MATH 373–01 (or –02, or –03)
Brent Deschamp
Homework #1
January 31, 2018

---

1 Problem Statement

Solve Kepler's equation, $E - e\sin(E) = M$, when $e = 0.048$ and $M = \pi/4$ using the methods of Bisection and False Position.

2 Solution

In order to determine an appropriate interval in which to search for a root, Kepler's equation was plotted for a range of E-values. The results can be seen in Figure 1.

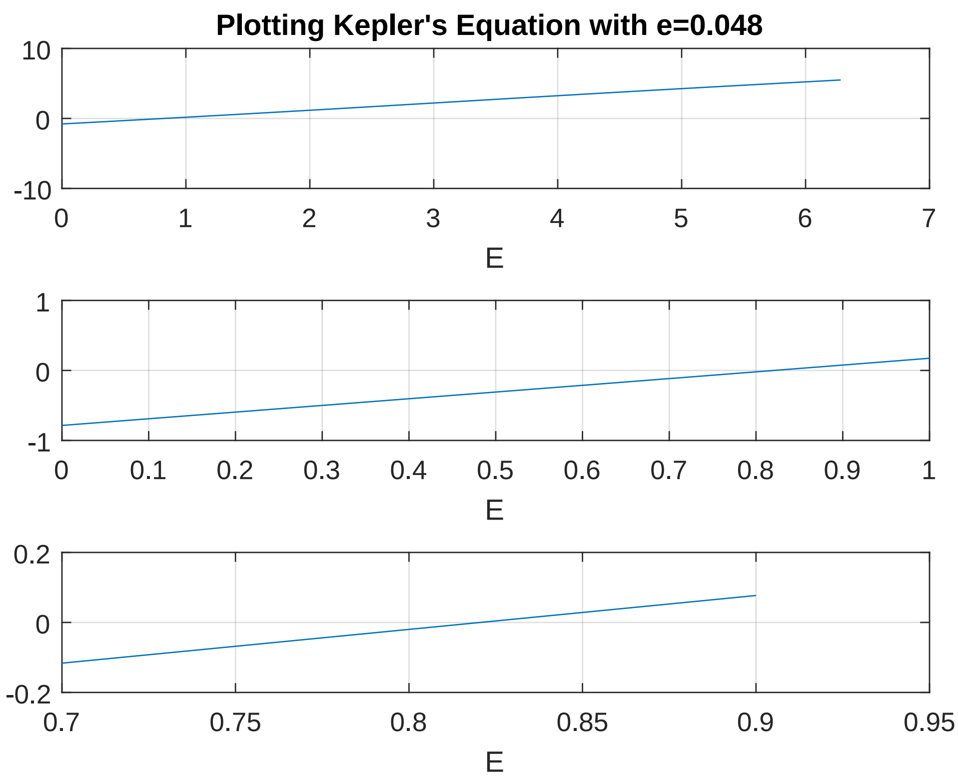

Figure 1: Plot of Kepler's equation.

An appropriate interval seems to be $[0.7, 0.9]$, and the results of Bisection and False Position can be seen in Table 1. Based on the methods, $E = 0.8205$ radians. This value is reliable because the function value at the root is very small for both methods. Also, the relative approximate error is small, and the methods terminated before reaching the maximum number of iterations, which is a good indication that the roots are accurate.

Table 1: Results from Bisection and False Position.

Method	`x_root`	`func_val`	ε_a	Iterations
Bisection	0.8205	2.3856×10^{-7}	9.2984×10^{-5}	18
False Position	0.8205	-3.6131×10^{-10}	3.0810×10^{-5}	3

3 Command-line Usage

Two programs were written to solve this problem, and the command-line usage is below.

```
>> Kepler_plot()

>> data = Kepler_problem()
```

4 Code

The code for plotting Kepler's equation can be found in Listing 1. A program was written to solve this problem, called `Kepler_problem`, and the code can be found in Listing 2. Note that `Kepler_problem` calls the programs `bisection` and `false_position`, but they are not included in this document.

Listing 1: MATLAB code for plotting Kepler's equation.

```
function Kepler_plot()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Brent Deschamp
% August 23, 2016
% Plot Kepler's Equation: E-e*sin(E) = M

% No inputs

% No outputs.  Plots are automatically generated
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Constants
e = 0.048;  % eccentricity for Jupiter
M = pi/4;   % mean anomaly

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Create values for E
E_vals = linspace(0,2*pi);
E_vals_finer = linspace(0,1);
E_vals_finer_still = linspace(0.7,0.9);

% Define function
Kepler = E_vals - e*sin(E_vals) - M;
```

```
Kepler_finer = E_vals_finer - e*sin(E_vals_finer) - M;
Kepler_finer_still = E_vals_finer_still - ...
        e*sin(E_vals_finer_still) - M;

% Plot the various functions
subplot(3,1,1);
plot(E_vals,Kepler);
xlabel('E');
grid
title(['Plotting Kepler''s Equation with e=' num2str(e)]);

subplot(3,1,2)
plot(E_vals_finer,Kepler_finer);
xlabel('E');
grid

subplot(3,1,3)
plot(E_vals_finer_still,Kepler_finer_still);
xlabel('E');
grid
```

Listing 2: MATLAB code for solving Kepler's equation.

```
function data = Kepler_problem()
% Solves Kepler's equation

% Allocate space for data
data = zeros(2,4);

% Save the working directory
current_folder = pwd;

% Constants
e = 0.048;
M=pi/4;

% Function definition
Kepler = @(E) E - e*sin(E) - M;

% Parameters
x_min = 0.7;
x_max = 0.9;

% Change to the appropriate directory
cd C:\Users\bdescham\Numerical\Root-Finding

% Bisection
[data(1,1),data(1,2),data(1,3),data(1,4)] = bisection(Kepler,x_min,x_max);
```

```
% False Position
[data(2,1),data(2,2),data(2,3),data(2,4)] =...
    false_position(Kepler,x_min,x_max);

% Change folder back
cd(current_folder);

end
```