Université Mohamed Khider Biskra
Département de Mathématiques
Master 2

2021/2022
Simulation et M.N
T.P

Devoir
(Corrigé type)

Exercice 1.

La méthode de différence finies consiste à approcher la solution exacte $y(x)$ sur $[0,1]$ aux $m+1$ points $0 = x_0 < x_1 < \ldots < x_m = 1$.
On prend $x_k = kh$ *avec* $k = 0, \ldots m$ *et* $h = L/m$, h est le pas de discrétisation.

On a les approximations suivantes

$$\frac{dy}{dx}(x) \simeq \frac{y_{k+1} - y_k}{h}, \quad \frac{d^2y}{dx^2}(x) \simeq \frac{y_{k+1} + y_{k-1} - 2y_k}{h^2}$$

On a $y_0 = 0$ et $y_m = 1$, et pour tout $k \neq 0, m$, le problème s'écrit

$$\frac{-y_{k+1} - y_{k-1} + 2y_k}{h^2} + p(x_k)\frac{y_{k+1} - y_k}{h} = f(x_k)$$

Nous obtenons alors un système de $m-1$ équations linéaires avec $m-1$ inconnus :

$$-y_{k-1} + (2 - h\ p(x_k))y_k + (h\ p(x_k) - 1)y_{k+1} = h^2 f(x_k)$$

$$\begin{cases} (2 - h\ p(x_1))y_1 + (h\ p(x_1) - 1)y_2 = h^2 f(x_1) + y_0 \\ -y_1 + (2 - h\ p(x_2))y_2 + (h\ p(x_2) - 1)y_3 = h^2 f(x_2) \\ \\ \ldots\ldots \\ \\ -y_{m-2} + (2 - h\ p(x_{m-1}))y_{m-1} = h^2 f(x_{m-1}) - (h\ p(x_{m-1}) - 1)y_m \end{cases}$$

Le système s'écrit sous la forme $A_h Y = F_h$, où $A_h = A_{h1} + hA_{h2}$

$$A_{h1} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0\ldots & 0 \\ -1 & 2 & -1 & 0 & 0\ldots & 0 \\ 0 & -1 & 2 & -1 & 0\ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0.. & 0 & ..0 & -1 & 2 & -1 \\ 0.. & 0 & .. & . & -1 & 2 \end{bmatrix},$$

$$A_{h2} = h \begin{bmatrix} -p(x_1) & p(x_1) & 0 & 0 & 0... & 0 \\ 0 & -p(x_2) & p(x_2) & 0 & 0... & 0 \\ 0 & 0 & -p(x_3) & p(x_3) & 0... & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0.. & 0 & ..0 & 0 & -p(x_{m-2}) & p(x_{m-2}) \\ 0.. & 0 & .. & . & 0 & -p(x_{m-1}) \end{bmatrix}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_{m-1} \end{bmatrix}, F_h = h^2 \begin{bmatrix} f(x_1) \\ f(x_2) \\ . \\ . \\ f(x_{m-1}) \end{bmatrix} + \begin{bmatrix} y_0 \\ 0 \\ . \\ 0 \\ -(h\ p(x_{m-1}) - 1)y_m \end{bmatrix},$$

L'implémentation

```
f=@(x) exp(2.*x.^2);
p=@(x) 2.*x;
%discrétisation
n = 100 ;  h = 1/(n) ; xh = 0 : h: 1 ;
ybegin=0;yend=1; %condition initiales
%Les matrices
A1 = diag(ones(n - 2, 1) ,1)*(-1) ;
A2 = diag(ones(n - 2, 1) ,-1)*(-1) ;
A3 = diag(ones(n-1, 1))*(2) ;
Ah1 = A1 + A2 + A3 ;

px1=p(xh(2:end-1));
px2=p(xh(2:end-2));
Ah2=h*(-diag(px1)+diag(px2,1));
A=Ah1+Ah2;
%Le vecteur fh
fx=f(xh(2:end-1));
fh=h^2*fx+[ybegin  zeros(1,n-3) (h*p(xh(end-1)-1))*yend];
%La résolution de A*yh=fh'
yh = inv(A)*fh';
yhh = [ybegin , yh', yend];
plot(xh, yhh,'ro')
xlabel('x')
ylabel('y')
title('solution numérique(problème de l'ex 1)')
```

Exercice 3.

a) Pour ce problème, on utilise la commande fmincon

```
function [ C,Ceq ] = nonlin( x )
C=[ ];
Ceq=[x(1)^2+x(2)^2-10];
end
```

```
F=@(x) -x(1)+2*x(2)^2;
[x,fva,existflag,output]=fmincon(F,[1,1],[1/2,-1],[1],
[],[],[0,0],[inf,inf],@nonlin)
```

b) On a besoin d'utiliser la commande fminimax

```
f=@(x) [-2*x(1)-3*x(2)+18,x(1)+x(2)+3,-x(1)+x(2)];
[x,fval,maxfval,existflag,output]=fminimax(f,[1;1],[1 1 ],
[4],[2 3 ],[-1],[-inf,-inf],[inf,inf])
```

c) Un problème de minimisation d'une fonction quadratique sous des contraintes linéaires :

```
[x,fva,existflag,output]=quadprog([1 -1 ; -1 2 ],[-2;-6],
[-1 -1 ; 3 1; -1 1],[-1;2;1],[],[],[0,0],[inf,inf],[0;0])
```

Exercice 2.

1. Le schémas explicite. Pour calculer une solution approchée, on se donne une discrétisation en temps et en espace.

- On se donne un ensemble de points $t_j, j = 0, 1, \dots M$ de $]0, 1[$ et $x_i, i = 0, 1, \dots N$ de $]0, 1[$.
- On considère un pas constant $h = \frac{1}{N}$ et $k = \frac{1}{M}$.
- On pose $t_j = jk$ pour $j = 0, \dots M$ et $x_i = ih$ pour $i = 0, \dots N$

On approche la dérivée en temps

$$\frac{\partial y}{\partial t}(x_i, t_j) \simeq \frac{y_i^{j+1} - y_i^j}{k}$$

D'après les conditions données, on utilisera un schéma qui contient les approximations y_i^{j+1}, y_{i+1}^j et y_i^j, donc on aura besoin d'utiliser l'approximation suivante :

$$\frac{\partial y}{\partial x}(x_i, t_j) \simeq \frac{y_{i+1}^j - y_i^j}{h}$$

Le problème s'écrira alors

$$\begin{cases} \dfrac{y_i^{j+1} - y_i^j}{k} + y_i^j \ \dfrac{y_{i+1}^j - y_i^j}{h} = 0 \ , \ i = 0, \ldots, N-1 \ \ j =, 0 \ldots, M-1, \\ y_0^j = 1, y_{n+1}^j = 0, \quad j = 0, \ldots, M \\ y_i^M = cos(\frac{\pi}{2})x_i, \quad i = 0, \ldots, N \end{cases}$$

On obtient

$$y_{i+1}^j = y_i^j - (\lambda(y_i^{j+1} - y_i^j)/y_i^j) \ \ , \ \lambda = h/k$$

2. L'implémentation.

```
%les paramètres
L=1;
T=1;
M=30;
k=T/M;
N=30;
h=L/N;
b=h/k;
%les valeurs initiales
for i=1:(N+1)
    x(i)=(i-1)*h;
        y(i,M+1)=cos((pi/2)*x(i));
    end

%valeurs aux limtes
for j=1:M+1
    y(1,j)=1;
    y(N+1,j)=0;
    t(j)=(j-1)*k;
end

for j=M:-1:2 %boucle du temps
    for i=1:N %boucle de l'espace
        y(i+1,j)=y(i,j)-b*(y(i,j+1)-y(i,j))/y(i,j);
    end
end

mesh(x,t,y')
xlabel('X')
ylabel('T')
title('solution implicite (problème de l'exercice 2)')
```