Sketches of Solutions

Assignment 2 Roots of Equations

$f(x) = \sqrt[4]{13}$

$x - \sqrt[4]{13}$

1. Use Bisection method to find the root of $\sqrt[4]{13}$ in the interval of 1.5 and 2.0

$x^4 - \sqrt[4]{13} = 0$

1.1 show your work for 4 iterations

$x^4 - 13 = 0$

Rearrange the given equation to

$$x^4 - \left(\sqrt[4]{13}\right)^4 = 0 \qquad (1)$$

$$x^4 - 13 = 0 \qquad (2)$$

From equation (2), we can apply Bisection method to approximate its root as follow

X_L = 1.5, X_R = 2.0, ε_S = 0.000001

Iteration	X_L	X_R	X_M	$f(X_M)$	$f(X_R)$	$f(X_M)*f(X_R)$	ε
1	1.5000000	2.000000	1.750000	-3.621094	3.000000	-10.863282	0.500000
2	1.750000	2.000000	1.875000	-0.640381	3.000000	-1.921143	0.333333
3	1.875000	2.000000	1.937500	1.091812	3.000000	3.275436	0.142857
4	1.875000	1.937500	1.906250	0.204423	1.091812	0.223191	0.066667
5	1.890625	1.906250	1.898438	-0.010702	0.204423	-0.002188	0.016393
…	…	…	…	…	…	…	…
20	1.898828	1.898829	1.898829	0.000002	0.000002	0.000000	0.000001

1.2 write a program to iteratively find the answer till there is no changes in the answer for six significant figures

```
// Node.js (JavaScript)
// Bisection Method
var xl, xr, xm, f_xr, f_xl, f_xm, esp, approx_err, true_err, i;
var error, prev_xm, exact_sol = 13**(1/4);
// equation: x^4 - 13 = 0
esp = 0.0000001;
xl = 1.5;
xr = 2.0;
xm = (xl+xr)/2;
```

var xl, xr, xm, f_xr, f_xl, f_xm, esp, approx_err, true_err, i;

var error, prev_xm, exact_sol = 13**(1/4);

esp = 0.0000001;

xl = 1.5

xr = 2.0

xm = (xl+xr)/2;

xl = 1.

(Math.pow(xr, 4) *)4 - 13;

i=0

f_xr = (xr ** 4) - 13;

f_xm = (xm**4) - 13;

true_err = 100* Math.abs((exact_sol-xm) / exactsol

console.log ('tre error = ' +true_err);

prev_xm = xm;

if(f_xr*f_xm <0) xl = xm;

else xr = xm;

i = 1;

var

```
i = 0;

console.log('iteration ' + i + ': xr = ' + xr + ', xl = ' + xl + ', xm = ' + xm);

f_xr = (xr**4) - 13;

f_xm = (xm**4) - 13;

true_err = 100*Math.abs((exact_sol -xm)/exact_sol);

console.log('tre error = ' + true_err);

prev_xm = xm;

if(f_xr*f_xm < 0)   xl = xm;

else   xr = xm;

i = 1;

do {   // determine f(x)

   xm = (xl+xr)/2;

   f_xr = (xr**4) - 13;

   f_xm = (xm**4) - 13;

   true_err = 100*Math.abs(exact_sol -xm)/exact_sol);

   approx_err = 100*Math.abs((xm - prev_xm)/xm);

   if(f_xr*f_xm < 0)   xl = xm;

   else   xr = xm;

   error = prev_xm - xm;

   prev_xm = xm;

   console.log('iteration ' + i + ': xr = ' + xr + ', xl = ' + xl + ', xm = ' + xm);

   console.log('true_err = ' + true_err);

   console.log('approx_err = ' + approx_err);

   i++;

} while(Math.abs(error) > esp);
```

2. Use False-Position method to find the value of $\frac{1}{43}$ in the interval of 0.02 and 0.03

$x = \frac{1}{43}$

2.1 show your work for 4 iterations

Rearrange the given equation to

$x - \frac{1}{43}$

$$\frac{1}{x} - 43 = 0 \qquad (1)$$

From equation (1), we can apply False position method to approximate its root as follow

X_L = 0.02, X_R = 0.03, ε_S = 0.000001%

Iteration	X_L	X_R	$f(X_R)$	X_1	$f(X_1)$	$f(X_1)* f(X_R)$	$\varepsilon(\%)$
1	0.020000	0.030000	-9.666667	0.024200	-1.677686	16.217632	23.966942
2	0.020000	0.024200	-1.677686	0.023388	-0.243031	0.407730	3.471866
3	0.020000	0.023388	-0.243031	0.023274	-0.033600	0.008166	0.489817
4	0.020000	0.023274	-0.033600	0.023258	-0.004042	0.000136	0.068794
5	0.020000	0.023258	-0.004042	0.023256	-0.000344	0.000001	0.008600
6	0.020000	0.023256	-0.000344	0.023256	-0.000344	0.000000	0.000000

2.2 write a program provided that the error is lesser than 0.000001%

```
// Node.js (JavaScript)
// False-Position Method
var xl, xr, x1, f_xr, f_xl, f_x1, esp, approx_err, true_err, i;
var error, prev_x1, exact_sol = 1/43;
// equation: (1/x) - 43 = 0
esp = 0.0000001;
xl = 0.02;
xr = 0.03;
f_xr = (1/xr) - 43;
f_xl = (1/xl) - 43;
x1 = (xl*f_xr - xr*f_xl)/(f_xr-f_xl);
```

```
f_x1 = (1/x1) - 43;
i = 0;
console.log('iteration ' + i + ': xr = ' + xr + ', xl = ' + xl + ', x1 = ' + x1);
true_err = 100*Math.abs((exact_sol - x1)/exact_sol);
console.log('tre error = ' + true_err);
prev_x1 = x1;
if(f_xr*f_x1 < 0)   xl = x1;
else   xr = x1;
i = 1;
do { // determine f(x)
    f_xr = (1/xr) - 43;
    f_xl = (1/xl) - 43;
    x1 = (xl*f_xr - xr*f_xl)/(f_xr-f_xl);
    f_x1 = (1/x1) - 43;
    true_err = 100*Math.abs((exact_sol - x1)/exact_sol);
    approx_err = 100*Math.abs((x1 - prev_x1)/x1);
    if(f_xr*f_x1 < 0)   xl = x1;
    else   xr = x1;
    prev_x1 = x1;
    console.log('iteration ' + i + ': xr = ' + xr + ', xl = ' + xl + ', x1 = ' + x1);
    console.log('true_err = ' + true_err);
    console.log('approx_err = ' + approx_err);
    i++;
} while(approx_err > esp);
```

3. Use ONE-POINT ITERATION method to find the value of $\frac{1}{2}$ with an initial guess of 0.00

Sketches of Solutions

3.1 show your work for 4 iterations

Rearrange the given equation to

$$x - \frac{1}{2} = 0 \qquad (1)$$

$$(x - \tfrac{1}{2})(x - \tfrac{1}{2})$$

$$x = x^2 + \frac{1}{4} \qquad (2)$$

$$x^2 - \frac{x}{2} - \frac{x}{2} + \frac{1}{4}$$

Then rewrite equation (2) into iterative form

$$x_{i+1} = x_i^{\,2} + \frac{1}{4} \qquad (3)$$

From equation (3), we can apply One-point iteration method to approximate its root as follow

Initial guess; X = 0.00, ε_S = 0.000001%

$$x^2 - \frac{\cancel{2}x}{\cancel{2}} + \frac{1}{4}$$

Iteration	X_i	X_{i+1}	ε_a(%)
1	0.000000	0.250000	N/A
2	0.250000	0.312500	20.000000
3	0.312500	0.347656	10.112360
4	0.347656	0.370865	6.257972
5	0.370865	0.387541	4.303001
…	…	…	…
11538	0.499913	0.499913	0.000001

$$x^2 - x + \frac{1}{4}$$

$$x = x^2 + \frac{1}{4}$$

Note that the convergence of the solution depends on how we formulate the equation. If we formulate as $x - \frac{1}{2} = 0$, the solution will be diverge since there is no term related to x on the right-hand side of the equation.

Iteration	X_i	X_{i+1}	ε_t (%)	ε_a (%)
1	0.000000	-0.500000	200%	N/A
2	-0.500000	-1.000000	300%	50.000000%
3	-1.000000	-1.500000	400%	33.333333%
4	-1.500000	-2.000000	500%	25.000000%
5	-2.000000	-2.500000	600%	20.000000%
…	…	…		…
350	-174.500000	-175.000000	35100%	0.285714%

3.2 write a program with error lesser than 0.000001%

```
// Node.js (JavaScript)
```

```
// One-Point Iteration Method
var x, new_x, i = 0;
var approx_err, eps;
// equation new_x = x^2 + 1/4
eps = 0.000001;
// initial guess : x = 0
x = 0.00;
new_x = x**2 + (1/4);
approx_err = 100*Math.abs((new_x - x)/new_x);
console.log('iteration ' + i + ': x_i = ' + x + ', x_i+1 = ' + new_x + ', approx_err = ' + approx_err);
do {
    i++;
    x = new_x;
    new_x = x**2 + (1/4);
    approx_err = 100*Math.abs((new_x - x)/new_x);
    console.log('iteration ' + i + ': x_i = ' + x + ', x_i+1 = ' + new_x + ', approx_err = ' +
approx_err);
} while(approx_err > eps);
```