homework - 4

COL726.

K. Laxman
2018CS50408

(2) given equation $f(x) = x^2 - 3x + 2 = 0 \rightarrow (1)$

given that $x = 2$, Now we have to find $|g_i'(2)| \Rightarrow g_1(x) = \frac{x^2+2}{3}$

i.e. from eq (1)

Let $3x = x^2 + 2$

$x = \frac{x^2+2}{3}$

i) $g_1'(x) = \left|\frac{2x}{3}\right|$

$g_1(2) = \frac{2 \times 2}{3} = 4/3$ so, here $g_1'(2) = 4/3 > 1$

This condition is not convergent.

ii) $g_2(x) = \sqrt{3x-2}$.

Let $x^2 - 3x + 2 = 0 \Rightarrow x^2 = 3x - 2$

$\Rightarrow \sqrt{3x-2} = x$

$g_2'(x) = \frac{1}{2\sqrt{3x-2}} \cdot 3 = \frac{3}{2\sqrt{3x-2}}$

$g_2'(2) = \frac{3}{2\sqrt{6-2}} = 3/4 < 1$ so, $g_2'(x)$ is convergent.

iii) $g_3(x) = \frac{\cancel{3x}}{3 - 2/x}$; Let $x^2 - 3x + 2 = 0$

$x^2 = 3x - 2$

$x = 3 - 2/x$

$g_3'(x) = 2/x^2$ i.e $2/(2)^2 = 1/2 < 1$

$\therefore$ $g_3'(x)$ is convergent

iv) $g_4(x) = \frac{(x^2-2)}{(2x-3)}$

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

$$= \frac{2x^2-6x+4}{(2x-3)^2}$$

$$g_4'(2) = \frac{2(2)^2-6(2)+4}{(2\times2-3)^2} \Rightarrow 0<1$$

$\therefore$ $g_4'(x)$ is convergent

q ⑦ :- By using the hint given in q5 and examining the optimisation problem

minimize $\quad x_1y_1 + \sqrt{\gamma}x_2y_2$

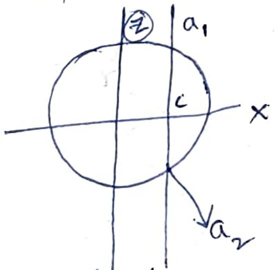
Subject to $\quad y_1^2+y_2^2 \le 1$

$\qquad y_1 \ge 1/\sqrt{1+\gamma}$ here y_1, y_2 are variables.

It is represented as

i.e. vertical line through

$y_1 = (1+\gamma)^{-1/2}$



To maximize the inner product of y with coeff vector $(x_1, \sqrt{\gamma}x_2)$. There are 3 cases depending on orientation of coeff. vector.

Case ① $x_1 > 0$ and $|x_2| \le x_1$, coeff. vector lies b/w $(1,-\sqrt{\gamma})(1,\sqrt{\gamma})$

optimum is $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{x_1^2+\gamma x_2^2}\begin{bmatrix} x_1 \\ \sqrt{\gamma}x_2 \end{bmatrix}$

optimal value is $(x_1^2+\gamma x_2^2)^{1/2}$

case ② If $x_2 \le 0$ and $x_2 < -x_2$, then $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{1+\gamma}}\begin{bmatrix} 1 \\ -\sqrt{\gamma} \end{bmatrix}$

optimal value is $\dfrac{(x_1-\gamma x_2)}{(1+\gamma)^{1/2}}$

③ If $x_2 \ge 0$ and $x_1 < x_2$ then $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{\sqrt{1+\gamma}}\begin{bmatrix} 1 \\ \sqrt{\gamma} \end{bmatrix}$

optimal value is $\dfrac{x_1+\gamma x_2}{(1+\gamma)^{1/2}}$

by considering the gradient descent algorithm.

Note that iterates satisfies $|x_2^{(k)}| < x_1^{(k)}$ so they are in the interior of the region where $f(x_1, x_2) = (x_1^2 + \gamma x_2^2)^{\frac{1}{2}}$

$$\nabla f(x) = \frac{1}{\sqrt{x_1^2 + \gamma x_2^2}} \begin{bmatrix} x_1 \\ \gamma x_2 \end{bmatrix}$$

Verify the expressions

$$x_1^{(k)} = \gamma\left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \quad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k.$$

for $k = 0$, starting point $x^{(0)} = (\gamma, 1)$ gradient at $x^{(k)} \propto (x_1^{(k)}, \gamma x_2^{(k)})$

$\therefore$ exact line search minimizes f along the line

$$\begin{bmatrix} (1-t) x_1^{(k)} \\ (1-\gamma t) x_2^{(k)} \end{bmatrix} = \left(\frac{\gamma - 1}{\gamma + 1}\right)^k \begin{bmatrix} (1-t)\gamma \\ (1-\gamma t)(-1)^k \end{bmatrix}$$

f can be written as

$$f((1-t)x_1^{(k)}, (1-\gamma t)x_2^{(k)}) = (\gamma^2(1-t)^2 + \gamma(1-\gamma t)^2)^{1/2} \left(\frac{\gamma - 1}{\gamma + 1}\right)^k$$

This is minimised by $t = 2/(1+\gamma)$ so,

$$x^{(k+1)} = \left(\frac{\gamma - 1}{\gamma + 1}\right)^k \begin{bmatrix} (1-t)\gamma \\ (1-\gamma t)(-1)^k \end{bmatrix}$$

$$= \left(\frac{\gamma - 1}{\gamma + 1}\right)^{k+1} \begin{bmatrix} \gamma \\ (-1)^{k+1} \end{bmatrix}$$

qs ①:- To show that :- x_n belongs to the Krylov subspace K_n.

To show this, we need to show that x_n can be written as linear combination of first n powers of A times b, by Induction

For $n=1$

$$X_2 = X_1 + \alpha(b - Ax_1) = (1 - \alpha A)b + \alpha x_1$$

$\because$ This is linear combn. of b & Ax_1 so, it belongs to K_1

Assume that X_n belongs to K_n for some n, then

$$X_{n+1} = X_n + \alpha(b - AX_n)$$
$$= X_n + \alpha b - \alpha A X_n$$
$$= (I - \alpha A) X_n + \alpha b$$

by induction hypothesis X_n written as

$$X_n = C_0 b + C_1 Ab + \dots + C_{n-1} A_{n-1} b$$ here $c_0, c_1, \dots c_{n-1}$ are constants

by substituting in above equation.

$$X_{n+1} = (1 - \alpha A)(C_0 b + C_1 Ab + \dots + C_{n-1} A_{n-1} b) + \alpha b$$
$$= C_0 (I - \alpha A) b + C_1 (I - \alpha A) Ab + \dots + C_{n-2} (I - \alpha A) A_{n-2} b +$$
$$(C_{n-1} - \alpha C_{n-2})(I - \alpha A) A_{n-1} b$$
$$+ (C_{n-1} \alpha)^n Ab$$
$$+ \alpha b$$ $\because$ This is linear combn. of $n+1$ powers.

To find degree $(n-1)$ polynomial corresponds to relation.

$$X_{n+1} = (I - \alpha A) X_n + \alpha b$$
$$= (I - \alpha A)(I - \alpha A) X(n-1) + \alpha (I - \alpha A) b$$
$$= (I - \alpha A)^2 X(n-1) + \alpha (I - \alpha A) b$$
$$= (I - \alpha A)^{n-1} + \alpha (I - \alpha A)^{(n-1)} b$$ which can be written as

This is polynomial (degree $n-1$) in A here I is Identity matrix.

$$P_{n-1}(x) = X_n + \alpha (XI - A)^{(-1)} b$$ we can use for value of α for last convergence

So, finally formula

$$\alpha = 2/(\lambda_{max} + \lambda_{min})$$ λ_{max} & λ_{min} are eigenvalues of A.

In this case W.K.T. eigenvalue of A is spread in circle $|z-2| \le \frac{1}{2}$ in complex plane

Let eigenvalue of A $= \lambda_1, \lambda_2, \dots \lambda_m$

$$|\lambda_i - 2| \le \frac{1}{2}$$ adding 2 on both sides

$$|\lambda_i| \le 5/2$$

$\therefore$ we have

$\lambda_{max} = \max\{|\lambda_1|, |\lambda_2|, \ldots |\lambda_m|\} \leq 5/2$

$\lambda_{min} = \min\{|\lambda_1|, |\lambda_2|, \ldots |\lambda_m|\} \geq 3/2$

substitue these value in above formula for α

$\alpha = 2/(\lambda_{max} + \lambda_{min}) \leq 4/5$ for fast convergence.

So, we could recommed $\alpha = 4/5$

https://drive.google.com/drive/folders/1jm9ZdrW-8yD8P0W9jhHf_jzQ4mzbgvh2?usp=sharing

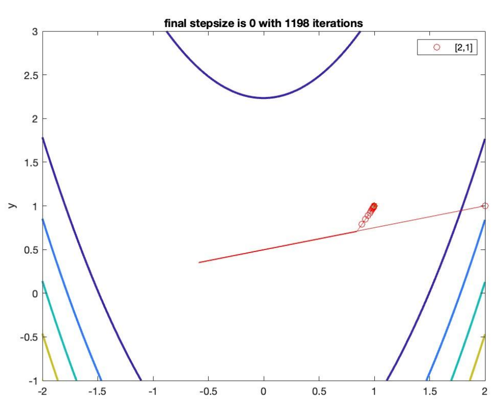
Q3

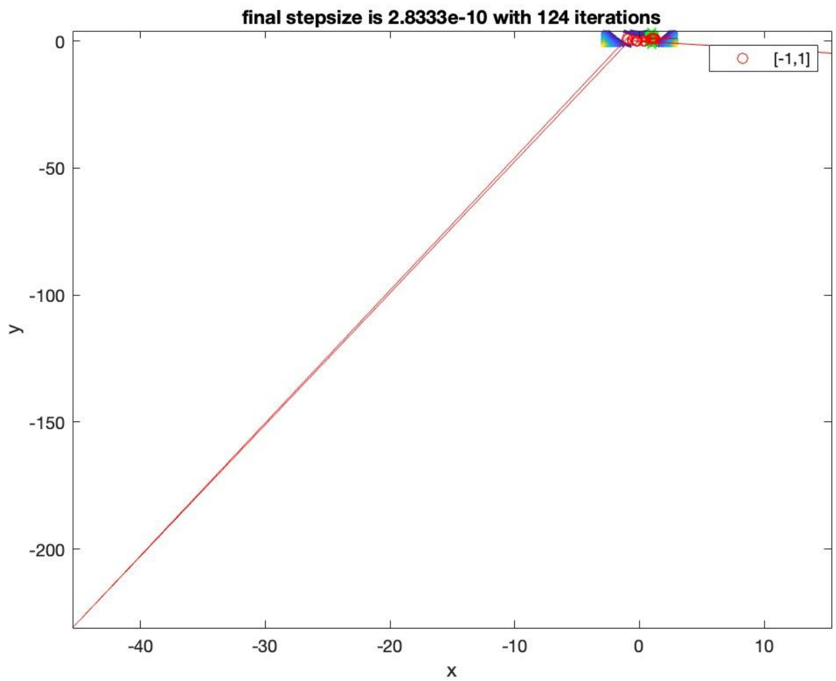
```
% Nonlinear system using Newton and Broyden's methods
x_initial = [-0.5; 1.40]; % Initial guess for x
x_true = [0; 1]; % True value of x
tolerance = eps; % Tolerance for convergence
max_iterations = 20; % Maximum number of iterations
fprintf('Newton method:\n');
fprintf('k   x(1)          x(2)          error\n');
k = 0;
x = x_initial;
step_size = ones(size(x));
error = norm(x - x_true);
fprintf('%3d %17.10e %17.10e  %17.10e\n', k, x(1), x(2), error)

while norm(step_size) > tolerance && k < max_iterations
        k = k + 1;
        step_size = -(Df(x)\f(x));
        x = x + step_size ;
        error = norm(x - true_x);
        fprintf('%3d %17 .10e %17 .10e  %17 .10e\n', k, x(1), x(2), error);
end

fprintf('\nBroyden method:\n');
fprintf('k   x(1)          x(2)          error\n');
k = 0;
x = x_initial;
fx = f(x);
B = Df(x);
step_size = ones(size(x));
error = norm(x - x_true);
fprintf('%3d %17.10e %17.10e  %17.10e\n', k, x(1), x(2), error)

while norm(step_size) > tolerance && k < max_iterations
   k = k + 1;
   step_size = -(B\fx);
   x = x + step_size;
   y = fx;
   fx = f(x);
   y = fx - y;
   B = B + ((y - B*step_size)*(step_size')) / (step_size'*step_size);
```

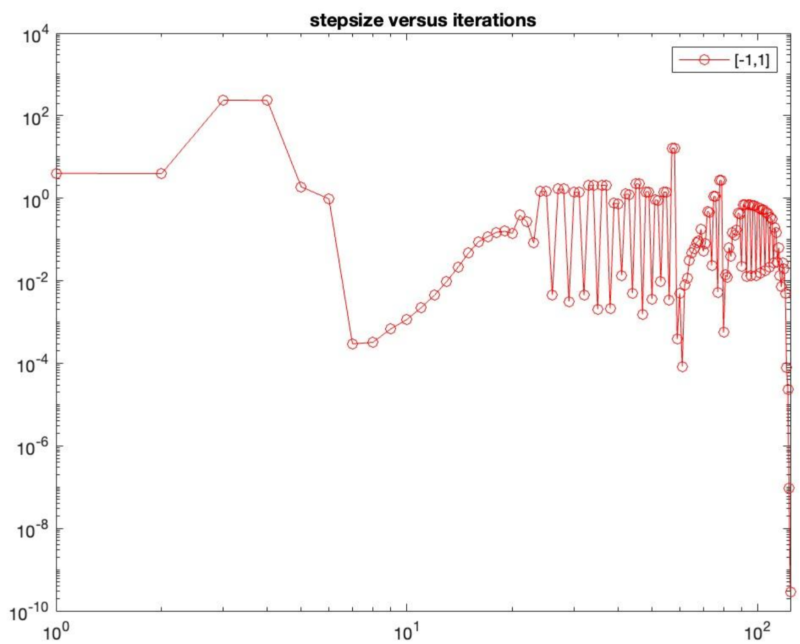
```
    error = norm(x - x_true);
    fprintf(' %3d %17.10e %17.10e  %17.10e\n', k, x(1), x(2), error);
end

% Function to calculate f(x)
function [y] = f(x)
    y = [(x(1) + 3)*(x(2)^3 - 7) + 18;
        sin(x(2)*exp(x(1)) - 1)];
end

% Function to calculate the Jacobian of f(x)
function [j] = Df(x)
    j = [x(2)^3 - 7, 3*x(2)^2*(x(1) + 3);
        x(2)*exp(x(1))*cos(x(2)*exp(x(1)) - 1), exp(x(1))*cos(x(2)*exp(x(1)) - 1)];
end
```

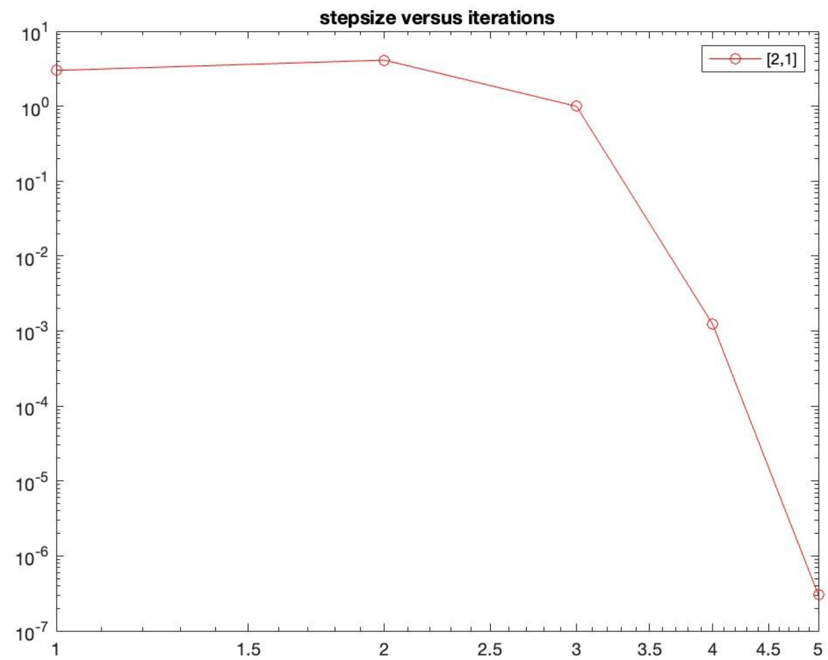
```
>> thirdqs
Newton method:
k   x(1)              x(2)              error
  0 -5.0000000000e-01  1.4000000000e+00   6.4031242374e-01
  1    2    3    4    5
Broyden method:
k   x(1)              x(2)              error
  0 -5.0000000000e-01  1.4000000000e+00   6.4031242374e-01
   1 -5.5315135718e-02  1.0280665838e+00   6.2028198166e-02
   2  5.0995307015e-04  1.0001236435e+00   5.2472835528e-04
   3 -2.3384786364e-04  1.0000765609e+00   2.4606176062e-04
   4 -4.0826221037e-05  1.0000135979e+00   4.3031185873e-05
   5 -1.3275089604e-07  1.0000000453e+00   1.4027960438e-07
   6 -5.3912058040e-10  1.0000000002e+00   5.6858615398e-10
   7  1.6678561809e-12  1.0000000000e+00   1.7584764462e-12
   8 -7.4352765976e-16  1.0000000000e+00   8.6605346667e-16
   9  3.5985405887e-17  1.0000000000e+00   1.1670861614e-16
  10  1.4149822529e-16  1.0000000000e+00   1.7985466189e-16
>>
```

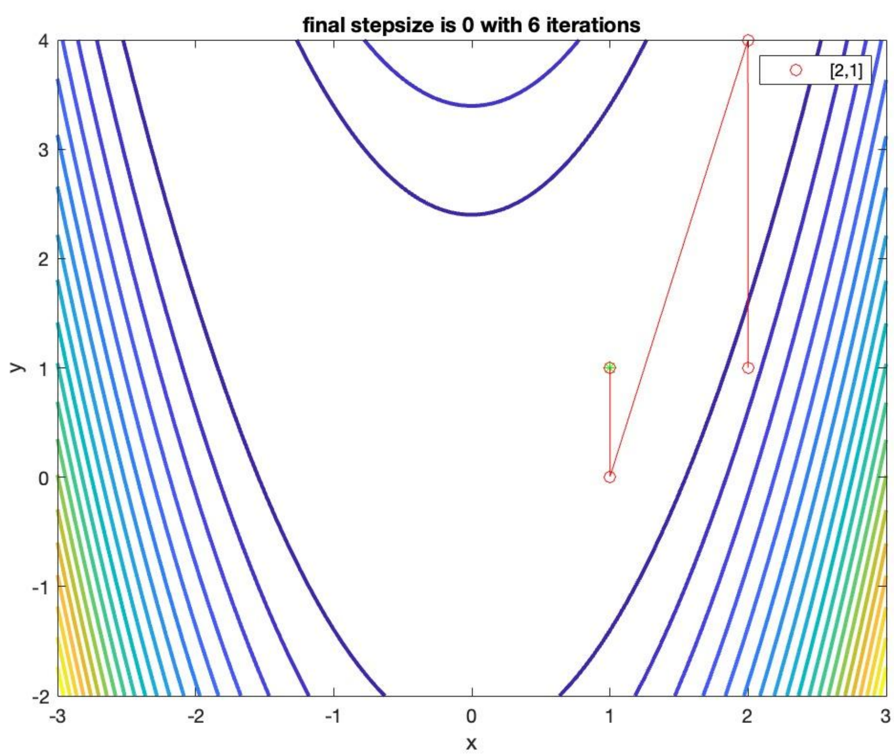
Q4



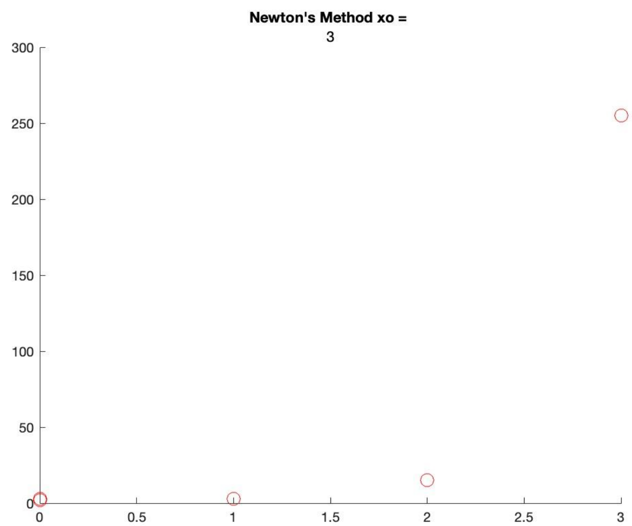

final stepsize is 2.8333e-10 with 124 iterations
[-1,1]
y
x
0
-50
-100
-150
-200
-40
-30
-20
-10
0
10

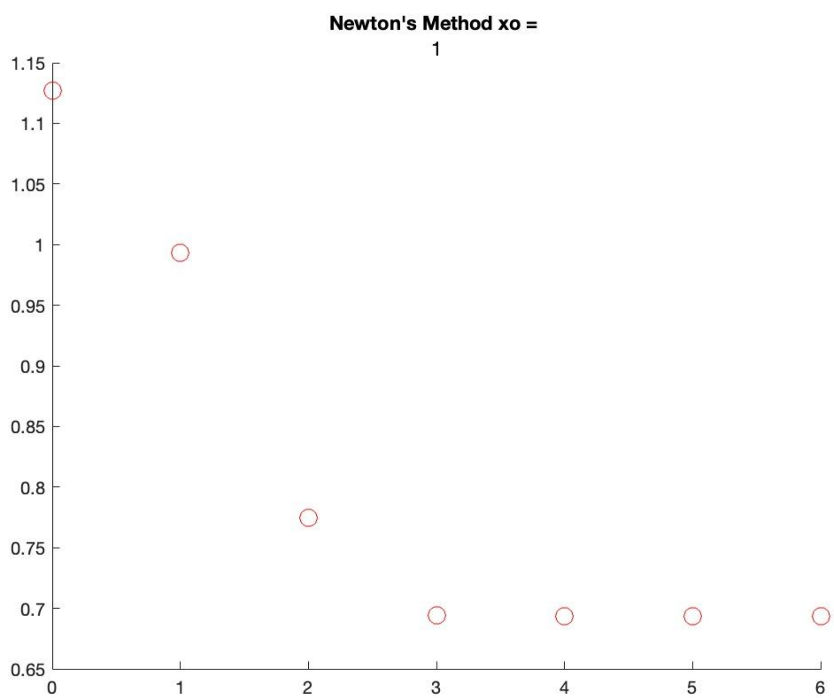

stepsize versus iterations
[-1,1]
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{0}
10^{1}
10^{2}

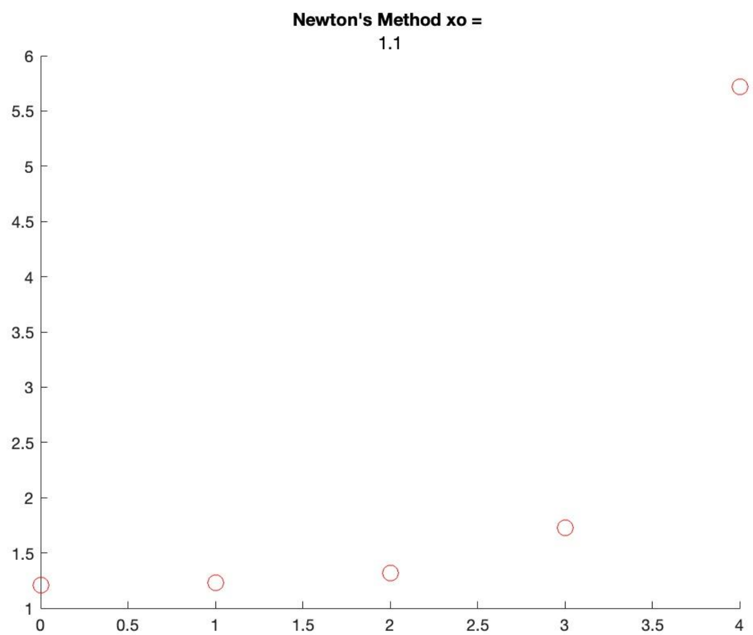

stepsize versus iterations
[2,1]
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
1
1.5
2
2.5
3
3.5
4
4.5
5


final stepsize is 0 with 6 iterations
[2,1]
4
3
2
1
0
-1
-2
y
-3
-2
-1
0
1
2
3
x

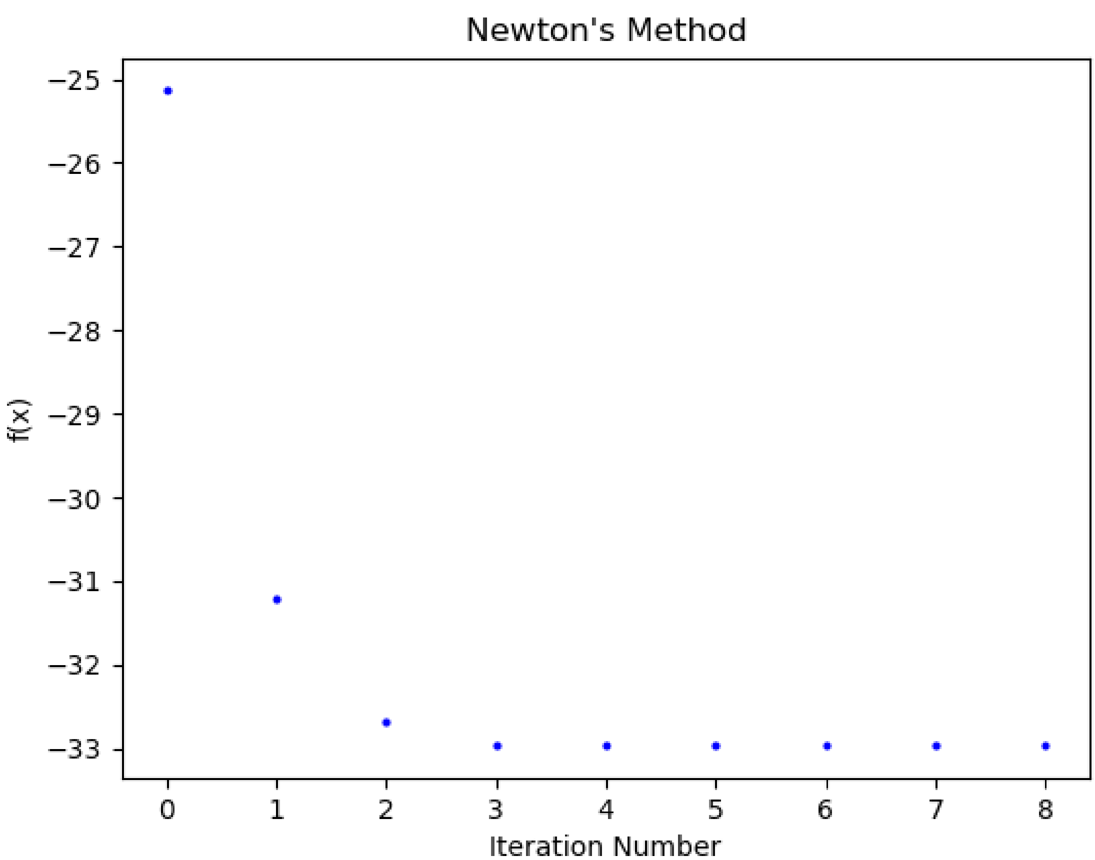
Q5


Newton's Method xo =
3
300
250
200
150
100
50
0
0
0.5
1
1.5
2
2.5
3


Newton's Method xo =
1
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
1
2
3
4
5
6


Newton's Method xo =
1.1
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
0.5
1
1.5
2
2.5
3
3.5
4

Q6


Newton's Method
−25
−26
−27
−28
−29
−30
−31
−32
−33
f(x)
0
1
2
3
4
5
6
7
8
Iteration Number