# Project 1

Due on Feb 12

## 1 Objective

The goal of this project is to combine two algorithms for solving an equation of one variable, namely Bisection method and Newton's method. The hybrid method should enjoy the benefit of guaranteed convergence of Bisection method as well as the fast convergence of Newton's method

## 2 Detailed Description

### 2.1 Motivation

The equation we want to solve has the form

$$f(x) = 0, \tag{1}$$

where $f(x)$ is a differentiable function on $[a, b]$. We know that if $f(a)$ and $f(b)$ has opposite signs, then Bisection method converges linearly to a zero of $f$, say $p$. We also know that for Newton's method, only if the initial guess $x_0$ is "sufficiently close" to $p$, it converges. If it converges, and $f'(p) \neq 0$, it converges quadratically – much faster than Bisection method. Sometimes "try and error" is needed to find a good initial guess. In this project we design a hybrid approach that automates the choice of initial guess for Newton's method.

### 2.2 Proposed Method

The pseudo code for the proposed method is Algorithm 1.

Two functions, say BISECTION and NEWTON are needed by Algorithm 1. BISECTION takes the function $f$, left and right end points, tolerance, and maximum number of iterations as inputs, and outputs the new end points and the solution. NEWTON takes the function $f$, its derivative $f'$, initial guess, tolerance and maximum number of iterations as inputs, and outputs the solution and a "flag" which denotes whether the algorithm converges or not (0 for not converged, 1 for converged).

**Algorithm 1** Hybrid method combining Bisection and Newton's method

```
input: end points a, b such that f(a) · f(b) < 0; tolerances for Bisection and
Newton's method tolb, toln; maximum number of iterations for Newton's
method nmax.
output solution p
function HybridBisectionNewton(f, f', a, b, tolb, toln, nmax)
    if f(a) · f(b) > 0 then
        return error message: end points are not valid
    end if
    error ← |b − a|
    tolb ← error/2                          ▷ set the tolerance for Bisection Method
    l ← a, r ← b                  ▷ set the initial end points for Bisection Method
    flag ← 0                       ▷ set the flag as the algorithm hasn't converged
    while flag = 0 do
        l, r, p ← Bisection(f, l, r, tolb, bmax = 1)        ▷ one step of Bisection
        p, flag ← Newton(f, f', p, toln, nmax)               ▷ try Newton's Method
        if flag = 1 then                              ▷ if Newton's Method converged
            output p
        else                                      ▷ if Newton's Method not converged
            tolb ← tolb/2                 ▷ decrease the Bisection tolerance by half
        end if
    end while
end function
```

### 2.3 Application

In orbital mechanics, Kepler's equation relates various geometric properties of the orbit of a body subject to a central force. It is written as

$$M(t) = E(t) - e\sin(E(t)), \tag{2}$$

where $t$ is the time, $e$ is the orbit constant called eccentricity, $M(t)$ is the average anomaly, and $E(t)$ is the eccentric anomaly.

$$M(t) = \frac{2\pi t}{T}, \tag{3}$$

where $T$ is the period of the orbit. Combining Equations (2) and (3) we have

$$\frac{2\pi t}{T} - E(t) + e\sin(E(t)) = 0, \tag{4}$$

where we want to solve $E(t)$ for any time $t \in [0, T]$.

In this project, we want to apply Algorithm 1 to solve Equation (4) for $E(t)$, given $e \in (0, 1)$ and $t \in [0, T]$. For trivial cases $t = 0$ , $t = T/2$ and $t = T$ we have $E(0) = 0, E(T/2) = \pi, E(T) = 2\pi$. Otherwise we can only solve $E(t)$ numerically.

In order to apply Algorithm 1, for fixed $t, e$ we denote

$$f(E) = \frac{2\pi t}{T} - E + e\sin(E). \tag{5}$$

Two end points $E_a, E_b$ are chosen such that

$$f(E_a) \cdot f(E_b) < 0. \tag{6}$$

For example, you can use $\frac{2\pi t}{T}$ and $\pi$ (why?). There are better choices than this (what is your choice?).

## 3 What to submit

Submit a zip file containing both computer codes and a report. The requirements are the following:

### 3.1 MATLAB/Octave Code

Write codes for the following functions:

1. Bisection method
2. Newton's method
3. Hybrid method combining Bisection and Newtons method (Algorithm 1)
4. Kepler equation solver

5. Code for testing: solving Kepler's equation for these parameters:

$$T = 1, e = 0.25, t = 0.01, 0.03, 0.05, \ldots, 0.97, 0.99.$$

   Find $E$ with 6 digits of accuracy.

## 3.2 Project Report

Write a PDF document containing the following components:

1. Documentation for MATLAB/Octave functions, including a description of what they do, what the variables in the input and output represent, and how to use the code (calling syntax).
2. List your choice of parameters like the end points, tolerance and maximum number of iterations, and explain why you think they are good.
3. Report the result for the testing case by plotting a graph of the curve $E(t)$ for $t \in [0, T]$.
4. Record the number of times Newton's method are called by Kepler's equation solver when solving $E(t)$ for each $t$ and plot a graph of this.