Assignment #8

Due Friday, 14 April at the start of class

Please read Chapters 7 and 8 of LeVeque. Most of Chapter 7 is important, but I am de-emphasizing multistep methods. In Chapter 8, but I will not cover 8.6.

P29. Consider the "θ-method" for $u' = f(u,t)$, namely

$$U^{n+1} = U^n + k[(1-\theta)f(U^n, t_n) + \theta f(U^{n+1}, t_{n+1})],$$

where $0 \le \theta \le 1$ is a fixed parameter.

a) Cases $\theta = 0, 1/2, 1$ are all familiar methods. Name them. Then show that the θ-method is A-stable for $\theta \ge 1/2$.

b) Plot the (absolute) stability regions for $\theta = 0, 1/4, 1/2, 3/4, 1$ and briefly comment on how the stability region will look for other values of θ (in $[0,1]$).

P30. Consider this one-step (Runge-Kutta) method, the *implicit midpoint method*,

$$U^* = U^n + \frac{k}{2} f\big(U^*, t_n + k/2\big),$$
$$U^{n+1} = U^n + k f\big(U^*, t_n + k/2\big).$$

The first equation (stage) is Backward Euler to determine an approximation to the value at the midpoint in time and the second stage is the midpoint method using this value.

a) Determine the order of accuracy of this method.

b) Determine the stability region.

c) Is this method A-stable? Is it L-stable?

P31. This problem is about a "typical" non-stiff ODE system, the *Lotka-Volterra predator-prey model*. The equations are

$$\frac{dx}{dt} = \alpha x - \beta x y$$
$$\frac{dy}{dt} = \delta x y - \gamma y$$

where $x(t)$ is the number of prey at time t and $y(t)$ the predators.[1] To keep things simple, we use $\alpha = 2/3, \beta = 4/3, \gamma = 1 = \delta$ and initial conditions $x(0) = y(0) = 2$.

Suppose we want to find $x(t)$ and $y(t)$ for the interval $0 \le t \le 20$, i.e. $t_0 = 0$ and $t_f = 20$. This question is about the choice of numerical methods for such a problem.

a) Using `forwardeuler.m` from the solutions to Assignment #7, or any other implementation of forward Euler (FE), confirm that $N = 50$ step results, i.e. with $k = t_f/N = 20/50 = 0.4$, produces exploding garbage. (*How do you know it is garbage? Observe that with larger N values the results are consistent and qualitatively-different.*)

[1] The famous case is snowshoe hares and lynx: http://www.pnas.org/content/94/10/5147.full.

b) Despite the explosion in part **a)**, no expert would describe this problem as "stiff."[2] We will show that, on this problem, implicitness does not yield a good solution with reduced computational cost; compare **P32** below for the opposite result. For concrete comparison of computational costs, fix $N = 50$ (and thus $k = 0.4$) for the rest of the problem, both parts **b)** and **c)**.

Implement both the backward Euler (BE) and RK2 methods[3] and apply them to this problem with $N = 50$. Plot the results and comment on their quality. (*Hints*. For BE you will need to solve nonlinear equations at each step, so a Newton iteration will be part of your code. You will find that using the current values as the initial iterate leads to convergence of the Newton iteration in a small, fixed number of steps; report what number of steps suffices, and why, and fix that number in your code. To evaluate "quality", compare to results for a larger N.)

c) Now count function evaluations for the $N = 50$ runs you just plotted, for all three methods (FE, BE, RK2). A fair way to count is to count "1" for each component of the right-hand-side of the ODE system, and count "1" for each evaluation of a Jacobian entry. Show that BE is actually more expensive than RK2 *and* that RK2 results are just as good.

In conclusion, on *this* ODE system, one's time is better spent improving order (FE $\to$ RK2) than adding implicitness (FE $\to$ BE).

P32. For a definitely-stiff problem, we apply the method-of-lines (MOL) to the heat equation,

$$u_t \overset{*}{=} u_{xx},$$

a PDE. Here $u(t, x)$ is the temperature in a rod of length one ($0 \le x \le 1$) and we set boundary temperatures to zero ($u(t, 0) = 0$ and $u(t, 1) = 0$). For an initial temperature distribution we set one part hotter than the rest:

$$u(0, x) = \begin{cases} 1, & 0.25 < x < 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

Suppose we seek $u(1, x)$, i.e. we set $t_f = 1$.

We discretize the spatial derivatives using the notation from Chapter 2. That is, suppose there are $m + 1$ subintervals, let $h = 1/(m + 1)$, and let $x_j = jh$ for $j = 0, 1, 2, \dots, m + 1$. Let $U_j(t) \approx u(t, x_j)$. By eliminating unknowns $U_0 = 0$ and $U_{m+1} = 0$, and noting that the time derivatives remain as derivatives, from $*$ we get a linear ODE system of dimension m,

$$U(t)' = AU(t)$$

where $U(t) \in \mathbb{R}^m$ and A is *exactly the matrix in equation* (2.10) *in the textbook*. For a given m, the components of $U(0)$ can be computed from the above formula for $u(0, x)$.

Finally the exercise itself: Implement both FE and BE on the above MOL heat equation system. In particular, store A using `sparse` storage and solve the equation (in BE) using backslash, which will automatically detect that the matrix is tridiagonal and then use the well-known efficient solution method for tridiagonal systems.

Now consider the $m = 100$ case. For BE, compute and show the solution using $N = 100$ time steps. For FE, $N = 100$ will generate extraordinary explosion. Instead, determine the time step k from the eigenvalues of A and the stability region of FE. Then compare the computational costs by counting floating-point multiplications.[4] You will conclude that an implicit is indeed effective in this case.

---

[2] The explosion does reflect a "stability problem," but nonlinearity confounds any easy classification.

[3] Note BE is L-stable while RK2 is not even A-stable.

[4] For an $m \times m$ tridiagonal matrix A, a Av costs $3m$ multiplications while $A\backslash v$ costs $5m$ multiplications.