Solution TP2 : Exemple de mise en oeuvre de la méthode d'éléments finis

Ce TP a pour but de suivre les étapes de mise en oeuvre de la méthode des éléments finis sur un problème modèle monodimensionnel. Les éléments finis concernés seront les éléments finis Lagrange de degré 1 et 2.

Exercice - 1

Problème à résoudre :

$$\left|\begin{array}{ll} -\Delta u = f & \text{dans } \Omega, \\ u = 0 & \text{sur } \partial\Omega, \end{array}\right. \qquad \text{où} \qquad \Omega =]0,1[, f \in L^2(\Omega) \tag{1}$$

Q-1 **: Formulation Variationnelle.** Soit u une fonction suffisamment régulière,

$$\begin{array}{rl} u \text{ solution de } (1) & \Longrightarrow \; u'' = f \\ & \Longrightarrow \; \int_0^1 -u''v\,dx = \int_0^1 fv\,dx \quad \forall v \\ & \Longrightarrow \; \int_0^1 u'v'\,dx - [u'v]_{x=0}^{x=1} = \int_0^1 fv\,dx \quad \forall v \\ & \Longrightarrow \; \int_0^1 u'v'\,dx = \int_0^1 fv\,dx \quad \forall v \text{ tel que } v(0) = v(1) = 0 \\ & \Longrightarrow \; \left\{ \begin{array}{l} u \in H_0^1(\Omega), \\ \int_0^1 u'v'\,dx = \int_0^1 fv\,dx \quad \forall v \in H_0^1(\Omega) \end{array} \right. , \end{array}$$

où $H^1(\Omega) = \left\{v \in L^2(\Omega) \text{ tel que } v' \in L^2(\Omega)\right\}, \quad H_0^1(\Omega) = \left\{v \in H^1(\Omega) : \; v(0) = v(1) = 0\right\}.$

Ainsi, la solution u de l'équation (1) vérifie : $\boxed{\forall v \in H_0^1(\Omega), \quad \int_\Omega u'(x)v'(x)\,dx = \int_\Omega f(x)v(x)\,dx}$.

Q-2 **: Discrétisation par éléments finis P_r−Lagrange ($r = 1, 2$).**

On commence par introduire un maillage de Ω, c'est-à-dire une suite de N_{so} points

$$0 = x_1 < \ldots < x_{N_{so}-1} < x_{N_{so}} = 1 \quad \text{ce qui génère } N_{ma} = N_{so} - 1 \text{ mailles } I_i = [x_i, x_{i+1}].$$

On introduit l'espace éléments finis P_r−Lagrange, $X_h^r = \left\{v_h \in \mathcal{C}^0([0,1]) \text{ tel que } u_h|_{[x_i,x_{i+1}]} \in \mathbb{P}_r\right\}$.
C'est bien un sous-espace de $H^1(\Omega)$. On introduit alors le sous-espace de X_h^r prenant en compte les conditions aux limites : $X_{0h}^r = \{v_h \in X_h^r \text{ tel que } v_h(0) = v_h(1) = 0\}$.

Q-3 **:** X_{0h}^1 est un sous-espace de $H_0^1(\Omega)$ de dimension $N_{so} - 2$, correspondant au nombre de sommets internes. Et X_{0h}^2 est un sous-espace de $H_0^1(\Omega)$ de dimension $N_{ma} + N_{so} - 2$ correspondant au total des degrés de liberté situés aux sommets des mailles (soit $N_{so} - 2$) plus ceux situés au centre des mailles (soit N_{ma}).

Et le problème discret s'écrit : $\boxed{\forall v_h \in X_{0h}^r, \quad \int_\Omega u_h'(x)v_h'(x)\,dx = \int_\Omega f(x)v_h(x)\,dx}$.

Q-4 **: Formulation matricielle.** Si on peut expliciter une base $(\varphi_i)_{1\le i\le \dim(X_{0h}^r)}$ de X_{0h}^r, alors en posant $u_h = \sum_{i=1}^{\dim(X_{0h}^r)} u_i\varphi_i$ et $U = (u_1, \ldots, u_{\dim(X_{0h}^r)})^t$ le problème discret ci-dessus se ramène à la résolution du système linéaire : $AU = b$, avec $A_{i,j} = \int_\Omega \varphi_i'(x)\varphi_j'(x)\,dx \quad 1 \le i,j \le \dim(X_{0h}^r)$, $b_i = \int_\Omega f(x)\varphi_i(x)\,dx \quad 1 \le i \le \dim(X_{0h}^r)$.

Seulement à part dans le cas particulier de la dimension une, il n'est pas aisé dans le cas général d'exhiber une base de X_{0h}^r et de calculer les coefficients de A et b à partir de cette base. Néanmoins, on sait identifier une base locale de X_h^r (c'est à dire une base sur chaque élément), à ce titre, on calcule la matrice A et le vecteur b directement dans X_h^r, par assemblage des matrices et vecteurs locaux. Puis on introduit les conditions aux limites après assemblage. **Les exercices 2 et 3 décrivent cette procédure. La compréhension de ces exercices assure la compréhension de la mise en oeuvre de la méthode des éléments finis dans un cadre général.**

Exercice - 2 *Éléments finis P1-Lagrange*

On désigne par `X` les sommets x_i du maillage qu'on numérote de 1 à N_{so}.

1. **Table des degrés de libertés.** La position des degrés de liberté sont les sommets du maillage. Sur une maille numéro i : $I_i = [x_i, x_{i+1}]$, Le sommet 1 est x_i son numéro global est i ; le sommet 2 est x_{i+1}, son numéro global est $i+1$. Donc le sommet numéro $j = 1, 2$ de la maille i a pour numéro $i + j - 1$. D'où la table de degré de liberté

$$l2g(i,j) = i + j - 1 \quad 1 \le i \le N_{ma}, 1 \le j \le 2.$$

2. **Traitement élémentaire**. Soit $I_i = [x_i, x_{i+1}]$, une maille quelconque.
   (a) Explicitation des fonctions de base locale $P1-$Lagrange

$$\phi_1(x) = -\frac{x - x_{i+1}}{x_{i+1} - x_i}, \quad \phi_2(x) = \frac{x - x_i}{x_{i+1} - x_i}$$

   On pose $h_i = x_{i+1} - x_i$ le diamètre de la maille I_i.

   (b) Calcul de la matrice élémentaire A^i et du vecteur élémentaire b^i

$$A^i = \int_{x_i}^{x_{i+1}} \begin{bmatrix} {\phi_1}' \\ {\phi_2}' \end{bmatrix} \begin{bmatrix} {\phi_1}' & {\phi_2}' \end{bmatrix} dx = \frac{1}{h_i} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$b^i = \int_{x_i}^{x_{i+1}} f(x) \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} dx = \frac{h_i}{2} \begin{bmatrix} f(x_i) \\ f(x_{i+1}) \end{bmatrix}$$

   Pour calculer le second membre, on a utilisé la formule des trapèzes : $\int_{x_i}^{x_{i+1}} \psi(x)\,dx = \frac{x_{i+1} - x_i}{2} (\psi(x_i) + \psi(x_{i+1}))$.

   (c) Script `Matlab` de construction de A^i, b^i

```
function [Ai,bi]=elementlaplaceP1(xi1,xi2, i)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hi=xi2 - xi1;
Ai = (1/hi)*[1,-1;-1,1];
bi=(hi/2)* [ f(xi1); f(xi2)];
```

3. **Assemblage** : Pour assembler la matrice A et le second membre b, on boucle sur les éléments et pour chaque élément, on construit sa matrice locale et sont vecteur local et on les accumulent dans A et b.

script Matlab d'assemblage pour P1

```
function [A,b] = assembleLaplaceP1(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function [A,b] = assembleLaplaceP1(x)
%%
%% fonction qui assemble la matrice du laplacien P1 en 1D
%% ENTREE: x les noeuds du maillage
%% SORTIE: A la matrice
```

```
%%         b le vecteur second membre
%
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nso = length(x); % nombre de sommets
Nma = Nso -1 ;   % nombre de mailles
% table des degres de liberté
for i = 1:Nma
    l2g(i,1) = i;
    l2g(i,2) = i+1;
end
%allocation d'espace
A=zeros(Nso,Nso);
b=zeros(Nso,1);
%boucle d'assemblage
for i=1:Nma
    %On recupère les sommets de l'élément
    x1 = x(i); x2 = x(i+1);
    % On calcule la matrice et le second membre élémentaires
    [Ai,bi] = elementlaplaceP1(x1,x2,i);
    %Et on assemble la matrice et le vecteur élémentaires dans les matrice et vecteur globaux
    A(l2g(i,1:2), l2g(i,1:2)) = A(l2g(i,1:2), l2g(i,1:2)) + Ai(1:2,1:2);
    b(l2g(i,1:2)) = b(l2g(i,1:2)) + bi(1:2);
end
```

4. **Conditions aux limites** :

   Pour introduire les conditions aux limites $u_h(0) = u_h(1) = 0$, on commence par localiser les degrés de liberté situés en $x = 0$ et $x = 1$, par leur numéro global et leurs coordonnées : pour $x = 0$, on a $idl = 1, xdl = 0$ et pour $x = 1$, on a $idl = N_{so}$ et $xdl = x_{Nso}$.

   On dispose alors après de plusieurs approches dont deux sont les suivantes :

   - Par pénalisation : on choisit une très grande valeur tgv, et pour chaque indice idl du degré de liberté devant porter la condition de Dirichlet $u_h(xdl) = g(xdl)$, on pose $A(idl, idl) = tgv, \quad b(idl) = g(xdl) * tgv$.
   - Par terme unité sur la diagonale : On crée le vecteur U_D nul de même taille que U. Puis pour chaque idl on pose $U_D(idl) = g(xdl)$. On remplace ensuite b par $b = b - AU_D$. Et pour chaque indice idl, on pose $b(idl) = U_D(idl)$ et on met tous les termes des lignes et colonnes idl de A à zéro, sauf le terme diagonal qu'on met à 1.

script Matlab d introduction des conditons auxlimites pour P1

```
function [A,b]=conditionLimiteLaplaceP1(x,A,b)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% function [A,b] = conditionLimiteLaplaceP1(x,A,b)
%%
%% fonction qui introduit les conditions aux limites pourle Laplacien P1 en 1D
%% ENTREE: x les noeuds du maillage
%          A la matrice assemblée
%          b le vecteur assemblé
%          entrée implicite: g la fonction de la condition de Dirichliet
%% SORTIE: A la matrice A modifiée
%%         b le vecteur b modofié
%
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nso = length(x); % nombre de sommets
%Nma = Nso -1 ;   % nombre de maille

%CONDITION AUX LIMITES PAR PENALISATION
% tgv = 1e+30;
% A(1,1) = tgv; b(1) = tgv * g(x(1)); % g est la condition de dirichlet
% A(Nso,Nso) = tgv; b(Nso) = tgv * g(x(Nso));

%CONDITION AUX LIMITES PAR MODIFICATION
ud=zeros(Nso,1);ud(1)=g(x(1));ud(Nso)=g(x(Nso));
b=b-A*ud;
b(1) = ud(1);b(Nso)=ud(Nso);
A(:,1) =0;A(1,:) =0;A(:,Nso) =0;A(Nso,:) =0;
A(1,1) =1;A(Nso,Nso) =1;
```

On peut alors créer la fonction `function [A,b] = laplaceP1(X)` qui assemble la matrice et le second membre de la discrétisation par éléments finis $P1-$Lagrange du Laplacien.

```
function [A,b]=laplaceP1(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [A,b]=laplaceP1(x)
% fonction qui assemble la matrice du laplacien P1
% et introduit les conditions aux limites
% Entree: x maillage
%
% Sortie: A matrice
%         b vecteur second membre global
%
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[A,b] = assembleLaplaceP1(x);
[A,b] = conditionLimiteLaplaceP1(x,A,b);
```

Exercice - 3 *Éléments finis P2-Lagrange*

On continue à désigner par `X` les sommets x_i du maillage, qu'il ne faut pas confondre avec la position des degrés de liberté.

1. **Table des degrés de libertés.**

   Au niveau global, on choisit de numéroter d'abord les degrés de libertés situés sur les sommets (ceci donne N_{so}) et ensuite ceux situés au milieu des mailles. Ainsi sur une maille $I_i = [x_i, x_{i+1}]$, le numéro du degré de liberté situé en son centre est $N_{so} + i$ c'est-à-dire somme du numéro de la maille et du nombre total des sommets. Les numéros globaux des degrés de liberté locaux sont donc $(i, N_{so}+i, i+1)$. D'où la table

   $$l2g(i,1) = i,\ l2g(i,2) = N_{so} + i,\ l2g(i,3) = i + 1, \quad 1 \leq i \leq N_{ma}.$$

   Ce choix permet de numéroter en dernier les degrés de liberté associés aux noeuds internes aux mailles. D'autres choix sont possibles comme par exemple, numéroter suivant l'ordre croissant des abscisses des positions des degrés de liberté.

2. **Traitement élémentaire**. Soit $I_i = [x_i, x_{i+1}]$, une maille quelconque et $x_{i+\frac{1}{2}}$ son point milieu.

   (a) Explicitation des fonctions de base locale $P2-$Lagrange

   $$\phi_1(x) = \frac{(x - x_{i+\frac{1}{2}})(x - x_{i+1})}{(x_i - x_{i+\frac{1}{2}})(x_i - x_{i+1})},\ \phi_2(x) = \frac{(x - x_i)(x - x_{i+1})}{(x_{i+\frac{1}{2}} - x_i)(x_{i+\frac{1}{2}} - x_{i+1})},$$

   $$\phi_3(x) = \frac{(x - x_i)(x - x_{i+\frac{1}{2}})}{(x_{i+1} - x_i)(x_{i+1} - x_{i+\frac{1}{2}})}.$$

   Attention à l'ordre (dite numérotation locale) des fonctions de base locale, imposé par la table des degrés de liberté $l2g$. En effet, la fonction de base locale n^o1 est associée à x_i, celle n^o2 est associée à $x_{i+\frac{1}{2}}$, et celle n^o3 à x_{i+1}.

   (b) Calcul de la matrice élémentaire A^i et du vecteur élémentaire b^i

   On a jugé utile d'utiliser les formules de quadrature pour calculer à la fois A^i et b^i. Cela nous a incité à programmer les fonctions de base de sorte à retourner pour chacune d'elles sa valeur et sa dérivée en un point (donné en argument). (voir ci-dessous)

   (c) script `Matlab` de construction de A^i, b^i (voir ci-dessous)

Fonctions de base locale P2

```
function [v,dv] = baselocalP2(x1,x2,i,x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [v,dv] = baselocalP1(x1,x2,i,x)
% fonction de base P1-Lagrange
% entrée: [x1,x2] ->intervalle
%   i ->numero local de la fonction de base
%   x ->point d'évaluation
% sortie :
%  v->valeur de la fonction de base
% dv->valeur de la derivee de la fonction de base
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xm =(x1+x2)/2;
if(i==1)
      d = (x1-xm)*(x1-x2);
      v = (x-xm)*(x-x2)/d;
     dv = (2*x - xm - x2)/d;
elseif(i==2)
      d = (xm-x1)*(xm-x2);
      v = (x-x1)*(x-x2)/d;
      dv= (2*x - x1 - x2)/d;
elseif(i==3)
       d = (x2-xm)*(x2-x1);
       v = (x-xm)*(x-x1)/d;
      dv = (2*x - xm - x1)/d;
end
```

Calcul élémentaire pour P2

```
function [Ai,bi]=elementlaplaceP2(x1,x2, i)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% matrice et le second membre élémentaires
% du Laplacien par EF P2-Lagrange
% Entrees:[x1,x2] intervalle
%         i numero de l'élément
% Sortie : Ai matrice 3 x 3
%          bi vecteur 3 x 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%formule de quadrature de simpson
w = [1, 4, 1]*(x2-x1)/6; %poids
z=[x1 , (x1+x2)/2, x2]; %points
%initialisation
Ai=zeros(3,3);bi=zeros(3,1);
%on boucle sur les points de quadrature
for q=1:3
    wq=w(q);
    zq = z(q);
    %boucle sur les lignes
    for l=1:3
        [vl,dvl] = baselocalP2(x1,x2,l,zq);
        bi(l)= bi(l) + wq * f(zq)*vl;
        %boucle sur les colonnes
        for m=1:3
            [vm,dvm] = baselocalP2(x1,x2,m,zq);
            Ai(l,m)=Ai(l,m) + wq * dvl * dvm;
        end
    end
```

3. **Assemblage** : L'assemblage est identique à celui précédent.
   Nous avons modifié le script pour retourner aussi la position des degrés de liberté, afin de faciliter la représentation graphique de la solution P2.

script Matlab d'assemblage pour P2

```
function [A,b,y] = assembleLaplaceP2(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [A,b,y] = assembleLaplaceP2(x)
%
% fonction qui assemble la matrice du laplacien P2 en 1D
% ENTREE: x les noeuds du maillage
% SORTIE: A la matrice
%         b le vecteur second membre
%         y position des degres de liberté
%
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nso = length(x); % nombre de sommets
Nma = Nso -1 ;   % nombre de mailles
Ndl=Nso + Nma;
% table des degrés de liberte
% et leur position
y=[];l2g=[];
for i = 1:Nma
    l2g(i,1) = i;
    l2g(i,2) = i+Nso;
    l2g(i,3) = i+1;
    y(l2g(i,1))=x(i);
    y(l2g(i,3)) = x(i+1);
    y(l2g(i,2)) = (x(i)+x(i+1))/2;
end
%allocation d'espace
A=zeros(Ndl,Ndl);
b=zeros(Ndl,1);
%boucle d'assemblage
for i=1:Nma
    %On recupere les sommets de l'element
    x1 = x(i); x2 = x(i+1);
    % On calcule la matrice et le second membre elementaires
    [Ai,bi] = elementlaplaceP2(x1,x2,i);
    % Et on assemble la matrice et le vecteur élémentaires
    A(l2g(i,1:3), l2g(i,1:3)) = A(l2g(i,1:3), l2g(i,1:3)) + Ai(1:3,1:3);
    b(l2g(i,1:3)) = b(l2g(i,1:3)) + bi(1:3);
end
```

4. **Conditions aux limites** : La prise en compte des conditions aux limites $u_h(0) = u_h(1) = 0$, se déroule comme expliqué plus haut. Il faut juste faire attention ici à la numérotation des degrés de liberté, qui fait que celui situé en x=1, n'est pas le dernier degré de liberté, son numéro est N_{so} et non $N_{so} + N_{ma}$.

script Matlab d'introduction des conditions aux limites pour P2

```
function [A,b]=conditionLimiteLaplaceP2(x,A,b)
Nso = length(x); % nombre de sommets
Nma = Nso -1 ;   % nombre de maille
Ndl=Nso + Nma;

% Attention le numero du degre de liberte ? la position
% x(Nso) n'est pas Ndl mais plutot Nso

%CONDITION AUX LIMITES PAR PENALISATION
% tgv = 1e+30;
% A(1,1) = tgv; b(1) = tgv * g(x(1)); % g est la condition de dirichlet
% A(Nso,Nso) = tgv; b(Nso) = tgv * g(x(Nso));

%CONDITION AUX LIMITES PAR MODIFICATION
ud=zeros(Ndl,1);
ud(1)=g(x(1));ud(Nso)=g(x(Nso));
b=b-A*ud;b(1) = ud(1);b(Nso)=ud(Nso);
A(:,1) =0;A(1,:) =0;A(:,Nso) =0;A(Nso,:) =0;
A(1,1) =1;A(Nso,Nso) =1;
```

On peut alors créer la fonction `function [A,b] = laplaceP2(X)` qui assemble la matrice et le second membre de la discrétisation par éléments finis $P2-$Lagrange du Laplacien.

```
function [A,b,y]=laplaceP2(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [A,b]=laplaceP2(x)
% fonction qui assemble la matrice du laplacien P2
% et introduit les conditions aux limites
% Entree: x maillage
%
% Sortie: A matrice
%         b vecteur second membre global
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[A,b,y] = assembleLaplaceP2(x);
[A,b] = conditionLimiteLaplaceP2(x,A,b);
```

Exercice - 4 *Une comparaison*

On s'intéresse à présent au problème suivant

$$\left| \begin{array}{ll} -\epsilon u'' + u' = 1 & \text{dans }]0,1[, \\ u(0) = u(1) = 0 & . \end{array} \right. \tag{2}$$

La solution exacte de cette équation aux dérivées ordinaires est

$$u(x) = x - \frac{\exp(\frac{x}{\epsilon}) - 1}{\exp(\frac{1}{\epsilon}) - 1}.$$

On généralise l'étude précédente pour obtenir un nouveau problème. On cherche une solution $u \in H_0^1(0,1)$ du problème

$$\forall v \in H_0^1(0,1), \quad \epsilon \int_0^1 u'v' + \int_0^1 u'v = \int_0^1 v.$$

On utilise la méthode des éléments finis $P1$ puis $P2$ pour trouver une solution approchée de ce problème.

Q-1 **:** Le problème discret est ici

$$\boxed{\text{Chercher } u_h \in X_{0h}^r \text{ tel que } \quad \epsilon \int_\Omega u_h'(x) v_h'(x)\,dx + \int_\Omega u_h'(x) v_h(x) ddx = \int_\Omega f(x) v_h(x)\,dx \quad \forall v_h \in X_{0h}^r,}$$

Avec $r = 1$ pour la discrétisation $P1$ et $r = 2$ pour la discrétisation $P2$, avec $f(x) = 1$ la fonction second membre.

Q-2 **: Constructions pour P1-Lagrange**

1. On commence par traitement élémentaire. Soit I_i une maille ; on construit sa matrice élémentaire et son vecteur élémentaire : $A_{l,m}^i = \epsilon \int_{I_i} (\phi_l^i)'(\phi_m^i)'\,dx + \int_{I_i} (\phi_l^i)(\phi_m^i)'\,dx \quad 1 \le l, m \le 2$
   La matrice locale associée au terme $\int_{I_i} (\phi_l^i)'(\phi_m^i)'\,dx$ est identique à celle de l'exercice précédent, celle associée au terme $\int_{I_i} (\phi_l^i)(\phi_m^i)'\,dx$ est
   $$\int_{x_i}^{x_{i+1}} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \begin{bmatrix} \phi_1{}' & \phi_2{}' \end{bmatrix} dx = \frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}.$$
   D'où
   $$A^i = \frac{\epsilon}{h_i} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}.$$
   Le calcul de b^i reste inchangé. On a finalement le script

```
function [Ai,bi]=elementconvdiffP1(xi1,xi2,epsi, i)
hi=xi2 - xi1;
Ai = (epsi/hi)*[1,-1;-1,1] + (1/2)*[-1,1;-1,1];
bi=(hi/2)* [ f(xi1); f(xi2)];
```

2. Puis on écrit un script qui assemble les matrices et vecteurs élémentaires et introduit les conditions aux limites

```
function [A,b]=convdiffP1(epsi,x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [A,b] =convdiffP1(x)
% fonction qui assemble la matrice du problème de convection diffusion
% pour les polynomes de degre 1 et introduit les conditions aux limites
% ENTREE: x -> les noeuds du maillage
%         eps -> le coefficient epsilon
% SORTIE: A -> la matrice avec condition aux limites incorporée
%         b -> le vecteur second membre avec condition aux limites incorporée
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nso = length(x); % nombre de sommets
Nma = Nso -1 ;   % nombre de maille
% table des degre de liberté
A=zeros(Nso,Nso);
b=zeros(Nso,1);
for i = 1:Nma
    l2g(i,1) = i;
    l2g(i,2) = i+1;
end
for i=1:Nma
    %On recupère les sommets de l'élément
    x1 = x(i); x2 = x(i+1);
    % On calcule la matrice et le second membre élémentaire
    [Ai,bi] = elementconvdiffP1(x1,x2,epsi,i);
    % et on assemble la matrice et le vecteur élémentaires
    A(l2g(i,1:2), l2g(i,1:2)) = A(l2g(i,1:2), l2g(i,1:2)) + Ai(1:2,1:2);
    b(l2g(i,1:2)) = b(l2g(i,1:2)) + bi(1:2);
end
% condition aux limites
ud=zeros(Nso,1); ud(1)=g(x(1)); ud(Nso)=g(x(Nso));
b=b-A*ud;
b(1) = ud(1);b(Nso)=ud(Nso);
A(:,1) =0;A(1,:) =0;A(:,Nso) =0;A(Nso,:) =0;
A(1,1) =1;A(Nso,Nso) =1;
```

Q-3 : Constructions pour P2-Lagrange

1. Matrice et vecteur élémentaires

```
function [Ai,bi]=elementconvdiffP2(x1,x2,epsi,i)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [Ai,bi]=elementconvdiffP2(x1,x2,epsi,i)
% Assemble la matrice élémentaire et le second membre élémentaire
% du problème de convection diffusion pour P2-Lagrange
%
% Entrees:[x1,x2] intervalle
%         epsi coefficien epsilon
%         i numero de l'élément
% Sortie : Ai matrice 3 x 3
%         bi vecteur 3 x 1
%
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%formule de quadrature de simpson
w = [1, 4, 1]*(x2-x1)/6; %poids
z=[x1 , (x1+x2)/2, x2]; %points
%initialisation
Ai=zeros(3,3);bi=zeros(3,1);
%on boucle sur les poind de quadrature
for q=1:3
    wq=w(q);
    zq = z(q);
    %boucle sur les lignes
    for l=1:3
        [vl,dvl] = baselocalP2(x1,x2,l,zq);
        bi(l)= bi(l) + wq * f(zq)*vl;
        %boucle sur les colonnes
        for m=1:3
            [vm,dvm] = baselocalP2(x1,x2,m,zq);
            Ai(l,m)=Ai(l,m) + wq * (epsi*dvl * dvm + vl*dvm);
        end
    end
end
```

2. Assemblage et introduction des conditions aux limites

```
function [A,b,y]=convdiffP2(epsi,x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function [A,b,y] =convdiffP2(x)
%
% fonction qui assemble la matrice du problème de convection diffusion
% pour les polynomes de degre 2 et introduit les conditions aux limites
%
% ENTREE: x les noeuds du maillage
%         eps le coefficient epsilon
%
% SORTIE: A la matrice avec condition aux limites incorporée
%         b le vecteur second membre avec condition aux limites incorporée
%         y position des degres de liberté
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Nso = length(x); % nombre de sommets
Nma = Nso -1 ;   % nombre de mailles
Ndl=Nso + Nma;   % nombre de degrés de liberté
% table des degrés de liberte
% et leur position
y=[];l2g=[];
for i = 1:Nma
    l2g(i,1) = i;
    l2g(i,2) = i+Nso;
    l2g(i,3) = i+1;
    y(l2g(i,1))=x(i);
    y(l2g(i,3)) = x(i+1);
    y(l2g(i,2)) = (x(i)+x(i+1))/2;
end
%réservation d'espace
A=zeros(Ndl,Ndl);
```

```
b=zeros(Ndl,1);
% boucle sur les éléments
for i=1:Nma
    %On recupere les sommets de l'element
    x1 = x(i); x2 = x(i+1);
    % On calcule la matrice et le second membre elementaire
    [Ai,bi] = elementconvdiffP2(x1,x2,epsi,i);
    % et on assemble la matrice et le vecteur élémentaires dans la
    % matrice et le vecteur globale
    A(l2g(i,1:3), l2g(i,1:3)) = A(l2g(i,1:3), l2g(i,1:3)) + Ai(1:3,1:3);
    b(l2g(i,1:3)) = b(l2g(i,1:3)) + bi(1:3);
end
%conditions aux limites
ud=zeros(Ndl,1);ud(1)=g(x(1));ud(Nso)=g(x(Nso));
b=b-A*ud;
b(1) = ud(1);b(Nso)=ud(Nso);
A(:,1) =0;A(1,:) =0;A(:,Nso) =0;A(Nso,:) =0;
A(1,1) =1;A(Nso,Nso) =1;
```

Q-4 : Comparaisons

On commence par écrire un script pour la solution exacte :

Solution exacte

```
function [y] = gexact(x,epsi)
y=x - (exp(x/epsi)-1)/(exp(1/epsi)-1);
```

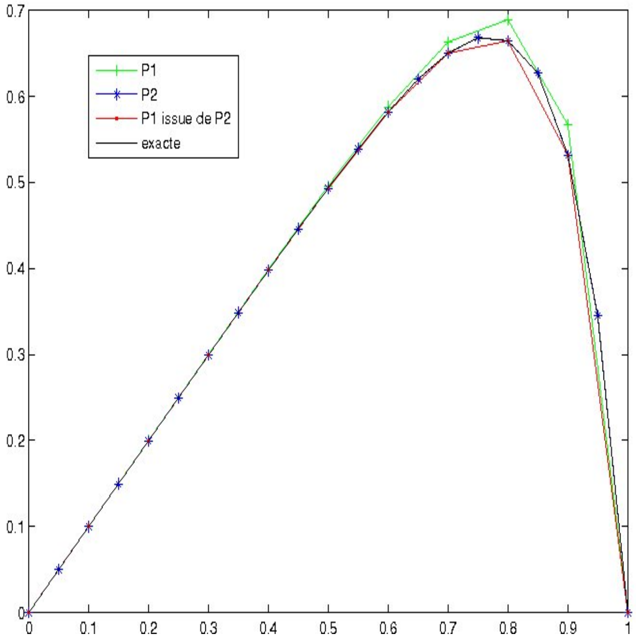
On introduit ensuite la fonction qui réalise le test pour un maillage x donné et une valeur de ϵ donnée.

Script de comparaison

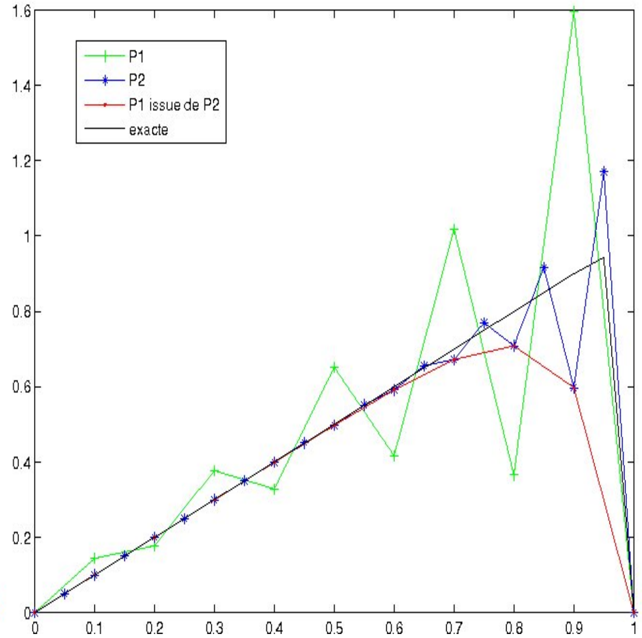
```
function  testconvdiff(x,epsi)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  testconvdiff(x,epsi)
%
% fonction qui compare les discrétisation P1, P2 Lagrange
% de l'équation de convection diffusion
% ENTREE: x   -> les noeuds du maillage
%        epsi -> valeur de epsilon
% SORTIE:
%  COURS D'ELEMENTS FINIS
%  M2PRO 2007-2008    Lab AN-EDP Univ d'ORSAY
% (c) Jean-Baptiste Apoung Kamga (July 2007)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%résolution P1
[A,b]=convdiffP1(epsi,x);
u=A\b;
%résolution P2
[A2,b2,y]=convdiffP2(epsi,x);
u2=A2\b2;
[dum,p]=sort(y);
%solution exacte
ue=gexact(y,epsi);
%représentation graphique
plot(x,u,'g+-',y(p),u2(p),'b*-',x,u2(1:length(x)),'r.-',y(p),ue(p),'k-');
legend('P1','P2','P1 issue de P2','exacte');
pause();
```

Représentation graphique de la solution exacte et la solution discrète sur les deux maillages.

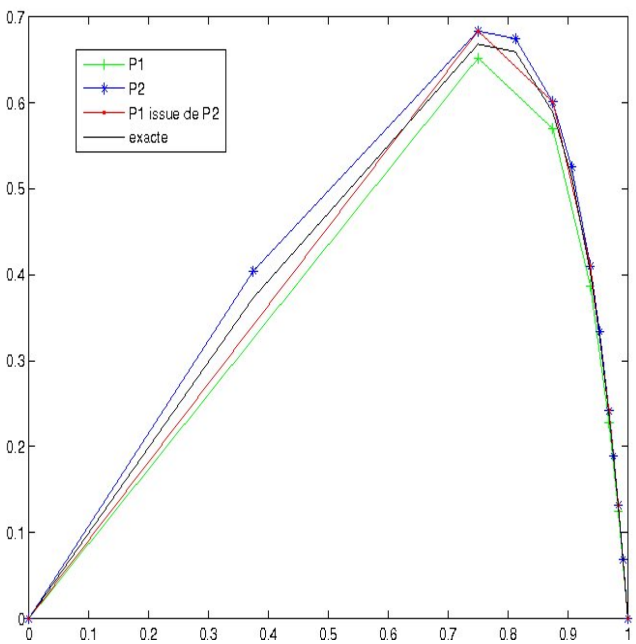


x=0 :0.1 :1; $\epsilon = 0.1$.

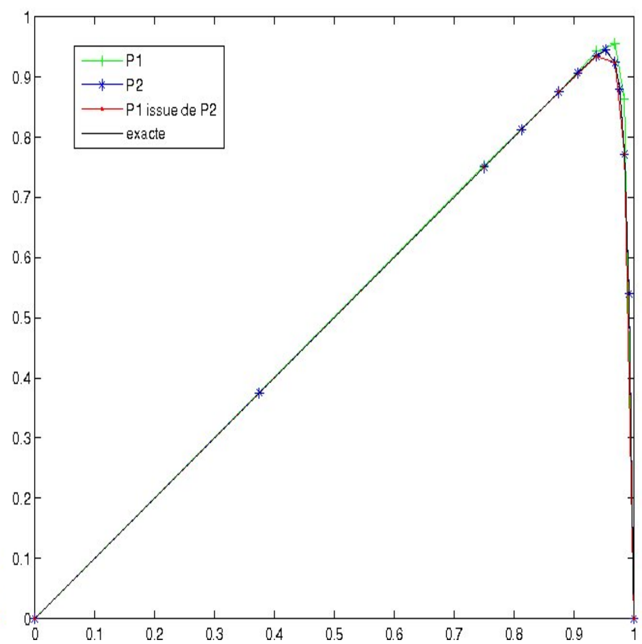


x=0 :0.1 :1; $\epsilon = 0.01$.

On reprend avec `x = [0, 0.75,0.875, 0.9375, 0.96875, 0.984375, 1];`.



$\epsilon = 0.1$



$\epsilon = 0.01$

Il vous reste à expliquer pour le problème considéré, les faits suivants :

1. **sur le maillage uniforme dont h est du même ordre que ϵ, il y a un bon comportement des éléments finis Lagrange. [*renseignez-vous sur le nombre de Péclet*].**
2. **l'interpolé P1 de la solution P2 présente moins d'oscillations et semble mieux se comporter même lorsque la solution P2 oscille.**
3. **le raffinement de maillage au voisinage de $x = 1$ conduit à un bon comportement des éléments finis Lagrange.[*renseignez-vous sur la couche limite*].**