Homework #9 • Numerical Analysis II (math 416) • Waves in Two-Dimensions

- due Wednesday 31 November.
- quality of presentation will count heavily.

A) (6 readable pages) This problem is about solving the wave equation in two space dimensions, $u(x,y,t)$. The aim is to produce something like the figure output by the *hw09c.m* script that is available from the webpage.

Consider the PDE problem

$$u_{tt} = u_{xx} + u_{yy} \quad \text{on} \quad 0 \le x \le 6 \ , \ 0 \le y \le 8 \ , \ 0 \le t \ .$$

You will need to specify IVs for $u(x,y,0)$ and $u_t(x,y,0)$ and BCs around the edges of the spatial domain.

A good test for your code is given by the exact solution to the wave equation

$$u(x,y,t) = \sin(\pi x)\ \sin(\pi y)\ \sin(\omega t)$$

where $\omega^2 = 2\pi^2$. This is consistent with zero Dirichlet BCs; the IVs can be calculated from the above exact solution. Use this case to do your debugging and convergence tests.

The scheme you should use is centered differencing in all variables,

$$\frac{u^{m+1}_{j,k} - 2u^m_{j,k} + u^{m-1}_{j,k}}{\Delta t^2} = \frac{u^m_{j+1,k} - 2u^m_{j,k} + u^m_{j-1,k}}{\Delta x^2} + \frac{u^m_{j,k+1} - 2u^m_{j,k} + u^m_{j,k-1}}{\Delta y^2}$$

where $u^m_{j,k} \approx u(x_j, y_k, t_m)$. Note that this is a 2-step method, and hence needs to be jump-started. A convenient way to do this using the IVs uses the Taylor series:

$$u(x,y,t+\Delta t) = u(x,y,0) + u_t(x,y,0)\,\Delta t + (u_{xx}(x,y,0) + u_{yy}(x,y,0))\ \frac{\Delta t^2}{2} + O(\Delta t^3) \ .$$

The real computation involves the following BCs & IVs:

$$\begin{aligned} \text{IVs}: \quad & u(x,y,0) = 0 \\ & u_t(x,y,0) = 0 \\ \text{BCs}: \quad & u(x,0,t) = b(x)\ \sin(4\pi\, t) \\ & u(x,8,t) = 0 \\ & u(0,y,t) = 0 \\ & u(6,y,t) = 0 \end{aligned}$$

where $b(x)$ is given by

$$b(x) = \begin{cases} e^{-50(x-2)^4} + e^{-50(x-4)^4} & \text{if greater than } 10^{-4} \\ 0 & \text{otherwise} \end{cases} .$$

The *pcolor* plot from *hw09c.m* illustrates a numerical approximation for $u(x,y,8)$. It uses $Nx = 97, Ny = 129, Nt = 200$ points (including the BC edges) so that $\Delta x = \Delta y$. Comment on whether or not this discretization represents an accurate solution.

Include pseudocode in your report.

Hints:

- I stored my solution in several matrix forms:

  u0 & u1 = $(Ny+1)\times(Nx+1)$ matrices for old & new solutions with BCs,

  u_old & u_new = $(Ny\text{-}1)\times(Nx\text{-}1)$ matrices for old & new solutions without BCs,

- here is my matrix structure:

$$\text{u0} = \left[\begin{array}{ccc} & b(x)\sin(4\pi\, t) & \\ \hdashline 0 & & 0 \\ \vdots & \text{u\_old} & \vdots \\ 0 & & 0 \\ \hdashline 0 & \cdots & 0 \end{array}\right]$$

  this allowed me to update u0 by *u0 = [b ; z1 u_old z1 ; z2];*

- the inverse procedure extracts u_old by *u_old = u0(2:Ny,2:Nx);*
- note that the matrix indices are switched, *u0(k,j)* is the value of $u(x_j, y_k, t_0)$
- the matlab *diff* function is useful for producing u_{xx} and u_{yy}
- there is a stability criterion for Δt