UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

School of Computer Science and Applied Mathematics

Numerical Methods
Lab 3

15 April 2021

Instructions

- Read all the instructions carefully.
- MATLAB has a help file for every function if you get stuck.
- There are also numerous sources available on the internet, Google is your friend!

SECTION 1

Main Exercises

1.1

Exercise 1

Program the Bisection method to find the roots of equations. Your function should take as inputs, some function `f` (you will need to look up how function handles work), a tolerance value `tol`, as well as some initial bracket $I_0 = (\texttt{ainitial}, \texttt{binitial})$ respectively. Your function should return as output the root c . Use the stopping criteria $\frac{|c_n - c_{n-1}|}{|c_n|} < tol$ for the convergence of the sequence c_n's. The first line of your function must look like the following:

```
function root = bisectionSearch(f, tol, I_0)
```

1.2

Exercise 2

Program the False position method to find the roots of equations. Your function should take as inputs, some function `f` (you will need to look up how function handles work), a tolerance value `tol`, as well as some initial bracket $I_0 = (\texttt{ainitial}, \texttt{binitial})$ respectively. Use a similar sopping criteria as in Exercise 1. Your function should return as output the root c. The first line of your function must look like the following:

```
function root = RegularfalsiSearch(f, tol, I_0)
```

1.3

Exercise 3

Program the Newton root finding method for a scaler equation. Your function should take as inputs, two function handles `f` and `fprime` (you will need to look up how function handles work). It must also take in an initial guess x_0 and a tolerance value `tol`. Your function should return as output the root x. Use the stopping criteria $\frac{|x_n - x_{n-1}|}{|x_n|} < tol$. The first line of your function must look like the following:

```
function x = Newtonmethodscaler(f, fprime, x0, tol)
```

1.4

Exercise 4

Program the Newton root finding method for systems. Your function should take in as inputs, two function handles (anonymous functions) - F (system of equations as a column vector) and its Jacobian J (a matrix). It must also take in an initial guess x_0 and an tolerance value `tol`. The output of your function must be the roots of the system as a column vector x. The first line of your function should look like:

```
function x = Newtonmethodsystem(F, J, x0, tol)
```

1.5

Alternatively, a single function can be written, which when called, can find roots of non-linear scaler equations as well as systems of equations.

That is, instead of coding Exercise 3 and Exercise 4, you may wish to program the Newton root finding method for non-linear scaler equations as well as systems of equations. The first line of your function should look like:

```
function x = Newtonmethodsystem(F, Fprime, x0, tol)
```

where F can be a single function or a system of functions as a column vector, Fprime can be the derivative of the single function or the Jacobian matrix if F is a system. x0 can be a scaler or a vector.

If you choose this approach, it means you will end up with only three exercises, as opposed to four.