3450:730 Advanced Numerical Partial Differential Equations
Using the tridiagonal solver on Au=b

A tridiagonal matrix has mostly zeros in it, so it is wasteful to store all of those values in memory, and it is especially wasteful to do Gauss Elimination on the entire system. The tridiagonal solver uses a special version of Gauss Elimination that utilizes the structure of the matrix to speed up the calculations. As an example, solving a 500×500 matrix using Gauss Elimination might take 18 seconds, while using the tridiagonal solver takes only 0.001 seconds. Going to a 1000×1000 system, the difference is more pronounced – 152 seconds versus 0.002 seconds.

You input only 3 columns of the matrix into the tridiagonal solver. I call them AL, AM, and AR. The structure appears below:

$$\begin{bmatrix} AM(1) & AR(1) & 0 & 0 & 0 & \cdots & 0 \\ AL(2) & AM(2) & AR(2) & 0 & 0 & \cdots & 0 \\ 0 & AL(3) & AM(3) & AR(3) & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & AL(imax-1) & AM(imax-1) & AR(imax-1) \\ 0 & 0 & 0 & 0 & \cdots & AL(imax) & AM(imax) \end{bmatrix}$$

If your matrix looks like

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -r & 1+2r & -r & 0 & 0 & 0 & 0 \\ 0 & -r & 1+2r & -r & 0 & 0 & 0 \\ 0 & 0 & -r & 1+2r & -r & 0 & 0 \\ 0 & 0 & 0 & -r & 1+2r & -r & 0 \\ 0 & 0 & 0 & 0 & -r & 1+2r & -r \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

then you would build the matrix in this way:

```
AL(1) = 0.d0
AM(1) = 1.d0
AR(1) = 0.d0
do i=2,imax-1
   AL(i) = -r
   AM(i) = 1.d0 + 2.d0*r
   AR(i) = -r
end do
AL(imax) = 0.d0
AM(imax) = 1.d0
AR(imax) = 0.d0
```