Please provide complete and well-written solutions to the following exercises.

Due December 1, 1159PM PST, to be uploaded as a single PDF document to blackboard (under the Assignments tab).

Homework 11

Exercise 1. Complete NCM Problem 7.21, from here. This exercise investigates an atmospheric simulation.

Exercise 2. Fix $\lambda > 0$. Suppose $y\colon [0,\infty) \to \mathbf{R}$ satisfies $y(0) := 10$, and y satisfies the following ODE:

$$y'(t) = f(y(t)) := -\lambda y(t), \qquad \forall\, 0 \leq t \leq 100.$$

Note that an exact solution is $y(t) := 10e^{-\lambda t}$. In this exercise, we will try out different iterative methods for solving this ODE.

Consider setting $\lambda = 20, 200$ or 2000 and set the step size h to be $1, .1, .01$ and $.001$ in the following solution methods.

(a) $y_{n+1} = y_n + hf(y_n)$ (Euler's)
(b) $y_{n+1} = y_n + hf(y_{n+1})$ (backwards Euler's) (You should solve for y_{n+1}.)
(c) $y_{n+1} = y_n + hf(y_n + hf(y_n))$ (predictor-corrector Euler's)
(d) $y_{n+1} = y_n + hf(y_n + \frac{h}{2}f(y_n))$ (modified Euler's)
(e) $y_{n+1} = y_{n-1} + 2hf(y_n)$ (Nystrom's midpoint)

For each iterative method and for each value of λ and h, report the computed value of $y(100)$, and compare this value to the actual value $10e^{-100\lambda}$ (which is basically zero). Describe which methods perform the best, and which methods perform the worst. How do the results compare to theoretical error bounds? For example, for a multistep method (which is the case for (a) and (e)), is the method stable and consistent?

Exercise 3. Find the values of h where the recursions satisfy stability for the numerical methods (c), (d) and (e) from Exercise 2, and also for

(f) $y_{n+1} = y_n + \frac{h}{2}[f(y_n) + f(y_{n+1})]$ (Trapezoid rule)

Exercise 4. In this exercise, we will solve the following boundary value problem in three different ways. Let $y\colon [0,1] \to \mathbf{R}$ satisfy

$$y''(t) = (y(t))^2 - 1, \quad \forall\, t \in [0,1], \qquad y(0) = 0, \quad y(1) = 1.$$

(a) First solve this problem using the shooting method. That is, ignore temporarily the condition $y(1) = 1$, and instead impose the initial value conditions

$$y'(0) = \eta, \qquad y(0) = 0.$$

Denote the solution y which depends on t and η as $y(t,\eta)$. Then, create a Matlab function f, defined to be

$$f(\eta) := y(1,\eta) - 1$$

and look for the zero of this function. If you find an η such that $f(\eta) = 0$, then the solution y satisfies the original boundary value problem.

(b) Now observe that we can re-write the differential equation as

$$\frac{d}{dt}\left(\frac{(y')^2}{2} - \frac{y^3}{3} + y\right) = 0$$

and if we assume y is continuously differentiable, then this means

$$\frac{(y')^2}{2} - \frac{y^3}{3} + y = K \qquad (*)$$

is a constant function K. Since $y(0) = 0$, we can solve $(*)$ for $y'(0)$ to get $y'(0) = \pm\sqrt{2K}$ (but trusting the validity of our previous result, we assume that $y'(0) = +\sqrt{2K}$). Solving $(*)$ for $\frac{1}{y'(t)}$ and recalling that $(d/dt)y^{-1}(t) = 1/y'(y^{-1}(t))$, we can integrate $1/y'(t)$ to obtain the inverse function of y, denoted as $t(y)$:

$$t(y) = \int_{s=0}^{s=y} \frac{1}{\sqrt{2(K + \frac{s^3}{3} - s)}} ds. \qquad (**)$$

and since we want to impose the condition $y(1) = 1$ we want equivalently that $t(1) = 1$ so that we must find the zero of the following equation

$$g(K) = \left(\int_{s=0}^{s=1} \frac{1}{\sqrt{2(K + \frac{s^3}{3} - s)}} ds\right) - 1$$

and this will solve our problem. That is, finding such a K will find the inverse function of y via $(**)$, so that y is then obtained from $(**)$, since y is the inverse of $t(y)$.

(c) Now try a finite difference method, choosing $n+1$ equal subintervals of length $h = \frac{1}{n+1}$ which turns the equation $y'' = y^2 - 1$ into a system involving n unknowns

$$y_{i+1} - 2y_i + y_{i-1} = h^2(y_i^2 - 1), \quad \forall\, i = 1, \ldots, n$$

where $y_0 = 0$ and $y_{n+1} = 1$. In matrix form this is

$$Ay + b = h^2(y^2 - 1)$$

where A has -2's on the diagonal and $1's$ on the superdiagonal and subdiagonal and zeros elsewhere, $b_n = 1$ and $b_i = 0$ for $i < n$, and y^2 denotes y with each of its entries squared. In this particular instance, we write

$$Ay = h^2(y^2 - 1) - b \qquad (\ddagger)$$

choosing an initial guess for y, and repeatedly solve the linear system, improving our guess more each time until we are sufficiently close to the solution. That is, if y is fixed on the right side of $(\ddagger)$, then we solve for x the linear system $Ax = h^2(y^2-1)-b$. We then solve the linear system $Az = h^2(x^2 - 1) - b$ for z, and so on.

Exercise 5. Now consider the boundary value problem

$$y''(t) = \frac{-t(y'(t) + \pi \sin(\pi t))}{d} - \pi^2 \cos(\pi t), \quad \forall\, t \in [-1, 1], \qquad y(-1) = -2, \quad y(1) = 0. \qquad (*)$$

Here $d > 0$ is a fixed parameter.

(a) First try solving this problem with the shooting method for fairly small d (e.g. try $d = .1$, $d = .01$ and $d = .001$). What is the smallest value of d for which the shooting method is able to find a solution y with $|y(1)| < 10^{-3}$? Denote this value of d as $\widehat{d}$.

(b) Now try to manipulate your initial guesses to improve the performance of the `fzero` function used in the shooting method. If f denotes the function used in our definition of the shooting method, we want to have $|f(1)| < 10^{-2}$. With $d > 0$ fixed, let $\eta(d)$ denote the value of η that `fzero` returns when trying to solve $f(\eta) = 0$. Let $d_k := \widehat{d} \cdot (.8)^k$ for any $k \geq 0$. Then, try to use $\eta(d_k)$ as an initial guess in `fzero` when you apply the shooting method in the case $d = d_{k+1}$. Does this method work well for values of d smaller than $\widehat{d}$?

(c) This ODE is linear in the following sense. If y and $\widetilde{y}$ are solutions of the differential equation $(*)$ with only the initial value specified (i.e. where $y(1)$ and $\widetilde{y}(1)$ are not fixed), then $ay + (1-a)\widetilde{y}$ also satisfies $(*)$ for any $a \in \mathbf{R}$. So, if we use two different shootings y and $\widetilde{y}$ with differential initial derivatives $y'(-1)$ and $\widetilde{y}'(-1)$, why can we not just make a linear combination of them fit any of our desired solutions? That is, why can we not just pick a particular a such that $z := ay + (1-a)\widetilde{y}$ satisfies $z(1) = 0$? This is exactly what the book suggests we do. Can we do this on the computer to solve $(*)$ when d is small (e.g. $d = .001$)?

(d) Since the ODE is linear, try to use a single iteration finite difference method. Do this first with evenly spaced nodes, and then estimate the L^∞ error for the $n-$node solution by comparing it with the $2n$-node solution. For select d values, plot the minimal solution on $n = 2^j$ nodes where the estimated error E_n is less than 10^{-2}.

(e) Try using unevenly spaced nodes in a finite difference method. In particular, try to use more nodes near $t = 0$. Do the unevenly spaced nodes perform better than the evenly spaced case?