Applied Numerical Methods - Lab 3

GOYENS Florentin & WEICKER David

13th October 2015

0.1 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.

Distretization to a linear system of equations

The z-axis is discretization 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

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

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 $x_N = L$. We also define the ghost point $x_{N+1} = L + h$ that will be used implicitly to write the system. The real unknowns we will include in the system are $T_1, \ldots, T_N$. T_0 is known from the start.

With finite difference approximation we rewrite the problem 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

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

As for the boundary condition we have the straightforward

$$T_0 = 400,$$

as well as

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

This allows to express T_{N+1} with other variables and remove it from the system of equations.

This is a linear system of $N + 1$ equations. As we would like to avoid dividing by the small value h^2, we rewrite the system as follows. This system is tridiagonal and the Matlab *band solver* will be very (very) efficient is the matrix is defined as sparse.