CEE 451
Coding Problem 4
Shooting Method

Implement the shooting method as defined in the tutorial below (eqns. 1a,b,c). Show your answer for two different numbers of intervals of N ($\Delta x = 1/N$), and two different initial guesses for s each (total of four plots) with a plot of the results for f as a function of x for each case.

How many Newton-Raphson iterations did you need in each case for convergence (i.e. when s^{new} is sufficiently close to s^{old})? (Don't confuse the counter M for the number of iterations with the number of intervals N used to define Δx. I am asking for convergence of the Newton-Raphson routine, not convergence in the sense of varying N until you get grid-independence.)

Perform the calculation also for the case

$$f'' + f^2 x = 0 \quad , \quad f(0) = 0 \quad , \quad f(1) = 2$$

In the above case, show the complete formulation of the variational problem (first-order ordinary differential equations and boundary conditions for f, g, F, G) along with your numerical results). Hint 1: as an example, $\partial(f^3)/\partial s = 3f^2 [\partial f/\partial s] = 3f^2 F$. Hint 2: the correct value of s is between 0 and 10.

To code this, you should use pieces of code that you have already developed in the first three coding problems.
Here is a sample outline: N is the number of intervals; M is the counter for Newton-Raphson.

```
Sub ShootSolver()
        Get_Input_Data 'xA, xB, fA, fB, gA = first guess of s = sinit, no of intervals
                'N, tolerance δ, Mmax = max no. of iterations etc.
        Setup 'compute x(i), set Converged = False, Failed = False, M = 0, sold =
                sinit etc.
        Do
                PD_little_f_and_little_g 'Predictor-corrector; you have code to do
                        this from                'coding assignment 3
                PD_big_F_and_big_G
                Find_new_s
                Test_For_Convergence 'snew sufficiently close to sold? You have
                        'code from coding assignment 1, Newton-Raphson, for this
        Loop Until Converged or Failed
        If Failed Then
                Complain
        Else
```

```
                PrintOut
        End If
End Sub
```

Please provide me with copies of code + numbers and plots for output, as you think is sufficient to characterize the solutions to the problems.

Shooting Method Tutorial

Let f = f(x) denote a function to be determined by the boundary value problem

$$f'' + f = 0 \quad , \quad f(0) = 0 \quad , \quad f(1) = 1 \qquad \text{(1a,b,c)}$$

This problem has an exact solution:

$$f(x) = \frac{\sin(x)}{\sin(1)} \qquad (2)$$

The shooting method can be used to find this solution numerically. Of course, a numerical method is not necessary to solve (1). But the shooting method also works for nonlinear boundary value problems for which there is no closed-form solution.

Before introducing the shooting method it is useful to review the Newton-Raphson method for solving algebraic equations iteratively. Let R(s) be some specified function of s (e.g. R(s) = 1-e^{-s}). We wish to solve the equation

$$R(s) = 0 \qquad (3)$$

Let s be a guess for the root. We can get an improved guess s^{new} by defining

$$\Delta s = s^{new} - s \qquad (4)$$

and solving the linearized form based on Taylor expansion

$$R(s + \Delta s) \cong R(s) + R'(s)\Delta s = 0 \qquad (5)$$

to obtain the relation

$$s^{new} = s - \frac{R(s)}{R'(s)} \qquad (6)$$

Equation (6) can now be solved iteratively for s.

To implement the solution of (1a,b,c) by the shooting method we first define

$$g \equiv f' \qquad (7)$$

so that (1a) reduces to two first-order equations,

$$\begin{aligned} f' &= g \\ g' &= -f \end{aligned} \qquad \text{(8a,b)}$$

We could solve this equation by marching from x = 0 using e.g. the Euler step method if both boundary conditions (1b) and (1c) were specified at x = 0. Boundary condition (1c), however, is specified at x = 1.

To overcome this problem, we replace (1c) with the condition

$$g(0) = s \qquad (9)$$

where the right value of s is the one that yields the value f = 1 at x = 1. We don't know what s should be yet, so we treat it as a free variable. Thus f and g are not only functions of x, but also of s;

$$f = f(x, s) \quad , \quad g = g(x, s) \tag{10a,b}$$

The boundary value problem (1) now becomes

$$\begin{gathered} \frac{df}{dx} = g(x,s) \\ \frac{dg}{dx} = -f(x,s) \\ f(0,s) = 0 \\ g(0,s) = s \end{gathered} \tag{11a,b,c,d}$$

where s is to be determined. Now (11a,b) can be solved numerically using e.g. the Euler step method. Let

$$\Delta x = \frac{1}{N} \tag{12}$$

where N is an appropriately large positive integer, and let $x_n = (n-1)\Delta x$, n = 1..N + 1, so that $x_1 = 0$ and $x_{N+1} = 1$. Further letting f_n and g_n denote the values of f and g at x_n, (11a,b) discretize to

$$\begin{gathered} f_{n+1} = f_n + g_n \Delta x \\ g_{n+1} = g_n - f_n \Delta x \end{gathered} \tag{13a,b}$$

where since $x_1 = 0$,

$$\begin{gathered} f_1 = 0 \\ g_1 = s \end{gathered} \tag{13c,d}$$

Equations (13a,b) are then solved by marching from n = 1 to n = N + 1.

Now the right value of s is the one that satisfies (1c), or thus

$$R(s) = f(1,s) - 1 = 0 \tag{14}$$

For any guess s, a better guess s^{new} is given as

$$s^{new} = s - \frac{R(s)}{\frac{dR}{ds}} = s - \frac{f(1,s) - 1}{\left.\frac{\partial f}{\partial s}\right|_{x=1}} \tag{15}$$

But in order to implement this we need to know ∂f/∂s at x = 1. So to do this we specify the *variational* parameters F(x,s) and G(x,s), where

$$\begin{gathered} F \equiv \frac{\partial f}{\partial s} \\ G = \frac{\partial g}{\partial s} \end{gathered} \tag{16a,b}$$

(Now in principle we should write ∂f/∂x in equation (11a) etc., because f is also a function of s, but for simplicity we will retain the notation df/dx and dg/dx.) We now take the derivatives of (11a,b,c,d) with respect to s to specify the *variational problem*:

$$\frac{dF}{dx} = G(x,s)$$
$$\frac{dG}{dx} = -F(x,s) \qquad (17a,b,c,d)$$
$$F(0,s) = 0$$
$$G(0,s) = 1$$

Now we can also solve the variational problem using the Euler step method;

$$F_{n+1} = F_n + G_n \Delta x$$
$$G_{n+1} = G_n - F_n \Delta x \qquad (18a,b)$$
$$F_1 = 0$$
$$G_1 = 1 \qquad (18c,d)$$

Thus (14) becomes

$$s^{new} = s - \frac{f_{N+1} - 1}{F_{N+1}} \qquad (19)$$

The problem now becomes fai-rly straightforward. Guess a value of s. Then solve (13a,b) and (18a,b) out to $x_N = 1$ to find f, g, F and G. Then get a better guess of s from (19). Repeat until convergence is obtained in s.

Try it out. The code is easy to implement in, for example, Matlab or Visual Basic for Applications in Excel. Since (1) has the analytical solution (2), it is easy to verify that the correct value of s is

$$s = \left.\frac{df}{dx}\right|_{x=0} = \frac{d}{dx}\left[\frac{\sin(x)}{\sin(1)}\right] = \frac{1}{\sin(1)} = 1.188 \qquad (20)$$

Try initial guesses of s that are different from 1.188 and see how many steps are needed for convergence. The shooting method often fails if the initial guess of s is too far from the correct value, so often one must play around to find first guesses that lead to convergence.