ENGR 695 Advanced Topics in Engineering Mathematics Fall 2023

Lecture Outline for Wednesday, Nov. 29

1. Crank-Nicholson Method (an implicit FD method) applied to heat equation

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

   a. Issues with explicit FD method:
      i. centered difference for *x*-derivative and forward difference for *t*-derivative
      ii. mixed differencing reduces accuracy slightly for a given Δx
      iii. stability criterion limits size of Δt
   b. One alternative: center all derivatives at time $t + 0.5\Delta t$ instead of at t or $t + \Delta t$. FD approximations become

$$t\text{-derivative: } \frac{\partial\left(x, t+0.5\Delta t\right)}{\partial t} \approx \frac{u\left(x, t+\Delta t\right) - u\left(x, t\right)}{\Delta t}$$

$$x\text{-derivative: } \frac{\partial^2 u\left(x, t+0.5\Delta t\right)}{\partial x^2} \approx \frac{u\left(x+\Delta x, t+0.5\Delta t\right) - 2u\left(x, t+0.5\Delta t\right) + u\left(x-\Delta x, t+0.5\Delta t\right)}{\Delta x^2}$$

   c. Indexing doesn't allow half time-steps, so

$$\begin{aligned}\frac{\partial^2 u\left(x, t+0.5\Delta t\right)}{\partial x^2} &\approx \frac{1}{2}\left[\frac{\partial^2 u\left(x, t+\Delta t\right)}{\partial x^2} + \frac{\partial^2 u\left(x, t\right)}{\partial x^2}\right] \\ &\approx \frac{1}{2}\left[\frac{u\left(x+\Delta x, t+\Delta t\right) - 2u\left(x, t+\Delta t\right) + u\left(x-\Delta x, t+\Delta t\right)}{\Delta x^2}\right. \\ &\qquad \left. + \frac{u\left(x+\Delta x, t\right) - 2u\left(x, t\right) + u\left(x-\Delta x, t\right)}{\Delta x^2}\right]\end{aligned}$$

   d. Approximation of *x*-derivative using index notation:

$$\frac{\partial^2 u\left(x, t+0.5\Delta t\right)}{\partial x^2} \approx \frac{1}{2\Delta x^2}\left(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} + u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right)$$

   e. FD approximation of heat equation becomes

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

(*continued on next page*)

f. Multiply both sides by $2\Delta x^2/c$, then gather $j + 1$ (new) terms on the left and j (old) terms on the right:

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

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).

g. 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 $u_{Nx,j+1} = u_{Nx,j} = u_b$):

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

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

h. Result is an $(N_x - 2) \times (N_x - 2)$ system of equations:

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

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

(*continued on next page*)

i. Note that the values of $u_{1,j+1}$ and $u_{1,j}$ are defined by the boundary condition at $x = a$. Likewise, the values of $u_{Nx,j+1}$ and $u_{Nx,j}$ are defined by the boundary condition at $x = b$.
j. The matrix does not change with time, so its inverse can be computed once and stored before the algorithm begins. The matrix is tridiagonal; efficient routines are available. (Could use Cholesky or LDL^T factorization, or special forms of them, for example.)
k. Pre-multiply the right-hand-side vector by the stored inverted matrix at every time step. It's time consuming, but at least Gaussian elimination is not required.
l. Implicit method → no restriction on size of Δt for stability purposes
m. Improved accuracy if Δx, Δt, or both are small, but computation time increases