Corrigé 8

Exercice 1

Notons $x_{j+1/2} = (j+1/2)h$, $j = 0, 1, ..., N$. L'approximation de la dérivée première d'une fonction v au point x_i par une formule de différences finies centrées est :

$$v'(x_i) \simeq \frac{v(x_{i+1/2}) - v(x_{i-1/2})}{h}.$$

Par conséquent, en posant $v(x) = (2+\sin(x))u'(x)$, on a

$$\begin{aligned}\frac{d}{dx}\big((2+\sin(x_i))u'(x_i)\big) &\simeq \frac{(2+\sin(x_{i+1/2}))u'(x_{i+1/2}) - (2+\sin(x_{i-1/2}))u'(x_{i-1/2})}{h} \\ &\simeq \frac{(2+\sin(x_{i+1/2}))(u(x_{i+1}) - u(x_i)) - (2+\sin(x_{i-1/2}))(u(x_i) - u(x_{i-1}))}{h^2}.\end{aligned}$$

Un schéma basé sur des différences centrées pour le problème donné revient à chercher les u_i, $i = 1, ..., N$ tels que

$$\begin{cases} -\dfrac{1}{h^2}\big((2+\sin(x_{i+1/2}))(u_{i+1}-u_i) - (2+\sin(x_{i-1/2}))(u_i - u_{i-1})\big) = f(x_i), \quad i = 1, ..., N, \\ u_0 = u_{N+1} = 0. \end{cases}$$

Ce problème peut s'écrire sous la forme du système linéaire de taille N donné par :

$$\frac{1}{h^2}\begin{pmatrix} 4+\sin(x_{1/2})+\sin(x_{3/2}) & -2-\sin(x_{3/2}) & & \\ -2-\sin(x_{3/2}) & \ddots & & \\ & \ddots & & -2-\sin(x_{N-1/2}) \\ & & -2-\sin(x_{N-1/2}) & 4+\sin(x_{N-1/2})+\sin(x_{N+1/2}) \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{pmatrix}.$$

Remarque : Une mauvaise méthode consiste à écrire l'équation sous la forme $((2+\sin(x))u')' = (2+\sin(x))u'' + \cos(x)u'$, puis appliquer la méthode des différences finies. En effet, en procédant de cette manière on obtient une matrice qui n'a pas la propriété de symétrie de la matrice ci-dessus.

Exercice 2

En utilisant la formule de différences finies centrées vue au cours pour l'approximation de $u''(x)$ et $u'(x)$, le schéma peut s'écrire :

$$\begin{cases} \dfrac{-u_{i-1}+2u_i-u_{i+1}}{h^2} = f(x_i), \quad i=1,...,N+1, \\ u_0 = 0, \\ u_{N+1} + \dfrac{u_{N+2}-u_N}{2h} = 5. \end{cases}$$

Remarque : Ce problème comporte $N+1$ inconnues, $u_1, \ldots, u_{N+1}$.

En éliminant l'inconnue u_{N+2}, ce schéma peut s'écrire sous la forme d'un système linéaire de taille $N+1$ donné par :

$$\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & \ddots & \ddots & & \\ & & \ddots & \ddots & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1+h \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_N \\ u_{N+1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ \vdots \\ f(x_N) \\ \frac{1}{2}f(x_{N+1}) + \frac{5h}{h^2} \end{pmatrix}.$$

Remarque : La dernière équation du système a été divisée par 2 afin d'obtenir une matrice symétrique.

Exercice 3

3.a) On utilise la formule de différences finies centrées pour approcher $u''(x)$:

$$u''(x) \simeq \frac{\delta_h^2 u(x)}{h^2} = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2}.$$

Considérons le N-vecteur $\vec{u} = (u_j)_{j=1}^N$, où les $u_j \simeq u(x_j)$, $j = 1, 2, ..., N$. Posons $u_0 = 0$ et $u_{N+1} = 0$. Le système d'équations pour discrétiser le problème est le suivant:

$$\begin{cases} \dfrac{-u_{j-1}+2u_j-u_{j+1}}{h^2} + (1+x_j)u_j + x_j e^{u_j} = f(x_j), \quad j=1,2,...,N, \\ u_0 = u_{N+1} = 0. \end{cases}$$

Ce problème est donc équivalent à chercher le vecteur $\vec{u} \in \mathbb{R}^N$ tel que $\vec{F}(\vec{u}) = \vec{0}$, où la fonction $\vec{F}$ est définie par :

$$\vec{F} : \vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} \in \mathbb{R}^N \mapsto \vec{F}(\vec{v}) = \begin{pmatrix} F_1(v_1, \ldots, v_N) \\ F_2(v_1, \ldots, v_N) \\ \vdots \\ F_N(v_1, \ldots, v_N) \end{pmatrix} \in \mathbb{R}^N,$$

avec

$$\begin{cases} F_1(\vec{v}) = \dfrac{2v_1 - v_2}{h^2} + (1+x_1)v_1 + x_1 e^{v_1} - f(x_1), \\ F_j(\vec{v}) = \dfrac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} + (1+x_j)v_j + x_j e^{v_j} - f(x_j), \quad j = 2, ..., N-1, \\ F_N(\vec{v}) = \dfrac{-v_{N-1} + 2v_N}{h^2} + (1+x_N)e^{v_N} - f(x_N). \end{cases}$$

3.b) La matrice jacobienne est définie par :

$$D\vec{F}(\vec{v}) = \begin{pmatrix} a_1 & c_1 & & & & \\ c_1 & a_2 & c_2 & & & \\ & c_2 & a_3 & \ddots & 0 & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & 0 & & \ddots & a_{N-1} & c_{N-1} \\ & & & & c_{N-1} & a_N \end{pmatrix},$$

avec

$$\begin{cases} a_j = \dfrac{2}{h^2} + (1+x_j) + x_j e^{v_j}, & j = 1, 2, ..., N, \\ c_j = -\dfrac{1}{h^2}, & j = 1, 2, ..., N-1. \end{cases}$$

Pour tout $\vec{v} \in \mathbb{R}^N$, la matrice $D\vec{F}(\vec{v})$ est donc bien symétrique. Elle est définie positive car $\forall \vec{z} \in \mathbb{R}^N$:

$$\vec{z}^T D\vec{F}(\vec{v})\vec{z} = \frac{1}{h^2}\left(z_1^2 + z_N^2 + \sum_{j=1}^{N-1}(z_j - z_{j+1})^2\right) + \sum_{j=1}^{N}\left((1+x_j) + x_j e^{v_j}\right) z_j^2 \geqslant 0,$$

et $\vec{z}^T D\vec{F}(\vec{v})\vec{z} = 0$ si et seulement si $\vec{z} = 0$.

Remarque: On peut également noter que c'est la somme d'une matrice symétrique définie positive et d'une matrice diagonale à coefficients positifs.

3.c) Comme la matrice jacobienne est symétrique définie positive, la décomposition de Cholesky $A = LL^T$ est toute indiquée pour résoudre le système linéaire à chaque pas. On complète l'algorithme de la méthode de Newton donné dans le fichier `diffinies.m` de la manière suivante :

```
function [u]=diffinies(N, max_iter = 10)
    % parametres
    %
    % N         : nombre d'inconnues du systeme non lineaire
    % h         : pas d'espace
    % u         : N-vecteur, approximation de la solution du probleme aux limites
    % a         : N-vecteur, diagonale de la matrice jacobienne A,
    %             puis diagonale de L tq A=LL^T
    % c         : (N-1)-vecteur, sous-diagonale de la matrice jacobienne A,
    %             puis sous-diagonale de L tq A=LL^T
    % b         : N-vecteur, second membre du systeme lineaire A y = b,
```

```
%               puis solution du systeme lineaire A y = b
%
h = 1 / (N + 1);
coeff = (N + 1)^2;
%
% initialisation : u^0
%
for i = 1 : N
  u(i) = 1.;
end

%
% algorithme de Newton : DF(u^n) ( u^n-u^{n+1} ) = F(u^n), c'est-a-dire
%                           A y = b, puis u^{n+1} = u^n - y
%
for n = 1 : max_iter
%
% remplir la matrice jacobienne A = DF(u^n) et le second membre b = F(u^n)
%
    for i = 1 : N
      a(i) = 2 * coeff + (1 + i * h) + i * h * exp(u(i));
    end
    for i=1:N-1
      c(i) = -1 * coeff;
    end
    b(1) = (2 * u(1) - u(2)) * coeff + (1 + h) * u(1) + h * exp(u(1)) - f(h);
    for i = 2 : N - 1
      b(i) = (-u(i-1) + 2 * u(i) - u(i + 1)) * coeff + (1 + i * h) * u(i) + i * h * exp(u(i)) - f(i * h);
    end
    b(N) = (-u(N-1) + 2 * u(N)) * coeff + (1 + N * h) * u(N) + N * h * exp(u(N)) - f(N * h);
%
% decomposition de cholesky A = LL^T
%
a(1) = sqrt(a(1));
for i = 1 : N - 1
  c(i) = c(i) / a(i);
  a(i+1) = sqrt(a(i+1) - c(i) * c(i));
end
%
% resolution du systeme lineaire Lz = b */
%
b(1) = b(1) / a(1);
for i=1:N-1
    b(i+1) = (b(i+1) - c(i) * b(i)) / a(i+1);
end
%
% resolution du systeme lineaire L^T y = z */
%
b(N) = b(N) / a(N);
for i = N - 1 : -1 : 1
    b(i) = (b(i) - c(i) * b(i+1)) / a(i);
end
%
% poser u^{n+1} = u^n - y, imprimer la norme euclidienne de y et l'erreur max.
%
```

```
    norm2 = 0;
    err = 0;
    for i=1:N
        u(i) = u(i) - b(i);
        norm2 =norm2 + b(i) * b(i);
        erri = abs(u(i) - uex(i * h));
        if (erri > err)
            err = erri;
        end
    end

  fprintf(' iteration %d norm2 %e err %e \n',n,norm2,err)
end
end
```

3.d) On constate, pour $N = 15$ par exemple, que la méthode de Newton converge après 5 itérations (`norm2` $= 1.459428e - 32$); l'erreur $\max_{1 \leq i \leq N} |u(x_i) - u_i|$ vaut alors $5.551115e - 17$. On a bien $u(x_i) = u_i$ à 16 décimales près.

3.e) Le tableau suivant représente l'erreur $\max_{1 \leq i \leq N} |u(x_i) - u_i|$ pour plusieurs valeurs de h:

$N+1$	h	Erreur
4	0.25	2.200004e-01
8	0.125	5.055203e-02
16	0.0625	1.237587e-02

On note que l'erreur est approximativement divisée par 4 à chaque fois que h est divisé par 2.