Numerical Methods 2014–15 Assignment 2 — A two-point boundary value problem **Due: 11pm on 19/12/2014**

See also the general information on assignments. Your submission must be a `.pdf` file containing a written account of your solutions to all the Questions below, together with any `.m` files created for use in MATLAB. All files must be zipped together in a single file and uploaded to Moodle by the submission deadline. **Do not deviate from the submission instructions.**

In this assignment, we will use the Newton-Raphson method to find the finite-difference solution to the two-point second-order nonlinear boundary value problem (BVP)

$$u'' + \exp(u)u' = \mu \sin(2\pi x), \qquad u(0) = 1, \quad u'(1) + u^3(1) = 0, \tag{1}$$

We will investigate the behaviour of the numerical scheme as a function of the grid spacing. We will also investigate the behaviour of the solution as a function of the parameter μ.

Below is a list of specific questions that are useful as a guide in completing these tasks.

Question 1 *Use a uniform grid composed of $N+1$ nodes. Define the grid spacing and the grid points x_i. Introduce appropriate notation to describe the approximate solution to the BVP at each node.* (Marks: 2)

Question 2 *Use second-order of convergence finite-difference approximations to generate difference equations that correspond to the ODE discretised at the interior nodes.* (Marks: 5)

Question 3 *Generate difference equations corresponding to the boundary conditions. Ensure second-order of convergence and use the backward-difference formula for u' where appropriate.* (Marks: 5)

Question 4 *Write the difference equations in the form $\mathbf{F}(\mathbf{u}) = 0$ and calculate the entries of the Jacobian $J_{ij} = \partial F_i / \partial u_j$.* (Marks: 8)

Question 5 *Complete the* MATLAB *code for the solution of BVP (1) provided below. You should specify the value of the parameter μ, the number of nodes N, the interval end points, the desired error tolerance, the initial approximation, the entries of the vector $\mathbf{F}$ and the entries of the Jacobian $\widehat{\mathbf{J}}$.*

When you run the script it should generate a solution matrix `SOL` whose columns are successive approximations to the solution to the difference equations. Plot the iterates (the columns of `SOL`) to visualize the steps the Newton-Raphson method takes as it converges towards the solution. Use $\mu = 10$, tolerance 10^{-8} and sufficient number of nodes. (Marks: 8)

Question 6 *Modify the script produced in Question 5 into a function that takes as its argument the value for μ and outputs the values of u at $x = 1/4$, $1/2$, and $3/4$. Use this function to produce a plot showing curves of $u(1/4)$, $u(1/2)$ and $u(3/4)$ against the parameter μ for 51 values of μ from -20 to 30. Use tolerance 10^{-8} and sufficient number of nodes.* (Marks: 6)

We now investigate the order of convergence of one of the two main numerical methods used – the finite-difference **discretisation** method. The solution depends on two independent numerical parameters, namely the number of of the finite-difference discretisation nodes N and the number of Newton-Raphson iterations n. The rate of convergence of the finite-difference method is related to N, and we keep all other parameter values fixed. To simplify the notation we denote the approximation to the solution u at $x = 1/2$ and $\mu = 10$, after $n = 10$ Newton-Raphson iterations by v, i.e.

$$v(N) = u\left(x = \frac{1}{2},\ \mu = 10;\ n = 10,\ N\right).$$

Question 7 *Demonstrate using a graph that the order of convergence of the finite-difference disretisation method is 2. One way to do this is to consider the convergence of the value of u at a fixed location, say $x = 1/2$, then plot the ratio*

$$\frac{\left|v(2^{k-1}) - v(2^{k-2})\right|}{\left|v(2^{k}) - v(2^{k-1})\right|}$$

as a function of k and observe that the ratio tends to 4 as k is increased. (See Theorem 3.15 and Tables such as 5.4 in the revised Lecture notes). (Marks: 6)

CONTINUED

Listing 1: Code template for the BVP (1).

```
% Code to calculate the solution to a
% nonlinear BVP with Dirichlet and Robin BCs

% Some physical parameter.
mu=;

% Define the interval over which solution is calculated.
a=; b=;

% Define N.
N=;

% Define the grid spacing.
h=(b-a)/N;

% Define the grid. Note that there is no need to solve at x=a.
x = reshape(linspace(a+h,b,N),N,1);

% Define an initial guess for the solution
% of the BVP (make sure it is a column vector)
U=;

% Define a tolerance for the termination of Newton—Raphson.
tol=;

% Ensure that F is such that at least one iteration is done.
F=ones(N,1);
J=zeros(N,N);

% Store the initial guess in SOL.
SOL= U;

% Loop while the norm(F) is greater than tol.
while (norm(F)>tol )

%% Define F(u).
% Boundary conditions.
F(N)=;

% Finite difference approximation to ODE at interior nodes.
F(1)=;
for i=2:N-1
   F(i)=;
end;

%% Define the Jacobian J.
% First row corresponds to BC at x=0.
J(1,:)=;

% Last row corresponds to BC at x=1.
```

```
J(N,:)=;

% Intermediate rows correspond to F(i)=...
for i=2:(N-1)
% Diagonal entries.
  J(i,i)=;
% There may be off-diagonal entries, check your calculation.
  J(i,:)=
end

%% Having defined F and J, update the approximate solution to
% the difference equations.
U=U-J\F ;

%% Store the new approximation.
SOL=[SOL,U];

end

%% Insert your plot commands here.
```

END