Due: Monday January 31, 2022

Problem Set 2

NPDE is the textbook *Numerical Partial Differential Equations*. Submissions are due in the LMS, and must be typeset (e.g. LaTeX).

1. (15 pts.) *Adopted from NPDE exercise 1.2.1:*
   Write a code to approximately solve

$$\begin{aligned} v_t &= \nu v_{xx}, \qquad x \in (0,1), \qquad t > 0 \\ x(x,0) &= f(x), \qquad x \in (0,1) \\ v(0,t) &= a(t), \qquad v(1,t) = b(t), \qquad t \geq 0. \end{aligned}$$

   Use the grid $x_j = j\Delta x$, with $j = 1, 2, \ldots, N$, and $\Delta x = 1/N$ (as described in the text), and apply the usual second-order centered spatial discretization with forward Euler time integration, i.e.

$$D_{+t} v_j^n = \nu D_{+x} D_{-x} v_j^n.$$

   Use $f(x) = \sin(2\pi x)$, $a = b = 0$, $\nu = 1/6$, and $N = 10$. Find solutions at $t = 0.06$, $t = 0.1$, $t = 0.9$, and $t = 50.0$. For the first three values of t, use $\Delta t = 0.02$. To speed the solution to the last value of t, you might use a larger value for Δt. Determine how large you can choose Δt and still get results that might be correct. Compare and contrast your solution to the exact solution.

2. (10 pts.) *Adopted from NPDE exercise 1.3.1:*
   Solve the IBVP from problem (1) using leapfrog temporal integration, i.e.

$$D_{0t} v_j^n = \nu D_{+x} D_{-x} v_j^n.$$

   Recall that $D_{0t} v_j^n = \frac{1}{2}(D_{+t} + D_{-t}) v_j^n = \frac{v_j^{n+1} - v_j^{n-1}}{2\Delta t}$. For convenience, use the values form the exact solution at $t = \Delta t$ to get the leapfrog scheme started. Do only the part with $\Delta t = 0.02$. It is also suggested that if the results of this problem are not nice, do note spend the rest of your life on in.

3. (15 pts.) Consider the heat equation

$$u_t = \nu u_{xx}, \qquad x \in (0,1), \qquad t > 0$$

   with initial conditions

$$u(x, t = 0) = \sin\left(\frac{5\pi}{2} x\right),$$

   and boundary conditions

$$\begin{aligned} u(x = 0, t) &= 0 \\ u_x(x = 1, t) &= 0. \end{aligned}$$

(a) Write a code to solve this problem using the usual second-order centered spatial discretization with forward Euler time integration, i.e.

$$D_{+t}v_j^n = \nu D_{+x}D_{-x}v_j^n$$

on the grid defined by $x_j = j\Delta x$, $j = 0, 1, \ldots, N$, $\Delta x = 1/N$, and the parameter $r = \nu\Delta t/\Delta x^2 = 0.4$. Ensure that your treatment of the boundary conditions are at least second-order accurate. Note that you may use a ghost cell to implement boundary conditions, but you are not required to do so.

(b) Setting $\nu = 1$, perform a grid refinement study using $N = 20, 40, 80, 160$ by computing the maximum errors in the approximation at a final time $t_f = .1$ for grid points $j = 0, 1, \ldots, N$ (i.e. do not compute errors in ghost cells if you use them). Produce a log-log plot showing the error as a function of Δx, and include a reference line indicating the expected rate of convergence.

(c) For each of the runs in part (b) determine the computational cost to obtain the approximation at the final time. For example you may use the MATLAB `tic` and `toc` commands. Discuss the scaling of the computational time with respect to the size of the grid. Produce a log-log plot showing the relation of the computational cost to the number of grid points, and include a reference line indicating the predicted growth in cost.