DEA Physique et Génie des Matériaux

—

Mastère Matériaux et Mise en Forme

—

# **Fiches** TD Elément Finis

P. Laure (Patrice.Laure@inln.cnrs.fr)
Institut Non Linéaire de Nice

### Séance 1 : Initiation à Scilab

**Ex 1** *Manipulation sur les vecteurs*
Ecrire (sans utiliser de boucle) les vecteurs suivants :
**1)** Nombres de 1 à 3 par pas de 0.1 .
**2)** Nombres de 3 à 1 par pas de -0.1 .

**3)** Les carrés des 10 premiers entiers.
**4)** Nombres de la forme $(-1)^n n^2$ pour $n = 1, 10$.

**5)** Vecteur formé de 10 “0” suivis de 10 “1”.

**Ex 2** *Manipulation sur les Matrices*

**1)** Ecrire (sans utiliser de boucle) la matrice d'ordre 6 contenant les entiers de 1 à 36, rangés par lignes.
**2)** Ecrire (avec ou sans boucle) la matrice d'ordre $n$ suivante :

$$\mathcal{C} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

**Ex 3** *Exemple de tracé*

Tracer la fonction $f(x) = \sin(x) + x$ entre 0 et 4 $\pi$.

**Ex 4** *Ecriture d'une fonction*
Ecrire la fonction `alterne2_colonnes` qui prend en entrée deux matrices quelconques $A$ et $B$ de même dimensions. Elle retourne la matrice formée en alternant les colonnes de $A$ et $B$.
On écrit le programme dans un fichier `alterne2_colonnes.sci` et on compile la fonction sous scilab avec la commande `exec(''alterne2_colonnes.sci'')`.

**Ex 5** *Calcul de la solution d'une edo*

On considére l'équation

$$-u'' + u = 1$$
$$u(0) = u(1) = 0$$

**1)** Vérifier $u(x) = 1 - \frac{e^x + e^{1-x}}{1+e}$ est la solution exacte du problème.
**2)** Résoudre numériquement en utilisant $n$ points de discrétisation et l'approximation

$$u'' = \frac{u_{i+1} - 2\ u_i + u_{i-1}}{\Delta_x^2} \quad \text{avec} \quad \Delta_x = 1/(n-1).$$

Ecrire le problème sous forme matricielle $AU = F$ où $U = (u_i)_{i=1,n}$ est le vecteur correspondant aux valeurs de $u$ aux points de discrétisation.

Ecrire un programme en scilab qui résoud ce problème. Comparer la solution calculée et la solution exacte. Etudier l'erreur quand on change $n$.

**Séance 2 : Initiation aux éléments finis 1D**

Pour pouvoir faire ce TD facilement, il faut prendre à l'adresse

**http ://www.inln.cnrs.fr/~laure/EF/TD2/Seance2/index.html**

les fichiers suivant :
- `P1.sci` et `P2.sci` ; fonctions qui donnent les fonctions de Lagrange pour l'élément $\mathbf{P_1}$ et $\mathbf{P_2}$ .
- `gauss.sci` ; une fonction qui donne les points de Gauss pour l'intégration numérique.

**Ex 1** *Eléments Finis Unidimensionnels*
On considère l'équation

$$-u'' + u = 1$$
$$u(0) = u(1) = 0$$

**1)** Ecrire la formulation faible de cette équation.

**2)** On prend une interpolation $\mathbf{P_1}$, c'est à dire sur l'intervalle de référence on utilise les polynômes de Lagrange $L_1(\xi) = (1-\xi)/2$ et $L_2(\xi) = (1+\xi)/2$ (cf. le programme `P1.sci`).
On prend `Nelt` éléments : écrire le programme qui calcule le vecteur `COOR` et la matrice `CONNEC`.
**3)** Ecrire le système élémentaire à résoudre sur chaque élément. Ecrire le programme correspondant. Pour l'intégration numérique, on utilisera les points de Gauss (cf. le programme `gauss.sci`).

**4)** Ecrire le programme qui fait l'assemblage.

**5)** Ecrire le programme qui résoud le Problème. Ne pas oublier d'imposer les conditions aux limites $u(0) = u(1) = 0$. Regarder l'évolution de l'erreur en fonction du pas de discrétisation.

**6)** Modifier le programme écrit pour utiliser des éléments $\mathbf{P_2}$ (programme `P2.sci`). Regarder l'évolution de l'erreur en fonction du pas de discrétisation.

**7)** Modifier le programme pour les conditions aux limites $u(0) = 0$ et $\partial_x u(1) = 0$.

**Ex 2** *A faire pour la prochaine séance*
Donner la formulation faible de l'equation $-\frac{d^2u}{dx^2} - u + x^2 = 0$ définie sur $[0, 1]$ avec les conditions aux limites $u(0) = 0$ et $u'(1) = 1$.
Il faudra aussi écrire le programme qui permet d'obtenir la solution numérique (on peut modifier les programmes déjà écrits à l'exercice 1).

**Séance 3 : Initiation aux éléments finis 2D**

Pour pouvoir faire ce TD facilement, il faut prendre à l'adresse

**http ://www.inln.cnrs.fr/∼laure/EF/TD2/Seance3/index.html**

les fichiers suivant :
- `Q1.sci, Q2.sci, P1.sci`; fonctions qui donne les fonctions de Lagrange pour les éléments $\mathbf{Q_1}$, $\mathbf{Q_2}$ et $\mathbf{P_1}$.
- `gauss.sci`; une fonction qui donne les points de Gauss pour l'intégration numérique.
- `maillage_2D_?.sci`; des fonctions qui donnent le maillage d'un carré par des quadrangles et des triangles.
- `cal_indice.sci, interpol.sci, trace_maillage_2D.sci, trace_u_2D.sci`; des fonctions qui permettent de tracer le maillage et la solution $u$.

On va utiliser le fichier `seance3.sci` qui devrait ressembler au programme écrit à la deuxième séance. Les instructions d'assemblage ont été légèrement modifier pour tenir compte de la structure creuse de la matrice globale. Le travail consistera à ajouter les instructions permettant de calculer le système élémentaire pour un problème bidimensionnel.

**Ex 1** *Ecoulements de Poiseuille*

On veut résoudre

$$-\Delta u = 1 \quad \text{sur un carré} \quad [0,1]\times[0,1]$$

avec les conditions aux limites de Dirichlet

$$u(0,y) \;=\; u(1,y) \;=\; u(x,0) \;=\; u(x,1) \;=\; 0 \tag{1}$$

en utilisant des éléments $\mathbf{Q_2}$ (programme `Q2.sci`).

**1)** Ecrire la forme variationnelle de l'équation et le système élémentaire à résoudre sur chaque élément.

**2)** On utilise des quadrilatères comme éléments (cf. `maillage_2D.sci` et `trace_maillage_2D.sci` pour tracer ce maillage). Détailler le calcul de la matrice jacobienne qui correspond à la transformation linéaire du quadrilatère de référence vers un quadrilatère quelconque. Ecrire les instructions correspondantes.

**3)** La matrice `CONNEC` et le vecteur `COOR` sont donnés par la fonction `maillage_2D.sci`. La matrice `ADRESS` et le vecteur `NUMER` sont calculés pour les conditions aux limites (1) dans le fichier `seance3.sci`. Ecrire les instructions qui permettent de calculer le système élémentaire (les matrices `A_el` et `F_el`) en complétant le fichier `seance3.sci`.

**4)** Ecrire une fonction scilab qui calcule le débit de la solution $u$ (c'est à dire $\int_0^1 \int_0^1 u\,dx\,dy$)

**5)** Modifier le calcul de `NUMER` pour avoir des conditions de Neumann pour $x = 0$ et $x = 1$. Vérifier que la solution trouvée ne dépend pas de $x$ et est égal à $u(y) = y(1-y)/2$.

**Ex 2** *Solution analytique*

Pour tester les différents maillages, on résoud un problème pour lequel on connaît une solution analytique. On considère l'équation sur $\Omega = [0,1] \times [0,1]$

$$-\Delta u \; + \; 3u \; = \; 0$$

et on peut montrer que $u(x,y) = e^{2x} \sin(y)$ est une solution si on ajuste correctement les conditions aux limites sur les bords.

**1)** Ecrire la formulation faible de ce problème.

**2)** En adaptant les programmes écrits à l'exercice précédents et les différentes fonctions `maillage_2D_?.sci` tester les élements $\mathbf{Q_1}$, $\mathbf{Q_2}$ et $\mathbf{P_1}$. On peut comparer l'erreur entre la solution calculée et la solution analytique.

**Séance 4 : Programmation modulaire - Problèmes non linéaires**

Pour faire ce TD, il faut prendre à l'adresse

**http ://www.inln.cnrs.fr/~laure/EF/TD2/Seance4/index.html**

les fichiers suivant :
- `P1.sci` et `P2.sci`, des fonctions qui donnent les fonctions de Lagrange.
- `gauss.sci`, une fonction qui donne les points de Gauss pour l'intégration numérique.
- `maillage_1D.sci`, fonction qui calcule le maillage en 1D.
- `trace_u_1D.sci`, pour tracer la solution $u$ .

**Ex 1** *Ecoulement de Poiseuille*
On veut résoudre $-\nabla.(\mu\nabla u) = \nabla P$ sur un domaine $[0,1]$ avec une condition aux limites de Diriclet ($u(0) = u(1) = 0$).
Le terme $\nabla P$ (la perte de charge) est constant (pour simplifier, on peut prendre $\nabla P = 1$) et $\mu$ est fonction du taux de cisaillement.

**1)** Ecrire la formulation faible du problème.

**2)** Afin d'avoir un programme plus simple et plus modulaire, on a définit les tableaux `PG, XG, COOR, CONNEC, NUMER` et `ADRESS` comme des variables globales. De la même façon, on a écrit des petites fonctions qui font l'assemblage (`assemblage.sci`), le calcul du système élementaire (`cal_Melm_?`), le maillage (`maillage_1D.sci`) et le calcul de la solution dans le cas linéaire où $\mu$ est constant (`cal_sol_lin.sci`). Prendre ces programmes à la même adresse et lire le programme `exemple_1D` qui utilisent ces fonctions. Ce dernier programme calculent la solution du problème de Poiseuille dans le cas 1D pour $\mu = 1$ .

**3)** On suppose que $\mu$ suit la loi de Carreau-Yasuda

$$\mu(\dot{\epsilon}) = \mu_o\ (1 + (\lambda\dot{\epsilon})^a)^{\frac{m-1}{a}} \qquad \text{avec} \qquad \dot{\epsilon} = \left|\frac{\partial u}{\partial x}\right|$$

On peut prendre $\mu_0 = 1.$, $\lambda = 2.$, $m = .2$ et $a = 2$. Puis les valeurs pour le polyéthylène : $\mu_0 = 1.$, $\lambda = 13.47$, $m = 0.294$ et $a = .381$.

**3.1)** On va utiliser une méthode de point fixe. Modifier la fonction `cal_Melm_1D.sci` afin de calculer le système élémentaire.

**3.2)** Ecrire une fonction que l'on peut appeller `cal_sol_ptfixe.sci` qui appelle la fonction `cal_sol_lin.sci` et la fonction écrite en **3.1)** (par exemple `cal_Melm_1D_a.sci`).

**3.3)** Ecrire la formulation faible à résoudre si on veut utiliser une méthode itérative de Newton.
Ecrire une nouvelle fonction (par exemple `cal_Melm_1D_b`) qui calcule le système élementaire.
Ecrire la fonction qui calcule la solution par une méthode de Newton (`cal_sol_newton.sci`). Cette fonction appellera la fonction `cal_sol_lin.sci` et la fonction `cal_Melm_1D_b.sci`.

**3.4)** Comparer les deux méthodes quand on augmente $\lambda$.

**4)** On peut regarder le cas 2D avec maintenant

$$\dot{\epsilon} = \sqrt{\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}}$$

Les programmes analogues à ceux du cas unidimensionnel se trouve à l'adresse

**http ://www.inln.cnrs.fr/~laure/EF/TD2/Seance4/index.html**

**Séance 5 : Eléments finis mixtes**

Pour faire ce TD, il faut prendre à l'adresse

**http ://www.inln.cnrs.fr/~laure/EF/TD2/Seance5/index.html**

les fichiers suivant :
- `Q1.sci`, `Q2.sci` et `P1.sci` ; des fonctions qui donnent les fonctions de Lagrange.
- `gauss.sci` ; une fonction qui donne les points de Gauss pour l'intégration numérique.
- `maillage_2D_Q?.sci` ; les fonctions qui calculent le maillage en 2D par des quadrangles ou des triangles
- `cond_limit_2D.sci` ; la fonction qui calcule les matrices `UC`, `NUMER` et `ADRESS`.
- `cal_indice.sci`, `interpol.sci`, `trace_p_2D.sci`, `trace_u_2D.sci`, `trace_vecteur.sci` et `trace_maillage.sci` pour tracer la solution $u$, $p$ et le maillage.
- `cal_sol_lin.sci`, `assemblage` ; les fonctions qui calcule la solution d'un problème linéaire et fait l'assemblage de la matrice globale.

**Ex 1** *Le problème de Stokes*
On veut résoudre sur un carré $\Omega = [0,1] \times [0,1]$ :

$$\nabla.\sigma = 0 \quad . \tag{2}$$

Pour tester les diverses méthodes, on va travailler sur une solution analytique. si $\mu = 1$ l'équation (2) se simplifie en $\Delta u = \nabla p$ et on peut montrer que

$$u_x = e^x \ (y \ \sin(y) - \cos(y)) \ \ ; \ \ u_y = e^x \ y \ \cos(y) \ \ ; \ \ p = 2e^x \ \cos(y)$$

est une solution du problème si on ajuste correctement les conditions aux limites.

**1)** Ecrire la formulation faible du problème, sachant que $\sigma = 2\mu\dot{\epsilon}(u) - p\mathbf{I}$.

**2)** Expliciter le calcul du système élémentaire.

**3)** Le programme `exemple_Q2Q0.sci` calcule la solution de ce problème en utilisant un élément $\mathbf{Q}_2 - \mathbf{Q}_0$.
Le calcul de la matrice élémentaire s'effectue avec la fonction `cal_Melm_Q2Q0.sci`.
Modifier ces programmes pour utiliser l'élément $\mathbf{Q}_2 - \mathbf{Q}_1$.

**4)** Vérifier que l'on a une *instabilité en damier* pour l'élément $\mathbf{Q}_1 - \mathbf{Q}_0$. Il faut écrire une nouvelle fonction `cal_Melm_Q1Q0.sci` en modifiant l'une des fonctions précédentes.

**5)** Prendre les programmes `exemple_P1P1.sci` et `cal_Melm_P1P1.sci` et vérifier que l'éléments $\mathbf{P}_1 - \mathbf{P}_1$ donne une solution admissible pour la vitesse bien qu'il ne vérifie la condition inf-sup.

**6)** Modifier la fonction `cal_Melm_P1P1.sci` pour pour ajouter une bulle (élément $\mathbf{P}_1^+ - \mathbf{P}_1$).

**7)** Le problème de la cavité entrainée :
Calculer la solution de (2) pour les conditions aux limites

$$u_x(x,0) = u_x(0,y) = u_x(1,y) = 0 \quad ; \quad u_x(x,1) = 1$$

$$u_y(x,0) = u_x(x,1) = u_y(0,y) = u_y(1,y) = 0$$

Il faut modifier le calcul de `UC` dans `cond_limit_2D.sci`.