Assignment 3

Table of Contents

Problem 4

```
clc; close all; clear all;
i = 0;
upper_bound_error = 0.01;
approx_error_bi = [10000]; %placeholder since first approx. error = N/
A
evals_b = 0;

% Bisection method
f_lower = p(0.000001);
evals_b = evals_b + 1;
f_upper = p(1);
evals_b = evals_b + 1;

if f_lower == 0
    root = 0;
end
if f_upper == 0
    root = 1;
end
if f_upper*f_lower > 0
    fprintf('Cannot use bisection method if both brackets are the same
 sign.');
end
upper = 1;
lower = 0.000001;
root = .5*(upper + lower);
f_root = p(root);
evals_b = evals_b + 1;

while approx_error_bi(end) > 0.01
    i = i + 1;
    if f_lower*f_root == 0
        root = lower;
        break
    end
    if f_lower*f_root < 0
        upper = root;
    else
        lower = root;
    end
    f_lower = p(lower);
    evals_b = evals_b + 1;

    f_upper = p(upper);
    evals_b = evals_b + 1;

    oldroot = root;
    root = .5*(upper + lower);
    approx_error_bi = [approx_error_bi, abs((root-oldroot)/root*100)];
    f_root = p(root);
    evals_b = evals_b + 1;
end

fprintf('Using the bisection method: \n');
fprintf('a) The APR is %f%%\n', root*100*12);
fprintf('b) The number of iterations is %d\n', i);
fprintf('c) The total number of evaluations made: %d\n\n', evals_b);

% False Position Method
approx_error_fp = [10000];
ifp = 0;
evals_p = 0;

f_lower = p(0.000001);
evals_p = evals_p + 1;
f_upper = p(1);
evals_p = evals_p + 1;

if f_lower == 0
    root = 0;
end
if f_upper == 0
    root = 1;
end
if f_upper*f_lower > 0
    fprintf('Cannot use false position method if both brackets are the
 same sign.');
end
upper = 1;
lower = 0.000001;
root = .5*(upper + lower);
f_root = p(root);
evals_p = evals_p + 1;

while approx_error_fp(end) > 0.01
    ifp = ifp + 1;
    if f_lower*f_root == 0
        root = lower;
        break
    end
    if f_lower*f_root < 0
        upper = root;
    else
```

```
            lower = root;
        end

        f_lower = p(lower);
        evals_p = evals_p + 1;
        f_upper = p(upper);
        evals_b = evals_b + 1;

        oldroot = root;
        root = (upper + f_upper*(upper - lower)/(f_lower - f_upper));
        approx_error_fp = [approx_error_fp, abs((root-oldroot)/root*100)];
        f_root = p(root);
        evals_p = evals_p + 1;
    end

    fprintf('Using the false position method: \n');
    fprintf('a) The APR is %f%%\n', root*100*12);
    fprintf('b) The number of iterations is %d\n', ifp);
    fprintf('c) The total number of evaluations made: %d\n\n', evals_p);
```

Problem 5

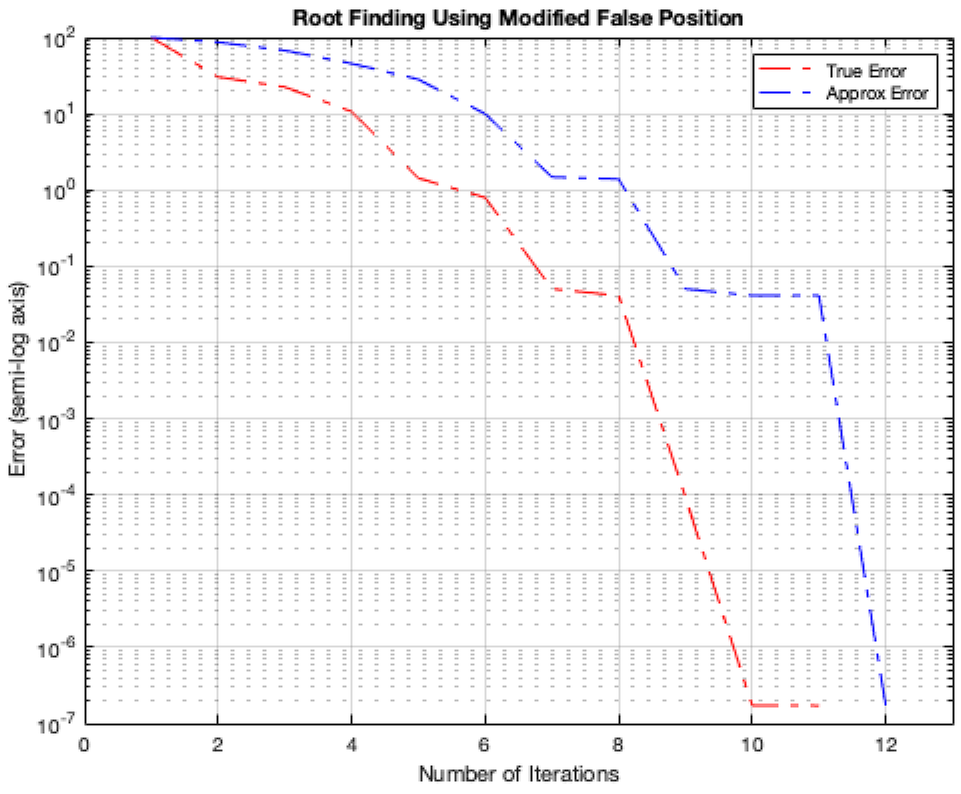
```
clc; close all; clear all;
wo = 2.3; %kN/cm
L = 750; %cm
E = 55000; %kN/cm^2
I = 25000; %cm^4
max_delta = -1;

f_lower = displacement(0);
f_upper = displacement(L - 1);
if f_lower == 0
    max_delta = 0;
end
if f_upper == 0
    max_delta = L - 0.01;
end
if f_upper*f_lower > 0
    fprintf('Cannot use bisection method if both brackets are the same
 sign.');
end
upper = 750;
lower = 0;

while 1
    root = .5*(upper + lower);
    f_root = displacement(root);
    if abs(f_lower*f_root) < 1*10^-15 % check if "equals 0"
        max_delta = lower;
        break
    end
    if f_lower*f_root < 0
        upper = root;
    else
        lower = root;
    end
end

fprintf('The maximum deflection occurs at %f cm\n', max_delta);
deflection = wo/(120*E*I*L)*(-(max_delta^5) + 2*L^2*(max_delta)^3 -
L^4*max_delta);
fprintf('The maximum deflection is %f cm\n', deflection);
```

```
The maximum deflection occurs at 335.410196 cm
The maximum deflection is -1.262362 cm
```

Problem 6

```
clc; close all; clear all;
upper_bound_error = 0.001;
true_error = [100];
approx_error = [100];
root = [];
i = 1;
iu = 0;
il = 0;

upper = 1.3;
lower = 0;
f_lower = f(lower);
f_upper = f(upper);

if f_lower == 0
    root = 0;
end
if f_upper == 0
    root = 1.3;
end
if f_upper*f_lower > 0
    fprintf('Cannot use false position method if both brackets are the
 same sign.');
end
root = [root, .5*(upper+lower)];
f_root = f(root(end));

while approx_error(end) > upper_bound_error
    i = i+1;
    if f_lower*f_root == 0
        root(end) = lower;
        break
    end
    if f_lower*f_root < 0
        upper = root(end);
        f_upper = f(upper);
        iu = 0;
        il = il + 1;
```

```
            if il >= 2
                f_lower = f_lower/2;
            end
        else
            lower = root(end);
            f_lower = f(lower);
            il = 0;
            iu = iu + 1;
            if iu >= 2
                f_upper = f_upper/2;
            end
        end

        oldroot = root(end);
        root = [root, upper + f_upper*(upper - lower)/(f_lower -
  f_upper)];
        ea1 = abs((root(end) - oldroot)/root(end)*100);
        eal = abs((root(end) - lower)/root(end)*100);
        eaf = abs((root(end) - upper)/root(end)*100);
        % The above 3 lines of code are necessary to make sure
        % the algorithm does not underestimate the error.
        approx_error = [approx_error, max([ea1, eal, eaf])];
        true_error = [true_error, abs((1-root(end))*100)];
        f_root = f(root(end));
    end

    fprintf('The root of the given function is at %f\n', root(end))

    semilogy(linspace(1, 12, 12), true_error, '-.r');
    grid on
    hold on
    semilogy(linspace(1, 12, 12), approx_error, '-.b');
    legend('True Error', 'Approx Error');
    title('Root Finding Using Modified False Position');
    xlabel('Number of Iterations');
    ylabel('Error (semi-log axis)');
    xlim([0, 13]);

    % Note: True Error is also a 1x12 vector, except the value of the true
    % error for the final root is equal to 0, which is not plotted on a
    % semilog plot.

    The root of the given function is at 1.000000
```



Function Calls from Each Section

Problem 4

```
function payment = p(i)
payment = 39190*i*(i+1)^72/((1+i)^72 - 1) - 587;
end

% Problem 5
function deltax = displacement(x)
wo = 2.3; %kN/cm
L = 750; %cm
E = 55000; %kN/cm^2
I = 25000; %cm^4
deltax = wo/(120*E*I*L)*(-5*x^4 + 6*L^2*x^2 - L^4);
end

% Problem 6
function y = f(x)
y = x^10 - 1;
end
```

Using the bisection method:
a) The APR is 2.516466%
b) The number of iterations is 22
c) The total number of evaluations made: 69

```
Using the false position method:
a) The APR is 2.516238%
b) The number of iterations is 12
c) The total number of evaluations made: 27
```

Published with MATLAB® R2020a