EM 314 - Assignment 2

Lecturer: Dr. Janitha Gunatilake

November 12, 2018

This assignment discusses iterative methods to solve nonlinear equations: bisection method, fixed point iteration and Newton's method.

Theory

1. Consider the bisection algorithm to solve an equation $f(x) = 0$, starting with an interval $[a, b]$. Show that the minimum number of iterations k required to achieve a tolerance τ satisfy

$$k > \log_2\left(\frac{b-a}{\tau}\right) - 1,$$

   Hint: Refer the error estimate of the bisection method.

2. Consider the function $g(x) = e^{-x}$.
   (a) Prove that g is a contraction on $G = [\ln 1.1, \ln 3]$.
   (b) Prove that $g : G \to G$.
   (c) Deduce that $x_{k+1} = g(x_k)$ converges to the unique fixed point $x_* \in G$ for **any** $x_0 \in G$.

3. Consider the fixed point iteration $x_{k+1} = g(x_k)$ where $g(x) = \tan^{-1}(2x)$.
   (a) Clearly, $x = 0$ is a fixed point of $g(x)$. Show that the fixed point iteration will not converge to this fixed point.
   Hint: Recall the definition of convergence: we say a sequence $\{x_k\}$ converges to x_*, equivalently $\lim_{k\to\infty} x_k = x_*$, if for every $\epsilon > 0$, there exists a number N such that

$$|x_k - x_*| < \epsilon \text{ whenever } k > N.$$

   Otherwise, the sequence diverges.
   (b) There is another fixed point x_* near $x = 1.16$.
      (i) Starting with an initial guess $x_0 = 2$, write 2 iterations of the fixed point iteration method to find x_*. At each iteration k, clearly indicate the approximate solution x_k and the error estimate e_k.
      (ii) Redo part (i) using the Newton's method.

Implementation

4. (a) Implement Newton's Method on MATLAB/GNU Octave.
Specifically, write a function `newtons()` in which input arguments are f, f', τ (tolerance), x_0, nmax (maximum number of iterations) that returns the approximate solution, residual and the number of iterations.
Note: Refer "function handles" to learn how to have mathematical functions as input arguments.

   (b) Test your code in part (a) solving the nonlinear equation $x^2 + 4x - 4 = 0$ with a tolerance 10^{-5}. Begin with $x_0 = 100$. Do you obtain the expected solution?
(Use a separate script `testNewtons.m` and save it in the same folder as `newtons.m`.)

   (c) For $k = 1, 2, \ldots$, tabulate your results as follows (here, x_* is the exact solution):

k	x_k	$e_k = \lvert x_k - x_* \rvert$	e_k / e_{k-1}^2

   You may modify your code to suit this test. Examine the last column of the table. Do you obtain quadratic convergence as expected?

   (d) Redo part (c) with $\tau = 10^{-8}$. Examine your results and comment.

Applications

5. **Kepler's Equation.**
The Cartesian coordinates of a planet in an elliptic orbit at time t are equal to $(ea \sin E, \cos E)$, where a is the semimajor axis, and e is the eccentricity of the ellipse. Using Kepler's laws of planetary motion, it can be shown that the angle E, called the eccentric anomaly, satisfies Kepler's Equation:
$$M = E - e \sin E, \quad 0 < |e| < 1,$$
where M is called the mean anomaly.

   Suppose $e = 0.8$, $M = 3$. Solve Kepler's Equation using Newton's method. Use your code in Question 4 with $\tau = 10^{-8}$.

   Some Historical Notes. The Kepler's Equation, which plays a vital role in celestial mechanics, has been first derived by the German mathematician/astronomer Johannes Kepler (1571-1630). For M not a multiple of π, Kepler's Equation has a unique solution. However, as Kepler's Equation is transcendental, it cannot be solved algebraically. Therefore, many numerical algorithms have been derived to solve this equation. Kepler has used an iterative method, similar to a fixed point iteration. Later, the English mathematician(/physicist/astronomer etc.) Isaac Newton (1643-1727) has used his method to solve the Kepler's Equation.

6. **State Equation of a Gas**
   The Van der Waals equation of state for a gas is given by

$$\left\{ p + a\left(\frac{N}{V}\right)^2 \right\}(V - Nb) = kNT,$$

   where V is the volume occupied by the gas, T is the temperature, p is the pressure, N is the number of molecules contained and k is the Boltzmann constant. a and b are coefficients that depend on the specific gas.

   Use bisection method to find the volume occupied by 1000 molecules of CO_2 at a temperature $T = 300K$ and a pressure $p = 3.5 \times 10^7 Pa$, with a tolerance of 10^{-12}. For carbon dioxide (CO_2) $a = 0.401 Pam^6, b = 42.7 \times 10^{-6} m^3$. The Boltzmann constant is $k = 1.3806503 \times 10^{-23} JK^{-1}$. You may use the code given in the Appendix.

Appendix

1. Bisection Method

```
function [zero, res, niter] = bisection(f,a,b,tol,nmax)

x = [a (a+b)/2 b]; y = f(x); niter = 0; I = (b-a)/2;
if y(1)*y(3)>0
    error('The signs of the function at the extrema must be opposite');
elseif y(1) == 0
    zero = a; res = 0; return
elseif y(3) == 0
    zero = b; res = 0; return
end
while ( I >= tol && niter <= nmax )
    if sign(y(1))*sign(y(2))<0
        x(3) = x(2); x(2) = (x(1) + x(3))/2;
        y = f(x); I = (x(3)-x(1))/2;
    elseif sign(y(2))*sign(y(3))<0
        x(1) = x(2); x(2) = (x(1) + x(3))/2;
        y = f(x); I = (x(3)-x(1))/2;
    else
        x(2) = x(find(y ==0)); I = 0;
    end
    niter = niter+1;
end

if niter > nmax
    fprintf('bisection method exited without convergence');
end

zero = x(2); res = f(x(2));
```

2. Example

Here I present an example we have already discussed in the class. Suppose we want to solve

$$x - \cos x = 0.$$

We write the function $f(x)$ in a file `f.m` in the same folder that contains `bisection.m`.

```
function y = f(x)
  y = x-cos(x);
```

Now, let's use the bisection method to solve the equation, with an initial interval $[0, \pi/2]$ as follows:

```
>> [zero, res, niter] = bisection(@f,0,pi/2,10^(-6),100)
```

Submission Guidelines:

- Produce a document that includes solutions, code, output and figures appropriately. Use a suitable cover page that includes the course title and the course code, assignment number, your name and the registration number.
- Submit your document in **PDF** format, online via **FEeLS**.
- **Due date**: November 23 (Friday), 2018, 11:55 pm. *Please note that marks will be deducted for late submissions.*