MAT-202E NUMERICAL METHODS
HOMEWORK 5

Dr. Onur Tuncer

December 30, 2008

Due date for source code submission via e-mail: 12.01.2009 @ 12 pm
Due date for the submission of the report: 13.01.2009

You need to solve the transient (i.e. time dependent) temperature distribution for a rod made of AISI 1040 steel with a length of $L = 1\ m$. The rod is initially at a uniform temperature of $0°C$, suddenly it is exposed to a temperature of $100°C$ on both ends. Thermal properties of AISI 1040 steel are as follows; thermal conductivity $k = 51.9\ W/m.K$, specific heat capacity $c_p = 486\ J/kg.K$ and density $\rho = 7845\ kg/m^3$.
Consider the below governing equation for a transient one-dimensional heat conduction problem.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2},\ 0 \leq x \leq L,\ t \geq 0$$

where $T(x,t)$ is the dependent variable, and α is a constant coefficient. Indeed $\alpha = k/(\rho c_p)$ is the thermal diffusivity of the material. Boundary conditions are $T(x,0) = 0$ (initial condition), $T(0,t>0) = 100$ (at the left boundary), and similarly $T(L,t>0) = 100$ (at the right boundary). In order to compute the numerical solution the governing PDE needs to be discretized in both time and space coordinates.Grid points can be uniformly distributed such that the grid point coordinates can be given with,

$$x_i = (i-1)\Delta x,\ i = 1,2,\cdots,N$$

where N is the number of spatial grid points, including ones on the boundary.Given L and N grid point spacing Δx can be computed as,

$$\Delta x = \frac{L}{N-1}$$

Similarly time t coordinate can also discretized with uniform intervals Δt from $0 < t < t_{final}$ such that,

$$t_m = (m-1)\Delta t,\ m = 1,2,\cdots,M$$

where K is the total number of time steps and Δt is the size of time step.

$$\Delta t = \frac{t_{final}}{M-1}$$

Approximations for the governing PDE can be obtained by replacing the continious derivatives with their finite difference approximations. If we use a first order backward time derivative combined with a centered approach for the spatial second derivative, we obtain an implicit algebraic system. Note

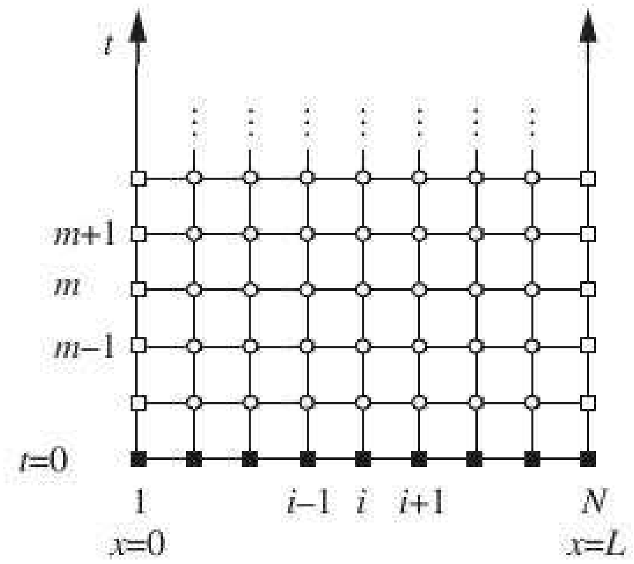

Figure 1: Mesh Used for the Solution of Transient One Dimensional Heat Conduction Equation. The solid squares incdicate the location of the (known) initial condition. The open squares indicate the locatin of known boundary values, whereas the open circles indicate the interior nodes for which the finite difference approximation is computed

that this particular fully implicit discretization is unconditionally stable.On the other hand to obtain the solution a tri-diagonal system has to be inverted at each time step. This can be efficiently done with the Thomas algorithm.

$$\frac{T_i^m - T_i^{m-1}}{\Delta t} = \alpha \frac{T_{i-1}^m - 2T_i^m + T_{i+1}^m}{\Delta x^2} + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$$

The above equation is valid at the internal nodes ($i = 2, 3, \cdots, N-1$). In order to see the system of equations more clearly we drop the truncation terms and re-arrange as follows,

$$-\frac{\alpha}{\Delta x^2} T_{i-1}^m + (\frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2}) T_i^m - \frac{\alpha}{\Delta x^2} T_{i+1}^m = \frac{1}{\Delta t} T_i^{m-1}$$

This system can be put in matrix form,

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} T_1^m \\ T_2^m \\ T_3^m \\ \vdots \\ T_{N-1}^m \\ T_N^m \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-1} \\ d_N \end{bmatrix}$$

where the coefficients for the interior nodes are,

$$\begin{aligned} a_i &= -\alpha/\Delta x^2, \ i = 2, 3, \cdots, N-1 \\ b_i &= (1/\Delta t) + (2\alpha/\Delta x^2), \\ c_i &= -\alpha/\Delta x^2, \\ d_i &= (1/\Delta t) T_i^{m-1} \end{aligned}$$

In order to enforce boundary conditions we need,

$$
\begin{aligned}
b_1 &= 1,\ c_1 = 0,\ d_1 = T(0,t) \\
a_N &= 0,\ b_N = 1,\ d_N = T(L,t)
\end{aligned}
$$

Tri-Diagonal Systems of Equations

Tri-diagonal systems are a special case of systems of linear equations which commonly arise in the solution of many of the engineering problems. Ordinary matrix inversion for a tri-diagonal system with n unknowns requires $\mathcal{O}(n^3)$ operations be made by the computer. Therefore such an approach is unnecessarily computationally intensive. Very efficient methods for solving tri-diagonal systems exist one of them being the so-called "Thomas Algorithm" which only requires $\mathcal{O}(n)$ operations. A tri-diagonal system can be written as,

$$
a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, with, a_1 = 0, c_n = 0 \tag{1}
$$

This system can be cast into matrix $A\mathbf{x} = \mathbf{b}$ form with,

$$
A = \begin{bmatrix} b_1 & c_1 & 0\cdots & 0 & 0 \\ a_2 & b_2 & c_2\cdots & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & a_n & b_n \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \tag{2}
$$

Note that in matrix A all entries are zero except the ones in the diagonal, the super-diagonal and the sub-diagonal. The name tri-diagonal refers to this property of the coefficient matrix A. The solution of this system is performed in two steps as per the Thomas Algorithm. First step involves modifying the coefficient vectors as in (3) and (4).

$$
c'_i = \begin{cases} \frac{c_1}{b_1} & i = 1 \\ \frac{c_i}{b_i - c'_{i-1} a_i} & i = 2, 3, \cdots, n-1 \end{cases} \tag{3}
$$

$$
d'_i = \begin{cases} \frac{d_1}{b_1} & i = 1 \\ \frac{d_i - d'_{i-1} a_i}{b_i - c'_{i-1} a_i} & i = 2, 3, \cdots, n-1 \end{cases} \tag{4}
$$

After the new coefficients are obtained solution is reached through back substitution (5).

$$
x_i = \begin{cases} d'_n & i = n \\ d'_i - c'_i x_{i+1} & i = n-1, n-2, \cdots, 1 \end{cases} \tag{5}
$$

a. Find the analytical solution (Hint: You can use seperation of variables technique to obtain the analytical solution by assuming $T(x,t) = \Theta(t)X(x)$. You will use this exact solution to check your numerical results.)

b. Write a subroutine named *analytical* that computes the analytical solution.

c. Write a subroutine named *Thomas* that implements the Thomas algorithm to invert a tri-diagonal system of equations.

d. Choose a suitable time step Δt and grid spacing Δx. What kind of judgement you use for choosing these values? State your thoughts in the Discussion section of your report.

e. Compute the numerical solution in time and space until the transient dies (i.e. until steady-state solution). At each time step output your results into a file for later plotting. Also write the analytical solution into a seperate file.

f. Plot your numerical results (temperature versus x-coordinate at different time steps). Plot all time steps on a single graph in order to reduce clutter.

g. Compare your numerical results with the analytical results. Comment on the accuracy of your numerical solution.

Homework Rules

You should present your own work during the course. Any unethical behaviour shall be penalized seriously.
You will post your homework in a zip file named as 'HW#-ID#.zip' to the address tuncero@itu.edu.tr
The zip file should include the necessary source codes and executables (m-files, C or FORTRAN codes, etc.)

Homework Format

Homework should consist of the following parts. Problem Definition and Discussion and Conclusion parts should be typed up using a word processor software. Remember neatness counts.

i. Cover Page (The cover page should state the name and ID number of the student clearly, cover page also needs to include the course name and homework number) In addition, cover page must also include the following statement at the bottom;
"I have neither given nor received any unauthorized help for this assignment."

Your Signature

ii. Problem Definition

iii. Results (In Numerical or Graphical Format)

iv. Discussion

v. Appendix (Source Code, etc.)

Source code should include the necessary comments and results should be presented with at least 5 digits.