

GIUDICE Anthony
SIMONET Théo
SONZOGNI Masc

TP 4
Equation de la chaleur

① Afin de modéliser plus simplement les différents schémas, on peut écrire l'équation suivante :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = C(1-\theta) \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + C\theta \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}$$

où on retrouve, si $\theta=0$, le schéma d'Euler explicite, si $\theta=1$, le schéma d'Euler implicite, et si $\theta=\frac{1}{2}$, le schéma de Crank-Nicholson.

On peut alors écrire :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = C(1-\theta) \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + C\theta \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}$$

$$\begin{aligned} (\Rightarrow) \quad & -\theta C \frac{\Delta t}{\Delta x^2} u_{i-1}^{n+1} + \left(1 + 2\theta C \frac{\Delta t}{\Delta x^2}\right) u_i^{n+1} - \theta C \frac{\Delta t}{\Delta x^2} u_{i+1}^{n+1} \\ & = (1-\theta) C \frac{\Delta t}{\Delta x^2} u_{i-1}^n + \left(1 - 2(1-\theta) C \frac{\Delta t}{\Delta x^2}\right) u_i^n + (1-\theta) C \frac{\Delta t}{\Delta x^2} u_{i+1}^n \end{aligned}$$

Si on note $r = C \frac{\Delta t}{\Delta x^2}$, on a alors :

$$\begin{aligned} & -\theta r u_{i-1}^{n+1} + \left(1 + 2\theta r\right) u_i^{n+1} - \theta r u_{i+1}^{n+1} \\ & = (1-\theta)r u_{i-1}^n + \left(1 - 2(1-\theta)r\right) u_i^n + (1-\theta)r u_{i+1}^n \end{aligned}$$

$$\begin{aligned} \Leftrightarrow & \begin{pmatrix} -\theta_n & 1+2\theta_n & -\theta_n \end{pmatrix} \begin{pmatrix} u_{i-1}^{n+1} \\ u_i^n \\ u_{i+1}^{n+1} \end{pmatrix} \\ & = \begin{pmatrix} (1-\theta)_n & 1-2(1-\theta)_n & (1-\theta)_n \end{pmatrix} \begin{pmatrix} u_{i-1}^n \\ u_i^n \\ u_{i+1}^n \end{pmatrix} \end{aligned}$$

En généralisant cette forme, on obtient

$$M U^{n+1} = N U^n \quad (*)$$

avec $U^{dc} = \begin{pmatrix} u_1^x \\ \vdots \\ u_{Ndc+1}^{dc} \end{pmatrix}$

$$M = \begin{pmatrix} 1+2\theta_n & -\theta_n & 0 & \cdots & 0 \\ -\theta_n & \ddots & \ddots & \ddots & | \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -\theta_n \\ 0 & 0 & \cdots & -\theta_n & 1+2\theta_n \end{pmatrix}$$

$$N = \begin{pmatrix} 1-2(1-\theta)_n & (1-\theta)_n & 0 & \cdots & 0 \\ (1-\theta)_n & \ddots & \ddots & \ddots & | \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & (1-\theta)_n \\ 0 & 0 & \cdots & (1-\theta)_n & 1-2(1-\theta)_n \end{pmatrix}$$

② Les conditions limites de Dirichlet s'écrivent sous la forme :

$$u(0, t) = k_0 \quad \text{et} \quad u(1, t) = k_1$$

Dans notre cas, on a :

$$u(0, t) = u_1^n \quad \text{et} \quad u(1, t) = u_{N_{\mathcal{C}}+1}^n$$

On peut alors réécrire (*) sous la forme

$$M U^{n+1} = N U^n + F^n$$

avec $F^x = \begin{pmatrix} k_0 \\ f_1^x \\ f_2^x \\ \vdots \\ f_{N_{\mathcal{C}}}^x \\ k_1 \end{pmatrix} \Delta t$ où $f_i^n = f(x_i, t_n)$

$$M = \begin{pmatrix} 1 & 0 & & & & & 0 \\ -\alpha_n & 1+2\alpha_n & -\alpha_n & 0 & & & 0 \\ 0 & & & & & & 0 \\ & & & & & & 0 \\ 0 & & & & & & 0 \\ 0 & & & & & & 1 \end{pmatrix}$$

$N =$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ (1-\theta)\pi & 1-2(1-\theta)\pi & (1-\theta)\pi & 0 \\ 0 & 0 & (1-\theta)\pi & 1-2(1-\theta)\pi \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

③ Les conditions limites de Neumann s'écrivent sous la forme

$$\frac{\partial u}{\partial x}(0, t) = k_0 \quad \text{et} \quad \frac{\partial u}{\partial x}(1, t) = k_1$$

Si on utilise la condition miroir pour approcher ces dérivées, on obtient

$$\frac{\partial u}{\partial x}(0, t) = \frac{u_2^n - u_1^n}{\Delta x} \quad \text{et} \quad \frac{\partial u}{\partial x}(1, t) = \frac{u_{N_x+1}^n - u_{N_x}^n}{\Delta x}$$

Ainsi, on a

$$\frac{u_2^n - u_1^n}{\Delta x} = k_0 \quad (\Rightarrow) \quad \left(-\frac{1}{\Delta x}\right)u_1^n + \left(\frac{1}{\Delta x}\right)u_2^n = k_0$$

et

$$\frac{u_{N_x+1}^n - u_{N_x}^n}{\Delta x} = k_1 \quad (\Rightarrow) \quad \left(-\frac{1}{\Delta x}\right)u_{N_x}^n + \left(\frac{1}{\Delta x}\right)u_{N_x+1}^n = k_1$$

On peut alors réécrire (*) sous la forme

$$M U^{n+1} = N U^n + F^n$$

avec

$$M = \begin{pmatrix} -\frac{1}{\Delta x_c} & \frac{1}{\Delta x_c} & 0 & & 0 \\ -\theta_n & 1+2\theta_n & -\theta_n & 0 & \\ 0 & & & & 0 \\ 0 & & & & 0 \\ 0 & & & & -\theta_n \\ 0 & & & & 1+2\theta_n \\ 0 & & & & -\theta_n \\ 0 & & & & \frac{1}{\Delta x_c} \\ 0 & & & & \frac{1}{\Delta x_c} \end{pmatrix}$$

A graph plotted on grid paper showing a function with two distinct points of inflection. The first point of inflection is a local maximum at approximately (-2, 3). The second point of inflection is a local minimum at approximately (1, -1).

The diagram illustrates a triangular mesh with 7 nodes arranged in a hexagonal pattern. The nodes are labeled with values corresponding to the entries in the matrix:

- Top row: $-\frac{1}{\Delta x}$, $\frac{1}{\Delta x}$, 0
- Middle row: $(1-\theta)r$, $1-2(1-\theta)r$, $(1-\theta)r$
- Bottom row: 0, 0, 0

The mesh consists of 6 triangles. Vertical lines connect the bottom row nodes to the central node, and diagonal lines connect the middle row nodes to the central node.

④ Les conditions limites de Robin s'écrivent sous la forme

$$a_0 u(0,t) + b_0 \frac{\partial u}{\partial x}(0,t) = k_0$$

$$\text{et } a_1 u(1,t) + b_1 \frac{\partial u}{\partial x}(1,t) = k_1$$

En utilisant la condition miroir pour approximer les dérivées, on obtient :

$$a_0 u(0,t) + b_0 \frac{\partial u}{\partial x}(0,t) = a_0 u_1^n + b_0 \frac{u_2^n - u_1^n}{\Delta x}$$

$$\text{et } a_1 u(1,t) + b_1 \frac{\partial u}{\partial x}(1,t) = a_1 u_{N_{\Delta x}+1}^n + b_1 \frac{u_{N_{\Delta x}+1}^n - u_{N_{\Delta x}}^n}{\Delta x}$$

Ainsi, on a

$$a_0 u_1^n + b_0 \frac{u_2^n - u_1^n}{\Delta x} = k_0$$

$$\Leftrightarrow \left(a_0 - \frac{b_0}{\Delta x} \right) u_1^n + \left(\frac{b_0}{\Delta x} \right) u_2^n = k_0$$

$$\text{et } a_1 u_{N_{\Delta x}+1}^n + b_1 \frac{u_{N_{\Delta x}+1}^n - u_{N_{\Delta x}}^n}{\Delta x} = k_1$$

$$\Leftrightarrow \left(-\frac{b_1}{\Delta x} \right) u_{N_{\Delta x}}^n + \left(a_1 + \frac{b_1}{\Delta x} \right) u_{N_{\Delta x}+1}^n = k_1$$

On peut alors réécrire (*) sous la forme

$$M U^{n+1} = N U^n + F^n$$

avec

$$M =$$

$$\begin{pmatrix} a_0 - \frac{b_0}{\Delta x} & \frac{b_0}{\Delta x} & 0 & & 0 \\ -\theta_n & 1+2\theta_n & -\theta_n & 0 & 0 \\ 0 & & & & 0 \\ 0 & & & & 0 \\ 0 & & & & 0 \\ 0 & & & -\theta_n & 1+2\theta_n & -\theta_n \\ 0 & & & 0 & -\frac{b_1}{\Delta x} & a_1 + \frac{b_1}{\Delta x} \end{pmatrix}$$

$$N =$$

$$\begin{pmatrix} a_0 - \frac{b_0}{\Delta x} & \frac{b_0}{\Delta x} & 0 & & 0 \\ (1-\theta)_n & 1-2(1-\theta)_n & (1-\theta)_n & 0 & 0 \\ 0 & & & & 0 \\ 0 & & & & 0 \\ 0 & & & (1-\theta)_n & 1-2(1-\theta)_n & (1-\theta)_n \\ 0 & & & 0 & -\frac{b_1}{\Delta x} & a_1 + \frac{b_1}{\Delta x} \end{pmatrix}$$

Question 5 :

```
clear all
close all
clc

c = 5; %coefficient de diffusion
CFL = 2*c./(2*c+1); %coefficient de Courant-Fredrichs-Lowy

Nx = 40; %Nombre de mailles -1 en espace
dx = 1./Nx; %Increment d'espace
x = [0:dx:1]; %Vecteur d'espace

Nt = 1000; %Nombre de mailles -1 en temps
dt = CFL*dx.^2./(2*c); %Increment de temps
t = [0:Nt]*dt; %Vecteur de temps

r = c*dt/(dx.^2); %voir question 1

%creation des matrices M et N
e = ones(Nx+1,1);
theta = [0 1 0.5];
for i=1:length(theta)
    N(i,:) = [(1-theta(i))*r 1-2*(1-theta(i))*r (1-theta(i))*r];
    M(i,:) = [-theta(i)*r 1+2*theta(i)*r -theta(i)*r];
end
o = input('Methodes : 1 = Euler Explicte, 2 = Euler Implicit, 3 =
Crank Nicolson ? ');
N = spdiags([N(o,1)*e N(o,2)*e N(o,3)*e],-1:1,Nx+1,Nx+1);
M = spdiags([M(o,1)*e M(o,2)*e M(o,3)*e],-1:1,Nx+1,Nx+1);

%creation de la matrice F et du vecteur u0
k = input('Cas ? ');
f = zeros(Nx+1,Nt+1);
u0 = zeros(Nx+1,1);
for i=1:length(x)
    if k==1
        if x(i)<0.75 && x(i)>0.25
            u0(i) = 1;
        end
    elseif k==2
        f(i,:) = 20*sin(pi*x(i));
        if x(i)<0.5
            u0(i) = 1;
        end
    elseif k==3
        for j=1:length(t)
            f(i,j) = (t(j)/2)*exp(-4096*((x(i)-0.5).^2));
        end
        u0(i) = 1+x(i)-sin(pi*x(i))-0.25*sin(10*pi*x(i));
    end
end
%creation de la matrice contenant les differents vecteurs Un
u = zeros(Nx+1,Nt+1); u(:,1) = u0;

%application des conditions limites de Dirichlet
M(1,:) = 0; M(1,1) = 1; M(Nx+1,:) = 0; M(Nx+1,Nx+1) = 1;
N(1,:) = 0; N(1,1) = 1; N(Nx+1,:) = 0; N(Nx+1,Nx+1) = 1;
```

```

%calcul de l'evolution de la temperature
for i=2:length(t)
    u(:,i) = inv(M)*(N*u(:,i-1)+f(:,i)*dt);
end

%creation et enregistrement (en commentaires) du film
if (o==1) m = 'EulerExplicite';
elseif (o==2) m = 'EulerImplicite';
elseif (o==3) m = 'CrankNicolson'; end
%mov=VideoWriter(strcat('./VideoDirichlet',strcat(m,strcat('Cas',str
cat(num2str(k),'.avi')))));
for i=1:length(t)
    plot(x,u(:,i)); grid on
    %F = getframe(gcf);
    %open(mov);
    %writeVideo(mov,F);
    pause(0.01)
end

%creation de la figure regroupant les temperatures aux temps donnees
figure
t0 = [1 5 10 20 30 40 50 75 100 200 300 400 500 600 700 800 900 Nt];
for i=t0
    hold on
    plot(x,u(:,i)); grid on
    hold off
end

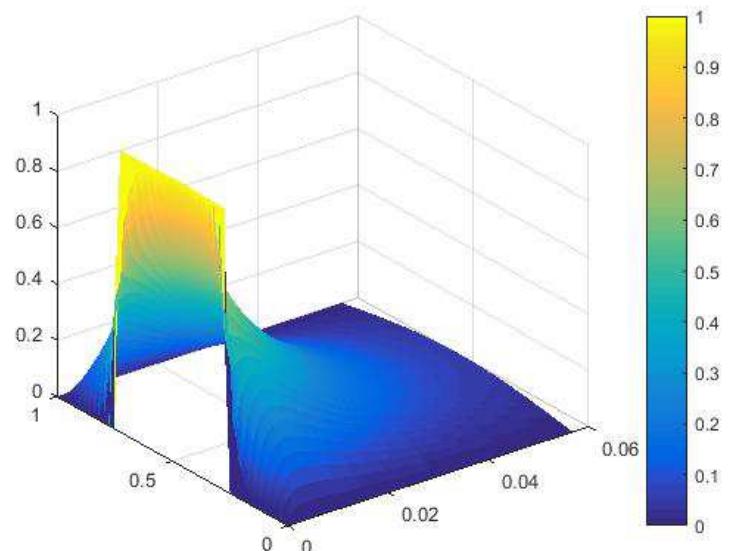
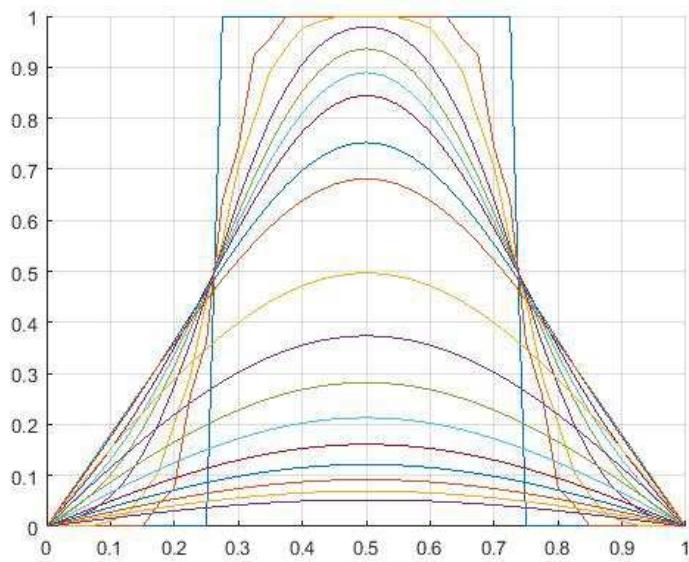
%creation de la figure representant la variation de la temperature
%par rapport au temps
figure; surf(t,x,u,'EdgeColor','none'); colorbar

```

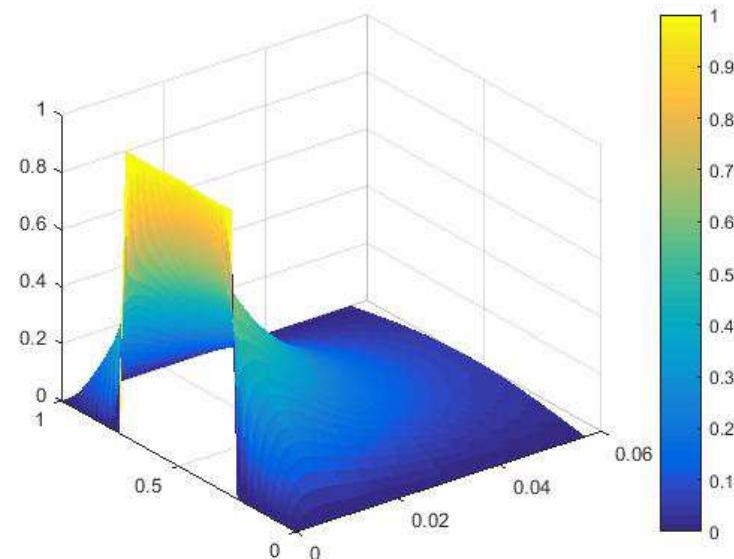
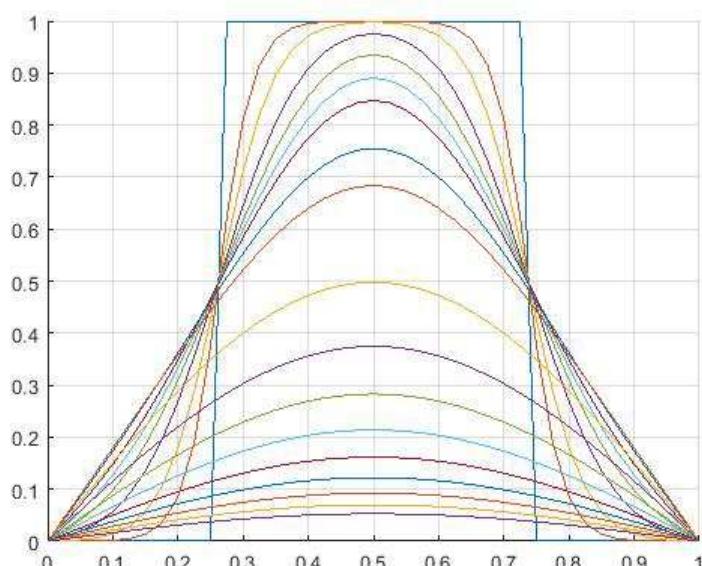
Question 6 :

Cas 1 :

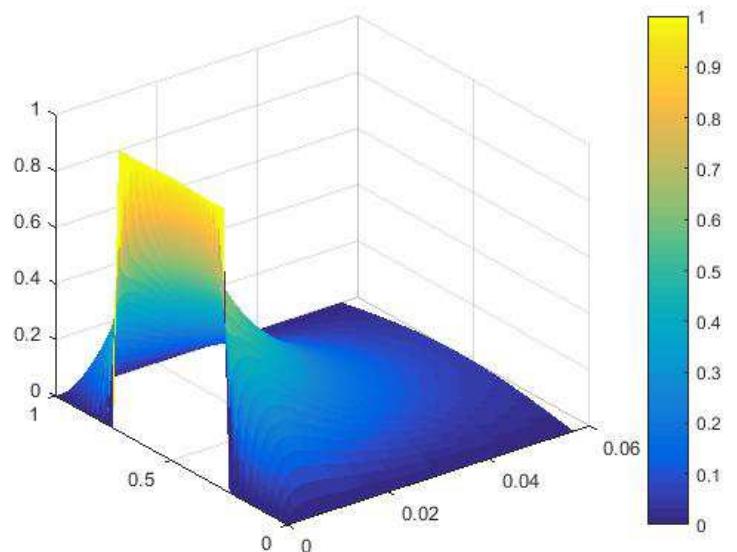
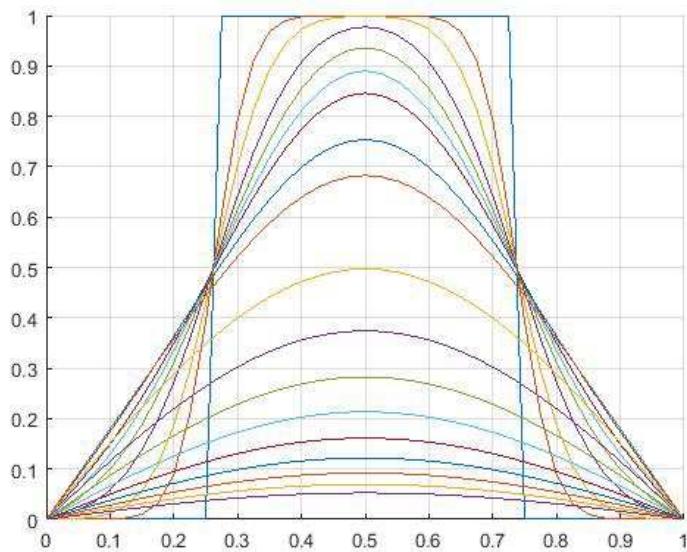
- Euler Explicite



- Euler Implicit

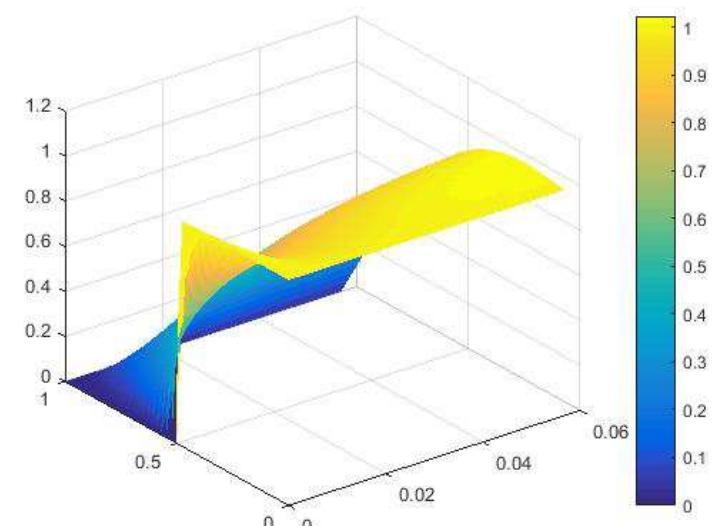
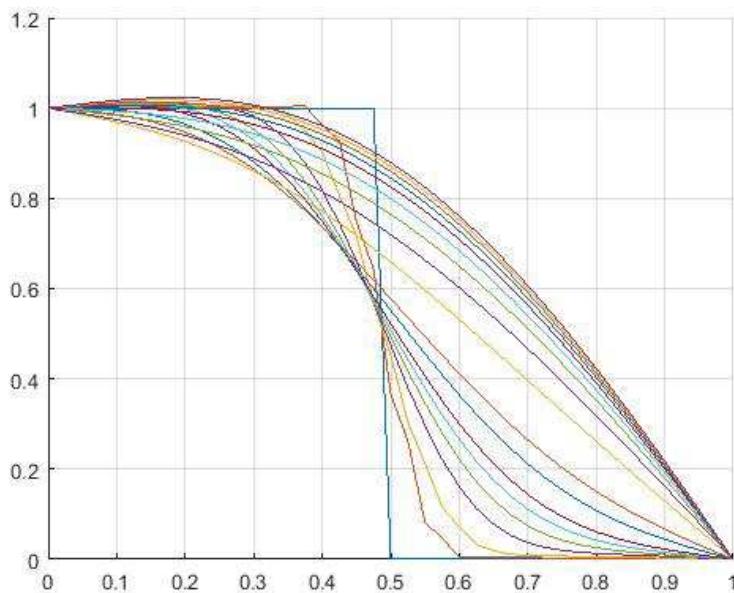


- Crank-Nicholson

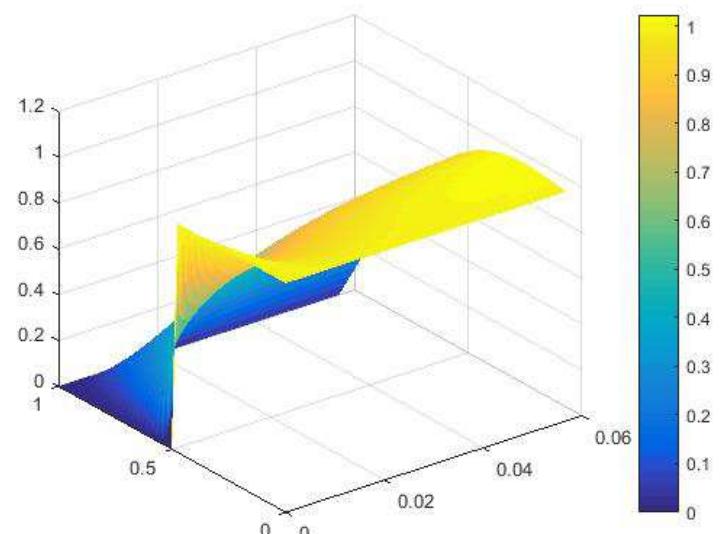
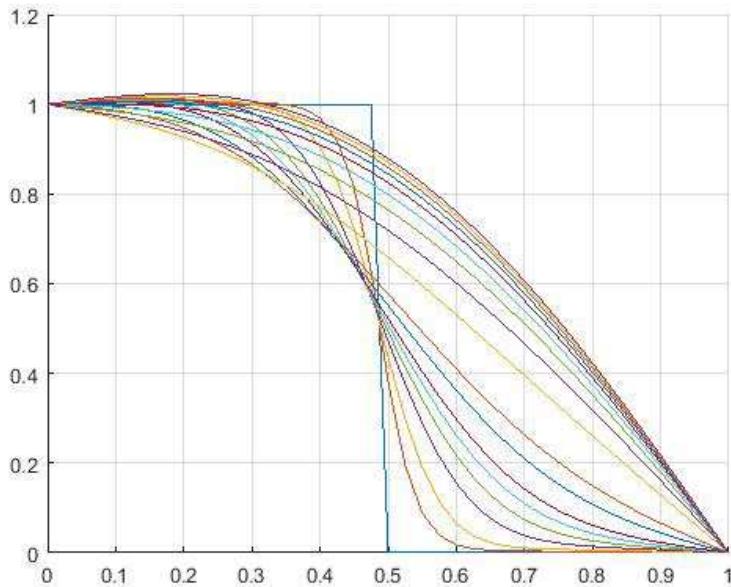


Cas 2 :

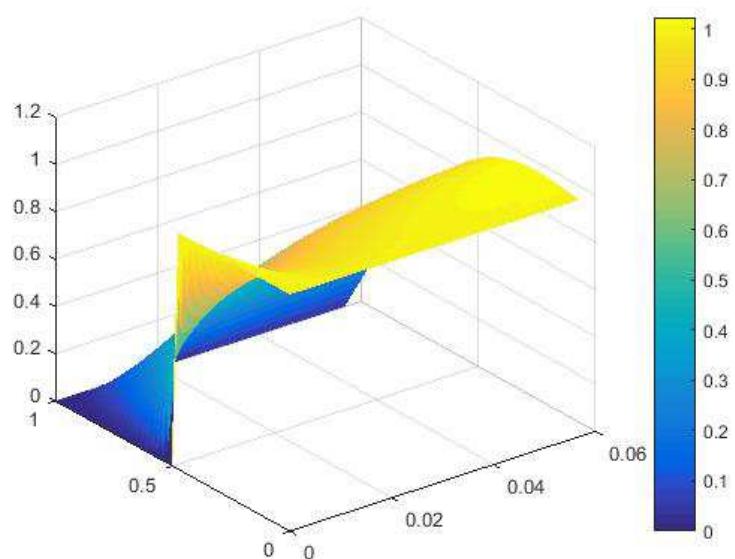
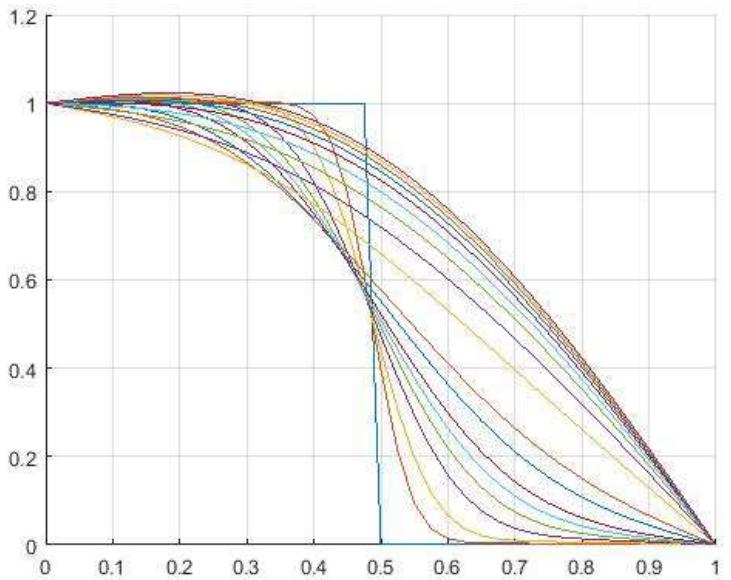
- Euler Explicite



- Euler Implicite

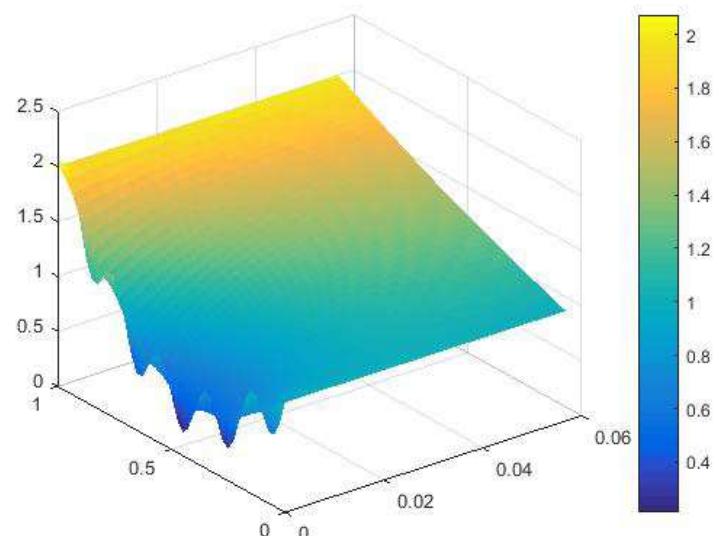
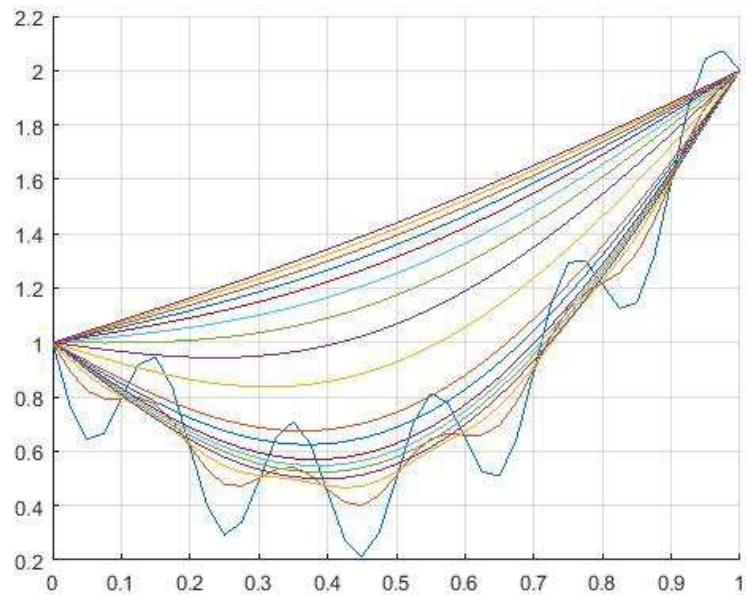


- Crank-Nicholson

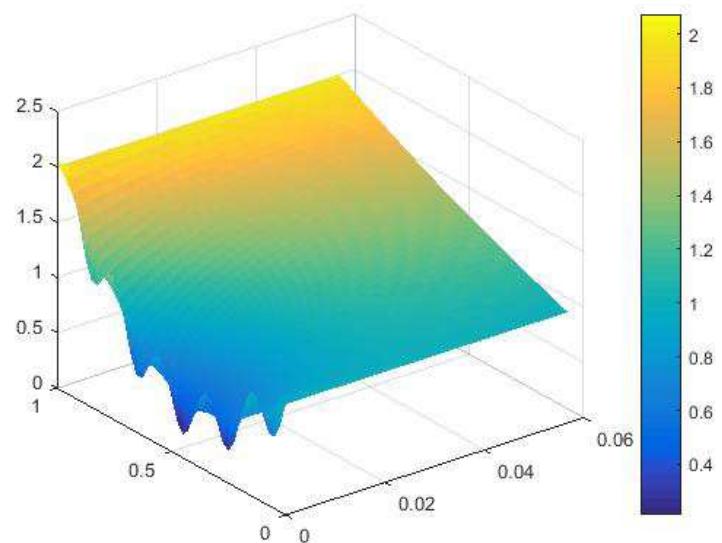
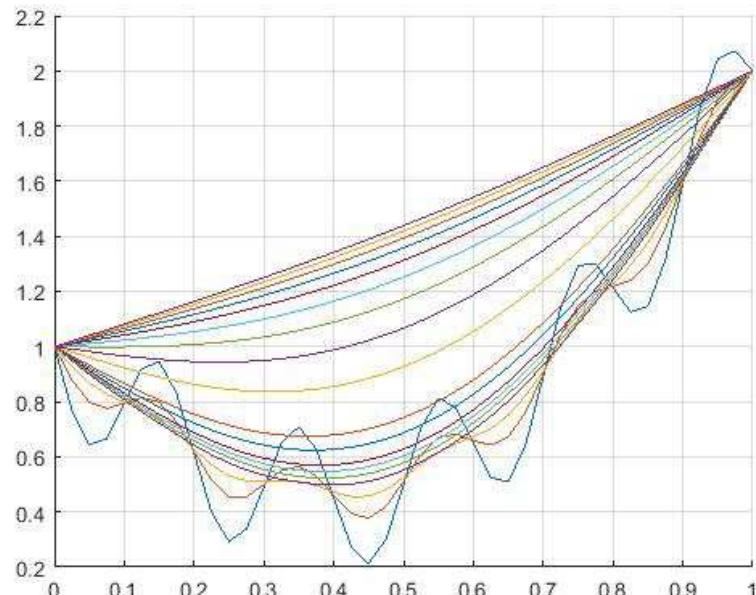


Cas 3 :

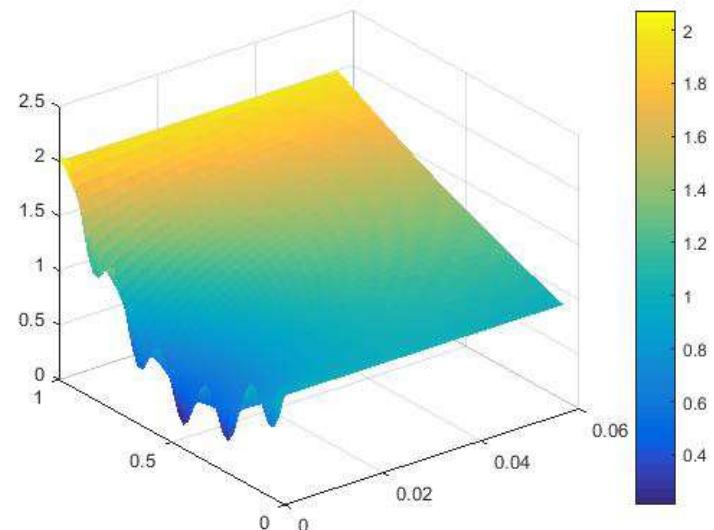
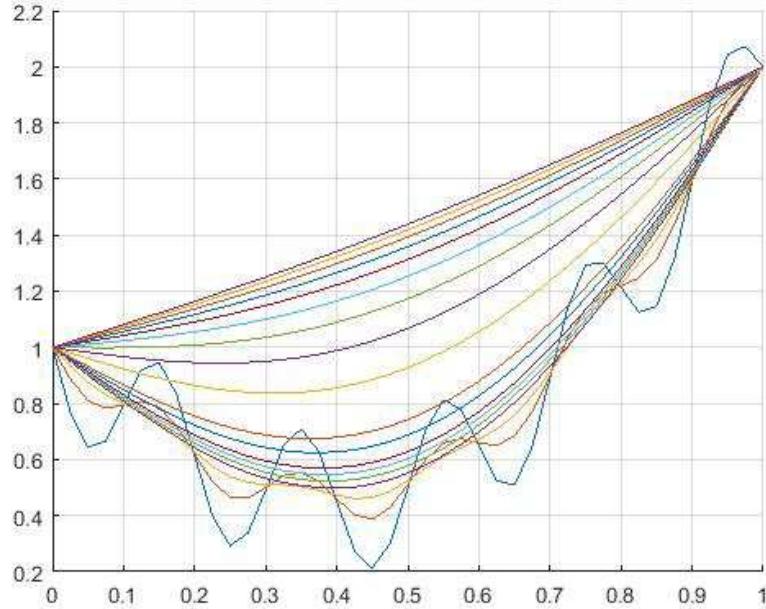
- Euler Explicite



- Euler Implicite



- Crank-Nicholson



On peut voir qu'avec des conditions limites de Dirichlet, la température, dans les cas 1 et 3, tend vers une différence constante entre les deux bords, qui peut alors se représenter par une droite affine d'équation $T(x) = (T(1) - T(0))x + T(0)$.

Ce serait aussi le cas dans le cas 2, sauf que la fonction $f(x, t)$, qui fait office de chauffage, crée une "bosse" sur la courbe. Cependant, comme la f ne dépend pas du temps dans ce cas, la température atteint tout de même une position d'équilibre au bout d'un certain moment.

Question 7 :

```
clear all
close all
clc

c = 5; %coefficient de diffusion
CFL = 2*c./(2*c+1); %coefficient de Courant-Fredrichs-Lowy

Nx = 40; %Nombre de mailles -1 en espace
dx = 1./Nx; %Increment d'espace
x = [0:dx:1]; %Vecteur d'espace

Nt = 1000; %Nombre de mailles -1 en temps
dt = CFL*dx.^2./(2*c); %Increment de temps
t = [0:Nt]*dt; %Vecteur de temps

r = c*dt/(dx.^2); %voir question 1

%creation des matrices M et N
e = ones(Nx+1,1);
theta = [0 1 0.5];
for i=1:length(theta)
    N(i,:) = [(1-theta(i))*r 1-2*(1-theta(i))*r (1-theta(i))*r];
    M(i,:) = [-theta(i)*r 1+2*theta(i)*r -theta(i)*r];
end
o = input('Methodes : 1 = Euler Explicte, 2 = Euler Implicite, 3 =
Crank Nicolson ? ');
M = spdiags([M(o,1)*e M(o,2)*e M(o,3)*e],-1:1,Nx+1,Nx+1);
N = spdiags([N(o,1)*e N(o,2)*e N(o,3)*e],-1:1,Nx+1,Nx+1);

%creation de la matrice F et du vecteur u0
k = input('Cas ? ');
f = zeros(Nx+1,Nt+1);
u0 = zeros(Nx+1,1);
for i=1:length(x)
    if k==1
        if x(i)<0.75 && x(i)>0.25
            u0(i) = 1;
        end
    elseif k==2
        f(i,:) = 20*sin(pi*x(i));
        if x(i)<0.5
            u0(i) = 1;
        end
    elseif k==3
        for j=1:length(t)
            f(i,j) = (t(j)/2)*exp(-4096*((x(i)-0.5).^2));
        end
        u0(i) = 1+x(i)-sin(pi*x(i))-0.25*sin(10*pi*x(i));
    end
end
%creation de la matrice contenant les differents vecteurs Un
u = zeros(Nx+1,Nt+1); u(:,1) = u0;
```

```

%application des conditions limites de Neumann
M(1,:) = 0; M(Nx+1,:) = 0; N(1,:) = 0; N(Nx+1,:) = 0;
M(1,1) = -1/dx; M(1,2) = 1/dx; M(Nx+1,Nx) = -1/dx; M(Nx+1,Nx+1) =
1/dx;
N(1,1) = -1/dx; N(1,2) = 1/dx; N(Nx+1,Nx) = -1/dx; N(Nx+1,Nx+1) =
1/dx;
if k==1
    f(1,:) = 0; f(Nx+1,:) = 0;
elseif k==2
    f(1,:) = 1; f(Nx+1,:) = 0;
elseif k==3
    f(1,:) = -1/10; f(Nx+1,:) = -1/10;
end
u(1,1) = u(2,1)-dx*f(1,1); u(Nx+1,1) = u(Nx,1)+dx*f(Nx+1,1);

%calcul de l'evolution de la temperature
for i=2:length(t)
    u(:,i) = inv(M)*(N*u(:,i-1)+f(:,i)*dt);
end

%creation et enregistrement (en commentaires) du film
if (o==1) m = 'EulerExplicite';
elseif (o==2) m = 'EulerImplicite';
elseif (o==3) m = 'CrankNicolson'; end
%mov =
VideoWriter(strcat('./VideoNeumann',strcat(m,strcat('Cas',strcat(num
2str(k),'.avi')))));
for i=1:length(t)
    plot(x,u(:,i)); grid on
    %F = getframe(gcf);
    %open(mov);
    %writeVideo(mov,F);
    pause(0.005)
end

%creation de la figure regroupant les temperatures aux temps donnees
figure
t0 = [1 5 10 20 30 40 50 75 100 200 300 400 500 600 700 800 900 Nt];
for i=t0
    hold on
    plot(x,u(:,i)); grid on
    hold off
end

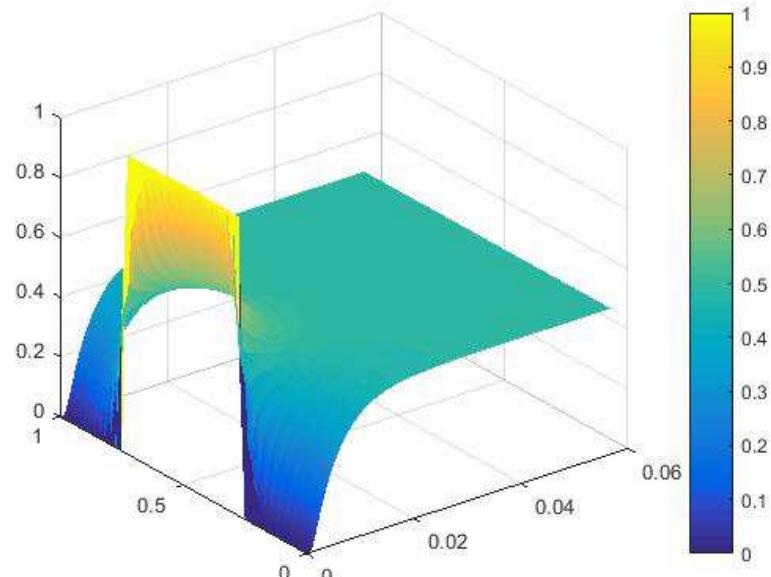
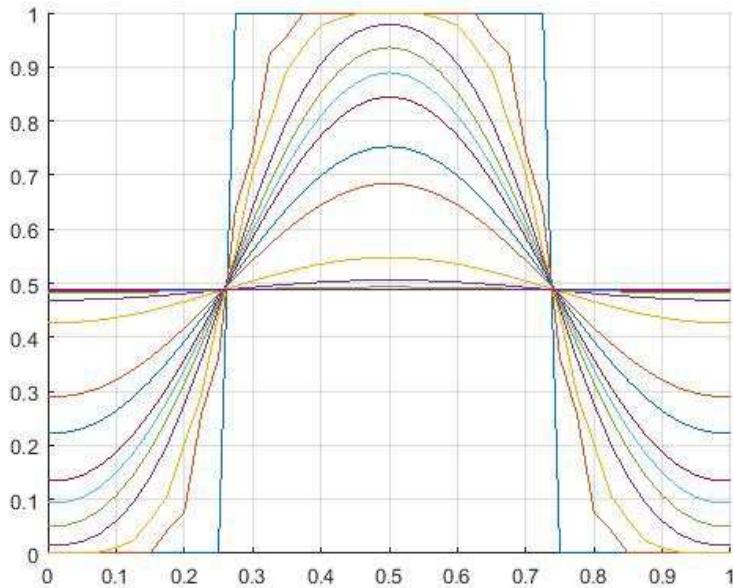
%creation de la figure representant la variation de la temperature
%par rapport au temps
figure; surf(t,x,u,'EdgeColor','none'); colorbar

```

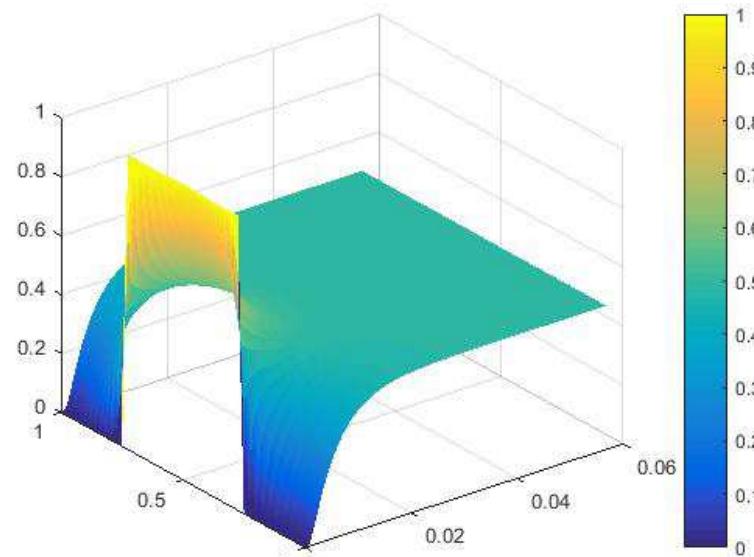
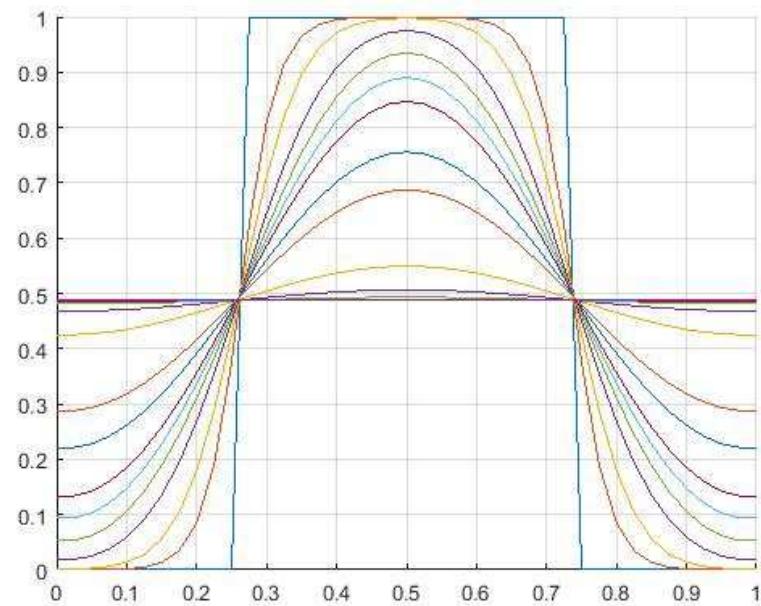
Question 8 :

Cas 1 :

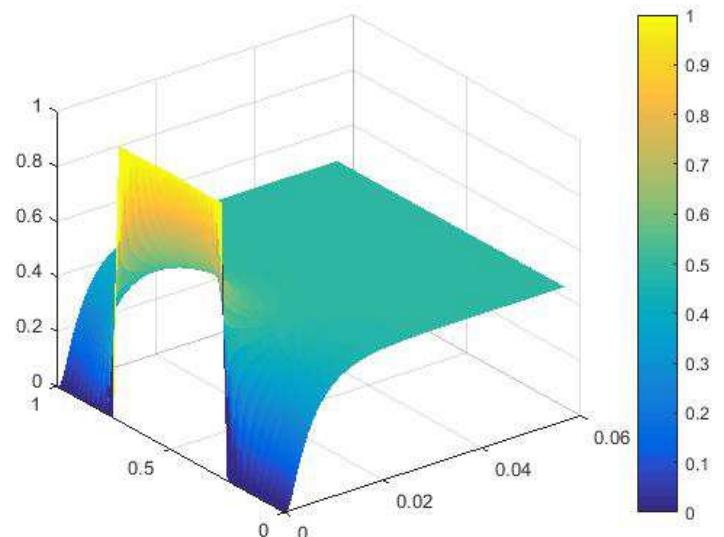
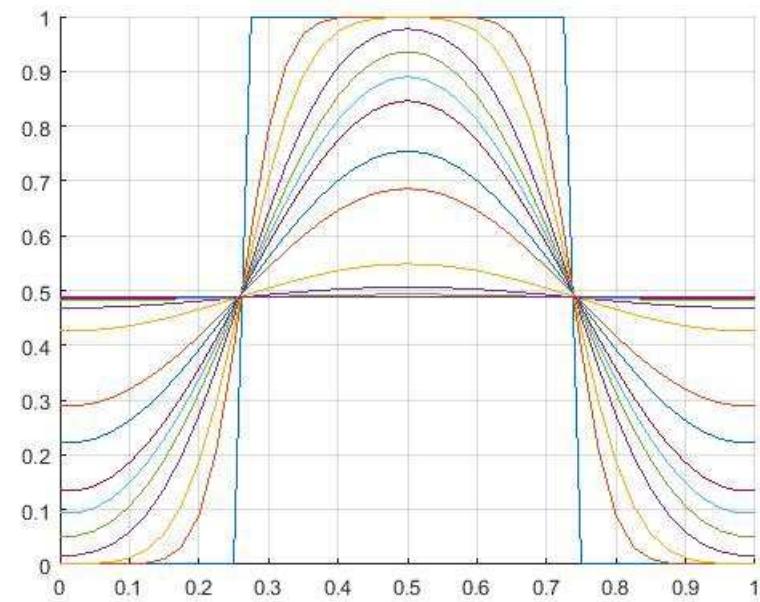
- Euler Explicite



- Euler Implicit

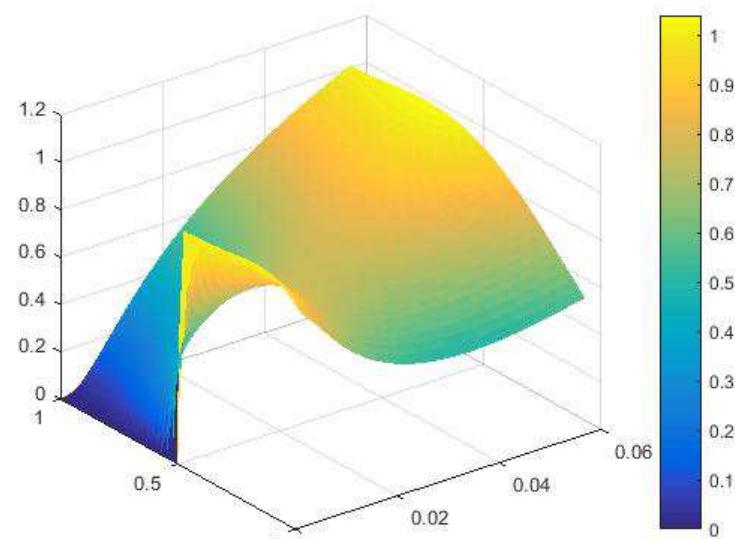
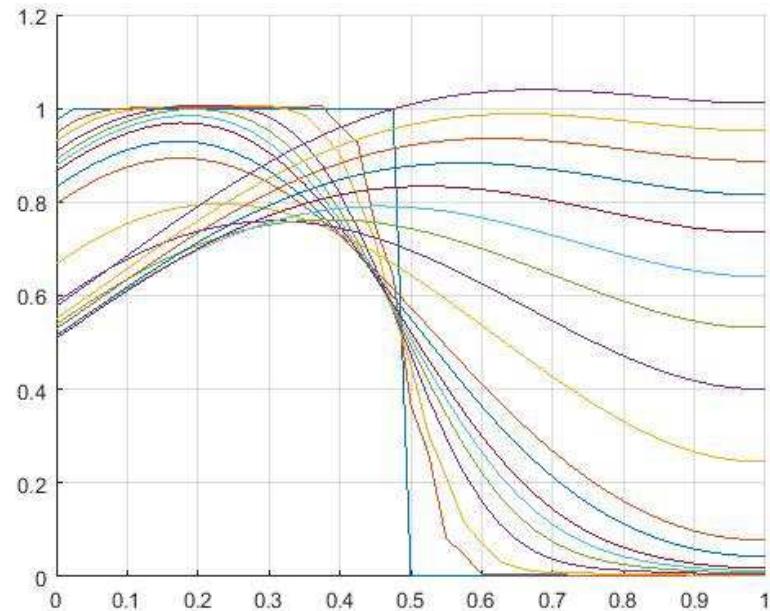


- Crank- Nicholson

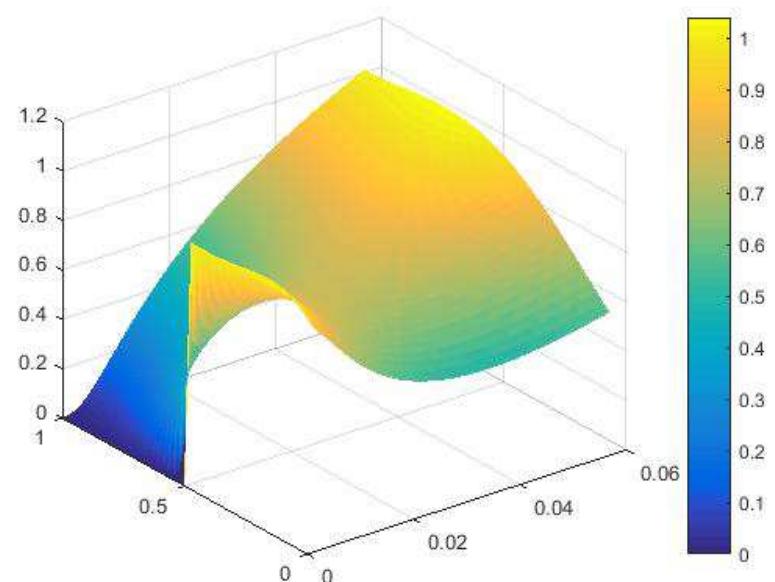
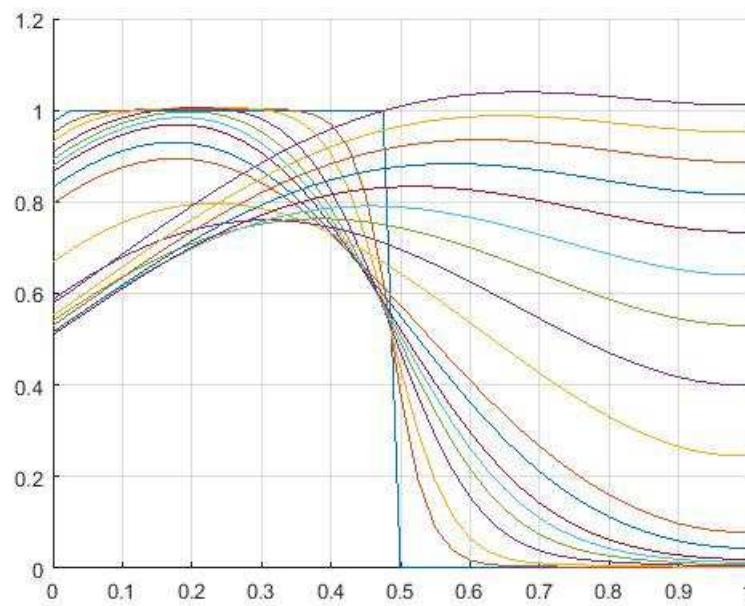


Cas 2 :

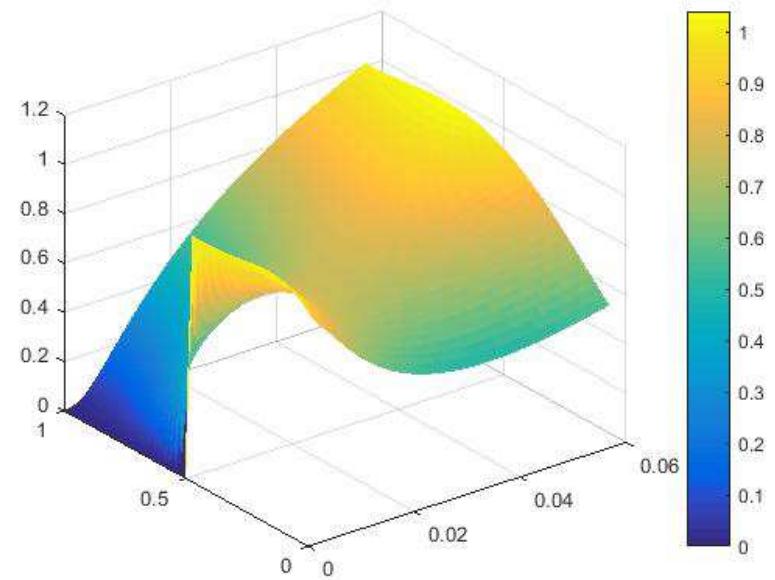
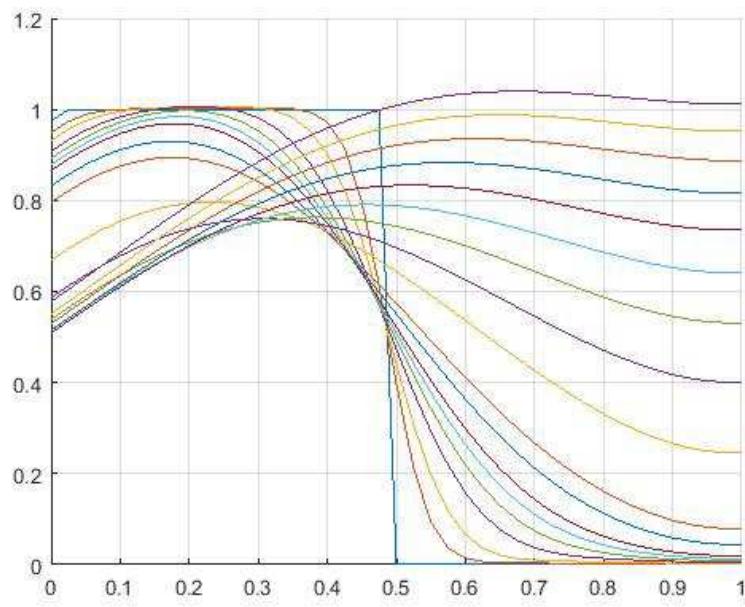
- Euler Explicite



- Euler Implicite

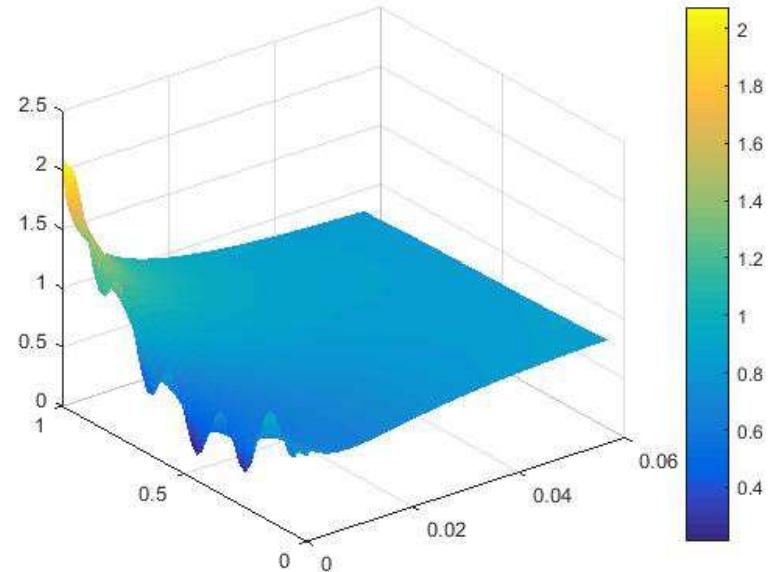
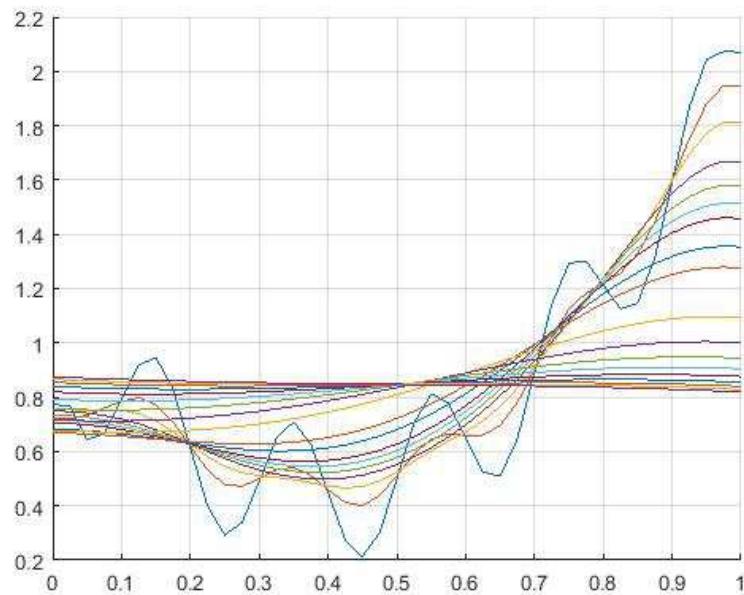


- Crank-Nicholson

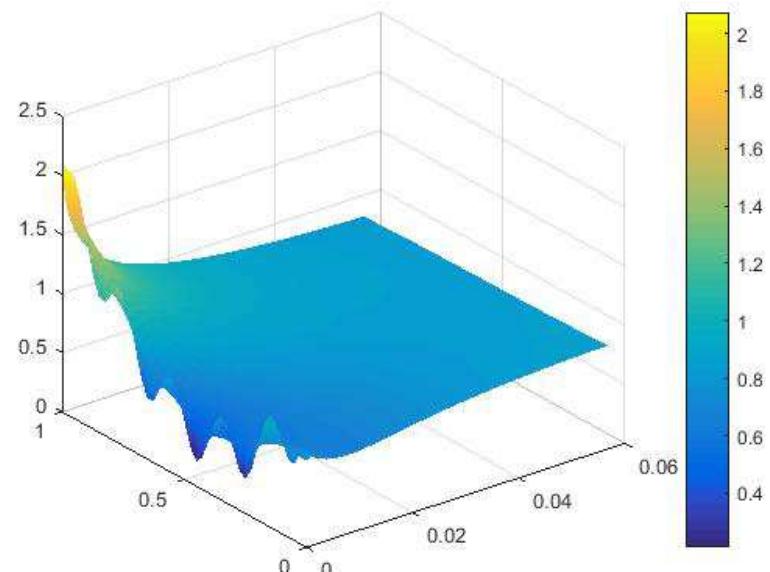
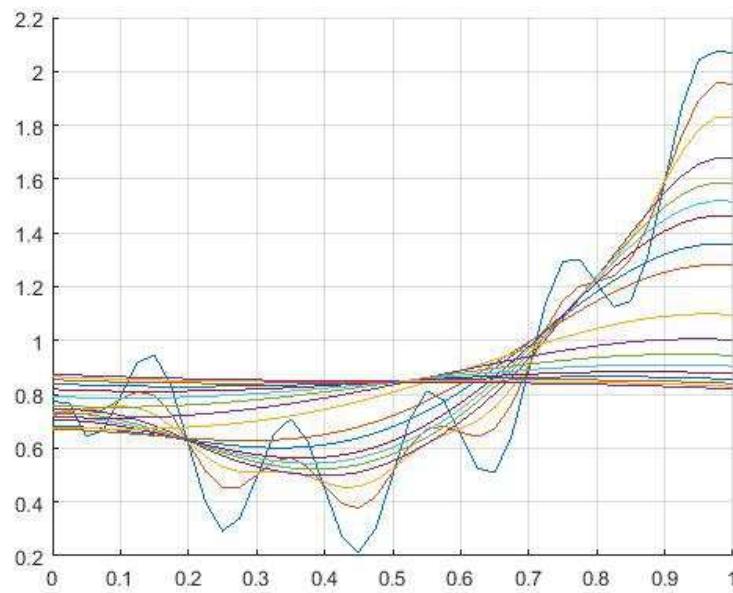


Cas 3 :

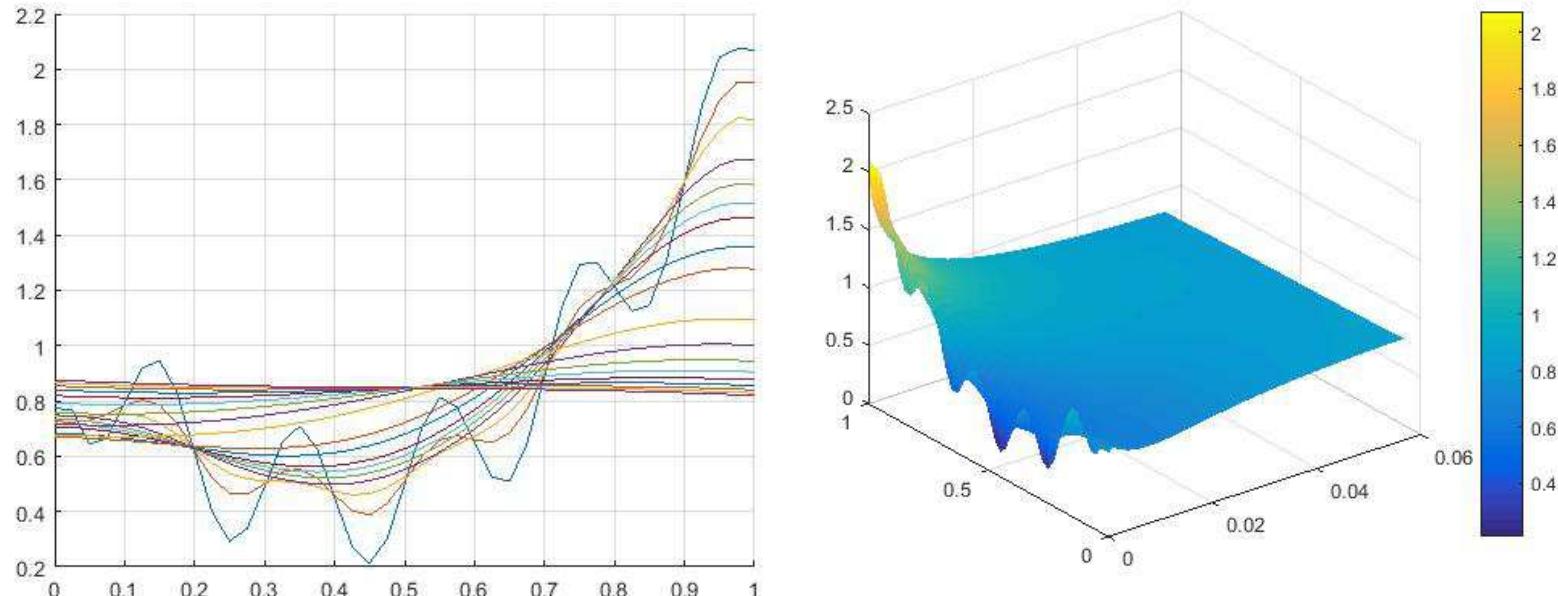
- Euler Explicite



- Euler Implicite



- Crank-Nicholson



Dans le cas de conditions limites de Neumann, on peut voir que l'évolution de la température dans la barre tend, dans les cas 1 et 3, vers une droite affine d'équation $T(x) = Ax + T(0)$. Cela est dû au fait que, dans ces cas-ci, les 2 conditions limites sont identiques ; ce n'est pas la même chose pour le cas 2, car, dans celui-ci, les conditions limites sont différentes sur chacun des bords.

On a en réalité la courbe de température qui tend vers une courbe "à minima d'énergie" : c'est-à-dire qu'elle tend vers une courbe dont la dérivée seconde est constante et vaut $a = \frac{\nu(1)-\nu(0)}{1-0} = \nu(1) - \nu(0)$, où $\nu(x)$ correspond à la dérivée première de la courbe au point x . Dans les cas 1 et 3, comme $\nu(x)$ et $\nu(x)$ sont identiques, on a la dérivée seconde qui est nulle, et donc la courbe qui tend vers une droite affine.

Question 9 :

```
clear all
close all
clc

c = 5; %coefficient de diffusion
CFL = 2*c./(2*c+1); %coefficient de Courant-Fredrichs-Lowy

Nx = 40; %Nombre de mailles -1 en espace
dx = 1./Nx; %Increment d'espace
x = [0:dx:1]; %Vecteur d'espace

Nt = 1000; %Nombre de mailles -1 en temps
dt = CFL*dx.^2./(2*c); %Increment de temps
t = [0:Nt]*dt; %Vecteur de temps

r = c*dt/(dx.^2); %voir question 1

%creation des matrices M et N
e = ones(Nx+1,1);
theta = [0 1 0.5];
for i=1:length(theta)
    N(i,:) = [(1-theta(i))*r 1-2*(1-theta(i))*r (1-theta(i))*r];
    M(i,:) = [-theta(i)*r 1+2*theta(i)*r -theta(i)*r];
end
o = input('Methodes : 1 = Euler Explicte, 2 = Euler Implicite, 3 =
Crank Nicolson ? ');
M = spdiags([M(o,1)*e M(o,2)*e M(o,3)*e],-1:1,Nx+1,Nx+1);
N = spdiags([N(o,1)*e N(o,2)*e N(o,3)*e],-1:1,Nx+1,Nx+1);

%creation de la matrice F et du vecteur u0
k = input('Cas ? ');
f = zeros(Nx+1,Nt+1);
u0 = zeros(Nx+1,1);
for i=1:length(x)
    if k==1
        if x(i)<0.75 && x(i)>0.25
            u0(i) = 1;
        end
    elseif k==2
        f(i,:) = 20*sin(pi*x(i));
        if x(i)<0.5
            u0(i) = 1;
        end
    elseif k==3
        for j=1:length(t)
            f(i,j) = (t(j)/2)*exp(-4096*((x(i)-0.5).^2));
        end
        u0(i) = 1+x(i)-sin(pi*x(i))-0.25*sin(10*pi*x(i));
    end
end
%creation de la matrice contenant les differents vecteurs Un
u = zeros(Nx+1,Nt+1); u(:,1) = u0;
```

```

%application des conditions limites de Robin
M(1,:) = 0; M(Nx+1,:) = 0; N(1,:) = 0; N(Nx+1,:) = 0;
if k==1
    f(1,:) = 0; f(Nx+1,:) = 0;
    M(1,1) = 0.2+1./dx; M(1,2) = -1./dx; M(Nx+1,Nx) = -1./dx;
    M(Nx+1,Nx+1) = 0.2+1./dx;
    N(1,1) = 0.2+1./dx; N(1,2) = -1./dx; N(Nx+1,Nx) = -1./dx;
    N(Nx+1,Nx+1) = 0.2+1./dx;
elseif k==2
    f(1,:) = 1; f(Nx+1,:) = 0;
    M(1,1) = 1+1./dx; M(1,2) = -1./dx; M(Nx+1,Nx) = -2./dx;
    M(Nx+1,Nx+1) = (1/7)+2./dx;
    N(1,1) = 1+1./dx; N(1,2) = -1./dx; N(Nx+1,Nx) = -2./dx;
    N(Nx+1,Nx+1) = (1/7)+2./dx;
elseif k==3
    f(1,:) = -0.5; f(Nx+1,:) = -0.5;
    M(1,1) = 1+1./dx; M(1,2) = -1./dx; M(Nx+1,Nx) = -1./dx;
    M(Nx+1,Nx+1) = 1+1./dx;
    N(1,1) = 1+1./dx; N(1,2) = -1./dx; N(Nx+1,Nx) = -1./dx;
    N(Nx+1,Nx+1) = 1+1./dx;
end
u(1,1) = (f(1,1)-M(1,2)*u(2,1))./M(1,1);
u(Nx+1,1) = (f(Nx+1,1)-M(Nx+1,Nx)*u(Nx,1))./M(Nx+1,Nx+1);

%calcul de l'evolution de la temperature
for i=2:length(t)
    u(:,i) = inv(M)*(N*u(:,i-1)+f(:,i)*dt);
end

%creation et enregistrement (en commentaires) du film
if (o==1) m = 'EulerExplicite';
elseif (o==2) m = 'EulerImplicite';
elseif (o==3) m = 'CrankNicolson'; end
%mov =
VideoWriter(strcat('./VideoRobin',strcat(m,strcat('Cas',strcat(num2str(k),'.avi')))));
for i=1:length(t)
    plot(x,u(:,i)); grid on
    %F = getframe(gcf);
    %open(mov);
    %writeVideo(mov,F);
    pause(0.005)
end

%creation de la figure regroupant les temperatures aux temps donnees
figure
t0 = [1 5 10 20 30 40 50 75 100 200 300 400 500 600 700 800 900 Nt];
for i=t0
    hold on
    plot(x,u(:,i)); grid on
    hold off
end

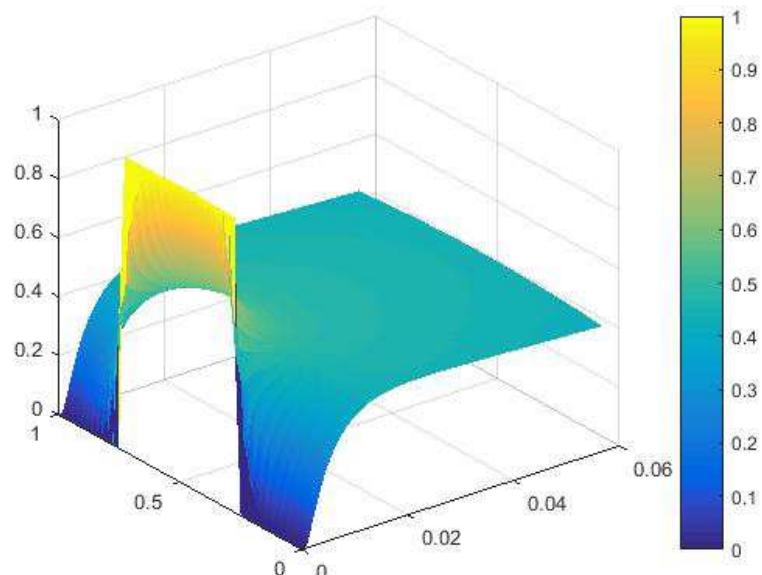
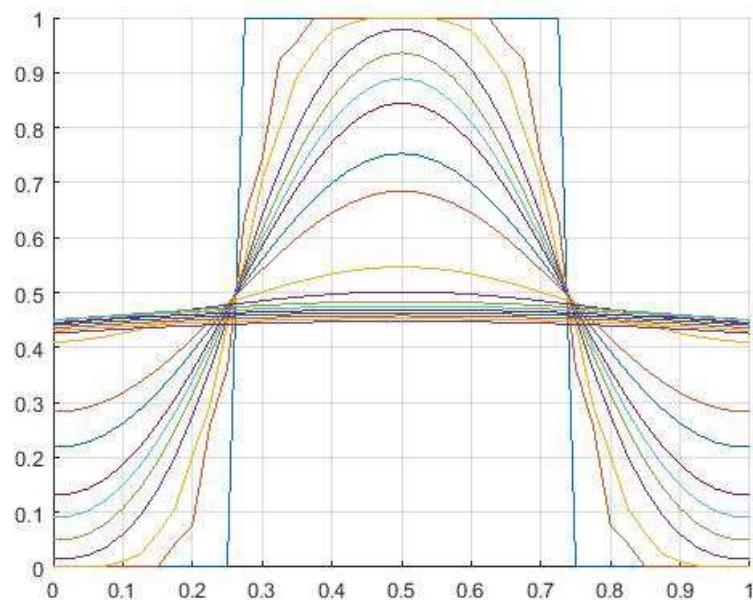
%creation de la figure representant la variation de la temperature
%par rapport au temps
figure; surf(t,x,u,'EdgeColor','none'); colorbar

```

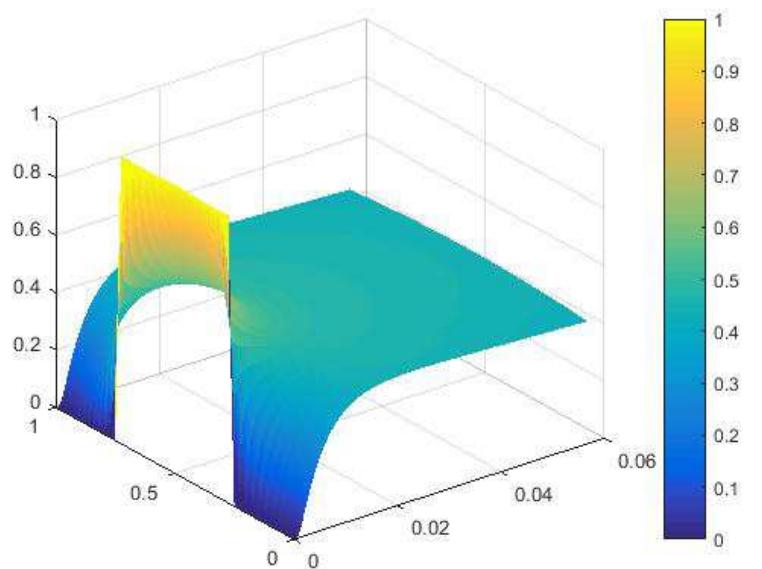
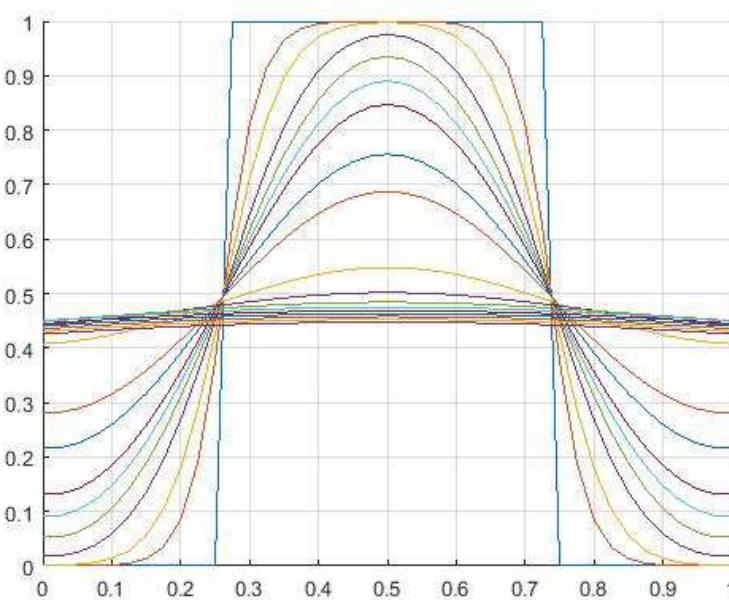
Question 10 :

Cas 1 :

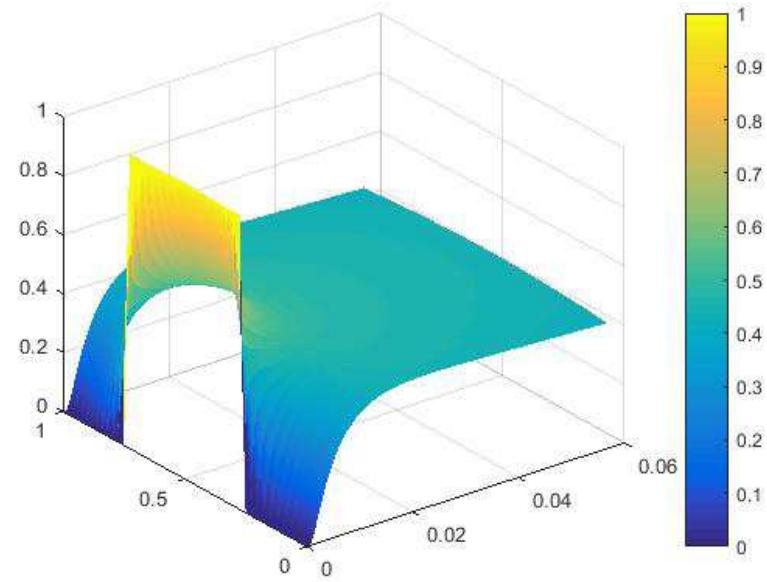
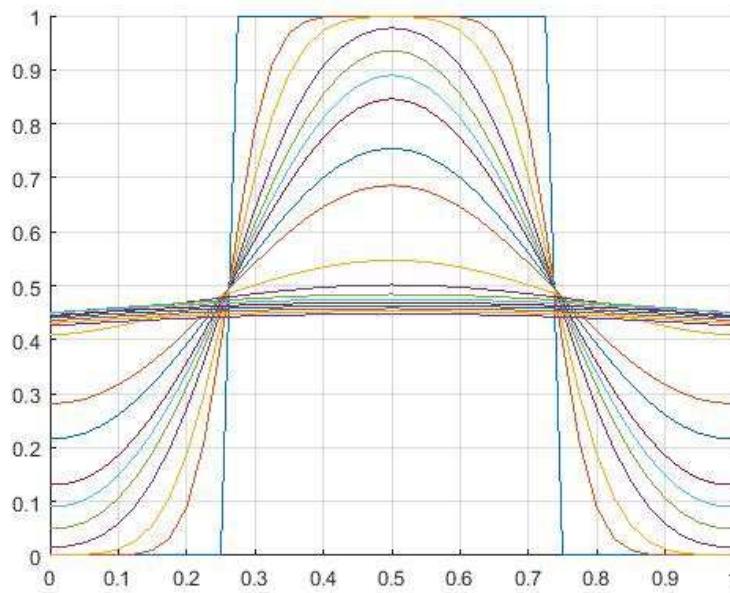
- Euler Explicite



- Euler Implicit

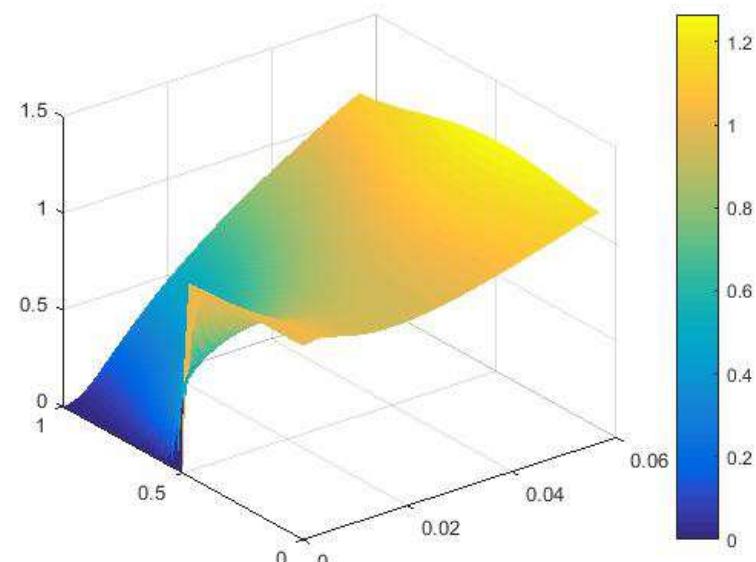
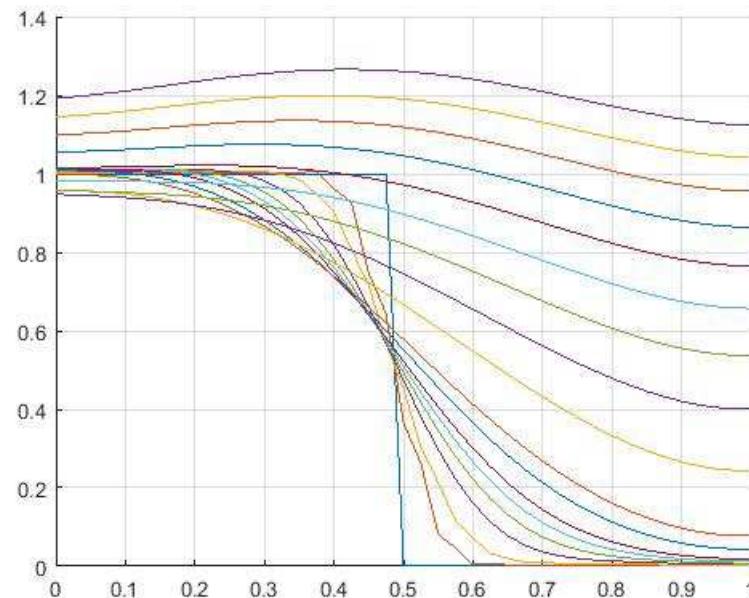


- Crank-Nicholson

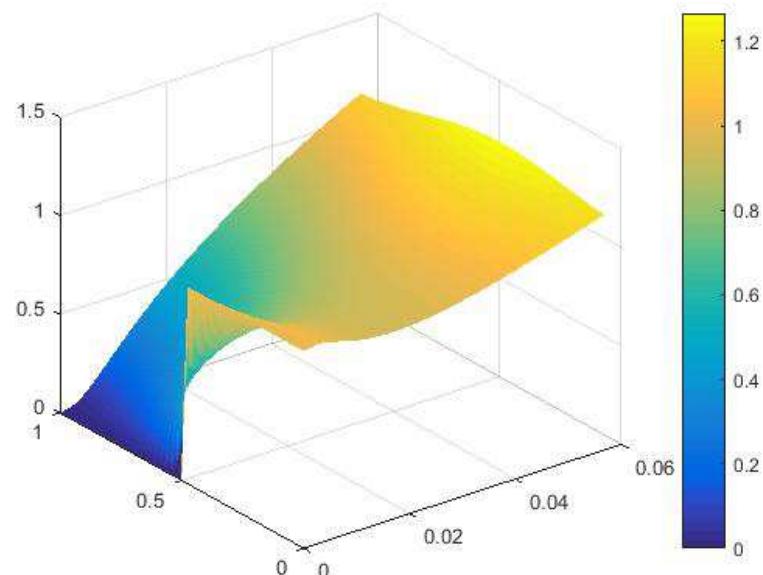
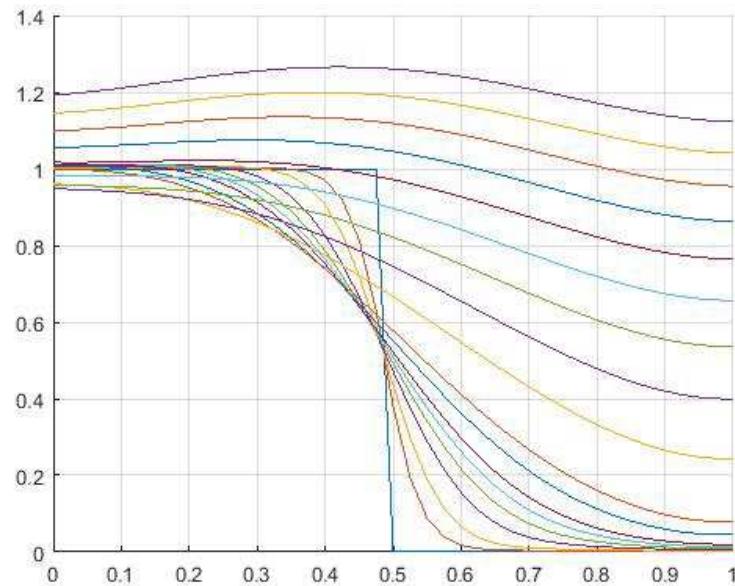


Cas 2 :

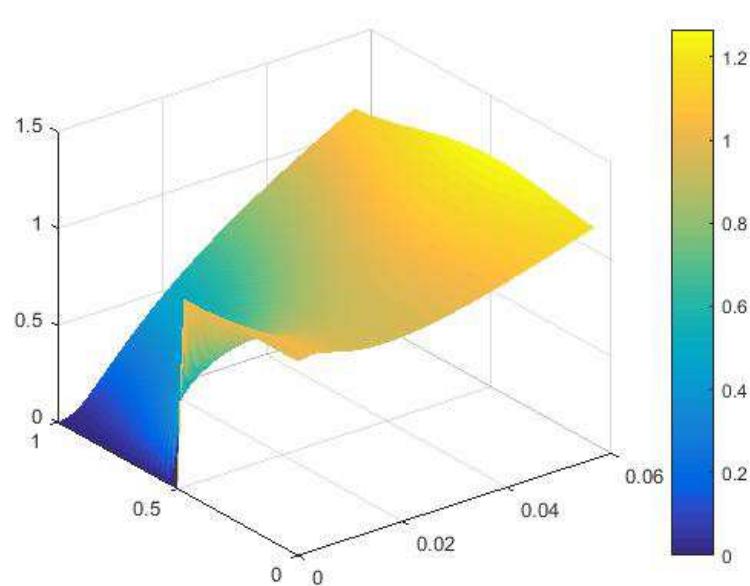
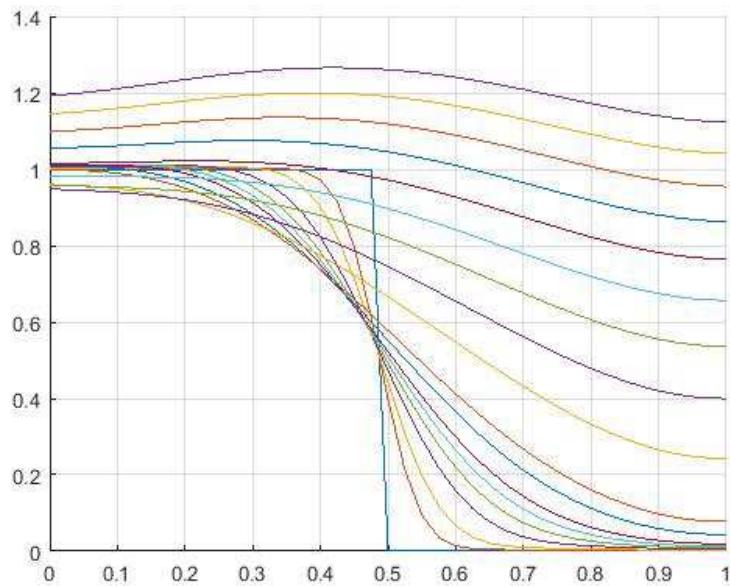
- Euler Explicite



- Euler Implicite

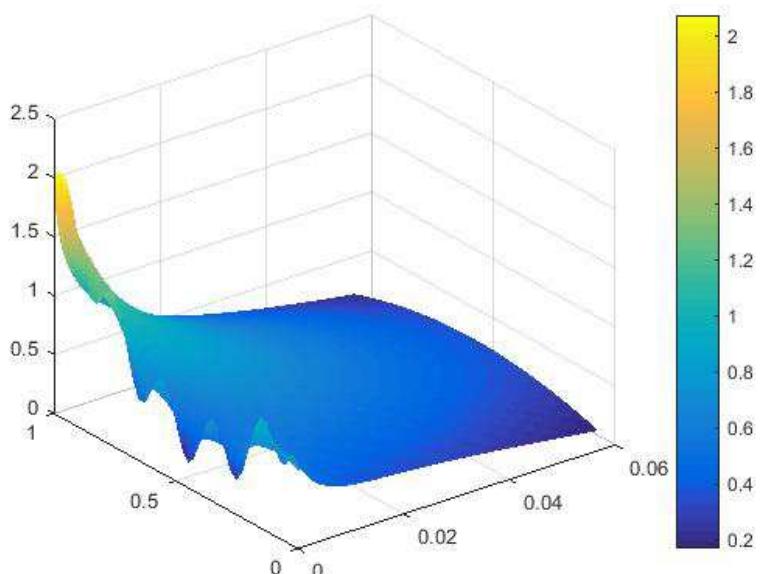
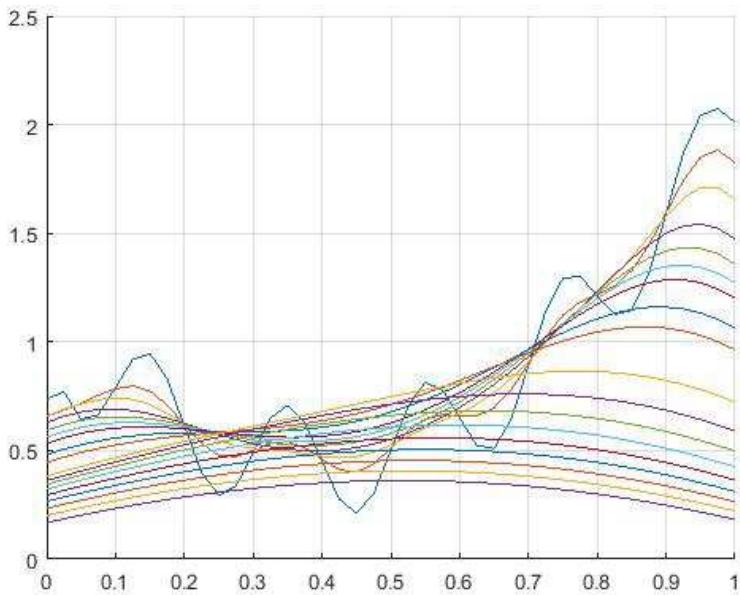


- Crank-Nicholson

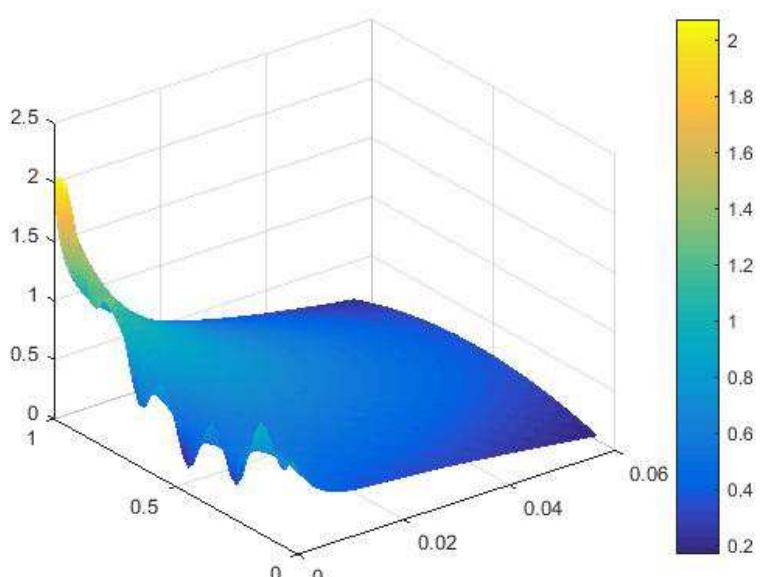
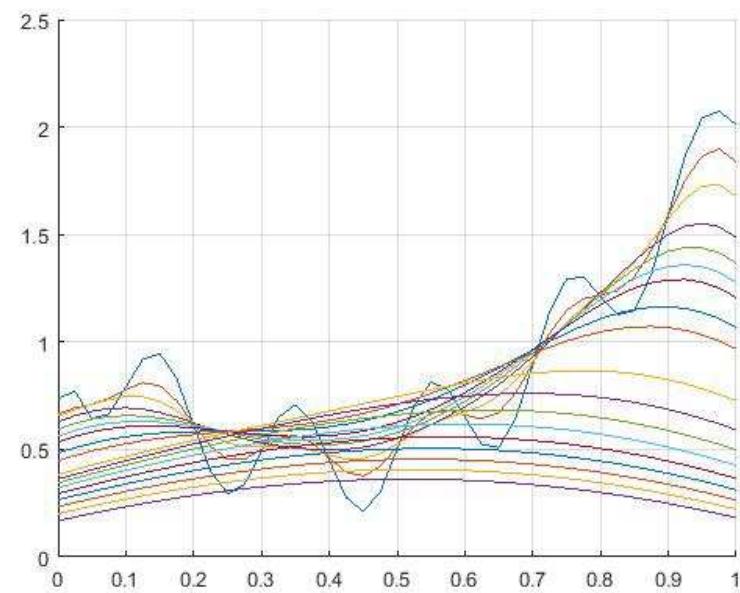


Cas 3 :

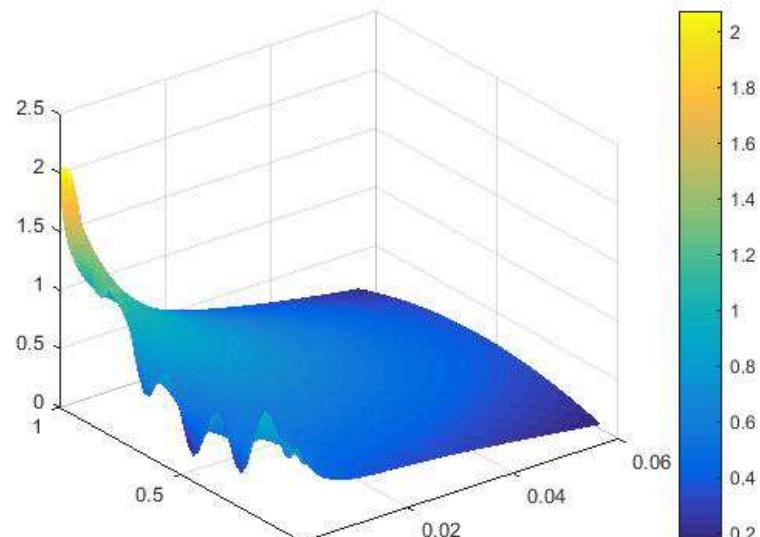
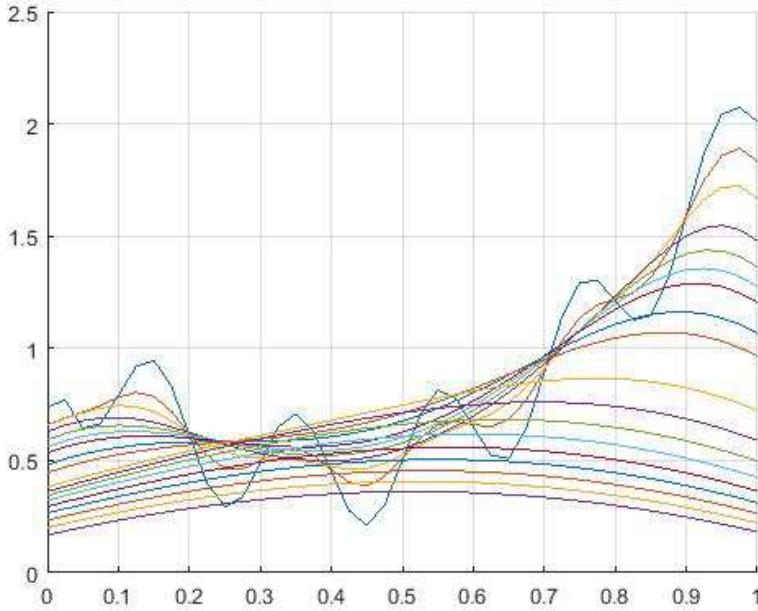
- Euler Explicite



- Euler Implicit



- Crank-Nicholson



Pour des conditions limites de Robin, on a une relation directe entre les valeurs aux bords et leurs dérivées premières. On peut ainsi remarquer sur les graphes obtenus que, si la valeur au bord est modifiée, il en va de même pour la valeur de la pente en ce point.

On peut voir que sur chacun des cas, la courbe tend vers une courbe “à minima” d’énergie, avec sa dérivée seconde qui tend vers une constante. Comme les conditions limites de Robin s’écrivent de la forme $a_i u(i, t) + b_i \frac{\partial u}{\partial x}(i, t) = f(i)$ (avec $i = 0$ ou 1 suivant le bord), on a alors $\frac{\partial u}{\partial x}(i, t) = \frac{f(i) - a_i u(i, t)}{b_i}$. On a alors, comme la dérivée seconde tend vers une constante à t très grand : $a(t) \approx \frac{\partial u}{\partial x}(1, t) - \frac{\partial u}{\partial x}(0, t) = \frac{f(1) - a_1 u(1, t)}{b_1} - \frac{f(0) - a_0 u(0, t)}{b_0} \approx \frac{a_0 b_1 u(0) - a_1 b_0 u(1) + f(1)b_0 - f(0)b_1}{b_0 b_1}$, avec $u(i)$ la valeur limite au bord i atteinte par la température.

Question 11 :

On peut voir l'instabilité du schéma d'Euler Explicite à l'aide du film nommé *VideoInstableRobinCas3.avi*. Normalement, il doit représenter l'évolution du cas 3 soumis à des conditions limites de Robin, sauf qu'ici, le CFL a été initialisé à la valeur 1.5, rendant ainsi le schéma d'Euler Explicite instable.