

Matthew LeDuc
APPM 5600
Fall 2023
HW 2

Question 1

Part i

Evaluating the function as $f(x) = (x - 5)^9$, the bisection code returns an approximate root of $x^* = 5$ in 10 iterations using the given parameters.

Part ii

If instead we evaluate the function using the binomial theorem as

$$f(x) = \sum_{n=0}^9 \binom{9}{n} x^n (-5)^{9-n}$$

the bisection code returns an approximate root of $x^* = 5.082$ in 10 iterations using the given parameters.

Part iii

The repeated truncation error from adding each term of the expanded form of the function leads to an error in the function evaluation, and this error changes where the algorithm will observe the root to be. Even though the actual root of the function is at $x^* = 5$, these repeated truncation errors add up in such a way that the evaluation of the function finds a root in the wrong place.

Code

The main script:

```
%% HW2_Problem1
fh1 = @(x) (x-5)^9;
fh2 = @(x) f1(x);
interval = [4.8, 5.31];
tol = 1e-4;
maxIters = 1000;
[root1, Niters1, errNum1] = bisection( fh1, interval, tol, maxIters );
[root2, Niters2, errNum2] = bisection( fh2, interval, tol, maxIters );

function y = f1(x)
y = zeros(size(x));
for ii = 0:9
```

```

y = y+nchoosek( 9, ii )*(x).^ii*(-5)^( 9-ii );
end
end

```

The bisection code:

```

function [root, cnt, errNum, errMsg] = bisection(fHandle, interval, tol, maxIters)
%BISECTION Implementation of bisection method
% Error messages:
% 0:Success
% 1:Root not found due to no detected sign change on interval
% 2:Root not found, max iterations reached
a = interval(1);b = interval(2);%
fa = fHandle(a);
fb = fHandle(b);
if fa*fb>0 %Handling the case where sign(a) and sign(b) are different
    newPts = linspace(a, b, 5);%checking if the initial choice of a and b
                                %were just bad
    for ii = 1:length(newPts)-1
        if fHandle(newPts(ii))*fHandle(newPts(ii+1))<0
            a = newPts(ii);%If we find a sign change in one of the new
                            %intervals we are cooking
            b = newPts(ii+1);
            fa = fHandle(a);
            fb = fHandle(b);
            rootExists = 1;
            break
        else
            rootExists = 0;
        end
    end
    if ~rootExists
        %If we don't find a sign change, output the midpoint
        root = (a+b)/2;
        cnt = 0;
        errNum = 1;
        errMsg = "Root not found: No sign change detected";
        return
    end
end
del = tol+1;
cnt = 0;
while del>tol && cnt<maxIters%Bisection method
    cnt = cnt+1;
    c = (a+b)/2;
    fc = fHandle(c);
    if fa*fc<0
        b = c;
        fb = fc;
    else
        a = c;
        fa = fc;
    end
end

```

```
        del = abs( (a-b)/b );  
end  
root = (a+b)/2;  
if cnt<maxIters  
    errNum = 0;  
    errMsg = "Successful";  
else  
    errNum = 2;  
    errMsg = "Root not found: Maximum iterations reached";  
end  
end
```