Assignment #10, the <u>Final</u> Assignment

Due *Wednesday 7 May, 2014*. Please put in my Dept. of Mathematics and Statistics (Chapman 101) box by 5:00pm.

Rules. You *may* use any reference, print or electronic, as long as it is clearly cited, but you *may not* search out online or other complete solutions to these particular problems, whether or not they exist. Please refer specifically to equations (or ideas) in the textbook if that promotes clarity. This assignment is worth a total of 90 points, twice the usual value. You *may not* talk or communicate about this exam with any person other than me:

`elbueler@alaska.edu` 474-7693

Read sections 4.9, 4.11, 6.1, 6.2 of MORTON & MAYERS, 2ND ED. This assignment also relates to material covered in earlier assignments.

1. For the advection PDE $u_t + a_0 u_x = 0$, with a_0 constant, the Lax-Friedrichs scheme is

$$(*) \qquad \frac{U_j^{n+1} - \frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right)}{\Delta t} + a_0 \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} = 0.$$

As you will see, this scheme has both some good properties and some surprisingly bad ones. It is rarely used in isolation, but often used as a part of another scheme, Lax-Wendroff.

(a) *5 points.* Show that the truncation error is $T(x,t) = A\Delta t + B\frac{\Delta x^2}{\Delta t} + C\Delta x^2$; identify the values of A, B, C in terms of a_0 and the derivatives of the exact solution $u(x,t)$. (*Hint: Use Taylor's theorem* with *remainder.*)

(b) *5 points.* By substituting von Neumann's expression $U_j^n = \lambda^n e^{ik(j\Delta x)}$, show that the scheme is conditionally stable, with the condition being $|\nu| \le 1$ where $\nu = a_0\Delta t/\Delta x$.

(c) *5 points.* By subtracting the definition of truncation error from the scheme, and using the standard definition $e_j^n = U_j^n - u(x_j, t_n)$ for the error, show that the scheme converges along the appropriate kind of refinement paths. (*State any assumptions about the refinement path!*)

(d) *5 points.* Calculate an expression for the phase of a numerical mode in the Lax-Friedrichs method. In particular, write down the analog of upwind formula (4.30) in MORTON & MAYERS.

(e) *5 points.* Modify my code

http://bueler.github.io/M615S14/upwindsquare.m

from the solutions to problem 2 on A#9, to create the Lax-Friedrichs version of Figure 4.6. Because in this case the velocity $a(x,t)$ is not constant, you should replace "a_0" in equation $(*)$ with "$a(x_j, t_n)$" when implementing the scheme. Describe the differences between the upwind and the Lax-Friedrichs result on this particular advection problem.

2. (*This problem is a simplification of Exercise 4.10 on page 149 of* Morton & Mayers.) Consider the system of first-order PDEs

$$\rho_t + u_x + v_y = 0, \tag{1a}$$

$$u_t + c^2 \rho_x = 0, \tag{1b}$$

$$v_t + c^2 \rho_y = 0. \tag{1c}$$

The solution of this system, subject to some initial and boundary conditions which do not worry us in this problem, is a set of three functions $\rho(x, y, t)$, $u(x, y, t)$, and $v(x, y, t)$. One lesson of this problem is that you often see first-order (coupled) systems of PDEs like system (1) which are hiding classical PDEs like the wave equation, namely equation (3) below.

Now, exercise 4.10 refers to a couple of different "staggered leap-frog" schemes for system (1). Here is one of them: Suppose an equally-spaced grid of points (x_j, y_k, t_n) with spacing $(h, h, \Delta t)$; this assumes $h = \Delta x = \Delta y$ for simplicity. Suppose $R^n_{jk} \approx \rho(x_j, y_k, t_n)$, $U^n_{jk} \approx u(x_j, y_k, t_n)$, $V^n_{jk} = v(x_j, y_k, t_n)$. The "alternative" scheme is:

$$\frac{R^{n+1}_{jk} - R^n_{jk}}{\Delta t} + \frac{U^{n+1/2}_{j+1/2,k} - U^{n+1/2}_{j-1/2,k}}{h} + \frac{V^{n+1/2}_{j,k+1/2} - V^{n+1/2}_{j,k-1/2}}{h} = 0, \tag{2a}$$

$$\frac{U^{n+1/2}_{j+1/2,k} - U^{n-1/2}_{j+1/2,k}}{\Delta t} + c^2 \frac{R^n_{j+1,k} - R^n_{jk}}{h} = 0, \tag{2b}$$

$$\frac{V^{n+1/2}_{j,k+1/2} - V^{n-1/2}_{j,k+1/2}}{\Delta t} + c^2 \frac{R^n_{j,k+1} - R^n_{jk}}{h} = 0. \tag{2c}$$

(a) *5 points.* Using three different colors (e.g. colored pens) for R, U, V, respectively, sketch the stencil for the scheme (2) as a "three-dimensional" figure in (x, y, t) space. Also sketch the projection of the stencil into the (x, y) plane and into the (x, t) plane. Do these by hand but make them reasonably neat, and large enough; together your sketches should take up close to one page. Label the points in your sketches clearly.

(b) *5 points.* Show from (1) that

$$\rho_{tt} = c^2(\rho_{xx} + \rho_{yy}). \tag{3}$$

That is, show that if ρ, u, v together solve the system (1) then in fact ρ solves the second-order wave equation (3). (*Hint. Differentiate* (1a) *with respect to* t. *Then use equality of mixed derivatives, and the other equations, to eliminate* u *and* v.)

(c) *5 points.* Show that if R, U, V solve the scheme (2) then R solves

$$\frac{R^{n+1}_{jk} - 2R^n_{jk} + R^{n-1}_{jk}}{\Delta t^2} = c^2 \left(\frac{R^n_{j+1,k} - 2R^n_{jk} + R^n_{j-1,k}}{h^2} + \frac{R^n_{j,k+1} - 2R^n_{jk} + R^n_{j,k-1}}{h^2} \right), \tag{4}$$

that is, the centered-time and centered-space scheme for (3); see also scheme (4.106). Sketch the stencil for (4) and compare to the sketch in part **(a)**.

(d) *5 points.* Do von Neumann stability analysis on (4). Specifically, by substituting

$$R^n_{jk} = \lambda^n e^{iq_x(jh)} e^{iq_y(kh)}$$

into (4), show that λ solves

$$\lambda^2 + 2\nu^2 \left(\sin^2(q_x h/2) + \sin^2(q_y h/2) - 1\right) \lambda + 1 = 0$$

where $\nu = c\Delta t/h$. Then, by applying the result of exercise 2.6(i) in the textbook, show that scheme (4) is conditionally-stable for PDE (3) if the CFL condition $|\nu| \leq 1$ applies.

3. Consider the advection-diffusion problem

$$u_t + a u_x = b u_{xx} \tag{5}$$

with particular constants $a = 1$ and $b = 0.1$. Observe that the basic method recommended in section 2.15 would use the upwind scheme on the u_x term and the centered-space scheme on the u_{xx} term to get a conditionally-stable explicit scheme. This problem describes a more accurate version of that idea.

(a) *5 points.* The MacCormack scheme for (5) is an explicit "predictor-corrector" scheme; these words are also used to describe some ODE schemes. In the case above where $a > 0$ and $b > 0$ are constant, the scheme is

(predictor)
$$\frac{U_j^{n+*} - U_j^n}{\Delta t} + a\frac{U_j^n - U_{j-1}^n}{\Delta x} = 0$$

(corrector)
$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\left(\frac{1}{2}\frac{U_j^n - U_{j-1}^n}{\Delta x} + \frac{1}{2}\frac{U_{j+1}^{n+*} - U_j^{n+*}}{\Delta x}\right)$$
$$= b\left(\frac{1}{2}\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} + \frac{1}{2}\frac{U_{j+1}^{n+*} - 2U_j^{n+*} + U_{j+1}^{n+*}}{\Delta x^2}\right)$$

Note that the quantities U_j^{n+*} are "tentative" values at the new time t_{n+1}. These tentative values are then used in the corrector step to get the "true" values U_j^{n+1} at the new time. Draw separate stencils for the predictor and corrector steps; use clearly-labeled color or symbols as needed to make the corrector step clear. What is the usual name for the predictor step?

(b) *10 points.* Implement the MacCormack scheme using the initial condition

$$u(x, 0) = e^{-5(x-1)^2}$$

and boundary conditions $u(0, t) = 0$ and $u(5, t) = 0$ on the interval $0 \le x \le 5$. In particular, compute an approximation of $u(x, t)$ at $t_f = 3$. For runs with $J = 50$ and $J = 200$ subintervals, and choosing Δt so that

$$\mu = \frac{b\Delta t}{\Delta x^2} = \frac{1}{2},$$

produce a plot of the initial and final U values. From these two runs, and possibly finer-grid runs if needed, estimate

$$\max_{0 \le x \le 5} u(x, t_f),$$

providing some evidence that your value is accurate to 4 digits.[1]

(Extra Credit) For 4 points extra credit, do Exercise 5.4 from Morton & Mayers. For another 4 points extra credit, find different initial and boundary conditions so that you do know the exact solution of (5), and evaluate the measured error of the scheme. (*Hint: These are not easy points to get.*)

[1]Since I don't know the exact solution, I cannot advise on how to measure the error, but my intuition for this problem is clear because I understand the advection ($u_t + a u_x = 0$) and diffusion ($u_t = b u_{xx}$) problems separately.

4. *15 points.* Modify my code

http://bueler.github.io/M615S14/ftcs.m

to use the leap-frog scheme (4.90) or (4.91) on the problem already solved by `ftcs.m`, namely $u_t - 2u_x = 0$ on $-1 \le x \le 1$ with initial condition $u(x,0) = \sin(5x) + 1$ and (upstream) boundary condition $u(1,t) = 0$. Use $t_0 = 0$ as the initial time and approximate $u(x,t_f)$ at $t_f = 0.99$. Compare runs with $J = 25, 50, 100, 200, 400$. As leap-frog is conditionally-stable, you will need to choose the time step in a clearly-explained manner. Because the leap-frog scheme is a three-level scheme, a choice is needed to get started: You may use the exact solution at t_0 *and the exact solution at* $t_0 + \Delta t$. Make a convergence plot of the maximum error $\max_j |U_j^N - u(x_j, t_f)|$ versus Δx, and comment on the apparent rate of convergence.

5. Sections 6.1 and 6.2 consider the Poisson equation

$$u_{xx} + u_{yy} + f = 0 \tag{6}$$

on a square, where $f(x,y)$ is a given function, and where $u = 0$ on the boundary. This problem asks you to consider a slightly more general problem: a rectangle instead of a square, and nonzero boundary conditions.

(a) *5 points.* Consider the rectangle $-1 \le x \le 1, 0 \le y \le 3$. Show that $u(x,y) = x^2 + \sin(y)$ solves (6) with $f(x,y) = \sin(y) - 2$ and boundary conditions

$$\begin{aligned} u(-1,y) &= 1 + \sin(y), \\ u(x,0) &= x^2, \\ u(1,y) &= 1 + \sin(y), \\ u(x,3) &= x^2 + \sin(y). \end{aligned}$$

Also explain, in a couple of sentences, how I generated this exact solution. (*Hint. It is fine to start your explanation with "You cheated by ..." But you still need to state clearly what choice I made and then what I did.*)

(b) *10 points.* Write a code which uses the centered-space finite difference scheme to approximately solve (6) with $J_x = 60$ equal spaces in the x-direction and $J_y = 50$ spaces in the y-direction, so that $\Delta x \ne \Delta y$. Measure the error from your scheme on this one grid by applying it to the problem in part **(a)**. Report the maximum value of the error (i.e. one number). (*Hint. In your solution, include a few hand-written equations to explain what your code does. Of course I expect you to solve the linear system by* `A\b` *in* MATLAB.)