Numerical Solution of 2D Heat Equation

21-11-2022

Deliverable Task 1

Anastasia Papadaki
anastasia.p.papadaki@fau.de

François Dugourd
francois.dugourd@etu.chimieparistech.psl.eu

Abstract

Consider the dimensionless 2D heat equation:

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w^2}{\partial x^2} + \frac{\partial^2 w^2}{\partial y^2} \tag{1}$$

$[x, y] \in [0,1]x[0,1], \quad t \in [0, 0.16]$

The initial and boundary conditions are the following:

- $w(x, y, 0) = 0$,
- $w(0, y, t) = 1 - y^3$,
- $w(1, y, t) = 1 - sin(\pi/2 * y)$,
- $w(x, 0, t) = 1$,
- $w(x, 1, t) = 0$.

1 Discretize the equation by CDS scheme in space and the Crank-Nicolson method in time:

In numerical algorithms for differential equations the concern is the growth of round-off errors and/or initially small fluctuations in initial data which might cause a large deviation of final answers from the exact solution. The von Neuman stability analysis is based on the decomposition of numerical errors of numerical approximations into Fourier series. Fourier series decomposes any periodic function or periodic signal into the sum of a (possibly infinite) set of simple oscillating functions, namely sines and cosines (or, equivalently, complex exponentials). We have chosen complex exponentials to represent errors as they are much easier to work with than real trigonometric functions.[1]

$$\frac{w_{i,j}^{n+1} - w_{i,j}^{n}}{\Delta t} = \frac{1}{2}\frac{\left(w_{i+1,j}^{n+1} - 2w_{i,j}^{n+1} + w_{i-1,j}^{n+1}\right)}{\Delta x^2} + \frac{1}{2}\frac{\left(w_{i,j+1}^{n+1} - 2w_{i,j}^{n+1} + w_{i,j-1}^{n+1}\right)}{\Delta y^2} = \frac{1}{2}\frac{\left(w_{i+1,j}^{n} - 2w_{i,j}^{n} + w_{i-1,j}^{n}\right)}{\Delta x^2} + \frac{1}{2}\frac{\left(w_{i,j+1}^{n} - 2w_{i,j}^{n} + w_{i,j-1}^{n}\right)}{\Delta y^2} \tag{2}$$

1.1 Show under what conditions the Crank-Nicolson scheme is stable.

a) In general, we know that the Crank-Nicolson scheme is **unstable**. However, using the Von Neumann Stability Analysis, we can prove that the Crank-Nicolson scheme is stable.

Consistency and stability $\Longleftrightarrow$ convergence (Taylor expansion) (property of numerical scheme) Idea in von Neumann stability analysis: Study growth of waves e^{ikx}. (Similar to Fourier methods). So for the Heat equation, we will get:

$$w_t = D \cdot w_{xx}$$

Solution:
$w(x,t) = e^{-Dk^2 t} \cdot e^{ikx} = \underbrace{G(k)}_{\text{growth factor}}$

no growth if $|G(k)| \leq 1 \forall k$

$$w_t = w_{xx} + w_{yy}$$

Forward Euler:

$$\frac{w_{ij}^{n+1} - w_{ij}^{n}}{\Delta t} = \frac{w_{i+1,j}^{n} - 2w_{i,j}^{n} + w_{i-1,j}^{n}}{(\Delta x)^2} + \frac{w_{i,j+1}^{n} - 2w_{i,j}^{n} + w_{i,j-1}^{n}}{(\Delta y)^2}$$

$$w(x, y, t_n) = e^{i(k,l)\cdot\begin{pmatrix} x \\ y \end{pmatrix} = e^{ikx}} \cdot e^{ily}$$

$$\frac{G-1}{\Delta t} = \frac{e^{ik\Delta x} - 2 + e^{-ik\Delta x}}{(\Delta x)^2} + \frac{e^{il\Delta y} - 2 + e^{-il\Delta y}}{(\Delta y)^2}$$

$$\Rightarrow G = 1 - 2\frac{\Delta t}{(\Delta x)^2} \cdot (1 - \cos(k\Delta x)) - 2\frac{\Delta t}{(\Delta y)^2} \cdot (1 - \cos(l\Delta y)) \tag{3}$$

Worst case: $k\Delta x = \pi = l\Delta y \Rightarrow G = 1 - 4\frac{\Delta t}{(\Delta x)^2} - 4\frac{\Delta t}{(\Delta y)^2}$
Stability condition:

$$\frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2} \leq \frac{1}{2} \Leftrightarrow \Delta t \leq \frac{1}{2} \cdot \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right)^{-1} \underbrace{=}_{\text{if } \Delta x = h = \Delta y} \frac{h^2}{4} \tag{4}$$

Furthermore, as we mention above we can use the Von Neumann stability analysis to determine the stability of Crank-Nicolson scheme.
More specifically, in 2D the differential equation is represented as Fourier series, with the temperature: $w_{i+1,j}^{n+1}$ represented as:

$$\begin{aligned} \frac{w^{n+1}e^{ik\Delta x}e^{il\Delta y} - w^n e^{ik\Delta x}e^{il\Delta y}}{\Delta t} = \\ \frac{1}{2}\frac{\left(w^{n+1}e^{ik\Delta x+1}e^{il\Delta y} - 2w_{i,j}^{n+1}e^{ik\Delta x}e^{il\Delta y} + w_{i-1,j}^{n+1}e^{ik\Delta x-1}e^{il\Delta y}\right)}{\Delta x^2} + \\ \frac{1}{2}\frac{\left(w^{n+1}e^{ik\Delta x}e^{il\Delta y+1} - 2w_{i,j}^{n+1}e^{ik\Delta x}e^{il\Delta y} + w_{i-1,j}^{n+1}e^{ik\Delta x}e^{il\Delta y-1}\right)}{\Delta y^2} + \\ \frac{1}{2}\frac{\left(w^{n}e^{ik\Delta x+1}e^{il\Delta y} - 2w_{i,j}^{n}e^{ik\Delta x}e^{il\Delta y} + w_{i-1,j}^{n}e^{ik\Delta x-1}e^{il\Delta y}\right)}{\Delta x^2} + \\ \frac{1}{2}\frac{\left(w^{n}e^{ik\Delta x}e^{il\Delta y+1} - 2w_{i,j}^{n}e^{ik\Delta x}e^{il\Delta y} + w_{i-1,j}^{n}e^{ik\Delta x}e^{il\Delta y-1}\right)}{\Delta y^2} \end{aligned}$$

Dividing the whole equation with $w_{i,j}^n = w^n e^{ik\Delta x}e^{il\Delta y}$, we ended up in the Eq. (3). Therefore, as of Von Neumann stability analysis, the Crank Nicolson scheme is **unconditionally stable** as the amplification factor is always smaller or equal to 1.
b) To check for convergence, we need to check both the stability and consistency of the scheme.
We have already showed the stability in the a), so in this section we will prove the consistency of the scheme.
Moreover, the terms in the Crank Nicolson scheme equation can be represented in terms of their Taylor Series approximations.
So, we have:

$$\begin{aligned} \mathrm{w}_{i,j}^{n+1} &= \mathrm{w}_{i,j}^n + \left.\frac{\partial w}{\partial t}\right|_{i,j}^n \Delta t + \left.\frac{\partial^2 w}{\partial t^2}\right|_{i,j}^n \frac{\Delta t^2}{2} + H \\ \mathrm{w}_{i\pm1,j}^n &= \mathrm{w}_{i,j}^n \pm \left.\frac{\partial w}{\partial x}\right|_{i,j}^n \Delta x + \left.\frac{\partial^2 w}{\partial x^2}\right|_{i,j}^n \frac{\Delta x^2}{2} \pm \left.\frac{\partial^3 w}{\partial x^3}\right|_{i,j}^n \frac{\Delta x^3}{6} + H \\ \mathrm{w}_{i,j\pm1}^n &= \mathrm{w}_{i,j}^n \pm \left.\frac{\partial w}{\partial y}\right|_{i,j}^n \Delta y + \left.\frac{\partial^2 w}{\partial y^2}\right|_{i,j}^n \frac{\Delta y^2}{2} \pm \left.\frac{\partial^3 w}{\partial y^3}\right|_{i,j}^n \frac{\Delta y^3}{6} + H \end{aligned} \tag{5}$$

Substituting these terms into equation (2), we will obtain the truncation error term $O(\Delta^2) + O(\Delta x^2) + O(\Delta y^2)$ which is dependent to the on $\Delta t, \Delta x^2 \text{and} \Delta y^2$. As the grid is more refined, the truncation error term will also reach zero, which thus means that the scheme is consistent. As such, the scheme also converges as the grid is more refined.

2 Results of the Explicit Euler scheme

Since the Crank Nicolson scheme is unconditionally stable, we only need to obtain the explicit Euler scheme without matrix division to be required as the implicit term and it is just the result of the multiplications and additions of all known explicit terms.
So, we can say a stable Δt for a stable explicit Euler scheme would be for the given $h = 1/40$:

$$\Delta t \leq \frac{h^2}{4} = \frac{(\frac{1}{40})^2}{4} = 0.000156 \tag{6}$$

Figure 2 represents the temperature evolution at the node location $x = y = 0.4$ for the Euler explicit scheme computed at a stable Δt of $0.0001s$. We can notice from the Fig.2 that the temperature at this node exhibits an increase from $T = 0$ at $t = 0s$ to a maximum value of $T = 0.63879$ at $t = 0.16s$. At the same time, it is visible that the temperature at the specific node stays at a value of $T = 0$ for the first few seconds of the computation but after that it experiences a sudden increase. This is because, for the very first iterations, this node has no connecting nodes with temperature values different than zero since the initial temperature condition is set to zero except for the boundary conditions.
Figure 3 represents the vertical temperature profile of the Euler explicit scheme at $x = 0.4$ and at the final time step $t = 0.16s$. Here, also, we can notice that the temperature is higher at the node $(0.4, 0)$ than at $x - y$ coordinates and at the same time is equal to $T = 1$ which is at the boundary condition. However, the temperature decreases when is moving away from this boundary with increasing y until it reaches a value of $T = 0$ which is the node of the second boundary where T is set to 0.

3 Results of the Crank Nickolson scheme

3.1 Influence of the width of the time step

The Crank Nickolson scheme was also computed following the code in the code section. The time step of dt=0.0001, dt=0.001 and dt=0.01 were tested. As expected each simulation was stable since the Crank-Nickolson scheme is unconditionally stable (see section 1.1). However for dt=0.01 the solution was found to be

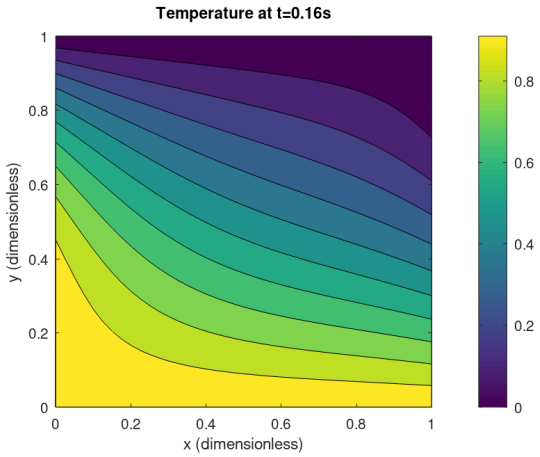


Figure 1: The contour plot of Explicit Euler

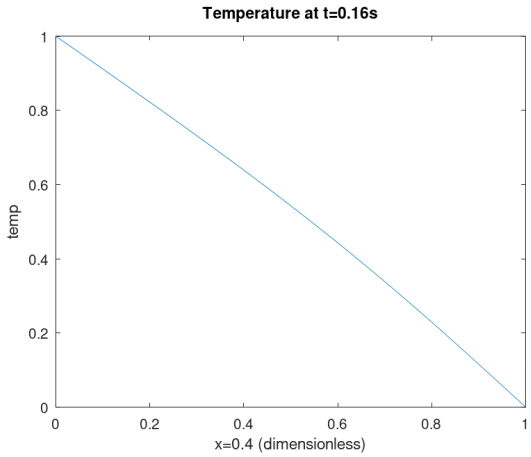


Figure 2: The Explicit Euler plot at x=0.4

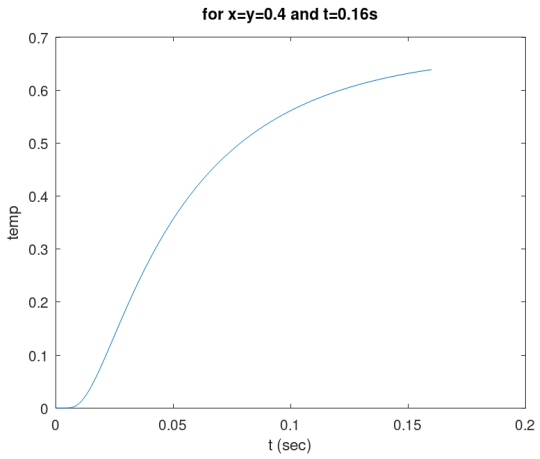


Figure 3: The Explicit Euler plot at x=y=0.4

different then for the two other time steps as shown in figure 4 compared to . So the next tests were made with dt = 0.001.

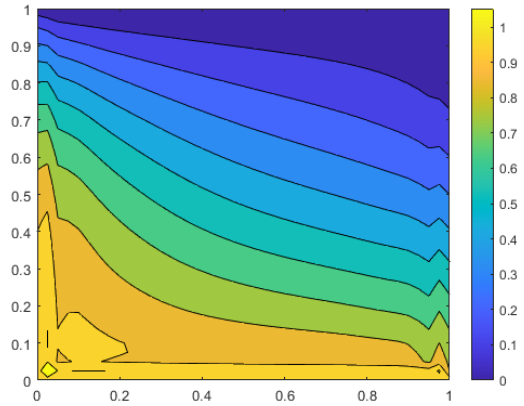


Figure 4: Temperature along the y axis for x=0.4 in the numerical solution of the equation (1) using the Crank-Nickolson scheme with dt=0.01

3.2 Change of the temperature in time at a specific point

As seen in figure 5 the temperature increases with time and seems to begin to stabilize around 0.65. That can be explained by the fact that the boundary conditions at y=1 and y=0 are respectively set to 0 and 1. We could however expect a linear diffusion of the temperature in space and therefore after a long enough time the temperature should stabilize at 0.4. The boundary conditions at x=0 and x=1 are nevertheless $1/y^3$ and $1-sin(n/2*y)$ respectively so for y=0.4 0.94 at the x=0 limit and 0.41 at the x=1 limit. Which can explain the bigger value observed.

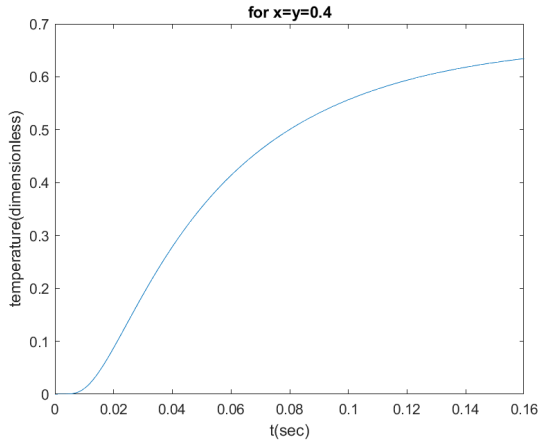


Figure 5: Evolution of the temperature with the time at a given point w=y=0.4, the chosen time step width id dt=0.001

3.3 Change of temperature at the final time step along the y axis.

Along the y axis the temperature decreases from 1 to 0 following the boundary conditions at y=0 und y=1 respecctively. We can also notice that the shape of the curve representing the decrease of temperature is located between the shape of the curves representing the boundary conditions at x=0 and x=1. (curve in red and orange on 6).

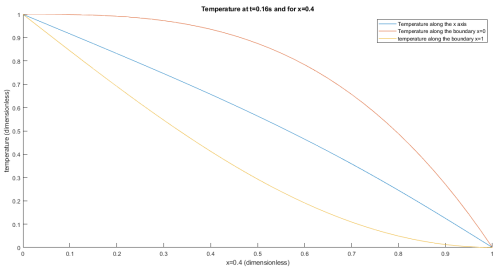


Figure 6: Evolution of the temperature along the y axis at the last frame(after 0.16s)

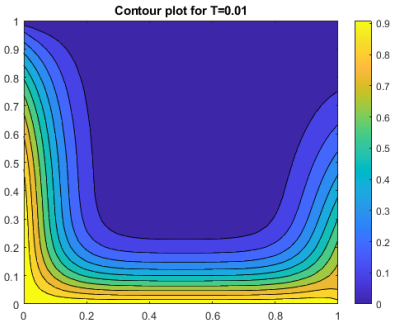
For the next part the Crank-Nickolson scheme with a dt=0.001 will be chosen over the euler scheme. Indeed we calculated that the euler scheme was unstable with a dt greater than dt=0.001 (equation (6)). However The Crank Nickolson scheme presents the same precision of results and for the same order of computational time with a dt=0.001.

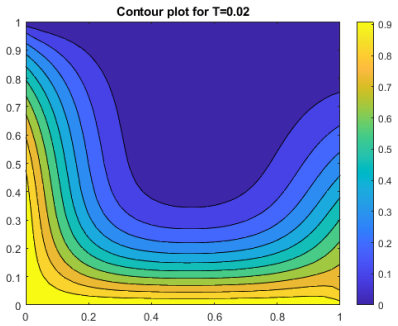
4 2D heat diffusion with time

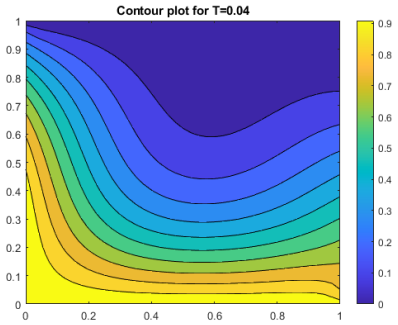
The heat diffusion is lead by the equation presented in the abstract. The darkest blue represents a temperature of 0 while the lightest yellow colour stands for a temperature of 1. The gradient of temperature seems to be the higher going from (x=0,y=0) to (x=1,y=1). It is important to notice that the equation was here chosen to be very global. Indeed depending on the material the 2D equation (1) has to be parametered with the heat diffusivity of the material. The temperature on a scale going from 0 to 1 is also very flexible and can represent any difference of temperature as long the boundary conditions have the same form has the conditions that are here described.

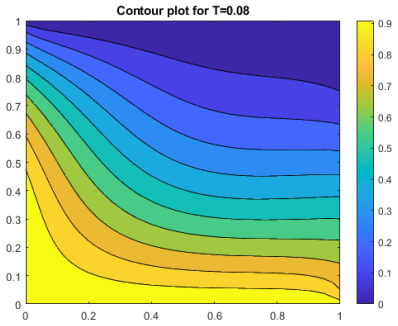
5 Conclusion

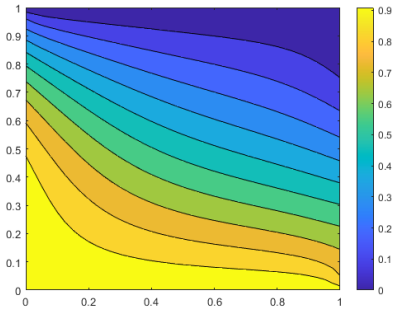
The heat diffusion case that was studied here is very global and lacks of physical meaning but has also the advantage to be very flexible and adaptable to a lot of different situation. The downside of Crank Nicolson is that the solution is more complex to solve where we have several variables which need to be solved in our matrix. On a computational point of view an explicit algorithm (Explicit Euler scheme) for computing time and space dependent equations was compared with an implicit program (Crank-Nickolson scheme). The Explicit euler was easier to program but can be unstable if the time steps are too big for the simulation. Meanwhile The Crank-Nickolson is always stable since it uses all the 4 neighbors on a grid to compute the point of interest for the next time step. Thus, it is critical to use very small time steps for Euler explicit compared to the Crank Nicolson scheme where a larger value of Δt can be used and still obtain the same result. This is due to the fact that Crank Nicolson uses temperature values at $n+1$ instead of only temperature values at n which dramatically increases the accurancy. The time of computation is however usually longer for the same number of time steps but the stability of the algorithm leads to the opportunity to chose lower time steps, so the opportunity to reduce the computational cost.
Finally, the temperature plot shows the physical phenomena of diffusion without convection and heat source/sink. As we know, the heat is being transferred from high to low temperature value nodes. Initially the highest value is at the lower boundary condition where the temperature is constant at $T = 1$ presumably due to constant heating at the bottom wall and the lowest value is at the upper boundary condition where the temperature is constant at $T = 0$ presumably due to constant cooling at the top wall. The right and left side boundaries are varying relative to the position of the node since they are expressed in terms of 2 different functions. In the beginning , the distribution of temperature is not uniform , with high temperature gradients especially near the bottom of the wall. As time passes, the temperature gradient gets smoother and more uniform throughout the space and the inner temperature value increases gradually from the initial value of 0.











6 Code

Explicit Euler

```
% The explicit Euler Method
clc
clear all
close all
% Parameters
T= 0.16;
dt=0.0001;
t= T/dt ;
dx=1/40;
dy=1/40;
Lx=1.0;
Ly=1.0;
Nx=(Lx/dx)+1;
Ny=(Ly/dy)+1;

% Space discretization
x = 0:dx:1;
y = 0:dy:1;

%Initial Condition
w=zeros(Nx,Ny,t); % we start with the initial temperature of
↪  zero everywhere

% Boundary conditions

for l=1:1:Ny % w(0, y, t) = 1-y3
  w(l,1,:) = 1-x(l)^3;
end

for r=1:1:Nx % w(1, y, t) = 1-sin(p/2 *y)
  w(r,Nx,:) = 1-sin(pi/2 * x(r));
end

w(1,:,:) = 1; % w(x, 0, t) = 1
w(Ny,:,:) = 0; % w(x, 1, t) = 0

% Discretization
for k=1:1:t; % time discretization
  k
  for i=2:1:Nx-1
    for j=2:1:Ny-1
      w(i,j,k+1) = w(i,j,k)+ dt
↪   *(((w(i-1,j,k)-2*w(i,j,k)+w(i+1,j,k))/dx^2)+
      ((w(i,j-1,k)-2*w(i,j,k)+w(i,j+1,k))/dy^2));
    endfor
  endfor
end

% Plotting
figure 1
w(:,:,1) % just to see the 1st iteration
contourf(x,y,w(:,:,t), 10)
colorbar
```

Crank Nicolson

```
% The 2D crank_Nickolson
clc
clear all
close all
% Parameters
T= 0.16;
dt=0.0001;
t= T/dt ;
dx=1/40;
dy=1/40;
Lx=1.0;
Ly=1.0;
Nx=(Lx/dx)+1;
Ny=(Ly/dy)+1;
Nxy=Nx*Ny;
alpha=dt/dx^2;
Result=zeros(Nx,Ny,t);

% Space discretization
x = 0:dx:1;
y = 0:dy:1;

%Initial Condition
w=zeros(Nxy,t); % we start with the initial temperature of
↪  zero everywhere

% Boundary conditions

for l=1:Nx:Nxy % w([1 1+Nx 1+2Nx... 1+Nx(Ny-1)], t) = 1-y3
  w(l,:) = 1-x((l-1)/Nx+1)^3;
end

for r=Nx:Nx:Nxy % w([Nx 2Nx... Nx*Ny)], t) = 1-sin(p/2 *y)
  w(r,:) = 1-sin(pi/2 * x(r/Nx));
end

for i=1:1:Nx
   w(i,:) = 1; % w(1->Nx, t) = 1
end
for j=0:1:Nx-1 %w(Nx(Ny-1)->Nx*Ny, t) = 1
   w(end-j,:) = 0;
end

%Crank-Nickolson penta diagonal matrix

diagonals = [2*(1+2*alpha)*ones(Nx*Ny,1),
↪  -alpha*ones(Nx*Ny,4)];
A=spdiags(diagonals,[0 -1 1 -Nx Nx],Nx*Ny,Nx*Ny);
I=speye(Nx*Ny);

%Replacement with identity lines into the matrix for the
↪  unchanging boundary
%conditions

for i=1:1:Ny
   A(i,:) = I(i,:);
   A(Nxy-Nx+i,:) = I(Nxy-Nx+i,:);
   A(i*Nx,:) = I(i*Nx,:);
   A(1+(i-1)*Nx,:) = I(1+(i-1)*Nx,:);
end
% In order to maintain the boundary conditions every
↪  element of boundary is facing the equivalent line of the
↪  identity
```

Crank Nicolson

```
% Discretization
for k=1:t
   b=ones(Nx,1);%boundary conditions (w(1->Nx, t) = 1)
   for i=1:1:Ny-2;

      %stocking the precedent results into the result matrix to
↪     avoid a
      %second double loop

      Result(i,:,k)=w((i-1)*Nx+1:i*Nx,k);
      Result(Ny-1,:,k)=w((Ny-2)*Nx+1:(Ny-1)*Nx,k);
      Result(Ny,:,k)=w((Ny-1)*Nx+1:Ny*Nx,k);
      %Even if we could directly implement this border in
↪     Result doing
      % it in the main loop is a good way to check for bugs
      leftside=[alpha*w(1+i*Nx:Nx*(i+1)-2,k) +
↪     (2-4*alpha)*w(2+i*Nx:Nx*(i+1)-1,k)+
↪     alpha*w(3+i*Nx:Nx*(i+1),k)+alpha*w(2+(i-1)*Nx:Nx*i-1,⌋
↪     k)+alpha*w(2+(i+1)*Nx:(i+2)*Nx-1,k)];
      b=[b;1-x(i)^3;leftside; sin(1-sin(pi/2 * x(i)))];
   end
   b=[b;zeros(Nx,1)];%boundary conditions
↪     w(Nx(Ny-1)->Nx*Ny, t) = 1
   w(1:Nx*Ny,k+1)=A\b;
end
%Stocking the value of the last iteration
for i=1:1:Ny
   Result(i,:,t)=w((i-1)*Nx+1:i*Nx,t);
end
% Plotting the temperature surface at the final time

figure (1)
contourf(x,y,Result(:,:,t), 10)
w(1:Nxy,2)
colorbar

%Plotting the time evolution of the temperature at x=y=0.4

 v(:)=w((0.4/dx+1)*Nx+0.4/dx,:);
 timevector=[0:dt:T]
 figure (2)
 plot(timevector,v)
 xlabel('t(sec)')
 ylabel('temperature(dimensionless)')
 title('for x=y=0.4')
 saveas(4,"x=y=0.4 dt= 0.01 .png")

%Plotting the vertical temperature profile at t=0.16s and
↪  x=0.4
z(:)=w(Nx*0.4/dx+1:(Nx)*0.4/dx+Nx,0.16/dt);
figure (3)
plot(x,z)
xlabel('x=0.4 (dimensionless)')
ylabel('temperature (dimensionless)')
title('Temperature at t=0.16s and for x=0.4')
saveas(5,'x=0.4 and t=0.16s dt=0.01.png')
```

References

[1] Von neumann stability analysis - matlab uprm). November 11, 2022.