UNIVERSITE OUAGA 3S

Année 2011-2012

FST/LST2 & ESSIN/LII

Calcul numérique : Pratique de Matlab

Enseignant
Dr SO Ousséni

Etudiant	
Nom :	BAMOUNI
Prénoms :	Kevin Cédric Y.
Faculté :	FST/LST2
N°IP :	11-12/FST/001/LST2

Rapport du projet N°8 de Matlab

Sommaire :

I. Présentation des Problèmes abordés.

Tableau 1 : Enoncé

Projet 8
Résoudre l'équation de la chaleur $\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0,\ (x,t) \in [0;1] \times [0;0.01]$ Conditions aux limites : $u(0,t) = 0$ et $u(1,t) = 10t$ pour tout t. Conditions initiale $u(x,0) = \begin{cases} 0 \text{ si } 0 \le x \le 0.25 \\ (x-0.25) \text{ si } 0.25 \le x \le 0.5 \\ (0.75-x) \text{ si } 0.5 \le x \le 0.75 \\ 0 \text{ si } 0.75 \le x \le 1 \end{cases}$

L'équation de la chaleur ci-dessus est une équation aux dérivées partielles sans second membre donc un problème parabolique.

II. La méthode utilisée pour la résolution.

La solution exacte de l'équation parabolique ci-dessus obtenue par intégration est la suivante :

$$u(x,t) = e^{-t} \sin x$$

La résolution numérique de l'équation de la chaleur soumis à notre étude consistera a approximer la solution exacte $u(x,t) = e^{-t} \sin x$ par une méthode bien définie que nous aurons choisies parmi celles connues pour résoudre (approximer numériquement) les équations paraboliques.

La méthode que nous allons utiliser pour résoudre numériquement l'équation de la chaleur sera la **Méthode des différences finies**.

La méthode des différences finies se présente comme suit :

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

on pose $u(x_i, t) = u_i$ avec $i \in \;\; 1, n-1 \quad n \in \mathbb{N}^*$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1} - 2u_i + u_{i-1}}{dx^2} = u_i'(t) \text{ avec } dx^2 = (\Delta x)^2$$

$$\text{pour } i = 1; \qquad u_1'(t) = \frac{u_2 - 2u_1 + u_0}{dx^2}$$

$$\text{pour } i = 2; \qquad u_2'(t) = \frac{u_3 - 2u_2 + u_1}{dx^2}$$

$$\text{pour } i = 3; \qquad u_3'(t) = \frac{u_4 - 2u_3 + u_2}{dx^2}$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\text{pour } i = n-1; \qquad u_{n-1}'(t) = \frac{u_n - 2u_{n-1} + u_{n-2}}{dx^2}$$

on peut alors poser u matriciellement :

$$\begin{pmatrix} u_1 \\ \vdots \\ \vdots \\ \vdots \\ u_{n-1} \end{pmatrix} = \frac{1}{(\Delta x)^2} \begin{pmatrix} -2 & 1 & \dots & \dots & 0 \\ 1 & -2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -2 & 1 \\ 0 & \dots & \dots & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{n-1} \end{pmatrix} + \frac{1}{(\Delta x)^2} \begin{pmatrix} u_0 \\ \vdots \\ \vdots \\ \vdots \\ u_n \end{pmatrix}$$

avec les conditions aux limites : $u_0 = 0$ et $u_n = 10t$

dans notre résolution on pose la matrice M= $\begin{pmatrix} -2 & 1 & \cdots & \cdots & 0 \\ 1 & -2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -2 & 1 \\ 0 & \cdots & \cdots & 1 & -2 \end{pmatrix}$

avec M qui est une matrice trigonale.

et $u = \begin{pmatrix} u_0 \\ \vdots \\ \vdots \\ u_n \end{pmatrix}$

III. Les algorithmes.

Les algorithmes utilisés pour l'approximation de l'équation parabolique sont :

Tableau 2 : Fonction pardef

Fonction pardef

global T NT NX dt dx
T=1;
NX=10;
NT=20;
dt=T/NT;
dx=pi/NX;

Tableau 3 : Fonction matrice

Fonction matrice
function M=matrice global T NT NX dt dx V=[-2 1 zeros(1,NX-3)]; M=toeplitz(V);

Tableau 4 : Fonction solution

Fontion solution

```
function [U,t,x]=solution
global T NT NX dt dx
t=0:dt:0.01;
x=0:dx:1;
 %comprndre et construire U0 la matrice de la condition initiale
%codition initiale
z1=0;
z2=0;
z3=0;
z4=0;
for  z=1:length(x)
if  x(z)>=0  && x(z)<=0.25
   z1=z1+1;
   U1(z1)=0;
else
   if  x(z)>0.25  &&  x(z)<=0.5
      z2=z2+1;
      U2(z2)=x(z)-0.25;%
   else
      if  x(z)>0.5 && x(z)<=0.75
```

```
        z3=z3+1;
        U3(z3)=0.75-x(z);%
      else
        if  x(z)>0.75 && x(z)<=1
          z4=z4+1;
         U4(z4)=0; %
        end
      end
    end
end
end
length(x)
U0=[U1';U2';U3' ;U4'];
UO=U0(2:end-1);
U=UO;
M=matrice;
for i=1:NT
   UO=UO+(1/(dx)^2)*M*UO;
   U=[U UO];
end
U=[zeros(1,NT+1);U;10*t];
```

Tableau 5 : Fonction courbe

Fonction courbe

```
global T NT NX dt dx
[U,t,x]=solution;
[tt,xx]=meshgrid(t,x);
mesh(tt,xx,U),xlabel temps,ylabel espace,zlabel quantité-de-chaleur
legend('Approximation')
```

Tableau 6 : Fonction simulation

Fonction simulation

```
function simulation(NX0,NT0)
global T NT NX dt dx
pardef
NX=NX0;
```

```
NT=NT0;
dt=T/NT;
dx=1/NX;
courbe
```

Tableau 7 : Fonction exacte

Fonction exacte

function exacte(nte,nxe)
t=linspace(0,0.01,nte);
x=linspace(0,1,nxe);
[X,Y]=meshgrid(x,t);
Z=sin(X)*exp(-1*Y);
mesh(X,Y,Z)
xlabel temps,ylabel espace,zlabel quantité-de-chaleur
legend('solution exacte.')

IV. Les figures issues de la résolution.

Nous avons essentiellement **3 figures** issues de la résolution :

La première figure est la courbe de la solution exacte obtenue à l'aide d'une calculatrice scientifique programmable sur ordinateur qui est Microsoft Mathematics, cette courbe nous permet d'avoir un aperçu général de la courbe de la solution exacte $(e^{-it}\sin x)$de l'équation de la chaleur$\left(\frac{\partial u}{\partial t}-\frac{\partial^2 u}{\partial x^2}=0\right)$ soumise à notre étude :

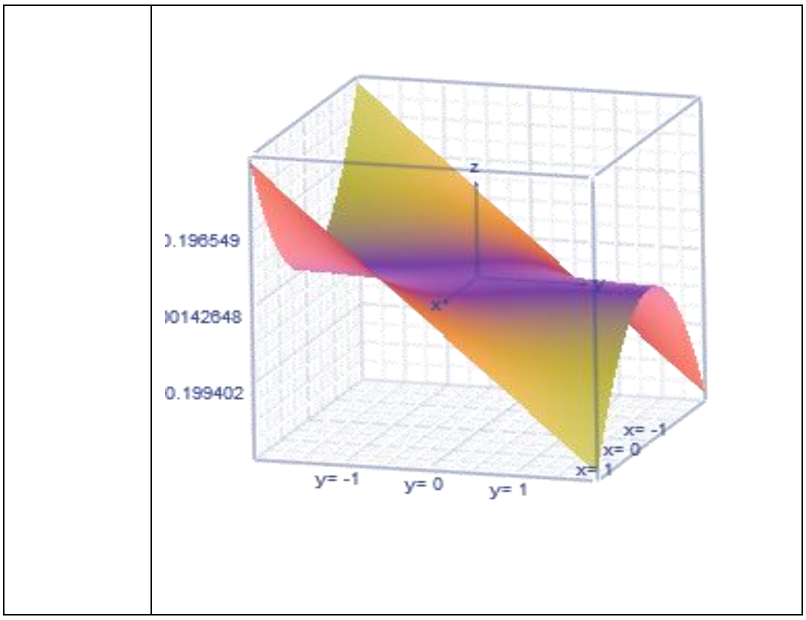

Figure 1 : Solution exacte avec Microsoft Mathematics

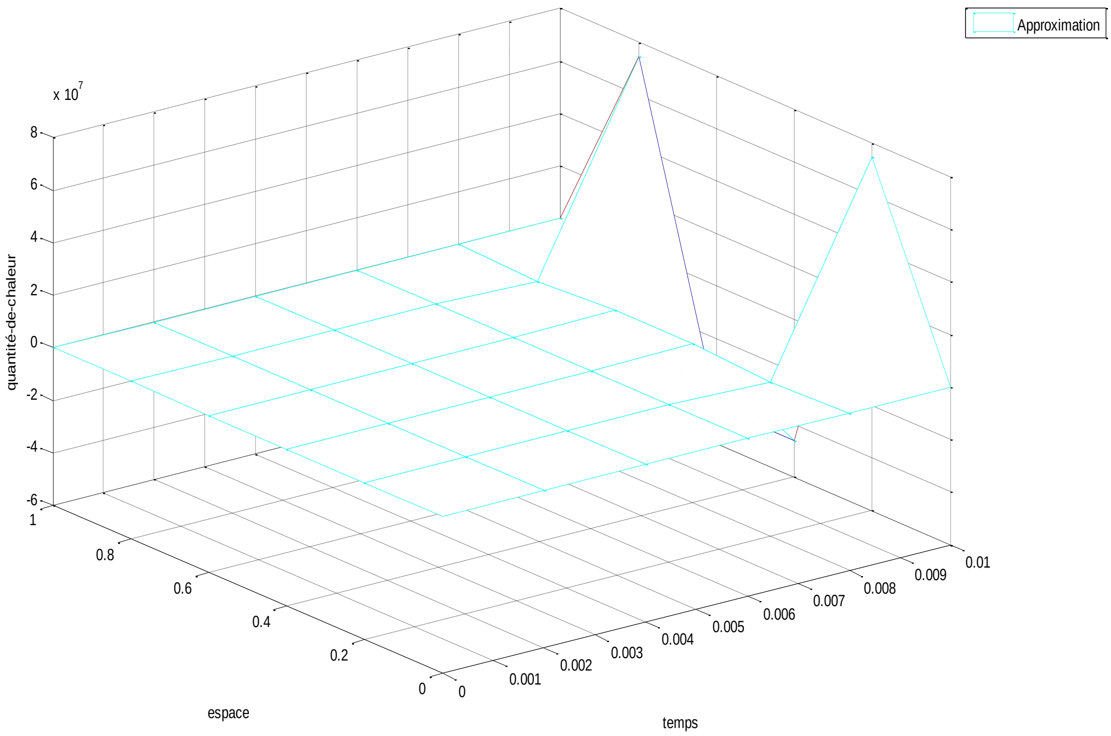

Figure 2: Courbe Obtenue par Simulation Numérique Avec les paramètres (5,5)

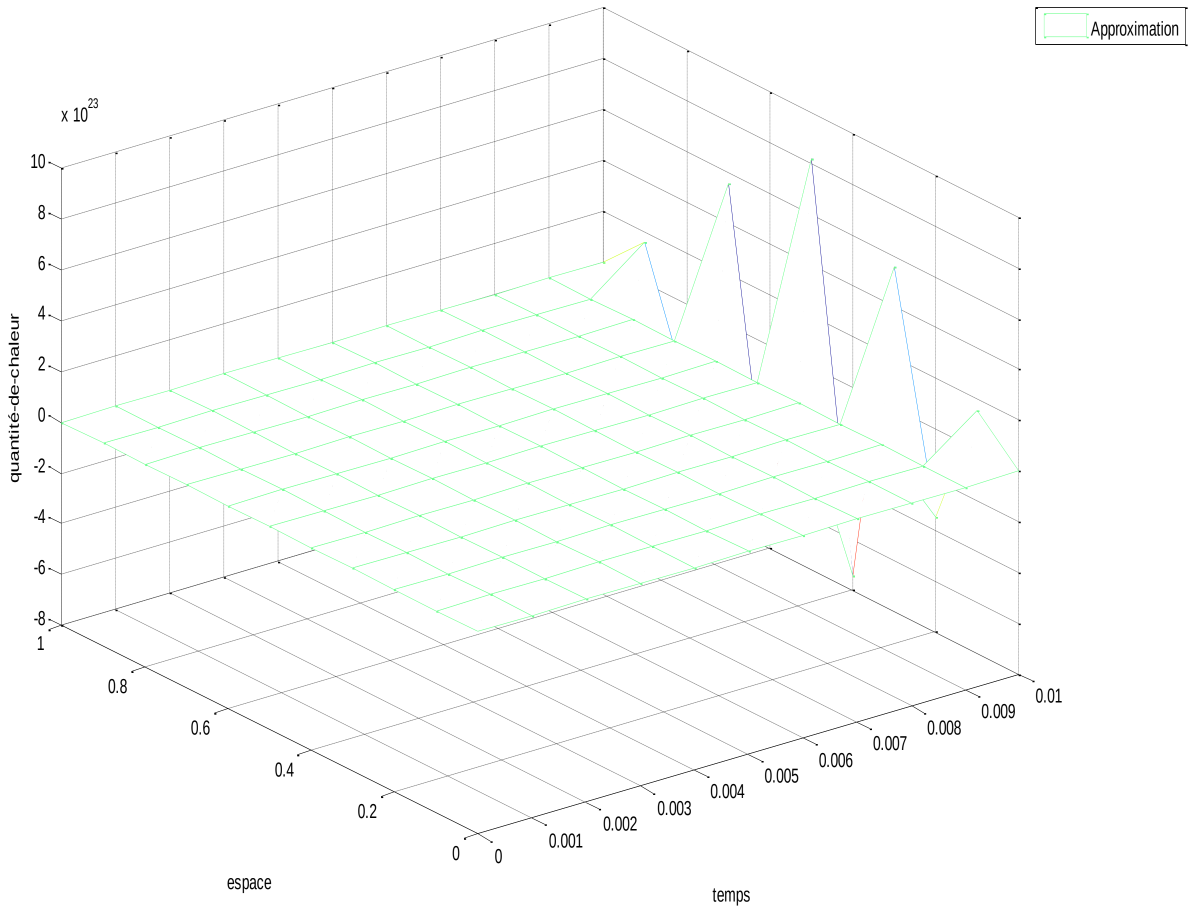

Figure 3 : Courbe Obtenue par Simulation Numérique Avec les paramètres (10,10)

V. Conclusion sur les résultats obtenus.

La figure 1 nous donne un aperçu sur l'allure générale de la solution exacte, ainsi nous pouvons faire une comparaison avec la courbe obtenue par simulation numérique, il fait aussi noter que la simulation se fait sur un espace et un intervalle de temps très réduit :

Avec les paramètres (5,5) on a une courbe qui a moins de nœuds et de détail mais lorsque nous passons a (10,10) nous observons une autre courbe avec plus de nœuds et de détail au niveau de l'origine.

En conclusion plus les intervalles de temps et d'espace sont petits plus la simulation se rapproche de la solution exacte, plus nous avons une courbe détaillée.

VI. Annexe

Tableau 8 : Aide Fonction pardef

La fonction pardef	
Description :	Correspond à l'initialisation des variables utilisées tout au long du programme.

Tableau 9 : Aide Fonction matrice

La fonction matrice	
Description :	Correspond à la création de la matrice trigonale M à l'aide de la fonction toeplitz de Matlab.
Utilisation :	L'exécution de cette fonction ne nessécite aucun paramètre et sa taille est déjà prédéfinition dans son algorithme, ainsi elle est utilisée pour la création de la matrice u , comme dans la méthode explicitée ci-dessus.

Tableau 10 : Aide Fonction solution

La fonction solution	
Description :	correspond à la considération des conditions initiales, à la création de la matrice de u et à
Utilisation :	Cette fonction ne nécessite aucun paramètre d'entré elle utilise la matrice trigonale créée par la fonction matrice pour créer la matrice u complète avec les conditions initiales et les conditions aux limites, qui sera utilisée pour générer le graphique de l'approximation

Tableau 11 : Aide Fonction courbe

La fonction courbe	
Description :	Génère la courbe de la simulation numérique en 3D.
Utilisation :	Cette fonction génère une courbe 3D en utilisant les méthodes classiques de création de graphique 3D sous matlab

Tableau 12 : Aide fonction simulation

La fonction simulation	
Description :	Fonction qui génère la simulation avec des paramètres en entré en utilisant toutes les méthodes précédentes.

Utilisation :	L'instruction pour déclencher la simulation numérique est « simulation(x,y) » avec x qui correspond au nombre de points de l'espace et y le nombre de points du temps.

VII. Table des figures