Jim Lambers
MAT 461/561
Spring Semester 2019-20
Final Exam (coding option)

This exam is due by 5pm on Thursday, May 14. You may use any resource, including the textbook, course notes, or other published information, but all sources outside of course material must be cited. You may not collaborate with anyone.

You must use the provided script `testcode.m` that is posted on the Assignments page of the course web site. When you are finished with the problems, you should be able to test your code using this script, *without modifications*.

1. Write a MATLAB function

`[X,Y]=fdlinear(p,q,r,a,b,alpha,beta,n)`

to solve the general linear two-point BVP

$$y'' = p(x)y' + q(x)y + r(x), \quad a < x < b, \quad y(a) = \alpha, \quad y(b) = \beta$$

using finite differences.

The parameters `p`, `q` and `r` are functions that evaluate the coefficients $p(x)$, $q(x)$ and $r(x)$, where x can be a scalar or vector. The parameters `a`, `b`, `alpha` and `beta` specify the boundary conditions above. The parameter `n` is the number of *interior* points (N in the textbook and notes) in the discretization of the solution that must be computed.

The function must return a `X`, a vector of x-values from $[a, b]$ that *includes* the boundary points a and b, and `Y`, a vector containing the values of the approximate solution at the x-values in `X`. Use your function to solve the linear BVP

$$y'' = 2y' - y + xe^x, \quad 0 < x < 2, \quad y(0) = 0, \quad y(2) = -4,$$

with $N = 100$ and $N = 200$ points. The script `testcode.m` is set up to do this for you, and plot the computed solutions for comparison to the exact solution.

Comment on how the performance varies between $N = 100$ and $N = 200$ points, in terms of the accuracy and efficiency, using the data reported by the `testcode.m` script.

2. Consider the general nonlinear two-point BVP Write the following functions to solve a problem of this form

$$y'' = f(x, y, y'), \quad a < x < b, \quad y(a) = \alpha, \quad y(b) = \beta.$$

(a) Write a function

`[X,Y,niter]=shootnewt(f,fy,fyp,a,b,alpha,beta,n,tol);`

that implements the shooting method, in which Newton's Method is used to solve the resulting nonlinear equation for the initial slope $t = y'(a)$. For the initial guess, use

$$t^{(0)} = \frac{\beta - \alpha}{b - a}.$$

The parameters f, fy and fyp are functions that evaluate f, f_y, and $f_{y'}$ at input parameters x, y ahd yp, representing x, y and y', respectively, that can be scalars or vectors. The parameters a, b, alpha and beta are used to specify the boundary conditions. The parameter n is the number of *interior* points in the discretization of the solution that must be computed, and tol is used to indicate when Newton's method should stop. Specifically, it should stop when the computed solution $y(b, t)$ satisfies $|y(b, t^{(k)}) - \beta| < \texttt{tol}$.

The function must return X, a vector of x-values from $[a, b]$ that *includes* the boundary points a and b, and Y, a vector containing the values of the approximate solution at the x-values in X, as well as niter, the number of iterations of Newton's Method.

(b) Write a function

`[X,Y,niter]=shootsec(f,a,b,alpha,beta,n,tol);`

that implements the shooting method, in which the Secant Method is used to solve the resulting nonlinear equation for the initial slope $t = y'(a)$. Use the same initial guess $t^{(0)}$ as in Newton's Method. For the second initial value $t^{(1)}$, use

$$t^{(1)} = t^{(0)} - \frac{y(b, t^{(0)}) - \beta}{b - a}.$$

The input and output parameters have exactly the same meaning as in shootnewt.

(c) Write a function

`[X,Y,niter]=fdnewton(f,fy,fyp,a,b,alpha,beta,n,tol);`

that solves the BVP using finite differences in conjunction with Newton's Method for a system of nonlinear equations. For the initial guess, use

$$y^{(0)}(x) = \alpha + \frac{\beta - \alpha}{b - a}(x - a).$$

The input and output parameters have exactly the same meaning as in shootnewt and shootsec, except that tol is used differently. Newton's Method should stop after k iterations when

$$\|\mathtt{y}^{(\mathtt{k})} - \mathtt{y}^{(\mathtt{k-1})}\|_\infty < \texttt{tol},$$

where $\mathtt{y}^{(\mathtt{k})}$ is the kth solution computed by Newton's Method.

Use all of your functions to solve the nonlinear BVP

$$y'' = y^3 - yy', \quad 1 < x < 2, \quad y(1) = \frac{1}{2}, \quad y(2) = \frac{1}{3},$$

with $N = 100$ and $N = 200$ points in each case. The script testcode.m is set up to do this for you, and plot the computed solutions for comparison to the exact solution.

Comment on how the performance varies between $N = 100$ and $N = 200$ points, in terms of the accuracy and efficiency, using the data reported by the testcode.m script.