CS 624: Numerical Solution of Differential Equations
Spring 2004
Problem Set 4

Handed out: Mon., Mar. 15.

Due: Wed., Mar. 31 in lecture.

1. Stability can be considered in norm other than L^2. For instance, the l_h^p-norm for $1 \le p < \infty$ is defined by
$$\|v\|_p = \left[h \sum_{j=-\infty}^{\infty} |v_j|^p \right]^{1/p}.$$
Show that if $\sigma \le 1/2$, then the Euler method for $u_t = u_{xx}$ (which is at the top of p. 122 of the text) is stable in the p-norm, $1 \le p < \infty$.

   [Hint: First note that the function $g(x) = |x|^p$ is convex for any p in the given range. Then apply Jensen's inequality where the weights are $1-2\sigma$, σ, σ. If you aren't familiar with Jensen's inequality, then see, e.g. http://mathcircle.berkeley.edu/trig/node2.html.]

2. In their original paper, CFL considered the following two-step explicit finite difference scheme for the full wave equation $u_{tt} = u_{xx}$:
$$v_j^{n+1} = 2v_j^n - v_j^{n-1} + \lambda^2(v_{j+1}^n - 2v_j^n + v_{j-1}^n).$$
Rewrite this as an explicit one-step vector finite difference method. Then apply a Fourier transform to the vector formula (see 3.6 of the text if you don't know how to do this) to determine the amplification factor $G(\xi)$. Finally, determine upper bounds on the absolute values of the eigenvalues of $G(\xi)$ assuming $\lambda \le 1$.

3. Design a finite difference method for the PDE
$$u_t = au + u_{xx}$$
where a is a nonpositive constant real number. This equation models heat flow in which the heat simultaneously diffuses and flows out of the object (because the object is soaking in an ice-bath, for instance). It is a special case of a class of PDE's called "reaction-diffusion equations." Show that the order of accuracy for your finite difference method is positive. Show that your method is stable. Show that your method is convergent (by citing the Lax equivalence theorem); your convergence argument should include an explicit relation among h, k, a.

4. Consider a chemical reaction taking place along a linear domain $[0, 1]$. The reaction is the same one as in PS2, except that the four species also diffuse as they react:
$$\begin{aligned}
\alpha_t &= c_1\alpha_{xx} - m_1\alpha + m_2\beta\gamma, \\
\beta_t &= c_2\beta_{xx} + m_1\alpha - m_2\beta\gamma, \\
\gamma_t &= c_3\gamma_{xx} + m_1\alpha - m_2\beta\gamma - m_3\gamma + m_4\delta, \\
\delta_t &= c_4\delta_{xx} + m_3\gamma - m_4\delta.
\end{aligned}$$

Use the same values for $m_1, \ldots, m_4$ as in PS2. Use $c_1 = c_3 = c_4 = 0.05$, and $c_2 = 1$. Solve these equations in matlab using the "method of lines" (see the first two pages of 3.3 of the text). That is, first discretize in space only: Replace the space-derivatives in the above equations with difference approximations. Assume that the four functions $\alpha, \beta, \gamma, \delta$ are continuous functions of t, each of which is defined at each grid point to obtain a system of ODEs. Then apply `ode15s` to this system of ODEs. (If you do this correctly, there will be $4N$ ODEs in the system, where N is the number of spatial grid points). Use as initial conditions $\alpha(x, 0) = 1$ for $x \leq 0.5$, $\alpha(x, 0) = 0$ for $x > 0.5$ and $\beta(x, 0) = \gamma(x, 0) = \delta(x, 0) = 0$ for all $x \in [0, 1]$. Use as boundary conditions $\alpha_x(0, t) = \alpha_x(1, t) = 0$ and similarly for β, γ, δ. To implement this boundary condition, copy the value at the 0 end of the domain into the -1 grid position when computing the finite difference approximation for α_{xx} at the 0 position, and similarly at the right end, and similarly for β, γ, δ.

Integrate to $t = 6$. Compare the performance of `ode15s` both with and without a user-specified Jacobian. Note that your user-specified Jacobian should be a matlab sparse matrix.

Hand in listings of your m-files, a few sentences of conclusions, and at least two interesting plots. In particular, make a surface plot of γ as a function of both x and t. Use `surf` or `surfl` for this purpose. Use the form of these functions in which the x- and y-axis arguments are vectors, and the z-argument is a matrix.