NUMERICAL SIMULATION OF PARTIAL DIFFERENTIAL EQUATIONS

VINCENT PEETERS AND MORITZ WOLTER

Project Report

Supervised by Stefan Vandewalle
Stefaan Poedts
December 2, 2014 – Version 1

1 HEAT EQUATION

In the first part of this report we are going to solve the heat equation:

$$\frac{\partial \phi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} \tag{1}$$

With the boundary conditions:

$$\phi(0,t) = 0,\ \phi(1,t) = 1,\ \phi(x,0) = \sin(5\pi x/2).$$

Using four different numerical solution schemes:

1.1 EXPLICIT EULER

The explicit Euler method is defined as:

$$U_j^{n+1} = U_j^n + \mu(U_{j+1}^n - 2U_j^n + U_{j-1}^n) \ \text{ with } \mu = \frac{\triangle t}{(\triangle x)^2}. \tag{2}$$

From this equation we can derive code that solves the heat equation:

```
mu = 0.3;
dX = 1/20;
dT = mu * dX^2;
%First plot:
t = 0:dT:0.5;
%second plot.
%t = 0:dT:0.05;
x = 0:dX:1;

U = zeros(length(t),length(x));
%boundary conditions:
U(:,1) = 0;
U(:,end) = 1;
U(1,:) = sin(5*pi*x/2);

%compute the solution:
for n = 1:1:(length(t)-1)
    for j = 2:1:(length(x)-1)
        U(n+1,j) = U(n,j) + mu*( U(n,j+1) - 2*U(n,j) + U(n,j-1));
    end
end
mesh(x,t,U)
```

Listing 1: explicit Euler

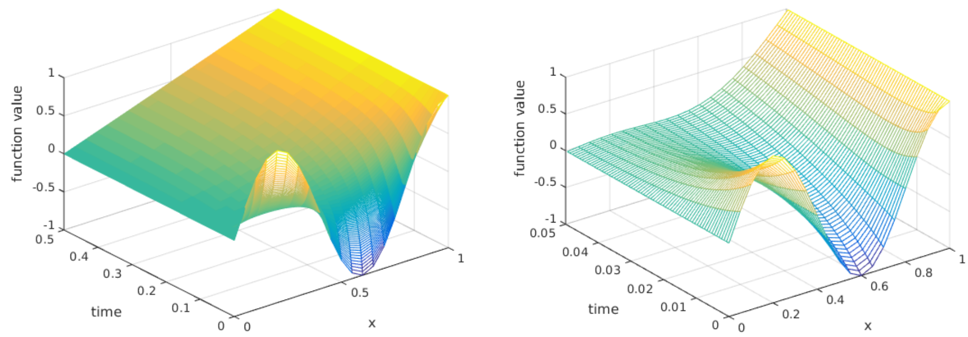


Figure 1: Solution of the heat equation with the explicid Euler method. Until time $t = 0.5$ (left) and until $t = 0.05$ (right). The boundary conditions are: 0, 1, $\sin(5\pi x/2)$.

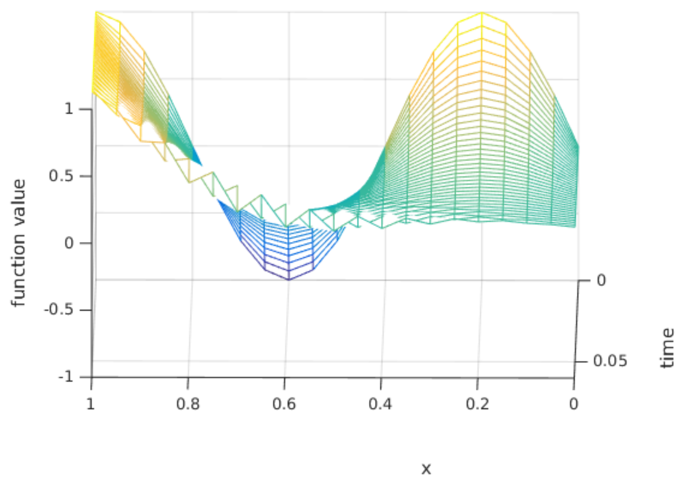


Figure 2: Instabilities forming with $\mu = 0.55$ at time $t = 0.05$.

Running this code leads to the images in figure 1. The computations are done using a mesh ratio $\mu = 0.3$ and $\triangle x = \frac{1}{20}$. Therefore we have time steps of size $\triangle t = 0.00075 = 7.5 * 10^{-4}$. This scheme is stable for mesh ratios $\mu \leqslant 0.5$. Therefore if we increase the time step to ≈ 0.0013 we are expecting to see instability. A plot of forming instabilities is given in figure 2

1.2 A SLIGHT VARIATION OF THE PROBLEM

Next we are going to consider a small variation of the problem. In fact the boundary conditions are going to change to:

$$\phi(0,t) = 0, \ \phi(1,t) = 0, \ \phi(x,0) = \sin(\pi x).$$

For this set of boundary conditions we know the exact solution:

$$\pi(x,t) = \exp(-\pi^2 t)\sin(\pi x). \tag{3}$$

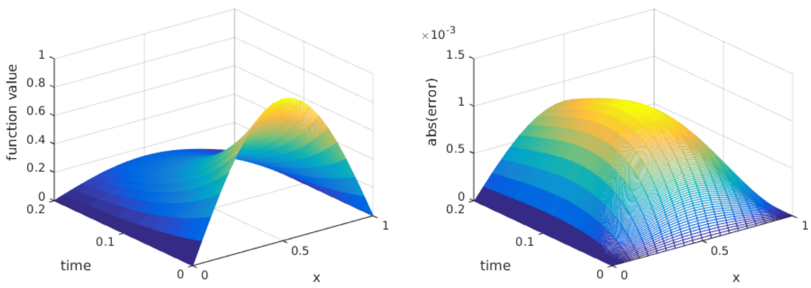


Figure 3: Numerical solution of the heat equation with the second boundary value set (left). Absolute value of the numerical solution (right).

Knowledge of the exact solution enables us to check the code we provided earlier. We are now able to compute the error in every grid point. The solution should not deviate too much from the exact solution. A plot of the numerical solution and it's error is given in figure 3. As the biggest error in any grid point is equal to 0.0011 we conclude our implementation is probably correct.

TODO: more math here...

1.3 EULER, CRANK-NICOLSON AND THE θ-METHOD

In this section we are going to use the more general theta-method-scheme to compare the errors of the explicit-Euler, implicit-Euler and Crank-Nicolson methods. Θ-type methods are defined as :

∂_x^2 denotes double application of a central difference

$$U_j^{n+1} - U_j^n = \mu[\theta\partial_x^2 U_j^{n+1} + (1-\theta)\partial_x^2 U_j^n]. \tag{4}$$

With $\theta = 0$ this we have the explicit Euler method, $\theta = 1$ leads to the implicit Euler method and finally $\theta = 0.5$ leads to the Crank-Nicolson method. In order do be able to implement a function in matlab that takes care of finding the error for each of these methods we have to derive two essential matrices. This is done by expanding the central differences from equation 4 and rearranging:

$$-\mu\theta U_{j-1}^{n+1} + U_j^{n+1}(1+2\mu\theta) - \mu\theta U_{j+1}^{n+1}$$

$$= \mu(1-\theta)U_{j-1}^n + (1-2\mu(1-\theta)U_j^n + \mu(1-\theta)U_{j+1}^n. \tag{5}$$

Here we are looking at an equation of the form $A\mathbf{U^{n+1}} = B\mathbf{U^n}$. Therefor the matrices A and B must be:

$$A = \begin{pmatrix} (1+2\mu\theta) & -\mu\theta & & \\ -\mu\theta & (1+2\mu\theta) & -\mu\theta & \\ & -\mu\theta & (1+2\mu\theta) & -\mu\theta \\ & \ddots & \ddots & \ddots \end{pmatrix}$$

$$B = \begin{pmatrix} 1-2\mu(1-\theta) & \mu(1-\theta) & & \\ \mu(1-\theta) & 1-2\mu(1-\theta) & \mu(1-\theta) & \\ & \mu(1-\theta) & 1-2\mu(1-\theta) & \mu(1-\theta) \\ & \ddots & \ddots & \ddots \end{pmatrix}. \quad (6)$$

Now we are able to write the following function, which when executed with different thetas θ and grid parameters $\triangle x, \triangle t$ will allow us to learn more about the error.

```
function error  = thetaMethod( dT,dX,tEnd,xEnd,leftBound,...
                                 rightBound,xBound,theta)
mu = dT/(dX^2);
t = 0:dT:tEnd;
x = 0:dX:xEnd;
U = meshgrid(x,t);
exSol = zeros(length(t),length(x));
error = zeros(length(t),length(x));
%Time boundary conditions.
U(:,1) = leftBound;
U(:,end) = rightBound;
%x boundary condition
U(1,:) = xBound;

%construct the step Matrix:
leftMat = toeplitz([(1 + 2*mu*theta) -mu*theta, ...
                    zeros(1,length(x)-4)],...
                   [(1 + 2*mu*theta) -mu*theta,...
                    zeros(1,length(x)-4)]);

rightMat = toeplitz([(1 - 2*mu*(1-theta)) mu*(1-theta), ...
                     zeros(1,length(x)-4)],...
                    [(1 - 2*mu*(1-theta)) mu*(1-theta), ...
                     zeros(1,length(x)-4)]);

for n = 1:1:(length(t)-1)
    %compute the values for the next time step.
    U(n+1,2:(end-1)) = leftMat\(rightMat*U(n,2:end-1)')...
                       + mu*[U(n+1,1); zeros(length(x)-4,1); ...
                         U(n+1,end)];
    %Exact solution:
    exSol(n,:) = exp(-pi^2*t(n))*sin(pi*x);
    %Error:
    error(n,:) = U(n,:) - exSol(n,:);
end
end
```

Listing 2: generic theta Method

In the following section we will describe and interpret the results we obtained.

	△x =1/20	△x =1/40	△x =1/80	△x =1/160	△x =1/320
△t =1/10	0.8919	0.6577	0.4729	0.3368	0.2390
△t =1/20	0.1427	0.1119	0.0817	0.0584	0.0415
△t =1/40	0.0196	0.0214	0.0166	0.0121	0.0086
△t =1/80	0.0047	0.0032	0.0035	0.0027	0.0019

Table 1: Error values of the Crank-Nicolson-scheme for different grid values. **Every entry has to be multiplied by 1.0e-03**.

	△x =1/20	△x =1/40	△x =1/80	△x =1/160	△x =1/320
△t =1/10	0.0068	0.0049	0.0035	0.0025	0.0018
△t =1/20	0.0023	0.0017	0.0012	0.0008	0.0006
△t =1/40	0.0009	0.0007	0.0005	0.0003	0.0002
△t =1/80	0.0004	0.0003	0.0002	0.0001	0.0001

Table 2: Error of the implicit Euler-scheme for different grid values.

1.3.1 *Results*

Table 1 and 2 show the results we obtained by running the code in listing 2 with different input values. Figure 4 shows a graphical representation of the results. The smaller the steps we take the smaller is the error we observe.

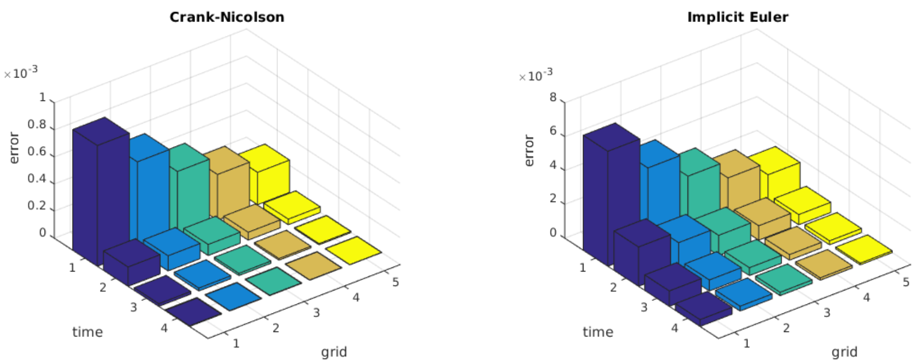
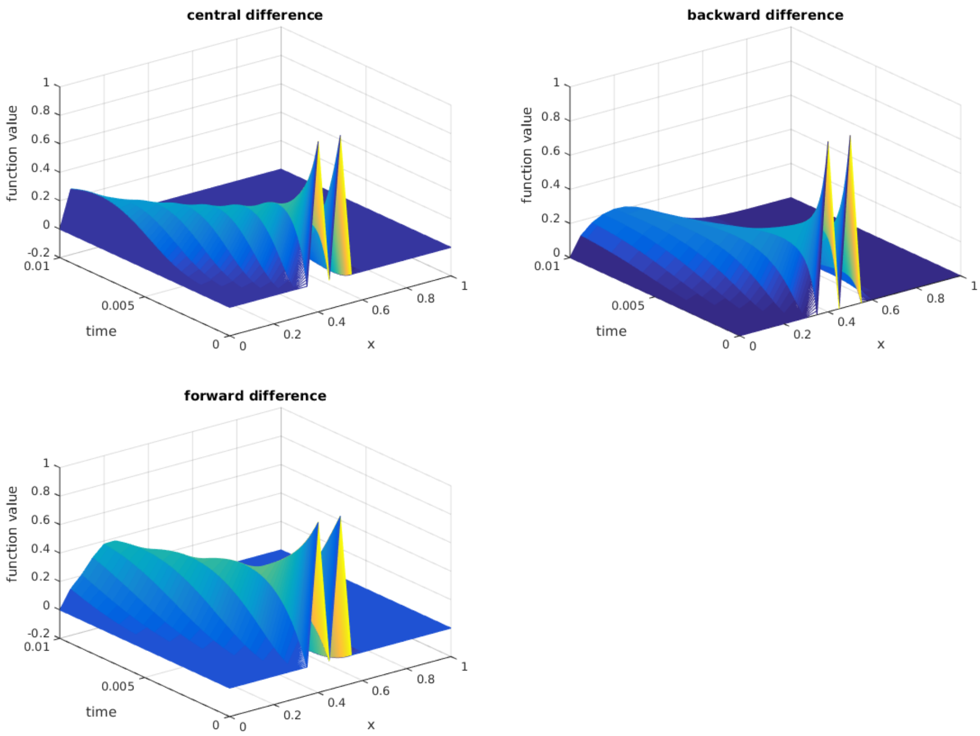


Figure 4: Bar plots of the results shown in table 1 (left) and 2 (right).

2 HEAT EQUATION WITH CONVECTION

In this part a convection term in introduced into the equation. We are now solving a different problem of the form:

$$\frac{\partial \phi}{\partial t} = 50\frac{\partial \phi}{\partial x} + \frac{\partial^2 \phi}{\partial x^2} \tag{7}$$


central difference
backward difference
forward difference
function value
time
x