Correction TD 2 : Résolution de systèmes linéaires et d'équations différentielles

1 Résolution de systèmes linéaires

Exercice 1 *Compléter l'ossature du code fournie afin que le programme résolve le système* $\mathbf{Ax} = \mathbf{b}$ *(avec* $\mathbf{A}$ *matrice carrée de taille* $m \times m$*) en utilisant la méthode de Jacobi.*

```
function [x] = Jacobi(A,b,x0,m,n)

   // on veut resoudre Ax = b en construisant une suite initialisee par x0
   // la matrice A est de taille m x m
   // on fait n iterations

   // remplir la matrice M (exemple pour le produit matriciel S*T)
   ........................
   Minv = M^(-1) ; // Minv contient la matrice inverse de M

   // remplir la matrice N
   ........................

   x = x0 ;
   for k = 1:n
      x = ........................
   end

endfunction
```

On veut résoudre le système $\mathbf{Ax} = \mathbf{b}$ avec la méthode de Jacobi. Celle-ci a besoin de construire la suite suivante

$$\begin{cases} \mathbf{x}_0 \text{ connu} \\ \mathbf{x}_{k+1} = \mathbf{M}^{-1}\mathbf{N}\mathbf{x}_k + \mathbf{M}^{-1}\mathbf{b} \end{cases}$$

avec $\mathbf{M} = \mathbf{D}$ la partie diagonale de la matrice $\mathbf{A}$ et $\mathbf{N} = \mathbf{E} + \mathbf{F}$, $-\mathbf{E}$ correspond à la partie triangulaire strictement inférieure de $\mathbf{A}$ et $-\mathbf{F}$ à la partie triangulaire strictement supérieure de $\mathbf{A}$. On a donc

$$\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}$$

Cette suite, sous certaines conditions, converge vers la solution du système $\mathbf{Ax} = \mathbf{b}$. Notons aussi que l'on peut exprimer $\mathbf{N}$ en fonction de $\mathbf{M}$ et $\mathbf{A}$ directement (cela nous sera utile pour le programme Scilab).

$$\mathbf{N} = \mathbf{E} + \mathbf{F} = \mathbf{D} - \mathbf{A} = \mathbf{M} - \mathbf{A}$$

On a maintenant tous les éléments pour remplir le programme Scilab.

```
function [x] = Jacobi(A,b,x0,m,n)

   // on veut resoudre Ax = b en construisant une suite initialisee par x0
   // la matrice A est de taille m x m
```

```
   // on fait n iterations

   // remplir la matrice M (exemple pour le produit matriciel S*T)
   M = zeros(m,m) // creation d'une matrice de taille m x m a coef nuls dans chaque case
   for k = 1:m
      M(k,k) = A(k,k) ;
   end
   // on pouvait aussi faire M = diag(diag(A)) ;
   // diag d'une matrice retourne un vecteur comprenant les elements
   // de la diagonale de la matrice
   // diag d'un vecteur retourne une matrice diagonale dont les elements diagonaux
   // sont ceux du vecteur
   Minv = M^(-1) ; // Minv contient la matrice inverse de M

   // remplir la matrice N
   N = M - A ;

   x = x0 ;
   for k = 1:n
      x = Minv*N*x + Minv*b ;
   end

endfunction
```

Exercice 2 *Voici 2 matrices décomposées de la manière suivante (on ne demande pas de vérifier l'égalité)*

$$\begin{pmatrix} 2 & -2 & 2 \\ -1 & 2 & -9 \\ -2 & 0 & -4 \end{pmatrix} = \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} -3 & 2 & -7 \\ 0 & -2 & 4 \\ 0 & 0 & -6 \end{pmatrix}$$

$$\begin{pmatrix} -8 & 21 & 3 \\ 4 & -3 & 3 \\ -2 & 3 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{4} & -\frac{3}{10} & 1 \end{pmatrix} \begin{pmatrix} -8 & 21 & 3 \\ 0 & \frac{15}{2} & \frac{9}{2} \\ 0 & 0 & -\frac{4}{10} \end{pmatrix}$$

1. *Sous quelles formes sont décomposées les matrices suivantes ? Justifier.*
2. *Quel est le rang de chaque matrice ? Justifier*
3. *Résoudre les systèmes suivants en utilisant les questions précédentes*

$$\begin{pmatrix} 2 & -2 & 2 \\ -1 & 2 & -9 \\ -2 & 0 & -4 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 2 \\ 4 \\ -3 \end{pmatrix}$$

$$\begin{pmatrix} -8 & 21 & 3 \\ 4 & -3 & 3 \\ -2 & 3 & -1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 8 \\ 20 \\ -8 \end{pmatrix}$$

1. La première décomposition est une décomposition QR avec

$$\mathbf{Q} = \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix} \qquad \mathbf{R} = \begin{pmatrix} -3 & 2 & -7 \\ 0 & -2 & 4 \\ 0 & 0 & -6 \end{pmatrix}$$

On voit facilement que $\mathbf{R}$ est une matrice triangulaire supérieure. Il nous faut aussi vérifier que $\mathbf{Q}$ est une matrice orthogonale. Pour cela, on effectue le calcul suivant

$$\mathbf{Q}\mathbf{Q}^T = \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{4}{9}+\frac{1}{9}+\frac{4}{9} & -\frac{2}{9}-\frac{2}{9}+\frac{4}{9} & -\frac{4}{9}+\frac{2}{9}+\frac{2}{9} \\ -\frac{2}{9}-\frac{2}{9}+\frac{4}{9} & \frac{1}{9}+\frac{4}{9}+\frac{4}{9} & \frac{2}{9}-\frac{4}{9}+\frac{2}{9} \\ -\frac{4}{9}+\frac{2}{9}+\frac{2}{9} & \frac{2}{9}-\frac{4}{9}+\frac{2}{9} & \frac{4}{9}+\frac{4}{9}+\frac{1}{9} \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

On a donc bien $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ (matrice identité : des 1 sur la diagonale et des 0 partout ailleurs). Donc $\mathbf{Q}$ est une matrice orthogonale.

La seconde décomposition est une décomposition LU avec

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{4} & -\frac{3}{10} & 1 \end{pmatrix} \qquad \mathbf{U} = \begin{pmatrix} -8 & 21 & 3 \\ 0 & \frac{15}{2} & \frac{9}{2} \\ 0 & 0 & -\frac{4}{10} \end{pmatrix}$$

En effet, $\mathbf{L}$ est une matrice triangulaire inférieure et $\mathbf{U}$ une matrice triangulaire supérieure.

2. On peut montrer que le rang des deux matrices étudiées est 3. Étudions d'abord la première matrice

$$\begin{pmatrix} 2 & -2 & 2 \\ -1 & 2 & -9 \\ -2 & 0 & -4 \end{pmatrix}$$

Clairement, les deux premières lignes sont indépendantes (on ne peut pas exprimer la seconde ligne en fonction de la première de manière linéaire) et le rang de la matrice est forcément supérieur ou égal à 2 (il est forcément inférieur ou égal à 3 car on travaille sur une matrice 3×3). Cherchons maintenant si la troisième ligne l_3 peut être écrite comme une combinaison linéaire des deux premières lignes l_1 et l_2. Cela à chercher deux réels λ et μ tel que

$$l_3 = \lambda l_1 + \mu l_2$$

et donc de voir si le système suivant a une solution

$$\begin{cases} 2\lambda - \mu &= -2 \\ -2\lambda + 2\mu &= 0 \\ 2\lambda - 9\mu &= -4 \end{cases}$$

Ce système n'admet aucune solution. On note l'_1, l'_2 et l'_3 les lignes de ce système. On a du côté gauche de l'égalité

$$l'_3 + 7l'_1 - 8l'_2 = 2\lambda - 9\mu + 7(2\lambda - \mu) + 8(-2\lambda + 2\mu) = (2 + 14 - 16)\lambda + (-9 - 7 + 16)\mu = 0$$

et du côté droit de l'égalité

$$l'_3 + 7l'_1 - 8l'_2 = -4 - 14 + 0 = -18$$

D'où, $0 = -18$ si ce système admet des solutions, ce qui n'est pas possible. Ce système n'admet pas de solutions donc les lignes de la matrices sont linéairement indépendantes et le **rang de la matrice est** 3.

Étudions maintenant la deuxième matrice

$$\begin{pmatrix} -8 & 21 & 3 \\ 4 & -3 & 3 \\ -2 & 3 & -1 \end{pmatrix}$$

Clairement, les deux premières lignes sont indépendantes et le rang de la matrice est forcément supérieur ou égal à 2. Cherchons maintenant si la troisième ligne l_3 peut être écrite comme une combinaison linéaire des deux premières lignes l_1 et l_2. Cela à chercher deux réels λ et μ tel que

$$l_3 = \lambda l_1 + \mu l_2$$

et donc de voir si le système suivant a une solution

$$\begin{cases} -8\lambda + 4\mu &= -2 \\ 21\lambda - 3\mu &= 3 \\ 3\lambda + 3\mu &= -1 \end{cases}$$

Ce système n'admet aucune solution. On note l'_1, l'_2 et l'_3 les lignes de ce système. On a du côté gauche de l'égalité

$$l_2 + 2l_1 - \frac{5}{3}l_3 = \lambda(21 - 16 - 5) + \mu(-3 + 8 - 5) = 0$$

et du côté droit

$$l_2 + 2l_1 - \frac{5}{3}l_3 = 3 - 4 + \frac{5}{3} = \frac{2}{3}$$

D'où, $0 = \frac{2}{3}$ si ce système admet des solutions, ce qui n'est pas possible. Ce système n'admet pas de solutions donc les lignes de la matrices sont linéairement indépendantes et le **rang de la matrice est** 3.

3. On veut résoudre le système suivant

$$\begin{pmatrix} 2 & -2 & 2 \\ -1 & 2 & -9 \\ -2 & 0 & -4 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 2 \\ 4 \\ -3 \end{pmatrix}$$

Or dans la question 1, on a étudié la décomposition QR de la matrice du système et pour résoudre le système, il faut tout d'abord calculer

$$\mathbf{y} = \mathbf{Q}^T \begin{pmatrix} 2 \\ 4 \\ -3 \end{pmatrix} = \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 2 \\ 4 \\ -3 \end{pmatrix} = \begin{pmatrix} -\frac{4}{3} + \frac{4}{3} - 2 \\ \frac{2}{3} - \frac{8}{3} - 2 \\ \frac{4}{3} + \frac{8}{3} - 1 \end{pmatrix} = \begin{pmatrix} -2 \\ -4 \\ 3 \end{pmatrix}$$

Puis, il faut résoudre le système $\mathbf{Rx} = \mathbf{y}$. Notons $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, on veut donc résoudre le système suivant

$$\left\{\begin{array}{rcl} -3x_1+2x_2-7x_3 &=& -2 \\ -2x_2+4x_3 &=& -4 \\ -6x_3 &=& 3 \end{array}\right.$$

On a immédiatement que $x_3 = -\frac{1}{2}$, d'où avec la seconde ligne $x_2 = 1$ et finalement avec la première ligne $x_1 = \frac{5}{2}$. D'où

$$\mathbf{x} = \begin{pmatrix} \frac{5}{2} \\ 1 \\ -\frac{1}{2} \end{pmatrix}$$

On veut maintenant résoudre le système suivant

$$\begin{pmatrix} -8 & 21 & 3 \\ 4 & -3 & 3 \\ -2 & 3 & -1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 8 \\ 20 \\ -8 \end{pmatrix}$$

Or dans la question 1, on a étudié la décomposition LU de la matrice du système. Pour le résoudre, il faut tout d'abord résoudre le système $\mathbf{Ly} = \mathbf{b}$. Si on note $\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$, cela revient à résoudre le système suivant

$$\left\{\begin{array}{rcl} y_1 &=& 8 \\ -\frac{1}{2}y_1 + y_2 &=& 20 \\ \frac{1}{4}y_1 - \frac{3}{10}y_2 + y_3 &=& -8 \end{array}\right.$$

On a immédiatement $y_1 = 8$, d'où avec la deuxième ligne $y_2 = 24$ et avec la troisième ligne $y_3 = -10 + \frac{36}{5} = -\frac{14}{5}$. On doit ensuite résoudre le système $\mathbf{Ux} = \mathbf{y}$. Notons $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$, on veut donc résoudre le système suivant

$$\left\{\begin{array}{rcl} -8x_1 + 21x_2 + 3x_3 &=& 8 \\ \frac{15}{2}x_2 + \frac{9}{2}x_3 &=& 24 \\ -\frac{4}{10}x_3 = -\frac{14}{5} & & \end{array}\right.$$

On a directement $x_3 = \frac{14}{5} \times \frac{10}{4} = 7$. Puis

$$x_2 = \frac{2}{15}(24 - \frac{9 \times 7}{2}) = \frac{1}{15}(48 - 63) = -1$$

et

$$x_1 = -\frac{1}{8}(8 - 21 + 21) = -1$$

D'où

$$\mathbf{x} = \begin{pmatrix} -1 \\ -1 \\ 7 \end{pmatrix}$$

2 Résolution d'équations différentielles

Exercice 3 *Soit une fonction $u(x,t)$ solution de l'équation différentielle suivante (équation de la chaleur)*

$$\frac{\partial u}{\partial t} - c\frac{\partial^2 u}{\partial x^2} = 0$$

avec :

- $t \geq 0$, *variable temporelle*
- x, *variable spatiale comprise entre* 0 *et* 1
- $c > 0$ *coefficient dit de diffusion thermique*
- $u(x,0) = u_0(x)$ *avec* u_0 *fonction connue*
- $u(0,t) = \alpha(t)$ *avec* α *fonction connue*
- $u(1,t) = \beta(t)$ *avec* β *fonction connue*

On veut résoudre de manière numérique cette équation. Pour cela, on subdivise le temps et l'espace (= discrétiser) de la manière suivante

- *pour l'espace,* $x_i = ih$, $i = 0, \ldots, n+1$ *avec* $h = \frac{1}{n+1}$
- *pour le temps,* $t_j = j\Delta t$, *avec* $\Delta t > 0$ *un pas de temps* ***choisi****.*
- *on note* $u_i^j = u(x_i, t_j)$

Les questions suivantes doivent vous permettre de poser le problème discret (version discrète du problème continu, soit passer de x et t aux x_i et t_j) et vous conduire aux systèmes linéaires à résoudre.

1. *Même sans la résolution de l'équation différentielle, on connait déjà un certain nombre de* u_i^j. *Lesquels ?*

On suppose maintenant qu'on connaisse à un instant t_j donné ***tous*** *les* u_i^j, $i = 0, \ldots, n+1$. *Par contre, on ne connait pas tous les* u_i^{j+1} *à l'instant suivant* t_{j+1}. *Le but des prochaines questions est de pouvoir exprimer les* u_i^{j+1} *en fonction des* u_i^j *en discrétisant l'équation*

$$\frac{\partial u}{\partial t}(x_i, t_j) - c\frac{\partial^2 u}{\partial x^2}(x_i, t_j) = 0$$

*Cette méthode est dite d'****Euler explicite***

2. *Discrétiser le terme suivant* $\frac{\partial u}{\partial t}(x_i, t_j)$ *en fonction uniquement de* u_i^j, u_i^{j+1} *et* Δt. *Quel est l'ordre de l'approximation en temps ?*
3. *Discrétiser le terme suivant* $\frac{\partial^2 u}{\partial x^2}(x_i, t_j)$ *en fonction uniquement de* u_{i-1}^j, u_i^j, u_{i+1}^j *et* h.
   Quel est l'ordre de l'approximation en espace ?
4. *Établir le système à résourdre pour trouver les* u_i^{j+1} *en fonction des* u_i^j.

On suppose toujours qu'on connaisse à un instant t_j donné ***tous*** *les* u_i^j, $i = 0, \ldots, n+1$. *On va construire les* u_i^{j+1} *en fonction des* u_i^j *en discrétisant l'équation*

$$\frac{\partial u}{\partial t}(x_i, t_{j+1}) - c\frac{\partial^2 u}{\partial x^2}(x_i, t_{j+1}) = 0$$

*Cette méthode est dite d'****Euler implicite***

5. *Discrétiser le terme suivant* $\dfrac{\partial u}{\partial t}(x_i, t_{j+1})$ *en fonction uniquement de* u_i^j, u_i^{j+1} *et* Δt.
Quel est l'ordre de l'approximation en temps ?

6. *Discrétiser le terme suivant* $\dfrac{\partial^2 u}{\partial x^2}(x_i, t_{j+1})$ *en fonction uniquement de* u_{i-1}^{j+1}, u_i^{j+1}, u_{i+1}^{j+1} *et* h.
Quel est l'ordre de l'approximation en espace ?

7. *Établir le système à résourdre pour trouver les* u_i^{j+1} *en fonction des* u_i^j.

8. *D'un point de vue pratique, quelle méthode (Euler explicite ou implicite) semble la plus facile à mettre en place ? Détailler votre point de vue.*

1. On sait que $u(x,0) = u_0(x)$ pour x réel dans $[0,1]$ avec u_0 une fonction connue (condition initiale sur la fonction u). Comme l'égalité est valable pour tout x de $[0,1]$, elle est aussi valable pour les x_i, $i = 0, \ldots, n+1$. On a donc $u(x_i, 0) = u_0(x_i)$. D'autre part, le temps initial noté t_0 vaut 0. On connait donc

$$u_i^0 = u(x_i, t_0) = u(x_i, 0) = u_0(x_i) \qquad i = 0, \ldots, n+1$$

On a aussi que $u(0,t) = \alpha(t)$ pour t réel positif avec α une fonction connue (condition au bord $x = 0$ sur la fonction u). Comme l'égalité est valable pour tout t réel positif, elle est aussi valable pour les t_j, j entier positif. On a donc $u(0, t_j) = \alpha(t_j)$. On rappelle aussi que $x_0 = 0$, donc on connait

$$u_0^j = u(x_0, t_j) = u(0, t_j) = \alpha(t_j) \qquad j \text{ entier } \geq 0$$

On a enfin que $u(1,t) = \beta(t)$ pour t réel positif avec β une fonction connue (condition au bord $x = 1$ sur la fonction u). Comme l'égalité est valable pour tout t réel positif, elle est aussi valable pour les t_j, j entier positif. On a donc $u(1, t_j) = \beta(t_j)$. On rappelle aussi que $x_{n+1} = 1$, donc on connait

$$u_{n+1}^j = u(x_{n+1}, t_j) = u(1, t_j) = \beta(t_j) \qquad j \text{ entier } \geq 0$$

Au final, on connait au préalable

$$\begin{cases} u_i^0 & i = 0, \ldots, n \\ u_0^j & j \text{ entier } \geq 0 \\ u_{n+1}^j & j \text{ entier } \geq 0 \end{cases}$$

Il reste à découvrir les u_i^j, $i = 1, \ldots, n$, j entier ≥ 1. Pour cela, il faut résoudre de manière discrète l'équation différentielle.

2. On veut approcher le terme $\dfrac{\partial u}{\partial t}(x_i, t_j)$ en utilisant u_i^j, u_i^{j+1} et Δt. Il s'agit donc d'approcher la dérivée temporelle de u en (x_i, t_j), $i = 1, \ldots, n$, j entier ≥ 1. Pour cela, on peut utiliser la formule de Taylor-Lagrange au point (x_i, t_{j+1}), on a

$$u(x_i, t_{j+1}) = u(x_i, t_j) + \Delta t \frac{\partial u}{\partial t}(x_i, t_j) + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \xi_j) \qquad \xi_j \in [t_j, t_{j+1}]$$

On peut faire l'approximation suivante

$$\frac{\partial u}{\partial t}(x_i, t_j) \approx \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} = \frac{u_i^{j+1} - u_i^j}{\Delta t}$$

et l'erreur commise sur $\dfrac{\partial u}{\partial t}(x_i, t_j)$ vaut

$$\frac{\partial u}{\partial t}(x_i, t_j) - \frac{u_i^{j+1} - u_i^j}{\Delta t} = -\frac{1}{\Delta t} \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \xi_j) = -\frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \xi_j)$$

L'erreur commise s'exprime par un terme en Δt le pas de temps, l'approximation est d'ordre 1. (Δt est à la puissance 1).

3. On veut approcher le terme $\dfrac{\partial^2 u}{\partial x^2}(x_i, t_j)$ en utilisant u_{i-1}^j, u_i^j, u_{i+1}^j et h. Il s'agit donc d'approcher la dérivée seconde en espace de u en (x_i, t_j), $i = 1, \ldots, n$, j entier ≥ 1. Pour cela, on peut utiliser la formule de Taylor-Lagrange aux points (x_{i+1}, t_j) et (x_{i-1}, t_j), on a ainsi

$$u(x_{i+1}, t_j) = u(x_i, t_j) + h\frac{\partial u}{\partial x}(x_i, t_j) + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}(x_i, t_j) + \frac{h^3}{6}\frac{\partial^3 u}{\partial x^3}(x_i, t_j) + \frac{h^4}{24}\frac{\partial^4 u}{\partial x^4}(\xi_i, t_j) \qquad \xi_i \in [x_i, x_{i+1}]$$

$$u(x_{i-1}, t_j) = u(x_i, t_j) - h\frac{\partial u}{\partial x}(x_i, t_j) + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}(x_i, t_j) - \frac{h^3}{6}\frac{\partial^3 u}{\partial x^3}(x_i, t_j) + \frac{h^4}{24}\frac{\partial^4 u}{\partial x^4}(\eta_i, t_j) \qquad \eta_i \in [x_{i-1}, x_i]$$

On peut faire l'approximation suivante

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) \approx \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{h^2} = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2}$$

et l'erreur commise sur $\dfrac{\partial^2 u}{\partial x^2}(x_i, t_j)$ vaut

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) - \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} = -\frac{1}{h^2}\left(\frac{h^4}{24}\frac{\partial^4 u}{\partial x^4}(\xi_i, t_j) + \frac{h^4}{24}\frac{\partial^4 u}{\partial x^4}(\eta_i, t_j)\right) = -\frac{h^2}{24}\left(\frac{\partial^4 u}{\partial x^4}(\xi_i, t_j) + \frac{\partial^4 u}{\partial x^4}(\eta_i, t_j)\right)$$

L'estimation de l'erreur peut se simplifier en utilisant le théorème des valeurs intermédiaires (u est suffisament dérivable en x), on a

$$\frac{1}{2}\left(\frac{\partial^4 u}{\partial x^4}(\xi_i, t_j) + \frac{\partial^4 u}{\partial x^4}(\eta_i, t_j)\right) = \frac{\partial^4 u}{\partial x^4}(\zeta_i, t_j) \qquad \zeta_i \in [x_{i-1}, x_{i+1}]$$

et

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) - \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} = -\frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\zeta_i, t_j)$$

L'erreur commise s'exprime par un terme en h^2 avec h le pas d'espace, l'approximation est d'ordre 2. (h est à la puissance 2).

4. On veut approcher l'équation différentielle en (x_i, t_j), $i = 1, \ldots, n$, j entier ≥ 1

$$\frac{\partial u}{\partial t}(x_i, t_j) - c\frac{\partial^2 u}{\partial x^2}(x_i, t_j) = 0$$

par

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} - c\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} = 0$$

Supposons que l'on connaisse à un instant donné t_j tous les u_i^j (j est fixé seul i varie), on veut connaitre à l'instant suivant t_{j+1} les u_i^{j+1}. Grâce à la relation précédente, on peut écrire à j fixé pour i variant de 1 à n que

$$u_i^{j+1} = \frac{c\Delta t}{h^2}u_{i-1}^j + \left(1 - \frac{2c\Delta t}{h^2}\right)u_i^j + \frac{c\Delta t}{h^2}u_{i+1}^j$$

En particulier, si $i = 1$,

$$u_1^{j+1} = \frac{c\Delta t}{h^2}\underbrace{\alpha(t_j)}_{=u_0^j} + \left(1 - \frac{2c\Delta t}{h^2}\right)u_2^j + \frac{c\Delta t}{h^2}u_3^j$$

et si $i = n$

$$u_n^{j+1} = \frac{c\Delta t}{h^2} u_{n-1}^j + \left(1 - \frac{2c\Delta t}{h^2}\right) u_n^j + \frac{c\Delta t}{h^2} \underbrace{\beta(t_j)}_{u_{n+1}^j}$$

On peut mettre tout ceci sous forme vectorielle en reliant le vecteur $\begin{pmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{n-1}^{j+1} \\ u_n^{j+1} \end{pmatrix}$ au vecteur $\begin{pmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_{n-1}^j \\ u_n^j \end{pmatrix}$.
Notons bien que dans les deux vecteurs sont exclues les valeurs de u aux bords $x = 0$ et $x = 1$ car on les connait déjà. On peut alors écrire le système suivant

$$\begin{pmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{n-1}^{j+1} \\ u_n^{j+1} \end{pmatrix} = \begin{pmatrix} 1 - 2\frac{c\Delta t}{h^2} & \frac{c\Delta t}{h^2} & 0 & \dots & 0 \\ \frac{c\Delta t}{h^2} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{c\Delta t}{h^2} \\ 0 & \dots & 0 & \frac{c\Delta t}{h^2} & 1 - 2\frac{c\Delta t}{h^2} \end{pmatrix} \begin{pmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_{n-1}^j \\ u_n^j \end{pmatrix} + \begin{pmatrix} \frac{c\Delta t}{h^2}\alpha(t_j) \\ 0 \\ \vdots \\ 0 \\ \frac{c\Delta t}{h^2}\beta(t_j) \end{pmatrix}$$

Le système n'a pas besoin d'être résolu, on peut calculer directement les u_i^{j+1} à partir des u_i^j.

5. On veut approcher le terme $\dfrac{\partial u}{\partial t}(x_i, t_{j+1})$ en utilisant u_i^j, u_i^{j+1} et Δt. Il s'agit donc d'approcher la dérivée temporelle de u en (x_i, t_{j+1}), $i = 1, \dots, n$, j entier ≥ 1. Pour cela, on peut utiliser la formule de Taylor-Lagrange au point (x_i, t_j), on a

$$u(x_i, t_j) = u(x_i, t_{j+1}) - \Delta t \frac{\partial u}{\partial t}(x_i, t_{j+1}) + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \xi_j) \qquad \xi_j \in [t_j, t_{j+1}]$$

On peut faire l'approximation suivante

$$\frac{\partial u}{\partial t}(x_i, t_{j+1}) \approx \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} = \frac{u_i^{j+1} - u_i^j}{\Delta t}$$

et l'erreur commise sur $\dfrac{\partial u}{\partial t}(x_i, t_{j+1})$ vaut

$$\frac{\partial u}{\partial t}(x_i, t_{j+1}) - \frac{u_i^{j+1} - u_i^j}{\Delta t} = \frac{1}{\Delta t} \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \xi_j) = \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \xi_j)$$

L'erreur commise s'exprime par un terme en Δt le pas de temps, l'approximation est d'ordre 1. (Δt est à la puissance 1).

6. On veut approcher le terme $\dfrac{\partial^2 u}{\partial x^2}(x_i, t_{j+1})$ en utilisant u_{i-1}^{j+1}, u_i^{j+1}, u_{i+1}^{j+1} et h. Il s'agit donc d'approcher la dérivée seconde en espace de u en (x_i, t_{j+1}), $i = 1, \dots, n$, j entier ≥ 1. On peut reprendre directement ce qui a été fait à la question 3 en remplaçant j par $j + 1$, on a donc l'approximation suivante

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_{j+1}) \approx \frac{u(x_{i+1}, t_{j+1}) - 2u(x_i, t_{j+1}) + u(x_{i-1}, t_{j+1})}{h^2} = \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2}$$

L'approximation reste d'ordre 2.

7. On veut approcher l'équation différentielle en (x_i, t_{j+1}), $i = 1, \ldots, n$, j entier ≥ 1

$$\frac{\partial u}{\partial t}(x_i, t_{j+1}) - c\frac{\partial^2 u}{\partial x^2}(x_i, t_{j+1}) = 0$$

par

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} - c\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2} = 0$$

Supposons que l'on connaisse à un instant donné t_j tous les u_i^j (j est fixé seul i varie), on veut connaitre à l'instant suivant t_{j+1} les u_i^{j+1}. Grâce à la relation précédente, on peut écrire à j fixé pour i variant de 1 à n que

$$-\frac{c\Delta t}{h^2}u_{i-1}^{j+1} + \left(1 + \frac{2c\Delta t}{h^2}\right)u_i^{j+1} - \frac{c\Delta t}{h^2}u_{i+1}^{j+1} = u_i^j$$

En particulier, si $i = 1$,

$$-\frac{c\Delta t}{h^2}\underbrace{\alpha(t_{j+1})}_{=u_0^{j+1}} + \left(1 + \frac{2c\Delta t}{h^2}\right)u_1^{j+1} - \frac{c\Delta t}{h^2}u_2^{j+1} = u_1^j$$

et si $i = n$

$$-\frac{c\Delta t}{h^2}u_{n-1}^{j+1} + \left(1 + \frac{2c\Delta t}{h^2}\right)u_n^{j+1} - \frac{c\Delta t}{h^2}\underbrace{\beta(t_{j+1})}_{=u_{n+1}^{j+1}} = u_n^j$$

On peut mettre tout ceci sous forme vectorielle en reliant le vecteur $\begin{pmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{n-1}^{j+1} \\ u_n^{j+1} \end{pmatrix}$ au vecteur $\begin{pmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_{n-1}^j \\ u_n^j \end{pmatrix}$.

Notons bien que dans les deux vecteurs sont exclues les valeurs de u aux bords $x = 0$ et $x = 1$ car on les connait déjà. On peut alors écrire le système suivant

$$\begin{pmatrix} 1 + 2\frac{c\Delta t}{h^2} & -\frac{c\Delta t}{h^2} & 0 & \ldots & 0 \\ -\frac{c\Delta t}{h^2} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\frac{c\Delta t}{h^2} \\ 0 & \ldots & 0 & -\frac{c\Delta t}{h^2} & 1 + 2\frac{c\Delta t}{h^2} \end{pmatrix} \begin{pmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{n-1}^{j+1} \\ u_n^{j+1} \end{pmatrix} = \begin{pmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_{n-1}^j \\ u_n^j \end{pmatrix} + \begin{pmatrix} \frac{c\Delta t}{h^2}\alpha(t_{j+1}) \\ 0 \\ \vdots \\ 0 \\ \frac{c\Delta t}{h^2}\beta(t_{j+1}) \end{pmatrix}$$

Cette fois-ci, on ne peut plus calculer directement les u_i^{j+1} à partir des u_i^j. Il faut résoudre un système linéaire.

8. D'un point de vue pratique, la méthode d'Euler explicite (la première) est plus pratique à mettre en oeuvre que la méthode d'Euler implicite (la seconde) car il n'y a pas de systèmes linéaires à résoudre. Cependant, la méthode d'Euler implicite a quelques propriétés plus intéressantes que la méthode d'Euler explicite (notamment a propos d'explosions numériques) mais le temps ne nous permet pas de les étudier dans le présent cours.

3 Extrait de l'examen de 2012

3.1 Représentation des nombres (4 points)

Lors d'un calcul, un ordinateur fournit des réponses approximatives pour des raisons de cardinalité. En effet, il utilise un nombre fini de bits (chiffre binaire c'est à dire 0 ou 1) pour représenter les entiers ou les réels. Pour comprendre comment l'ordinateur effectue ses calculs et éviter les situations pathogènes, il est intéressant de comprendre le calcul en système binaire et la représentation des réels en nombres à virgule flottante.

Question 1 *"There are only 10 types of people in the world : those who understand binary and those who don't." Expliquer la blague.*

La réponse est simple. En binaire, 10 veut dire 2. Si on traduit en français la phrase, on a donc : "Il y a seulement 10 (= 2 en binaire) types de personnes dans le monde : ceux qui comprennent le binaire et ceux qui ne le comprennent pas". A priori, si vous avez compris la blague, c'est que vous comprenez le binaire.

Question 2 *Convertir les nombres 12 et 13 en binaire. Effectuer l'addition 12+13 en binaire.*

Convertissons tout d'abord 12 en binaire. Cela donne

$$\begin{array}{rcrcrcr} 12 & = & 2 & \times & 6 & + & \mathbf{0} \\ 6 & = & 2 & \times & 3 & + & \mathbf{0} \\ 3 & = & 2 & \times & 1 & + & \mathbf{1} \\ 1 & = & 2 & \times & 0 & + & \mathbf{1} \end{array}$$

On a donc $(12)_{10} = (1100)_2$. Convertissons maintenant 13 en binaire. On a

$$\begin{array}{rcrcrcr} 13 & = & 2 & \times & 6 & + & \mathbf{1} \\ 6 & = & 2 & \times & 3 & + & \mathbf{0} \\ 3 & = & 2 & \times & 1 & + & \mathbf{1} \\ 1 & = & 2 & \times & 0 & + & \mathbf{1} \end{array}$$

On a ainsi $(13)_{10} = (1101)_2$. On effectue maintenant l'addition de $(1100)_2$ et $(1101)_2$. Pour rappel, l'addition en binaire fonctionne de la manière suivante

+	0	1
0	0	1
1	1	10

D'où l'opération suivante

$$\begin{array}{rccccc} & & 1 & 1 & 0 & 0 \\ + & & 1 & 1 & 0 & 1 \\ \hline = & 1 & {}^{1}1 & {}^{1}0 & 0 & 1 \end{array}$$

On a $(1100)_2 + (1101)_2 = (11001)_2$. Or $(12)_{10} + (13)_{10} = (25)_{10}$, vérifions si $(25)_{10} = (11001)_2$.

$$\begin{array}{rcrcrcr} 2 & \times & 0 & + & \mathbf{1} & = & 1 \\ 2 & \times & 1 & + & \mathbf{1} & = & 3 \\ 2 & \times & 3 & + & \mathbf{0} & = & 6 \\ 2 & \times & 6 & + & \mathbf{0} & = & 12 \\ 2 & \times & 12 & + & \mathbf{1} & = & 25 \end{array}$$

On a bien $(25)_{10} = (11001)_2$, le résultat obtenu en binaire est bien conforme au résultat obtenu en base 10.

Question 3 *Comment sont représentés les réels sur un ordinateur ? Quels sont les avantages et les inconvénients d'une telle représentation ? (4 lignes maximum)*

Les réels sont stockés sur un ordinateur sous forme de flottants en base 2, soit par 2 nombres écrits en binaire, la mantisse contenant la valeur normalisée du nombre que l'on veut représenter et l'exposant qui définit le décalage par rapport à la normalisation. L'avantage principal est la rapidité des calculs sur ces nombres (opérations en binaires faciles), l'inconvéniant principal est la perte de précision sur certains réels.

Question 4 *Expliquer les résultats de la session Scilab présentée ci-dessous :*

```
->
->a=1
a = 1.
->b=10^ 20
b = 1.000D+20
->c=-b
c = - 1.000D+20
->(a+b)+c
ans = 0.
->a+(b+c)
ans = 1.
```

Les 3 premières lignes sont évidentes à comprendre, on affecte correctement 1 à a, 10^{20} à b et -10^{20} à c. La quatrième ligne ne donne pas le résultat attendu car il effectue tout d'abord l'opération $a + b$ mais en pratique du fait de l'écart trop important entre a et b, l'ordinateur va évaluer $a + b$ à 10^{20} et non $10^{20} + 1$. Puis l'ordinateur soustrait 10^{20} au résultat précédent et on obtient 0 au lieu du 1 attendu. La cinquième ligne permet un ordre des calcul évitant ce problème, mais il suffit de remplacer a par b et vice versa dans cette ligne pour retrouver le même genre de problème qu'à la ligne précédente.

3.2 Integration (3 points)

Soit g une fonction de [0,1] dans $\mathbb{R}$, en général les méthodes analytiques d'évaluation de $I = \int_0^1 g(x)dx$ se basent sur le calcul d'une primitive de g. Cependant, pour la plupart des fonctions, on ne connaît pas d'expression analytique des primitives. Il est alors intéressant d'utiliser une méthode numérique. On considère la méthode de quadrature de la forme :

$$\int_0^1 g(x)dx \approx g(0) \tag{1}$$

Question 5 *Donner l'interprétation géométrique d'une telle méthode.*

Il faut tout d'abord remarquer que l'approximation vaut

$$\int_0^1 g(x)dx \approx g(0) = g(0) \times (1 - 0)$$

$g(0) \times (1 - 0)$ peut être vu comme l'aire du rectangle de longueur $g(0)$ en ordonnée et de largeur $1 - 0$ entre les abscisses 0 et 1. On retrouve ici la formule des rectangles (formule de Newton-Cotes fermé $n = 0$)

Question 6 *Quelle est le degré de précision de ce schéma ?*

On a vu dans le cours que la formule des rectangles (dans le cas Newton-Cotes fermé) était de degré de précision 0. Néanmoins, si on n'avait pas reconnu la formule des rectangles ici, on pouvait directement le vérifier. En effet, si $g(x) = 1$ sur $[0, 1]$,

$$\int_0^1 g(x)\,dx = 1 \qquad g(0) = 1$$

La formule de quadrature est donc exacte pour $g(x) = 1$, elle est de degré de précision au moins 0. Si $g(x) = x$ sur $[0, 1]$,

$$\int_0^1 g(x)\,dx = \frac{1}{2} \qquad g(0) = 0$$

La formule de quadrature n'est pas exacte pour $g(x) = x$, son degré de précision est donc bien 0.

Question 7 *Cette méthode parait-elle intéressante ? Justifier.*

Cette méthode n'est pas intéressante car l'erreur commise est rapidement importante (voir à la question précédente pour $g(x) = x$) même pour des fonctions g vraiment simple. Elle serait plus intéressante sous forme composite mais cela implique forcément plus d'évaluation de la fonction g que la méthode simple (soit 1 seule fois).

3.3 Méthode itérative du point fixe (3 points)

En analyse numérique, une méthode itérative est un procédé algorithmique. Pour résoudre un problème donné, après le choix d'une valeur initiale considérée comme une première ébauche de solution, la méthode procède par itérations au cours desquelles elle détermine une succession de solutions approximatives raffinées qui se rapprochent graduellement de la solution cherchée.

Par exemple, on cherche à résoudre numériquement l'équation $\cos(x^*) = x^*$ sur [0,1].

Question 8 *Montrer en utilisant le théorème du point fixe que l'itération $x_n = \cos(x_{n-1})$ avec $x_0 \in [0, 1]$ converge vers la solution unique d'une telle équation. On rappelle que $\forall x \in [0, 1], \sin(x) \leq \sin(1) < 1$.*

On suppose que l'équation $\cos(x^*) = x^*$ admet une unique solution sur $[0, 1]$ (implicitement supposé dans l'énoncé, une démonstration simple existe, pour cela regarder dans la correction du TP 1). Pour tout n, $x_n \in [0, 1]$, donc pour montrer que la suite converge bien vers x^*, il suffit de montrer que la suite (x_n) converge.

On pose $g(x) = \cos(x)$ est une fonction continue et dérivable sur $[0, 1]$ et $g'(x) = \sin(x)$ est une fonction continue sur $[0, 1]$. g' admet donc un maximum sur $[0, 1]$ et comme elle est strictement croissante sur $[0, 1]$

$$\text{Pour tout } x \text{ de } [0, 1] \qquad 0 \leq g'(x) \leq g'(1) < 1$$

donc par passage au maximum

$$\max_{x\in[0,1]} |g'(x)| \leq g'(1) < 1$$

Le maximum de $|g'|$ sur $[0, 1]$ est strictement inférieur à 1, donc d'après le cours (théorème du point fixe), la suite (x_n) converge vers x^* unique solution de l'équation $x^* = g(x^*)$ sur $[0, 1]$.

Question 9 *Compléter l'algorithme qui permet de calculer x_n :*

```
function [ res ] = pointfixe( x0,n )
res=x0 ;
```

```
.............
.............
.............
endfunction
```

```
function [ res ] = pointfixe(x0,n)
res = x0 ;
for i = 1:n
   res = cos(y) ;
end
endfunction
```

Question 10 *On note $e_n = |x_n - x^*|$ l'erreur de la méthode après n itérations. La figure présente l'évolution de e_{n+1}/e_n en fonction de n. Pourquoi est-ce un graphique intéressant ? Pouvait-t-on prédire ce comportement ?*

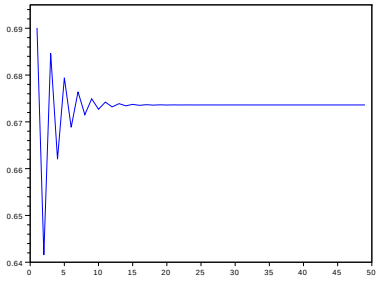

FIG. 1 – **Vitesse de convergence de la méthode du point fixe appliquée à la fonction** $cos(x)$

Ce graphique est intéressant, il permet de montrer que e_{n+1}/e_n admet une limite réelle strictement positive lorsque n tend vers l'infini donc que la méthode utilisée est d'ordre 1. Ce comportement était prévisible. En effet puisque $g'(x^*) \neq 0$, on démontre de manière théorique que la méthode est d'ordre 1.