ENGR 695   Advanced Topics in Engineering Mathematics   Fall 2023

Lab #11: The Crank-Nicholson Algorithm with Neumann Boundary Conditions

<u>Introduction</u>

In this lab session, you will examine the solution of the basic one-dimensional heat equation with Neumann boundary conditions using the Crank-Nicholson method. You will also have the opportunity to investigate the accuracy of the method relative to spatial and temporal step sizes.

<u>Theoretical Background</u>

We have seen in class that one of the many possible finite difference algorithms that can be applied to heat equation problems is the Crank-Nicholson method. The one-dimensional form of the heat equation problem with Dirichlet boundary conditions can be expressed as

$$c\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}, \qquad a \le x \le b \quad \text{and} \quad t \ge 0,$$

$$u(a,t) = u_a \qquad u(b,t) = u_b \qquad u(x,0) = f(x),$$

where c is the thermal diffusivity of the material. The corresponding finite difference update equation used in the Crank-Nicholson method is given by

$$u_{i-1,j+1} - \alpha u_{i,j+1} + u_{i+1,j+1} = -u_{i-1,j} + \beta u_{i,j} - u_{i+1,j},$$

where

$$\alpha = 2 + \frac{2\Delta x^2}{c\Delta t} \quad \text{and} \quad \beta = 2 - \frac{2\Delta x^2}{c\Delta t},$$

and where $u_{i,j}$ is the value of the dependent variable (heat or temperature) at location $x = a + (i-1)\Delta x$ for $i = 1, 2, 3, \ldots, N_x$ (where $x = a$ is the left-most boundary) and at time $t = j\,\Delta t$ for $j = 1, 2, 3, \ldots, N_t$. The quantities N_x and N_t are the total number of discrete spatial locations and time steps, respectively. As with the other finite difference methods, the dependent variable u is evaluated only at a finite number N_x of discrete points in the solution space. Thus, all of the values of u at a given moment in time are stored in a software vector of length N_x. Two special-case update equations are applied at the boundaries of the solution space. They are given by

$$-\alpha u_{2,j+1} + u_{3,j+1} = -u_{1,j} - u_{1,j+1} + \beta u_{2,j} - u_{3,j}$$

and

$$u_{N_x-2,j+1} - \alpha u_{N_x-1,j+1} = -u_{N_x-2,j} + \beta u_{N_x-1,j} - u_{N_x,j} - u_{N_x,j+1}.$$

The terms on the right-hand sides in each equation ($u_{1,j}$ and $u_{1,j+1}$ in the first one and $u_{Nx,j}$ and $u_{Nx,j+1}$ in the second one) correspond to the boundaries at $x = a$ (for which $i = 1$) and at $x = b$ (for which $i = N_x$), respectively. Thus, for Dirichlet boundary conditions the two terms would be set equal to the value u_a in the first equation and to u_b in the second equation. If the boundary

conditions were time dependent, then the two terms would be evaluated at times $j\Delta t$ and $(j + 1)\Delta t$, respectively. Thus, the general forms of the special update equations are

$$-\alpha u_{2,j+1} + u_{3,j+1} = \beta u_{2,j} - u_{3,j} - u_a\left(j\Delta t\right) - u_a\left[\left(j+1\right)\Delta t\right]$$

and

$$u_{N_x-2,j+1} - \alpha u_{N_x-1,j+1} = -u_{N_x-2,j} + \beta u_{N_x-1,j} - u_b\left(j\Delta t\right) - u_b\left[\left(j+1\right)\Delta t\right].$$

The Crank-Nicholson finite difference method is referred to as an *implicit* method because new values of the dependent variable u must be calculated at multiple points in the solution space simultaneously. That is, the left-hand side of the regular update equation for the interior points has three terms corresponding to three adjacent locations that are evaluated at time $(j + 1)\Delta t$. The determination of the new values of u requires the evaluation of an $(N_x - 2)$ by $(N_x - 2)$ system of equations involving tridiagonal matrices if the boundary conditions are of the Dirichlet type.

The system of equations must be modified if the problem is defined by Neumann boundary conditions, which have the form

$$\left.\frac{\partial u}{\partial x}\right|_{x=a} = u_{xa} \quad \text{and} \quad \left.\frac{\partial u}{\partial x}\right|_{x=b} = u_{xb},$$

where u_{xa} and u_{xb} could be constants or be time dependent. If $u_{xa} = u_{xb} = 0$, then the boundary conditions are homogeneous. The finite difference approximations of the boundary conditions are centered at the respective boundaries and lead to

$$\left.\frac{\partial u}{\partial x}\right|_{x=a} \approx \frac{u_{2,j} - u_{0,j}}{2\Delta x} = u_{xa} \quad \rightarrow \quad u_{0,j} = u_{2,j} - 2\Delta x\, u_{xa}$$

and

$$\left.\frac{\partial u}{\partial x}\right|_{x=b} \approx \frac{u_{N_x+1,j} - u_{N_x-1,j}}{2\Delta x} = u_{xb} \quad \rightarrow \quad u_{N_x+1,j} = u_{N_x-1,j} + 2\Delta x\, u_{xb}.$$

These expressions apply at time j. The corresponding approximations that apply at time $j + 1$ are

$$u_{0,j+1} = u_{2,j+1} - 2\Delta x\, u_{xa}$$

and

$$u_{N_x+1,j+1} = u_{N_x-1,j+1} + 2\Delta x\, u_{xb}.$$

The expressions above are substituted into the regular update equations centered at $i = 1$ and $i = N_x$, respectively, to eliminate the terms that correspond to locations outside the solution space (i.e., at $i = 0$ and at $i = N_x + 1$). The regular update equations are given by

$$u_{0,j+1} - \alpha u_{1,j+1} + u_{2,j+1} = -u_{0,j} + \beta u_{1,j} - u_{2,j}$$

and

$$u_{N_x-1,j+1} - \alpha u_{N_x,j+1} + u_{N_x+1,j+1} = -u_{N_x-1,j} + \beta u_{N_x,j} - u_{N_x+1,j}.$$

The special update equations that result after the substitutions (boxed here for easy reference) are

$$\boxed{-\alpha u_{1,j+1} + 2u_{2,j+1} = \beta u_{1,j} - 2u_{2,j} + 4\Delta x\, u_{xa}}$$

and

$$\boxed{2u_{N_x-1,j+1} - \alpha u_{N_x,j+1} = -2u_{N_x-1,j} + \beta u_{N_x,j} - 4\Delta x\, u_{xb}},$$

where both forms assume that the quantities u_{xa} and u_{xb} are time-invariant constants.

The Crank-Nicholson algorithm for Neumann boundary conditions can be expressed in matrix form as

$$A\mathbf{u}_{j+1} = B\mathbf{u}_j + \mathbf{c} \quad \rightarrow \quad \mathbf{u}_{j+1} = A^{-1}B\mathbf{u}_j + A^{-1}\mathbf{c} = D\mathbf{u}_j + \mathbf{d},$$

where $D = A^{-1}B$, $\mathbf{d} = A^{-1}\mathbf{c}$, and where

$$A = \begin{bmatrix} -\alpha & 2 & 0 & 0 & \cdots & 0 \\ 1 & -\alpha & 1 & 0 & \cdots & 0 \\ 0 & 1 & -\alpha & 1 & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & \cdots & 0 & 1 & -\alpha & 1 \\ 0 & \cdots & 0 & 0 & 2 & -\alpha \end{bmatrix}, \quad \mathbf{u}_{j+1} = \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \\ \vdots \\ u_{N_x-1,j+1} \\ u_{N_x,j+1} \end{bmatrix}, \quad \mathbf{u}_j = \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ \vdots \\ u_{N_x-1,j} \\ u_{N_x,j} \end{bmatrix},$$

$$B = \begin{bmatrix} \beta & -2 & 0 & 0 & \cdots & 0 \\ -1 & \beta & -1 & 0 & \cdots & 0 \\ 0 & -1 & \beta & -1 & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & \cdots & 0 & -1 & \beta & -1 \\ 0 & \cdots & 0 & 0 & -2 & \beta \end{bmatrix}, \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} 4\Delta x\, u_{xa} \\ 0 \\ 0 \\ \vdots \\ 0 \\ -4\Delta x\, u_{xb} \end{bmatrix}.$$

As with the explicit method, the initial condition is accommodated by filling the solution vector at time step $j = 0$ with the values of $f(x)$ evaluated at each location in the solution space using

$$u_{i,0} = f\big[a + (i-1)\Delta x\big] \qquad i = 1, 2, 3, \ldots, N_x.$$

The primary advantage of implicit methods is that they are inherently stable. There is no stability constraint on either the spatial step size Δx or the time step size Δt. Moreover, their relationship to each other is not constrained. Nevertheless, the accuracy of the solution is highly dependent on the spatial and time step sizes. Reducing either or both leads to a more accurate solution. The observation of this characteristic forms part of this lab exercise.

<u>Procedure</u>

- Your assignment is to add a few lines of code to an incomplete *Matlab* m-file to solve a 1-D heat distribution problem using the Crank-Nicholson method for Neumann boundary conditions. Download the following *Matlab* m-files available at the course Moodle site. You should set up a separate folder to contain them.

  `CrankNicholson_lab.m` – primary m-file containing most of the procedural commands
  `f.m` – function that defines the initial temperature distribution $f(x)$

  You may change the name of `CrankNicholson_lab.m` if you wish.

  The m-files are heavily commented, and the places where you need to add code are clearly indicated. Edit the m-files, and then run a test case with the following problem parameters. They should already be defined in the file `f.m`:

$$u(a,t) = 300 \quad u(b,t) = 400 \quad u(x,0) = f(x) = 300 + \frac{100x}{b-a} + 100\sin^2\left(\frac{2\pi x}{b-a}\right)$$

  The other important parameters such as the boundary locations a and b, the thermal diffusivity c of the material, the number of spatial steps N_x and time steps N_t, etc. are already coded near the top of the main m-file.

- Run your modified m-files to verify that the routine is stable and producing accurate results. With the indicated spatial and time step sizes, the solution at $t = 42.75$ s (simulation time) should look like the one shown in Fig. 1. Write down the initial and final total energy values displayed in the *Matlab* command window after the script finishes executing.
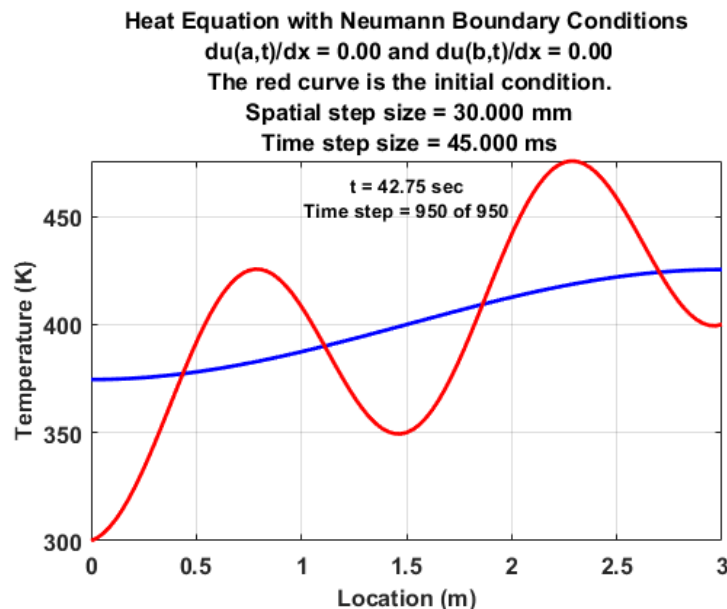

Figure 1. Solution of the homogeneous heat equation problem at time $t = 42.75$ s for the indicated boundary conditions and initial condition.

- Reduce the spatial step size Δx by a factor of one-half (by appropriately increasing the value of variable Nx) and run the simulation again. Note that the time step size Δt is set automatically according to the explicit algorithm stability constraint,

$$\Delta t = \frac{0.5\Delta x^2}{c}.$$

- Although it is not necessary to limit the value of Δt in the Crank-Nicholson method, it is being constrained here to provide a consistent basis for comparison to previous results. (Feel free to try a larger value of Δt to see what happens.) You will need to double the value of variable Nt to arrive at the same end point in time ($t = 42.75$ s). As before, write down the initial and final total energy values displayed in the *Matlab* command window after the script finishes executing.

- Reduce the spatial step size Δx (and Δt, which happens automatically) by a factor of one-half one more time and run the simulation again. Write down the initial and final total energy values displayed in the *Matlab* command window after the script finishes executing. In comments added to the end of your CrankNicholson_lab.m script, list the spatial and time step sizes and the difference between the initial and final total energy for each one. Also add some brief comments about what you observed and what the implications are.

- Set the Neumann boundary condition constants to $u_{xa} = 50$ and $u_{xb} = -100$ and run the script one more time with the same initial condition as above and with any reasonable values for Δx and Δt. Confirm that the result at simulation time $t = 42.75$ s looks like Figure 2. In comments added to the end of your CrankNicholson_lab.m script, explain how you know (or don't know) that the slopes at the left and right ends of the blue curve are approximately correct.
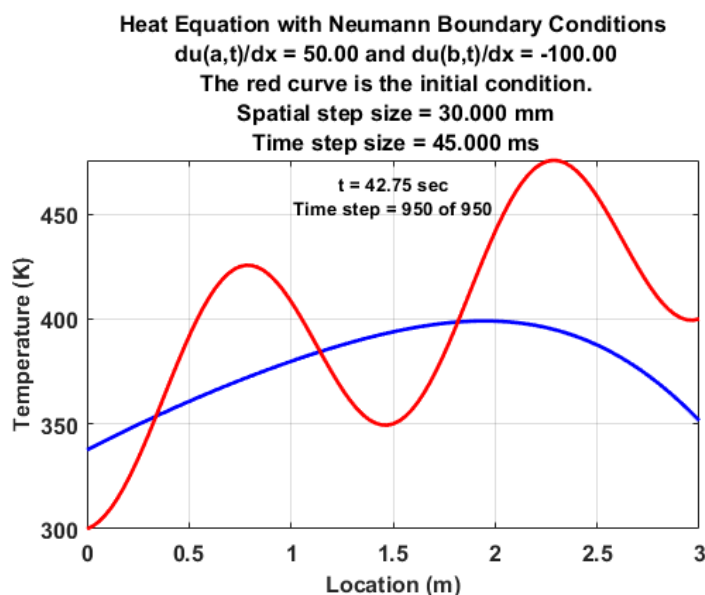

Figure 2. Solution of the homogeneous heat equation problem at time $t = 42.75$ s for $u_{xa} = 50$ and $u_{xb} = -100$ and the indicated initial condition.

Lab Work Submission and Scoring

Save a copy of your edited `CrankNicholson_lab.m` script. Change the name to `LName_Lab11_fa23.m`, where `LName` is your last name. Add comments to the script in response to the prompts above, which are repeated below:

- List the spatial and time step sizes and the difference between the initial and final total energy for each one. Also add some brief comments about what you observed and what the implications are.
- Explain how you know (or don't know) that the slopes at the left and right ends of the blue curve that you obtain for the $u_{xa} = 50$ and $u_{xb} = -100$ case are approximately correct.

E-mail your edited script with comments to me.

Your score will be assigned according to the lab scoring rubric posted on the Laboratory page at the course web site.

If you do not complete the exercises during the lab session, then you may submit your documentation as late as 11:59 pm on Tuesday, December 5. Since the semester ends on December 5, submissions after the deadline will not be accepted unless extenuating circumstances apply.