1. (a)

$$|g_1'(x)| = |\frac{2x}{3}| \Rightarrow |g_1'(2)| = \frac{4}{3} > 1$$

$$|g_2'(x)| = |\frac{3}{2\sqrt{3x-2}}| \Rightarrow |g_2'(2)| = \frac{3}{4} < 1$$

$$|g_3'(x)| = |\frac{2}{x^2}| \Rightarrow |g_3'(2)| = \frac{1}{2} < 1$$

$$|g_4'(x)| = |\frac{2x(2x-3) - 2(x^2-2)}{(2x-3)^2}| \Rightarrow |g_4'(2)| = \frac{4-4}{1} = 0$$

Then we know that $|g_1'(2)|$ is bigger than 1, which means divergence. $|g_2'(2) = \frac{3}{4}|$ means it is linear convergence with constant $|\frac{3}{4}|$. $|g_3'(2) = \frac{1}{2}|$ means it is linear convergence with constant $|\frac{1}{2}|$. And $|g_4'(2) = 0|$ means it is quadratic convergence.

(b)

```
function fixpoint % fixed-point iteration
    fs = {'(x*x+2)/3' 'sqrt(3*x-2)' '3-2/x' '(x*x-2)/(2*x-3)'};
    tol = 0.01;
    x_rel = 2;
    for i = 1 : 4
        fprintf('g_%g(x) = %s\n', i, fs{i});
        g = inline(fs{i},'x');
        disp('  k               x              err                ratio');
        k = 0;
        x = 3.0;
        err = abs(x-x_rel);
        fprintf('%3d %17e %17e\n', k, x, err);
        while err > tol && k < 12;
            k = k+1;
            x = g(x);
            err_new = abs(x-x_rel);
            ratio = err_new/err;
            err = err_new;
            fprintf('%3d %17e %17e %17e\n', k, x, err, ratio);
        end;
        disp(' ');
    end
```

```
g_1(x) = (x*x+2)/3
  k            x               err              ratio
  0     3.000000e+00     1.000000e+00
  1     3.666667e+00     1.666667e+00     1.666667e+00
  2     5.148148e+00     3.148148e+00     1.888889e+00
  3     9.501143e+00     7.501143e+00     2.382716e+00
  4     3.075724e+01     2.875724e+01     3.833714e+00
  5     3.160026e+02     3.140026e+02     1.091908e+01
  6     3.328655e+04     3.328455e+04     1.060009e+02
  7     3.693315e+08     3.693315e+08     1.109618e+04
  8     4.546858e+16     4.546858e+16     1.231105e+08
  9     6.891304e+32     6.891304e+32     1.515619e+16
 10     1.583003e+65     1.583003e+65     2.297101e+32
 11     8.352990e+129    8.352990e+129    5.276675e+64
 12     2.325748e+259    2.325748e+259    2.784330e+129

g_2(x) = sqrt(3*x-2)
  k            x               err              ratio
  0     3.000000e+00     1.000000e+00
  1     2.645751e+00     6.457513e-01     6.457513e-01
  2     2.436648e+00     4.366481e-01     6.761862e-01
  3     2.304332e+00     3.043316e-01     6.969723e-01
  4     2.216528e+00     2.165277e-01     7.114859e-01
  5     2.156289e+00     1.562892e-01     7.217977e-01
  6     2.113970e+00     1.139696e-01     7.292227e-01
  7     2.083725e+00     8.372475e-02     7.346235e-01
  8     2.061838e+00     6.183759e-02     7.385820e-01
  9     2.045853e+00     4.585258e-02     7.415001e-01
 10     2.034099e+00     3.409875e-02     7.436605e-01
 11     2.025413e+00     2.541261e-02     7.452652e-01
 12     2.018970e+00     1.896950e-02     7.464600e-01

g_3(x) = 3-2/x
  k            x               err              ratio
  0     3.000000e+00     1.000000e+00
  1     2.333333e+00     3.333333e-01     3.333333e-01
  2     2.142857e+00     1.428571e-01     4.285714e-01
  3     2.066667e+00     6.666667e-02     4.666667e-01
  4     2.032258e+00     3.225806e-02     4.838710e-01
  5     2.015873e+00     1.587302e-02     4.920635e-01
  6     2.007874e+00     7.874016e-03     4.960630e-01

g_4(x) = (x*x-2)/(2*x-3)
  k            x               err              ratio
  0     3.000000e+00     1.000000e+00
  1     2.333333e+00     3.333333e-01     3.333333e-01
  2     2.066667e+00     6.666667e-02     2.000000e-01
  3     2.003922e+00     3.921569e-03     5.882353e-02
```

Clearly, the converging rate is approximately like what we calculated.

2. (a) Starting with the secant method update formula is given as

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

$$= \frac{x_k(f(x_k) - f(x_{k-1})) - f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

$$= \frac{x_k f(x_k) - x_k f(x_{k-1}) - x_k f(x_k) + x_{k-1} f(x_k)}{f(x_k) - f(x_{k-1})}$$

$$= \frac{x_{k-1} f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

(b) When we close to solution then x_{k-1} and x_k are close to each other which mean there difference is close to 0, and also can cause catastrophic cancellation. For formula in $part(a)$ it is hard for us to get rid of cancellation. For formula in $part(b)$, catastrophic cancellation is the only thing affecting x_{k+1}

3.

```
function bisection
% xr: rel solu from fzero
% fs: input functions
% [as,bs]: input interval
diary result.out
fs = {'x*x*x-2*x-5' 'exp(-x)-x' 'x*sin(x)-1' 'x*x*x - 3*x*x + 3*x - 1'};
as = {1 -0.4 1 0};
bs = {6  0.8 2 3};

tol = 0.01;
for k = 1:4
    i = 1;
    a = as{k};
    b = bs{k};
    fprintf('g_%g(x) = %s\n', k, fs{k});
    f = inline(fs{k},'x');
    disp(sprintf('%3s %17s %17s %17s %17s', 'k', 'a','f(a)','b','f(b)'));
    disp(sprintf('%3d %17f %17.12f %17.12f %17.12f', i, a, f(a), b, f(b)));
    if f(a) * f(b) > 0
        disp(sprintf('Wrong Interval [%d, %d]', a, b))
    else
        while (b-a) > tol && i<50;
            i = i+1;
            m = a + (b-a)/2;
            if f(a)*f(m)  < 0
                b = m;
                disp(sprintf('%3d %17.12f %17.12f %17.12f %17.12f', i, a, f(a), b, f(b)));
            end
            if f(a)*f(m)  > 0
                a = m;
                disp(sprintf('%3d %17.12f %17.12f %17.12f %17.12f', i, a, f(a), b, f(b)));
            end
        end
    end
    disp(sprintf('result = %3f',(a + (b-a)/2)));
    disp(sprintf('\n') );
end
diary off
```

```
function newton
% xs: intial guess
% iteration: ieration times
% fs: input functions
% gs: g(x) = f'(x)
diary result.out
fs = {'x*x*x-2*x-5' 'exp(-x)-x' 'x*sin(x)-1' 'x*x*x - 3*x*x + 3*x - 1'};
gs = {'3*x*x-2' '-exp(-x)-1' 'sin(x) + x*cos(x)' '3*x*x -6*x +3'};
xs = {1 0 1 0};
tol = 0.001;


for k = 1:4
    i = 1;
    f = inline(fs{k},'x');
    g = inline(gs{k},'x');
    x = xs{k};
    fprintf('g_%g(x) = %s\n', k, fs{k});
    fprintf('%3s %17s %17s %17s %17s\n', 'k', 'xk','f(xk)','g(xk)', 'hk');
    fprintf('%3d %17f %17f %17f %17f\n', i, x, f(x), g(x), -f(x)/g(x));
    while abs(f(x)) > tol && i<20
        i = i+1;
        x = x - f(x)/g(x);
        fprintf('%3d %17f %17f %17f %17f \n', i, x, f(x), g(x), -f(x)/g(x));
    end
    fprintf('\n');
end
diary off
```

```
function secant
% x1,x2: intinal guess
% iteration: ieration times
% fs: input functions
diary result.out
fs = {'x*x*x-2*x-5' 'exp(-x)-x' 'x*sin(x)-1' 'x*x*x - 3*x*x + 3*x - 1'};
x1s = {1 -1 1 0};
x2s = {3 0 2 3};
tol = 0.001;

for k = 1:4
    i = 2;
    f = inline(fs{k},'x');
    x1 = x1s{k};
    x2 = x2s{k};
    fprintf('g_%g(x) = %s\n', k, fs{k});
    fprintf('%3s %17s %17s %17s\n', 'k', 'xk','f(xk)','hk');
    fprintf('%3d %17f %17f %17f \n', 1, x1, f(x1));
    fprintf('\n');
    fprintf('%3d %17f %17f %17f\n',2, x2, f(x2), -f(x2)*(x2-x1)/(f(x2)-f(x1)));
    while abs(f(x2)) > tol && i<20
        i = i+1;
        s = x2 - f(x2)*(x2-x1)/(f(x2)-f(x1));
        x1 = x2;
        x2 = s;
        fprintf('%3d %17f %17f %17f \n', i, x2, f(x2), -f(x2)*(x2-x1)/(f(x2)-f(x1)));
    end
    fprintf('\n');
end
diary off
```

For termination criteria,

using $b - a > tol$ for bisection;

using $f(x) > tol$ for Newton's method;

using $f(x) > tol$ for secant.

(a)

$$f(x) : x^3 - 2x - 5 = 0$$
$$f'(x) : 3x^2 - 2$$

$bisection:$

iteration	interval	tolerance	root	convergence rate
12	[1,3]	0.001	2.0947	linear around(0.5)

$Newton'smethod:$

iteration	initial guess	tolerance	root	convergence rate
8	1	0.001	2.0946	linear around 10^{-1}

$secant:$

iteration	initial guess	tolerance	root	convergence rate
8	1,3	0.001	2.0946	linear(10^{-1})

$fzero:$
Using matlab $fzero$ with initial guess 1, we get a root 2.0946.

(b)

$$e^{-x} = x$$
$$-e^{-x} = 1$$

$bisection:$

iteration	interval	tolerance	root	convergence rate
11	[0,1]	0.001	0.5674	linear around(0.5)

$Newton'smethod:$

iteration	initial guess	tolerance	root	convergence rate
4	0	0.001	0.5671	linear(10^{-1})

$secant:$

iteration	initial guess	tolerance	root	convergence rate
5	0,1	0.001	0.5672	linear(10^{-1})

$fzero:$

Using matlab $fzero$ with initial guess 1, we get a root 0.5671.

(c)

$$x\sin(x) = 1$$

$$\sin(x) + x\cos(x) = 0$$

$bisection$:

iteration	interval	tolerance	root	convergence rate
11	[1,2]	0.001	1.1143	linear around(0.5)

$Newton'smethod$:

iteration	initial guess	tolerance	root	convergence rate
2	0.5	0.001	1.1147	linear(10^{-1})

$secant$:

iteration	initial guess	tolerance	root	convergence rate
5	1,2	0.001	1.1142	super linear (1.5)

$fzero$:

Using matlab $fzero$ with initial guess 1, we get a root 1.1142.

(d)

$$x^3 - 3x^2 + 3x - 1 = 0$$

$$3x^2 - 6x + 3 = 0$$

$bisection$:

iteration	interval	tolerance	root	convergence rate
12	[0,3]	0.001	0.9998	linear(0.5)

$Newton'smethod$:

iteration	initial guess	tolerance	root	convergence rate
7	0	0.001	0.9122	linear(0.66)

$secant$:

iteration	initial guess	tolerance	root	convergence rate
11	0,3	0.001	0.9232	linear(0.75)

fzero :
Using matlab *fzero* with initial guess 0, we get a root 1.0000.

4. (a)

$$f(x) = \sqrt[3]{1 - \frac{3}{4x}} = 0 \Rightarrow 1 - \frac{3}{4x} = 0$$

$$4x = 3 \Rightarrow x = \frac{3}{4}$$

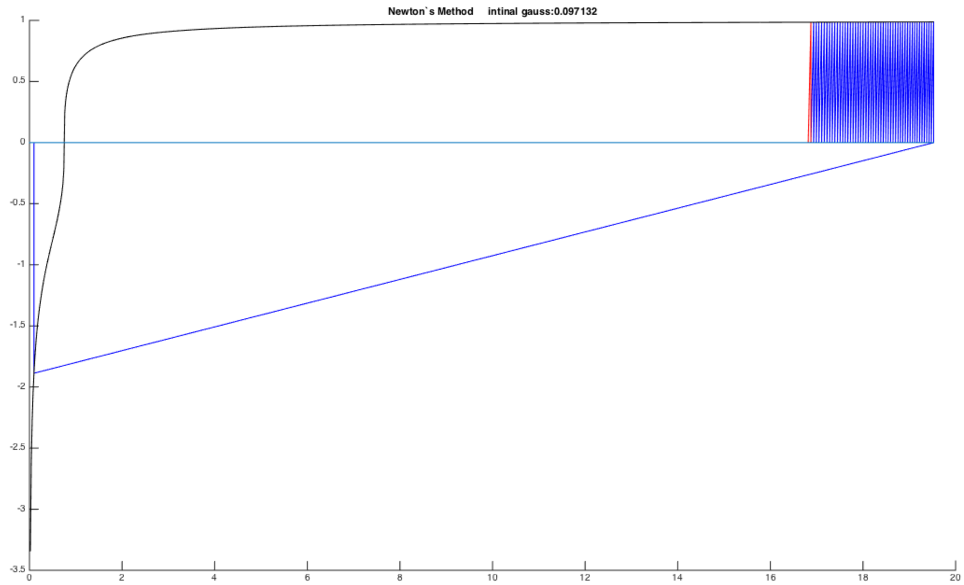
Hence, clearly function $f(x) = \sqrt[3]{1 - \frac{3}{4x}}$ only has one root which is $\frac{3}{4}$.

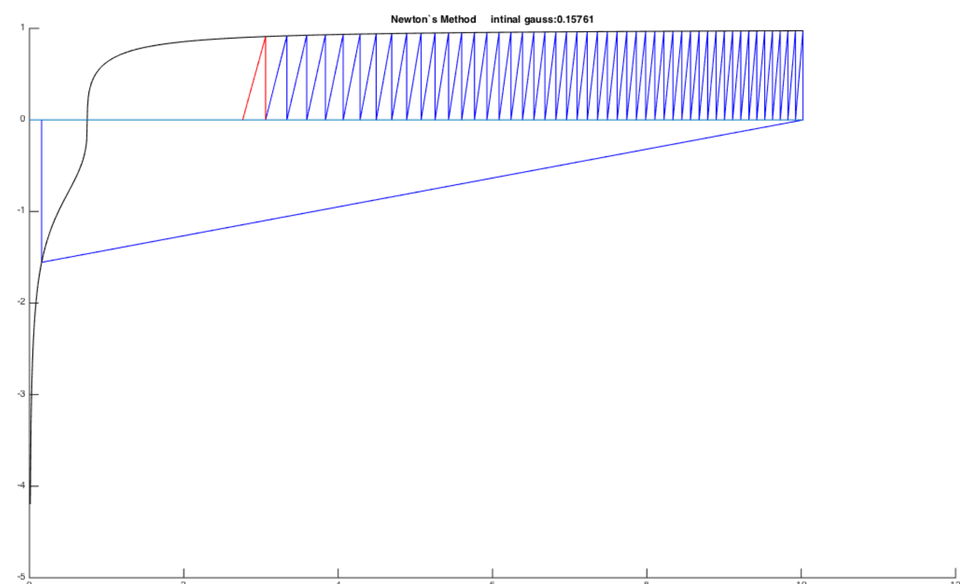
(b)

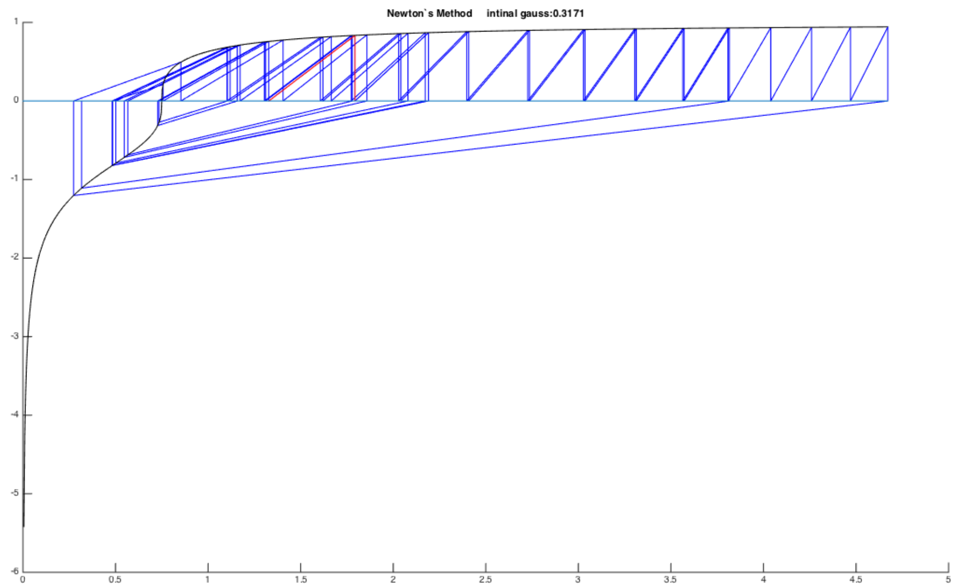
```
function root = pNewton(iteration)
% iteration: ieration times
% f: input function

x = rand();

figure
hold on;

title(['Newton`s Method     intinal gauss:',num2str(x)]);

root = x;
right = x;
left = 0;

for i = 0:iteration
    % calculate root
    % eliminate division by x
    root = x - f(x)/x;
    % update color
    color = floor(i/iteration);
    % plot root
    rangeX = [x, x];
    rangeY = [0 f(x)];
    plot(rangeX,rangeY,'Color',[color 0 1-color]);
    %plot tanget line
    rangeX = [x root];
    rangeY = [f(x) 0];
    plot(rangeX,rangeY,'Color',[color 0 1-color])
    % update right and left(total x-axis range)
    right = max(right, root);
    left = min(left, root);
    %update x
    x = root;
end
% plot f(x)
a = linspace(left,right,1001);
b = f(a);
plot(a,b,'black');
line([left right], [0 0]);

function r = f(x)
%f(x)
r = nthroot((1 - 3*power(4*x,-1)), 3);
```

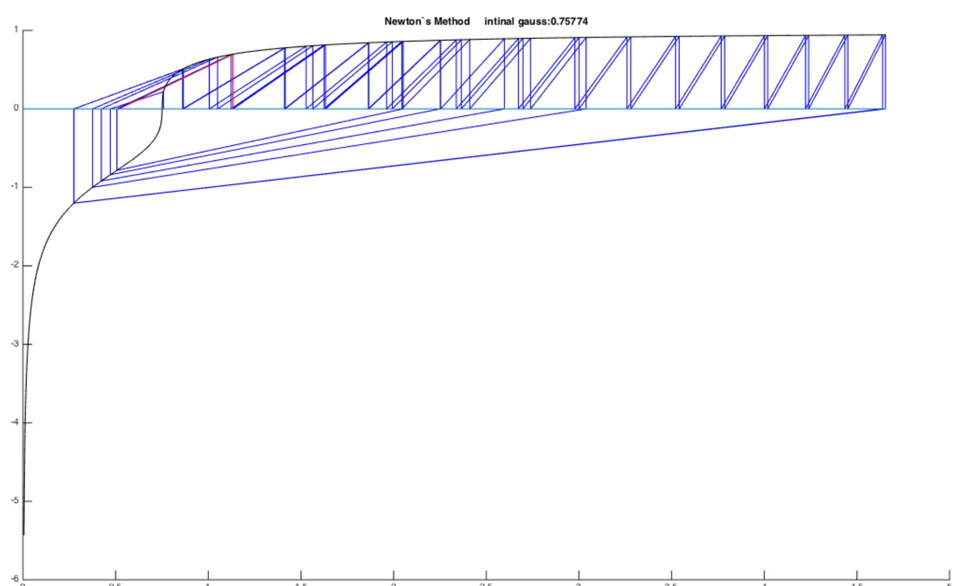
5 plots of *Newton's method* with random starting points show below, and the red line indicates

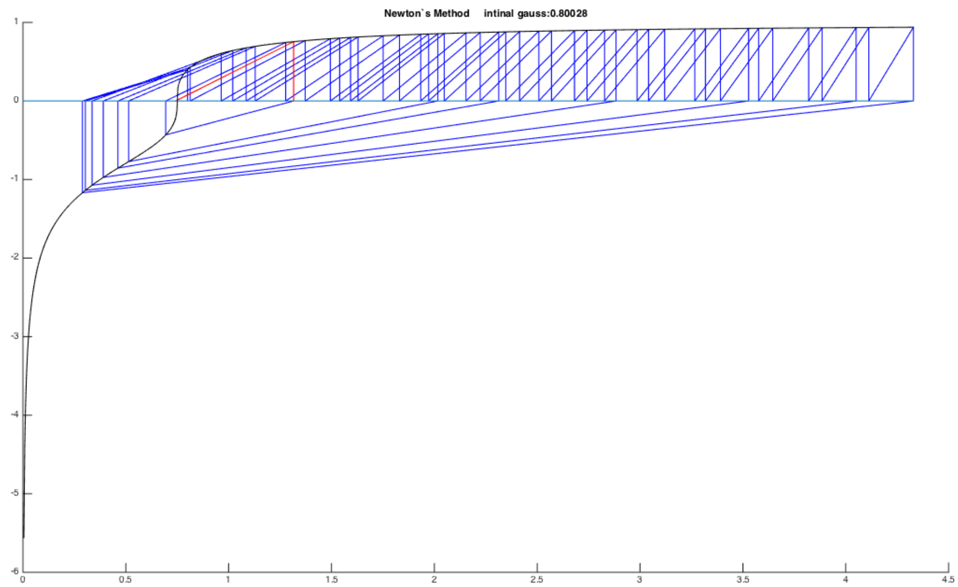
the last iteration of the total 50 iterations.






Newton`s Method intinal gauss:0.3171
1
0
-1
-2
-3
-4
-5
-6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Newton`s Method intinal gauss:0.75774
1
0
-1
-2
-3
-4
-5
-6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5



(c) From the plots, we could observe,

- Base on different initial guess with fixed iteration the *Newton's method* might converge or diverge.
- For a *Newton's method* with some initial guess, the function may converge for some certain iterations (i.e it may first converge then diverge).