Applied Numerical Methods - Lab 3

GOYENS Florentin & WEICKER David

13th October 2015

Stationary heat conduction in 1-D

In a one dimensional pipe we are interested in the temperature evolution along the z-axis. We will study the behaviour of the numerical solution based on finite differences.

1) Distretization to a linear system of equations

The z-axis is first discretized in $N + 1$ points spreading from 0 to L. The differential equation for the temperature is the following

$$-\kappa \frac{d^2T}{dz^2} + v\rho C \frac{dT}{dz} = Q(z)$$

with

$$Q(z) = \begin{cases} 0 & if \quad 0 \leq z < a \\ Q_0 \sin\left(\frac{z-a}{b-a}\pi\right) & if \quad a \leq z \leq b \\ 0 & if \quad b \leq z \leq L. \end{cases}$$

We assumed that κ as a constant value through the pipe.

The boundary conditions are

$$T(0) = T_0$$

and

$$-\kappa \frac{dT}{dz}(L) = k_v(T(L) - T_{out}).$$

DG: Discretize the interval to a Grid For the discretization, let $T_0, T_1, \ldots, T_N$ be the unknown temperature we will solve for. The step is $h = L/N$. Define $z_i = i * h$ and $T_i \approx T(z_i)$. It is clear that $z_0 = 0$ and $z_N = L$. We also define the ghost point $z_{N+1} = L + h$ that will be used implicitly to write the system. The real unknowns we will include in the system are $T_1, \ldots, T_N$. Obviously T_0 is known from the start. We therefore need N linearly independent equations.

The figure 1 illustrates the discretization and the usage of ghost points depending on the boundary conditions.

Figure 1: Discretization and ghost points.

DD: Discretize the differential equation With finite difference we discretize the equation as

$$-\kappa \frac{T_{i+1} - 2T_i + T_{i-1}}{h^2} + v\rho C \frac{T_{i+1} - T_{i-1}}{2h} = Q(z_i),$$

for all the points, $i = 1, \ldots N$. This is equivalent to

$$\underbrace{\Big(-\frac{v\rho Ch}{2} - \kappa\Big)}_{\alpha} T_{i-1} + \underbrace{\Big(2k\Big)}_{\beta} T_{i-1} + \underbrace{\Big(\frac{v\rho Ch}{2} - \kappa\Big)}_{\gamma} T_{i+1} = h^2 Q(z_i),$$

The previous are N linear equations that feature the $N + 2$ variables $T_0, T_1, \ldots, T_{N+1}$.

DB: Discretize the Boundary conditions The first boundary condition gives the straightforward $T_0 = 400$ which removes T_0 from the problem.

We also have

$$-\kappa \frac{T_{N+1} - T_{N-1}}{h} = k_v(T_N - T_{out}).$$

This allows to express T_{N+1} with other variables and remove it from the system of equations. We will then have N unknowns in our system of N difference equations.

$$T_{N+1} = T_{N-1} - \underbrace{\frac{2hk_v}{k}}_{\delta} T_N + \underbrace{\frac{2hk_v}{k}}_{\delta} T_{out}. \tag{1}$$

The final difference equation is the only one where T_{N+1} appears. That is

$$\alpha T_{N-1} + \beta T_N + \gamma T_{N-1} = h^2 Q(z_N).$$

We remove T_{N+1} using equation 1 and this yields

$$(\alpha + \gamma) T_{N-1} + (\beta - \gamma\delta) T_N + \gamma\delta T_{out} = h^2 Q(z_N).$$

We now have a linear system of N equations. This system is tridiagonal and the Matlab *band solver* will be very efficient is the matrix is defined as sparse. The system in our Matlab code can be visualized as follows.

$$\begin{pmatrix} \beta & \gamma & & & & \\ \alpha & \beta & \gamma & & & \\ & \alpha & \ddots & \ddots & & \\ & & \ddots & & & \\ & & & \alpha & \beta & \gamma \\ & & & & (\alpha+\gamma) & (\beta-\gamma\delta) \end{pmatrix} * \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{pmatrix} + \begin{pmatrix} \alpha T_0 \\ 0 \\ \vdots \\ 0 \\ \gamma\delta T_{out} \end{pmatrix} = h^2 \begin{pmatrix} Q(z_1) \\ \vdots \\ Q(z_N) \end{pmatrix}$$

2) Convergence of solution without convection

We now set $v = 0$ and solve the equation for increasing values of N. Results on figure 2.

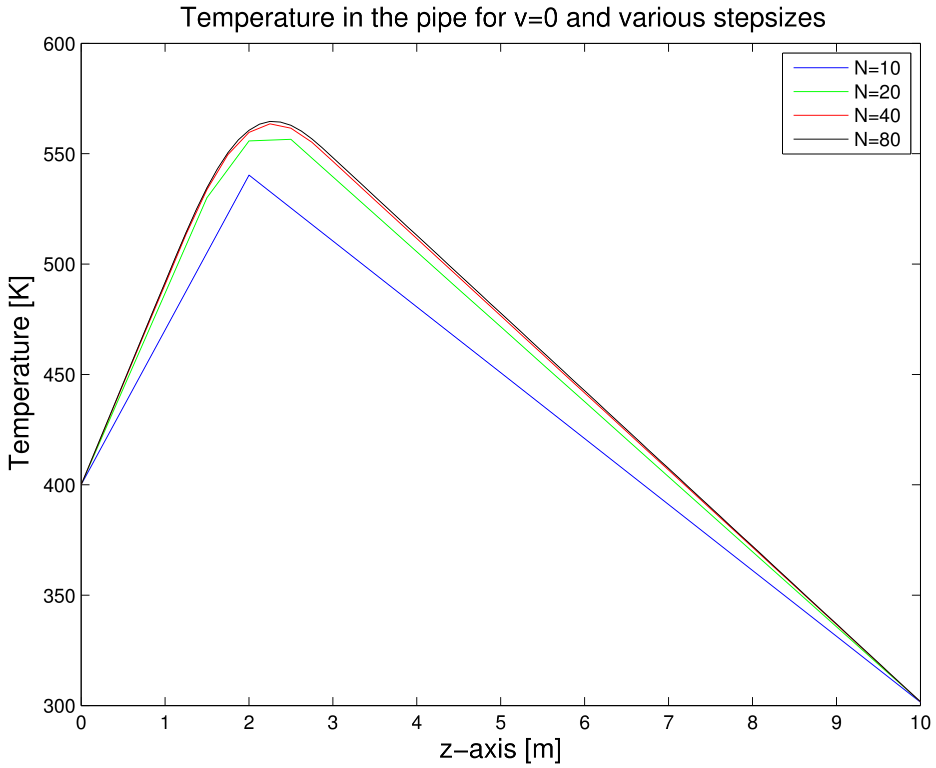

Figure 2: Convergence of solution without convection.

We see that increasing N has a great effect on the quality of the solution. It seems to converge towards a smooth curve.

The solution looks like a straight line on the intervals $[0, 1]$ and $[3, 10]$. This is to be expected. Since $v = 0$ and on these intervals the function Q is zero, the equation simplifies to

$$\frac{d^2T}{dz^2} = 0$$

whose solution is a straight line.

We also check that the convexity is correct. Since Q is non-negative, and again $v = 0$, the solution must be concave. Indeed, second derivative is non-positive thanks to the equation.

3) Increasing speed for $N = 40$

Let us set $N = 40$. We will now have v vary and take the values $0.1, 0.5, 1, 10$. Results on figure 3.

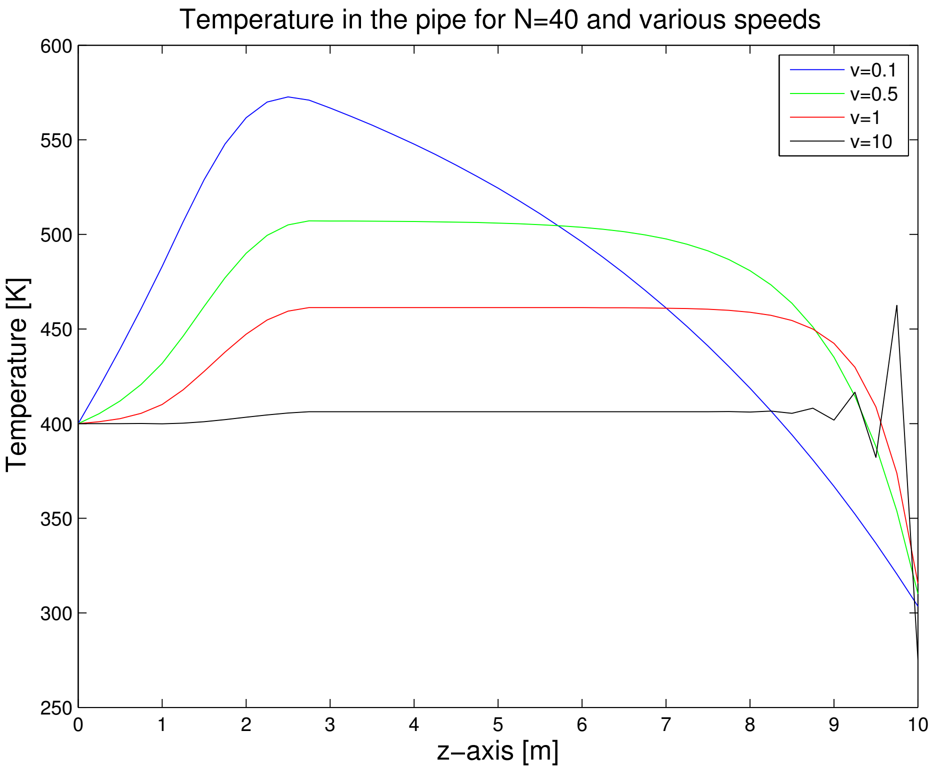

Figure 3: Solution with increasing speed v.

The maximal temperature decreases as v increases. This makes sense physically since the fluid in the pipe has less time to heat up. For high speeds like $v = 10$, the temperature barely changes as the fluid stays a very short time in the pipe.

We clearly notice that oscillations occur when $v = 10$. The slope of the solution near the end of the rope increases with v. This seems to create numerical instabilities for large v.

As we want to see better how the oscillations behave, we now reduce the precision with N for v fixed at 10. Results on figure 4.

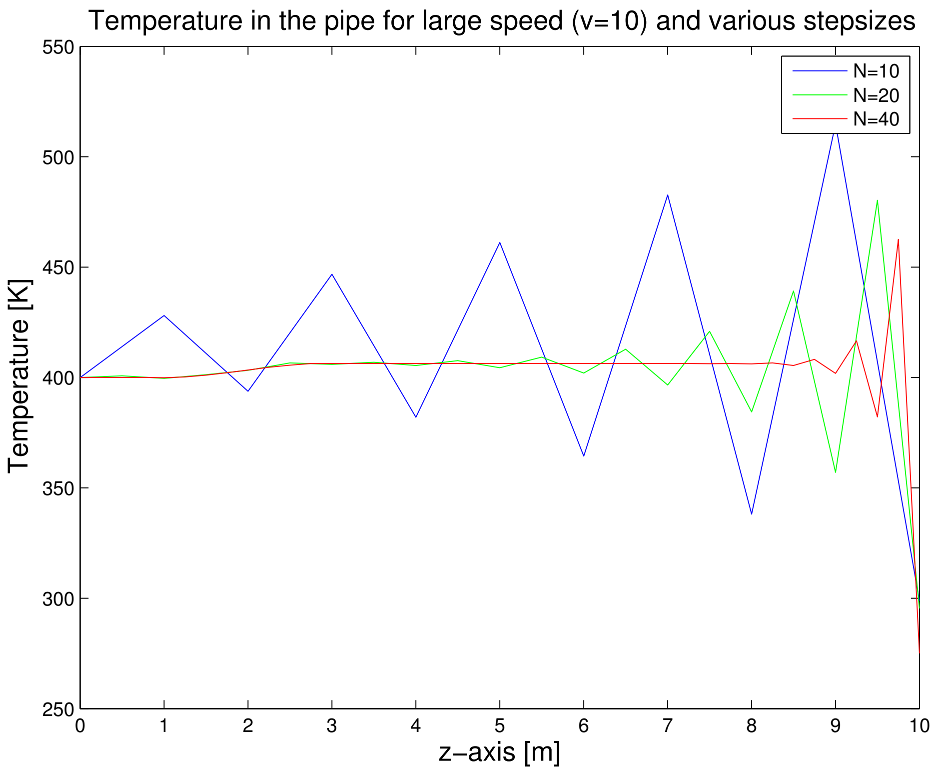

Figure 4: Capture the oscillation with increasing N.

As the level of the discretization decreases with N, we see more and more oscillations. At first ($N = 40$) the instabilities are located at the right end of the pipe. As N decreases, the instabilities propagate towards the left end of the pipe.

Here is the Matlab code used to produce the figures needed for this rapport.

```
function [] = code3()
% Function for LAB3
% Authors: David Weicker and Florentin Goyens
close all;

%no speed and varible N
N = [10 20 40 80];
v = 0;
T = {0,0,0,0};
z = {0,0,0,0};
for i = 1:4
    [T{i},z{i}] = temp(N(i),v);
end
figure;
plot(z{1},T{1},'b',z{2},T{2},'g',z{3},T{3},'r',z{4},T{4},'k');title('Temperature in ↩
    the pipe for v=0 and various stepsizes','FontSize',14)
xlabel('z-axis [m]','FontSize',14);ylabel('Temperature [K]','FontSize',14);legend('N↩
    =10','N=20','N=40','N=80');

%fixed N=40 and variable speed
N = 40;
v = [0.1 0.5 1 10];
for i = 1:4
    [T{i},z{i}] = temp(N,v(i));
```

```
end
figure;
plot(z{1},T{1},'b',z{2},T{2},'g',z{3},T{3},'r',z{4},T{4},'k');title('Temperature in ↩
    the pipe for N=40 and various speeds','FontSize',14);
xlabel('z-axis [m]','FontSize',14);ylabel('Temperature [K]','FontSize',14);legend('v↩
    =0.1','v=0.5','v=1','v=10');

%large v=10 and variable N
N = [10 20 40];
v = 10;
for i = 1:3
    [T{i},z{i}] = temp(N(i),v);
end
figure;
plot(z{1},T{1},'b',z{2},T{2},'g',z{3},T{3},'r');title('Temperature in the pipe for ↩
    large speed (v=10) and various stepsizes','FontSize',14);
xlabel('z-axis [m]','FontSize',14);ylabel('Temperature [K]','FontSize',14);legend('N↩
    =10','N=20','N=40');

end

function q = Q(z)
% Right-hand side of the system
% z is the discretized axis
[a,b,~,~,~,~,~,~,Q0,~,~,~]=param();

q = (a<=z) & (z<=b);
q = Q0*q.*sin(pi*(z-a)/(b-a));
end

function [T,z] = temp(N,v)
%Solve the problem for N+1 points between 0 and 10 and for speed v
[~,~,L,k,kv,rho,C,~,T0,Tout,~]=param();

h = L/N;
z = linspace(0,L,N+1);
q = Q(z(2:end))*h*h;

alpha = -k-v*rho*C*h/2; %tridiagonal terms
beta = 2*k;
gamma = -k+v*rho*C*h/2;

delta = 2*h*kv/k; %boundary conditions
q(1) = q(1) - alpha*T0;
q(end) = q(end) - gamma*delta*Tout;

e = ones(N,1);
A = spdiags([alpha*e beta*e gamma*e],-1:1,N,N); %sparse matrix
A(N,N-1) = A(N,N-1) + gamma;
A(N,N) = A(N,N) - gamma*delta;

T = T0*ones(N+1,1);
T(2:end) = A\q';
end

function [a,b,L,k,kv,rho,C,Q0,T0,Tout,v] = param()
%Initialize all the parameters for the model
L=10;
k=0.5;
kv=10;
v=0;
rho=1;
C=1;
a=1;
b=3;
Q0=50;
T0=400;
Tout=300;
end
```