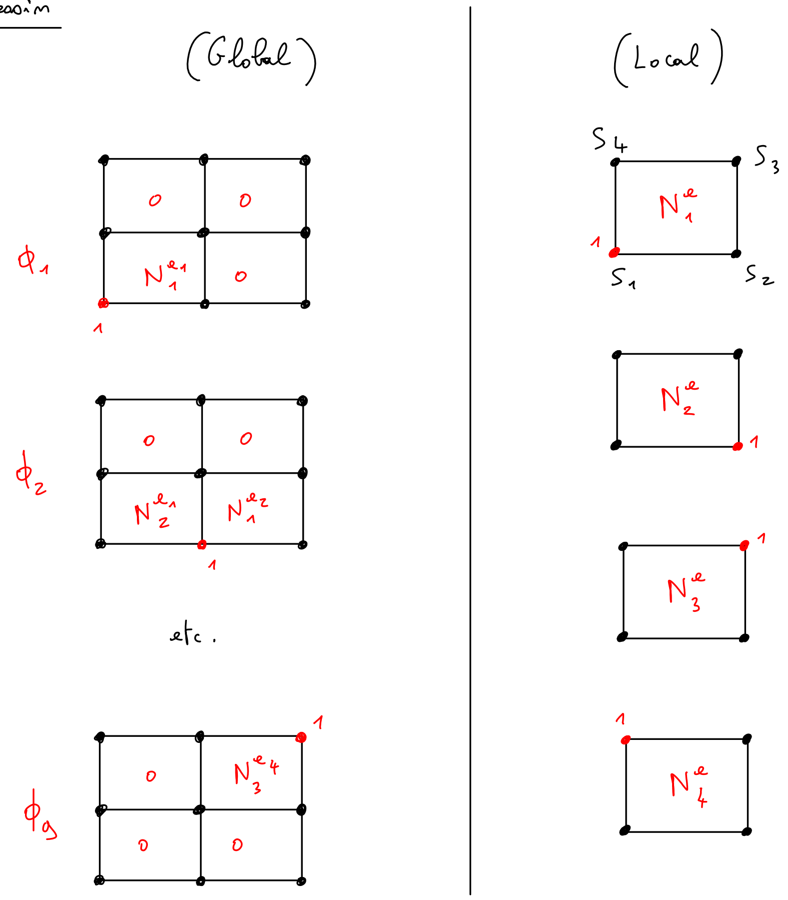
$$\boxed{\begin{aligned} L_i(\phi_j) &= \delta_{ij} \qquad \forall i,j \in \{1, \dots, 9\}, \\ L_k^e(N_m^e) &= \delta_{km} \qquad \forall k,m \in \{1, \dots, 4\}. \end{aligned}}$$

Sur le dessin ci-dessous, on précise les valeurs prises par les fonctions sur les noeuds M_i et S_k, ainsi que les correspondances entre les ϕ_i et les N_k^e.

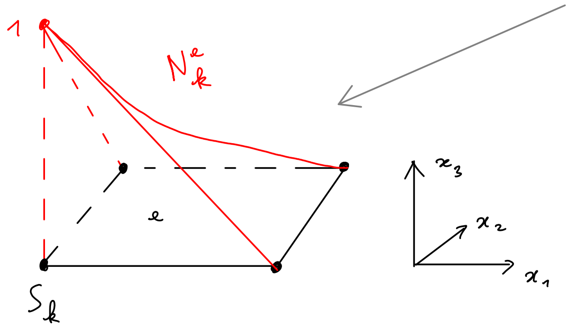
Légende

- $\bullet$ = noeud où ϕ_i (ou N_k^e) prend la valeur 0 ;
- $\overset{1}{\bullet}$ = ——————————— 1.

Dessin



Graphe de $x_3 = N_k^e(x_1, x_2)$



C'est une droite le long des axes x_1 et x_2, mais c'est une parabole le long de la droite $\{x_1 = x_2\}$.

3) Problème approché

On considère maintenant une approximation u_h de u, comme étant la solution du problème

$$\boxed{(\mathcal{P}_h) \begin{cases} u_h \in V_h, \\ a(u_h, \phi_h) = \ell(\phi_h) \quad \forall \phi_h \in V_h \text{ nulle sur } \Gamma_D, \\ u_h = \alpha_h \quad \text{sur } \Gamma_D, \end{cases}}$$

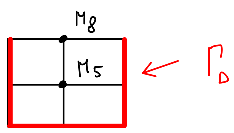
où $\boxed{\alpha_h = \sum_{i \in I_D} \alpha(M_i)\, \phi_i}$

avec $\boxed{I_D = \{ i \mid \phi_i \not\equiv 0 \text{ sur } \Gamma_D \}.}$

En général, c'est $\alpha(M_{L_i})$ où M_{L_i} est le noeud qui supporte le ddl L_i, comme au TD no.2 ; mais ici $M_{L_i} = M_i$.

Remarques

1) Ici $I_D = \{1, \ldots, 9\} \setminus \{5, 8\}$ car :



2) α_h est une approximation polynômiale par morceaux de α.

3) On doit considérer une telle condition de Dirichlet approchée car $u_h|_{\Gamma_D}$ est polynômiale par morceaux, mais α peut ne pas l'être.

Voici maintenant l'écriture matricielle de $(\mathcal{P}_h)$.

Théorème

Notons $U_h = \begin{pmatrix} L_1(u_h) \\ \vdots \\ L_g(u_h) \end{pmatrix} \in \mathbb{R}^g$ les coordonnées de $u_h \in V_h$

sur la base $\{\phi_1, \dots, \phi_g\}$. Alors u_h est solution de $(\mathcal{P}_h)$

si et seulement si

$$\boxed{AU_h = F}$$

où

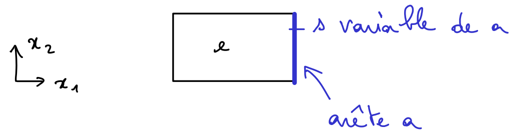
$$\boxed{\begin{array}{l} A_{ij} = \begin{cases} a(\phi_j, \phi_i) & \text{si } i \notin I_D, \\ \delta_{ij} & \text{si } i \in I_D, \end{cases} \\ \\ F_i = \begin{cases} \ell(\phi_i) & \text{si } i \notin I_D, \\ \alpha(M_i) & \text{si } i \in I_D. \end{cases} \end{array}}$$

(c'est encore $\alpha(M_{L_i})$ en général, mais ici $M_{L_i} = M_i$.)

Preuve (formelle)

Vérifions que les équations données par les lignes $i \in I_D$ sont bien équivalentes à avoir « $u_h = \alpha_h$ sur Γ_D ».

Pour celà, on utilise le fait que pour toute arête a d'une maille e, $u_h|_a$ est un polynôme de degré au plus 1 en la variable de l'arête. C'est-à-dire, avec les notations ci-dessous



la fonction $s \mapsto u_h(s)$ appartient à l'espace de polynômes $\text{Vect}\{1, s\}$, qui sera rigoureusement définie plus tard. Celà est facile à comprendre de manière intuitive. En effet, faire varier s revient à fixer l'une des deux variables, x_1 ou x_2, et faire varier l'autre toute seule. Puisque de plus

$$u_h \in \text{vect}\{1, x_1, x_2, x_1 x_2\},$$

on comprends que $u_h|_a \in \text{vect}\{1, s\}$. Par suite, $u_h|_{\Gamma_D}$ est entièrement déterminée par ses valeurs aux noeuds

$$\{M_i \mid i \in I_D\}.$$

On raisonne de la même manière pour montrer que $d_h|_{\Gamma_D}$ est aussi entièrement déterminée par ses valeurs en ces noeuds.

Donc $\left[u_h = \alpha_h \text{ sur } \Gamma_D\right] \Longleftrightarrow \left[u_h(M_i) = \alpha_h(M_i) \quad \forall i \in I_D\right]$

$$\Longleftrightarrow \left[L_i(u_h) = \alpha_h(M_i) \quad \forall i \in I_D\right],$$

ce qui nous donne l'équivalence voulue.

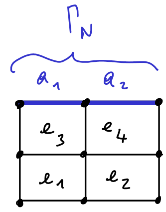
Pour conclure la preuve du théorème, on montre que les équations données par les lignes $i \notin I_D$ sont équivalentes à « $a(u_h, \phi_h) = l(\phi_h) \quad \forall \phi_h \in V_h$ nulle sur Γ_D », de la même façon que pour le théorème de la première partie sur la dimension 1. ■

IV) Assemblage

1) Préliminaires

On implémentera A et F avec le même algorithme d'assemblage qu'au TD no. 2. On devra en plus prendre en compte la condition de Neuman. Pour calculer par exemple un coefficient F_i, on procèdera ainsi :

$$F_i = \underbrace{\int_\Omega \cdots}_{\displaystyle = \sum_{\text{mailles } e} \int_e \cdots} + \underbrace{\int_{\Gamma_N} \cdots}_{\displaystyle = \sum_{\text{arêtes } a} \int_a \cdots}$$

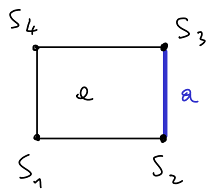


où le domaine Ω est découpé en mailles, comme tout à l'heure, et le bord Γ_D sera aussi découpé en arêtes élémentaires, comme ci-dessus. On calculera alors les coefficients élémentaires $\int_e \cdots$ et $\int_a \cdots$, puis on les assemblera pour obtenir les coefficients globaux.

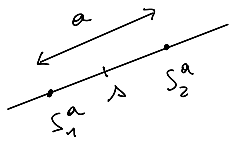
2) Eléments finis sur les arêtes

Pour l'assemblage des coefficients sur les arêtes, on a besoin de définir une structure d'éléments finis sur celles-ci. Cette structure sera induite de celle que l'on avait choisie sur les mailles.

Dans la suite, on considère une maille e et une de ses arêtes a comme sur le dessin suivant :



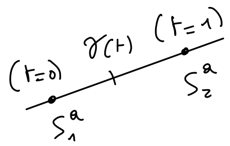
On note les nœuds de a comme ci-dessous :



On note encore s la variable générale de l'arête, c'est-à-dire un point général de a. Il est utile de paramétriser l'arête par

$$\begin{aligned} \gamma :\ [0,1] &\longrightarrow a \\ t &\longmapsto S_1^a + t\,(S_2^a - S_1^a) \end{aligned}$$

(ainsi :



).

<u>Définition</u> (polynôme en s)

Soient $n \in \mathbb{N}$ et $q : a \to \mathbb{R}$. On dit que q est un polynôme en s de degré $\leq n$ si la fonction $t \in [0,1] \longrightarrow q(\gamma(t)) \in \mathbb{R}$ est un polynôme en t (de $\mathbb{R}$ dans $\mathbb{R}$) de degré $\leq n$.

Notation

On note l'ensemble de ces polynômes par

$$\mathcal{P}_m^s \quad \text{ou} \quad \operatorname{vect}\{1, s, s^2, \dots, s^m\}.$$

C'est une notation formelle

car $s \in a \subset \mathbb{R}^2$, d'où les s^k

ne sont évidemment pas les

puissances d'un réel s !

Exercice

Soit $q : s \in a \mapsto (s - S_1^a) \underbrace{\cdot}_{\text{produit scalaire}} (s - S_2^a) \in \mathbb{R}$. Montrez que $q \in \mathcal{P}_2^s$.

On munit maintenant a de l'espace d'interpolation

$$\boxed{\mathcal{P}_a = \left\{ q : a \to \mathbb{R} \;\middle|\; q = p|_a \text{ pour un certain } p \in \mathcal{P}_e \right\}.}$$

restriction de p à a

Remarque : $p = p(x_1, x_2)$ est un polynôme à 2 variables,

alors que $q = q(s)$ est « comme un polynôme à 1 variable ».

Exemple

On rappelle que $\mathcal{P}_e$ est l'espace d'interpolation choisi sur la maille e. On avait pris

$$\mathcal{P}_e = Q_1 = \text{vect}\{1, x_1, x_2, x_1 x_2\}.$$

Si on prend le polynôme $p(x_1, x_2) = x_1$, alors sa restriction $q = p|_a$ est la fonction

$$q : s \in a \longmapsto s_1 \in \mathbb{R},$$

où $s = (s_1, s_2)$ désigne les coordonnées de $s \in \mathbb{R}^2$.

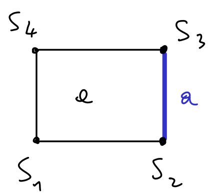
Exercice

Montrez que $\boxed{\mathcal{P}_a = \mathcal{P}_1^s = \text{vect}\{1, s\}.}$

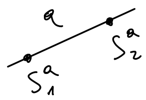
Précisons maintenant les ddls que l'on choisit sur l'arête.

Sur la maille e, on avait pris les suivants :

$$\begin{cases} L_1^e(p) = p(S_1), \\ \quad \vdots \\ L_4^e(p) = p(S_4). \end{cases}$$



Sur l'arête



on choisit les mêmes ddls (en ne gardant que « ceux qui sont supportés par l'arête »). Cela nous donne :

$$\boxed{\begin{cases} L_1^a(q) = q(S_1^a), \\ L_2^a(q) = q(S_2^a). \end{cases}}$$

<u>Remarque</u>

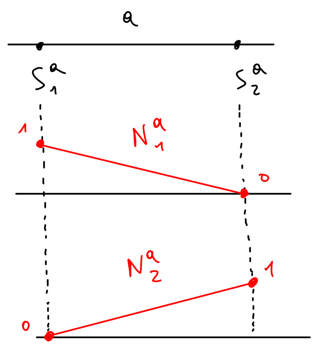
Le triplet $\left(a, \mathcal{P}_a, \{L_1^a, L_2^a\}\right)$ est un élément fini.

De plus, avec le choix des éléments finis Q_1 sur les mailles, on retrouve les éléments finis de Lagrange $\mathcal{P}_1$ « mono-dimensionnels » sur les arêtes ; voir le cours en dimension 1.

Base

Notons $\{N_1^a, N_2^a\}$ la base d'interpolation de $\mathcal{P}_a$. On rappelle que : $\boxed{L_k^a(N_m^a) = \delta_{km} \quad \forall k, m \in \{1, 2\}.}$

D'où :



Exercice

Montrez que pour tout $s \in a \subset \mathbb{R}^2$,

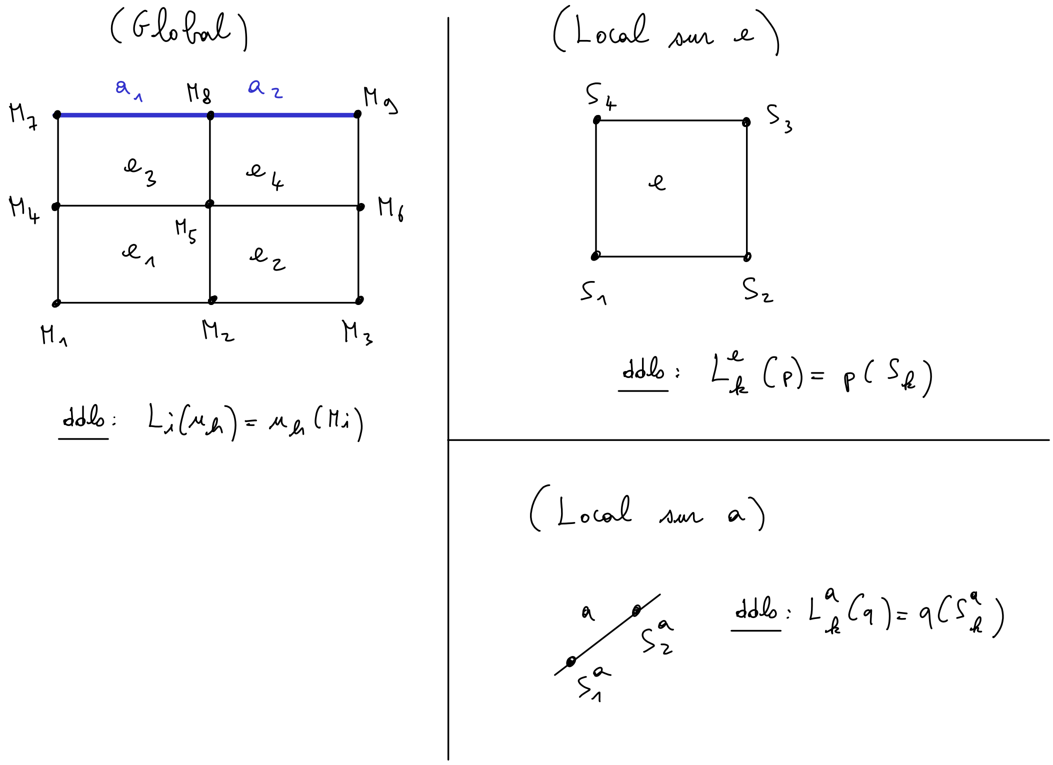
$$\boxed{\begin{aligned} N_1^a(s) &= \frac{(s - S_2^a) \cdot (S_1^a - S_2^a)}{\| S_1^a - S_2^a \|^2}, \\ N_2^a(s) &= \frac{(s - S_2^a) \cdot (S_2^a - S_1^a)}{\| S_2^a - S_1^a \|^2}. \end{aligned}}$$

norme euclidienne

3) Matrices des correspondances

Nous avons introduit trois façons différentes de numéroter les ddls, selon que l'on considère les ddls L_i sur Ω, L_k^e sur une maille e, ou L_k^a sur une arête a.

Les numéros choisis sur $\overline{\Omega}$ s'appellent les « numéros globaux » et ceux choisis sur e et a les « numéros locaux ». Rappelons les ci-dessous :



Dans la suite, nous aurons besoin de connaître toutes les correspondances entre les numéros globaux i et locaux k. Rappelons d'abord celles que l'on avait entre $\overline{\Omega}$ et les mailles. Elles sont données sous la forme d'une matrice (comme au TD no. 2). Avec les choix de numérotations que l'on a fait précédemment, on obtient :

$$\text{Elements} = \begin{array}{cc} \begin{pmatrix} 1 & 2 & 5 & 4 \\ 2 & 3 & 6 & 5 \\ 4 & 5 & 8 & 7 \\ 5 & 6 & 9 & 8 \end{pmatrix} & \begin{array}{l} \leftarrow e = e_1 \\ \leftarrow e_2 \\ \leftarrow e_3 \\ \leftarrow e_4 \end{array} \\ \begin{array}{cccc} \uparrow & \uparrow & \uparrow & \uparrow \\ k=1 & 2 & 3 & 4 \end{array} & \end{array}$$

où

$$\boxed{i = \text{Elements}(e, k) \Longleftrightarrow \left[\phi_i = N_k^e \text{ sur } e\right].}$$

Choisissons maintenant des correspondances entre Ω et les arêtes, sous la forme de la matrice suivante :

$$\text{Aretes} = \begin{array}{cc} \begin{pmatrix} 7 & 8 \\ 8 & 9 \end{pmatrix} & \begin{array}{l} \leftarrow a = a_1 \\ \leftarrow a_2 \end{array} \\ \begin{array}{cc} \uparrow & \uparrow \\ k=1 & 2 \end{array} & \end{array}$$

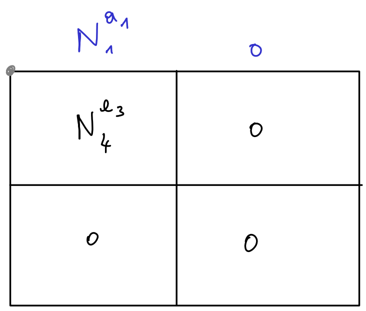
où

$$\boxed{i = \text{Aretes}(a, k) \Longleftrightarrow \left[\phi_i = N_k^a \text{ sur } a\right].}$$

Celà nous permet d'écrire les fonctions de base globales ϕ_i à l'aide des fonctions de base locales N_k^e et N_k^a.

Par exemple, les restrictions de ϕ_7 sur chaque maille et arête sont

(ϕ_7 prends la valeur 1 en ce noeud, et 0 sur les autres)



$(= \phi_7)$

Ce qui veut dire que

$$\phi_7 = \begin{cases} 0 & \text{sur } e_1, e_2, e_4, \\ N_4^{e_3} & \text{sur } e_3, \end{cases} \qquad \text{(sur les mailles)}$$

et

$$\phi_7 = \begin{cases} N_1^{a_1} & \text{sur } a_1, \\ 0 & \text{sur } a_2 . \end{cases} \qquad \text{(sur les arêtes)}$$

4) Assemblage

Détaillons maintenant la méthode que nous allons utiliser pour calculer les A_{ij} et F_i. Il est utile de rappeler les formules de ces coefficients. On a vu que

$$A_{ij} = \begin{cases} a(\phi_j, \phi_i) & \text{si } i \notin I_D, \\ \delta_{ij} & \text{si } i \in I_D, \end{cases}$$

lignes de Dirichlet

$$F_i = \begin{cases} \ell(\phi_i) & \text{si } i \notin I_D, \\ \alpha(M_i) & \text{si } i \in I_D, \end{cases}$$

où $I_D = \{1, \dots, 9\} \setminus \{5, 8\}$,

$$a(\phi_j, \phi_i) = \underbrace{\int_\Omega \vec{\nabla}\phi_j \cdot \vec{\nabla}\phi_i}_{\text{termes de l'équation}} + \underbrace{\int_{\Gamma_N} b\, \phi_j\, \phi_i}_{\text{termes de Neuman}}$$

et $$\ell(\phi_i) = \overbrace{\int_\Omega f\, \phi_i} + \overbrace{\int_{\Gamma_N} \beta\, \phi_i}.$$

Cela nous donne les formes générales suivantes :

(avec des zéros hors de la diagonale)

$$A = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ \hline & & & & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ \hline & & & & & & & & \\ & & & & & & & & 1 \end{pmatrix}, \quad \begin{array}{c} \leftarrow i=5 \rightarrow \\ \\ \leftarrow i=8 \rightarrow \end{array} \quad F = \begin{pmatrix} \alpha(M_1) \\ \vdots \\ \alpha(M_4) \\ \hline \alpha(M_6) \\ \alpha(M_7) \\ \hline \alpha(M_9) \end{pmatrix}$$

Il reste à calculer les A_{ij} et F_i, pour $i \notin I_D$. Pour cela, on doit introduire la définition ci-dessous.

Définition (Coefficients élémentaires)

Pour toute maille e et tout $k, m \in \{1, \dots, 4\}$, on pose

$$\boxed{\begin{cases} A^e_{km} = \int_e \vec{\nabla} N^e_m \cdot \vec{\nabla} N^e_k \,, \\ F^e_k = \int_e f\, N^e_k \,. \end{cases}}$$

Pour toute arête a et tout $k \in \{1, 2\}$, on pose

$$\boxed{\begin{cases} A^a_{km} = \int_a b\, N^a_m N^a_k \,, \\ F^a_k = \int_a \beta\, N^a_k \,. \end{cases}}$$

Terminologie

Les matrices

$$A^e = (A^e_{km})_{1 \le k, m \le 4} \quad \text{et} \quad A^a = (A^a_{km})_{1 \le k, m \le 2}$$

s'appellent les « matrices élémentaires ». Les vecteurs

$$F^e = (F^e_k)_{1 \le k \le 4} \quad \text{et} \quad F^a = (F^a_k)_{1 \le k \le 2}$$

s'appellent les « vecteurs second membres élémentaires ».

Ces coefficients sont des versions locales des intégrales obtenues dans les formules de $a(\phi_j, \phi_i)$ et $l(\phi_i)$. On les a définis en remplaçant respectivement Ω et Γ_N, par e et a, et les fonctions de bases globales par leurs expressions locales. Ainsi, il est possible d'écrire les A_{ij} et F_i comme des sommes de coefficients élémentaires (lorsque $i \notin I_D$).

Par exemple,

car $i \notin I_D$

$$
\begin{aligned}
A_{87} = a(\phi_7, \phi_8) &= \int_\Omega \vec{\nabla}\phi_7 \cdot \vec{\nabla}\phi_8 + \int_{\Gamma_N} b\, \phi_7\, \phi_8 \\
&= \int_{e_3} \vec{\nabla} N_4^{e_3} \cdot \vec{\nabla} N_3^{e_3} + \int_{a_1} b\, N_1^{a_1}\, N_2^{a_1} \\
&= A_{34}^{e_3} + A_{21}^{a_1}
\end{aligned}
$$

car d'après Elements et Aretes, les expressions locales de ϕ_7 et ϕ_8 sont

ϕ_7

$N_1^{a_1}$	0
$N_4^{e_3}$	0
0	0

ϕ_8

$N_2^{a_1}$	$N_1^{a_2}$
$N_3^{e_3}$	$N_1^{e_4}$
0	0

On constate que cette décomposition peut se déduire directement des matrices des correspondances en

* remplaçant les numéros globaux i et j par leurs numéros locaux correspondants k et m sur chaque maille et arête, d'après Elements et Aretes,
* et en ne considérant que les mailles et arêtes, dont les lignes correspondantes de Element et Aretes contiennent ces numéros globaux (sinon les fonctions globales ϕ_i et/ou ϕ_j sont nulles).

(Voir le TD no. 2 pour plus de détails.)

En effet :

$j = 7$ corresponds respectivement à $m = 4$ sur e_3 et $m = 1$ sur a_1

$$A_{87} = A^{e_3}_{34} + A^{a_1}_{21}.$$

$i = 8$ corresponds respectivement à $k = 3$ sur e_1 et $k = 1$ sur a_1

On décompose les F_i de la même manière et on trouve que

$$F = \begin{pmatrix} \alpha(M_1) \\ \alpha(M_2) \\ \alpha(M_3) \\ \alpha(M_4) \\ F_3^{e_1} + F_4^{e_2} + F_2^{e_3} + F_1^{e_4} \\ \alpha(M_6) \\ \alpha(M_7) \\ F_3^{e_3} + F_4^{e_4} + F_2^{a_1} + F_1^{a_2} \\ \alpha(M_9) \end{pmatrix}$$

et $A =$ (à faire à la maison).

5) Algorithme d'assemblage

Dans la pratique, on implémente A et F par ordinateur à l'aide d'un algorithme basé sur les calculs précédents.

Algorithme

(Initialisation)

$A = 0$; $F = 0$; (matrice et vecteur nuls)

(Contribution de l'équation)

Pour tout $e = e_1, \dots, e_4$, faire :

Calcul de A^e et F^e;

Pour tout $k = 1, \dots, 4$, faire :

$i = \text{Elements}(e, k)$;

$F_i = F_i + F^e_k$;

Pour tout $m = 1, \dots, 4$, faire :

$j = \text{Elements}(e, m)$;

$A_{ij} = A_{ij} + A^e_{km}$;

fin

fin

fin

(Contribution de Neuman)

Pour tout $a = a_1, a_2$, faire :

Calcul de A^a et F^a;

Pour tout $k = 1, 2$, faire :

$i = \text{Aretes}(a, k)$;

$F_i = F_i + F_k^a$;

Pour tout $m = 1, 2$, faire :

$j = \text{Aretes}(a, m)$;

$A_{ij} = A_{ij} + A_{km}^a$;

fin

fin

fin

(Contribution de Dirichlet)

Pour tout $i \in I_D$, faire :

$F_i = \alpha(M_i)$; (En général, c'est $\alpha(M_{L_i})$.)

Pour tout $j = 1, \ldots, 12$, faire:

⋮

⋮

⋮ ⋮

$A_{ij} = \delta_{ij}$;

fin

fin

V) Intégration des coefficients élémentaires

Il nous reste à calculer A^e, F^e, A^a et F^a. Nous allons le faire de manière approchée, en utilisant des formules d'intégration numérique.

1) Intégration sur $\mathbb{R}$

Voici un exemple de formule en dimension 1.

<u>Lemme</u> (Trapèzes)

Soient deux réels $S < T$. Alors

$$\frac{1}{T-S}\int_S^T g(t)\,dt = \sum_{i=1}^{2} \omega_i\, g(t_i) \qquad \forall g \in \text{vect}\{1,t\} = \mathbb{P}_1^t,$$

où les t_i et ω_i sont donnés par $\underset{t_1 = S}{(\omega_1 = \frac{1}{2})}$ —— $\underset{t_2 = T}{(\omega_2 = \frac{1}{2})}$.

(Voir les TDs et TPs pour d'autres formules.)

Remarque

Si $g \notin \mathcal{P}_1^t$, on obtient seulement une approximation qui « sera d'autant plus précise que $T-S$ sera petit ».

Terminologie

Les t_i et w_i sont respectivement les points et poids d'intégration. Le plus haut degré des polynômes pour lesquels cette formule est vraie mesure la précision de l'approximation.

Exercice

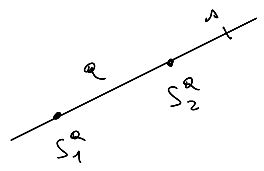
1) Montrez cette formule pour $g(t) = 1$ puis $g(t) = t$.

2) En déduire qu'elle est vraie pour tout $g \in \mathcal{P}_1^t$.

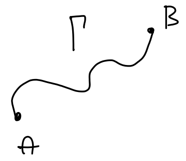
3) Montrons qu'elle est fausse pour $g(t) = t^2$.

2) Intégration sur les arêtes

On aura exactement les mêmes formules. Montrons le pour une arête générale



(de variable s). Rappelons d'abord la notion d'intégrale le long du bord, qui apparaît dans la formule de Green-Gauss. Soit donc une courbe du plan



c'est-à-dire un sous ensemble non vide Γ de $\mathbb{R}^2$ que l'on peut paramétriser au sens ci-dessous.

Définition

On dit que γ est une paramétrisation de Γ si

(i) γ est une fonction de $[0,1]$ dans $\mathbb{R}^2$ suffisamment régulière,

(ii) $\gamma([0,1]) = \Gamma$, c'est-à-dire

$$\Gamma = \left\{ \gamma(t) \mid t \in [0,1] \right\} \subseteq \mathbb{R}^2,$$

(iii) et $\gamma :]0,1[\longrightarrow \Gamma$ est injective, c'est-à-dire

$$(\forall t_1, t_2 \in]0,1[) \quad [\gamma(t_1) = \gamma(t_2) \Rightarrow t_1 = t_2]$$

(avec $\gamma(0) = A$ et $\gamma(1) = B$, ou l'inverse).

Exemple

$\gamma : t \mapsto S_1^a + t(S_2^a - S_1^a)$ est une paramétrisation de l'arête a.

L'intégrale sur Γ qui intervient dans la formule de Green-Gauss est la suivante :

Théorème et définition

Soit $q : \Gamma \longrightarrow \mathbb{R}$ suffisament régulière. Pour toute paire de paramétrisations γ_1 et γ_2 de Γ, on a

$$\int_0^1 q(\gamma_1(t)) \, \|\gamma_1'(t)\| dt = \int_0^1 q(\gamma_2(t)) \, \|\gamma_2'(t)\| dt .$$

Ce réel dépends donc seulement de q et de Γ, mais pas de la paramétrisation choisie. On le définit alors comme étant l'intégrale de q sur Γ, que l'on note $\int_\Gamma q(s)\, ds$

ou encore $\int_\Gamma q$ (en abrégé) ; c'est-à-dire

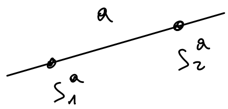
$$\int_\Gamma q = \int_0^1 q(\gamma(t))\, \|\gamma'(t)\|\, dt$$

où γ est n'importe quelle paramétrisation possible de Γ.

On peut maintenant montrer la formule ci-dessous.

<u>Lemme</u> (Trapèzes sur les arêtes)

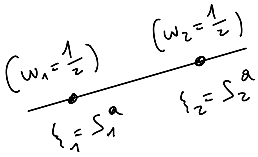
Soit $a \subset \mathbb{R}^2$ un segment comme ci-dessous



Alors

$$\boxed{\frac{1}{|a|}\int_a g(s)\, ds = \sum_{i=1}^{2} w_i\, g(\xi_i) \qquad \forall g \in \mathbb{P}_1^1}$$

où les points et poids d'intégration sont



et $|a| = \|S_2^a - S_1^a\|$ désigne la longueur de a.

<u>Preuve</u>

D'après les définitions précédentes, on a

$$\frac{1}{|a|}\int_a q(s)\,ds = \frac{1}{\|S_2^a - S_1^a\|}\int_0^1 q(\gamma(t))\,\|\gamma'(t)\|\,dt$$

où on choisit la paramétrisation

$$\gamma : t \mapsto S_1^a + t(S_2^a - S_1^a).$$

Par suite, $\gamma'(t) = S_2^a - S_1^a$ et

$$\frac{1}{|a|}\int_a q(s)\,ds = \int_0^1 q(\gamma(t))\,dt.$$

Or, si $q \in \mathcal{P}_1^s$ alors $t \mapsto q(\gamma(t)) \in \mathcal{P}_1^t$ (par définition de l'espace $\mathcal{P}_1^s$). On peut donc utiliser la formule des trapèzes sur $\mathbb{R}$, qui implique que

$$\frac{1}{|a|}\int_a q(s)\,ds \underset{\begin{cases} t_1 = 0 \\ t_2 = 1 \\ \omega_1 = \omega_2 = \frac{1}{2}\end{cases}}{=} \sum_{i=1}^{2} \omega_i\, q(\gamma(t_i))$$

$$\underset{\begin{cases} \xi_1 = \gamma(t_1) = S_1^a \\ \xi_2 = \gamma(t_2) = S_2^a \end{cases}}{=} \sum_{i=1}^{2} \omega_i\, q(\xi_i).$$

Ceci termine la preuve. ∎

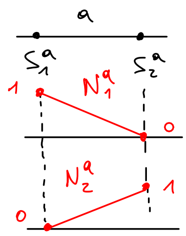
3) Calcul de A^a et F^a

On peut maintenant intégrer les coefficients élémentaires sur les arêtes de manière approchée, en choisissant la formule des trapèzes. Détaillons les calculs pour

$$F^a = \left(F_k^a\right)_{1 \leq k \leq 2},$$

par exemple.

On a $\quad F_k^a = \int_a \beta\, N_k^a \quad$ où



par les trapèzes

$$\simeq \frac{|a|}{2} \sum_{m=1}^{2} \beta(S_m^a)\, \underbrace{N_k^a(S_m^a)}_{= \delta_{km}}$$

d'où $\quad F_k^a \simeq \frac{|a|}{2}\, \beta(S_k^a)$; c'est-à-dire

$$\boxed{F^a \simeq \frac{|a|}{2} \begin{pmatrix} \beta(S_1^a) \\ \beta(S_2^a) \end{pmatrix}}.$$

De la même manière, on trouve que

$$\boxed{A^a \simeq \frac{|a|}{2}\begin{pmatrix} b(S_1^a) & 0 \\ 0 & b(S_2^a) \end{pmatrix}.}$$

4) <u>Calcul de A^e et F^e</u>

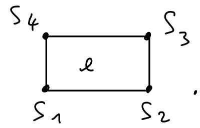
Les formules d'intégration numérique sur les rectangles sont obtenues simplement à partir des formules sur $\mathbb{R}$, lors des intégrations successives en x_1 et x_2

<u>Remarque</u>

Pour des éléments triangulaires, voir l'appendice.

Si on choisit les trapèzes en x_1 et aussi en x_2, on obtient :

<u>Lemme</u> (Trapèzes $\times$ trapèzes)

Soit $e \subset \mathbb{R}^2$ un rectangle de la forme 

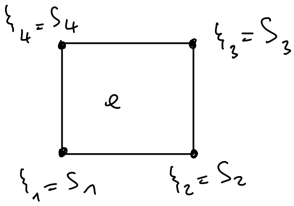
.

Notons $|e|$ l'aire de e ; c'est-à-dire $|e| = \|S_2 - S_1\| \times \|S_4 - S_1\|$.

Alors

$$\boxed{\frac{1}{|e|}\int_e g(x_1,x_2)\,dx_1\,dx_2 = \sum_{i=1}^{4} \omega_i\, g(\xi_i) \qquad \forall g \in Q_1 = \mathrm{Vect}\{1, x_1, x_2, x_1x_2\}}$$

avec



$\left(\omega_1 = \dots = \omega_4 = \frac{1}{4}\right)$.

<u>Preuve</u> (Exercice).

<u>Remarque</u>: on peut choisir d'autres formules en x_1 et x_2,

Comme par exemple

(degré ≤ 1 avec le choix des trapèzes)

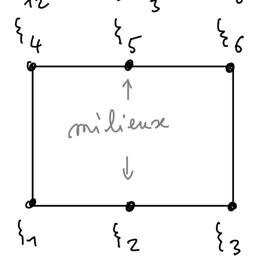
$$\frac{1}{|e|}\int_e g = \sum_{i=1}^{6} \omega_i\, g(\xi_i) \qquad \forall g \in \mathrm{vect}\{1, x_1, x_2, x_1x_2, x_1^2x_2, x_1^3x_2\}$$

(Simpson $\times$ trapèzes)

(degré ≤ 3 avec le choix de Simpson (voir le TD no. 1))

où

$\omega_4 = \frac{1}{12}$ $\quad\omega_5 = \frac{1}{3}$ $\quad\omega_6 = \frac{1}{12}$



poids de Simpson

poids des trapèzes

$\omega_1 = \frac{1}{6} \times \frac{1}{2} = \frac{1}{12}$ $\quad \omega_2 = \frac{4}{6} \times \frac{1}{2} = \frac{1}{3}$ $\quad \omega_3 = \frac{1}{6} \times \frac{1}{2} = \frac{1}{12}$

En utilisant la formule « trapèzes × trapèzes », on obtient les approximations suivantes :

$$
\boxed{
\begin{aligned}
(1) \qquad A^e_{km} &= \int_e \vec{\nabla} N^e_m \cdot \vec{\nabla} N^e_k \\
&\simeq \frac{|e|}{4} \sum_{i=1}^{4} \vec{\nabla} N^e_m(\xi_i) \cdot \vec{\nabla} N^e_k(\xi_i) \\
\text{et} \qquad F^e_k &= \int_e f N^e_k \simeq \frac{|e|}{4} \sum_{i=1}^{4} f(\xi_i)\, N^e_k(\xi_i), \\
\text{où } \xi_i &= S_i.
\end{aligned}
}
$$

Pour continuer, il nous faut calculer

* les $N^e_k(\xi_i)$,
* les $\vec{\nabla} N^e_k(\xi_i)$ et les $\vec{\nabla} N^e_m(\xi_i)$,

pour toutes fonctions de base locales, N^e_k et N^e_m, et tout point d'intégration ξ_i (de la formule choisie).

Faisons le avec une méthode générale qui sera implémentable par ordinateur. Dans la suite, on note

$$S_k = (S_{k1}, S_{k2}) \in \mathbb{R}^2$$

les coordonnées des noeuds.

On rappelle que, puisque le triplet

$$(e, \mathcal{P}_e, \{L_1^e, \dots, L_4^e\})$$

choisi sur les rectangles lors du maillage est un élément fini, on a la formule suivante :

$$\boxed{\begin{array}{c} p(x_1, x_2) = \displaystyle\sum_{k=1}^{4} L_k^e(p) \; N_k^e(x_1, x_2) \\ \forall p \in \mathcal{P}_e, \; \forall x_1, x_2 \in \mathbb{R} \end{array}}$$

(où $\{N_1^e, \dots, N_4^e\}$ était la base d'interpolation).

Utilisons cette formule pour écrire la base canonique de $\mathcal{P}_e = \text{Vect}\{1, x_1, x_2, x_1 x_2\}$ sur la base d'interpolation.

En choisissant par exemple le polynôme

$$p(x_1, x_2) = x_2,$$

on trouve que pour tout $x_1, x_2 \in \mathbb{R}$,

$$x_2 = \sum_{k=1}^{4} \underbrace{L_k^e(p)}_{\substack{\parallel \\ p(S_k) \\ \parallel \\ S_{k2}}} \; N_k^e(x_1, x_2)$$

d'où $\quad x_2 = S_{12} N_1^e(x_1,x_2) + \dots + S_{42} N_4^e(x_1,x_2)$.

On fait la même chose pour tout les autres polynômes de la base canonique, puis on écrit le système obtenu sous forme matricielle. Cela donne

$$(2) \qquad \boxed{\begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ S_{11} & S_{21} & S_{31} & S_{41} \\ S_{12} & S_{22} & S_{32} & S_{42} \\ S_{11}S_{12} & S_{21}S_{22} & S_{31}S_{32} & S_{41}S_{42} \end{pmatrix}}_{\text{notée } M} \begin{pmatrix} N_1^e(x_1,x_2) \\ \vdots \\ N_4^e(x_1,x_2) \end{pmatrix} \quad \forall x_1, x_2 \in \mathbb{R}.}$$

En dérivant en x_1 et x_2, on a aussi

$$(3) \qquad \boxed{\begin{pmatrix} 0 \\ 1 \\ 0 \\ x_2 \end{pmatrix} = M \begin{pmatrix} \frac{\partial N_1^e}{\partial x_1}(x_1,x_2) \\ \vdots \\ \frac{\partial N_4^e}{\partial x_1}(x_1,x_2) \end{pmatrix} \qquad \forall x_1, x_2 \in \mathbb{R}}$$

et

$$(4)\qquad \boxed{\begin{pmatrix} 0 \\ 0 \\ 1 \\ x_1 \end{pmatrix} = M \begin{pmatrix} \dfrac{\partial N_1^e}{\partial x_2}(x_1, x_2) \\ \vdots \\ \dfrac{\partial N_4^e}{\partial x_2}(x_1, x_2) \end{pmatrix} \qquad \forall x_1, x_2 \in \mathbb{R}.}$$

La suite des calculs de A^e_{km} et F^e_k se fait alors par ordinateur de la manière suivante :

* on rappelle que M est inversible car c'est la matrice de passage entre la base canonique et la base d'interpolation ;

* on calcule alors les

$$N_k^e(\xi_i), \quad \overrightarrow{\nabla} N_k^e(\xi_i) \quad \text{et} \quad \overrightarrow{\nabla} N_m^e(\xi_i)$$

en choisissant $(x_1, x_2) = \xi_i$ dans (2), (3) et (4), puis en multipliant ces égalités par M^{-1} ;

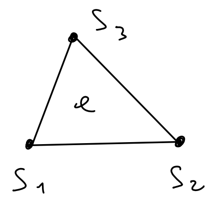
* et enfin on injecte les résultats obtenus dans (1).

(Voir les TDs et TPs).

$\boxed{\text{Annexe}}$

A) Eléments finis triangulaires

Dans la pratique, on peut utiliser différents types de mailles. En plus des rectangulaires, les plus communes sont les triangulaires de la forme



Les éléments finis les plus simples sont alors ceux de Lagrange $\mathcal{P}_1$ « bidimensionnels », où

$$\mathcal{P}_e = \text{Vect}\{1, x_1, x_2\} \underset{\text{noté}}{=} \mathcal{P}_1^{(x_1,x_2)} \underset{\text{ou}}{=} \mathcal{P}_1$$

et les ddls sont

$$L_1^e(p) = p(S_1), \quad L_2^e(p) = p(S_2) \quad \text{et} \quad L_3^e(p) = p(S_3).$$

(Voir le TD no. 3 pour plus de détails.)

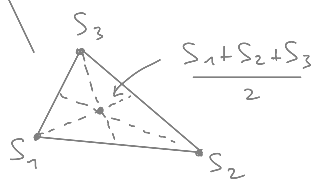
B) Intégration numérique sur les triangles

Les formules sont alors légèrement plus compliquées.

En voici quelques unes (où $|e|$ désigne l'aire de e) :

Lemme (Centre)

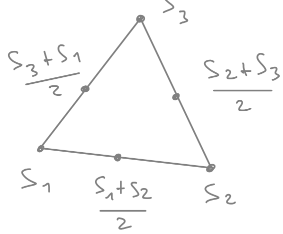
$$\frac{1}{|e|}\int_e g = g\left(\frac{S_1+S_2+S_3}{3}\right) \qquad \forall g \in \mathbb{P}_1 = \text{vect}\{1, x_1, x_2\}$$



Lemme (milieux des arêtes)

$$\frac{1}{|e|}\int_e g = \frac{1}{3}\left(g\left(\frac{S_1+S_2}{2}\right) + g\left(\frac{S_2+S_3}{2}\right) + g\left(\frac{S_3+S_1}{2}\right)\right)$$

$$\forall g \in \mathbb{P}_2 = \text{vect}\{1, x_1, x_2, x_1x_2, x_1^2, x_2^2\}.$$



Chapitre 2

Travaux dirigées corrigés

2.1 Un problème de poutre 1-d

2.1.1 TD no. 1 : interpolation et intégration

(Tournez la page.)

TD no. 1

Approximation des EDPs, Printemps 2016-2017

Exercice 1 (Eléments finis)**.** Soient

- $e = [S_1, S_2]$ avec $S_1 < S_2$,
- $\mathcal{P}_e = \mathcal{P}_2 = \text{vect}\,\{1, x, x^2\}$,
- et les fonctions $L_k^e : \mathcal{P}_e \to \mathbb{R}$ définies par

$$L_1^e(p) = p(S_1), \quad L_2^e(p) = \frac{1}{|e|}\int_e p(x)\,dx \quad \text{et} \quad L_3^e(p) = p(S_2),$$

où $|e| = S_2 - S_1$ désigne la longueur de e.

1. Montrez que les L_k^e sont linéaires.
2. Montrez que

$$(\forall p \in \mathcal{P}_e) \quad [L_1^e(p) = L_2^e(p) = L_3^e(p) = 0 \Rightarrow p = 0]\,.$$

3. En déduire que pour tout $n = 1, 2, 3$, il existe un unique $q \in \mathcal{P}_e$ tel que

$$L_k^e(q) = \delta_{kn} \quad \forall k = 1, 2, 3$$

(où $\delta_{kn} = 1$ si $k = n$ et 0 sinon). On notera ce polynôme par $q = N_n^e$.

4. Montrez que

$$p = \sum_{k=1}^{3} L_k^e(p)\,N_k^e \quad \forall p \in \mathcal{P}_e.$$

5. En déduire que $\{N_1^e, N_2^e, N_3^e\}$ est une base et précisez les coordonnées de $p(x) = 2\,x^2 - x + 3$ sur les bases $\{1, x, x^2\}$ et $\{N_1^e, N_2^e, N_3^e\}$.

Exercice 2 (Formule de Simpson)**.** Soit $e = [S_1, S_2]$, $S_1 < S_2$.

1. Montrez que

$$\frac{1}{|e|}\int_e g(x)\,dx = \frac{1}{6}\,(g(S_1) + 4g(m) + g(S_2)) \quad \forall g \in \mathcal{P}_3,$$

où $m = \frac{S_1+S_2}{2}$, $|e| = S_2 - S_1$ et $\mathcal{P}_3 = \text{vect}\,\{1, x, x^2, x^3\}$.

2. Montrez qu'il existe $g \in \mathcal{P}_4$ telle que cette formule soit fausse.

Exercice 3 (Base d'interpolation)**.** Soient e, $\mathcal{P}_e$ et $\{L_1^e, L_2^e, L_3^e\}$, définis comme dans l'exercice 1, dont on reprendra les notations.

1. Montrez qu'il existe une constante $c \in \mathbb{R}$ telle que

$$N_2^e(x) = c(x - S_1)(x - S_2) \quad \forall x \in \mathbb{R}.$$

   Vous préciserez de plus la valeur exacte de cette constante.

2. Montrez qu'il existe $x_* \in]S_1, S_2[$ tel que

$$N_1^e(x) = \frac{(x - x_*)(x - S_2)}{(S_1 - x_*)(S_1 - S_2)} \quad \forall x \in \mathbb{R}.$$

3. Montrez que $x_* = \frac{2S_1+S_2}{3}$.
4. En déduire les formules explicites de la base $\{N_1^e, N_2^e, N_3^e\}$ et dessinez les graphes de ces fonctions.

Corrigé du TD no. 1

Rappels d'algèbre linéaire

Soient $I \subseteq \mathbb{R}$, $I \neq \emptyset$, et

$$E = \{ \text{fonctions } f : I \to \mathbb{R} \} .$$

On peut alors définir une structure « naturelle » d'espace vectoriel sur E. Rappelons là :

$\boxed{\text{Somme } +_E}$

Etant données $f, g \in E$, on définit $f +_E g \in E$ par

$$f +_E g : \begin{array}{l} I \longrightarrow \mathbb{R} \\ x \mapsto f(x) + g(x) \end{array} .$$

$\boxed{\text{Multiplication } \cdot_E}$

Etant donnés $f \in E$ et $\lambda \in \mathbb{R}$, on définit $\lambda \cdot_E f$ par

$$\lambda \cdot_E f : \begin{array}{l} I \longrightarrow \mathbb{R} \\ x \mapsto \lambda f(x) \end{array} .$$

Alors :

tout nos espaces vectoriels seront sur $\mathbb{R}$

Proposition (à faire à la maison)

$(E, +_E, \cdot_E)$ est un espace vectoriel (sur $\mathbb{R}$) dont l'élément neutre pour $+_E$ est la fonction

$$0_E : \begin{array}{rcl} I & \longrightarrow & \mathbb{R} \\ x & \longmapsto & 0 \end{array}.$$

Dans la suite, on note simplement

$$f +_E g = f + g, \quad \lambda \cdot_E f = \lambda f, \quad 0_E = 0, \quad \text{etc.}$$

Exemples

Les espaces V, V_e, $\mathcal{P}_e$, $\mathcal{P}_M$, etc., du cours, sont tous des espaces vectoriels avec la même structure.

Exercice 1

1) Montrons que

$$L_1^e : \begin{array}{rcl} \mathcal{P}_e & \longrightarrow & \mathbb{R} \\ p & \longmapsto & p(S_1) \end{array}$$

est linéaire.

On rappelle que $e = [S_1, S_2]$ et $\mathcal{P}_e = \text{vect}\{1, x, x^2\}$ est l'espace des polynômes de $\mathbb{R}$ dans $\mathbb{R}$ de degré ≤ 2.

(C'est un espace de fonctions et aussi un espace vectoriel avec la structure précédente.)

On doit montrer que pour tout $p, q \in \mathcal{P}_e$ et $\lambda \in \mathbb{R}$,

$$\begin{cases} L_1^e(p+q) = L_1^e(p) + L_1^e(q), & (*) \\ L_1^e(\lambda p) = \lambda L_1^e(p). & (**) \end{cases}$$

On a

$$\begin{aligned} L_1^e(p+q) &= (p+q)(S_1) \\ &= p(S_1) + q(S_1) \\ &= L_1^e(p) + L_1^e(q), \end{aligned}$$

définition de L_1^e

définition de la somme de fonctions

ce qui montre $(*)$. On montre $(**)$ de la même manière.

On a donc montré que L_1^e est linéaire. On raisonne de même pour L_2^e et L_3^e.

2) Soit $p \in \mathcal{P}_e$ tel que $L_1^e(p) = L_2^e(p) = L_3^e(p) = 0$.

On doit montrer que p est le polynôme nul, ie que

$$p(x) = 0 \qquad \forall x \in \mathbb{R}.$$

Utilisons d'abord que

$$\begin{cases} L_1^e(p) = p(S_1) = 0 \\ \text{et } L_3^e(p) = p(S_2) = 0. \end{cases}$$

Ce polynôme $p \in \mathcal{P}_2$ a donc 2 racines et par suite :

$$p(x) = c\,(x - S_1)(x - S_2) \qquad \forall x \in \mathbb{R}$$

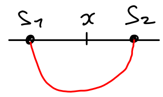
pour une certaine constante $c \in \mathbb{R}$. Mais alors

$$0 = L_2^e(p) = \frac{1}{|e|} \int_e p(x)\, dx$$
$$= \frac{c}{|e|} \underbrace{\int_{S_1}^{S_2} (x - S_1)(x - S_2)\, dx}_{\neq 0}.$$

car ce polynôme a un signe constant entre ses 2 racines

On en déduit que $c = 0$

et ceci termine la preuve.



$x \mapsto (x - S_1)(x - S_2)$

3) Posons $T : p \in \mathcal{P}_e \mapsto \begin{pmatrix} L_1^e(p) \\ L_2^e(p) \\ L_3^e(p) \end{pmatrix} \in \mathbb{R}^3$.

Cette application est linéaire car les L_k^e le sont.

Elle est de plus injective, car si $T(p) = T(q)$ alors $T(p-q) = 0$,

ce qui veut dire que $L_1^e(p-q) = L_2^e(p-q) = L_3^e(p-q) = 0$

et on en déduit que $p - q = 0$ (d'après la question précédente).

De plus, $\dim \mathcal{P}_e = \dim \mathbb{R}^3 = 3$ et T est donc bijective

d'après le théorème du rang. Ceci implique qu'il existe

un unique $q \in \mathcal{P}_e$ tel que $T(q) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, ie tel que

$$L_1^e(q) = 1, \quad L_2^e(q) = 0 \quad \text{et} \quad L_3^e(q) = 0.$$

Cela répond à la question pour $n = 1$ et on raisonne de la

même manière pour $n = 2$ et 3.

4) Soit $p \in \mathcal{P}_e$ et posons $q = \sum_{k=1}^{3} L_k^e(p) N_k^e \in \mathcal{P}_e$.

Il suffit de montrer que

$$(1) \qquad L_k^e(p) = L_k^e(q) \quad \forall k = 1,2,3,$$

d'après ce qui précède ; en effet on aura alors $T(p) = T(q)$

et l'injectivité de T impliquera que

$$p = q = \sum_{k=1}^{3} L_k^e(p)\, N_k^e .$$

Montrons donc (1).

Par linéarité de $L_m^e : \mathcal{P}_e \longrightarrow \mathbb{R}$, on a

$$L_m^e(q) = L_m^e\Big(\sum_{k=1}^{3} \underbrace{L_k^e(p)}_{\substack{\in \mathbb{R} \\ \text{ce sont des scalaires}}} N_k^e \Big)$$

$$= \sum_{k=1}^{3} L_k^e(p) \underbrace{L_m^e(N_k^e)}_{= \delta_{km}}$$

$$= L_m^e(p),$$

pour tout $m = 1, 2, 3$. Ceci termine la preuve.

5) La question précédente implique que $\{N_1^e, N_2^e, N_3^e\}$ est génératrice de $\mathcal{P}_e$. C'est donc une base, car $\dim \mathcal{P}_e = 3$.

Soit maintenant $p \in \mathcal{P}_e$ tel que

$$p(x) = 2x^2 - x + 3 \quad \forall x \in \mathbb{R}.$$

La formule précédente implique que

$$\underbrace{3 - x + 2x^2}_{\text{C'est l'écriture de } p \text{ sur la base canonique } \{1, x, x^2\}} = \underbrace{L_1^e(p)\, N_1^e(x) + L_2^e(p)\, N_2^e(x) + L_3^e(p)\, N_3^e(x)}_{\text{C'est celle sur la base } \{N_1^e, N_2^e, N_3^e\}} \quad \forall x \in \mathbb{R}.$$

Les coordonnées de p sont donc respectivement $\begin{pmatrix} 3 \\ -1 \\ 2 \end{pmatrix}$ et $\begin{pmatrix} L_1^e(p) \\ L_2^e(p) \\ L_3^e(p) \end{pmatrix}$ sur ces deux bases. Précisons les dernières coordonnées. On a

$$\begin{cases} L_1^e(p) = p(S_1), \\ L_2^e(p) = \dfrac{1}{|e|} \displaystyle\int_e p(x)\, dx = \dfrac{1}{S_2 - S_1} \int_{S_1}^{S_2} (2x^2 - x + 3)\, dx, \\ L_3^e(p) = p(S_2). \end{cases}$$

D'où

$$\boxed{\begin{aligned} L_1^e(p) &= 2S_1^2 - S_1 + 3, \\ L_2^e(p) &= \frac{2}{3}(S_1^2 + S_1 S_2 + S_2^2) - \frac{S_1 + S_2}{2} + 3, \\ L_3^e(p) &= 2S_2^2 - S_2 + 3. \end{aligned}}$$

<u>Remarque</u>

Un tel triplet $(e, \mathbb{P}_e, \{L_1^e, L_2^e, L_3^e\})$ s'appelle un élément fini et $\{N_1^e, N_2^e, N_3^e\}$ la base d'interpolation (voir le cours).

<u>Exercice 2</u>

1) On veut montrer que

(2) $$\boxed{\frac{1}{|e|}\int_e g(x)\,dx = \frac{1}{6}\Big(g(S_1) + 4\,g(m) + g(S_2)\Big) \qquad \forall g \in \mathcal{P}_3,}$$

où $e = [S_1, S_2]$ et $m = \dfrac{S_1+S_2}{2}$.

Montrons le pour les polynômes de base

$$g(x) = 1, \; g(x) = x, \text{ etc.}$$

$\boxed{\text{Pour } g(x) = 1}$

On a $\displaystyle \frac{1}{|e|}\int_e g(x)\,dx = \frac{1}{|e|}\int_e dx = 1$

et $\displaystyle \frac{1}{6}\Big(g(S_1) + 4g(m) + g(S_2)\Big) = \frac{1}{6}\Big(1 + 4\times 1 + 1\Big) = 1.$

Ceci montre que (2) est vraie pour $g(x) = 1$.

$\boxed{\text{Pour } g(x) = x,\ x^2 \text{ et } x^3}$

On montre de la même manière que (2) est vraie pour ces polynômes.

$\boxed{\text{Conclusion}}$

Par linéarité de la formule (2) en g, on en déduit que (2) est vraie pour tout $g \in \text{Vect}\{1, x, x^2, x^3\} = \mathcal{P}_3$.

2) On doit trouver $g \in \mathcal{P}_4$ tel que (2) soit fausse.

Supposons d'abord que $S_1 = 0$, $S_2 = 1$, et prenons $g(x) = x^4$.

On a $\quad \frac{1}{|e|} \int_e g(x)\,dx = \int_0^1 x^4\,dx = \frac{1}{5}$

et $\quad \frac{1}{6}\left(g(0) + 4g(\tfrac{1}{2}) + g(1)\right) = \frac{1}{6}\left(0 + 4\frac{1}{16} + 1\right) = \frac{5}{24}.$

Donc (2) est bien fausse dans ce cas.

Pour S_1 et S_2 généraux, posons

$$g(x) = \left(\frac{x - S_1}{S_2 - S_1}\right)^4.$$

Par changement de variable, on a encore

$$\frac{1}{|e|}\int_e g(x)\,dx \underset{\substack{y = \frac{x-S_1}{S_2-S_1} \\ dy = \frac{dx}{|e|}}}{=} \int_0^1 y^4\,dy = \frac{1}{5}$$

Comme tout à l'heure

et $\quad \frac{1}{6}\left(g(S_1) + 4g(m) + g(S_2)\right) = \frac{1}{6}\left(0^4 + 4\left(\frac{1}{2}\right)^4 + 1^4\right) = \frac{5}{24}.$

Remarque

Les formules du type (2) s'appellent des « formules d'intégration numérique ». Elles sont utilisées pour le calcul approché des intégrales (voir le TD no. 2, le TP no. 2 et le cours sur la dimension 2).

Exercice 3

On reprends les notations de l'exercice 1 et on calcule la base d'interpolation $\{N_1^e, N_2^e, N_3^e\}$ de $\mathcal{P}_e$ définie par les équations

$$\boxed{L_k^e(N_m^e) = \delta_{km} \quad \forall k,m \in \{1,2,3\}.}$$

1) On a $N_2^e \in \mathcal{P}_e = \mathcal{P}_2$ qui satisfait

$$\begin{cases} L_1^e(N_2^e) = 0 = N_2^e(S_1), \\ L_2^e(N_2^e) = 1 = \frac{1}{|e|}\int_e N_2^e(x)\,dx, \\ L_3^e(N_2^e) = 0 = N_2^e(S_2). \end{cases} \quad \text{racines}$$

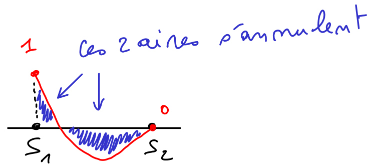
Donc $N_2^e(x) = c\,(x-S_1)(x-S_2)$ pour une certaine constante $c \in \mathbb{R}$. Pour la calculer, utilisons que

$$1 = c \underbrace{\frac{1}{|e|}\int_e \overbrace{(x-S_1)(x-S_2)}^{\text{noté } g(x)\in\mathcal{P}_2}\,dx}_{\overset{\text{Simpson}}{=} \frac{1}{6}\left(\cancel{g(S_1)} + 4g(m) + \cancel{g(S_2)}\right) = \frac{1}{6}\left(-4\frac{|e|^2}{4}\right) = -\frac{|e|^2}{6}} .$$

Donc $c = -\dfrac{6}{|e|^2}$ et $\boxed{N_2^e(x) = -\dfrac{6(x-S_1)(x-S_2)}{|e|^2} \quad (\forall x \in \mathbb{R}).}$

2) Pour $N_1^e \in \mathcal{P}_1$, on a

$$\begin{cases} L_1^e(N_1^e) = 1 = N_1^e(S_1), \\[2mm] L_2^e(N_1^e) = 0 = \dfrac{1}{|e|}\displaystyle\int_e N_1^e(x)\,dx, \\[2mm] L_3^e(N_1^e) = 0 = N_1^e(S_2). \longleftarrow \text{racine} \end{cases}$$

Donc N_1^e est de la forme 

et il a forcément une autre racine $x_* \in]S_1, S_2[$

(sinon on ne pourrait pas avoir $\int_e N_1^e(x)\,dx = 0$!)

D'où

$$\boxed{N_1^e(x) = \frac{\overbrace{(x-x_*)}^{\text{racines}}\overbrace{(x-S_2)}}{\underbrace{(S_1-x_*)(S_1-S_2)}_{\text{pour avoir } N_1^e(S_1)=1}} \quad \forall x \in \mathbb{R}.}$$

3) Pour calculer x_*, on utilise que

$$0 = \frac{1}{|e|}\int_e N_1^e(x)\,dx \underset{\text{Simpson}}{=} \frac{1}{6}\Big(\overbrace{N_1^e(S_1)}^{=1} + 4\underbrace{N_1^e(m)}_{\substack{= \frac{(m-x_*)(m-S_2)}{(S_1-x_*)(S_1-S_2)} \\ = \frac{1}{2}\frac{m-x_*}{S_1-x_*}}} + \cancel{\overbrace{N_1^e(S_2)}^{=0}}\Big).$$

D'où $\quad 0 = 1 + 2\,\dfrac{m - x_*}{S_1 - x_*} \quad \left(\text{où } m = \dfrac{S_1+S_2}{2}\right).$

Un simple calcul montre alors que $\boxed{x_* = \dfrac{2S_1+S_2}{3}\,.}$

4) En utilisant les deux questions précédentes, on conclut que

$$N_1^e(x) = \frac{\left(x - \frac{2S_1+S_2}{3}\right)(x - S_2)}{\left[(S_1-S_2)/3\right] \times (S_1 - S_2)}$$

ie

$$\boxed{N_1^e(x) = \frac{3}{|e|^2}\left(x - \frac{2S_1+S_2}{3}\right)(x - S_2)\,;}$$

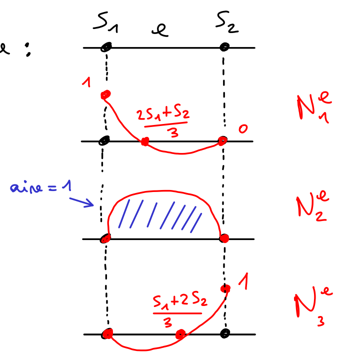
à la question 1, on a aussi vu que

$$\boxed{N_2^e(x) = -\frac{6}{|e|^2}(x - S_1)(x - S_2)\,;}$$

et par symétrie (avec N_1^e), on a finalement

$$\boxed{N_3^e(x) = \frac{3}{|e|^2}(x - S_1)\left(x - \frac{S_1+2S_2}{3}\right)}$$

(pour tout $x \in \mathbb{R}$). C'est-à-dire :



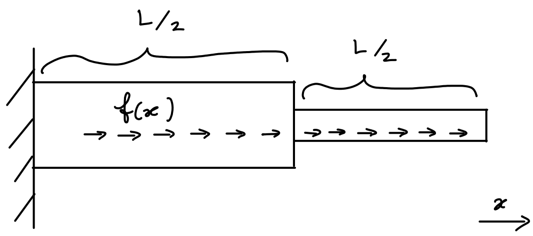
2.1.2 TD no. 2 : algorithme d'assemblage

(Tournez la page.)

TD no. 2

Approximation des EDPs, Printemps 2016-2017

Exercice 1 (Problème de Dirichlet)**.** On considère deux poutres élastiques de même longueur $L/2$ soumises à un effort axial linéique donné $f(x)$:



On cherche à calculer le déplacement à l'équilibre $u(x)$ de la section d'abscisse x. On note

$$c(x) = \begin{cases} c_1 & \text{si } x < \frac{L}{2}, \\ c_2 & \text{sinon,} \end{cases}$$

où $c_i > 0$ est une caractéristique du matériau [1]. On suppose que la première poutre est encastrée en $x = 0$ et que le déplacement de la seconde en $x = L$ est connu et égal à α_L. On admet [2] que $u : [0, L] \to \mathbb{R}$ satisfait

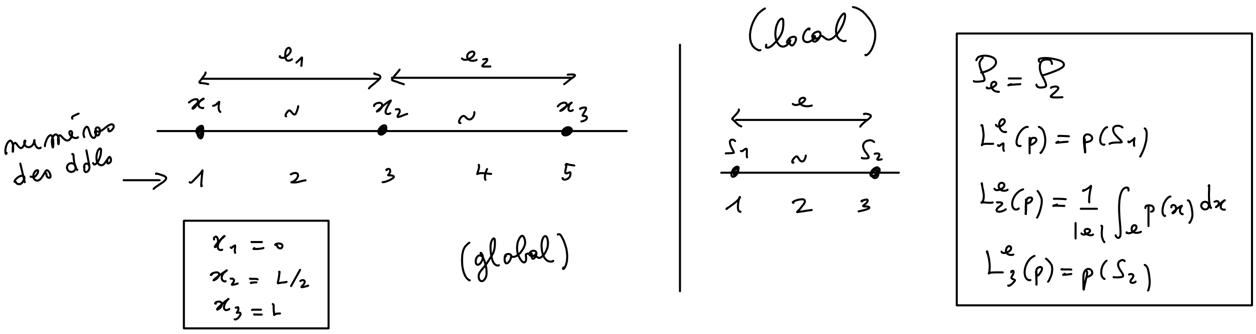
$$\begin{cases} -(c\,u')'(x) = f(x) \quad \forall x \in]0, L[, \\ u(0) = 0 \quad \text{et} \quad u(L) = \alpha_L. \end{cases} \tag{$\mathscr{P}$}$$

Dans la suite, on notera $\Omega =]0, L[$ le domaine physique et $\overline{\Omega} = [0, L]$.

1. On note respectivement Γ_D et Γ_N les parties du bord du domaine où on a une condition de Dirichlet et de Neuman. Précisez Γ_D et Γ_N.
2. Reformulez $(\mathscr{P})$ sous forme variationnelle.
3. On considère l'approximation de $(\mathscr{P})$ par le méthode de Galerkin. Rappelez la forme générale du problème approché en notant V_h l'espace d'approximation.

1. Module de Young × section.
2. Cf. le calcul des structures.

4. On considère le maillage de $\overline{\Omega}$ suivant :



où on muni chaque maille des éléments finis du TD no. 1. On choisit

$$V_h = \Big\{ u_h \in C(\overline{\Omega}) \text{ tel que } u_{h|_e} \in \mathcal{P}_e \text{ pour toute maille } e \Big\}.$$

On note $\{L_1, \dots, L_5\}$ les degrés de liberté globaux et $\{\phi_1, \dots, \phi_5\}$ la base de V_h associée (cf. le cours).

(a) Précisez ce que sont ces ddls.

(b) Dessinez les graphes des ϕ_i en précisant leurs liens avec les N_k^e (du TD no. 1).

5. Ecrivez le problème approché sur la base $\{\phi_1, \dots, \phi_5\}$ (cf. le cours). Vous obtiendrez que

$$A\,U_h = F,$$

où vous préciserez U_h ainsi que $A = (A_{ij})$ et $F = (F_i)$.

6. Pour tout e et tout $k, n \in \{1, 2, 3\}$, on définit :

$$A_{kn}^e = \int_e c(x)\,(N_n^e)'(x)(N_k^e)'(x)\,dx \quad \text{et} \quad F_k^e = \int_e f(x) N_k^e(x)\,dx.$$

(a) Montrez que $A_{43} = A_{21}^{e_2}$ et $F_3 = F_3^{e_1} + F_1^{e_2}$.

(b) Ecrivez de la même façon tous les A_{ij} et F_i en fonction des A_{kn}^e et F_k^e.

7. Nous allons maintenant calculer des approximations des A_{kn}^e et F_k^e à l'aide de la formule de Simpson. Vous noterez m le milieu de e et $|e|$ sa longueur.

(a) Calculez $N_k^e(x)$ pour $k = 1, 2, 3$ et $x = S_1, m, S_2$.

(b) En déduire des approximations de F^e en fonction de $|e|$, $f(S_1)$, $f(m)$ et $f(S_2)$.

(c) Utilisez le même raisonnement pour obtenir des approximations de A^e en fonction de $|e|$, c_1 et c_2.

(d) Ces calculs sont-ils exacts ?

8. En déduire le calcul (approché) de A et F en fonction de $h = |e_1| = |e_2|$, etc.

$$\boxed{\text{Corrigé du TD no. 2}}$$

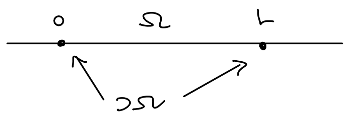
Exercice 1

Dans la suite, on considère le problème aux limites

$$(\mathcal{P}) \quad \begin{cases} -(c\,u')'(x) = f(x) \quad \forall x \in \Omega =]0, L[, \\ u(0) = 0 \quad \text{et} \quad u(L) = \alpha_L, \end{cases}$$

où $u : \overline{\Omega} = [0, L] \longrightarrow \mathbb{R}$ est l'inconnue et le reste est donné (comme dans l'énoncé).

1) Notons $\partial\Omega = \{0, L\}$ le bord du domaine physique Ω.



On appelle « Condition de Dirichlet », une condition au bord sur l'inconnue u, du type

$$\boxed{u = \ldots}$$

En d'autres termes, c'est une condition d'ordre 0 qui ne fait intervenir aucune dérivée de u. Dans le cas contraire, on parle de conditions de Neuman, Robin, etc.

Pour simplifier, on appellera << Condition de Neuman >> tout ces types de conditions au bord faisant intervenir une dérivée de u, comme

$$\boxed{\begin{array}{l} c u' = \dots \quad \text{ou} \quad c u' + b u = \dots, \text{ etc.} \\ \text{(voir le cours pour des exemples plus précis).} \end{array}}$$

On note Γ_D le partie du bord où on a Dirichlet et Γ_N celle où on a Neuman. Pour $(\mathcal{P})$, on a Dirichlet sur tout le bord et donc $\boxed{\Gamma_D = \partial\Omega = \{0, L\} \text{ et } \Gamma_N = \emptyset.}$

2) (Calcul formel)

On suppose que u satisfait $(\mathcal{P})$ et on considère une fonction test arbitraire $\phi : \overline{\Omega} \longrightarrow \mathbb{R}$ telle que

$$\boxed{\phi = 0 \quad \text{sur} \quad \Gamma_D.}$$

(On connaît déjà u sur Γ_D et donc il n'est pas nécessaire de reformuler la condition de Dirichlet avec des fonctions test.)

On a
$$\underbrace{-\int_\Omega (cu')'(x)\,\phi(x)\,dx}_{\overset{\text{IPP}}{=}\ \int_\Omega c(x)\,u'(x)\,\phi'(x)\,dx \,+\, c(0)\,\cancel{u'(0)}\,\phi(0) \,-\, c(L)\,\cancel{u'(L)}\,\phi(L)} = \int_\Omega f(x)\,\phi(x)\,dx .$$

$\phi(0) = 0$, $\phi(L) = 0$ car $\phi = 0$ sur Γ_D

Donc
$$\boxed{\underbrace{\int_\Omega c(x)\,u'(x)\,\phi'(x)\,dx}_{\text{// noté } a(u,\phi)} = \underbrace{\int_\Omega f(x)\,\phi(x)\,dx}_{\text{// noté } \ell(\phi)}}$$

et celà conduit à :

Définition

Notons $V = \{$ fonctions de $\bar{\Omega}$ dans $\mathbb{R}\}$. On dit que u est solution variationnelle de $(\mathcal{P})$ si

$$\boxed{\begin{cases} u \in V, \\ a(u,\phi) = \ell(\phi) \quad \forall \phi \in V \text{ nulle sur } \Gamma_D, \\ u = \alpha \quad \text{sur } \Gamma_D, \end{cases}}$$

où α est la fonction
$$\boxed{\begin{aligned} \alpha : \Gamma_D &\longrightarrow \mathbb{R} . \\ 0 &\longmapsto 0 \\ L &\longmapsto \alpha_L \end{aligned}}$$

3) Étant donné un sous espace vectoriel $V_h \subset V$, de dimension finie, on considère le problème approché qui consiste à chercher u_h tel que

$$
\boxed{(\mathcal{P}_h) \begin{cases} u_h \in V_h, \\ a(u_h, \phi_h) = \ell(\phi_h) \quad \forall \phi_h \in V_h \text{ nulle sur } \Gamma_D, \\ u_h = \alpha \quad \text{sur } \Gamma_D. \end{cases}}
$$

4-a) On choisit V_h comme dans l'énoncé. Les ddls globaux sont les fonctions $L_i : V_h \longrightarrow \mathbb{R}$ $(i = 1, \ldots, 5)$ définies par

$$
\begin{cases} L_1(u_h) = u_h(x_1), \\ L_2(u_h) = \dfrac{1}{|e_1|} \displaystyle\int_{e_1} u_h(x)\, dx, \\ L_3(u_h) = u_h(x_2), \\ L_4(u_h) = \dfrac{1}{|e_2|} \displaystyle\int_{e_2} u_h(x)\, dx, \\ L_5(u_h) = u_h(x_3), \end{cases}
$$

déplacement (approché) du noeud x_1

déplacement moyen (approché) de la première poutre (égale à la maille e_1)

etc.

où $|e|$ désigne la longueur d'une maille e.

b) **Exercice** (à faire à la maison). Montrez que

$\dim V_h = \mathrm{Card}\{L_1, \dots, L_5\}$, où les L_i sont linéaires et

$$(\forall u_h \in V_h) \quad \left[L_1(u_h) = \dots = L_5(u_h) = 0 \Rightarrow u_h = 0 \right].$$

Cela montre que le triplet global $(\bar{\Omega}, V_h, \{L_1, \dots, L_5\})$ est un « élément fini », au sens de la définition du cours. D'où

* pour tout $j = 1, \dots, 5$, il existe un unique $\phi_j \in V_h$

tel que $\boxed{L_i(\phi_j) = \delta_{ij} \quad (\forall i = 1, \dots, 5),}$

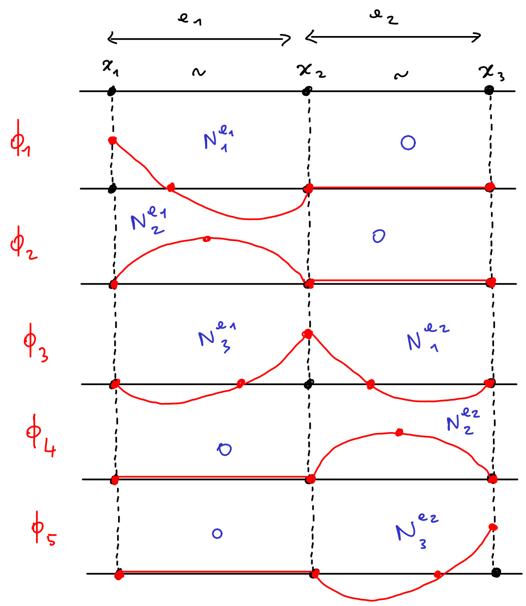
* et la famille $\{\phi_1, \dots, \phi_5\}$ obtenue est une base de V_h

telle que $$\boxed{u_h = \sum_{i=1}^{5} L_i(u_h)\, \phi_i \quad \forall u_h \in V_h.} \qquad (*)$$

Remarque : ces propriétés seront des conséquences de la proposition de l'annexe A du cours sur la dimension 1, une fois que vous aurez établi l'exercice 1 ci-dessus.

Dessinons maintenant la base d'interpolation globale $\{\phi_1, \dots, \phi_5\}$.

On a besoin des bases d'interpolation locales $\{N_1^e, N_2^e, N_3^e\}$ définies pour chaque maille e, comme au TD no. 1 (exercice 3).



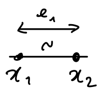
<u>Détails (pour ϕ_1)</u>

On a $\phi_1 \in V_h$ et

$$(1) \quad \begin{cases} L_1(\phi_1) = \phi_1(x_1) = 1, \\ L_2(\phi_1) = \dfrac{1}{|e_1|}\displaystyle\int_{e_1} \phi_1 = 0, \\ L_3(\phi_1) = \phi_1(x_2) = 0, \end{cases}$$

$$L_4(\phi_1) = \frac{1}{|e_2|}\int_{e_2} \phi_1 = 0,$$

$$L_5(\phi_1) = \phi_1(x_3) = 0.$$

Sur l'élément 

, on a l'espace d'interpolation

et les ddls locaux suivants :

$$\mathcal{P}_{e_1} = \mathcal{P}_2 \quad \text{et} \quad \begin{cases} L_1^{e_1}(p) = p(x_1), \\ L_2^{e_1}(p) = \dfrac{1}{|e_1|}\displaystyle\int_{e_1} p(x)\, dx, \\ L_3^{e_1}(p) = p(x_2). \end{cases}$$

Posons $q = \phi_1|_{e_1}$. On a alors $q \in \mathcal{P}_{e_1}$ et

$$\boxed{L_k^{e_1}(q) = \delta_{k1} \qquad \forall k = 1, 2, 3,}$$

d'après (1). L'exercice 1 du TD no. 1 implique donc que

$q = N_1^{e_1}$. On raisonne de même pour justifier tout le dessin.

5) Notons $U_h \in \mathbb{R}^5$ les coordonnées de $u_h \in V_h$ sur la base des ϕ_i. D'après la formule précédente $(*)$, on a

$$\boxed{U_h = \begin{pmatrix} L_1(u_h) \\ \vdots \\ L_5(u_h) \end{pmatrix}.}$$

D'après le cours, on prétends aussi que u_h résoud (P_h) si et seulement si

$$\boxed{AU_h = F}$$

où

$$\boxed{\begin{gathered} A_{ij} = \begin{cases} a(\phi_j, \phi_i) & \text{si } i \notin I_D, \\ \delta_{ij} & \text{si } i \in I_D, \end{cases} \\ F_i = \begin{cases} \ell(\phi_i) & \text{si } i \notin I_D, \\ \alpha(x_{L_i}) & \text{si } i \in I_D, \end{cases} \end{gathered}}$$

avec $\boxed{I_D = \{ i \mid \phi_i \not\equiv 0 \text{ sur } \underbrace{\Gamma_D}_{\substack{\| \\ \{0,L\} \\ \| \\ \{x_1, x_3\}}} \} = \{1,5\}}$

et où

$\boxed{x_{L_i} \text{ désigne « le nœud qui supporte le ddl } L_i \text{ ».}}$

Cela veut dire que

$$\boxed{x_{L_1} = x_1 \quad \text{et} \quad x_{L_5} = x_3}$$

(Car $L_1 : u_h \mapsto u_h(x_1)$ et $L_5 : u_h \mapsto u_h(x_3)$).

⚠ La condition de dirichlet est différente de celle du cours où Γ_D était réduit à un singleton. Ici, on a $\Gamma_D = \{x_1, x_5\}$ d'où l'introduction de la notation x_{L_i}.

<u>preuve</u> (de la nouvelle condition de Dirichlet)

Vérifions qu'on a bien

$$\Big[\underbrace{u_h = \alpha \text{ sur } \Gamma_D}_{(**)}\Big] \iff \Big[L_i(u_h) = \alpha(x_{L_i}) \quad \forall i \in I_D\Big].$$

On a

$$(**) \iff \Big[u_h(0) = \alpha(0) = 0 \quad \text{et} \quad u_h(L) = \alpha(L) = \alpha_L\Big]$$

voir la définition de la fonction $\alpha : \Gamma_D \to \mathbb{R}$ à la question 2

$$\iff \Big[u_h(x_1) = \alpha(x_1) \quad \text{et} \quad u_h(x_3) = \alpha(x_3)\Big]$$

$$\Updownarrow$$

$$\Big[u_h(x_{L_i}) = \alpha(x_{L_i}) \quad \forall i \in I_D = \{1, 5\}\Big]$$

d'après ce qui précède.

∎

6-a) On a

$$A_{43} = a(\phi_3, \phi_4) \qquad (\text{car } i = 4 \notin I_D)$$

$$= \int_\Omega c(x)\, \phi_3'(x)\, \phi_4'(x)\, dx$$

$$= \underbrace{\cancel{\int_{e_1} \text{—}\, dx}}_{\substack{\| \\ 0}} + \int_{e_2} \text{—}\, dx,$$

car $\phi_4 = 0$ sur e_1 et donc $\phi_4' = 0$ sur e_1

où $\int_{e_2} \text{—}\, dx = \int_{e_2} c(x) \left(N_1^{e_2}\right)'(x) \left(N_2^{e_2}\right)'(x)\, dx$

Voir le dessin des ϕ_i

$= A_{21}^{e_2}$. par définition des coefficients A_{km}^e

D'où $\boxed{A_{43} = A_{21}^{e_2}.}$

On montre de la même façon que

$$\boxed{F_3 = F_3^{e_1} + F_1^{e_2}.}$$

b) $\boxed{\text{Formule d'assemblage}}$

Donnons une formule générale pour faire ces calculs, qui aura l'avantage d'être implémentable par ordinateur.

Définition (numéros locaux et globaux)

On définit la « matrice des correspondances » par

$$\text{Elements} = \begin{array}{cc} \begin{pmatrix} 1 & 2 & 3 \\ & & \\ 3 & 4 & 5 \end{pmatrix} & \begin{array}{l} \leftarrow e = e_1 \\ \\ \leftarrow e = e_2 \end{array} \\ \begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ k=1 & 2 & 3 \end{array} & \end{array}$$

où

$$\boxed{i = \text{Elements}(e, k) \iff \left[\phi_i = N_k^e \text{ sur } e\right].}$$

On dit que « le $i^{\text{ème}}$ ddl global corresponds au $k^{\text{ème}}$ ddl local sur e ».

Remarque

On a aussi la formule :

$$\boxed{\left[\phi_i = 0 \text{ sur } e\right] \iff \left[i \text{ n'est dans la ligne } e \text{ de Elements}\right].}$$

Exemple : $\phi_4 = 0$ sur e_1 et on a bien que :

ne contient pas le numéro $i = 4$

$$\text{Elements} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix} \begin{array}{l} \leftarrow e = e_1 \\ \\ \end{array}.$$

Le calcul précédent montre alors que :

Théorème (formel)

Les « coefficients globaux A_{ij} et F_i » s'écrivent comme des sommes de « coefficients locaux A^e_{km} et F^e_k », en remplaçant les numéros globaux i et j par leurs numéros locaux correspondants k et m.

Exemple (Pour F_3)

[$i = 3$ correspond à $k = 1$ sur e_2]

$\Updownarrow$

$3 = \text{Elements}(e_2, 1)$

$$F_3 = F_3^{e_1} + F_1^{e_2}$$

[$i = 3$ correspond à $k = 3$ sur e_1] $\Leftrightarrow$ $3 = \text{Elements}(e_1, 3)$

Exemple (Pour A_{43})

$3 = \text{Elements}(e_2, 1)$

$$A_{43} = A^{e_2}_{21}.$$

$4 = \text{Elements}(e_2, 2)$

⚠ L'élément e_1 ne contribue pas au calcul de A_{43}, car au moins l'une des deux fonctions ϕ_4 ou ϕ_3 est nulle sur e_1. Ici c'est ϕ_4. Cela se voit sur la matrice Elements car le numéro $i = 4$ n'est pas sur la ligne de e_1.

On trouve donc finalement que :

$$\boxed{F = \begin{array}{cl} \begin{pmatrix} \alpha(x_1) \\ F_2^{e_1} \\ F_3^{e_1} + F_1^{e_2} \\ F_2^{e_2} \\ \alpha(x_3) \end{pmatrix} & \begin{array}{l} \longleftarrow I_D \\ \\ \\ \\ \longleftarrow I_D \end{array} \end{array}}$$

et

$$\boxed{A = \begin{array}{cl} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ A_{21}^{e_1} & A_{22}^{e_1} & A_{23}^{e_1} & 0 & 0 \\ A_{31}^{e_1} & A_{32}^{e_1} & A_{33}^{e_1} + A_{11}^{e_2} & A_{12}^{e_2} & A_{13}^{e_2} \\ 0 & 0 & A_{21}^{e_2} & A_{22}^{e_2} & A_{23}^{e_2} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} & \begin{array}{l} \longleftarrow I_D \\ \\ \\ \\ \longleftarrow I_D \end{array} \end{array}.}$$

Algorithme d'assemblage

Voici maintenant un algorithme qui nous permettra d'implémenter A et F par ordinateur, selon la formule du théorème précédent.

Algorithme

Initialisation

$A = 0$;

$F = 0$;

Contribution de l'équation

Pour tout $e = e_1, e_2,$ faire :

Calcul des A^e_{km} et F^e_k ;

Pour tout $k = 1, 2, 3,$ faire :

$i = \text{Elements}(e, k)$;

$F_i = F_i + F^e_k$;

Pour tout $m = 1, 2, 3,$ faire :

$j = \text{Elements}(e, m)$;

$A_{ij} = A_{ij} + A^e_{km}$;

fin

fin

⋮

$\vdots$

fin

Pour tout $i \in I_D$, faire :

$F_i = \alpha(x_{L_i})$;

Pour tout $j = 1, \dots, 5$, faire :

$A_{ij} = \delta_{ij}$;

fin

fin

Condition de Dirichlet

7) Pour calculer les A^e_{lm} et F^e_l, on va utiliser la formule de Simpson :

$$\frac{1}{|e|}\int_e g(x)\,dx = \frac{1}{6}g(S_1) + \frac{2}{3}g(m) + \frac{1}{6}g(S_2) \qquad \forall g \in \mathbb{P}_3 .$$

$m = \frac{S_1+S_2}{2}$

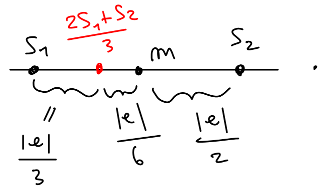
$|e| = S_2 - S_1$

Bien que nous allons devoir intégrer des fonctions non nécessairement polynômiales, nous utiliserons tout de même cette formule pour faire un calcul approché. On admettra que l'erreur commise est négligeable. Pour d'autres formules du même type, voir le TP no. 2 et le cours sur la dimension 2.

a) D'après l'exercice 3 du TD no. 1, on a

$$N_1^e(x) = \frac{3}{|e|^2}\left(x - \frac{2S_1+S_2}{3}\right)(x - S_2)$$

où



Donc

$$\boxed{N_1^e(m) = \frac{3}{|e|^2}\,\frac{|e|}{6}\left(-\frac{|e|}{2}\right) = -\frac{1}{4}.}$$

m est le milieu de e

De plus, on sait déjà que

$$N_1^e(S_1) = L_1^e(N_1^e) = \delta_{11} = 0,$$

etc.

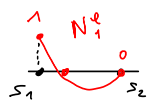
On montre de la même façon que

$$\boxed{N_2^e(m) = \frac{3}{2} \quad \text{et} \quad N_3^e(m) = -\frac{1}{4}.}$$

b) Pour F_1^e, on a

$$F_1^e = \int_e \underbrace{f(x)\,N_1^e(x)}_{\text{noté } g(x)}\,dx$$

$$\approx \frac{|e|}{6}\left(\underbrace{g(S_1)}_{=f(S_1)\underbrace{N_1^e(S_1)}_{=1}} + 4\,\overbrace{g(m)}^{=f(m)\overbrace{N_1^e(m)}^{=-1/4}} + \underbrace{\cancel{g(S_2)}}_{=f(S_2)\underbrace{\cancel{N_1^e(S_2)}}_{=0}}\right)$$



donc

$$\boxed{F_1^e \simeq \frac{|e|}{6}\left(f(S_1) - f(m)\right).}$$

On calcule les autres coefficients de la même façon et on obtient le « second membre élémentaire »

$$\boxed{F^e = (F^e_k)_{k=1,2,3} \simeq \frac{|e|}{6} \begin{pmatrix} f(S_1) - f(m) \\ 6 f(m) \\ f(S_2) - f(m) \end{pmatrix}}$$

c) On a

$$A^e_{km} = \int_e c(x) (N^e_m)'(x) (N^e_k)'(x)\, dx\,.$$

On remarque qu'avec notre choix de maillage, $c = c(x)$ est constante sur les mailles ($= c_i$ sur e_i). Notons c_e cette constante (ie $c_{e_i} = c_i$ pour $i = 1, 2$). Alors,

(2)
$$\begin{aligned} A^e_{km} &= c_e \int_e (N^e_m)'(x) (N^e_k)'(x)\, dx \\ &\underset{(\text{Simpson})}{\simeq} \frac{c_e |e|}{6} \Big(g(S_1) + 4 g(m) + g(S_2) \Big), \end{aligned}$$

où $g(x) = (N^e_m)'(x) (N^e_k)'(x)$.

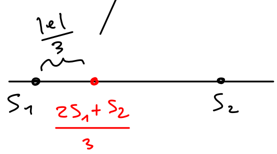
Il nous faut donc calculer les $(N^e_k)'(x)$ pour $x = S_1, m, S_2$.

Détaillons le calcul de $(N^e_1)'(S_1)$.

On a $$\left(N_1^e\right)'(x) = \frac{3}{|e|^2}\left(\left(x - \frac{2S_1+S_2}{2}\right) + (x - S_2)\right).$$

Donc,

$$\boxed{\left(N_1^e\right)'(S_1) = \frac{3}{|e|^2}\left(-\frac{|e|}{3} - |e|\right) = -\frac{4}{|e|}.}$$



On calcule toutes les autres dérivées de la même manière et on trouve que

$$\boxed{\left(\left(N_k^e\right)'(x)\right)_{\substack{k=1,2,3\\ x=S_1,m,S_2}} = \frac{1}{|e|}\begin{array}{cc} \begin{pmatrix} -4 & -1 & 2\\ 6 & 0 & -6\\ -2 & 1 & 4 \end{pmatrix} & \begin{array}{l} \leftarrow k=1\\ \leftarrow 2\\ \leftarrow 3 \end{array}\\ \begin{array}{ccc} \uparrow & \uparrow & \uparrow\\ x=S_1 & m & S_2 \end{array} & \end{array}}$$

Détaillons maintenant le calcul de A_{21}^e, par exemple.

En injectant les résultats ci-dessus dans (2), on trouve que

$$\begin{aligned} A_{21}^e \simeq c_e \frac{|e|}{6}\Big(&\left(N_1^e\right)'(S_1)\left(N_2^e\right)'(S_1)\\ &+ 4\left(N_1^e\right)'(m)\left(N_2^e\right)'(m)\\ &+ \left(N_1^e\right)'(S_2)\left(N_2^e\right)'(S_2)\Big) \end{aligned}$$

$$= c_e \frac{\cancel{|e|}}{6}\,\frac{1}{|e|^{\cancel{2}}}\left((-4)\times 6 + 4\times(-1)\times(0) + 2\times(-6)\right)$$

ie $$\boxed{A_{21}^e \simeq -\frac{6}{|e|}c_e.}$$

On raisonne de la même manière pour les autres coefficients et on trouve la « matrice élémentaire »

$$C_e = \begin{cases} C_1 & \text{si } e = e_1 \\ C_2 & \text{si } e = e_2 \end{cases} \qquad \boxed{A^e = \left(A^e_{km}\right)_{k,m \in \{1,2,3\}} \simeq \frac{C_e}{|e|} \begin{pmatrix} 4 & -6 & 2 \\ -6 & 12 & -6 \\ 2 & -6 & 4 \end{pmatrix}}$$

d) On rappelle que $N^e_k \in \mathcal{P}_e = \mathcal{P}_2$, donc

$$(N^e_m)' (N^e_k)' \in \mathcal{P}_2 \subset \mathcal{P}_3 .$$

On déduit que le calcul de A^e est en fait exact. Celui de F^e peut ne pas l'être si $f \notin \mathcal{P}_1$.

8) En combinant les questions 6 et 7, on trouve que

$$\boxed{F \simeq \begin{pmatrix} \alpha(x_1) = 0 \\ h\, f\left(\frac{x_1 + x_2}{2}\right) \\ \frac{h}{6}\left(2 f(x_2) - f\left(\frac{x_1+x_2}{2}\right) - f\left(\frac{x_2+x_3}{2}\right)\right) \\ h\, f\left(\frac{x_2+x_3}{2}\right) \\ \alpha(x_3) = \alpha_L \end{pmatrix}}$$

et

$$\boxed{A = \frac{1}{h}\begin{pmatrix} h & 0 & 0 & 0 & 0 \\ -6c_1 & 12c_1 & -6c_1 & 0 & 0 \\ 2c_1 & -6c_1 & 4(c_1+c_2) & -6c_2 & 2c_2 \\ 0 & 0 & -6c_2 & 12c_2 & -6c_2 \\ 0 & 0 & 0 & 0 & h \end{pmatrix},}$$

où $h = |e_1| = |e_2|$ $\left(= \frac{L}{2}\right.$ avec deux mailles$\left.\right)$.

2.1.3 Cours d'introduction à matlab

(Tournez la page.)

Quelques notions de base de matlab

Les commandes

Elles doivent être tapées dans la fenêtre *command window* suivies de « entrée ». On les notera par

> *commande à effectuer*

On peut par exemple assigner la valeur numérique 2 à une variable `x`, puis créer une autre variable `y=x+3`, etc. Pour celà, tapez

```
> x=2
> y=x+3
```

Remarque. *On peut donner d'autres noms aux variables (`a`, `b`, `B`, `nom_au_hasard`, etc.) en faisant attention aux majuscules (`b` et `B` ne sont pas les mêmes variables).*

Voici quelques commandes utiles :

`> A=[1 4 6;0 1 2;2 0 2]`	Matrice $A = \begin{pmatrix} 1 & 4 & 6 \\ 0 & 1 & 2 \\ 2 & 0 & 2 \end{pmatrix}$; le « ; » sert à changer de ligne.
`> F=[3;7;5]`	Vecteur $F = \begin{pmatrix} 3 \\ 7 \\ 5 \end{pmatrix}$.
`> A(1,2)`	Coefficient A_{12}.
`> F(1)`	Coefficient F_1.
`> size(F,1)`	Nombre de lignes.
`> size(F,2)`	Nombre de colonnes.
`> A(2,:)`	Sous-vecteur ligne de A correspondant à la ligne $i = 2$; les « : » servent à faire varier les colonnes.
`> A(:,2)`	Sous-vecteur colonne de A correspondant à la colonne $j = 2$.
`> F([1 3])`	Sous-vecteur $\begin{pmatrix} F_1 \\ F_3 \end{pmatrix}$ de F. Le vecteur `[1 3]` précise les numéros des coefficients que l'on veut garder.
`> inv(A)`	Inverse de A.
`> A'`	Transposée.
`> A*F`	Produit matriciel.
`> U=inv(A)*F`	Calcule $U = A^{-1}F$.
`> U=A\F`	Commande équivalente à la précédente.
`> min(F)`	Plus petit coefficient de F.
`> max(F)`	Plus grand coefficient de F.
etc.	Pour en savoir plus, utilisez l'aide de matlab ou simplement un moteur de recherche.

Les fichiers d'un programme

Les programmes sont des ensembles de fichiers (exécutables, de données, etc.) Pour les exécuter, on les regroupe dans un dossier que l'on choisira comme *current directory* de matlab.

Exemple. Créez un dossier `Test` dans votre repertoire personnel et choisissez le comme *current directory* en cliquant sur le bouton $\boxed{...}$ (en haut et à droite de la fenêtre principale). Vous y mettrez tout les fichiers des exemples qui vont suivre.

Les fichiers de données

Ce sont des fichiers `.dat` contenant des données numériques, etc.

Exemple. Créez un nouveau fichier `Elements.dat` en cliquant sur le bouton représentant une page blanche (en haut et à gauche). Ecrivez les valeurs ci-dessous dans ce fichier, en respectant les espaces.

Elements.dat

```
1  2  3
3  4  5
```

Enregistrez et tapez

```
> load Elements.dat
> Elements
```

dans *command window*. Celà définit une matrice

$$\texttt{Elements} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix}.$$

Les scripts

Ce sont des fichiers exécutables constitués d'une liste de commandes.

Exemple. Supposons que l'on veuille calculer la somme

$$\sum_{k=1}^{n} \frac{\cos\left(1+e^k\right)}{k^2},$$

pour $n = 10$. On peut le faire à l'aide du script suivant :

Somme.m

```
n=10;
S=0;
for k=1:n
  S=S+cos(1+exp(k))/k^2;
end
S
% S est la somme voulue.
```

Pour l'exécuter, enregistrez et tapez `> Somme` dans *command window*.

Remarques.
- *Attention aux priorités des opérations : puissance « ^ », division « / », multiplication « * », etc. Si vous n'êtes pas sûr, mettez des parenthèses.*
- *Pour ne pas afficher une ligne calcul, il suffit de mettre « ; » à la fin de cette ligne.*
- *On n'a pas mis de « ; » à l'avant dernière ligne, pour afficher la valeur de `S` à la sortie de la boucle « `for ... end` ».*
- *Les lignes de commentaires doivent être précédées d'un « % », comme à la dernière ligne.*

Les fonctions

Ce sont des fichiers exécutables qui calculent des variables de sortie en fonction de variables d'entrée.

Exemple (Avec une seule sortie)**.** Considérons la fonction

$$f : (x, y) \mapsto x^2 - y^2.$$

On peut la définir avec le programme ci-dessous.

f.m

```
function z=f(x,y)

z=x^2-y^2;
```

Pour calculer $f(x, y)$ pour $x = 3$ et $y = 1$, on utilise la commande `> f(3,1)`

Remarques.

- *Le nom du fichier*

  $$\underbrace{f}_{nom}.m$$

  doit être le même que celui de la fonction

  $$\texttt{function y=}\underbrace{\texttt{f}}_{nom}\texttt{(x)}$$

- *Les variables des fonctions sont locales. Par exemple, tapez*

```
> x=10
> f(3,1)
> x
```

  et remarquez que l'exécution de `f.m` *n'a pas changé la valeur de* `x` *(en* $\texttt{x} = 3$*).*
- *Ces variables peuvent être aussi des matrices, etc., du moment que les calculs sont cohérents.*

Exemple (Avec plusieurs sorties)**.** Considérons la fonction

$$g : (x, y) \mapsto (xy, x + y).$$

On peut la définir avec le programme ci-dessous.

g.m

```
function [u,v]=g(x,y)

u=x*y;
v=x+y;
```

Prenons encore $x = 3$ et $y = 1$, par exemple. Maintenant, la commande

```
> g(3,1)
```

ne donne que la valeur de la première sortie. Pour avoir toutes les sorties, il faut considérer des variables intermédiaires, par exemple `a` et `b`, et procéder comme ci-dessous.

```
> [a,b]=g(3,1)
> a
> b
```

Les boucles et instructions

Les deux principaux types de boucles sont « `for ... end` » et « `while ... end` ». On peut aussi considérer plusieurs options avec les instructions « `if ... end` », « `if ... else ... end` » et « `if ... elseif ... end` ».

Voici ci-dessous un exemple d'instruction ; cf. le dernier programme `Graphe.m` pour un exemple de boucle `while`.

Exemple. Considérons la fonction $c = c(x)$ définie par

$$c(x) = \begin{cases} 8 \times 10^7 & \text{si } x < 1/2, \\ 10^7 & \text{si } x \geq 1/2. \end{cases}$$

On la définit sur matlab avec le programme ci-dessous.

c.m

```
function y=c(x)

if (x<1/2)
  y=8*10^7;
else
  y=10^7;
end
```

Les dessins

Pour dessiner le point d'abscisse $x = 2$ et d'ordonnée $y = 3$, on utilise la commande

```
> plot(2,3)
```

On peut rajouter des options ; par exemple,

```
> plot(2,3,'b*')
```

donne le motif `*` avec la couleur `b`=*blue*. Pour les TPs, il sera utile de savoir dessiner le graphe d'une fonction point par point.

Exemple. Le graphe de $x \mapsto c(x)$, pour $x \in [0, 1]$, peut se dessiner avec un pas de 0.01 à l'aide du script ci-dessous.

Graphe.m

```
clf; % efface les figures précédentes
hold on; % dessine tout les points sur la même figure
grid on; % rajoute une grille
pas=0.01;
x=0;
while (x<=1)
  plot(x,c(x),'bo');
  x=x+pas;
end
title('Graphe de y=c(x)');xlabel('x'); ylabel('y');
```

Tapez `> Graphe` pour obtenir votre dessin.

2.1.4 TPs no. 1 et 2 : programmation

(Tournez la page.)

TP no. 1

Approximation des EDPs, Printemps 2016-2017

Table des matières

1 Problématique

On reprends l'exercice 1 du TD no. 2 et on se propose de faire un programme matlab pour calculer u_h. On prendra les valeurs numériques suivantes :

$$L = 1, \quad c_1 = 8 \times 10^7, \quad c_2 = 10^7, \quad f(x) = 9 \times 10^5, \quad \alpha_L = 10^{-2}$$

(L en mètres, c_i en Pascal$\times$mètres^2, f en Newton/mètres et α_L en mètres).

2 Assemblage et résolution de $AU_h = F$

1. Créez un dossier `TP1`, que vous choisirez comme *current directory* de matlab et dans lequel vous placerez vos fichiers `.m` et `.dat`.
2. Créez des fichiers de données `Elements.dat` et `ID.dat`, dans lesquels vous stockerez les matrices

$$\texttt{Elements} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix} \quad \text{et} \quad \texttt{ID} = \begin{pmatrix} 1 \\ 5 \end{pmatrix}.$$

Vérification. Vérifiez que vos fichiers sont corrects en tapant

```
> load Elements.dat;Elements
> load ID.dat;ID
```

3. Définissez les fonctions `c.m`, `f.m` et `alpha.m` suivantes :

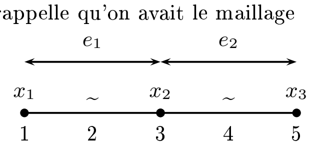
$$\begin{aligned} c : x &\mapsto \begin{cases} 8\times 10^7 & \text{si } x < 1/2, \\ 10^7 & \text{sinon,} \end{cases} \\ f : x &\mapsto 9\times 10^5, \\ \alpha : x &\mapsto \begin{cases} 0 & \text{si } x = 0, \\ 10^{-2} & \text{si } x = 1. \end{cases} \end{aligned}$$

   Vérification. Testez vos fonctions pour différentes valeurs de x.

4. Créez un fichier `Noeuds.dat` dans lequel vous stockerez le vecteur suivant

$$\texttt{Noeuds} = \begin{pmatrix} x_{L_1} \\ \vdots \\ x_{L_5} \end{pmatrix} = \begin{pmatrix} 0 \\ 0.25 \\ 0.5 \\ 0.75 \\ 1 \end{pmatrix}.$$

   Commentaires. C'est le vecteur des noeuds x_{L_i} qui supportent les ddls globaux. On rappelle qu'on avait le maillage



   avec

$$\begin{cases} L_1(u_h) = u_h(x_1), \\ L_2(u_h) = \frac{1}{|e_1|}\int_{e_1} u_h(x)\,dx, \\ L_3(u_h) = u_h(x_2), \\ L_4(u_h) = \frac{1}{|e_2|}\int_{e_2} u_h(x)\,dx, \\ L_5(u_h) = u_h(x_3). \end{cases}$$

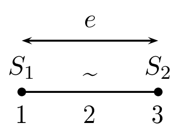
   D'où $x_{L_1} = x_1 = 0$, $x_{L_3} = x_2 = 0.5$ et $x_{L_5} = x_3 = 1$. Il est aussi pratique d'associer des noeuds à L_2 et L_4 pour avoir la formule

$$x_{L_i} = \texttt{Noeuds(i);}$$

   mais les valeurs de x_{L_2} et x_{L_4} n'interviendront pas dans les calculs et on peut donc les choisir au hasard. Prenons par exemple les milieux des mailles, ce qui donne $x_{L_2} = 0.25$ et $x_{L_4} = 0.75$.

   Vérification. Tapez `> load Noeuds.dat;Noeuds`

5. Pour chaque élément



on note

$$\texttt{Noeuds\_loc} = \begin{pmatrix} x_{L_1^e} \\ x_{L_2^e} \\ x_{L_3^e} \end{pmatrix} = \begin{pmatrix} S_1 \\ m \\ S_2 \end{pmatrix}$$

(où m est le milieu de e).

Remarque. C'est le vecteur des noeuds qui supportent les ddls locaux.

Créez et complétez la fonction ci-dessous qui calcule A^e et F^e en fonction de ce vecteur. Vous noterez `h` la longueur de e.

AeFe.m

```
function [Ae,Fe]=AeFe(Noeuds_loc)

S1=% A COMPLETER
S2=% A COMPLETER
m=(S1+S2)/2;
h=% A COMPLETER
ci=c(m);% calcule le bon ci

Ae=% A COMPLETER
Fe=% A COMPLETER
```

Rappels.

$$A^{e_i} = \frac{c_i}{|e_i|} \begin{pmatrix} 4 & -6 & 2 \\ -6 & 12 & -6 \\ 2 & -6 & 4 \end{pmatrix} \quad \text{et} \quad F^e = \frac{|e|}{6} \begin{pmatrix} f(S_1) - f(m) \\ 6\, f(m) \\ f(S_2) - f(m) \end{pmatrix},$$

où $|e| = S_2 - S_1$.

Vérification. Si $e = e_1$, on a

$$\texttt{Noeuds\_loc} = \begin{pmatrix} x_1 \\ \frac{x_1+x_2}{2} \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.25 \\ 0.5 \end{pmatrix}.$$

Calculez A^{e_1} et F^{e_1} en tapant

```
> [Ae,Fe]=AeFe([0;0.25;0.5])
```

Vous devez obtenir

$$A^{e_1} \simeq 10^9 \times \begin{pmatrix} 0.64 & -0.96 & 0.32 \\ -0.96 & 1.92 & -0.96 \\ 0.32 & -0.96 & 0.64 \end{pmatrix}, \quad F^{e_1} \simeq \begin{pmatrix} 0 \\ 450000 \\ 0 \end{pmatrix}.$$

6. Pour continuer, vous devez apprendre à calculer `Noeuds_loc` en fonction de e. Pour celà, lisez attentivement la remarque ci-dessous.

Remarque. Pour toute maille e et numéro local k, on a

$$\texttt{Noeuds\_loc(k)} = x_{L_k^e} = x_{L_i} = \texttt{Noeuds(i)},$$

où i est le numéro global correspondant à k sur e. D'où

$$\texttt{Noeuds\_loc} = \begin{pmatrix} \texttt{Noeuds(Elements(e,1))} \\ \texttt{Noeuds(Elements(e,2))} \\ \texttt{Noeuds(Elements(e,3))} \end{pmatrix}.$$

Sur matlab, ce calcul pour $e = e_1$ se fait par la commande suivante :

```
> Noeuds_loc=Noeuds(Elements(1,:))
```

Tapez la commande ci-dessus sur *command window* et vérifiez que vous obtenez les bons noeuds.

7. Créez et complétez le script ci-dessous qui assemble A et F (en considérant seulement la contribution de l'équation).

Principal.m

```
%% Initialisation

clear all;% des variables
clc;% de command window

%% Génération du maillage

load Noeuds.dat;
load Elements.dat;
load ID.dat;

%% Initialisation de A et F

N=size(Noeuds,1);% nombre de ddls globaux
A=zeros(N,N);% matrice nulle de taille N*N
F=zeros(N,1);% vecteur nul de taille N*1

%% Assemblage : contribution de l'équation

Nelem=size(Elements,1);% nombre d'éléments
N_loc=size(Elements,2);% nombre de ddls locaux

for e=1:Nelem
       Noeuds_loc=% A COMPLETER
       [Ae,Fe]=% A COMPLETER
       for k=1:N_loc
              % A COMPLETER
              % A COMPLETER
              for n=1:N_loc
                     % A COMPLETER
                     % A COMPLETER
              end
       end
end
```

Vérification. Tapez

```
> Principal
> A
> F
```

et vérifiez que

$$A \simeq 10^9 \times \begin{pmatrix} 0.64 & -0.96 & 0.32 & 0 & 0 \\ -0.96 & 1.92 & -0.96 & 0 & 0 \\ 0.32 & -0.96 & 0.72 & -0.12 & 0.04 \\ 0 & 0 & -0.12 & 0.24 & -0.12 \\ 0 & 0 & 0.04 & -0.12 & 0.08 \end{pmatrix}$$

et

$$F \simeq 10^5 \begin{pmatrix} 0 \\ 450000 \\ 0 \\ 450000 \\ 0 \end{pmatrix}.$$

8. Rajoutez la contribution de la condition de Dirichlet.

Principal.m (suite)

```
⋮

%% Assemblage : contribution de Dirichlet

CardID=size(ID,1);% cardinal de ID

for temp=1:CardID% temp est un indice muet, il n'est pas dans ID
        i=ID(temp);% i est l'indice général de ID
        % A COMPLETER
        for j=1:N
                A(i,j)=0;
        end
        % A COMPLETER
end
```

Vérification. Exécutez votre programme et vérifiez que les lignes de I_D sont correctes.

9. Résolvez le système $A\,U_h = F$ en complétant la ligne de code ci-dessous

Principal.m (fin)

```
⋮
```

```
%% Résolution du problème matriciel

Uh=% A COMPLETER
```

et en tapant `> Principal;Uh`

Vérification. Vous devez trouver

$$U_h \simeq \begin{pmatrix} 0 \\ 0.002 \\ 0.0036 \\ 0.0087 \\ 0.01 \end{pmatrix}.$$

3 Dessin de u_h par interpolation

3.1 Préliminaires

Vous avez calculé les $L_i(u_h) \simeq \mathtt{Uh}(i)$. Pour en déduire les déplacements de toutes les sections de poutre d'abscisses $x \neq x_i$, on va faire une interpolation. Commencez d'abord par lire attentivement ce qui suit.

Comment calculez $u_h(x)$ pour tout $x \in \overline{\Omega}$?

On a

$$u_h(x) = \sum_{i=1}^{5} L_i(u_h)\, \phi_i(x),$$

où

$$\begin{pmatrix} L_1(u_h) \\ \vdots \\ L_5(u_h) \end{pmatrix} = \underbrace{U_h}_{\text{“ au tableau ”}} \simeq \underbrace{\mathtt{Uh}}_{\text{“ sur matlab ”}} = \begin{pmatrix} \mathtt{Uh(1)} \\ \vdots \\ \mathtt{Uh(5)} \end{pmatrix}.$$

Par suite,

$$u_h(x) \simeq \sum_{i=1}^{5} \mathtt{Uh(i)}\ \phi_i(x).$$

On peut réécrire cette formule sur e en utilisant que

$$\big[\phi_i = N_k^e \text{ sur } e\big] \Leftrightarrow \big[i = \mathtt{Elements(e,k)}\big].$$

On obtient finalement

$$u_h(x) \simeq \sum_{k=1}^{3} \mathtt{Uh(Elements(e,k))}\ N_k^e(x) \quad \forall x \in e, \tag{1}$$

où on rappelle que

$$\begin{cases} N_1^e(x) = \frac{3}{|e|^2}\left(x - \frac{2\,S_1+S_2}{3}\right)(x - S_2)\,, \\ N_2^e(x) = -\frac{6}{|e|^2}\,(x - S_1)\,(x - S_2)\,, \\ N_3^e(x) = \frac{3}{|e|^2}\,(x - S_1)\left(x - \frac{S_1+2\,S_2}{3}\right). \end{cases}$$

Sur matlab, on notera Uh_loc et Ne les variables correspondantes aux vecteurs ci-dessous :

$$\texttt{Uh\_loc} = \begin{pmatrix} \texttt{Uh(Elements(e,1))} \\ \texttt{Uh(Elements(e,2))} \\ \texttt{Uh(Elements(e,3))} \end{pmatrix} \quad \text{et} \quad \texttt{Ne} = \begin{pmatrix} N_1^e(x) \\ N_2^e(x) \\ N_3^e(x) \end{pmatrix}.$$

Celà permettra d'écrire (1) comme le produit scalaire suivant :

$$u_h(x) \simeq \underbrace{\texttt{Uh\_loc} \cdot \texttt{Ne}}_{\text{“ au tableau ”}} = \underbrace{\texttt{Uh\_loc'*Ne}}_{\text{“ sur matlab ”}}. \tag{2}$$

3.2 Un exemple de calcul

Calculer $u_h(0.17)$ sur *command window* en suivant les étapes ci-dessous.

1. Calculez le vecteur Uh_loc pour $e = e_1 \ni 0.17$ en tapant

```
> Uh_loc=Uh(Elements(1,:))
```

2. Pour calculer le vecteur Ne en $x = 0.17$, on a besoin des noeuds et de la longueur h de $e = e_1$. Ces derniers se calculent avec les commandes suivantes :

```
> Noeuds_loc=Noeuds(Elements(1,:))
> S1=min(Noeuds_loc)
> S2=max(Noeuds_loc)
> h=S2-S1
```

3. En déduire le calcul de Ne (en $x = 0.17$) en tapant

```
> x=0.17
> Ne=zeros(3,1)
> Ne(1)=% A COMPLETER
> Ne(2)=% A COMPLETER
> Ne(3)=% A COMPLETER
```

   où vous complèterez les lignes manquantes.

4. Calculez $u_h(0.17)$ par (2) en tapant

```
> Uh_loc'*Ne
```

   Vous devez trouver $u_h(0.17) \simeq 0.0015$.

3.3 Dessin de u_h

1. Créez et complétez le script ci-dessous. Il réitère les calculs précédents pour dessiner le graphe de $x \mapsto u_h(x)$, avec un pas de 0.01.

Dessin.m

```
clf;
hold on;
grid on;

pas=0.01;

for e=1:Nelem

       Uh_loc=% A COMPLETER

       Noeuds_loc=% A COMPLETER
       S1=% A COMPLETER
       S2=% A COMPLETER
       h=% A COMPLETER

       x=S1;
       while (x<=S2)
             Ne=zeros(3,1);
             Ne(1)=% A COMPLETER
             Ne(2)=% A COMPLETER
             Ne(3)=% A COMPLETER
             % A COMPLETER
             x=x+pas;
       end
end

% titre, légende, etc.

legend('Graphe de x -> uh(x)');
title('Solution approchee');
xlabel('x'); ylabel('y');
```

Vérification. Tapez `> Dessin`

2. Complétez le programme principal :

Principal.m (avec le dessin)

```
⋮

%% Dessin de la solution approchée

Dessin;
```

Votre programme est terminé.

Corrigé du TP no. 1

Les programmes sont donnés à la fin de ce corrigé.

Ci-dessous, on commente seulement certaines questions.

(Pour toutes les autres, voir les programmes à la fin.)

Section 2 : assemblage et ...

3) Pour tester vos fonctions, tapez par exemple

```
> c(0.2)
> c(0,7)
```

(dans command window), etc.

4) La variable Noeuds_loc est la variable d'entrée de la fonction AeFe.m. C'est comme une « variable muette », dont la valeur n'est précisée que lorsqu'on fait appel à cette fonction.

7-8) Vous devez utiliser l'algorithme d'assemblage que l'on a vu au TD no. 2.

Pour vérifier que votre boucle de Dirichlet est correcte, vous devez taper

```
> Principal
> A(1,:)
> A(5,:)
> F(1)
> F(5)
```

dans command window. Vous devez obtenir

$$\begin{cases} A(1,:) = (1 \quad 0 \quad 0 \quad 0 \quad 0), \\ A(5,:) = (0 \quad 0 \quad 0 \quad 0 \quad 1), \\ F(1) = 0, \\ F(5) = 0.01. \end{cases}$$

$A_{ij} = \delta_{ij}$ car $i \in I_D$

$F_i = \alpha(x_{L_i})$ car $i \in I_D$

⚠ Si vous tapez $\boxed{> A}$ et $\boxed{> F}$, vous obtiendrez une matrice et un vecteur factorisés par un grand nombre, d'où le 0.0000 en A_{11} (qui vaut 10^{-9}), etc.

9) Le programme Principal.m assemble la matrice A

et le vecteur F de l'exercice 1 du TD no. 2. Il résoud aussi

$$\boxed{AU_h = F,}$$

où l'inconnue $U_h \in \mathbb{R}^5$ est le vecteur des coordonnées de la fonction $u_h : \bar{\Omega} \to \mathbb{R}$ sur la base d'interpolation ; c'est-à-dire, c'est le vecteur

$$\boxed{U_h = \begin{pmatrix} L_1(u_h) \\ \vdots \\ L_5(u_h) \end{pmatrix} = \begin{pmatrix} u_h(x_1) \\ \frac{1}{|e_1|}\int_{e_1} u_h(x)\,dx \\ u_h(x_2) \\ \frac{1}{|e_2|}\int_{e_2} u_h(x)\,dx \\ u_h(x_3) \end{pmatrix}.}$$

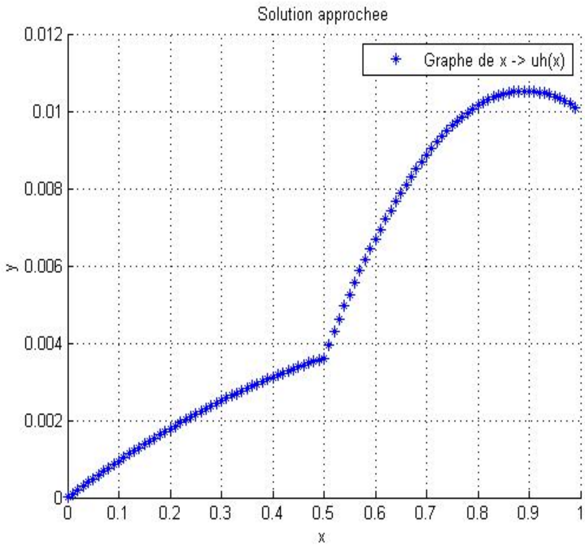
(Voir le TD no. 2)

La reconstruction de la fonction $u_h : \bar{\Omega} \to \mathbb{R}$ à partir de ces coordonnées est expliquée dans la suite du TP (voir l'énoncé).

Section 3 : dessin de ...

A la fin du TP, vous devez obtenir le déplacement

$u_h : \bar{\Omega} \to \mathbb{R}$, dont le graphe est dessiné ci-dessous.



Programmes corrigés du TP no. 1

Scripts

Principal.m

```
%% Initialisation

clear all;% des variables
clc;% de command window

%% Generation du maillage

load Noeuds.dat;
load Elements.dat;
load ID.dat;

%% Initialisation de A et F

N=size(Noeuds,1);% nombre de ddls globaux
A=zeros(N,N);% matrice nulle de taille N*N
F=zeros(N,1);% vecteur nul de taille N*1

%% Assemblage : contribution de l'equation

Nelem=size(Elements,1);% nombre d'elements
N_loc=size(Elements,2);% nombre de ddls locaux

for e=1:Nelem
    Noeuds_loc=Noeuds(Elements(e,:));
    [Ae,Fe]=AeFe(Noeuds_loc);
    for k=1:N_loc
        i=Elements(e,k);
        F(i)=F(i)+Fe(k);
        for n=1:N_loc
            j=Elements(e,n);
            A(i,j)=A(i,j)+Ae(k,n);
        end
    end
end

%% Assemblage : contribution de Dirichlet

CardID=size(ID,1);% cardinal de ID

for temp=1:CardID% temps est un indice muet, il n'est pas dans ID
    i=ID(temp);% i est l'indice general de ID
    F(i)=alpha(Noeuds(i));% c'est le noeud qui supporte Li
    for j=1:N
        A(i,j)=0;
    end
    A(i,i)=1;
end

%% Resolution du probleme matriciel

Uh=A\F;

%% Dessin de la solution approchee

Dessin;
```

Dessin.m

```
clf;
hold on;
grid on;

pas=0.01;

for e=1:Nelem

    Uh_loc=Uh(Elements(e,:));

    Noeuds_loc=Noeuds(Elements(e,:));
    S1=min(Noeuds_loc);
    S2=max(Noeuds_loc);
    h=S2-S1;

    x=S1;
    while (x<=S2)
        Ne=zeros(3,1);
        Ne(1)=((x-(2*S1+S2)/3)*(x-S2)*3)/(h^2);
        Ne(2)=-((x-S1)*(x-S2)*6)/(h^2);
        Ne(3)=((x-S1)*(x-(S1+2*S2)/3)*3)/(h^2);
        uh=Uh_loc'*Ne;
        plot(x,uh,'b*');
        x=x+pas;
    end
end

% titre, legende, etc.

legend('Graphe de x -> uh(x)');
title('Solution approchee');
xlabel('x');ylabel('y');
```

Fonctions

AeFe.m

```
function [Ae,Fe]=AeFe(Noeuds_loc)

S1=min(Noeuds_loc);
S2=max(Noeuds_loc);
m=(S1+S2)/2;
h=S2-S1;% longueur de e
ci=c(m);% calcule le bon ci

Ae=[4 -6 2;-6 12 -6;2 -6 4]*ci/h;
Fe=[f(S1)-f(m);6*f(m);f(S2)-f(m)]*h/6;
```

alpha.m

```
function y=alpha(x)

if (x==0)
    y=0;
elseif (x==1)
    y=10^(-2);
end
```

c.m

```
function y=c(x)

if (x<0.5)
    y=8*10^7;
else
    y=10^7;
end
```

f.m

```
function y=f(x)

y=9*10^5;
```

Fichiers de données

Noeuds.dat

```
0
0.25
0.5
0.75
1
```

Elements.dat

```
1 2 3
3 4 5
```

ID.dat

```
1
5
```

TP no. 2

Approximation des EDPs, Printemps 2016-2017

Dans le TP précédent, on a utilisé les formules de A^e et F^e que l'on avait calculées « à la main » en TD. La première partie de ce TP explique comment faire ces calculs directement par ordinateur et la deuxième partie est une introduction aux maillages.

Table des matières

1 Calcul pratique de A^e et F^e

On reprends le problème (de l'exercice 1 du TD no. 2) :

$$\begin{cases} -(c\,u')' = f, & \text{dans }]0, L[, \\ u(0) = 0 \quad \text{et} & u(L) = \alpha_L, \end{cases} \tag{$\mathscr{P}$}$$

avec

$$\begin{cases} L = 1, \\ c_1 = 8 \times 10^7, \\ c_2 = 10^7, \\ f(x) = 9 \times 10^5, \\ \alpha_L = 10^{-2}. \end{cases}$$

1.1 Préliminaires

On considère l'élément fini de la figure 1 et on note $\{N_1^e, \ldots, N_3^e\}$ sa base

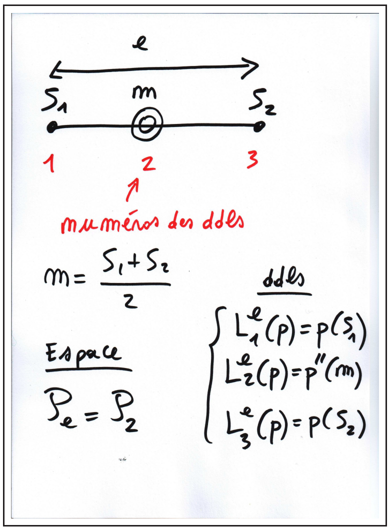


FIGURE 1 – Choix des éléments finis

d'interpolation.[1] On rappelle que

$$L_k^e(N_n^e) = \delta_{kn}.$$

On rappelle aussi que pour tout $p \in \mathcal{P}_e$ et tout $x \in e$,

$$p(x) = \sum_{k=1}^{3} L_k^e(p)\, N_k^e(x).$$

1. En déduire que
$$\begin{cases} 1 = N_1^e(x) + \dots, \\ x = \dots, \\ x^2 = \dots, \end{cases}$$
où vous complèterez les ...
2. Ecrivez ce système sous la forme
$$\begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} = \mathtt{M} \begin{pmatrix} N_1^e(x) \\ N_2^e(x) \\ N_3^e(x) \end{pmatrix}. \tag{1}$$
où vous préciserez la matrice M.

    Rappels. M est la matrice de passage entre la base canonique et la base d'interpolation de $\mathcal{P}_e$.

1. **A faire à la maison :** vérifiez que $(e, \mathcal{P}_e, \{L_1^e, \dots, L_3^e\})$ est bien un élément fini.

3. En déduire que

$$\begin{pmatrix} * \\ * \\ * \end{pmatrix} = \mathtt{M} \begin{pmatrix} (N_1^e)'(x) \\ (N_2^e)'(x) \\ (N_3^e)'(x) \end{pmatrix},$$

où vous préciserez les coefficients $*$.

4. On se propose de calculer $N_k^e(x)$ et $(N_k^e)'(x)$ par matlab en un certain point x. Prenons par exemple

$$S_1 = 0, \quad S_2 = 0.5 \quad \text{et} \quad x = 0.1.$$

(a) Avec ce x fixé, on notera `Nc` et `Ne` les vecteurs

$$\mathtt{Nc} = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} \quad \text{et} \quad \mathtt{Ne} = \begin{pmatrix} N_1^e(x) \\ N_2^e(x) \\ N_3^e(x) \end{pmatrix}.$$

Remarque. Ces vecteurs correspondent à la base canonique et la base d'interpolation dans la formule (1).

Utilisez les commandes suivantes pour calculer `Ne` :

```
> S1=0
> S2=0.5
> x=0.1
> M=% A COMPLETER
> Nc=[1;x;x^2]
> Ne=M\Nc
```

Que vaut $N_2^e(0.1)$?

(b) Les vecteurs des dérivées se noteront ainsi :

$$\mathtt{dNc} = \begin{pmatrix} 0 \\ 1 \\ 2\,x \end{pmatrix} \quad \text{et} \quad \mathtt{dNe} = \begin{pmatrix} (N_1^e)'(x) \\ (N_2^e)'(x) \\ (N_3^e)'(x) \end{pmatrix}.$$

Utilisez les commandes suivantes pour calculer `dNe` :

```
> dNc=[0;1;2*x]
> dNe=% A COMPLETER
```

Que vaut $(N_3^e)'(0.1)$?

1.2 Intégration des coefficients élémentaires

On rappelle que les coefficients de la matrice $A^e \in \mathbb{R}^{3\times 3}$ et du vecteur $F^e \in \mathbb{R}^3$ sont définis par :

$$A^e_{kn} = \int_e c(x)\,(N^e_n)'\,(x)\,(N^e_k)'\,(x)\,dx, \quad F^e_k = \int_e f(x)\,N^e_k(x)\,dx.$$

On approchera ces intégrales avec la formule de Gauss à 2 points suivante :

$$\frac{1}{|e|}\int_e g(x)\,dx = \sum_{i=1}^{2} \omega_i\, g(\xi_i) \quad \forall g \in \mathcal{P}_3,$$

où

$$\begin{cases} \xi_1 = m - \frac{|e|}{2}\frac{\sqrt{3}}{3}, \\ \xi_2 = m + \frac{|e|}{2}\frac{\sqrt{3}}{3}, \\ \omega_1 = \omega_2 = \frac{1}{2}; \end{cases}$$

voir la figure 2.

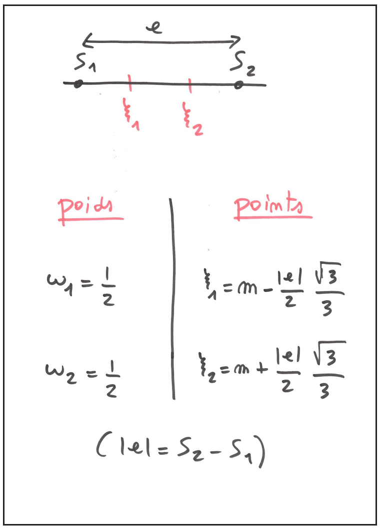


FIGURE 2 – Points et poids de la formule de Gauss à 2 points

Commentaire. Cette formule ressemble à celle de Simpson, mais avec d'autres ξ_i et ω_i. Ce type de formule s'appelle une « formule d'intégration numérique ». Plus de détails seront donnés dans le cours sur la dimension 2 ; pour l'instant, disons seulement que les ξ_i s'appellent les « points d'intégration » et les ω_i les « poids ».

Ceci conduit aux approximations suivantes :

$$\begin{aligned} A^e_{kn} &\simeq |e| \sum_{i=1}^{2} \omega_i \, c(\xi_i) \, (N^e_n)' \, (\xi_i) \, (N^e_k)' \, (\xi_i), \\ F^e_k &\simeq |e| \sum_{i=1}^{2} \omega_i \, f(\xi_i) \, N^e_k(\xi_i). \end{aligned} \tag{2}$$

1. Créez des sous-dossiers

   `.../TP2/Calcul_pratique`.

   Vous copier/collerez les fichiers de `TP1` dans `Calcul_pratique`, que vous choisirez comme *current directory* de matlab.

2. On rappelle que `AeFe.m` définit la fonction

   $$\texttt{Noeuds\_loc} \mapsto \texttt{[Ae,Fe]},$$

   où

   $$\texttt{Noeuds\_loc} = \begin{pmatrix} x_{L^e_1} \\ x_{L^e_2} \\ x_{L^e_3} \end{pmatrix}.$$

   On rappelle que $x_{L^e_k}$ est le noeud qui supporte le ddl L^e_k. Avec les éléments finis de la figure 1, on a

   $$\texttt{Noeuds\_loc} = \begin{pmatrix} S_1 \\ m \\ S_2 \end{pmatrix}.$$

   Modifiez `AeFe.m`, comme ci-dessous, pour que les variables de sortie soient la matrice `Ae` et le vecteur `Fe` définis par la formule (2).

AeFe.m

```
function [Ae,Fe]=AeFe(Noeuds_loc)

%% Noeuds et longueur de l'élément

S1=min(Noeuds_loc);
S2=max(Noeuds_loc);
m=(S1+S2)/2;
h=S2-S1;% longueur

%% Points et poids d'intégration

xi=zeros(2,1);
xi(1)=% A COMPLETER
xi(2)=% A COMPLETER
```

```
omega=zeros(2,1);
omega(1)=% A COMPLETER
omega(2)=% A COMPLETER

%% Matrice de passage

M=% A COMPLETER

%% Intégration des coefficients élémentaires

N_loc=size(Noeuds_loc,1);% nombre de ddls locaux
Ae=zeros(N_loc,N_loc);
Fe=zeros(N_loc,1);

Nint=size(xi,1);% nombre de points d'intégration
for i=1:Nint

        x=xi(i);% point d'intégration
        Nc=% A COMPLETER
        dNc=% A COMPLETER
        Ne=% A COMPLETER
        dNe=% A COMPLETER

        for k=1:N_loc
              Fe(k)=% A COMPLETER
              for n=1:N_loc
                    Ae(k,n)=% A COMPLETER
              end
        end
end
```

3. Si $S_1 = 0.5$ et $S_2 = 1$, on doit obtenir

$$\texttt{Ae} \simeq 10^7 \times \begin{pmatrix} 2 & 0 & -2 \\ 0 & 0.010417 & 0 \\ -2 & 0 & 2 \end{pmatrix},$$

et

$$\texttt{Fe} \simeq 10^5 \times \begin{pmatrix} 2.25 \\ -0.09375 \\ 2.25 \end{pmatrix}.$$

Tapez `> [Ae,Fe]=AeFe([0.5;0.75;1])` et vérifiez que votre programme donne les bons résultats.

1.3 Résolution

On considère l'approximation du problème ($\mathscr{P}$) avec le maillage de la figure 3 (avec des éléments de même longueurs). Dessinez le graphe du déplacement approché $x \mapsto u_h(x)$. Pour celà :

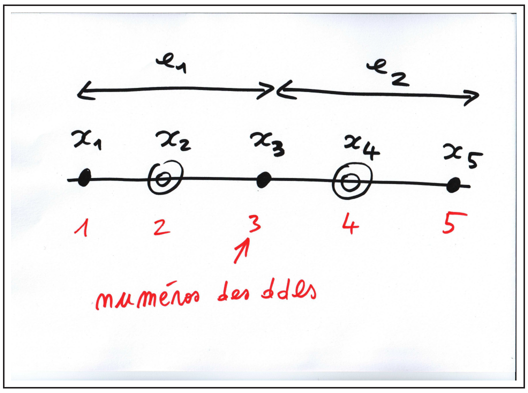


FIGURE 3 – Maillage de $\overline{\Omega}$

1. vérifiez que les fichiers `Noeuds.dat`, `Elements.dat` et `ID.dat` sont corrects ;
2. modifiez le calcul de `Ne` dans `Dessin.m` (comme ci-dessous) ;

Dessin.m

```
⋮
for e=1:Nelem
        Uh_loc=Uh(Elements(e,:));
        Noeuds_loc=Noeuds(Elements(e,:));

        S1=min(Noeuds_loc);
        S2=max(Noeuds_loc);
        M=% A COMPLETER

        x=S1;
        while (x<=S2)
              Nc=% A COMPLETER
              Ne=% A COMPLETER

              uh=Uh_loc'*Ne;
              plot(x,uh,'b*');
              x=x+pas;
        end
end
⋮
```

3. exécutez `Principal.m`.

2 Un exemple simple de génération de maillage

On considère maintenant l'approximation de ($\mathscr{P}$) avec le maillage de la figure 4. On prends des éléments de mêmes longueurs et on note

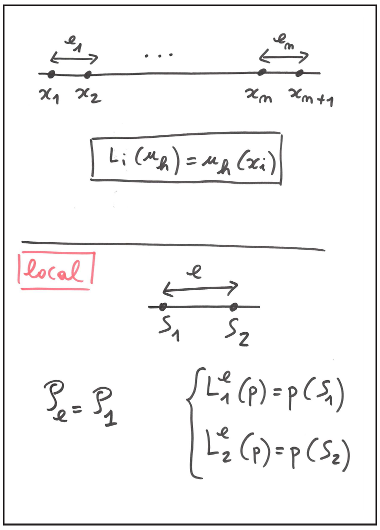


FIGURE 4 – Maillage avec n éléments finis de Lagrange $\mathcal{P}_1$

$$h = |e_1| = \cdots = |e_n| = \frac{1}{n}.$$

Les vecteurs et matrices définissant le maillage sont

$$\texttt{Noeuds} = \begin{pmatrix} x_1 \\ \vdots \\ x_{n+1} \end{pmatrix}, \quad \texttt{Elements} = \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ \vdots & \vdots \\ n & n+1 \end{pmatrix}, \quad \texttt{ID} = \begin{pmatrix} 1 \\ n+1 \end{pmatrix}. \tag{3}$$

On va écrire un script qui permet de générer ce maillage.

1. Lisez attentivement l'annexe A avant de continuer.
2. Créez un sous-dossier

   `.../TP2/Maillage`

   dans lequel vous copier/collerez les fichiers de `Calcul_pratique`. Vous choisirez `Maillage` comme *current directory* de matlab.

3. Créez et complétez le script ci-dessous qui génère les fichiers de données `Noeuds.dat`, `Elements.dat` et `ID.dat` définis précédemment, cf. (3). Vous commencerez avec $n = 4$, mais en écrivant un script qui fonctionne pour tout n.

Mailleur.m

```
%% Initialisation

clear all;
clc;
fclose('all');% ferme tout les fichiers

n=4;

%% D'abord ID

fidID=fopen('ID.dat','w');
fprintf(fidID,'%i\n%i',1,n+1);
fclose(fidID);

%% Maintenant Noeuds et Elements

fidNoeuds=% A COMPLETER
fidElements=% A COMPLETER

% A COMPLETER
⋮
% A COMPLETER

fclose(fidNoeuds);
fclose(fidElements);
```

4. Exécutez `Mailleur.m` et vérifiez qu'il génère les bons fichiers, pour $n = 4, 10$.
5. Soient un élément e comme à la figure 4 et $\{N_1^e, N_2^e\}$ sa base d'interpolation. Précisez la matrice `M` qui permet de calculer ces fonctions par la formule :
$$\begin{pmatrix} 1 \\ x \end{pmatrix} = \mathtt{M} \begin{pmatrix} N_1^e(x) \\ N_2^e(x) \end{pmatrix}.$$
6. Dessinez le graphe de $x \mapsto u_h(x)$, pour $n = 4, 10$. Pour celà, exécutez successivement `Mailleur.m` et `Principal.m`, sans oublier de modifier les calculs de `M`, `Nc` et `dNc` dans `AeFe.m` et `Dessin.m`.
7. On admet maintenant que le déplacement exact est donnée par :
$$u(x) = \begin{cases} a_1\, x^2 + b_1\, x & \text{si } x < 1/2, \\ a_2\, x^2 + b_2\, x + c_2 & \text{sinon,} \end{cases} \tag{4}$$

où

$$\begin{cases} a_1 = -0.005625, \\ b_1 = 0.010035, \\ a_2 = -0.045, \\ b_2 = 0.080278, \\ c_2 = -0.025278. \end{cases}$$

Définissez la fonction `u.m`, qui sera la fonction $x \mapsto u(x)$ de (4).

8. Modifiez `Dessin.m` afin de dessiner le graphe de $x \mapsto u(x)$ en même temps que celui de $x \mapsto u_h(x)$. Pour les différencier, vous dessinerez le graphe de u avec des ronds rouges, en utilisant l'option `'ro'` dans la commande `plot`.
9. A la fin de `Dessin.m`, mettez à jour la légende par :

```
⋮
legend('Solution approchée','Solution exacte');
title('Graphes des solutions');
⋮
```

10. Dessinez les graphes de u et u_h pour $n = 2, 4, 10, 50$. Que constatez vous ?

A Comment écrire sur un fichier de données ?

Voici une liste de commandes permettant d'écrire sur un fichier :

La commande	sert à :
`> fid=fopen('Exemple.dat','w')`	Ouvrir un fichier `Exemple.dat` pour y écrire dessus. Si le fichier n'existe pas, il sera créé ; sinon, les anciennes données seront effacées. Un numéro est attribué à la variable `fid` pour identifier le fichier.
`> fprintf(fid,'%f',1.2)`	Ecrire `1.2` dans le fichier. On précise le format des données par : — `%f` pour les réels (*floating number*) ; — `%i` pour les entiers (*integer*). Les modifications seront lisibles avec un éditeur de texte, après la fermeture du fichier (voir ci-dessous).
`> fprintf(fid,'\n')`	Retourner à la ligne.
`> fprintf(fid,' ')`	Laisser un espace.
`> fprintf(fid,'%f %f\n%i',0.1,0.9,2)`	Effectuer la suite de commandes : — écrire le réel `0.1` ; — laisser un espace ; — écrire le réel `0.9` ; — retourner à la ligne ; — écrire l'entier `2`.
`> fclose(fid)`	Fermer le fichier.

Par exemple, le script

Ecriture.m

```
fid=fopen('Fichier.dat','w');
fprintf(fid,'%f %f %f ',1.2,3.6,7.8);
fprintf(fid,'%f\n%i',2.9,2);
fclose(fid);
```

permet de générer le fichier

Fichier.dat

```
1.2   3.6   7.8   2.9
2
```

où la dernière donnée est enregistrée au format « entier ».

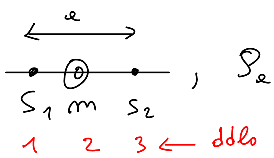
Corrigé du TP no. 2

(Voir la fin du corrigé pour les programmes ; ci-dessous on commente seulement quelques questions.)

Section 1 : calcul pratique ...

<u>Préliminaires</u> (sous-section 1.1)

1-2) On considère l'élément fini de la figure 1, c'est-à-dire



, $\mathcal{P}_e = \text{vect}\{1, x, x^2\}$ et $\begin{cases} L_1^e(p) = p(S_1), \\ L_2^e(p) = p''(m), \\ L_3^e(p) = p(S_2), \end{cases}$

$\left(\text{avec } m = \dfrac{S_1+S_2}{2}\right)$.

<u>Exercice</u> (à faire à la maison)

Montrez que ce triplet est bien un élément fini.

Notons $\{N_1^e, N_2^e, N_3^e\}$ sa base d'interpolation (voir l'annexe du cours sur la dimension 1). D'après le cours, on sait que

$$(1) \qquad p(x) = \sum_{k=1}^{3} L_k^e(p)\, N_k^e(x) \qquad \forall p \in \mathcal{P}_e,\ \forall x \in \mathbb{R}.$$

Pour trouver la matrice de passage M entre la base canonique de $\text{Vect}\{1, x, x^2\}$ et celle d'interpolation, on doit écrire les polynômes de la base canonique comme combinaisons linéaires des N_k^e. Pour celà, on utilise (1). Prenons d'abord le polynôme

$$p(x) = x^2 \qquad \forall x \in \mathbb{R}.$$

On a bien $p \in \mathcal{P}_e$, donc d'après (1)

$$x^2 = \underbrace{L_1^e(p)}_{= p(S_1) = S_1^2} N_1^e(x) + \underbrace{L_2^e(p)}_{= p''(m) = 2} N_2^e(x) + \underbrace{L_3^e(p)}_{= p(S_2) = S_2^2} N_3^e(x) \qquad \forall x \in \mathbb{R}.$$

(Car $p'(x) = 2x$ et $p''(x) = 2$)

On en déduit que

$$x^2 = S_1^2 \, N_1^e(x) + 2 \times N_2^e(x) + S_2^2 \, N_3^e(x) \qquad (\forall x \in \mathbb{R}).$$

On fait de même pour les autres polynômes de la base canonique; c'est-à-dire, on applique (1) en choisissant successivement $p : x \mapsto 1$ puis $p : x \mapsto x$. On obtient

$$1 = N_1^e(x) + 0 \times N_2^e(x) + N_3^e(x)$$

$$\text{et} \quad x = S_1 N_1^e(x) + 0 \times N_2^e(x) + S_2 N_3^e(x),$$

pour tout $x \in \mathbb{R}$. Par suite,

$$(2) \qquad \boxed{\begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 1 \\ S_1 & 0 & S_1 \\ S_1^2 & 2 & S_2^2 \end{pmatrix}}_{\text{// notée } M} \begin{pmatrix} N_1^e(x) \\ N_2^e(x) \\ N_3^e(x) \end{pmatrix} \quad (\forall x \in \mathbb{R}).}$$

3) En dérivant (2), on a

$$(3) \qquad \boxed{\begin{pmatrix} 0 \\ 1 \\ x \end{pmatrix} = M \left(\left(N_k^e\right)'(x) \right)_{1 \leq k \leq 3} \quad \forall x \in \mathbb{R}.}$$

4- a) $N_2^e(0.1) = -0.02.$

b) $\left(N_3^e\right)'(0.1) = 2.$

Intégration des coefficients élémentaires (sous-section 1.2)

On utilise évidemment (2) et (3) pour calculer

$$\boxed{\begin{aligned} Ne &= \left(N_k^e(x)\right)_{1\leq k\leq 3} = M^{-1}\begin{pmatrix}1\\ x\\ x^2\end{pmatrix} \\ \text{et}\quad dNe &= \left(\left(N_k^e\right)'(x)\right)_{1\leq k\leq 3} = M^{-1}\begin{pmatrix}0\\ 1\\ 2x\end{pmatrix},\end{aligned}}$$

avec $\boxed{x = \xi_i}$ et où la maille e ainsi que le point d'intégration ξ_i dépendent des boucles.

Ces vecteurs nous donneront alors les valeurs des $N_k^e(\xi_i)$ et $\left(N_k^e\right)'(\xi_i)$ nécessaires au calcul. Pour les récupérer, on doit se rappeler que (si $x=\xi_i$, alors)

$$Ne = \begin{pmatrix} N_1^e(\xi_i) \\ N_2^e(\xi_i) \\ N_3^e(\xi_i)\end{pmatrix}, \qquad dNe = \begin{pmatrix} \left(N_1^e\right)'(\xi_i) \\ \left(N_2^e\right)'(\xi_i) \\ \left(N_3^e\right)'(\xi_i)\end{pmatrix},$$

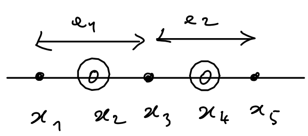
d'où (sur matlab)

$$\boxed{N_k^e(\xi_i) = Ne(k) \quad \text{et} \quad \left(N_k^e\right)'(\xi_i) = dNe(k).}$$

Résolution (sous-section 1.3)

1) On considère le maillage de la figure 3,

c'est-à-dire



$\left(\Gamma_D = \{x_1, x_5\} \right)$.

1 2 3 4 5 ← ddls

$$\begin{cases} L_1(u_h) = u_h(x_1) \\ L_2(u_h) = u_h''(x_2) \\ L_3(u_h) = u_h(x_3) \\ L_4(u_h) = u_h''(x_4) \\ L_5(u_h) = u_h(x_5) \end{cases}$$

D'où

noeuds qui supportent les ddls

$$\text{Noeuds} = \begin{pmatrix} x_{L_1} \\ x_{L_2} \\ x_{L_3} \\ x_{L_4} \\ x_{L_5} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.25 \\ 0.5 \\ 0.75 \\ 1 \end{pmatrix},$$

Les ddls sont numérotés comme les noeuds.

$$\text{Elements} = \begin{array}{c} \\ \begin{pmatrix} 1 & \overset{i}{2} & 3 \\ & & \\ 4 & 5 & 6 \end{pmatrix} \\ \uparrow \quad \uparrow \quad \uparrow \\ k=1 \quad 2 \quad 3 \end{array} \begin{array}{l} \leftarrow e = e_1 \\ \\ \leftarrow e_2 \end{array}$$

$$\boxed{\begin{array}{c} i = \text{Elements}(e,k) \\ \Updownarrow \\ \phi_i\big|_e = N_k^e \end{array}}$$

$$\left(\begin{array}{ll} \{\phi_i\}_i : & \text{base globale} \\ \{N_k^e\}_k & \text{— locale} \\ L_i(\phi_j) = \delta_{ij}, & L_k^e(N_m^e) = \delta_{km} \end{array} \right)$$

et $I_D = \{ i \mid \phi_i \not\equiv 0 \text{ sur } \Gamma_D \}$

$= \{ i \mid \phi_i(x_1) \neq 0 \text{ ou } \phi_i(x_5) \neq 0 \}$

$= \{ i \mid L_1(\phi_i) \neq 0 \text{ ou } L_5(\phi_i) \neq 0 \}$

$= \{1, 5\}$.

2) On rappelle que

$$Ne = \left(N_k^e(x) \right)_{1 \leq k \leq 3}$$

dans Dessin.m (où e et x dépendent des boucles).
Pour le calculer, on doit utiliser (2) comme précédemment.

3) Après exécution de Principal.m, on obtient le même dessin qu'au TP no. 1. C'est normal, car on a considéré le même problème aux limites continu, dont on a approché la solution avec le même type de fonctions polynômiales par morceaux (de degré ≤ 2 à chaque fois sur les mailles).
L'espace d'interpolation V_h est donc resté le même

et c'est seulement les fonctions de base ϕ_i, qui ont changé. En d'autres termes, on a simplement fait un changement de base pour calculer la même solution $u_h : \overline{\Omega} \longrightarrow \mathbb{R}$.

Remarque

Il se trouve que ce u_h est aussi la solution exacte du problème, mais c'est un cas particulier. Celà vient du fait qu'avec les données $c = c(x)$, $f = f(x)$, etc., considérées, la solution exacte est déjà continue et polynômiale de degré 2 par morceaux. On ne fait donc pas d'erreur en remplaçant V par V_h. De plus, $c = c(x)$ et $f = f(x)$ sont aussi des polynômes de degrés suffisamment bas pour qu'on ne fasse pas d'erreur dans les calculs de A^e et F^e, avec la formule de Gauss.

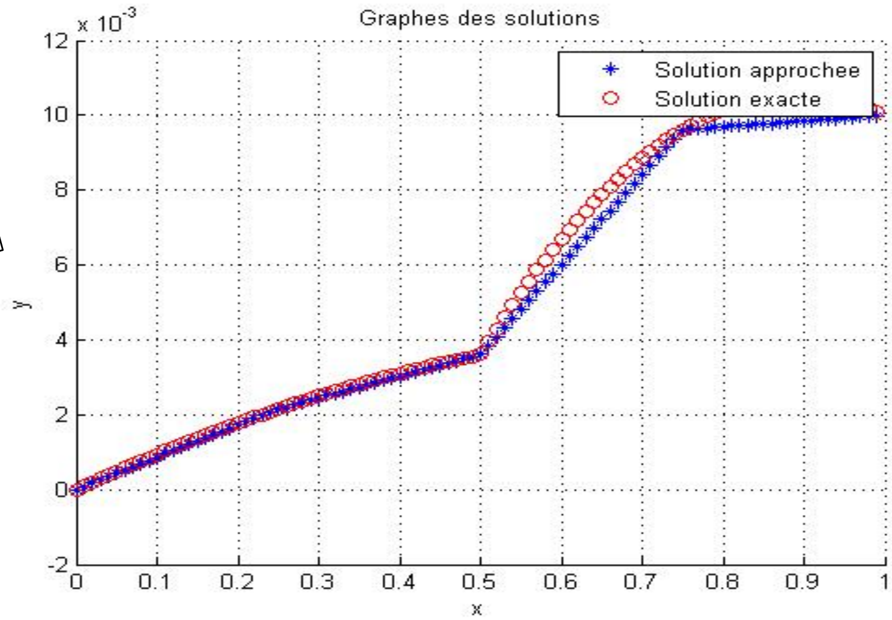
Section 2 : un exemple ...

En raisonnant comme précédemment avec nos nouveaux éléments finis, on trouve que pour tout élément $e=[S_1,S_2]$, on a la matrice de passage $\boxed{M=\begin{pmatrix}1 & 1\\ S_1 & S_2\end{pmatrix}}$ et pour tout $x\in\mathbb{R}$, on a

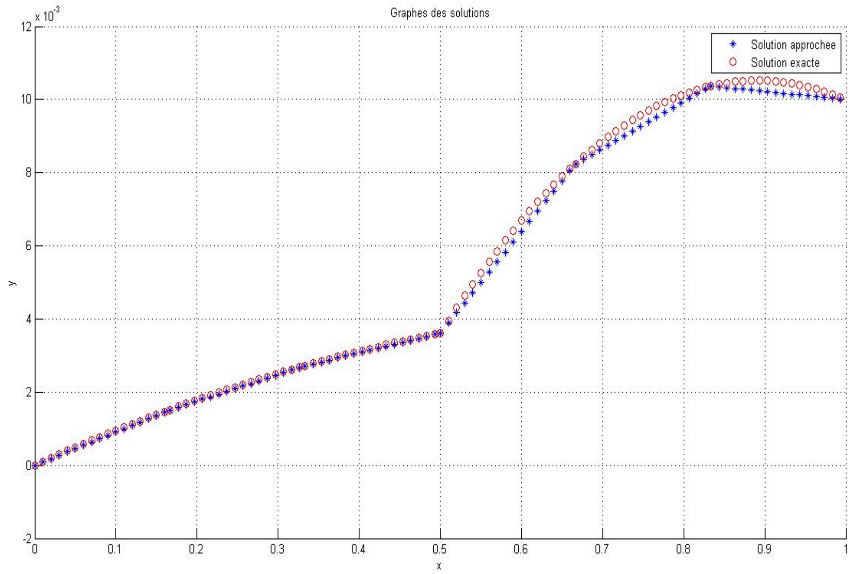
$$\boxed{\begin{pmatrix}1\\ x\end{pmatrix}=M\begin{pmatrix}N_1^e(x)\\ N_2^e(x)\end{pmatrix} \quad \text{et} \quad \begin{pmatrix}0\\ 1\end{pmatrix}=M\begin{pmatrix}(N_1^e)'(x)\\ (N_2^e)'(x)\end{pmatrix}.}$$

Voici enfin quelques graphes pour différents maillages :

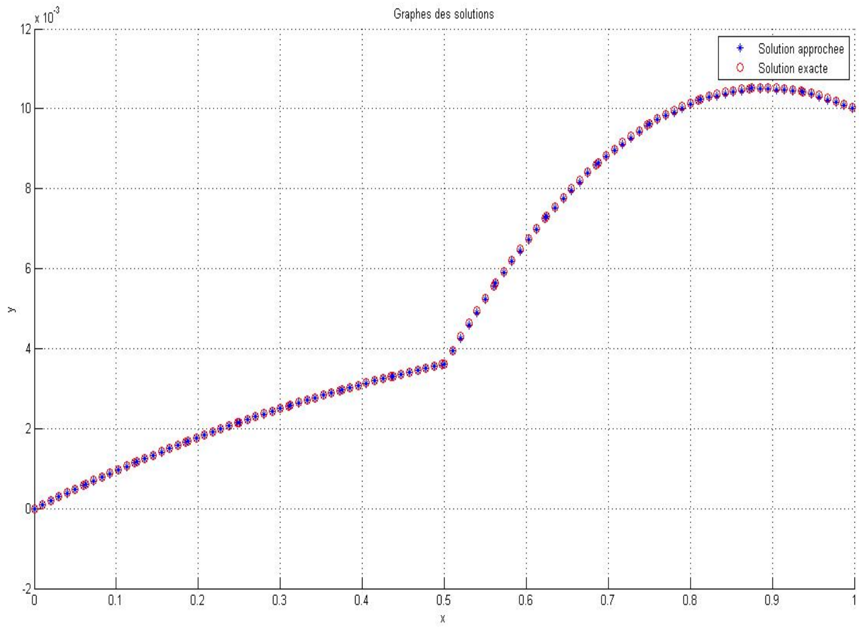
$\boxed{n=4}$



$n = 6$



$n = 16$



<u>Remarque :</u> la solution approchée u_h est maintenant affine par morceaux ; elle converge vers u lorsque $n \to \infty$.

Programmes corrigés du TP no. 2 : calcul pratique

Scipts

Principal.m

(Voir les programmes du TP no. 1.)

Dessin.m

```
clf;
hold on;
grid on;

pas=0.01;

for e=1:Nelem

    Uh_loc=Uh(Elements(e,:));
    Noeuds_loc=Noeuds(Elements(e,:));

    S1=min(Noeuds_loc);
    S2=max(Noeuds_loc);
    M=[1 0 1;S1 0 S2;S1^2 2 S2^2];

    x=S1;
    while (x<=S2)
        Nc=[1;x;x^2];
        Ne=M\Nc;

        uh=Uh_loc'*Ne;
        plot(x,uh,'b*');
        x=x+pas;
    end
end

% titre, legende, etc.

legend('Graphe de x -> uh(x)');
title('Solution approchee');
xlabel('x');ylabel('y');
```

Fonctions

AeFe.m

```
function [Ae,Fe]=AeFe(Noeuds_loc)

%% Noeuds et longueur de l'element

S1=min(Noeuds_loc);
S2=max(Noeuds_loc);
m=(S1+S2)/2;
h=S2-S1;% longueur

%% Points et poids d'integration

xi=zeros(2,1);
xi(1)=m-(h/2)*(sqrt(3)/3);
xi(2)=m+(h/2)*(sqrt(3)/3);

omega=zeros(2,1);
```

```
omega(1)=1/2;
omega(2)=1/2;

%% Matrice de passage

M=[1 0 1;S1 0 S2; S1^2 2 S2^2];

%% Integration des coefficients elementaires

N_loc=size(Noeuds_loc,1);% nombre de ddls locaux
Ae=zeros(N_loc,N_loc);
Fe=zeros(N_loc,1);

Nint=size(xi,1);% nombre de points d'integration
for i=1:Nint

    x=xi(i);% point d'integration
    Nc=[1;x;x^2];
    Ne=M\Nc;
    dNc=[0;1;2*x];
    dNe=M\dNc;

    for k=1:N_loc
        Fe(k)=Fe(k)+h*omega(i)*f(x)*Ne(k);
        for n=1:N_loc
            Ae(k,n)=Ae(k,n)+h*omega(i)*c(x)*dNe(n)*dNe(k);
        end
    end
end
```

alpha.m, c.m et f.m

(Voir les programmes du TP no. 1)

Fichiers de données

Noeuds.dat, Elements.dat et ID.dat

(Voir les programmes du TP no. 1.)

Programmes corrigés du TP no. 2 : maillage

Scripts

Mailleur.m

```
%% Initialisation

clear all;
clc;
fclose('all');% ferme tout les fichiers

n=4;

%% D'abord ID

fidID=fopen('ID.dat','w');
fprintf(fidID,'%i\n%i',1,n+1);
fclose(fidID);

%% Maintenant Noeuds et Elements

fidNoeuds=fopen('Noeuds.dat','w');
fidElements=fopen('Elements.dat','w');

x=0;
pas=1/n;

for i=1:n+1
    fprintf(fidNoeuds,'%f\n',x);
    x=x+pas;
end

for i=1:n
    fprintf(fidElements,'%f %f\n',i,i+1);
end

fclose(fidNoeuds);
fclose(fidElements);
```

Principal.m

(Voir les programmes du TP no. 1.)

Dessin.m

```
clf;
hold on;
grid on;

pas=0.01;

for e=1:Nelem

    Uh_loc=Uh(Elements(e,:));
    Noeuds_loc=Noeuds(Elements(e,:));

    S1=min(Noeuds_loc);
    S2=max(Noeuds_loc);
    M=[1 1;S1 S2];

    x=S1;
```

```
    while (x<=S2)
        Nc=[1;x];
        Ne=M\Nc;

        uh=Uh_loc'*Ne;
        plot(x,uh,'b*');
        plot(x,u(x),'ro');
        x=x+pas;
    end
end

% titre, legende, etc.

legend('Solution approchee','Solution exacte');
title('Graphes des solutions');
xlabel('x');ylabel('y');
```

Fonctions

AeFe.m

```
function [Ae,Fe]=AeFe(Noeuds_loc)

%% Noeuds et longueur de l'element

S1=min(Noeuds_loc);
S2=max(Noeuds_loc);
m=(S1+S2)/2;
h=S2-S1;% longueur

%% Points et poids d'integration

xi=zeros(2,1);
xi(1)=m-(h/2)*(sqrt(3)/3);
xi(2)=m+(h/2)*(sqrt(3)/3);

omega=zeros(2,1);
omega(1)=1/2;
omega(2)=1/2;

%% Matrice de passage

M=[1 1;S1 S2];

%% Integration des coefficients elementaires

N_loc=size(Noeuds_loc,1);% nombre de ddls locaux
Ae=zeros(N_loc,N_loc);
Fe=zeros(N_loc,1);

Nint=size(xi,1);% nombre de points d'integration
for i=1:Nint

    x=xi(i);% point d'integration
    Nc=[1;x];
    Ne=M\Nc;
    dNc=[0;1];
    dNe=M\dNc;

    for k=1:N_loc
        Fe(k)=Fe(k)+h*omega(i)*f(x)*Ne(k);
        for n=1:N_loc
            Ae(k,n)=Ae(k,n)+h*omega(i)*c(x)*dNe(n)*dNe(k);
```

```
        end
    end
end
```

alpha.m, c.m, f.m

(Voir les programmes du TP no. 1)

u.m

```
function y=u(x)

a1=-0.005625;
b1=0.010035;
a2=-0.045;
b2=0.080278;
c2=-0.025278;

if (x<0.5)
    y=a1*x^2+b1*x;
else
    y=a2*x^2+b2*x+c2;
end
```

2.2 Un problème de barrage 2-d

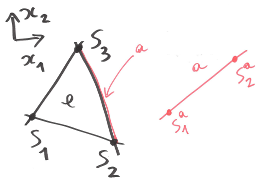
2.2.1 TD no. 3 : aspects théoriques

(Tournez la page.)

TD no. 3

Approximation des EDPs, Printemps 2016-2017

Exercice 1 (Eléments finis triangulaire)**.** Soit e un triangle de $\mathbb{R}^2$ et a une de ses arêtes.



On muni e de l'espace d'interpolation

$$\mathcal{P}_e = \text{vect}\,\{1, x_1, x_2\} \overset{\text{noté}}{=} \mathcal{P}_1$$

et des ddls $L_k^e : \mathcal{P}_e \to \mathbb{R}$ définis par

$$\begin{cases} L_1^e(p) = p(S_1), \\ L_2^e(p) = p(S_2), \\ L_3^e(p) = p(S_3). \end{cases}$$

1. Montrez que $(e, \mathcal{P}_e, \{L_1^e, \ldots, L_3^e\})$ est un élément fini.
2. Soit $p \in \mathcal{P}_e$ et q la restriction de p à a, i.e.

$$\begin{array}{rccl} q: & a & \to & \mathbb{R} \\ & s & \mapsto & p(s). \end{array}$$

   On considère la paramétrisation suivante de l'arête a :

$$\gamma : t \in [0,1] \mapsto S_1^a + t\,(S_2^a - S_1^a) \in a.$$

   Montrez que la fonction

$$t \in [0,1] \mapsto q(\gamma(t)) \in \mathbb{R}$$

   est un polynôme dont vous préciserez le degré.
3. En déduire la structure d'élément fini induite $(a, \mathcal{P}_a, \{L_1^a, L_2^a\})$.

Exercice 2 (Eléments finis à valeurs vectorielles)**.** Soit e et a comme à l'exercice précédent. Etant donné une fonction

$$\vec{p}\colon e \to \mathbb{R}^2,$$

on note $\vec{p}= \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ ses applications composantes ; i.e.

$$\vec{p}\colon (x_1, x_2) \mapsto \begin{pmatrix} p_1(x_1, x_2) \\ p_2(x_1, x_2) \end{pmatrix}.$$

On muni maintenant e de l'espace d'interpolation

$$\vec{\mathcal{P}}_e = \left\{ \vec{p} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \text{ tels que } p_1, p_2 \in \mathcal{P}_1 \right\} = \mathcal{P}_1 \times \mathcal{P}_1$$

et des ddls $\mathscr{L}_k^e : \vec{\mathcal{P}}_e \to \mathbb{R}$ définis par

$$\mathscr{L}_k^e(\vec{p}) = \begin{cases} p_1(S_k) & \text{si } k = 1, 2, 3, \\ p_2(S_{k-3}) & \text{si } k = 4, 5, 6. \end{cases}$$

Ces numérotations se représentent par le schéma suivant :

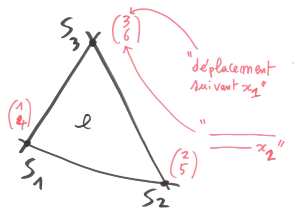


FIGURE 1 – Numérotation des ddls.

1. Montrez que $\left(e, \vec{\mathcal{P}}_e, \{\mathscr{L}_1^e, \ldots, \mathscr{L}_6^e\}\right)$ est un élément fini.
2. On note $\left(a, \vec{\mathcal{P}}_a, \{\mathscr{L}_1^a, \ldots, \mathscr{L}_4^a\}\right)$ la structure d'élément fini induite sur a, dont on numérotera les ddls comme à la figure 2.

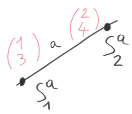


FIGURE 2 – Ddls sur une arête.

Précisez ce que sont $\vec{\mathcal{P}}_a$ et $\mathscr{L}_k^a$.

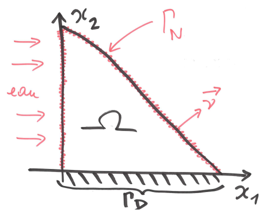
2

3. On note $\{\vec{N_1^e}, \ldots, \vec{N_6^e}\}$ et $\{\vec{N_1^a}, \ldots, \vec{N_4^a}\}$ les bases d'interpolation associées aux éléments finis précédents. On note ensuite $\{N_1^e, N_2^e, N_3^e\}$ et $\{N_1^a, N_2^a\}$ les bases d'interpolation associées aux éléments finis de l'exercice 1.
   (a) Montrez que
   $$\vec{N_1^e} = \begin{pmatrix} N_1^e \\ 0 \end{pmatrix} \quad \text{et} \quad \vec{N_1^a} = \begin{pmatrix} N_1^a \\ 0 \end{pmatrix}.$$
   (b) Ecrivez tout les $\vec{N_k^e}$ et $\vec{N_k^a}$ de la même manière.

Exercice 3 (Calcul explicite des bases)**.** Dans cet exercice, on se propose de calculer les N_k^e et les N_k^a (qui sont les fonctions d'interpolation associées aux éléments finis de l'exercice 1).

1. Etant donnés deux points $M_1, M_2 \in \mathbb{R}^2$, on note $\overrightarrow{M_1M_2}$ le vecteur colonne
   $$\overrightarrow{M_1M_2} = \begin{pmatrix} x_{21} - x_{11} \\ x_{22} - x_{12} \end{pmatrix},$$
   où $M_k = (x_{k1}, x_{k2})$.
   Etant donnés $M_3, M_4 \in \mathbb{R}^2$, on note
   $$\| \overrightarrow{M_1M_2}, \overrightarrow{M_3M_4} \|,$$
   la matrice dont les colonnes sont données par ces vecteurs.
   Montrez que pour tout $M \in \mathbb{R}^2$,
   $$N_1^e(M) = \frac{\det \| \overrightarrow{S_3M}, \overrightarrow{S_3S_2} \|}{\det \| \overrightarrow{S_3S_1}, \overrightarrow{S_3S_2} \|}.$$
2. En déduire les formules explicites de N_2^e et N_3^e.
3. On note $\{N_1^a, N_2^a\}$ la base d'interpolation associée. Montrez que pour tout $s \in a$,
   $$\begin{aligned} N_1^a(s) &= (s - S_2^a) \cdot \frac{S_1^a - S_2^a}{\|S_1^a - S_2^a\|^2}, \\ N_2^a(s) &= (s - S_1^a) \cdot \frac{S_2^a - S_1^a}{\|S_2^a - S_1^a\|^2}, \end{aligned}$$
   où $\cdot$ et $\| \cdot \|$ désignent le produit scalaire et la norme euclidienne de $\mathbb{R}^2$.

Exercice 4. On considère un barrage triangulaire encastré dans une fondation rigide et soumis à la pression de l'eau.



On note $\vec{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$ le déplacement à l'équilibre et on admet que $\vec{u}$ satisfait

$$\begin{cases} -\overrightarrow{\operatorname{div}}\,(\sigma) = \vec{f} \quad \text{dans } \Omega, \\ \sigma\,\vec{\nu} = \vec{\beta} \quad \text{sur } \Gamma_N, \\ \vec{u} = 0 \quad \text{sur } \Gamma_D, \end{cases} \tag{$\mathscr{P}$}$$

où :

- $\vec{\nu} = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}$ est la normale unitaire extèrieure,
- $\mu, \lambda > 0$ sont des constantes données,
- $\vec{f} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$, $\vec{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$ sont des fonctions données,
- $\sigma = \left(\sigma_{ij}(\vec{u})\right)_{1 \le i,j \le 2}$ est le tenseur des contraintes,
- et $\epsilon = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)_{1 \le i,j \le 2}$ est le tenseur des déformations.

On rappelle la loi de Hooke :

$$\sigma = 2\,\mu\,\epsilon + \lambda\,\theta\,\mathrm{I},$$

avec

$$\theta = \operatorname{tr}(\epsilon) = \epsilon_{11} + \epsilon_{22}$$

et $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. On précise aussi que

$$\overrightarrow{\operatorname{div}}\,(\sigma) = \begin{pmatrix} \frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} \\ \frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} \end{pmatrix}.$$

Enfin, on note comme d'habitude $\overline{\Omega} = \Omega \cup \Gamma_D \cup \Gamma_N$.

1. Montrer que pour toute fonction test

$$\vec{\phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} : \overline{\Omega} \to \mathbb{R}^2$$

nulle sur Γ_D,

$$\underbrace{\int_\Omega \left(\sum_{1\le i,j\le 2} \sigma_{ij}(\vec{u})\, \frac{\partial \phi_i}{\partial x_j} \right)}_{\text{noté } a(\vec{u},\vec{\phi})} = \underbrace{\int_\Omega \vec{f}\cdot\vec{\phi} + \int_{\Gamma_N} \vec{\beta}\cdot\vec{\phi}}_{\text{noté } l(\vec{\phi})}.$$

2. Montrer que

$$\sum_{1\le i,j\le 2} \sigma_{ij}(\vec{u})\, \frac{\partial \phi_i}{\partial x_j} = \sum_{1\le i,j\le 2} \sigma_{ij}(\vec{u})\, \epsilon_{ij}(\vec{\phi}).$$

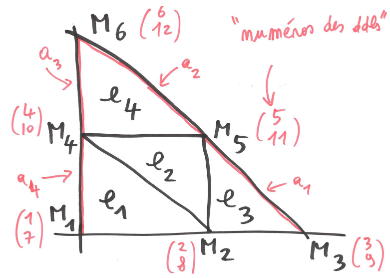
3. En déduire que

$$a(\vec{u},\vec{\phi}) = 2\,\mu \int_\Omega \epsilon(\vec{u}) : \epsilon(\vec{\phi}) + \lambda \int_\Omega \theta(\vec{u})\,\theta(\vec{\phi}),$$

où : désigne le double produit contracté $A : B = \sum_{1\le i,j\le 2} A_{ij}\, B_{ij}$.

4. En déduire la formulation variationnelle de $(\mathscr{P})$.

5. On considère l'approximation de ce problème avec les éléments finis de l'exercice 2 et les numérotations ci-dessous :



On obtient l'espace d'interpolation

$$\vec{V_h} = \left\{ \vec{u_h} \in C(\overline{\Omega}, \mathbb{R}^2) \text{ tels que } \vec{u_h}_{|_e} \in \mathcal{P}_1 \times \mathcal{P}_1 \text{ pour tout } e \right\}$$

et les ddls

$$\begin{cases} \mathscr{L}_1(\vec{u_h}) = *, \\ \vdots \\ \mathscr{L}_{12}(\vec{u_h}) = *, \end{cases}$$

que vous préciserez.

6. Soit $\{\vec{\phi}_1, \dots, \vec{\phi}_{12}\}$ la base d'interpolation associée. On rappelle que la méthode des éléments finis consiste à approcher le problème continu $(\mathscr{P})$ par un problème discret, qui peut s'écrire sous la forme matricielle

$$A\,U_h = F,$$

avec

$$A_{ij} = \begin{cases} \delta_{ij} & \text{si } i \in I_D, \\ a(\vec{\phi}_j, \vec{\phi}_i) & \text{sinon,} \end{cases}$$

$$F_i = \begin{cases} * & \text{si } i \in I_D, \\ l(\vec{\phi}_i) & \text{sinon,} \end{cases}$$

où vous préciserez I_D et $*$.

7. On choisit les correspondances suivantes entre les numérotations locales et globales des ddls :

$$\texttt{Elements} = \begin{pmatrix} 1 & 2 & 4 & 7 & 8 & 10 \\ 4 & 2 & 5 & 10 & 8 & 11 \\ 2 & 3 & 5 & 8 & 9 & 11 \\ 4 & 5 & 6 & 10 & 11 & 12 \end{pmatrix},$$

$$\texttt{Aretes} = \begin{pmatrix} 3 & 5 & 9 & 11 \\ 5 & 6 & 11 & 12 \\ 6 & 4 & 12 & 10 \\ 4 & 1 & 10 & 7 \end{pmatrix}.$$

Exprimez $\vec{\phi}_{11}$ en fonction des $\vec{N}_k^e$ et $\vec{N}_k^a$.

8. Etant donné un élément e et une arête a, comme aux figures 1 et 2, on définit les coefficients élémentaires

$$A_{kn}^e = 2\,\mu \int_e \epsilon(\vec{N}_n^e) : \epsilon(\vec{N}_k^e) + \lambda \int_e \theta(\vec{N}_n^e)\,\theta(\vec{N}_k^e),$$

$$F_k^e = \int_e \vec{f} \cdot \vec{N}_k^e,$$

$$F_k^a = \int_a \vec{\beta} \cdot \vec{N}_k^a\ .$$

Ecrivez $A_{11,6}$ et F_{10} comme des sommes de ces coefficients élémentaires.

9. Calculez F_k^a de manière approchée par la formule de Simpson (pour $k = 1, \dots, 4$). Vous noterez $|a|$ la longueur de a, $m^a = \frac{S_1^a + S_2^a}{2}$ le milieu, et vous obtiendrez des formules qui dépendent seulement de $|a|$ et des valeurs de $\vec{\beta}$ en S_1^a, S_2^a et m^a.

(Voir le TP no. 3 pour la suite.)

$\boxed{\text{Corrigé du TD no. 3}}$

<u>Exercice 1</u>

1) Montrons seulement l'unisolvance, le reste pouvant se vérifier facilement avec les arguments développés au TD no. 1.

Soit donc $p \in \mathcal{P}_e = \text{vect}\{1, x_1, x_2\}$ tel que

$$L_1^e(p) = L_2^e(p) = L_3^e(p) = 0,$$

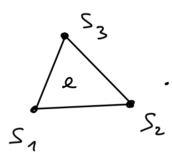
et montrons que nécessairement p est le polynôme nul.

On rappelle que p est une fonction de $\mathbb{R}^2$ dans $\mathbb{R}$, de la forme

$$p(x_1, x_2) = A + B x_1 + C x_2 \qquad \forall x_1, x_2 \in \mathbb{R},$$

pour certaine constantes $A, B, C \in \mathbb{R}$. Notons

$$S_k = (S_{k1}, S_{k2}) \in \mathbb{R}^2$$

les coordonnées des noeuds du triangle 

.

Par hypothèse, on a

$$p(S_1) = A + B S_{11} + C S_{12} = 0$$

avec la même chose pour S_2 et S_3.

Cela veut dire que le triplet (A, B, C) est solution du système

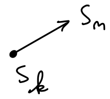
$$(\mathcal{S})\quad \begin{cases} A + B S_{11} + C S_{12} = 0\,, \\ A + B S_{21} + C S_{22} = 0\,, \\ A + B S_{31} + C S_{32} = 0\,. \end{cases}$$

C'est un système linéaire carré, qui admet donc une et une seule solution si et seulement si son déterminant est non nul. Or, ce déterminant vaut

$$\det = \begin{vmatrix} 1 & S_{11} & S_{12} \\ 1 & S_{21} & S_{22} \\ 1 & S_{31} & S_{32} \end{vmatrix} \begin{array}{l} (L_1) \\ (L_2) \\ (L_3) \end{array}$$

$$= \begin{vmatrix} 1 & S_{11} & S_{12} \\ 0 & S_{21}-S_{11} & S_{22}-S_{12} \\ 0 & S_{31}-S_{11} & S_{32}-S_{12} \end{vmatrix} \begin{array}{l} (L_1) \\ (L_1)-(L_2) \\ (L_1)-(L_3) \end{array}$$

$$= \begin{vmatrix} S_{21}-S_{11} & S_{22}-S_{12} \\ S_{31}-S_{11} & S_{32}-S_{12} \end{vmatrix}.$$

Notons $\overrightarrow{S_k S_m}$ le vecteur 

dont les coordonnées sont $(S_{m1} - S_{k1}, S_{m2} - S_{k2}) \in \mathbb{R}^2$.

On a alors

$$\det = \left\| \begin{matrix} \overrightarrow{S_1 S_2} \\ \overrightarrow{S_1 S_3} \end{matrix} \right\| \neq 0$$

matrice constituée de ces 2 vecteurs lignes

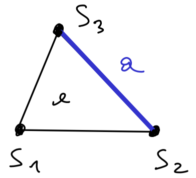
car on suppose évidemment que le triangle n'est pas plat, d'où $\overrightarrow{S_1 S_2}$ et $\overrightarrow{S_1 S_3}$ ne sont pas colinéaires.

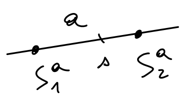
Ceci montre que $(\mathcal{S})$ admet une et une seule solution.

On déduit que $(A, B, C) = (0, 0, 0)$, car le triplet nul est une solution triviale.

Par suite, p est bien le polynôme nul.

2) Dans la suite, a est une arête donnée de e,



dont on notera 

les nœuds et la variable.

Soit $p \in \mathcal{P}_a$, c'est-à-dire

$$p : \mathbb{R}^2 \longrightarrow \mathbb{R} ,$$

et notons q la restriction de p à a , c'est-à-dire

$$\begin{aligned} q : a \subset \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ s=(s_1, s_2) &\longmapsto p(s_1, s_2) \end{aligned}$$

(notée $q = p|_a$).

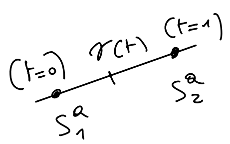
Montrons qu'alors $q \in \mathcal{P}_1^a = \mathrm{vect}\{1, s\}$, avec la définition du cours de cet espace de polynômes. D'après le cours, on doit montrer que la fonction composée

$$\begin{aligned} q \circ \gamma : [0, 1] &\longrightarrow \mathbb{R} \\ t &\longmapsto q(\gamma(t)) \end{aligned}$$

est un polynôme de $\mathbb{R}$ dans $\mathbb{R}$, de degré ≤ 1 , où

$$\begin{aligned} \gamma : [0, 1] &\longrightarrow a \subset \mathbb{R}^2 \\ t &\longmapsto S_1^a + t(S_2^a - S_1^a) \end{aligned}$$

est une paramétrisation (affine) de a :



On continue de noter $S_k^a = (S_{k1}^a, S_{k2}^a) \in \mathbb{R}^2$, les coordonnées des noeuds. On a donc

$$q(\gamma(t)) = p\left(S_{11}^a + t(S_{21}^a - S_{11}^a), S_{12}^a + t(S_{22}^a - S_{12}^a)\right),$$

pour tout $t \in [0,1]$. Or $p \in \mathrm{vect}\{1, x_1, x_2\}$ est de la forme

(1) $$p(x_1, x_2) = A + Bx_1 + Cx_2 \quad \forall x_1, x_2 \in \mathbb{R},$$

pour certaines constantes $A, B, C \in \mathbb{R}$. D'où

$$q(\gamma(t)) = A + B\left[S_{11}^a + t(S_{21}^a - S_{11}^a)\right] + C\left[S_{12}^a + t(S_{22}^a - S_{12}^a)\right]$$

et il est clair que c'est un polynôme de degré ≤ 1 en t. On vient donc de montrer que :

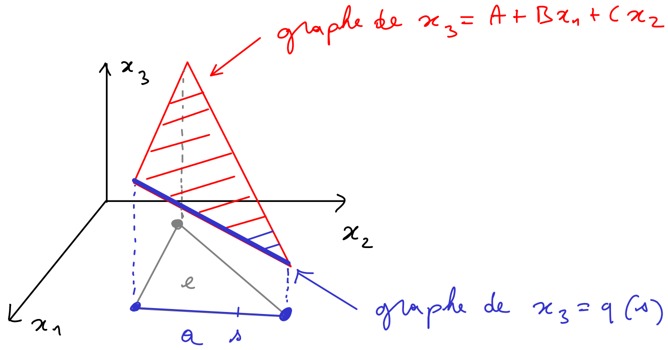
<u>Lemme</u>

$(\forall p \in \mathcal{P}_e) \qquad q = p|_a \in \mathcal{P}_1^\Lambda = \mathrm{vect}\{1, s\}$.

<u>Remarque</u>

Le graphe de p est un plan de $\mathbb{R}^3$, d'après (1).

Si on considère le graphe de $q = p|_a$, ce qui revient à faire une coupe le long de l'arête, on obtient une droite (ce qui confirme que $q \in \text{Vect}\{1,s\}$).



3) Nous allons maintenant pouvoir préciser la structure d'éléments finis que l'on obtient sur les arêtes, suite aux choix de prendre les éléments finis de l'énoncé sur e. D'après le cours, on rappelle que l'espace d'interpolation est défini par

$$\boxed{\mathcal{P}_a = \{ q = p|_a \text{ tel que } p \in \mathcal{P}_e \}.}$$

D'après le lemme précédent, on sait que $\mathcal{P}_a \subseteq \mathcal{P}_1^s$.

Montrons qu'on a en fait égalité.

<u>Preuve</u> (de $\mathcal{P}_1^s \subseteq \mathcal{P}_a$)

Il est facile de voir que $\mathcal{P}_a$ est un espace vectoriel.

Comme $\mathcal{P}_1^a$ est dimension 2, il suffit donc de trouver 2 polynômes linéairement indépendants dans $\mathcal{P}_a$.

On choisit les polynômes

$$q_1 : s \in a \longmapsto 1 \in \mathbb{R}$$

produit scalaire

$$\text{et} \quad q_2 : s \in a \longmapsto \frac{(s - S_1^a) \cdot (S_2^a - S_1^a)}{\| S_2^a - S_1^a \|^2} \in \mathbb{R}.$$

norme euclidienne

Vérifions d'abord qu'ils sont linéairement indépendants.

Soient donc $\lambda_1, \lambda_2 \in \mathbb{R}$ tels que $\lambda_1 q_1 + \lambda_2 q_2 = 0$ et montrons que nécessairement $\lambda_1 = \lambda_2 = 0$.

Par hypothèse,

$$\lambda_1 \, q_1(s) + \lambda_2 \, q_2(s) = 0 \qquad \forall s \in a.$$

En choisissant successivement $s = S_1^a$ et S_2^a, on obtient le système en (λ_1, λ_2) suivant:

$$\begin{cases} \lambda_1 & = 0, \\ \lambda_1 + \lambda_2 & = 0, \end{cases}$$

car $q_1(S_1^a) = q_1(S_2^a) = 1$, $q_2(S_1^a) = 0$ et $q_2(S_2^a) = 1$.

On déduit facilement que $\lambda_1 = \lambda_2 = 0$ et donc que la famille $\{q_1, q_2\}$ est libre. Il reste donc seulement à montrer que $q_1, q_2 \in \mathcal{P}_a$. Cela veut dire qu'on doit trouver $p_1, p_2 \in \mathcal{P}_e$ tels que $q_i = p_i /_a \quad (i = 1, 2)$.

Pour q_1, prenons le polynôme

$$p_1(x_1, x_2) = 1 \qquad \forall x_1, x_2 \in \mathbb{R}.$$

Il est clair que ce polynôme convient.

Pour définir p_2, notons

$$M = (x_1, x_2) \in \mathbb{R}^2$$

la variable générale du plan. Prenons alors la fonction

$$p_2 : \mathbb{R}^2 \longrightarrow \mathbb{R}$$
$$M \longmapsto \frac{(M - S_1^a) \cdot (S_2^a - S_1^a)}{\| S_2^a - S_1^a \|^2}$$

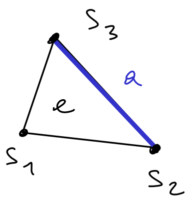
Par construction, il est clair que $q_2 = p_2 /_a$ et il reste seulement à vérifier que $q_2 \in \mathrm{Vect}\{1, x_1, x_2\}$.

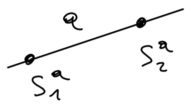
Mais cela est facile à voir en écrivant $M = (x_1, x_2)$, $S_k^a = (S_{k1}^a, S_{k2}^a)$, puis en développant le produit scalaire. La preuve de l'inclusion $\mathcal{P}_1^1 \subseteq \mathcal{P}_a$ est donc terminée. ∎

<u>Conclusion</u>

On vient de montrer que $\boxed{\mathcal{P}_a = \mathcal{P}_1^1.}$

D'après le cours, on sait aussi qu'on munit cet espace d'interpolation des ddls que l'on avait sur $\mathcal{P}_e$, en ne gardant que ceux qui sont supportés par l'arête.

On avait 

avec la notation locale 

et les ddls $L_k^e(p) = p(S_k)$ $(k = 1, 2, 3)$

sur $\mathcal{P}_e$. D'où les ddls sur $\mathcal{P}_a$ sont

$$\boxed{L_k^a(q) = q(S_k^a) \qquad (k = 1, 2).}$$

Terminologie

L'espace $\mathcal{P}_e = \text{Vect}\{1, x_1, x_2\}$ se note aussi $\mathcal{P}_1^{(x_1,x_2)}$ ou $\mathcal{P}_1$ (si il n'y a pas de confusion avec l'espace des polynômes de $\mathbb{R}$ dans $\mathbb{R}$, noté de la même manière).

Les éléments finis de cet exercice s'appellent les « éléments finis de Lagrange $\mathcal{P}_1$ bidimensionnels ». En les restreignant sur les arêtes, on récupère les « éléments finis de Lagrange $\mathcal{P}_1$ monodimensionnels », d'après ce qui précède.

Exercice 2

1) Montrons d'abord l'unisolvance. Soit donc $\vec{p} \in \mathcal{P}_e$ tel que $\mathcal{L}_1^e(\vec{p}) = \dots = \mathcal{L}_6^e(\vec{p}) = 0$ et montrons que nécessairement $\vec{p}$ est le polynôme nul. On rappelle que $\vec{p} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ a deux applications composantes, notées $\vec{p} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ et telles que

$$\vec{p}(x_1, x_2) = \begin{pmatrix} p_1(x_1, x_2) \\ \\ p_2(x_1, x_2) \end{pmatrix} \qquad \forall x_1, x_2 \in \mathbb{R},$$

où $p_i : \mathbb{R}^2 \longrightarrow \mathbb{R} \in \mathrm{Vect}\{1, x_1, x_2\}$ $(i = 1, 2)$.

On doit montrer que p_1 et p_2 sont les polynômes nuls (de $\mathbb{R}^2$ dans $\mathbb{R}$).

Pour p_1, on a

$$\begin{cases} p_1 \in \mathrm{Vect}\{1, x_1, x_2\} = \mathcal{P}_e, \\ \mathcal{L}_1^e(\vec{p}) = p_1(S_1) = L_1^e(p_1) = 0, \\ \mathcal{L}_2^e(\vec{p}) = p_1(S_2) = L_2^e(p_1) = 0, \\ \mathcal{L}_3^e(\vec{p}) = p_1(S_3) = L_3^e(p_1) = 0, \end{cases}$$

avec l'élément fini $\left(e, \mathcal{P}_e, \{L_1^e, L_2^e, L_3^e\}\right)$ de l'exercice 1. D'après l'unisolvance de cet exercice, on en déduit que $p_1 = 0$ (polynôme nul).

On raisonne de la même manière pour p_2 à partir des équations $\mathcal{L}_4^e(\vec{p}) = \mathcal{L}_5^e(\vec{p}) = \mathcal{L}_6^e(\vec{p}) = 0$, ce qui termine la preuve de l'unisolvance du triplet $\left(e, \vec{\mathcal{P}_e}, \{\mathcal{L}_1^e, \dots, \mathcal{L}_6^e\}\right)$.

Montrons maintenant que la condition

$$\dim\left(\vec{\mathcal{P}_e}\right) = \mathrm{Card}\left(\{\mathcal{L}_1^e, \dots, \mathcal{L}_6^e\}\right)$$

est vérifiée. On a

$$\begin{aligned}
\vec{\mathcal{P}_e} &= \left\{ \vec{p} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \;\middle|\; p_1, p_2 \in \mathcal{P}_1^{(x_1, x_2)} \right\} \\
&= \mathcal{P}_1^{(x_1,x_2)} \times \mathcal{P}_1^{(x_1,x_2)}
\end{aligned}$$

et donc

$$\begin{aligned}
\dim\left(\vec{\mathcal{P}_e}\right) &= 2 \times \overbrace{\left[\dim\left(\mathcal{P}_1^{(x_1,x_2)}\right)\right]}^{=3} \\
&= 6 = \mathrm{Card}\left(\{\mathcal{L}_1^e, \dots, \mathcal{L}_6^e\}\right).
\end{aligned}$$

Pour terminer la preuve que $(e, \vec{\mathcal{P}}_e, \{\mathcal{L}_1^e, \dots, \mathcal{L}_6^e\})$ est un élément fini, il reste à vérifier que les fonctions $\mathcal{L}_k^e : \vec{\mathcal{P}}_e \longrightarrow \mathbb{R}$ sont linéaires (pour $k = 1, \dots, 6$). Mais, ceci se montre facilement avec les mêmes idées qu'au TD no. 1 et la preuve est donc terminée.

2) On a

$$\vec{\mathcal{P}}_a = \{ \vec{q} = \vec{p}|_a \text{ tel que } \vec{p} \in \vec{\mathcal{P}}_e \}.$$

$$= \{ \vec{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \text{ tel que } q_i = p_i|_a \text{ pour un certain } p_i \in \mathcal{P}_e \ (\forall i = 1,2) \}$$

(de l'exercice 1)

$$= \{ \vec{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \text{ tel que } q_i \in \mathcal{P}_a \ (\forall i = 1,2) \}$$

(d'après l'exercice 1)

$$= \{ \vec{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \text{ tel que } q_i \in \mathcal{P}_1^s \ (\forall i = 1,2) \},$$

d'où $\boxed{\vec{\mathcal{P}}_a = \mathcal{P}_1^s \times \mathcal{P}_1^s .}$

Pour les ddls, les choix de leurs numérotations aux figures 1 et 2 de l'énoncé, signifient que

$$\boxed{\left\{\begin{array}{l} \mathcal{L}_1^a(\vec{q}) = q_1(S_1^a), \\ \mathcal{L}_2^a(\vec{q}) = q_1(S_2^a), \\ \mathcal{L}_3^a(\vec{q}) = q_2(S_1^a), \\ \mathcal{L}_4^a(\vec{q}) = q_2(S_2^a). \end{array}\right.}$$

3-a) Montrons que $\vec{N}_1^e = \begin{pmatrix} N_1^e \\ 0 \end{pmatrix}$, avec les notations de l'énoncé. On rappelle que les fonctions d'interpolation de base de l'élément fini $(e, \mathcal{P}_e, \{L_1^e, L_2^e, L_3^e\})$ de l'exercice 1 sont les uniques solutions des systèmes

$$(\mathcal{G}_m) \quad \left\{\begin{array}{l} N_m^e \in \mathcal{P}_e = \text{Vect}\{1, x_1, x_2\}, \\ L_k^e(N_m^e) = N_m^e(S_k) = \delta_{km} \quad \forall k = 1, 2, 3, \end{array}\right.$$

pour tout $m = 1, 2, 3$. Pour celles de l'élément fini $(e, \vec{\mathcal{P}}_e, \{\mathcal{L}_1^e, \dots, \mathcal{L}_6^e\})$ de l'exercice 2, on a

$$\begin{cases} \vec{N}^e_m \in \vec{\mathcal{P}}_e = \left(\mathrm{Vect}\{1, x_1, x_2\}\right)^2, \\ \mathcal{L}^e_k(\vec{N}^e_m) = \delta_{km} \quad \forall k = 1, \dots, 6, \end{cases}$$

pour tout $m = 1, \dots, 6$. On a donc, en notant

$$\vec{N}^e_1 = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix},$$

les équations suivantes pour p_1 et p_2 :

$$\begin{cases} p_1 \in \mathrm{Vect}\{1, x_1, x_2\} \\ \mathcal{L}^e_1(\vec{N}^e_1) = 1 = p_1(S_1), \\ \mathcal{L}^e_2(\vec{N}^e_1) = 0 = p_1(S_2), \\ \mathcal{L}^e_3(\vec{N}^e_1) = 0 = p_1(S_3), \end{cases}$$

et de la même façon

$$\begin{cases} p_2 \in \mathrm{Vect}\{1, x_1, x_2\}, \\ p_2(S_1) = p_2(S_2) = p_2(S_3) = 0. \end{cases}$$

On déduit que p_1 résoud le système $(\mathcal{S}_1)$ précédent, d'où $p_1 = N^e_1$ par unicité de la solution. De même $p_2 = 0$ (polynôme nul) par l'unisolvance de l'exercice 1. Ceci termine la preuve que $\vec{N}^e_1 = \begin{pmatrix} N^e_1 \\ 0 \end{pmatrix}$.

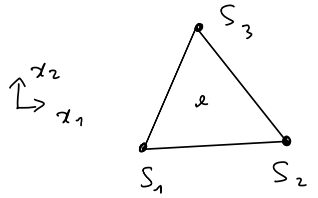
On raisonne de la même manière pour montrer que

$$\vec{N}_1^a = \begin{pmatrix} N_1^a \\ 0 \end{pmatrix}.$$

b) Donnons maintenant les formules de toutes les fonctions de base. C'est l'occasion de faire un bilan.

EF à valeurs scalaires (exercice 1)

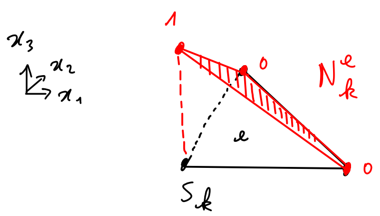
(Sur une maille)



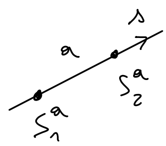
$P_e = \text{Vect}\{1, x_1, x_2\}$

$L_k^e(p) = p(S_k) \quad (k = 1,2,3)$

Base : $\boxed{L_k^e(N_m^e) = \delta_{km}}$



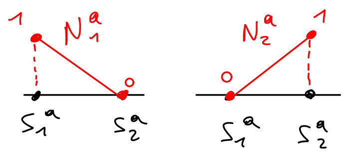
(Sur une arête)



$P_a = \text{Vect}\{1, s\}$

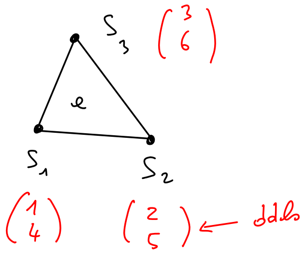
$L_k^a(q) = q(S_k^a) \quad (k = 1,2)$

Base : $\boxed{L_k^a(N_m^a) = \delta_{km}}$



$$\boxed{\text{EFs à valeurs vectorielles (exercice 2)}}$$

(Sur une maille)



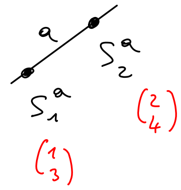
$$\vec{P}_e = \left(\mathrm{Vect}\{1, x_1, x_2\}\right)^2$$

$$\mathcal{L}_k^e\left(\begin{pmatrix}p_1\\p_2\end{pmatrix}\right) = \begin{cases} p_1(S_k) & \text{si } k=1,2,3 \\ p_2(S_{k-3}) & \text{si } k=4,5,6 \end{cases}$$

Base: $\boxed{\mathcal{L}_k^e(\vec{N}_m^e) = \delta_{km}}$

$$\boxed{\vec{N}_k^e = \begin{cases} \begin{pmatrix} N_k^e \\ 0 \end{pmatrix} & \text{si } k=1,2,3 \\ \begin{pmatrix} 0 \\ N_{k-3}^e \end{pmatrix} & \text{si } k=4,5,6 \end{cases}}$$

(Sur une arête)



$$\vec{P}_a = \left(\mathrm{Vect}\{1, s\}\right)^2$$

$$\mathcal{L}_k^a\left(\begin{pmatrix}q_1\\q_2\end{pmatrix}\right) = \begin{cases} q_1(S_k^a) & \text{si } k=1,2 \\ q_2(S_{k-2}^a) & \text{si } k=3,4 \end{cases}$$

Base: $\boxed{\mathcal{L}_k^a(\vec{N}_m^a) = \delta_{km}}$

$$\boxed{\vec{N}_k^a = \begin{cases} \begin{pmatrix} N_k^a \\ 0 \end{pmatrix} & \text{si } k=1,2 \\ \begin{pmatrix} 0 \\ N_{k-2}^a \end{pmatrix} & \text{si } k=3,4 \end{cases}}$$

Remarque

Dans la suite, nous n'aurons pas besoin de calculer les N_k^e de manière explicite. Nous utiliserons plutôt la matrice de passage avec la base canonique, puis nous calculerons les $N_k^e(x_1, x_2)$, pour n'importe quels x_1 et x_2, directement par ordinateur ; voir le cours sur la dimension 2, ainsi que les TPs no. 2 et 3. Néanmoins, pour ceux qui veulent connaître les formules explicites de ces fonctions, faites l'exercice 3 (qui n'est pas corrigé ici).

Exercice 4

Préliminaires

Voici des notions qui nous seront utiles pour la suite.

Définition

Soient $A, B \in \mathbb{R}^{2\times 2}$. On appelle « double produit contracté » de A et B, le réel noté $A : B$ et défini par

(2) $$A : B = \sum_{i,j=1}^{2} A_{ij}\, B_{ij}.$$

Remarque

C'est le produit scalaire canonique des matrices ; il est bilinéaire, symmétrique, etc.

Lemme (Formule de Green pour les systèmes)

Soient $\Omega \subseteq \mathbb{R}^2$, $\vec{\phi} : \overline{\Omega} \to \mathbb{R}^2$ et $\sigma : \overline{\Omega} \to \mathbb{R}^{2\times 2}$, un domaine et des fonctions suffisamment réguliers.

On note $\sigma = (\sigma_{ij})_{1 \le i,j \le 2}$ les applications composantes de σ.

On note aussi

$$\overrightarrow{\mathrm{div}}\,(\sigma) = \begin{pmatrix} \dfrac{\partial \sigma_{11}}{\partial x_1} + \dfrac{\partial \sigma_{12}}{\partial x_2} \\[2ex] \dfrac{\partial \sigma_{21}}{\partial x_2} + \dfrac{\partial \sigma_{22}}{\partial x_2} \end{pmatrix},$$

divergence de la 1ère ligne

2ème ligne

l'opérateur de divergence sur les lignes.

On note enfin

(3) $$\mathrm{Jac}(\vec{\phi}) = \begin{pmatrix} \dfrac{\partial \phi_1}{\partial x_1} & \dfrac{\partial \phi_1}{\partial x_2} \\[2ex] \dfrac{\partial \phi_2}{\partial x_1} & \dfrac{\partial \phi_2}{\partial x_2} \end{pmatrix}$$

la matrice Jacobienne de $\vec{\phi}$, où ϕ_1 et ϕ_2 désignent les applications composantes de $\vec{\phi}$; c'est-à-dire

$$\vec{\phi}(x_1, x_2) = \begin{pmatrix} \phi_1(x_1, x_2) \\ \phi_2(x_1, x_2) \end{pmatrix} \quad \forall (x_1, x_2) \in \overline{\Omega}.$$

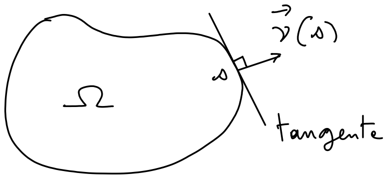
Alors,

(4) $$\boxed{-\int_\Omega \vec{\phi} \cdot \overrightarrow{\mathrm{div}}\,(\sigma) = \int_\Omega \sigma : \mathrm{Jac}(\vec{\phi}) - \int_{\partial\Omega} \vec{\phi} \cdot (\sigma \vec{\nu}),}$$

où $\vec{\nu} = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix}$ est la normale au bord, unitaire et extérieure,

et $\sigma\vec{\nu}$ le produit matriciel $\sigma\vec{\nu} = \begin{pmatrix} \sigma_{11}\nu_1 + \sigma_{12}\nu_2 \\ \sigma_{21}\nu_1 + \sigma_{22}\nu_2 \end{pmatrix}$.

Rappel :



Le vecteur $\vec{\nu}(s)$ est orthogonal à la tangente au bord $\partial\Omega$ de Ω, dirigé vers l'extérieur de Ω, et de norme $\|\vec{\nu}(s)\| = 1$.

Il est défini en tout point $s \in \partial\Omega$, ce qui nous donne un champ de vecteur

$$\vec{\nu} : s \in \partial\Omega \longmapsto \vec{\nu}(s) = \begin{pmatrix} \nu_1(s) \\ \nu_2(s) \end{pmatrix} \in \mathbb{R}^2.$$

Preuve (du lemme)

On va utiliser la formule de Green du cours (sur la dimension 2).

On a

$$-\int_\Omega \vec{\phi} \cdot \vec{\operatorname{div}}(\sigma) = \overbrace{-\int_\Omega \phi_1 \left(\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{12}}{\partial x_2} \right)}^{I_1 \text{ noté}} \underbrace{-\int_\Omega \phi_2 \left(\frac{\partial \sigma_{21}}{\partial x_1} + \frac{\partial \sigma_{22}}{\partial x_2} \right)}_{\text{noté } I_2}.$$

On $I_1 = -\int_\Omega \phi_1 \operatorname{div}(\vec{F})$, où $\operatorname{div}(\cdot)$ est la divergence classique et $\vec{F} : \bar{\Omega} \to \mathbb{R}^2$ le champ de vecteur $\vec{F} = (\sigma_{1j})_{1\le j\le 2}$. D'après la formule du cours, on a

$$I_1 = \int_\Omega \underbrace{\vec{F}\cdot\vec{\nabla}\phi_1}_{\sigma_{11}\frac{\partial \phi_1}{\partial x_1} + \sigma_{12}\frac{\partial \phi_1}{\partial x_2}} - \int_{\partial\Omega} \phi_1 \underbrace{\vec{F}\cdot\vec{\nu}}_{\sigma_{11}\nu_1 + \sigma_{12}\nu_2}.$$

De même, $I_2 = \int_\Omega \left(\sigma_{21}\frac{\partial \phi_2}{\partial x_1} + \sigma_{22}\frac{\partial \phi_2}{\partial x_2}\right) - \int_{\partial\Omega} \phi_2(\sigma_{21}\nu_1 + \sigma_{22}\nu_2)$.

En regroupant, on trouve que

$$I_1 + I_2 = \int_\Omega \left(\sum_{1\le i,j\le 2} \sigma_{ij}\frac{\partial \phi_i}{\partial x_j}\right)$$

$$- \int_{\partial\Omega} \begin{pmatrix}\phi_1\\ \phi_2\end{pmatrix} \cdot \begin{pmatrix}\sigma_{11}\nu_1 + \sigma_{12}\nu_2\\ \sigma_{21}\nu_1 + \sigma_{22}\nu_2\end{pmatrix},$$

ce qui est exactement le résultat voulu. ∎

On peut maintenant faire l'exercice.

1) – 4) <u>Formulation variationnelle</u> (Calculs formels)

On doit reformuler le problème $(\mathcal{P})$ à l'aide de fonctions test. Supposons donc que l'on ait une solution $\vec{u} : \overline{\Omega} \longrightarrow \mathbb{R}^2$ de

$$
\boxed{(\mathcal{P}) \begin{cases} -\overrightarrow{\mathrm{div}}(\sigma) = \vec{f} & \text{dans } \Omega, \quad (*) \\ \sigma \vec{\nu} = \vec{\beta} & \text{sur } \Gamma_N, \quad (**) \\ \vec{u} = \vec{0} & \text{sur } \Gamma_D, \quad (***) \end{cases}}
$$

où Ω, Γ_D, Γ_N, $\vec{f}$, etc., sont comme dans l'énoncé.

<u>Remarque</u> (Système d'EDPs d'ordre 2)

L'équation $(*)$ est un système d'équations aux dérivées partielles d'ordre 2, dont les inconnues sont les composantes $u_1 : \overline{\Omega} \to \mathbb{R}$ et $u_2 : \overline{\Omega} \to \mathbb{R}$ du déplacement $\vec{u}$. En effet, σ dépends des u_i par la formule

$$\sigma = 2\mu\, \varepsilon + \lambda \theta I$$

où $\lambda, \mu > 0$ sont des réels, $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$,

trace

$$\varepsilon = \varepsilon(\vec{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)_{1 \le i,j \le 2} \quad \text{et} \quad \theta = \mathrm{tr}(\varepsilon).$$

C'est de plus un système de 2 équations de la forme

$$(*) \iff \begin{cases} -\dfrac{\partial \sigma_{11}}{\partial x_1} - \dfrac{\partial \sigma_{12}}{\partial x_2} = f_1 & \text{dans } \Omega, \\ -\dfrac{\partial \sigma_{21}}{\partial x_1} - \dfrac{\partial \sigma_{22}}{\partial x_2} = f_2 & \text{———} . \end{cases}$$

On dit finalement qu'il est d'ordre 2, car il fait intervenir des dérivées d'ordre ≤ 2 des inconnues u_i.

Remarque (Conditions au bord)

La condition $(***)$ est une condition dite de Dirichlet, qui ne fait intervenir que des dérivées d'ordre 0 des inconnues.

Enfin, la condition $(**)$ est d'ordre 1 et est donc du type Neuman.

D'un point de vue mathématique, le problème (S) de cet exercice et celui avec le Laplacien du cours appartiennent tout les deux à une grande classe de problèmes, dits elliptiques. Le problème du cours en est le prototype. Bien que nous ayons maintenant un système d'EDPs, nous allons pouvoir utiliser les mêmes idées que celles qui ont été dévellopées en cours pour les équations scalaires.

Considérons donc des fonctions test

$$\vec{\phi} : \overline{\Omega} \longrightarrow \mathbb{R}^2,$$

telles que $\boxed{\vec{\phi} = \vec{0} \text{ sur } \Gamma_D.}$ Après multiplication de $(*)$ par $\vec{\phi}$ (avec le produit scalaire) et intégration, on a

$$\underbrace{-\int_{\Omega} \vec{\phi} \cdot \vec{\mathrm{div}}(\sigma)}_{\text{IPP}} = \int_{\Omega} \vec{f} \cdot \vec{\phi}.$$

$\shortparallel$ d'après (4)

$$\int_{\Omega} \sigma : \mathrm{Jac}(\vec{\phi}) - \underbrace{\int_{\partial\Omega} \vec{\phi} \cdot (\sigma \vec{\nu})}_{\shortparallel}$$

$$\underbrace{\int_{\Gamma_D} \cancel{\vec{\phi} \cdot \sigma \vec{\nu}}}_{\substack{\shortparallel \\ 0}} + \int_{\Gamma_N} \vec{\phi} \cdot \vec{\beta}$$

d'après Neuman

car $\vec{\phi} = 0$ sur Γ_D

C'est-à-dire,

$$\text{(5)}\quad \underbrace{\int_\Omega \sigma(\vec{u}) : \mathrm{Jac}(\vec{\phi})}_{\substack{\text{// noté} \\ a(\vec{u},\vec{\phi})}} = \overbrace{\int_\Omega \vec{f}\cdot\vec{\phi} + \int_{\Gamma_N} \vec{\beta}\cdot\vec{\phi}}^{\substack{\ell(\vec{\phi}) \\ \text{// noté}}}$$

$\forall \vec{u}$ solution de $(\mathcal{P})$, $\forall \vec{\phi}$ fonction test nulle sur Γ_D.

$\boxed{\text{Réécriture de } a(\vec{u},\vec{\phi})}$

On peut réécrire la forme $a(\vec{u},\vec{\phi})$ pour faire apparaître une symmétrie entre $\vec{u}$ et $\vec{\phi}$. Pour cela utilisons que

$$\sigma(\vec{u}) = 2\mu\, \varepsilon(\vec{u}) + \lambda\, \theta(\vec{u})\, I .$$

On obtient que

$$\sigma(\vec{u}) : \mathrm{Jac}(\vec{\phi}) = \left(2\mu\, \varepsilon(\vec{u}) + \lambda\, \theta(\vec{u})\, I\right) : \mathrm{Jac}(\vec{\phi})$$

$$\text{(6)}\quad = 2\mu \left(\varepsilon(\vec{u}) : \mathrm{Jac}(\vec{\phi})\right) + \lambda\, \theta(\vec{u}) \underbrace{\left(I : \mathrm{Jac}(\vec{\phi})\right)}_{\substack{\| \\ \frac{\partial \phi_1}{\partial x_1} + \frac{\partial \phi_2}{\partial x_2}}} .$$

le produit doublement contracté (2) est bilinéaire

voir (3)

On, en notant

$$\varepsilon(\vec{\phi}) = \frac{1}{2}\left(\frac{\partial \phi_i}{\partial x_j} + \frac{\partial \phi_j}{\partial x_i}\right)_{1\leq i,j\leq 2}$$

et $\theta(\vec{\phi}) = tr(\varepsilon(\vec{\phi}))$, on voit que

(7) $$\frac{\partial \phi_1}{\partial x_1} + \frac{\partial \phi_2}{\partial x_2} = tr(\varepsilon(\vec{\phi})) = \theta(\vec{\phi}).$$

De plus, toujours d'après (3),

$$\varepsilon(\vec{\phi}) = \frac{1}{2}\, Jac(\vec{\phi}) + \frac{1}{2}\left(Jac(\vec{\phi})\right)^T$$

où ^{T} désigne la transposée. Mais alors,

$$\varepsilon(\vec{u}) : Jac(\vec{\phi}) = \frac{1}{2}\left[\left(\varepsilon(\vec{u}) : Jac(\vec{\phi})\right) + \underbrace{\left(\varepsilon(\vec{u}) : Jac(\vec{\phi})\right)}\right]$$

d'après (2), il est facile de voir que $A:B = A^T:B^T$ pour toutes matrices A et B

$$= \left(\varepsilon(\vec{u})\right)^T : \left(Jac(\vec{\phi})\right)^T$$

$= \leftarrow$ $\varepsilon(\vec{u})$ est symétrique

$$\varepsilon(\vec{u}) : \left(Jac(\vec{\phi})\right)^T$$

Ce qui veut dire que

$$\varepsilon(\vec{u}) : Jac(\vec{\phi}) = \varepsilon(\vec{u}) : \left[\frac{1}{2}\left(Jac(\vec{\phi}) + \left(Jac(\vec{\phi})\right)^T\right)\right]$$

(8) $$= \varepsilon(\vec{u}) : \varepsilon(\vec{\phi}).$$

En injectant maintenant (7) et (8) dans (6),

on trouve que

(9) $$\boxed{a(\vec{u},\vec{\phi}) = 2\mu \int_\Omega \varepsilon(\vec{u}) : \varepsilon(\vec{\phi}) + \lambda \int_\Omega \theta(\vec{u})\, \theta(\vec{\phi}).}$$

Conclusion

Donnons maintenant la formulation variationnelle, motivée par (5).

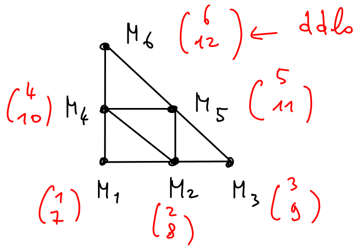
<u>Définition</u>

On dit que $\vec{u}$ est solution variationnelle de (P) si

$$\boxed{\begin{cases} \vec{u} \in \vec{V} = \left\{ \text{fonctions de } \overline{\Omega} \text{ dans } \mathbb{R}^2 \right\}, \\ a(\vec{u},\vec{\phi}) = \ell(\vec{\phi}) \quad \forall \vec{\phi} \in \vec{V} \text{ nulle sur } \Gamma_D, \\ \vec{u} = \vec{0} \quad \text{sur } \Gamma_D. \end{cases}}$$

5) <u>Maillage</u>

On considère le maillage et l'espace d'interpolation $\vec{V}_h$ de l'énoncé. Les numérotations



signifient qu'on munit $\vec{V}_h$ des ddls

$$\boxed{\mathcal{L}_i(\vec{u}_h) = \begin{cases} u_{h1}(M_i) & \text{si } i=1,\dots,6, \\ u_{h2}(M_{i-6}) & \text{si } i=7,\dots,12, \end{cases}}$$

où $\vec{u}_h$ désigne une fonction générale de $\vec{V}_h$ et

$$\boxed{\vec{u}_h = \begin{pmatrix} u_{h1} \\ u_{h2} \end{pmatrix}}$$

sont ses applications composantes.

Comme d'habitude, on peut voir que la famille

$\{\mathcal{L}_1, \dots, \mathcal{L}_{12}\}$ est un système de coordonnées sur V_h.

Plus précisément :

<u>Exercice</u> (à faire à la maison)

En utilisant les propriétés d'unisolvance de l'exercice 2,

vous montrerez que :

1) $\dim \vec{V}_h = \mathrm{Card}\{\mathcal{L}_1, \dots, \mathcal{L}_{12}\}$;

2) les $\mathcal{L}_i : \vec{V}_h \to \mathbb{R}$ sont linéaires ;

3) $(\forall \vec{u}_h \in \vec{V}_h) \ \left[\mathcal{L}_1(\vec{u}_h) = \dots = \mathcal{L}_{12}(\vec{u}_h) = 0 \Rightarrow \vec{u}_h = \vec{0}\right]$.

fonction nulle de $\bar{\Omega}$ dans $\mathbb{R}^2$

D'après la théorie du cours, voir l'annexe sur la dimension 1, on conclut que le triplet global

$$\left(\bar{\Omega}, \vec{V}_h, \{\mathcal{L}_1, \dots, \mathcal{L}_{12}\}\right)$$

est un « élément fini » et il admet une base d'interpolation. On la note $\{\vec{\phi}_1, \dots, \vec{\phi}_{12}\}$ dans la suite. On rappelle que pour tout $j = 1, \dots, 12$, $\vec{\phi}_j$ est l'unique solution du système

$$(10) \qquad \boxed{\begin{cases} \vec{\phi}_j \in \vec{V}_h, \\ \mathcal{L}_i(\vec{\phi}_j) = \delta_{ij} \quad \forall i = 1, \dots, 12. \end{cases}}$$

6) <u>Problème approché</u> (sous forme matricielle)

A partir de maintenant, nous allons considérer le problème approché de (3), qui est donné par la méthode de Galerkin. On rappelle qu'il s'agit de remplacer formellement $\vec{V}$ par $\vec{V}_h$ dans la formulation variationnelle. Cela conduit au problème de trouver une fonction $\vec{u}_h : \overline{\Omega} \longrightarrow \mathbb{R}^2$ telle que

$$\boxed{(\mathcal{P}_h) \begin{cases} \vec{u}_h \in \vec{V}_h , \\ a(\vec{u}_h , \vec{\phi}_h) = \ell(\vec{\phi}_h) \quad \forall \vec{\phi}_h \text{ nulle sur } \Gamma_D , \\ \vec{u}_h = \vec{0} \quad \text{sur } \Gamma_D . \end{cases}}$$

Notons $U_h \in \mathbb{R}^{12}$ les coordonnées de $\vec{u}_h$ sur la base d'interpolation. D'après le cours,

$$\boxed{U_h = \begin{pmatrix} \mathcal{L}_1(\vec{u}_h) \\ \vdots \\ \mathcal{L}_{12}(\vec{u}_h) \end{pmatrix} .}$$

De plus :

<u>Théorème</u>

La fonction $\vec{u}_h : \overline{\Omega} \longrightarrow \mathbb{R}^2$ résout $(\mathcal{P}_h)$ si et seulement si le vecteur U_h résout

$$\boxed{AU_h = F,}$$

où $A \in \mathbb{R}^{12 \times 12}$ et $F \in \mathbb{R}^{12}$ sont donnés par

$$\boxed{\begin{array}{l} A_{ij} = \begin{cases} a(\vec{\phi}_j, \vec{\phi}_i) & \text{si } i \notin I_D, \\ \delta_{ij} & \text{si } i \in I_D, \end{cases} \\ \\ F_i = \begin{cases} \ell(\vec{\phi}_i) & \text{si } i \notin I_D, \\ 0 & \text{si } i \in I_D, \end{cases} \end{array}}$$

avec $\boxed{I_D = \{ i \in \{1, \dots, 12\} \mid \vec{\phi}_i \not\equiv \vec{0} \text{ sur } \Gamma_D \}.}$

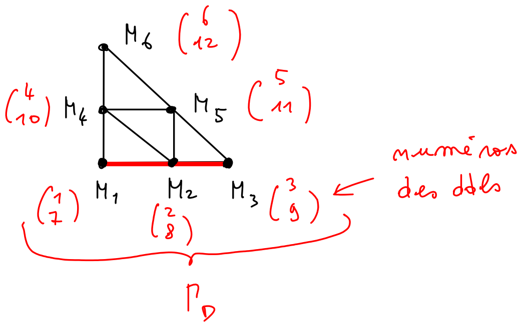
La preuve de ce résultat a été faite en cours pour la dimension 1. Pour le problème modèle que nous avions considéré en dimension 2, nous avions donné seulement une preuve formelle car les idées étaient très similaires. Ci-dessous, nous allons donner tout les arguments nécessaires à la dimension 2. Pour celà, montrons que :

<u>Lemme</u>

(11) $$\left[\vec{u}_h = \vec{0} \text{ sur } \Gamma_D \right] \Longleftrightarrow \left[\mathcal{L}_i(\vec{u}_h) = 0 \quad \forall i \in I_D \right].$$

<u>Preuve du lemme</u>

Calculons d'abord I_D. On a le maillage

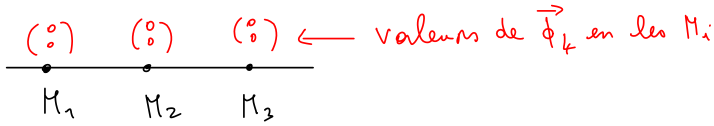


où chaque ddl $\mathcal{L}_j$ est associé à une fonction de base $\vec{\phi}_j$, étant l'unique solution de (10).

Par exemple, pour $j=4$, le système (10) s'écrit

$$\begin{cases} \vec{\phi}_4 \in \vec{V}_h, \\ \vec{\phi}_4(M_4) = \begin{pmatrix}1\\0\end{pmatrix}, \\ \vec{\phi}_4(M_i) = \begin{pmatrix}0\\0\end{pmatrix} \qquad \forall i \neq 4. \end{cases}$$

Donc, $\vec{\phi}_4$ prends les valeurs suivantes sur les noeuds de Γ_D :



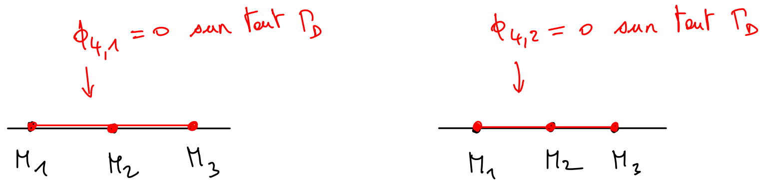
Mais on rappelle que $\vec{\phi}\Big|_e \in \left(\mathcal{P}_1(x_1,x_2)\right)^2$ pour chaque maille e, en tant que fonction de $\vec{V}_h$. On a aussi vu à l'exercice 2 qu'alors

$$\vec{\phi}_4\Big|_a \in \left(\mathbb{P}_1^1\right)^2,$$

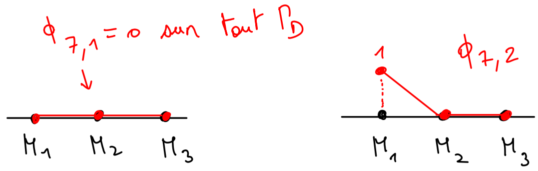
pour chaque arête a d'une maille. Ainsi, si on note

$$\vec{\phi}_4 = \begin{pmatrix} \phi_{4,1} \\ \phi_{4,2} \end{pmatrix},$$

les composantes $\phi_{4,i}$ sont des droites le long de chaque arête. Leurs graphes le long de Γ_D sont donc :



On en déduit que $4 \notin I_D$. En raisonnant de la même manière pour $\vec{\phi}_7 \underset{\text{noté}}{=} \begin{pmatrix} \phi_{7,1} \\ \phi_{7,2} \end{pmatrix}$, on a les graphes suivants le long de Γ_D :



Donc cette fois-ci $7 \in I_D$. On fait de même pour les autres $\vec{\phi}_i$ et on voit que $\boxed{I_D = \{1, 2, 3, 7, 8, 9\}.}$

Montrons maintenant (11). On a

$$\left[\vec{u}_h = \vec{0} \text{ sur } \Gamma_D\right] \iff \left[\vec{u}_h(M_i) = 0 \quad \forall i = 1,2,3\right]$$

Car on sait que les composantes de tout $\vec{u}_h \in \vec{V}_h$ sont des droites le long des arêtes

$$\iff \left[\begin{pmatrix} u_{h1}(M_i) \\ u_{h2}(M_i) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \forall i = 1,2,3\right]$$

d'après la définition des $\mathcal{L}_i$ à la question 5)

$$\iff \left[\begin{cases} \mathcal{L}_i(\vec{u}_h) = 0 \\ \mathcal{L}_{i+6}(\vec{u}_h) = 0 \end{cases} \quad \forall i = 1,2,3\right]$$

$$\iff \left[\mathcal{L}_i(\vec{u}_h) = 0 \quad \forall i \in I_D\right].$$

Ceci termine la preuve de (11). ∎

On est prêt pour démontrer le théorème précédent sur l'écriture matricielle de $(\mathcal{P}_h)$.

Preuve du théorème

Notons $\vec{V}_h^0 = \left\{\vec{\phi}_h \in \vec{V}_h \mid \vec{\phi}_h = \vec{0} \text{ sur } \Gamma_D\right\}$. D'après (11), il est facile de voir que $\boxed{\vec{V}_h^0 = \text{vect}\{\vec{\phi}_i \mid i \notin I_D\}.}$ Une fois que l'on a établie cette propriété, la preuve du théorème est alors exactement comme celle du cours sur la dimension 1. ∎

7) On considère les matrices Elements et Aretes de l'énoncé. On rappelle qu'elles définissent des correspondances entre les numéros globaux et locaux des ddls par les formules :

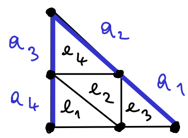
$$\boxed{\begin{cases} i = \text{Elements}(e,k) \iff \vec{\phi}_i\big|_e = \vec{N}^e_k, \\ i = \text{Aretes}(a,k) \iff \vec{\phi}_i\big|_a = \vec{N}^a_k, \end{cases}}$$

pour tout $i = 1, \dots, 12$, toutes mailles e et arêtes a, et tout k variant de 1 à 6 et de 1 à 4 respectivement sur les mailles et sur les arêtes. On rappelle aussi que les $\vec{N}^e_k$ et $\vec{N}^a_k$ sont les fonctions de l'exercice 2.

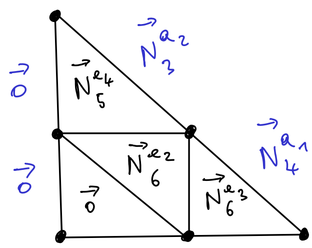
Remarque : il est possible de choisir d'autres correspondances, avec la seule contrainte d'avoir des bijections entre les numéros globaux et locaux sur chaque maille et arête.

Exprimons par exemple $\vec{\phi}_{11}$ en fonction des $\vec{N}^e_k$ et $\vec{N}^a_k$.

Rappelons d'abord les numéros



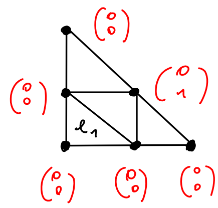
Ensuite, d'après Elements et Aretes, $\vec{\phi}_{11}$ corresponds aux fonctions de bases locales suivantes sur ces mailles et arêtes :



Liens entre $\vec{\phi}_{11}$ et les $\vec{N}_k^e$ et $\vec{N}_k^a$

Remarque

$*$ On a bien $\vec{\phi}_{11} = \vec{0}$ sur e_1, car elle prends les valeurs suivantes aux nœuds



et $\vec{\phi}_{11}\Big|_{e_1} \in \vec{\mathcal{P}}_{e_1}$

(d'où par l'unisolvance de l'exercice 2, $\vec{\phi}_{11} = \vec{0}$ sur e_1).

$*$ De même, $\vec{\phi}_{11} = \vec{0}$ sur a_3 et a_4.

8) Si $i \notin I_D$, on a

(voir (9))

$$A_{ij} = a(\vec{\phi}_j, \vec{\phi}_i) = 2\mu \int_\Omega \varepsilon(\vec{\phi}_j) : \varepsilon(\vec{\phi}_i) + \lambda \int_\Omega \theta(\vec{\phi}_j)\,\theta(\vec{\phi}_i),$$

(voir (5))

$$F_i = \ell(\vec{\phi}_i) = \int_\Omega \vec{f} \cdot \vec{\phi}_i + \int_{\Gamma_N} \vec{\beta} \cdot \vec{\phi}_i .$$

Le calcul des ces coefficients se fera par assemblage de coefficients élémentaires, comme pour le problème du cours. On rappelle que les coefficients élémentaires se définissent comme des versions locales des formules précédentes en remplaçant Ω et Γ_N par les mailles e et les arêtes a, respectivement, puis les fonctions de base globales par leurs expressions locales. Celà donne tout les coefficients suivants :

pour les mailles

$$\begin{cases} A^e_{km} = 2\mu \int_e \varepsilon(\vec{N}^e_m) : \varepsilon(\vec{N}^e_k) + \lambda \int_e \theta(\vec{N}^e_m)\,\theta(\vec{N}^e_k), \\ F^e_k = \int_\Omega \vec{f} \cdot \vec{N}^e_k , \end{cases}$$

pour les arêtes

$$\begin{cases} A^a_{km} = 0 , \\ F^a_k = \int_a \vec{\beta} \cdot \vec{N}^a_k . \end{cases}$$

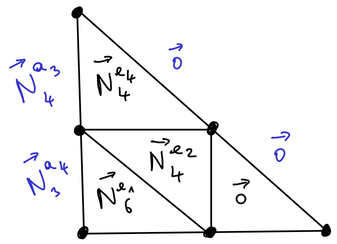
Il n'y a pas d'intégrale sur Γ_N dans la formule de A_{ij}.

Celà permet d'écrire les A_{ij} et F_i comme des sommes de coefficients élémentaires. Par exemple, pour F_{10}, On a :

$$F_{10} \overset{i \notin I_D}{=} \ell(\vec{\phi}_{10})$$

$$= \underbrace{\int_\Omega \vec{f} \cdot \vec{\phi}_{10}}_{\displaystyle\sum_{\text{mailles } e} \int_e \cdots} + \underbrace{\int_{\Gamma_N} \vec{\beta} \cdot \vec{\phi}_{10}}_{\displaystyle\sum_{\text{arêtes } a} \int_a \cdots} .$$

Or, d'après Elements et Aretes, $\vec{\phi}_{10}$ corresponds aux fonctions de bases locales suivantes :



En remplaçant $\vec{\phi}_{10}$ par ces $\vec{N}_k^e$ et $\vec{N}_k^a$, on trouve que

$$F_{10} = \int_{e_1} \vec{f} \cdot \vec{N}_6^{e_1} + \int_{e_2} \vec{f} \cdot \vec{N}_4^{e_2} + \cdots ,$$

ie $\boxed{F_{10} = F_6^{e_1} + F_4^{e_2} + F_4^{e_4} + F_4^{a_3} + F_3^{a_4} .}$

De même $\boxed{A_{11,6} = A_{53}^{e_4} \Big(+ \underbrace{A_{32}^{a_3}}_{0} \Big) .}$

0 car tout les A_{km}^a sont nuls

(Voir le TP no. 3 pour le programme d'assemblage de tout les autres coefficients A_{ij} et F_i.)

9) <u>Calcul de F^a</u>

Intégrons maintenant les F_k^a de manière approchée, par Simpson par exemple.

On doit calculer tout le vecteur

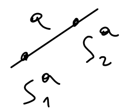
$$F^a = \begin{pmatrix} F_1^a \\ F_2^a \\ F_3^a \\ F_4^a \end{pmatrix} \in \mathbb{R}^4.$$

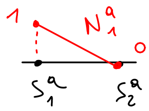
On a
$$F_1^a = \int_a \vec{\beta} \cdot \vec{N_1^a}$$
$$= \int_a \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \cdot \begin{pmatrix} N_1^a \\ 0 \end{pmatrix}$$

exercice

$$= \int_a \beta_1 N_1^a$$
$$\simeq \frac{|a|}{6} \Big(\beta_1(S_1^a)\, N_1^a(S_1) + 4\, \beta_1(m^a)\, N_1^a(m^a) + \beta_1(S_2^a)\, N_1^a(S_2^a) \Big),$$

Simpson

où $|a| = \|S_2^a - S_1^a\|$ est la longueur de l'arête 

et $m^a = \dfrac{S_1^a + S_2^a}{2}$ son milieu. On rappelle aussi que $N_1^a \in \mathrm{Vect}\{1, x\}$ est une droite de graphe 

(voir le bilan final de l'exercice 2).

Donc $N_1^a(S_1^a) = 1$, $N_1^a(m^a) = \frac{1}{2}$ et $N_1^a(S_2^a) = 0$.

Par suite,

$$\boxed{F_1^a \simeq \frac{|a|}{6}\left(\beta_1(S_1^a) + 2\,\beta_1(m^a)\right).}$$

De même

$$\boxed{F^a \simeq \frac{|a|}{6}\begin{pmatrix} \beta_1(S_1^a) + 2\,\beta_1(m^a) \\ \beta_1(S_2^a) + 2\,\beta_1(m^a) \\ \beta_2(S_1^a) + 2\,\beta_2(m^a) \\ \beta_2(S_2^a) + 2\,\beta_2(m^a) \end{pmatrix}.}$$

(Voir le TP no. 3 pour la suite.)

2.2.2 TP no. 3 : programmation

(Tournez la page.)

TP no. 3

Approximation des EDPs, Printemps 2016-2017

Continuons le problème du barrage du TD no. 3. On se propose de calculer le déplacement approché $\vec{u_h}$ par matlab avec l'algorithme « classique » d'assemblage pour les éléments finis.

Table des matières

1 Calcul de A^a et F^a

Soit a une arête du maillage comme ci-dessous :

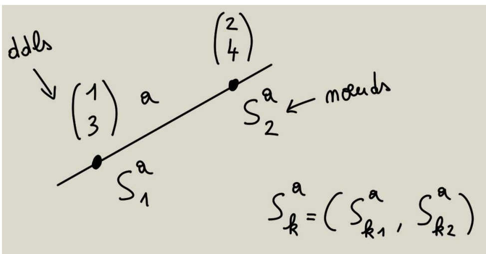


Figure 1 – Noeuds et ddls d'une arête.

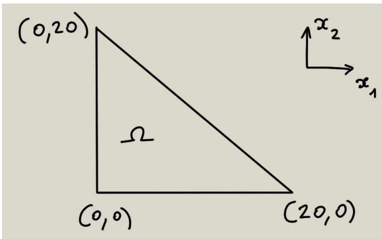
On rappelle que la matrice élémentaire $A^a \in \mathbb{R}^{4\times 4}$ et le vecteur élémentaire $F^a \in \mathbb{R}^4$ sont définis par :

$$A^a = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix} \quad \text{et} \quad F^a = \frac{|a|}{6} \begin{pmatrix} \beta_1(S_1^a) + 2\,\beta_1(m^a) \\ \beta_1(S_2^a) + 2\,\beta_1(m^a) \\ \beta_2(S_1^a) + 2\,\beta_2(m^a) \\ \beta_2(S_2^a) + 2\,\beta_2(m^a) \end{pmatrix}, \qquad (1)$$

où $m^a = \frac{S_1^a + S_2^a}{2}$ est le milieu de a et $|a|$ sa longueur. On rappelle aussi que β_1 et β_2 sont les applications composantes de la force

$$\vec{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} : \Gamma_N \to \mathbb{R}^2.$$

On prendra un barrage de longueur et hauteur $2L = 20$ mètres, comme ci-dessous.



On a alors

$$\vec{\beta}\,(x_1, x_2) = \begin{cases} \begin{pmatrix} \rho g(2L - x_2) \\ 0 \end{pmatrix} & \text{si } x_1 = 0, \\ \vec{0} & \text{sinon,} \end{cases}$$

où $\rho = 1000$ (kg/m^3) est la masse volumique de l'eau et $g \simeq 10$ (N/kg) l'intensité de la pesanteur (voir le calcul des structures).

1. Copiez/collez les programmes du TP no. 1 dans un nouveau dossier `TP3`.

   Remarque. Vous pouvez effacer `alpha.m`, `c.m` et `f.m`, car vous n'en aurez pas besoin.

2. Créez et complétez la fonction `betaa.m` qui suit, dont les variables de sorties seront les composantes β_1 et β_2 de la fonction $\vec{\beta}$.

betaa.m

```
function [beta1,beta2]=betaa(x1,x2)

rho=1000;g=10;L=10;

% A COMPLETER
```

Vérification. Pour calculez $\vec{\beta}(x_1, x_2)$ avec $x_1 = 0$ et $x_2 = 10$, vous devez taper

```
> [beta1,beta2]=betaa(0,10)
```

Vérifiez que vous obtenez les bonnes valeurs. Vérifiez aussi que vous obtenez le vecteur nul pour $x_1 \neq 0$ (par exemple avec $x_1 = x_2 = 10$).

3. Vous allez devoir créez une fonction `AaFa.m` qui calcule A^a et F^a, qui ont été définis en (1). La variable d'entrée sera la matrice 4×2 suivante :

$$\texttt{Noeuds\_loc} = \begin{pmatrix} S^a_{11} & S^a_{12} \\ S^a_{21} & S^a_{22} \\ S^a_{11} & S^a_{12} \\ S^a_{21} & S^a_{22} \end{pmatrix},$$

(chaque $k^{\text{ème}}$ ligne étant l'abscisse et l'ordonnée du noeud supportant le $k^{\text{ème}}$ ddl de l'arête a, comme sur la figure 1).

Complétez maintenant cette fonction, comme ci-dessous.

AaFa.m

```
function [Aa,Fa]=AaFa(Noeuds_loc)

%% Noeuds de l'arête

Sa=Noeuds_loc([1 2],:);% abscisses et ordonnées des noeuds

%% Milieu et longueur de l'arête

ma=% A COMPLETER
h=% A COMPLETER

%% Calcul de Aa et Fa

% A COMPLETER
```

<u>Vérification.</u> Si $S_1^a = (0,10)$ et $S_2^a = (0,0)$, on a

$$A^a \simeq \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad F^a \simeq 10^5 \times \begin{pmatrix} 6.6667 \\ 8.3333 \\ 0 \\ 0 \end{pmatrix}.$$

Vérifier que vous obtenez les bons résultats en tapant

```
> [Aa,Fa]=AaFa([0 10;0 0;0 10;0 0])
```

2 Calcul de A^e et F^e

Soit une maille e et les coefficients élémentaires définis par

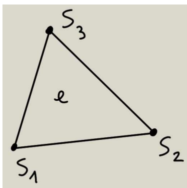
$$A^e_{kn} = \int_e \left\{ 2\,\mu \left(\epsilon(\vec{N^e_n}) : \epsilon(\vec{N^e_k}) \right) + \lambda\, \theta(\vec{N^e_n})\, \theta(\vec{N^e_k}) \right\}, \tag{2}$$

avec les coefficients de Lamé

$$\mu \simeq 25 \times 10^9 \quad \text{et} \quad \lambda \simeq 11.11 \times 10^9 \quad (N/m^2)$$

(voir le calcul des structures). On rappelle que :

- e est un triangle de la forme



- les matrices $\epsilon(\vec{N^e_k}) \in \mathbb{R}^{2\times 2}$ se calculent par

$$\epsilon_{ij}(\vec{N^e_k}) = \frac{1}{2}\left(\frac{\partial p_i}{\partial x_j} + \frac{\partial p_j}{\partial x_i} \right) \quad \text{où} \quad \vec{N^e_k} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix};$$

- leurs traces sont notées par

$$\theta(\vec{N^e_k}) = \mathrm{tr}\left(\epsilon(\vec{N^e_k}) \right);$$

- les $\vec{N^e_k}$ se calculent par

$$\vec{N^e_k} = \begin{cases} \begin{pmatrix} N^e_k \\ 0 \end{pmatrix} & \text{si } k = 1,2,3, \\ \begin{pmatrix} 0 \\ N^e_{k-3} \end{pmatrix} & \text{si } k = 4,5,6, \end{cases} \tag{3}$$

où $\{N^e_1, N^e_2, N^e_3\}$ est la base d'interpolation de $\mathcal{P}_1 = \mathrm{vect}\,\{1, x_1, x_2\}$ associée aux ddls

$$L^e_k : p \in \mathcal{P}_1 \mapsto p(S_k) \quad (\text{pour } k = 1,2,3).$$

1. On rappelle que pour tout $p \in \mathcal{P}_1$,

$$p(x_1, x_2) = \sum_{k=1}^{3} \overbrace{L_k^e(p)}^{=p(S_k)} \, N_k^e(x_1, x_2) \quad \forall x_1, x_2 \in \mathbb{R}.$$

   Appliquez cette formule pour les polynômes de la base canonique et déduisez en que pour tout $x_1, x_2 \in \mathbb{R}$,

$$\begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} = \underbrace{\begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}}_{\text{noté } \mathtt{M}} \times \begin{pmatrix} N_1^e(x_1, x_2) \\ N_2^e(x_1, x_2) \\ N_3^e(x_1, x_2) \end{pmatrix},$$

   où vous préciserez les $*$ en notant (S_{k1}, S_{k2}) les coordonnées de S_k.

2. En déduire que

$$\begin{pmatrix} * \\ * \\ * \end{pmatrix} = \mathtt{M} \times \underbrace{\begin{pmatrix} \frac{\partial N_1^e}{\partial x_1} \\ \frac{\partial N_2^e}{\partial x_1} \\ \frac{\partial N_3^e}{\partial x_1} \end{pmatrix}}_{\text{noté } \mathtt{dNedx1}} \tag{4}$$

   et

$$\begin{pmatrix} * \\ * \\ * \end{pmatrix} = \mathtt{M} \times \underbrace{\begin{pmatrix} \frac{\partial N_1^e}{\partial x_2} \\ \frac{\partial N_2^e}{\partial x_2} \\ \frac{\partial N_3^e}{\partial x_2} \end{pmatrix}}_{\text{noté } \mathtt{dNedx2}}, \tag{5}$$

   où vous préciserez les $*$.

   Remarque. Les $\partial N_k^e / \partial x_i$ ne dépendent plus de x_1 et x_2, car $N_k^e \in \text{vect}\{1, x_1, x_2\}$; en particulier, l'intégrale de (2) est égal à

$$A_{kn}^e = \underbrace{|e|}_{\text{aire de } e} \times \underbrace{\left\{ 2\,\mu \left(\epsilon(\overrightarrow{N_n^e}) : \epsilon(\overrightarrow{N_k^e}) \right) + \lambda \, \theta(\overrightarrow{N_n^e}) \, \theta(\overrightarrow{N_k^e}) \right\}}_{\text{fonction constante indépendante de } x_1 \text{ et } x_2} . \tag{6}$$

3. Vous allez commencer par implémenter le calcul de A^e dans la fonction `AeFe.m`. La variable d'entrée sera la matrice 6×2 suivante :

$$\mathtt{Noeuds\_loc} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \\ S_{31} & S_{32} \\ S_{11} & S_{12} \\ S_{21} & S_{22} \\ S_{31} & S_{32} \end{pmatrix}$$

(chaque $k^{\text{ème}}$ ligne étant l'abscisse et l'ordonnée du noeud supportant le $k^{\text{ème}}$ ddl de e).
Complétez maintenant les calculs ci-dessous, à l'aide de (4)–(5).

AeFe.m

```
function [Ae,Fe]=AeFe(Noeuds_loc)

mu=25*10^9;
lambda=11.11*10^9;

S=Noeuds_loc([1 2 3],:);% abscisses et ordonnées des noeuds

aire=abs(det([S(2,:)-S(1,:);S(3,:)-S(1,:)])/2);% aire de e

M=% A COMPLETER

%% Calcul des derivées des N^e_k

dNedx1=% A COMPLETER
dNedx2=% A COMPLETER
```

4. Faites un petit calcul sur votre feuille de brouillon pour compléter les formules ci-dessous, à partir de (3) :

$$\epsilon(\overrightarrow{N_k^e}) = \begin{cases} \begin{pmatrix} \frac{\partial N_k^e}{\partial x_1} & \frac{1}{2}\frac{\partial N_k^e}{\partial x_2} \\ * & * \end{pmatrix} & \text{si } k = 1, 2, 3, \\ \\ \begin{pmatrix} * & * \\ * & * \end{pmatrix} & \text{si } k = 4, 5, 6. \end{cases}$$

5. Sur matlab, on notera

$$\texttt{eps(i,j,k)} = \epsilon_{i,j}(\overrightarrow{N_k^e}),$$

où
- `i` est l'indice de ligne,
- `j` de colonne,
- et `k` de page.

En fixant k et en faisant varier i et j, on obtient la matrice 2×2

$$\texttt{eps(:,:,k)} = \epsilon(\overrightarrow{N_k^e}),$$

dont on notera la trace par

$$\texttt{theta(k)} = \theta(\overrightarrow{N_k^e}).$$

Complétez les calculs de ces matrices puis de leurs traces ci-dessous.

AeFe.m

```
⋮

%% Tenseurs des déformations

eps=zeros(2,2,6);
for k=1:3
       eps(:,:,k)=[dNedx1(k) dNedx2(k)/2;% A COMPLETER
       eps(:,:,k+3)=% A COMPLETER
end

%% Traces

theta=zeros(6,1);
for k=1:6
       theta(k)=trace(% A COMPLETER
end
```

Remarque. Etant donnée une matrice `A` sur matlab, on peut calculer sa trace par la commande `trace(A)`.

6. Il reste à implémenter la formule (6). Pour calculer le double produit contracté entre deux matrices A et B, on peut utiliser la formule :

$$A : B = \mathrm{tr}\,(A\,B^{\mathrm{T}})\,,$$

où B^{T} désigne la matrice transposée de B. On peut donc réécrire (6) de la manière suivante :

$$A^e_{kn} = |e| \left\{ 2\,\mu\,\mathrm{tr}\left(\epsilon(\overrightarrow{N^e_n})\,\epsilon(\overrightarrow{N^e_k}) \right) + \lambda\,\theta(\overrightarrow{N^e_n})\,\theta(\overrightarrow{N^e_k}) \right\}$$

(car $\epsilon(\overrightarrow{N^e_k})$ est symmétrique).
En déduire la fin du calcul de `Ae` ci-dessous.

AeFe.m

```
⋮
%% Intégration des coefficients élémentaires

Ae=zeros(6,6);
for k=1:6
       for n=1:6
              Ae(k,n)=aire*(2*mu*trace(% A COMPLETER
       end
end
```

7. Terminez par le calcul de F^e ci-dessous.

AeFe.m

```
⋮

Fe=zeros(6,1);
```

(Ce vecteur est nul car la force exercée par la pression de l'eau intervient seulement dans $\vec{\beta}$; voir le calcul des structures.)

Vérification. Si $S_1 = (0,0)$, $S_2 = (10,0)$ et $S_3 = (0,10)$, on doit trouver

$$\texttt{Ae} \simeq 10^{10}\times$$
$$\begin{pmatrix} 4.3055 & -3.0555 & -1.2500 & 1.8055 & -1.2500 & -0.5555 \\ -3.0555 & 3.0555 & 0 & -0.5555 & 0 & 0.5555 \\ -1.2500 & 0 & 1.2500 & -1.2500 & 1.2500 & 0 \\ 1.8055 & -0.5555 & -1.2500 & 4.3055 & -1.2500 & -3.0555 \\ -1.2500 & 0 & 1.2500 & -1.2500 & 1.2500 & 0 \\ -0.5555 & 0.5555 & 0 & -3.0555 & 0 & 3.0555 \end{pmatrix}$$

(F^e étant le vecteur nul). Vérifiez que vous obtenez les bons résultats en tapant

```
> [Ae,Fe]=AeFe([0 0;10 0;0 10;0 0;10 0;0 10])
```

3 Maillage

On reprends le maillage du TD, voir la figure 2. Vous allez éditer, « à la main », les fichier de données qui définissent ce maillage.

1. La variable `Noeuds` contiendra les abscisses et les ordonnées des noeuds $M_{\mathscr{L}_i}$ qui supportent chaque ddl $\mathscr{L}_i$. Celà donnera une matrice 12×2 de la forme suivante :

$$\texttt{Noeuds} = \begin{pmatrix} M_{\mathscr{L}_1} \\ \vdots \\ M_{\mathscr{L}_{12}} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 20 \end{pmatrix}.$$

   Stockez là dans le fichier `Noeuds.dat` (en complétant les pointillés).

2. Editez maintenant les fichiers `Elements.dat`, `Aretes.dat` et `ID.dat`.

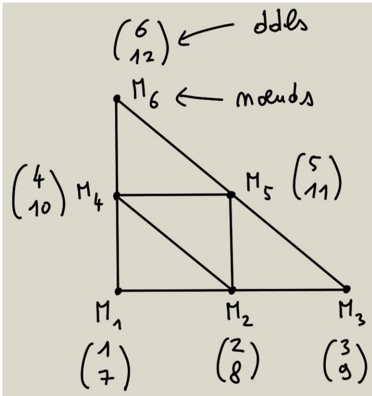


FIGURE 2 – Maillage ; voir `Elements` et `Aretes` pour les numérotations des éléments et des arêtes.

Rappels. On a vu en TD que

$$\texttt{Elements} = \begin{pmatrix} 1 & 2 & 4 & 7 & 8 & 10 \\ 4 & 2 & 5 & 10 & 8 & 11 \\ 2 & 3 & 5 & 8 & 9 & 11 \\ 4 & 5 & 6 & 10 & 11 & 12 \end{pmatrix},$$

$$\texttt{Aretes} = \begin{pmatrix} 3 & 5 & 9 & 11 \\ 5 & 6 & 11 & 12 \\ 6 & 4 & 12 & 10 \\ 4 & 1 & 10 & 7 \end{pmatrix} \quad \text{et} \quad \texttt{ID} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 7 \\ 8 \\ 9 \end{pmatrix}.$$

4 Assemblage

1. Commencez par mettre la dernière ligne du programme `Principal.m` en commentaire (l'ancien programme de dessin n'étant pas adapté à notre problème).

Principal.m

```
⋮
%% Dessin de la solution approchée
%Dessin;
```

2. Modifiez la ligne ci-dessous (qui prends en compte le fait que les noeuds ont maintenant deux coordonnées : abscisse et ordonnée).

Principal.m

```
⋮
%% Assemblage : contribution de l'équation
⋮
                Noeuds_loc=Noeuds(Elements(e,:),:);
⋮
```

3. Complétez la condition de Dirichlet ci-dessous (signifiant que le barrage est fixé sur Γ_D).

Principal.m

```
⋮
%% Assemblage : contribution de Dirichlet
⋮
        F(i)=0;
⋮
```

4. Rajoutez la contribution de Neuman dans le programme `Principal.m` (voir l'algorithme du cours).
5. Calculez les déplacements des noeuds en exécutant `Principal.m` et en tapant `> Uh`

Vérification. Vous devez trouver

$$U_h \simeq 10^{-4} \begin{pmatrix} -0.0000 \\ 0.0000 \\ 0.0000 \\ 0.4347 \\ 0.3357 \\ 0.7408 \\ 0.0000 \\ -0.0000 \\ 0.0000 \\ 0.1212 \\ -0.0516 \\ 0.1392 \end{pmatrix}.$$

5 Dessin du barrage

Pour continuer, vous devez lire attentivement l'appendice A avant de suivre les étapes ci-dessous.

1. Effacez le contenu du programme `Dessin.m` pour le remplacer par les lignes de code ci-dessous. Vous complèterez ce code pour dessiner chaque maille e, à la fois avant et après son déplacement. Vous utiliserez respectivement les options `'k'` et `'r--'` pour différencier la position initiale de la position d'équilibre.

Dessin.m

```
clf;
hold on;
grid on;

Coef=5000;% amplifie les déplacements pour le dessin

Nelem=size(Elements,1);

for e=1:Nelem

        %% Dessin de l'élément avant déplacement

        Sommets=Noeuds(Elements(e,[1 2 3 1]),:);
        % A COMPLETER

        %% après déplacement

        Deplacements=zeros(4,2);
        Deplacements(:,1)=Uh(Elements(e,[1 2 3 1]));% en x1
        Deplacements(:,2)=% A COMPLETER

        Equilibre=Sommets+Coef*Deplacements;
        % A COMPLETER

end

%% Légendes

legend('avant déplacement','après');
```

2. Remettez enfin la dernière ligne de `Principal.m` comme une ligne de code :

```
⋮
Dessin
```

A Dessin d'un triangle

Soit e un triangle de sommets S_1, S_2 et S_3. Pour fixer les idées, disons que $S_1 = (1, 2)$, $S_2 = (2, 5)$ et $S_3 = (0, 4)$. Pour dessiner e avec matlab, on peut utiliser les lignes de codes ci-dessous.

```
> clf
> Sommets=[1 2;2 5;0 4;1 2]
> plot(Sommets(:,1),Sommets(:,2),'k','LineWidth',2)
```

La variable `Sommets` est la matrice des abscisses et des ordonnées de chaque sommets, c'est-à-dire :

$$\texttt{Sommets} = \begin{pmatrix} 1 & 2 \\ 2 & 5 \\ 0 & 4 \\ 1 & 2 \end{pmatrix}.$$

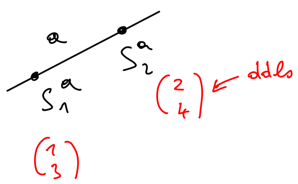
On a remis le premier sommet à la fin pour fermer le triangle. Pour dessiner le même triangle avec des traits rouges discontinus, utiliser l'option `'r--'` au lieu de `'k'`.

$\boxed{\text{Corrigé du TP no. 3}}$

(Voir la fin du corrigé pour les programmes ; ci-dessous on commente seulement quelques questions.)

$\boxed{\text{Section 1 : calcul de } A^a \ldots}$

3) La variable d'entrée Noeuds_loc de AaFa.m représente la matrice des coordonnées des noeuds qui supportent les ddls de l'arête. On rappelle qu'on a



$$\boxed{\begin{aligned} L_1^a(\vec{q}) &= q_1(S_1^a) \\ L_2^a(\vec{q}) &= q_1(S_2^a) \\ L_3^a(\vec{q}) &= q_2(S_1^a) \\ L_4^a(\vec{q}) &= q_2(S_2^a) \end{aligned}}$$

avec $S_k^a = (\underbrace{S_{k1}^a}_{\text{abscisse}}, \underbrace{S_{k2}^a}_{\text{ordonnée}})$.

D'où

$$\text{Noeuds\_loc} = \begin{pmatrix} S_{11}^a & S_{12}^a \\ S_{21}^a & S_{22}^a \\ S_{11}^a & S_{12}^a \\ S_{21}^a & S_{22}^a \end{pmatrix} \begin{array}{l} \leftarrow \text{coordonnées de } S_1^a \\ \leftarrow \text{———————— } S_2^a \\ \leftarrow \text{———————— } S_1^a \\ \leftarrow \text{———————— } S_2^a \end{array} .$$

Il peut paraître inutile de répéter ces noeuds deux fois, mais on a déjà vu au TP no. 1 qu'il est pratique de procéder ainsi pour retrouver facilement le noeud qui supporte n'importe quel ddl (notamment lors de l'assemblage dans Principal.m).

La variable

conserve les lignes 1 et 2

$$Sa = Noeuds\_loc([1\ \ 2];\ :)$$

conserve toutes les colonnes

représente alors simplement la matrice

$$Sa = \begin{pmatrix} S^a_{11} & S^a_{12} \\ S^a_{21} & S^a_{22} \end{pmatrix}$$

(sans répéter les noeuds).

(Voir les programmes corrigés à la fin pour d'autres explications concernant AaFa.m).

$\boxed{\text{Section 2 : calcul de } A^e \ldots}$

1-2) La matrice de passage est

$$\boxed{M = \begin{pmatrix} 1 & 1 & 1 \\ S_{11} & S_{21} & S_{31} \\ S_{12} & S_{22} & S_{32} \end{pmatrix}.}$$

En effet, on obtient la deuxième ligne, par exemple, en partant de la formule

$$p(x_1, x_2) = \sum_{k=1}^{3} L_k^e(p) \; N_k^e(x_1, x_2)$$

$$\forall p \in \text{vect}\{1, x_1, x_2\}, \quad \forall x_1, x_2 \in \mathbb{R},$$

que l'on applique au polynôme $p(x_1, x_2) = x_1$. On trouve

que
$$\begin{aligned} x_1 &= L_1^e(p) \; N_1^e(x_1, x_2) + \cdots \\ &= p(S_1) \; N_1^e(x_1, x_2) + \cdots \\ &= S_{11} \; N_1^e(x_1, x_2) + \cdots, \end{aligned}$$

en notant $S_k = (S_{k_1}, S_{k_2})$ les coordonnées des noeuds, etc.

En dérivant la formule

$$\boxed{\begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} = M \left(N_k^e(x_1, x_2) \right)_{1 \le k \le 3}}$$

par rapport à x_1 et x_2, on trouve que

$$\boxed{\begin{aligned} &\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = M \left(\frac{\partial N_k^e}{\partial x_1} \right)_{1 \le k \le 3} \\ \text{et} \quad &\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = M \left(\frac{\partial N_k^e}{\partial x_2} \right)_{1 \le k \le 3} . \end{aligned}}$$

4) On a

$$\boxed{\varepsilon(\vec{N}_k^e) = \begin{cases} \begin{pmatrix} \dfrac{\partial N_k^e}{\partial x_1} & \dfrac{1}{2} \dfrac{\partial N_k^e}{\partial x_2} \\ \dfrac{1}{2} \dfrac{\partial N_k^e}{\partial x_2} & 0 \end{pmatrix} & \text{si } k = 1, 2, 3, \\ \begin{pmatrix} 0 & \dfrac{1}{2} \dfrac{\partial N_{k-3}^e}{\partial x_1} \\ \dfrac{1}{2} \dfrac{\partial N_{k-3}^e}{\partial x_1} & \dfrac{\partial N_{k-3}^e}{\partial x_2} \end{pmatrix} & \text{si } k = 4, 5, 6 . \end{cases}}$$

En effet, calculons par exemple $\varepsilon_{ij}(\vec{N}^e_k)$ pour $k = 4,5,6$ et $i=1$ et $j=2$. D'après le TD no. 3, on a

$$\vec{N}^e_k = \begin{pmatrix} 0 \\ N^e_{k-3} \end{pmatrix} \overset{\text{noté}}{=} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

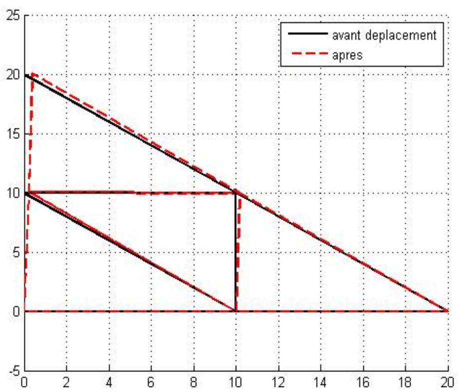
et de plus

$$\begin{aligned}
\varepsilon_{ij}(\vec{N}^e_k) = \varepsilon_{12}(\vec{N}^e_k) &= \frac{1}{2}\left(\frac{\partial p_1}{\partial x_2} + \frac{\partial p_2}{\partial x_1}\right) \\
&= \frac{1}{2}\left(0 + \frac{\partial N^e_{k-3}}{\partial x_1}\right) \\
&= \frac{1}{2}\frac{\partial N^e_{k-3}}{\partial x_1}.
\end{aligned}$$

On raisonne pareil pour les autres coefficients.

$\boxed{\text{Section 5 : dessin ...}}$

A la fin, vous devez obtenir le dessin du barrage suivant :



Programmes corrigés du TP no. 3

Scripts

Principal.m

```
%% Initialisation

clear all;
clc;

%% Generation du maillage

load Noeuds.dat;
load Elements.dat;
load ID.dat;
load Aretes.dat;

%% Initialisation de A et F

N=size(Noeuds,1);
A=zeros(N,N);
F=zeros(N,1);

%% Assemblage : contribution de l'equation

Nelem=size(Elements,1);
N_loc=size(Elements,2);

for e=1:Nelem

      Noeuds_loc=Noeuds(Elements(e,:),:);
      [Ae,Fe]=AeFe(Noeuds_loc);

      for k=1:N_loc
            i=Elements(e,k);
            F(i)=F(i)+Fe(k);
            for n=1:N_loc
                  j=Elements(e,n);
                  A(i,j)=A(i,j)+Ae(k,n);
            end
      end
end

%% Assemblage : contribution de Neuman

Naretes=size(Aretes,1);
N_loc=size(Aretes,2);

for a=1:Naretes

      Noeuds_loc=Noeuds(Aretes(a,:),:);
      [Aa,Fa]=AaFa(Noeuds_loc);

      for k=1:N_loc
            i=Aretes(a,k);
            F(i)=F(i)+Fa(k);
            for n=1:N_loc
                  j=Aretes(a,n);
                  A(i,j)=A(i,j)+Aa(k,n);
            end
      end
end
```

```
%% Assemblage : contribution de Dirichlet

CardID=size(ID,1);
for temp=1:CardID
      i=ID(temp);
      F(i)=0;% deplacement nul sur Gamma_D
      for j=1:N
            A(i,j)=0;
      end
      A(i,i)=1;
end

%% Resolution

Uh=A\F;

%% Representation graphique

Dessin;
```

__Dessin.m__

```
clf;
hold on;
grid on;

Coef=5000;% amplifie les deplacements pour le dessin

Nelem=size(Elements,1);

for e=1:Nelem

    %% Dessin de l'element avant deplacement

    Sommets=Noeuds(Elements(e,[1 2 3 1]),:);
    plot(Sommets(:,1),Sommets(:,2),'k','Linewidth',2);

    %% apres deplacement

    Deplacements=zeros(4,2);
    Deplacements(:,1)=Uh(Elements(e,[1 2 3 1]));% en x1
    Deplacements(:,2)=Uh(Elements(e,[4 5 6 4]));% en x2

    Equilibre=Sommets+Coef*Deplacements;
    plot(Equilibre(:,1),Equilibre(:,2),'r--','Linewidth',2);

end

%% Legendes

legend('avant deplacement','apres');
```

Fonctions

__AeFe.m__

```
function [Ae,Fe]=AeFe(Noeuds_loc)

mu=25*10^9;
lambda=11.11*10^9;

S=Noeuds_loc([1 2 3],:);% abscisses et ordonnees des noeuds
```

```
aire=abs(det([S(2,:)-S(1,:);S(3,:)-S(1,:)])/2);% aire de e

M=[1 1 1;S(1,1) S(2,1) S(3,1);S(1,2) S(2,2) S(3,2)];% matrice de passage

%% Calcul des derivees des Nek

dNedx1=M\[0;1;0];
dNedx2=M\[0;0;1];

%% Tenseur des deformations

eps=zeros(2,2,6);
for k=1:3
      eps(:,:,k)=[dNedx1(k) dNedx2(k)/2;dNedx2(k)/2 0];
      eps(:,:,k+3)=[0 dNedx1(k)/2;dNedx1(k)/2 dNedx2(k)];
end

%% Traces

theta=zeros(6,1);
for k=1:6
      theta(k)=trace(eps(:,:,k));
end

%% Integration des coefficients elementaires

Ae=zeros(6,6);
for k=1:6
        for n=1:6
                Ae(k,n)=aire*...
                    (2*mu*trace(eps(:,:,n)*eps(:,:,k))...
                    +lambda*theta(n)*theta(k));
        end
end
Fe=zeros(6,1);
```

AaFa.m

```
function [Aa,Fa]=AaFa(Noeuds_loc)

%% Noeuds de l'arete

Sa=Noeuds_loc([1 2],:);% abscisses et ordonnees des noeuds

%% Milieu et longueur de l'arete

h=norm(Sa(2,:)-Sa(1,:),2);% longueur de a
ma=(Sa(1,:)+Sa(2,:))/2;% milieu de a

[a,b]=betaa(Sa(1,1),Sa(1,2));% calcul de la force en S^a_1
[c,d]=betaa(ma(1),ma(2));% en m
[e,f]=betaa(Sa(2,1),Sa(2,2));% en S^a_2

%% Integration des coefficients elementaires

Aa=zeros(4,4);
Fa=(h/6)*[a+2*c;e+2*c;b+2*d;f+2*d];
```

betaa.m

```
function [beta1,beta2]=betaa(x1,x2)
```

```
rho=1000;g=10;L=10;

if (x1==0)
    beta1=rho*g*(2*L-x2);
else
    beta1=0;
end

beta2=0;
```

Fichiers de données

Noeuds.dat

```
0 0
10 0
20 0
0 10
10 10
0 20
0 0
10 0
20 0
0 10
10 10
0 20
```

Elements.dat

```
1 2 4 7 8 10
4 2 5 10 8 11
2 3 5 8 9 11
4 5 6 10 11 12
```

Aretes.dat

```
3 5 9 11
5 6 11 12
6 4 12 10
4 1 10 7
```

ID.dat

```
1
2
3
7
8
9
```

2.2.3 TP no. 4 (facultatif) : un maillage 2-d

(Tournez la page.)

TP no. 4 (facultatif)

Approximation des EDPs, Printemps 2016-2017

Ce dernier TP est une introduction aux maillages en dimension 2. L'objectif est de vous donner une idée des concepts mathématiques qu'il y a derrière les programmes. Il se trouve que c'est un problème trés difficile à partir de la dimension 2. Nous considèrerons donc seulement un maillage trés simple pour le barrage du TP no. 3. Théoriquement, le plus dur est de créer une matrice `Elements` « cohérente ». Etant donné que celà dépasse le cadre de ce cours, nous utiliserons la fonction `delaunay.m` (de matlab) sans en expliquer les idées. En ce qui concerne les conditions aux bords, vous apprendrez à les prendre en compte à partir des équations du domaine.

Table des matières

1 Préliminaires

Cette première section explique la stratégie que nous allons utiliser.

1.1 Grille cartésienne de $\overline{\Omega}$

On a besoin des équations du domaine de la figure 1. Nous allons considérer un maillage cartésien de ce domaine, avec des noeuds de la forme

$$h(i-1, j-1) \quad \text{pour } 1 \leq i, j \leq n+1,$$

où $h = 2L/n$ est le « pas en abscisse et en ordonnée ». Un exemple est donné à la figure 2 avec $n = 4$.

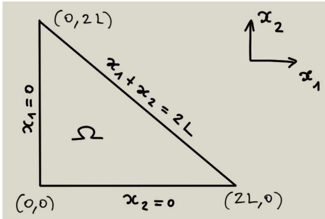


FIGURE 1 – Domaine $\overline{\Omega} = \{(x_1, x_2) : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 2L\}$.

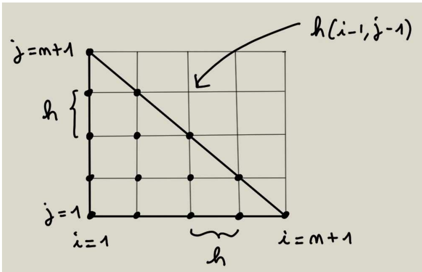


FIGURE 2 – Maillage cartésien de $\overline{\Omega}$ pour $n = 4$.

Pour conserver seulement les noeuds qui sont dans $\overline{\Omega}$, on utilise les équation de la figure 1. L'ensemble des noeuds de notre maillage est donc

$$\mathcal{N}_{\overline{\Omega}} = \{h(i-1, j-1) : 1 \leq i, j \leq n+1 \text{ et } h(i+j-2) \leq 2L\}.$$

En pratique, on doit considérer des équations approchées pour éviter de perdre des noeuds à cause des erreurs numériques. Celà veut dire qu'on va remplacer la condition $h(i+j-2) \leq 2L$ par une condition du type

$$h(i+j-2) \leq 2L + \varepsilon.$$

Un bon choix de ε est $\varepsilon = h/2$, car il est suffisamment grand pour éviter les erreurs d'arrondis et suffisamment petit pour ne pas rajouter de noeud ; voir la figure ci-dessous.

Conclusion. *L'ensemble des noeuds de notre maillage est*

$$\mathcal{N}_{\overline{\Omega}} = \{h(i-1, j-1) : 1 \leq i, j \leq n+1 \textit{ et } h(i+j-2) \leq 2L + \varepsilon\}, \tag{1}$$

où $h = 2L/n$ et $\varepsilon = h/2$.

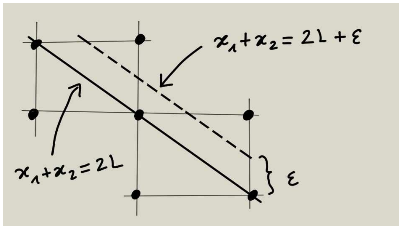


FIGURE 3 – Equation approchée de $\overline{\Omega}$ avec $\varepsilon = h/2$: les noeuds au-dessus de la droite en pointillé ne seront pas rajoutés.

1.2 Triangulation de Delaunay

Une fois qu'on a établi la liste des noeuds, on doit choisir une matrice des correspondances « cohérente ». Celà repose sur l'algorithme de triangulation de Delaunay. Il est déjà implémenté dans la fonction `delaunay.m`.

Entrées

Etant donnés $K \in \mathbb{N}$ et des noeuds $M_1, \ldots, M_K \in \mathbb{R}^2$, les variables d'entrée de `delaunay.m` sont les vecteurs

$$\mathtt{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix} \quad \text{et} \quad \mathtt{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_K \end{pmatrix},$$

où $M_i = (x_i, y_i)$.

Sortie

La variable de sortie nous donne une « matrice des correspondances cohérente pour les éléments finis $\mathcal{P}_1$ » — les noeuds du maillage étant les M_i ci-dessus.

Notation

On notera cette matrice `Tri`, pour éviter de la confondre avec notre matrice `Elements` qui doit considérer les éléments finis $\mathcal{P}_1 \times \mathcal{P}_1$; voir les exercices 1 et 2 du TD no. 3.

Exemple d'utilisation

Avec les noeuds

$$\begin{cases} M_1 = (0,0), \\ M_2 = (2,1), \\ M_3 = (-1,4), \\ M_4 = (0.5,6), \\ M_5 = (0.5,3), \end{cases}$$

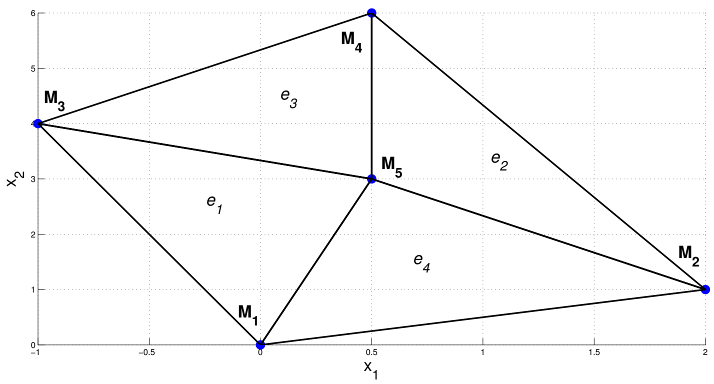
les commandes

```
> X=[0;2;-1;0.5;0.5]
> Y=[0;1;4;6;3]
> Tri=delaunay(X,Y)
```

nous donnent la matrice

$$\texttt{Tri} = \begin{pmatrix} 3 & 1 & 5 \\ 5 & 2 & 4 \\ 3 & 5 & 4 \\ 1 & 2 & 5 \end{pmatrix}$$

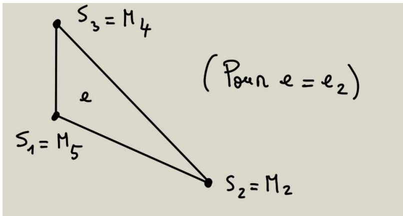
(qui peut être différente selon les versions de matlab). Celà corresponds à la triangulation ci-dessus.



Rappels. Pour $e = e_2$, par exemple, les correspondances entre les M_i et les noeuds locaux S_k sont donnés par la deuxième ligne `[5 2 4]` de `Tri` ; voir la figure ci-dessous.

Compatibilité

La fonction `delaunay.m` doit être utilisée avec les versions antèrieures à 2007 de matlab — elle existe toujours dans les versions plus récentes mais ne fonctionne plus pour les maillages de grandes tailles.



1.3 Comment déterminer les noeuds de Γ_D ?

Pour générer le fichier `ID.dat`, on doit être capable de vérifier si un noeud est sur Γ_D. D'après la figure 4 de la page suivante, l'ensemble de ces noeuds

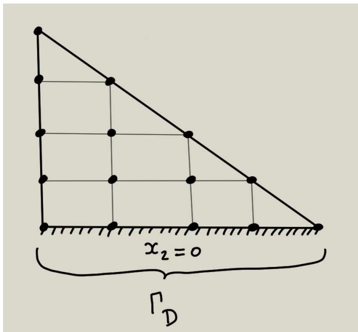


FIGURE 4 – Equation de Γ_D.

est

$$\mathcal{N}_{\Gamma_D} = \left\{(x_1, x_2) \in \mathcal{N}_{\overline{\Omega}} : x_2 = 0\right\},$$

où $\mathcal{N}_{\overline{\Omega}}$ est l'ensemble des noeuds de $\overline{\Omega}$; voir la formule (1). On doit encore remplacer cette condition par une condition approchée, de la forme

$$|x_2| \leq \varepsilon,$$

pour éviter les erreurs numériques. En raisonnant comme précédemment, on peut voir que $\varepsilon = h/2$ est satisfaisant — car celà ne rajoute aucun noeud comme à la figure 3.

Conclusion. *L'ensemble des noeuds qui sont sur Γ_D est*

$$\mathcal{N}_{\Gamma_D} = \left\{(x_1, x_2) \in \mathcal{N}_{\overline{\Omega}} : |x_2| \leq \varepsilon\right\}, \tag{2}$$

où $\varepsilon = h/2$ avec $h = 2L/n$.

1.4 Comment déterminer les arêtes de Γ_N ?

Pour le fichier `Aretes.dat`, on doit être capable de vérifier si une arête a d'un triangle e est incluse dans Γ_N. On a deux cas possibles représentés ci-dessous.

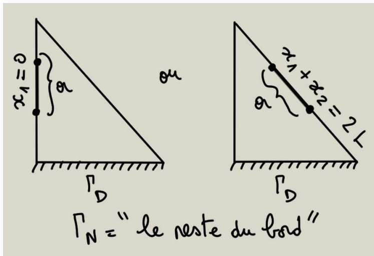


FIGURE 5 – Cas possibles pour qu'une arête a soit incluse dans Γ_N.

A partir de là, on raisonne comme précédemment en utilisant les équations de la figure 5 — ou plutôt des équations approchées pour éviter les erreurs numériques !

Conclusion. *Soit a une arête de sommets $S_1^a, S_2^a \in \mathcal{N}_{\overline{\Omega}}$. Notons*

$$S_k^a = (S_{k1}^a, S_{k2}^a) \in \mathbb{R}^2$$

les abscisses et les ordonnées de ces noeuds. Alors a est incluse dans Γ_N si et seulement si :

$$\Big[|S_{11}^a| \leq \varepsilon \text{ et } |S_{21}^a| \leq \varepsilon\Big] \text{ ou } \Big[|S_{11}^a + S_{12}^a - 2L| \leq \varepsilon \text{ et } |S_{21}^a + S_{22}^a - 2L| \leq \varepsilon\Big], \quad (3)$$

où $\varepsilon = h/2$ avec $h = 2L/n$.

2 Implémentation

Rappelons notre numérotation des ddls. Chaque noeud M_i supporte deux ddls : « ses déplacements suivant x_1 et x_2 ». Notons les $\mathscr{L}_i$ et $\mathscr{L}_{i+NN}$, où

$$NN = \text{« nombre de noeuds du maillage »}.$$

Celà veut dire que tout les ddls du problème sont les formes linéaires

$$\mathscr{L}_i(\overrightarrow{u_h}) = u_{h1}(M_i) \quad \text{et} \quad \mathscr{L}_{i+NN}(\overrightarrow{u_h}) = u_{h2}(M_i),$$

où le déplacement s'écrit sous la forme

$$\overrightarrow{u_h} = \begin{pmatrix} u_{h1} \\ u_{h2} \end{pmatrix}$$

et l'indice i varie de 1 à NN. Cette numérotation peut se représenter par la figure 6.

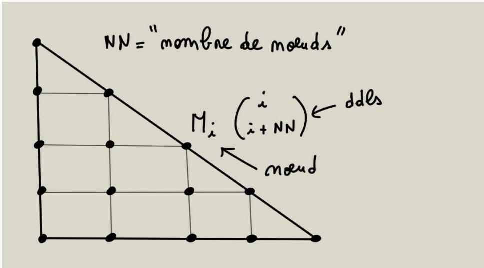


FIGURE 6 – Numéros des noeuds et des ddls.

1. Copiez/collez les programmes du `TP3` dans un nouveau dossier `TP4`. Créez et complétez le script `Mailleur.m` ci-dessous, afin de générer le fichier `Noeuds.dat`.

   Astuce. Commencez avec $n = 2$ pour avoir les mêmes noeuds et ddls qu'au TP no. 3.

Mailleur.m

```
%% Initialisation

clear all;
clc;
fclose('all');

n=2;
L=10;

h=2*L/n; % pas de la grille

eps=h/2; % pour les erreurs d'arrondis

%% Noeuds

fid=fopen('Noeuds.dat','w');

for j=1:n+1
      for i=1:n+1

            % test pour être dans le domaine

            test=h*(i+j-2)<=2*L+eps;

            % écriture dans Noeuds.dat

            if test
                  fprintf(fid,'%f %f\n',(i-1)*h,(j-1)*h);
```

```
            end

        end
end

% noeuds pour les ddls " déplacements suivant x2 "

% A COMPLETER (PLUSIEURS LIGNES)

fclose(fid)
```

Remarques. Le test vient de la formule (1). La variable logique `test` vaut 1 si $h(i-1, j-1) \in \overline{\Omega}$ et 0 sinon. On a commencé par la boucle en j (avant i) pour conserver la numérotation du TP no. 3.

Vérification. Tapez

```
> Mailleur
> load Noeuds.dat
> Noeuds
```

2. Complétez `Mailleur.m` pour générer le fichier `Elements.dat`. Vous utiliserez `delaunay.m`, pour avoir une matrice des correspondances « cohérente » entre les ddls « déplacements suivant x_1 ».

Mailleur.m

```
⋮

%% Elements

fid=fopen('Elements.dat','w');

% triangulation de Delaunay

load Noeuds.dat;
NN=size(Noeuds,1)/2;% nombre de noeuds (pris une seule fois)
X=Noeuds(1:NN,1);% abscisses des noeuds
Y=% A COMPLETER
Tri=% A COMPLETER

% écriture dans Elements.dat

Nelem=size(Tri,1);% nombre d'éléments

for e=1:Nelem
        fprintf(fid,% A COMPLETER
end

fclose(fid);
```

3. Tapez
```
> Mailleur
> load Elements.dat
> Elements
```
et précisez quels sont vos nouveaux triangles $e_1, \ldots, e_4$ sur le dessin ci-dessous.

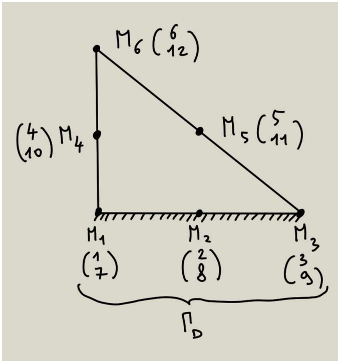


FIGURE 7 – Schéma du maillage à compléter ($n = 2$).

Remarque. Il se peut que vous ayiez des triangulations différentes, selon les versions de matlab.

4. Complétez `Mailleur.m` pour générer le fichier `ID.dat` ; vous utiliserez la formule (2).

Mailleur.m

```
⋮

%% ID

fid=fopen('ID.dat','w');

for i=1:size(Noeuds,1) % boucle sur les noeuds

        % test pour avoir un noeud de Gamma_D

        test=% A COMPLETER

        % écriture dans ID

        if test
              fprintf(fid,% A COMPLETER
        end
```

```
end

fclose(fid);
```

Vérification. Tapez

```
> Mailleur
> load ID.dat
> ID
```

et vérifiez que ce fichier corresponds à la figure 7.

5. Complétez `Mailleur.m` pour générer le fichier `Aretes.dat` ; vous utiliserez la formule (3).

Mailleur.m

```
⋮

%% Aretes

fid=fopen('Aretes.dat','w');

for e=1:Nelem% boucle sur les éléments

        % calcul des sommets du triangle

        Num=[Tri(e,1) Tri(e,2) Tri(e,3) Tri(e,1)];

        % (Explication : si on note Num=[i j l i], alors les
        % sommets du triangle sont M_i, M_j et M_l; on
        % répète M_i à la fin pour fermer le triangle.)

        for a=1:3% boucle sur les arêtes

                % calcul des noeuds de l'arête

                Sa=Noeuds(Num([a a+1]),:);

                % test pour que l'arête soit incluse dans Gamma_N

                test=% A COMPLETER

                % écriture dans Aretes.dat

                if test
                        fpintf(fid,'%i %i %i %i\n',Num(a),% ...
                        % ... A COMPLETER
                end

        end
```

```
end

fclose(fid);
```

Vérification. Tapez

```
> Mailleur
> load Aretes.dat
> Aretes
```

et terminez le schéma de la figure 7, en précisant quelles sont vos nouvelles arêtes $a_1, \ldots, a_4$.

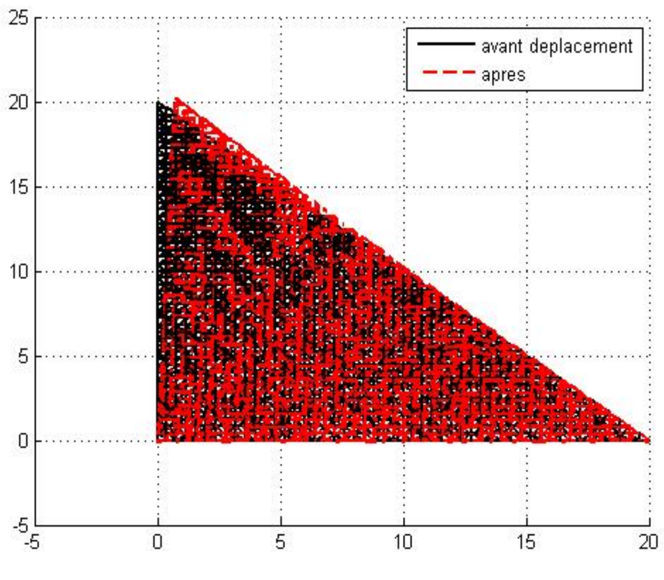
6. Exécutez `Mailleur.m` puis `Principal.m` avec $n = 4$, 10, 25, etc.

$\boxed{\text{Corrigé du TP no. 4}}$

Les programmes corrigés sont donnés à la page suivante.

Voici un exemple de dessin que vous devez obtenir

avec $\boxed{n = 50}$:



Programmes corrigés du TP no. 4

Tout les programmes sont comme dans le TP no. 3, sauf celui qui génère le maillage ci-dessous.

Mailleur.m

```
%% Initialisation

clear all;
clc;
fclose('all');

n=2;
L=10;

h=2*L/n;% pas de la grille

eps=h/2;% pour les erreurs d'arrondis

%% Noeuds

fid=fopen('Noeuds.dat','w');

for j=1:n+1
   for i=1:n+1

       % test pour etre dans le domaine

       test=h*(i+j-2)<=2*L+eps;

       % ecriture dans Noeuds.dat

       if test
           fprintf(fid,'%f %f\n',(i-1)*h,(j-1)*h);
       end

   end
end

% noeuds pour les ddls " deplacements suivant x2 "

for j=1:n+1
   for i=1:n+1

       % test pour etre dans le domaine

       test=h*(i+j-2)<=2*L+eps;

       % ecriture dans Noeuds.dat

       if test
           fprintf(fid,'%f %f\n',(i-1)*h,(j-1)*h);
       end

   end
end

fclose(fid);

%% Elements

fid=fopen('Elements.dat','w');
```

```
% triangulation de Delaunay

load Noeuds.dat;
NN=size(Noeuds,1)/2;% nombre de noeuds (pris une seule fois)
X=Noeuds(1:NN,1);% abscisses des noeuds
Y=Noeuds(1:NN,2);% ordonnees des noeuds
Tri=delaunay(X,Y);

% ecriture dans Elements.dat

Nelem=size(Tri,1);% nombre d'elements

for e=1:Nelem
    fprintf(fid,'%i %i %i %i %i %i\n',...
        Tri(e,1),Tri(e,2),Tri(e,3),...% ddls " deplacements suivant x1 "
        Tri(e,1)+NN,Tri(e,2)+NN,Tri(e,3)+NN);% " suivant x2 "
end

fclose(fid);

%% ID

fid=fopen('ID.dat','w');

for i=1:size(Noeuds,1)% boucle sur les noeuds

    %test pour avoir un noeud de Gamma_D

    test=abs(Noeuds(i,2))<=eps;

    % ecriture dans ID

    if test
        fprintf(fid,'%i\n',i);
    end

end

fclose(fid);

%% Aretes

fid=fopen('Aretes.dat','w');

for e=1:Nelem% boucle sur les elements

    % calcul des sommets du triangle

    Num=[Tri(e,1) Tri(e,2) Tri(e,3) Tri(e,1)];

    % (Explication : si on not Num=[i j l i], alors les
    % sommets du triangle sont M_i, M_j et M_l ; on
    % repete M_i a la fin pour fermer le triangle.)

    for a=1:3% boucle sur les aretes

        % calcul des noeuds de l'arete

        Sa=Noeuds(Num([a a+1]),:);

        % test pour que l'arete soit incluse dans Gamma_N

        test=((abs(Sa(1,1))<=eps)...
```

```
            &&(abs(Sa(2,1))<=eps))...
            ||...
            ((abs(Sa(1,1)+Sa(1,2)-2*L)<=eps)...
            &&(abs(Sa(2,1)+Sa(2,2)-2*L)<=eps));

        % ecriture dans Aretes.dat

        if test
            fprintf(fid,'%i %i %i %i\n',...
                Num(a),Num(a+1),...% ddls " deplacements suivant x1 "
                Num(a)+NN,Num(a+1)+NN);% " suivant x2 "
        end

    end

end

fclose(fid);
```

Bibliographie

[1] DHATT G., TOUZOT G. ET LEFRANÇOIS E. *Méthode des éléments finis.* Hermes Sciences Publications, 2015.

(Destiné avant tout aux ingénieurs ; normalement disponible à la bibliothèque de l'ens2m, sinon au moins une version antèrieure.)

[2] FORTIN A. ET GARON A. *Les éléments finis : de la théorie à la pratique.* Polycopié d'un cours de l'université de Laval, Canada. Version de 2016.

(Nécessite un peu plus de prérequis théoriques.)

`http://giref.ulaval.ca/files/afortin/Publications/elements_finis.pdf`

[3] LAYDI M. R. *Introduction à la méthode des éléments finis.* Polycopié du cours du semestre vert d'automne de l'ens2m. Version de 2004.

(Destiné avant tout aux ingénieurs.)

`https://maths.ens2m.fr/Laydi/Enseignement/EF2004.pdf`

[4] SAYAS F.-J. *A gentle introduction to the Finite Element Method.* Polycopié d'un cours de l'université de Delaware, Etats-Unis. Version de 2015.

(Nécessite un peu plus de prérequis théoriques.)

`http://www.math.udel.edu/~fjsayas/documents/anIntro2FEM_2015.pdf`