Corrigé 11

Exercice 1

Nous commençons la résolution numérique du problème $(\mathcal{P})$ en discrétisant par rapport à la variable x. Considérons les approximations $u_i(t)$ de $u(x_i,t)$. Le problème semi-discrétisé en espace consiste donc à trouver les N fonctions $u_i : \mathbb{R}_+ \to \mathbb{R}$, $i = 1, ..., N$ telles que

$$\dot{u}_i(t) - \frac{u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)}{h^2} = u_i(t)^3, \qquad i = 1, ..., N,$$

avec les conditions initiales $u_i(0) = w(x_i)$, $i = 1, ..., N$ et les conditions de bord $u_0(t) = u_{N+1}(t) = 0$, $\forall\ t \geqslant 0$. Ce schéma peut s'écrire sous forme matricielle, en définissant les N-vecteurs $\vec{u}(t) = \big(u_i(t)\big)_{i=1}^N$, $\vec{w} = \big(w(x_i)\big)_{i=1}^N$ et $\big(\vec{u}(t)\big)^3 = \big(u_i(t)^3\big)_{i=1}^N$, ainsi que la $N \times N$-matrice suivante

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & 0 & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \ddots & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}.$$

Il s'agit alors de trouver la fonction $\vec{u} : t \in \mathbb{R}_+ \mapsto \vec{u}(t) \in \mathbb{R}^N$ telle que

$$\begin{cases} \dot{\vec{u}}(t) &= -A\vec{u}(t) + \big(\vec{u}(t)\big)^3, \qquad \forall\ t \geqslant 0, \\ \vec{u}^0 &= \vec{w}. \end{cases}$$

1.a) Le problème discrétisé en temps par la méthode d'Euler rétrograde revient à chercher les N-vecteurs $\vec{u}^n = (u_i^n)_{i=1}^N$, $n \geqslant 0$ tels que

$$\begin{cases} \dfrac{\vec{u}^{n+1} - \vec{u}^n}{\tau} &= -A\vec{u}^{n+1} + (\vec{u}^{n+1})^3, \qquad n \geqslant 0, \\ \vec{u}^0 &= \vec{w}. \end{cases}$$

L'équation qui permet d'obtenir $\vec{u}^{n+1}$ à partir de $\vec{u}^n$ s'écrit

$$(I + \tau A)\vec{u}^{n+1} - \tau(\vec{u}^{n+1})^3 = \vec{u}^n, \qquad n = 0, 1, 2, ...$$

Ce schéma est donc bien implicite et non linéaire, car pour calculer $\vec{u}^{n+1}$ à partir de $\vec{u}^n$ il faut résoudre un système de N équations non linéaires à N inconnues.

1.b) Soit $\vec{F} : \vec{v} \in \mathbb{R}^N \mapsto \vec{F}(\vec{v}) \in \mathbb{R}^N$ la fonction définie par $\vec{F}(\vec{v}) = (I + \tau A)\vec{v} - \tau(\vec{v})^3 - \vec{u}^n$. A chaque pas de temps, le problème issu de la méthode d'Euler rétrograde revient à trouver $\vec{u}^{n+1} \in \mathbb{R}^N$ tel que $\vec{F}(\vec{u}^{n+1}) = 0$. La méthode de Newton pour approcher $\vec{u}^{n+1}$ s'écrit

$$\begin{cases} D\vec{F}(\vec{v}^k)(\vec{v}^k - \vec{v}^{k+1}) = \vec{F}(\vec{v}^k), \qquad k \geqslant 0, \\ \vec{v}^0 \quad \text{donné}, \end{cases}$$

où $D\vec{F}(\vec{v})$ est la matrice jacobienne de $\vec{F}$ au point $\vec{v}$:

$$D\vec{F}(\vec{v}) = I + \tau A - 3\tau B(\vec{v}) \qquad \text{avec} \qquad B(\vec{v}) = \begin{pmatrix} v_1^2 & & & \\ & v_2^2 & 0 & \\ & & \ddots & \\ & 0 & & v_N^2 \end{pmatrix}.$$

Cependant, on se contente de faire un unique pas de la méthode de Newton pour approcher $\vec{u}^{n+1}$ et c'est le résultat de ce premier pas qui sera alors pris comme solution $\vec{u}^{n+1}$. En choisissant $\vec{v}^0 = \vec{u}^n$ le schéma issu du premier pas de la méthode de Newton s'écrit

$$D\vec{F}(\vec{u}^n)(\vec{u}^n - \vec{u}^{n+1}) = \vec{F}(\vec{u}^n), \qquad n \geqslant 0,$$

c'est-à-dire, en considérant les définitions de $\vec{F}(\vec{v})$ et de sa jacobienne,

$$\big(I + \tau A - 3\tau B(\vec{u}^n)\big)(\vec{u}^n - \vec{u}^{n+1}) = \tau A\vec{u}^n - \tau(\vec{u}^n)^3, \qquad n \geqslant 0.$$

On constate que ce schéma est implicite et linéaire, car il suffit de résoudre un système linéaire pour calculer $\vec{u}^{n+1}$ à partir de $\vec{u}^n$.

1.c) Le problème discrétisé en temps par la méthode de Crank-Nicholson revient à chercher les N-vecteurs $\vec{u}^n = (u_i^n)_{i=1}^N$, $n \geqslant 0$ tels que

$$\begin{cases} \dfrac{\vec{u}^{n+1} - \vec{u}^n}{\tau} = \dfrac{\big(-A\vec{u}^{n+1} + (\vec{u}^{n+1})^3\big) + \big(-A\vec{u}^n + (\vec{u}^n)^3\big)}{2}, \qquad n \geqslant 0, \\ \vec{u}^0 = \vec{w}, \end{cases}$$

ou, de façon équivalente

$$\begin{cases} \left(I + \dfrac{\tau}{2}A\right)\vec{u}^{n+1} - \dfrac{\tau}{2}(\vec{u}^{n+1})^3 = \left(I - \dfrac{\tau}{2}A\right)\vec{u}^n + \dfrac{\tau}{2}(\vec{u}^n)^3, \qquad n \geqslant 0, \\ \vec{u}^0 = \vec{w}. \end{cases}$$

Ce schéma est donc implicite et non linéaire, car pour calculer $\vec{u}^{n+1}$ à partir de $\vec{u}^n$, il faut résoudre un système non linéaire de N équations à N inconnues.

Exercice 2

2.a) On pose $u_i^0 = x_i(1 - x_i)$, $i = 0, ..., N+1$. Pour tout $n \geqslant 0$, étant donné u_i^n, $i = 0, ..., N+1$, le problème discrétisé en temps et en espace par la méthode d'Euler progressive revient à chercher les u_i^{n+1}, $i = 0, ..., N+1$ tels que

$$\begin{cases} \dfrac{u_i^{n+1} - u_i^n}{\tau} - \dfrac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{h^2} + (1 + x_i)u_i^n = 0, \qquad i = 1, ..., N, \\ u_0^{n+1} = u_{N+1}^{n+1} = 0. \end{cases}$$

La relation ci-dessus nous permet également d'écrire :

$$u_i^{n+1} = \left(1 - \frac{2\tau}{h^2} - (1 + x_i)\tau\right) u_i^n + \frac{\tau}{h^2} u_{i-1}^n + \frac{\tau}{h^2} u_{i+1}^n, \quad i = 1, \dots, N. \qquad (1)$$

Ce schéma est donc bien explicite, car il permet d'exprimer explicitement u_i^{n+1}, $i = 1, \dots, N$ à partir de u_i^n, $i = 0, \dots, N+1$.

2.b) On a $\left(1 - \frac{2\tau}{h^2} - (1 + x_i)\tau\right) \geqslant 1 - \frac{2\tau}{h^2} - 2\tau \geqslant 0$ par hypothèse. Par conséquent, si $u_i^n \geqslant 0$, $i = 0, \dots N+1$, alors $u_i^{n+1} \geqslant 0$ pour $i = 1, \dots, N$ grâce à la relation (1) ci-dessus. Comme de plus, $u_0^{n+1} = u_{N+1}^{n+1} = 0$, les u_i^{n+1} sont positifs pour tout $i = 0, \dots, N+1$.

Exercice 3

3.a) On pose $u_i^0 = \sin(2\pi x_i)$, $i = 0, ..., N+1$. Pour tout $n \geqslant 0$, étant donné u_i^n, $i = 0, ..., N+1$, le problème discrétisé revient à chercher les u_i^{n+1}, $i = 0, ..., N+1$ tels que

$$(\mathcal{R}_1) \begin{cases} \dfrac{u_i^{n+1} - u_i^n}{\tau} - \dfrac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{h^2} + \alpha(x_i)^3 u_i^n = f(x_i, t_n), & i = 1, ..., N, \\ u_0^{n+1} = u_{N+1}^{n+1} = 0. \end{cases}$$

Nous pouvons mettre la relation ci-dessus sous la forme:

$$u_i^{n+1} = \left(1 - \frac{2\tau}{h^2} - \alpha\tau(x_i)^3\right) u_i^n + \frac{\tau}{h^2} u_{i-1}^n + \frac{\tau}{h^2} u_{i+1}^n + \tau f(x_i, t_n), \qquad i = 1, ..., N.$$

Ce schéma est donc explicite, car il permet d'exprimer u_i^{n+1}, $i = 1, ..., N$ à partir de u_i^n, $i = 0, ..., N+1$.

3.b) Le fichier `paraprog.m` est complété de la manière suivante :

```
%
% schema d'Euler progressif
%
for n = 1 : M
  unew(1) = (1. - 2 * tau / h / h - alpha * tau * h^3) * uold(1) + tau
     / h / h * uold(2) + tau * f(h,t);
  for i = 2 : N - 1
    unew(i) = (1. - 2 * tau / h / h - alpha * tau * (i * h)^3) * uold(
   i) + tau / h / h * (uold(i - 1) + uold(i + 1)) + tau * f(i * h,t) ;
  end
  unew(N) = (1. - 2 * tau / h / h - alpha * tau * (N * h)^3) * uold(N)
     + tau / h / h * uold(N - 1) + tau * f(N * h,t) ;
  t = t + tau;
end
```

3.c) Nous remarquons que le schéma est stable pour $N = 19$ et $\tau = 0.00125$, tandis qu'il est instable pour $N = 19$ et $\tau = 0.0013$. La condition $\tau \leqslant h^2/2$ est violée dans ce deuxième cas.

3.d) On a les résultats suivants au temps $t = 0.8$:

$N+1$	h	τ	M	Erreur
10	0.1	0.005	160	1.345654e-07
20	0.05	0.00125	640	3.526849e-08
40	0.025	0.0003125	2560	8.915923e-09
80	0.0125	0.000078125	10240	2.235105e-09

On note que l'erreur est approximativement divisée par quatre lorsque h est divisé par deux et τ est divisé par quatre. Le schéma est donc bien d'ordre $h^2 + \tau$.

3.e) On pose $u_i^0 = \sin(2\pi x_i)$, $i = 0, ..., N+1$. Pour tout $n \geqslant 0$, étant donné u_i^n, $i = 0, ..., N+1$, le problème discrétisé revient à chercher les u_i^{n+1}, $i = 0, ..., N+1$ tels que

$$(\mathcal{R}_2)\left\{\begin{array}{l} \dfrac{u_i^{n+1} - u_i^n}{\tau} - \dfrac{u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}}{h^2} + \alpha(x_i)^3 u_i^{n+1} = f(x_i, t_{n+1}), \qquad i = 1, ..., N, \\ u_0^{n+1} = u_{N+1}^{n+1} = 0. \end{array}\right.$$

On peut écrire ce schéma sous la forme $A\vec{u}^{n+1} = \vec{u}^n + \tau\vec{f}(t_{n+1})$ avec:

$$A = \begin{pmatrix} a_1 & -\dfrac{\tau}{h^2} & & \\ -\dfrac{\tau}{h^2} & \ddots & \ddots & \\ & \ddots & \ddots & -\dfrac{\tau}{h^2} \\ & & -\dfrac{\tau}{h^2} & a_N \end{pmatrix},$$

où $a_i = 1 + \dfrac{2\tau}{h^2} + \alpha\tau(x_i)^3$.

3.f) Le fichier `pararetro.m` est complété de la manière suivante :

```
function [err, u, u_ex] = pararetro(N, M, tau)
    %
    %  Schema d'Euler retrograde pour un probleme parabolique
    unidimensionnel.
    %  A chaque pas de temps, il s'agit de resoudre le systeme
    lineaire :
    %             A u^n+1 = u^n + tau f^{n+1}
    %
    %  parametres
    %
    %  N         : nombre d'inconnues du systeme lineaire
    %  h         : pas d'espace
    %  M         : nombre de pas de temps
    %  tau       : pas de temps
    %  t         : temps courant
    %  u         : N-vecteur, a chaque pas de temps,
```

```
  %              u est second membre du systeme lineaire,
  %              puis solution du systeme lineaire
  %  a         : N-vecteur, diagonale de la matrice A,
  %              puis diagonale de L tq  A=LL^T
  %  c         : (N-1)-vecteur, sur-diagonale et sous-diagonale
  %              de la matrice A, puis sous-diagonale de L tq  A=LL^T
  %
  h = 1/(N + 1);
  t = 0;
  alpha = 1.;

  %
  % condition initiale
  %
  for i = 1 : N
    u(i) = w(i * h);
  end

  %
  % remplissage de la matrice  A
  %
  for i = 1 : N
    a(i) = 1. + (2 * tau) / h / h + alpha * tau * (i * h)^3;
  end
  for i = 1 : N-1
    c(i) = (-tau) / h / h;
  end

  %
  % decomposition LL^T 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

  %
  % schema d'Euler retrograde: A u^n+1 = u^n + tau f^{n+1}
  %
  for n = 1 : M
    t = t + tau;

    %
    % second membre du systeme lineaire
    %
    for i = 1 : N
      u(i) = u(i) + tau * f(i * h, t);
    end

    %
    % resolution du systeme lineaire Ly = u^n+ tau f^{n+1}
    %
    u(1) = u(1) / a(1);
    for i = 1 : N - 1
        u(i + 1) = (u(i + 1) - c(i) * u(i)) / a(i + 1);
    end

  %
  % resolution du systeme lineaire L^T u^n+1 = y
  %
```

```
      u(N) = u(N) / a(N);
      for i = N - 1 : -1 : 1
        u(i) = (u(i) - c(i) * u(i+1)) / a(i);
      end
    end

    err = 0;
    for i = 1 : N
      x(i) = i * h;
      u_ex(i) = uex(i * h, t);
      erri = abs(u(i) - uex(i * h, t));
      if (erri > err)
        err = erri;
      end
    end
    fprintf(' erreur maximum au temps final  %e \n', err)
end
```

3.g) Nous remarquons que le schéma est stable quel que soit le choix des paramètres τ et h.

3.h) On a les résultats suivants au temps $t = 0.8$:

$N+1$	h	τ	M	Erreur
10	0.1	0.02	40	1.590178e-06
20	0.05	0.005	160	2.702037e-07
40	0.025	0.00125	640	6.044994e-08
80	0.0125	0.0003125	2560	1.468320e-08

On note que l'erreur est approximativement divisée par quatre lorsque h est divisé par deux et τ est divisé par quatre. Le schéma est donc bien d'ordre $h^2 + \tau$.