

Assignment 1

0.1 Exercises: 6

Q:

Explain how to evaluate the polynomial for a given input x , using as few operations as possible. How many multiplications and how many additions are required?

(a) $P(x) = a_0 + a_5x^5 + a_{10}x^{10} + a_{15}x^{15}$

(b) $P(x) = a_7x^7 + a_{12}x^{12} + a_{17}x^{17} + a_{22}x^{22} + a_{27}x^{27}$

A:

Use the method *Nested Multiplication* to rewrite the polynomial so that it can be evaluated from the inside out:

(a)

$$P(x) = a_0 + x^5(a_5 + x^5(a_{10} + x^5(a_{15})))$$

15 multiplications and 3 additions are required.

(b)

$$P(x) = x^7(a_7 + x^5(a_{12} + x^5(a_{17} + x^5(a_{22} + x^5(a_{27}))))))$$

27 multiplications and 4 additions are required.

0.1 Computer Problems: 1

Q:

Use the function `nest` to evaluate $P(x) = 1 + x + \dots + x^{50}$ at $x = 1.00001$. (Use the MATLAB ones command to save typing.) Find the error of the computation by comparing with the equivalent expression $Q(x) = (x^{51} - 1)/(x - 1)$.

A:

code:

nest.m:

```

%Program 0.1 Nested multiplication
%Evaluates polynomial from nested form using Horner's Method
%Input: degree d of polynomial,
% array of d+1 coefficients c (constant term first),
% x-coordinate x at which to evaluate, and
% array of d base points b, if needed
%Output: value y of polynomial at x
function y=nest(d,c,x,b)
if nargin<4, b=zeros(d,1); end
y=c(d+1);
for i=d:-1:1
y = y.*(x-b(i))+c(i);
end

```

Q.m:

```

function Q=Q(x)
Q=(x^(51)-1)/(x-1);

```

run.m:

```

nest(50,ones([1,51]),1.00001,zeros([1,51]))
nest(50,ones([1,51]),1.00001,zeros([1,51]))-Q(1.00001)

```

Results:

```

ans =

51.012752082749991

```

```

ans =

4.760636329592671e-12

```

P(x)=51.012752082749991

Absolute Error = 4.760636329592671e-12

0.4 Exercises: 2

Q:

Find the roots of the equation $x^2 + 3x - 8^{-14} = 0$ with three-digit accuracy.

A:

The roots of a quadratic equation of form $ax^2 + bx + c = 0$ are given by the quadratic formula: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

For our problem, this translates to $x = \frac{-3 \pm \sqrt{9 + 4 * (8^{-14})}}{2}$.

Using the minus sign gives the root $x_1 = -3.000$.

Using the plus sign gives the root $x_2 = 7.572 * 10^{-14}$.

However, the loss of significance may appear, so we restructure the quadratic formula: $x_2 =$

$$\frac{-3 + \sqrt{9 + 4 * (8^{-14})}}{2} = \frac{2 * 8^{-14}}{\sqrt{9 + 4 * 8^{-14}} + 3} = 7.579 * 10^{-14}.$$

$$\therefore x_1 = -3.000, x_2 = 7.579 * 10^{-14}$$

0.4 Computer Problems: 1

Q:

Calculate the expressions Solutions for Computer Problems that follow in double precision arithmetic (using MATLAB, for example) for $x = 10^{-1}, \dots, 10^{-14}$. Then, using an alternative form of the expression that doesn't suffer from subtracting nearly equal numbers, repeat the calculation and make a table of results. Report the number of correct digits in the original expression for each x.

- (a) $\frac{1 - \sec(x)}{\tan^2(x)}$
(b) $\frac{1 - (1-x)^3}{x}$

A:

(a)

$$\frac{1 - \sec(x)}{\tan^2(x)} = \frac{(1 - \sec(x))(1 + \sec(x))}{(\tan^2(x))(1 + \sec(x))} = -\frac{1}{1 + \sec(x)}$$

code:

get_data.m:

```
function [a,b,c]=get_data(x)
%%a is x.
%%b is the result of the original expression at x.
%%c is the result of the alternative form of the expression at x.
a=x;
b=(1-sec(x))/(tan(x))^2;
c=-1/(1+sec(x));
```

main.m:

```

format long
%%get 1^(-1) to 1^(-14)
for i=1:1:14
    x(i)=10^(-i);
end
%%get the results and save them in data
data=ones(14,3);
for i=1:1:14
    [data(i,1),data(i,2),data(i,3)]=get_data(x(i));
end
data

```

results:

```

data =

0.10000000000000000 -0.498747913711435 -0.498747913711429
0.01000000000000000 -0.499987499790956 -0.499987499791664
0.00100000000000000 -0.499999875014289 -0.499999874999979
0.00010000000000000 -0.499999993627931 -0.499999998750000
0.00001000000000000 -0.500000041336852 -0.499999999987500
0.00000100000000000 -0.500044450290837 -0.49999999999875
0.00000010000000000 -0.510702591327569 -0.499999999999999
0.00000001000000000 0 -0.500000000000000
0.00000000100000000 0 -0.500000000000000
0.00000000010000000 0 -0.500000000000000
0.00000000001000000 0 -0.500000000000000
0.00000000000100000 0 -0.500000000000000
0.00000000000010000 0 -0.500000000000000
0.00000000000001000 0 -0.500000000000000
0.00000000000000100 0 -0.500000000000000

```

(b)

$$\frac{1-(1-x)^3}{x} = \frac{(1-x+x)-(1-x)^3}{x} = \frac{(1-x)(1-(1-x)^2)+x}{x} = 1 + \frac{(1-x)(2x-x^2)}{x} = x^2 - 3x + 3$$

code:

get_data.m:

```

function [a,b,c]=get_data(x)
%%a is x.
%%b is the result of the original expression at x.
%%c is the result of the alternative form of the expression at x.
a=x;
b=(1-(1-x)^(3))/x;
c=x^(2)-3*x+3;

```

main.m:

```

format long
%%get 1^(-1) to 1^(-14)
for i=1:1:14
    x(i)=10^(-i);
end
%%get the results and save them in data
data=ones(14,3);
for i=1:1:14
    [data(i,1),data(i,2),data(i,3)]=get_data(x(i));
end
data

```

results:

```

data =

    0.100000000000000    2.709999999999999    2.710000000000000
    0.010000000000000    2.970099999999998    2.970100000000000
    0.001000000000000    2.997000999999999    2.997001000000000
    0.000100000000000    2.999700010000161    2.999700010000000
    0.000010000000000    2.999970000083785    2.999970000100000
    0.000001000000000    2.999997000041610    2.999997000001000
    0.000000100000000    2.999999698660716    2.999999700000010
    0.000000010000000    2.999999981767587    2.999999970000000
    0.000000001000000    2.999999915154206    2.999999970000000
    0.000000000100000    3.000000248221113    2.999999999700000
    0.000000000010000    3.000000248221113    2.999999999970000
    0.000000000001000    2.999933634839635    2.999999999997000
    0.000000000000100    3.000932835561798    2.999999999999700
    0.000000000000010    2.997602166487923    2.999999999999970

```

1.1 Exercises: 5

Q:

Consider the equation $x^4 = x^3 + 10$.

(a) Find an interval $[a,b]$ of length one inside which the equation has a solution.

(b) Starting with $[a,b]$, how many steps of the Bisection Method are required to calculate the solution within 10^{-10} ? Answer with an integer.

A:

(a)

Let $f(x) = x^4 - x^3 - 10$

Then, $f(0) = -10$, $f(1) = -10$, $f(2) = -2$, $f(3) = 44$

$\therefore f(2) < 0$ and $f(3) > 0$

$$\therefore x_0 \in [2, 3]$$

(b)

$$\frac{1}{2^{n+1}} < 0.5 * 10^{-10}$$

$$n > \frac{10}{\log_{10} 2} \approx 33.2$$

\therefore 34 steps are required.

1.1 Computer Problems: 2

Q:

Use the Bisection Method to find the root to eight correct decimal places.

(a) $x^5 + x = 1$

(b) $\sin x = 6x + 5$

(c) $\ln x + x^2 = 3$

A:

(a)

code:

bisect.m:

```

%Program 1.1 Bisection Method
%Computes approximate solution of f(x)=0
%Input: function handle f; a,b such that f(a)*f(b)<0,
% and correct decimal places
%Output: Approximate solution xc

function xc=bisect(f,a,b,d)
if sign(f(a))*sign(f(b)) >= 0
    error('f(a)f(b)<0 not satisfied!') %ceases execution
end
n = ceil(d/log10(2)); %1/(2^(n+1))<0.5*10^(-d)
i=0;
fa=f(a);
fb=f(b);

while i<=n
    i=i+1;
    c=(a+b)/2;
    fc=f(c);
    if fc == 0 %c is a solution, done
        break
    end
    if sign(fc)*sign(fa)<0 %a and c make the new interval
        b=c;fb=fc;
    else %c and b make the new interval
        a=c;fa=fc;
    end
end
xc=(a+b)/2; %new midpoint is best estimate

```

main.m:

```

f=@(x)x^5+x-1;
xc=bisect(f,0,1,8);
round(xc,8)

```

results:

```

ans =

    0.754877670000000

```

$\therefore x_0 = 0.75487767$

(b)

code:

bisect.m: same as (a)

main.m:

```
f=@(x)sin(x)-6*x-5;  
xc=bisect(f,-1,0,8);  
round(xc,8)
```

results:

ans =

-0.970898920000000

$\therefore x_0 = 0.97089892$

(c)

code:

bisect.m: same as (a)

main.m:

```
f=@(x)(cos(x))^2+6-x;  
xc=bisect(f,6,7,8);  
round(xc,8)
```

results:

ans =

6.776092320000000

$\therefore x_0 = 6.77609232$

1.3 Exercises: 8

Q:

Let $f(x) = x^n - ax^{n-1}$, and set $g(x) = x^n$.

(a) Use the Sensitivity Formula to give a prediction for the nonzero root of $f(x) = x^n - ax^{n-1} + \epsilon x^n$ for small ϵ .

(b) Find the nonzero root and compare with the prediction.

A:

(a)

$$r = a$$

$$\Delta r = -\frac{\epsilon g(r)}{f'(r)} = -\frac{\epsilon r^n}{nr^{n-1} - a(n-1)r^{n-2}} = -\epsilon a$$

$$\therefore r + \Delta r = a(1 - \epsilon)$$

(b)

$$x^n - ax^{n-1} + \epsilon x^n = 0$$

$$x^{n-1}(-a + x + \epsilon x) = 0$$

We are looking for nonzero root, so $-a + x + \epsilon x = 0$

$$\therefore x_0 = a\left(\frac{1}{1+\epsilon}\right)$$

$$\therefore \frac{1}{1+\epsilon} = 1 - \epsilon + \epsilon^2 + \dots (|\epsilon| < 1)$$

\therefore the prediction root $a(1 - \epsilon)$ is actually the the first term of Taylor expansion of the nonzero root $a\left(\frac{1}{1+\epsilon}\right)$.

1.3 Computer Problems: 3

Q:

(a) Use fzero to find the root of $f(x) = 2x\cos x - 2x + \sin x^3$ on $[-0.1, 0.2]$. Report the forward and backward errors.

(b) Run the Bisection Method with initial interval $[-0.1, 0.2]$ to find as many correct digits as possible, and report your conclusion.

A:

(a)

code:

```
f=@(x)2*x*cos(x)-2*x+sin(x^3);
fzero(f,[-0.1,0.2])
```

results:

```
ans =
```

```
1.688148348885743e-04
```

forward error= $|x_a - r| = 1.69 * 10^{-4}$

backward error= $f(x_a) - 0 = 3.368 * 10^{-11}$ (???)

(b)

code:

bisect.m:

```

%Program 1.1 Bisection Method
%Computes approximate solution of  $f(x)=0$ 
%Input: function handle f; a,b such that  $f(a)*f(b)<0$ ,
% and tolerance tol
%Output: Approximate solution xc

function [i,xc]=bisect(f,a,b,tol)
if sign(f(a))*sign(f(b)) >= 0
    error('f(a)f(b)<0 not satisfied!') %ceases execution
end

i=0;
fa=f(a);
fb=f(b);
while (b-a)/2>tol
    i=i+1;
    c=(a+b)/2;
    fc=f(c);
    if fc == 0 %c is a solution, done
        break
    end
    if b-a<2^(-52)%the change is now not obvious
        break
    end
    if sign(fc)*sign(fa)<0 %a and c make the new interval
        b=c;fb=fc;
    else %c and b make the new interval
        a=c;fa=fc;
    end
end
xc=(a+b)/2; %new midpoint is best estimate

```

main.m:

```

f=@(x)2*x*cos(x)-2*x + sin(x^3);
[times,xc]=bisect(f,-0.1,0.2,2^(-52));
times
round(xc,8)

```

results:

times =

13

ans =

-6.104000000000000e-05

After 13 calculations, we find as many correct digits as possible.

The answer is $-6.104 * 10^{-5}$.

1.4 Exercises: 1

Q:

Apply two steps of Newton's Method with initial guess $x_0 = 0$.

(a) $x^3 + x - 2 = 0$

(b) $x^4 - x^2 + x - 1 = 0$

(c) $x^2 - x - 1 = 0$

A:

(a)

$$f(x) = x^3 + x - 2$$

$$f'(x) = 3x^2 + 1$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{-2}{1} = 2$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 2 - \frac{8}{13} = \frac{18}{13}$$

(b)

$$f(x) = x^4 - x^2 + x - 1 = 0$$

$$f'(x) = 4x^3 - 2x + 1$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{-1}{1} = 1$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 1 - \frac{0}{3} = 1$$

(c)

$$f(x) = x^2 - x - 1 = 0$$

$$f'(x) = 2x - 1$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{-1}{-1} = -1$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = -1 - \frac{1}{-3} = -\frac{2}{3}$$

1.4 Computer Problems: 1

Q:

Each equation has one root. Use Newton's Method to approximate the root to eight correct decimal places.

(a) $x^3 = 2x + 2$

(b) $e^x + x = 7$

(c) $e^x + \sin x = 4$

A:

(a)

code:

Gauss.m:

```
function x=Gauss(f,df,a)
b=a-f(a)/df(a);
while abs(a-b)>10^(-9)%% answer is accurate enough
    a=b;
    b=b-f(b)/df(b);
end
x=b;
```

main.m:

```
format long
f=@(x) x^3-2*x-2;
df=@(x) 3*x^2-2;
round(Gauss(f,df,1),8)
```

results:

ans =

1.769292350000000

$$x_0 = 1.76929235$$

(b)

code:

Gauss.m: same as (a)

main.m:

```
format long
f=@(x) exp(x)+x-7;
df=@(x) exp(x)+1;
round(Gauss(f,df,1),8)
```

results:

```
ans =
```

```
1.672821700000000
```

$$x_0 = 1.67282170$$

(c)

code:

Gauss.m: same as (a)

main.m:

```
format long
f=@(x) exp(x)+sin(x)-4;
df=@(x) exp(x)+cos(x);
round(Gauss(f,df,1),8)
```

results:

```
ans =
```

```
1.129980500000000
```

$$x_0 = 1.12998050$$

1.5 Exercises: 1

Q:

Apply two steps of the Secant Method to the following equations with initial guesses $x_0 = 1$ and $x_1 = 2$.

(a) $x^3 = 2x + 2$

(b) $e^x + x = 7$

(c) $e^x + \sin x = 4$

A:

$$x_0 = 1$$

$$x_1 = 2$$

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

(a)

$$f(x) = x^3 - 2x - 2$$

$$x_2 = 2 - \frac{2 \cdot 1}{2 - (-3)} = \frac{8}{5}$$

$$x_3 = \frac{8}{5} - \frac{((\frac{8}{5})^3 - 2 \cdot \frac{8}{5} - 2) \cdot (\frac{8}{5} - 2)}{((\frac{8}{5})^3 - 2 \cdot \frac{8}{5} - 2) - 2} = 1.74227$$

(b)

$$f(x) = e^x + x - 7$$

$$x_2 = 1.57871$$

$$x_3 = 1.66016$$

(c)

$$f(x) = e^x + \sin x - 4$$

$$x_2 = 1.09291$$

$$x_3 = 1.11936$$

1.5 Computer Problems: 1

Q:

Use the Secant Method to find the (single) solution of each equation in Exercise 1.

A:

(a)

code:

Secant.m:

```
%% f is the function, a=x0, b=x1
function x=Secant(f,a,b)
c = b-(f(b)*(b-a))/(f(b)-f(a));
while abs(b-a)>2^(-52) %% answer is accurate enough
    a=b;
    b=c;
    c = b-(f(b)*(b-a))/(f(b)-f(a));
end
x=b;
```

main.m:

```
x0=1;  
x1=2;  
f=@(x) x^3-2*x-2;  
Secant(f,x0,x1)
```

results:

```
ans =
```

```
1.769292354238631
```

$x_0 = 1.76929235$

(b)

code:

Secant.m: same as (a)

main.m:

```
x0=1;  
x1=2;  
f=@(x) exp(x)+x-7;  
Secant(f,x0,x1)
```

results:

```
ans =
```

```
1.672821698628907
```

$x_0 = 1.67282170$

(c)

code:

Secant.m: same as (a)

main.m:

```
x0=1;  
x1=2;  
f=@(x) exp(x)+sin(x)-4;  
Secant(f,x0,x1)
```

results:

ans =

1.129980498650832

$$x_0 = 1.12998050$$