University of Puerto Rico

Mayaguez Campus

Department of Electrical & Computer Engineering

Assignment #2: False Position Method

Osvaldo M. Cardona – 841-08-0990

Diana Rivera Negrón– 802-08-6908

Ricardo I. López Martínez – 802-08-3856

INGE 4035 – Section 096

Dr. Arocha Ordoñez

Specifications:

Propose a known function with 3 or more roots, whose roots are known (at least by the graphical method). Graph the function showing the approximate roots. Write code for the method assigned to your group. The code must be a modification of the bisection code. Construct variables similar to the bisection method, so both methods (or codes) could be easily compared. Best codes will be published in our website. The code must be developed to find ONE root. Run the code as many times as roots. Use the MATLAB functions fzero or roots, whichever applies and consider their results as the exact ones. Compute the error of your own code computation, error=|xtrue-xcomp|. Construct a table to show your results. Hand in over the graph, the code and the output. Engineering report quality with the group members clearly identified (name, ID, section). Send one PDF report per group to marcoantonioarochaordonez@gmail.com with the subject: ROOT FINDING before next Friday (August 28, 2015), midnight. Report sent to wrong email or with the wrong subject won't be graded.

I. The function and the plot to be worked with are shown below:

$$f(x) = x^3 - 151x^2 + 5150x - 5000$$

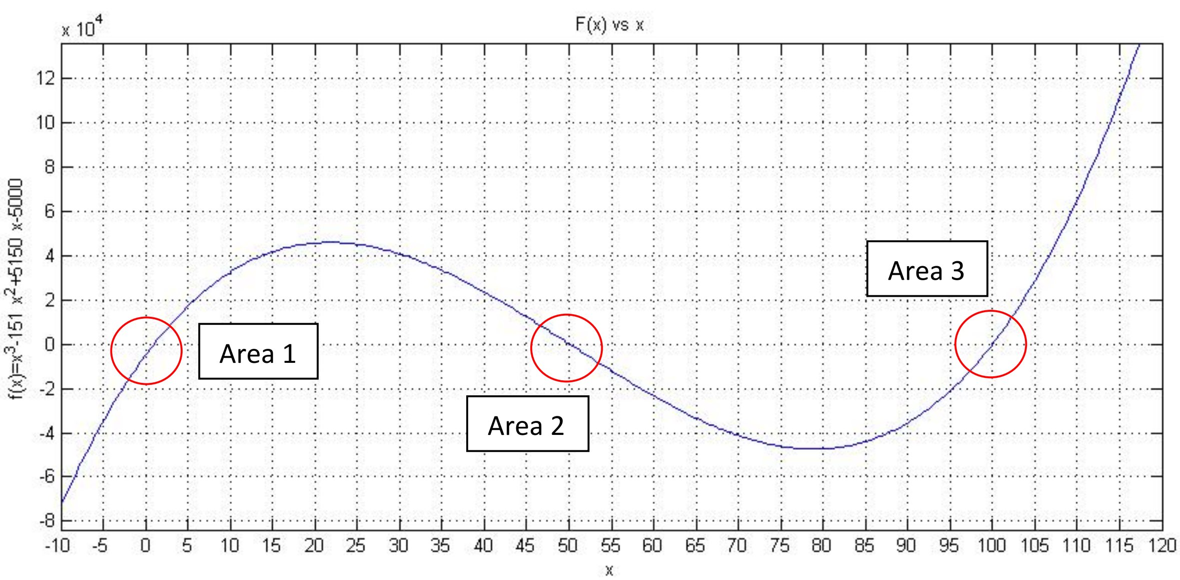

Figure 1.0: Plot for the function f(x). The red circles show the approximate areas where the roots occur.

II. Zoom-in for the three roots:

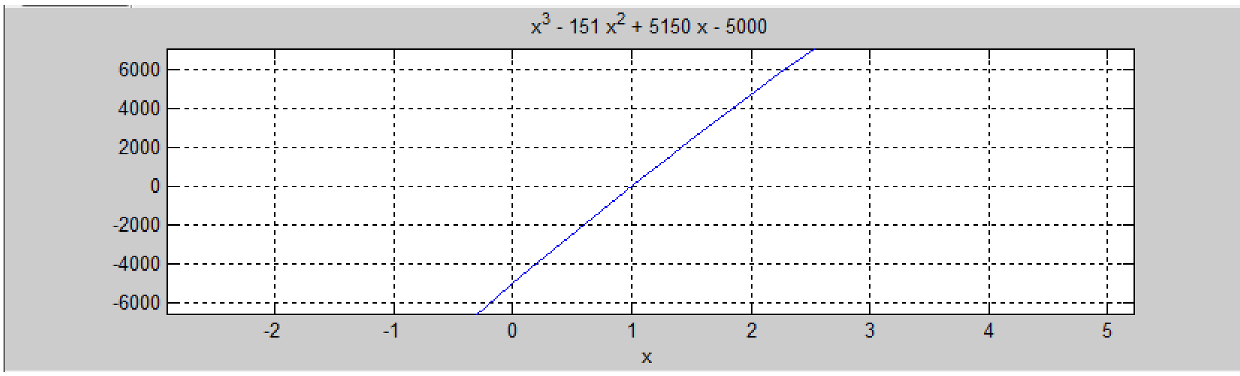

Figure 2.1: This figure shows that the root in area 1 of the function seems to be at (1,0)

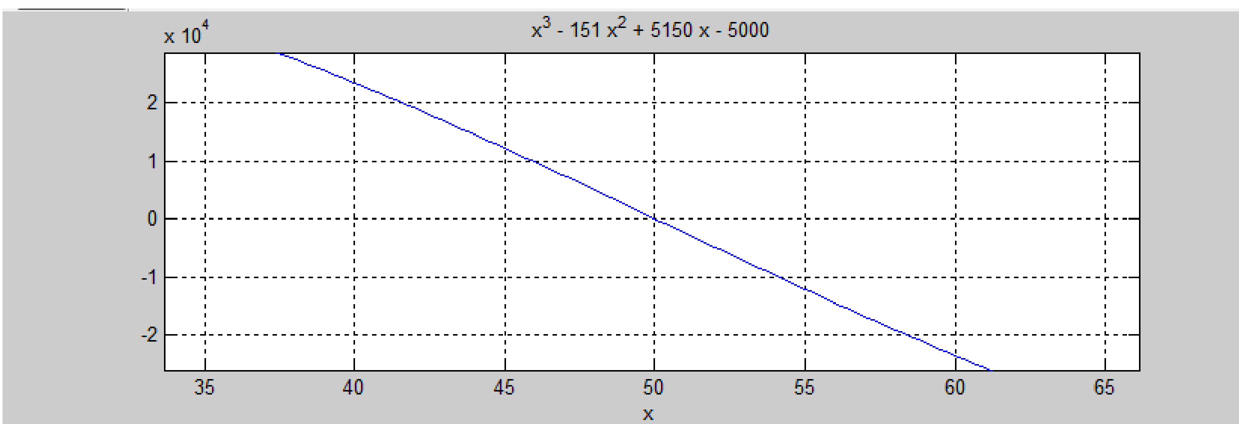

Figure 2.2: This figure shows that the root in area 2 of the function seems to be at (50,0)

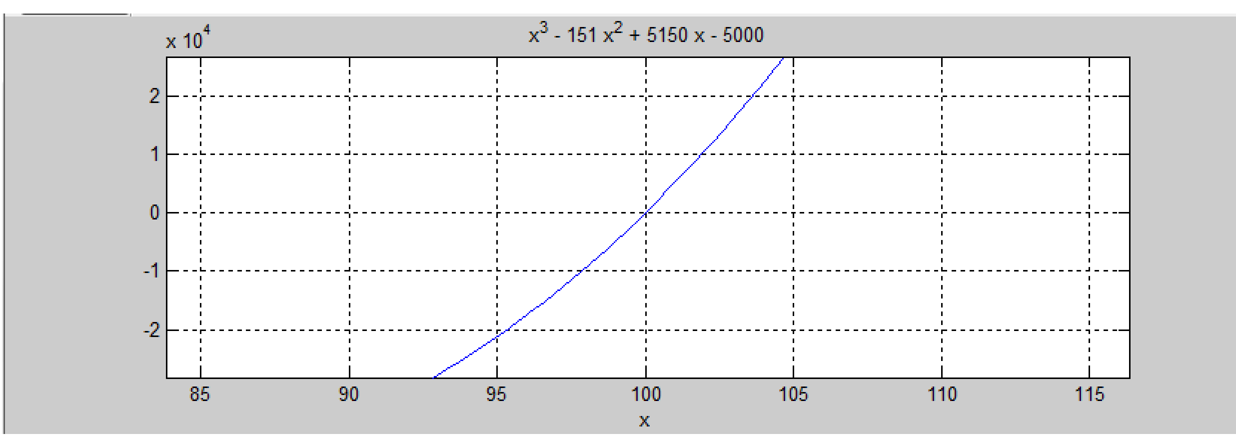

Figure 2.3: This figure shows that the root in area 3 of the function seems to be at (100,0)

III. Matlab code and outputs for the False Position Method used in solving the equation for the real roots:

1) Code for root at area 1 (a=-10, b=10)

```
% FALSE POSITION METHOD (Based on Bisection Method code)
% Find one positive root of f(x)=x^3-151*x^2+5150*x-5000
% Root is bracketed between [a,b]
% c is the initial estimate calculated from points a and b
    clc, clear, close
    a=-10; b=10; % initial guesses from graph f(x) vs x
    f=@(x) x^3-151*x^2+5150*x-5000; % f is a function handler
    c=b-f(b)*((a-b)/(f(a)-f(b))); % function to find new root estimate
    ainit=a; binit=b; %keep initial guesses as backup for reference
    er=abs(b-a);
    tol=1e-6; % error tolerance
    ii=1; % iteration counter
    ervect=zeros(1,ii); %creates a vector to store error values for each
                        %iteration
    iivect=zeros(1,ii); %creates a vector to store iteration number at a
                        %specific moment
    fprintf('%3s %7s %7s %7s %10s %10s %10s %10s \n',
    'ii','a','b','c','fa','fb','fc','er' ); % Table Title
    while er>= tol && ii<500
    fprintf('%3d %7.3f %7.3f %7.3f %10.2e %10.2e %10.2e %10.2e \n',ii,
    a, b, c, f(a), f(b), f(c), er);
    if f(a)*f(c)>0 % c substitutes a or b
    a=c;
    elseif f(a)*f(c)<0
    b=c;
    else
    root=c;
    break
    end
    c=(a+b)/2;
    er=abs(b-a);
    ervect(ii)=er; %stores current error value
    iivect(ii)=ii; %stores current iteration value
```

```
ii=ii+1;
end
% Among a, b, c find the closest to the root:
if abs(f(b))<abs(f(c)) && abs(f(b))<abs(f(a))
root=b;
elseif abs(f(c))<abs(f(a)) && abs(f(c))<abs(f(b))
root=c;
else
root=a
end
coeff=[1 -151 5150 -5000]; %creates a coefficient vector from f(x)
                        %to be used by 'roots' function in next line
roottrue=roots(coeff); %calculates the true roots and puts them in a
                        %vector named roottrue
error=abs(roottrue(3)-root); %calculates the error between true root
                             %and calculated root; roottrue(x) refers
                             %to the root to be compared in the
                             %roottrue vector
fprintf('\n The initial guess for a was %d \n', ainit);
fprintf(' The initial guess for b was %d \n', binit);
fprintf('\n The computed value of the root with the False Position
method code for the initial guesses is %e \n', root);
fprintf('\n The root was found in %d iterations \n', ii-1);
fprintf('\n The true values of the root is \n');
fprintf('%e \n',roottrue(3)); %index x of roottrue(x) shows the
                               %value of one of the solutions
fprintf('\n The error diference from computed and true value is %e',
error);
fplot (f,[-10,120]); %plots f(x) with x ranging from -10 to 120
```

MATLAB Output for root at area 1 (a=-10, b=10)

```
The initial guess for a was -10
The initial guess for b was 10

The computed value of the root with the False Point method code for the initial guesses is 9.999998e-01

The root was found in 25 iterations

The true values of the root is
1.000000e+00

The error diference from computed and true value is 1.873289e-07>> root

root =

    1.0000
```

MATLAB Output for root at area 2 (a=20, b=70)

```
The initial guess for a was 20
The initial guess for b was 70

The computed value of the root with the False Point method code for the initial guesses is 5.000000e+01

The root was found in 26 iterations

The true values of the root is
5.000000e+01

The error diference from computed and true value is 1.592883e-07>> root

root =

   50.0000
```

MATLAB Output for root at area 3 (a=80, b=120)

```
The initial guess for a was 80
The initial guess for b was 120

The computed value of the root with the False Point method code for the initial guesses is 1.000000e+02

The root was found in 26 iterations

The true values of the root is
1.000000e+02

The error diference from computed and true value is 6.517527e-08>>
```

Comments:

The function 'roots' was used to obtain the true values for the roots at every run of the code. This function takes the coefficients of a function (in this case, the coefficients are stored in the vector named 'coeff') and stores all its roots in a vector (in this case the vector 'roottrue'). The first value in the vector 'roottrue' is the highest, and the last value in this vector is the lowest. For each run, the index of the vector was changed everywhere in the code according to the solution that was being calculated; roottrue(1) contains the largest value (x=100), roottrue(2) contains the middle value (x=50) and roottrue(3) is the lowest value solution (x=1). Also, for each run of the code, the values for a and b were changed according to the area of the function where the roots were located.

IV. Comparison between calculated value (code) and true value (Matlab function 'roots')

Root #	Initial values for interval		Root value found with MATLAB code	Root value found with MATLAB function 'roots'	Error difference $=\lvert x_{true}-x_{comp}\rvert$
	a	b			
1	-10	10	1.0000	1.000000e+00	1.873289e-07
2	20	70	50.0000	5.000000e+01	1.592883e-07
3	80	120	100.0000	1.000000e+01	6.517527e-08

Comments:

The error difference is very small between using the code and using the 'roots' function to obtain the root values. This tells us that the False Position Method is very accurate for finding the value of the roots.