Math 686: Chapter 8a Homework – Spring 2020

DUE: TUESDAY, APRIL 14, 2020

1. Solve the boundary value problem (BVP)

$$\begin{aligned} u''' &= f(x,u,u',u''), \\ u(0) &= \alpha_0, \\ u'(0) &= \alpha_1 \\ u(1) &= \beta_0 \end{aligned}$$

using a shooting method with bisection used for the nonlinear solver. For either case you can use your favorite method for solving initial value problems (e.g. Euler, Runge-Kutta, Matlab's ode solvers, ...) to help with the necessary function evaluations. Solve two different problems – the first with $\alpha_0 = 0$, $\alpha_1 = 1$ and $\beta_0 = 2$ and

$$f(x,u,u',u'') = u'' + uu' + 6 - 7x - 4x^3 - 3x^5$$

for which the BVP has exact solution $u(x) = x^3 + x$, and the second with $\alpha_0 = e^{-r}$, $\alpha_1 = re^{-r}$ and $\beta_0 = 1$ and

$$f(x,u,u',u'') = uu'' - u' + (r^3 + r)e^{r(x-1)} - r^2 e^{2r(x-1)}$$

for which the BVP has exact solution $u(x) = e^{r(x-1)}$. Keep the value of r easily varied in your code but use $r = 10$.

Please list in each case your best estimate for the shooting variable s (see comments below) and plot comparisons of your solution prediction with the exact solution.

See next page for hints and suggestions.

Hints and Suggestions:

1. Before you attempt to solve these boundary value problems write a code in Matlab that implements the bisection method (see notes from Chapter 7) that can be used to find a root of a function $Z(s) = 0$. This should involve two Matlab files (something like 'bisection.m' and a Matlab function file 'Z.m', or however you wish to name these files). Test this bisection code to make sure it can find a zero of some simple function $Z(s)$ on some interval $s \in [a, b]$. This will be a critical part of your BVP solver so make sure you get this working first.

2. Write the third order differential equation given above as a system of three first order differential equations

$$\frac{d}{dx}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} f_1(x, y_1, y_2, y_3) \\ f_2(x, y_1, y_2, y_3) \\ f_3(x, y_1, y_2, y_3) \end{bmatrix}$$

   where $y_1 = u$, $y_2 = u'$ and $y_3 = u''$. You will treat the x variable like a 'time' variable and solve an initial value problem (IVP). Your initial conditions for y_1, y_2 and y_3 will come from the boundary conditions at $x = 0$. Since there are only two boundary conditions at $x = 0$ you will need to introduce a third initial condition $y_3(x = 0) = s$, where s is known as a 'shooting' variable. How do you choose s and enforce the boundary condition $u(1) = \beta_0$? Define a function $Z(s) := y_1(x = 1; s) - \beta_0$ and use a method like bisection with a reasonable guess for s and bracket $s \in [a, b]$ where $Z(a)Z(b) < 0$. Finding a value for s that makes $Z(s) = 0$ (i.e. what 'bisection.m' will do for you) is equivalent to finding where $y_1(x = 1; s) = \beta_0$ (that is, $u(x = 1) = \beta_0$). For such a value of s the IVP solution and the BVP solution are the same. Observe that in the shooting problem in order to evaluate your function $Z(s)$ for some value of s you will need to solve the IVP and evaluate the quantity $y_1(x = 1; s) - \beta_0$. All of this means that the shooting method can be though of as a nonlinear solver method (like bisection) where the function evaluation is more complicated (i.e. given a value of s, to find $Z(s)$ you need to solve in IVP).

3. You will want to make some decisions about plotting your solution $u(x)$ for $x \in [0, 1]$ and comparing it with the corresponding exact solution. This could be helpful throughout the process even before you find a suitable value of s. However, once you have found a suitable value of s the corresponding solution to the IVP (or more specifically the y_1 component) should be in good agreement with the exact solution.