ENGR 695 Advanced Topics in Engineering Mathematics Fall 2023

Lecture Outline for Friday, Dec. 1

1. Crank-Nicholson (often spelled Crank-Nicolson) method applied to heat equation (continued)

$$c\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}$$

   a. Implicit update equation:

$$u_{i-1,j+1} - \left(2 + \frac{2\Delta x^2}{c\Delta t}\right)u_{i,j+1} + u_{i+1,j+1} = -u_{i-1,j} + \left(2 - \frac{2\Delta x^2}{c\Delta t}\right)u_{i,j} - u_{i+1,j}$$

   Left-hand side has terms at three adjacent locations ($i + 1$, i, and $i - 1$), which leads to a set of coupled equations, that is, a system of equations (matrix equation).

   b. Special cases at boundaries. For Dirichlet BCs ($u_{1,j+1} = u_a$ and $u_{Nx,j+1} = u_b$) use:

   at $x = a$, substitute $u_{1,j+1} = u_{a,j+1}$ and $u_{1,j} = u_{a,j}$ (if u_a does not vary with time, then substitute $u_{1,j+1} = u_{1,j} = u_a$):

$$u_{a,j+1} - \left(2 + \frac{2\Delta x^2}{c\Delta t}\right)u_{2,j+1} + u_{3,j+1} = -u_{a,j} + \left(2 - \frac{2\Delta x^2}{c\Delta t}\right)u_{2,j} - u_{3,j}$$

$$\rightarrow \quad -\left(2 + \frac{2\Delta x^2}{c\Delta t}\right)u_{2,j+1} + u_{3,j+1} = -u_{a,j+1} - u_{a,j} + \left(2 - \frac{2\Delta x^2}{c\Delta t}\right)u_{2,j} - u_{3,j}$$

   at $x = b$, substitute $u_{Nx,j+1} = u_{b,j+1}$ and $u_{Nx,j} = u_{b,j}$ (if u_b does not vary with time, then substitute $u_{Nx,j+1} = u_{Nx,j} = u_b$):

$$u_{N_x-2,j+1} - \left(2 + \frac{2\Delta x^2}{c\Delta t}\right)u_{N_x-1,j+1} + u_{b,j+1} = -u_{N_x-2,j} + \left(2 - \frac{2\Delta x^2}{c\Delta t}\right)u_{N_x-1,j} - u_{b,j}$$

$$\rightarrow \quad +u_{N_x-2,j+1} - \left(2 + \frac{2\Delta x^2}{c\Delta t}\right)u_{N_x-1,j+1} = -u_{N_x-2,j} + \left(2 - \frac{2\Delta x^2}{c\Delta t}\right)u_{N_x-1,j} - u_{b,j} - u_{b,j+1}$$

   c. Note that the first equation involves terms at $i = 2$ and $i = 3$ (but not at $i = 1$) and that the last equation involves terms at $i = N_x - 2$ and $i = N_x - 1$ (but not at $i = N_x$). The total number of equations is therefore equal to $N_x - 2$, which results in an $(N_x - 2) \times (N_x - 2)$ system of equations:

(*continued on next page*)

$$\begin{bmatrix} -\alpha & 1 & 0 & 0 & \cdots & 0 \\ 1 & -\alpha & 1 & 0 & \cdots & 0 \\ 0 & 1 & -\alpha & 1 & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & \cdots & 0 & 1 & -\alpha & 1 \\ 0 & \cdots & 0 & 0 & 1 & -\alpha \end{bmatrix} \begin{bmatrix} u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ \vdots \\ u_{N_x-2,j+1} \\ u_{N_x-1,j+1} \end{bmatrix} = \begin{bmatrix} -u_{a,j+1} - u_{a,j} + \beta u_{2,j} - u_{3,j} \\ -u_{2,j} + \beta u_{3,j} - u_{4,j} \\ -u_{3,j} + \beta u_{4,j} - u_{5,j} \\ \vdots \\ -u_{N_x-3,j} + \beta u_{N_x-2,j} - u_{N_x-1,j} \\ -u_{N_x-2,j} + \beta u_{N_x-1,j} - u_{b,j} - u_{b,j+1} \end{bmatrix},$$

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

Furthermore, the right-hand side can be expressed in matrix form as

$$\begin{bmatrix} -u_{a,j+1} - u_{a,j} + \beta u_{2,j} - u_{3,j} \\ -u_{2,j} + \beta u_{3,j} - u_{4,j} \\ -u_{3,j} + \beta u_{4,j} - u_{5,j} \\ \vdots \\ -u_{N_x-3,j} + \beta u_{N_x-2,j} - u_{N_x-1,j} \\ -u_{N_x-2,j} + \beta u_{N_x-1,j} - u_{b,j} - u_{b,j+1} \end{bmatrix} = \begin{bmatrix} \beta & -1 & 0 & 0 & \cdots & 0 \\ -1 & \beta & -1 & 0 & \cdots & 0 \\ 0 & -1 & \beta & -1 & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & \cdots & 0 & -1 & \beta & -1 \\ 0 & \cdots & 0 & 0 & -1 & \beta \end{bmatrix} \begin{bmatrix} u_{2,j} \\ u_{3,j} \\ u_{4,j} \\ \vdots \\ u_{N_x-2,j} \\ u_{N_x-1,j} \end{bmatrix} + \begin{bmatrix} -u_{a,j+1} - u_{a,j} \\ 0 \\ 0 \\ \vdots \\ 0 \\ -u_{b,j} - u_{b,j+1} \end{bmatrix}.$$

d. Can express the matrix equation as

$$A\mathbf{u}_{j+1} = B\mathbf{u}_j + \mathbf{c},$$

$$\text{where} \quad A = \begin{bmatrix} -\alpha & 1 & 0 & 0 & \cdots & 0 \\ 1 & -\alpha & 1 & 0 & \cdots & 0 \\ 0 & 1 & -\alpha & 1 & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & \cdots & 0 & 1 & -\alpha & 1 \\ 0 & \cdots & 0 & 0 & 1 & -\alpha \end{bmatrix}, \quad \mathbf{u}_{j+1} = \begin{bmatrix} u_{2,j+1} \\ u_{3,j+1} \\ u_{4,j+1} \\ \vdots \\ u_{N_x-2,j+1} \\ u_{N_x-1,j+1} \end{bmatrix}, \quad \mathbf{u}_j = \begin{bmatrix} u_{2,j} \\ u_{3,j} \\ u_{4,j} \\ \vdots \\ u_{N_x-2,j} \\ u_{N_x-1,j} \end{bmatrix},$$

$$B = \begin{bmatrix} \beta & -1 & 0 & 0 & \cdots & 0 \\ -1 & \beta & -1 & 0 & \cdots & 0 \\ 0 & -1 & \beta & -1 & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & \cdots & 0 & -1 & \beta & -1 \\ 0 & \cdots & 0 & 0 & -1 & \beta \end{bmatrix}, \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} -u_{a,j+1} - u_{a,j} \\ 0 \\ 0 \\ \vdots \\ 0 \\ -u_{b,j} - u_{b,j+1} \end{bmatrix}.$$

(*continued on next page*)

e. At each iteration (time step), evaluate the matrix expression

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

Matrices A and B (and vector **c** as well if the boundary conditions are not time varying) do not change with time, so $A^{-1}B$ and $A^{-1}\mathbf{c}$ can be computed once and stored before the algorithm begins. If boundary conditions are time varying, then $A^{-1}\mathbf{c}$ must be evaluated at each time step, but A^{-1} can be precalculated.

Matrix multiplication is time consuming, but at least Gaussian elimination is not required.

Matrix A is tridiagonal and positive definite; efficient routines are available to compute inverse.

f. Implicit method → no restriction on size of Δt for stability purposes. The method is unconditionally stable when applied to the heat equation.

g. Accuracy is second order in space and time, which means that errors are proportional to Δx^2 and Δt^2. Accuracy improved if Δx, Δt, or both are decreased, but computation time increases.