Newton-Raphson Method for Nonlinear Systems

Recall that Newton-Raphson method was predicated on employing the derivative of a function to estimate its intercept with the axis of the independent variable-that is the root. This estimate was based on the <u>first order</u> Taylor Series expansion

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

Where x_i is the initial guess at the root and x_{i+1} is the point at which the slope intercepts the x axis. At this intercept equating the zero yields

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

which is the single-equation form of the Newton's method.

We begin with the forms

$$f(x,y)=0,$$
$$g(x,y)=0.$$

The multiequation form is derived in an identical fashion. However, a multivariable Taylor series must be used to account for the fact that more than one variable contributes to the determination of the root. For the two variable cases, a <u>first order</u> Taylor Series can be written for <u>each nonlinear equation</u> as

$$f(x_{i+1},y_{i+1})=f(x_i,y_i)+(x_{i+1}-x_i)\frac{\partial f}{\partial x}+(y_{i+1}-y_i)\frac{\partial f}{\partial y}$$

and

$$g(x_{i+1},y_{i+1})=g(x_i,y_i)+(x_{i+1}-x_i)\frac{\partial g}{\partial x}+(y_{i+1}-y_i)\frac{\partial g}{\partial y}$$

Just as for the single equation version, the root estimate corresponds to the values of x and y, where $f(x_{i+1},y_{i+1})$ **and** $g(x_{i+1},y_{i+1})$ **<u>equal zero</u>.**

Equations can be rearranged to give

$$\frac{\partial f}{\partial x}x_{i+1}+\frac{\partial f}{\partial y}y_{i+1}=-f(x_i,y_i)+x_i\frac{\partial f}{\partial x}+y_i\frac{\partial f}{\partial y}$$

$$\frac{\partial g}{\partial x}x_{i+1}+\frac{\partial g}{\partial y}y_{i+1}=-g(x_i,y_i)+x_i\frac{\partial g}{\partial x}+y_i\frac{\partial g}{\partial y}$$

Thus we obtain is a set of two linear equations with two unknowns.

We convert nonlinear system solution to the linear system solution

From these equations we obtain

$$x_{i+1} = x_i - \frac{f(x_i, y_i)\left\{\frac{\partial g}{\partial y}\right\}_{x_i, y_i} - g(x_i, y_i)\left\{\frac{\partial f}{\partial y}\right\}_{x_i, y_i}}{\begin{bmatrix} \left\{\frac{\partial f}{\partial x}\right\} & \left\{\frac{\partial f}{\partial y}\right\} \\ \left\{\frac{\partial g}{\partial x}\right\} & \left\{\frac{\partial g}{\partial y}\right\} \end{bmatrix}_{x_i, y_i}}$$

and

$$y_{i+1} = y_i - \frac{g(x_i, y_i)\left\{\frac{\partial f}{\partial x}\right\}_{x_i, y_i} - f(x_i, y_i)\left\{\frac{\partial g}{\partial x}\right\}_{x_i, y_i}}{\begin{bmatrix} \left\{\frac{\partial f}{\partial x}\right\} & \left\{\frac{\partial f}{\partial y}\right\} \\ \left\{\frac{\partial g}{\partial x}\right\} & \left\{\frac{\partial g}{\partial y}\right\} \end{bmatrix}_{x_i, y_i}}$$

The denominator of each of these equations is formally referred to as the determinant of the *Jacobian* of the system.

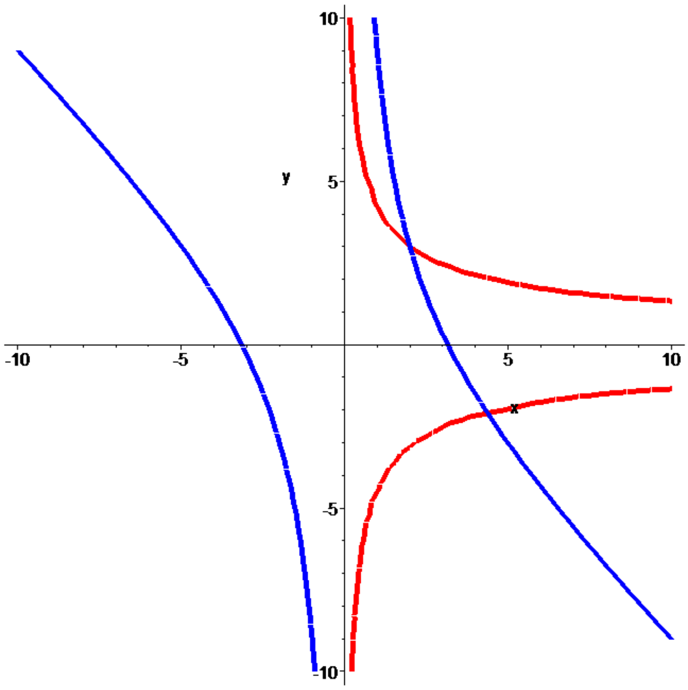
Jacobian of the system

$$\begin{bmatrix} \left\{\dfrac{\partial f}{\partial x}\right\} & \left\{\dfrac{\partial f}{\partial y}\right\} \\ \left\{\dfrac{\partial g}{\partial x}\right\} & \left\{\dfrac{\partial g}{\partial y}\right\} \end{bmatrix}$$

Example:

$$f(x,y) = x^2 + xy - 10 = 0$$
$$g(x,y) = y + 3xy^2 - 57 = 0$$

Obtain the first iterates with guesses of x=1.5 and y=3.5.



Solution:

First compute the partial derivatives and evaluate them at the initial value

$$\left\{\frac{\partial f}{\partial x}\right\}_{x_0,y_0} = \{2x+y\}_{x_0,y_0} = \{2x+y\}_{1.5,3.5} = 6.5$$

$$\left\{\frac{\partial f}{\partial y}\right\}_{x_0,y_0} = \{x\}_{x_0,y_0} = \{x\}_{1.5,3.5} = 1.5$$

$$\left\{\frac{\partial g}{\partial x}\right\}_{x_0,y_0} = \{3y^2\}_{x_0,y_0} = \{3y^2\}_{1.5,3.5} = 36.75$$

$$\left\{\frac{\partial g}{\partial y}\right\}_{x_0,y_0} = \{1+6xy\}_{x_0,y_0} = \{1+6xy\}_{1.5,3.5} = 32.5$$

Thus the determinant of the **Jacobian** for the first iteration is

$$\begin{bmatrix} \left\{\frac{\partial f}{\partial x}\right\} & \left\{\frac{\partial f}{\partial y}\right\} \\ \left\{\frac{\partial g}{\partial x}\right\} & \left\{\frac{\partial g}{\partial y}\right\} \end{bmatrix}_{(x_0,y_0)} = \begin{bmatrix} 6.5 & 1.5 \\ 36.75 & 32.5 \end{bmatrix}$$

$$= 6.5(32.5) - 1.5(36.75) = 156.25$$

The values of the functions can be evaluated at x_0, y_0 as

$$f(x_0, y_0) = -2.5$$
$$g(x_0, y_0) = 1.625$$

Then

$$x_1 = 1.5 - \frac{-2.5(32.5) - 1.625(1.5)}{156.25} = 2.03603$$

$$y_1 = 3.5 - \frac{1.625(6.5) - (-2.5)(36.75)}{156.25} = 2.84388$$

Compute the second iterations x_2 and y_2.

$$\begin{bmatrix} \left\{\dfrac{\partial f}{\partial x}\right\} \left\{\dfrac{\partial f}{\partial y}\right\} \\ \left\{\dfrac{\partial g}{\partial x}\right\} \left\{\dfrac{\partial g}{\partial y}\right\} \end{bmatrix}_{(x_1, y_1)} = ?$$

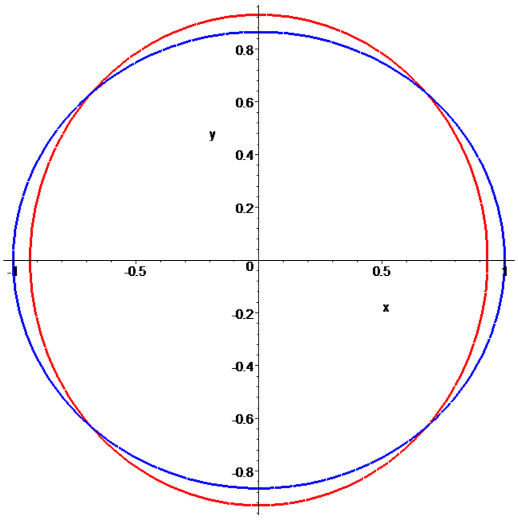
MAPLE SOLUTION!!!

```
>> [x,y]=solve('x^2+x*y-10=0','y+3*x*y^2-57=0')
x = [
2][
1/6*(4340+4*581717^(1/2))^(1/3)+106/3/(4340+4*581717^(1/2))^(1/3)-2/3]
[ -1/12*(4340+4*581717^(1/2))^(1/3)-53/3/(4340+4*581717^(1/2))^(1/3)-
2/3+1/2*i*3^(1/2)*(1/6*(4340+4*581717^(1/2))^(1/3)-
106/3/(4340+4*581717^(1/2))^(1/3))]
[ -1/12*(4340+4*581717^(1/2))^(1/3)-53/3/(4340+4*581717^(1/2))^(1/3)-2/3-
1/2*i*3^(1/2)*(1/6*(4340+4*581717^(1/2))^(1/3)-
106/3/(4340+4*581717^(1/2))^(1/3))]
y =[  3]
[6/31*(1/6*(4340+4*581717^(1/2))^(1/3)+106/3/(4340+4*581717^(1/2))^(1/3
)-2/3)^2-256/93-19/186*(4340+4*581717^(1/2))^(1/3)-
2014/93/(4340+4*581717^(1/2))^(1/3)]
[ 6/31*(-1/12*(4340+4*581717^(1/2))^(1/3)-
53/3/(4340+4*581717^(1/2))^(1/3)-
2/3+1/2*i*3^(1/2)*(1/6*(4340+4*581717^(1/2))^(1/3)-
106/3/(4340+4*581717^(1/2))^(1/3)))^2-
256/93+19/372*(4340+4*581717^(1/2))^(1/3)+1007/93/(4340+4*581717^(1/2
))^(1/3)-19/62*i*3^(1/2)*(1/6*(4340+4*581717^(1/2))^(1/3)-
106/3/(4340+4*581717^(1/2))^(1/3))]
[ 6/31*(-1/12*(4340+4*581717^(1/2))^(1/3)-
53/3/(4340+4*581717^(1/2))^(1/3)-2/3-
1/2*i*3^(1/2)*(1/6*(4340+4*581717^(1/2))^(1/3)-
106/3/(4340+4*581717^(1/2))^(1/3)))^2-
256/93+19/372*(4340+4*581717^(1/2))^(1/3)+1007/93/(4340+4*581717^(1/2
))^(1/3)+19/62*i*3^(1/2)*(1/6*(4340+4*581717^(1/2))^(1/3)-
106/3/(4340+4*581717^(1/2))^(1/3))]
>>
```

Example:

$$3x^2 + 4y^2 - 3 = 0,$$
$$x^2 + y^2 - \sqrt{3}/2 = 0.$$

The first equation represents an ellipse.
The second equation represents a circle.
Both curves are centered at the origin.



The Jacobian of this system

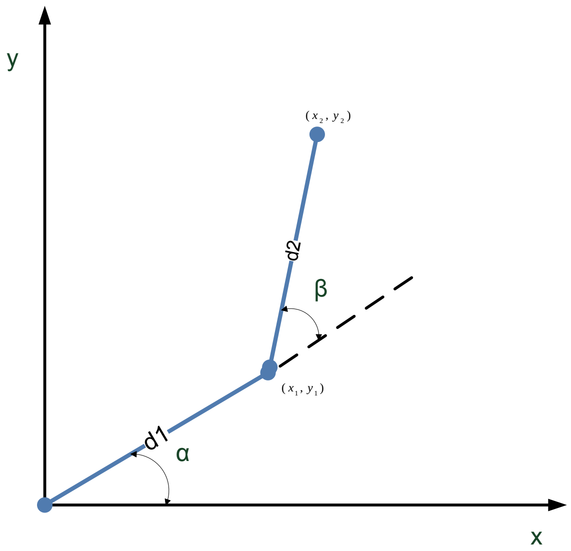
$$\begin{bmatrix} 6x & 8y \\ 2x & 2y \end{bmatrix}$$

$$X^{(0)} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

k	$x^{(k)}$	$y^{(k)}$
0	**0.5**	**0.5**
1	**0.7141**	**0.65192**
2	**0.68201**	**0.63422**
3	**0.68125**	**0.63397**
4	**0.68125**	**0.63397**

Example:

Two-link robot arm.



We need to solve for the unknown angles α and β.

$$x = d_1\cos(\alpha) + d_2\cos(\alpha+\beta)$$
$$y = d_1\sin(\alpha) + d_2\sin(\alpha+\beta)$$

Let $d_1 = 5, d_2 = 6$.

We wish to find the angles so that the arm will move to the point (10, 4).

Initial angles $\alpha^{(0)} = 0.7$ $\beta^{(0)} = 0.7$

The system of equations in this case is

$$5\cos(\alpha) + 6\cos(\alpha+\beta) - 10 = 0,$$
$$5\sin(\alpha) + 6\sin(\alpha+\beta) - 4 = 0.$$

Obtain Jacobian of the given system and check the following table.

k	$\alpha^{(k)}$	$\beta^{(k)}$	$\lVert\Delta\rVert$
0	**0.7**	**0.7**	
1	**-0.59855**	**1.8339**	**1.724**
2	**-0.10782**	**0.89987**	**1.0551**
3	**0.086882**	**0.53893**	**0.4101**
4	**0.14791**	**0.426**	**0.12837**
5	**0.155585**	**0.41139**	**0.016621**
6	**0.15598**	**0.41114**	**0.00029053**

$\Delta = x_{new} - x_{old}$.

The General Form of a System of Nonlinear Equations

Let

$$F(x_1, x_2, \ldots, x_n) = \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ f_3(x_1, x_2, \ldots, x_n) \\ . \\ . \\ . \\ f_n(x_1, x_2, \ldots, x_n) \end{pmatrix}$$

Defining the Jacobian matrix J(x) by

$$J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ . & & & \\ . & & & \\ . & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

We know from the fixed point iteration

$$x^{(k)} = G(x^{(k-1)})$$

The function G is defined*[*] by

$$G(x) = x - J(x)^{-1} F(x)$$

And the functional iteration procedure evolves from selecting $x^{(0)}$ **and generating for** k≥ 1 ,

$$x^{(k)} = G(x^{(k-1)}) = x^{(k-1)} - J(x^{(k-1)})^{-1} F(x^{(k-1)})$$

This method is called, Newton's method for nonlinear systems and is generally expected to give quadratic converge, provided that a sufficiently accurate starting value is known and inverse of the Jacobian matrix exists.

*** Burden, R. Numerical Analysis p.498**

- **First calculate**

$$F(x) \text{ and } J(x)$$

- **Then solve n x n linear system**

$$J(x)y = -F(x)$$

- **And set**

$$x = x + y$$

Newton's Method for Systems Algorithm

To approximate the solution of the nonlinear system F(x)=0 given an initial approximation x:

INPUT number n of equations and unknowns; initial approximation

$x=(x_1,x_2\ldots,x_n)^t$, Tolerance TOL, Maximum iterations.

OUTPUT approximate solution $x=(x_1,x_2\ldots,x_n)^t$ or a message that

***number of iteration was exceeded*.**

Step 1 Set k=1

Step 2 While(k≤) do Steps 3-7.

Step 3 Calculate F(x) and J(x)

$J(x)_{i,j} = (\partial f_i(x)/\partial x_j)$ **for $1 \le i, j \le n$**

Step 4 Solve n x n linear system J(x)y=-F(x)

Step 5 Set x=x + y

Step 6 If||y|| < TOL Then Output (x)

(Procedure completed successfully)

Step 7 Set k= k+1,

Step 8 OUTPUT('Maximum number of iterations exceeded');

STOP

Example:

Given the nonlinear system

$$
\begin{aligned}
x_1^2 + x_2^2 + x_3^2 - 1 &= 0,\\
x_1^2 \qquad + x_3^2 - 1/4 &= 0,\\
x_1^2 + x_2^2 - 4x_3 &= 0.
\end{aligned}
$$

$$
F(x) = \begin{Bmatrix} x_1^2 + x_2^2 + x_3^2 - 1 \\ x_1^2 + x_3^2 - 1/4 \\ x_1^2 + x_2^2 - 4x_3 \end{Bmatrix}
$$

Newton's method to obtain the first seven iterates with the initial approximation is

$$
x^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}
$$

Solution:

The Jacobian matrix J(x) for this system is given by

$$J(x)=\begin{bmatrix} 2x_1 & 2x_2 & 2x_3 \\ 2x_1 & 0 & 2x_3 \\ 2x_1 & 2x_2 & -4 \end{bmatrix}$$

and

$$\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix}=\begin{bmatrix} x_1^{(k-1)} \\ x_2^{(k-1)} \\ x_3^{(k-1)} \end{bmatrix}+\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix}$$

Where

$$\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix}=-(J(x_1^{(k-1)},x_2^{(k-1)},x_3^{(k-1)}))^{-1}F(x_1^{(k-1)},x_2^{(k-1)},x_3^{(k-1)})$$

$$J(x^{(0)}) = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & -4 \end{bmatrix}$$

Inverse of the Jacobian matrix:

$$J^{-1}(x^{(0)}) = \begin{bmatrix} -0.1667 & 0.5000 & 0.1667 \\ 0.5000 & -0.5000 & 0 \\ 0.1667 & 0 & -0.1667 \end{bmatrix}$$

$$F(x^{(0)}) = \begin{Bmatrix} 2 \\ 1.75 \\ -2 \end{Bmatrix}$$

$$J^{-1}(x^{(0)})F(x^{(0)}) = \begin{bmatrix} -0.1667 & 0.5000 & 0.1667 \\ 0.5000 & -0.5000 & 0 \\ 0.1667 & 0 & -0.1667 \end{bmatrix} \begin{bmatrix} 2 \\ 1.75 \\ -2 \end{bmatrix} = \begin{bmatrix} 0.2083 \\ 0.1250 \\ 0.6667 \end{bmatrix}$$

$$\begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \end{bmatrix} = -\begin{bmatrix} 0.20833 \\ 0.1250 \\ 0.6667 \end{bmatrix}$$

$$x = x + y$$

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} + \begin{bmatrix} y_1^{(0)} \\ y_2^{(0)} \\ y_3^{(0)} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.20833 \\ -0.1250 \\ -0.6667 \end{bmatrix} = \begin{bmatrix} 0.79167 \\ 0.875 \\ 0.3333 \end{bmatrix}$$

$$\Delta x^{(1)} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} - \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} = \begin{bmatrix} 0.79167 \\ 0.875 \\ 0.3333 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.20833 \\ -0.125 \\ -0.6667 \end{bmatrix}$$

$$\left\| \Delta x^{(1)} \right\| = \sqrt{0.043401 + 0.015625 + 0.444489} = \sqrt{0.503503} = 0.709588$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\|\Delta x\|$
0	**1.00000**	**1.00000**	**1.00000**	
1	**0.79167**	**0.875**	**0.33333**	**0.70959**
2	**0.44365**	**0.86607**	**0.42875**	**0.36111**
3	**0.28927**	**0.86603**	**0.44538**	**0.16405**
4	**0.2296**	**0.86603**	**0.44705**	**0.0507**
5	**0.22371**	**0.86603**	**0.4472**	**0.0058853**
6	**0.22361**	**0.86603**	**0.44721**	**0.00010352**
7	**0.22361**	**0.86603**	**0.44721**	**2.4665e-06**

Example:

Given the nonlinear system

$$
\begin{aligned}
&3x_1 - \cos(x_2x_3) - \frac{1}{2} = 0,\\
&x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 = 0,\\
&e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0
\end{aligned}
$$

$$
F(x) = \begin{Bmatrix} 3x_1 - \cos(x_2x_3) - 0.5 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 \\ e^{-x_1x_2} + 20x_3 + \dfrac{10\pi - 3}{3} \end{Bmatrix}
$$

Use Newton's method to obtain the first five iterates with the initial approximation is

$$
x^{(0)} = \begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \end{bmatrix}
$$

Solution:

The Jacobian matrix J(x) for this system is given by

$$J(x_1,x_2,x_3)=\begin{bmatrix} 3 & x_3\sin(x_2x_3) & x_2\sin(x_2x_3) \\ 2x_1 & 162(x_2+0.1) & \cos(x_3) \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix}$$

and

$$\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix}=\begin{bmatrix} x_1^{(k-1)} \\ x_2^{(k-1)} \\ x_3^{(k-1)} \end{bmatrix}+\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix}$$

Where

$$\begin{bmatrix} y_1^{(k-1)} \\ y_2^{(k-1)} \\ y_3^{(k-1)} \end{bmatrix}=-(J(x_1^{(k-1)},x_2^{(k-1)},x_3^{(k-1)}))^{-1}F(x_1^{(k-1)},x_2^{(k-1)},x_3^{(k-1)})$$

$$F(x_1^{k-1},x_2^{k-1},x_3^{k-1})=\begin{bmatrix} 3x_1^{(k-1)}-\cos(x_2^{(k-1)}x_3^{(k-1)})-0.5 \\ (x_1^2)^{(k-1)}-81(x_2^{(k-1)}+0.1)^2+\sin(x_3^{(k-1)})+1.06 \\ e^{x_1^{(k-1)}x_2^{(k-1)}}+20x_3^{(k-1)}+\dfrac{10\pi-3}{3} \end{bmatrix}$$

The results obtained using these iterative procedures are shown in the following table.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\lVert \Delta x \rVert$
0	**0.10000000**	**0.10000000**	**-0.10000000**	
1	**0.50003702**	**0.01946686**	**-0.52152047**	
2	**0.50004593**	**0.00158859**	**-0.52355711**	
3	**0.50000034**	**0.00001244**	**-0.52359845**	
4	**0.50000000**	**0.00000000**	**-0.52359877**	
5	**0.50000000**	**0.00000000**	**-0.52359877**	**0.00000**

$$\left\| x^{(5)} - x^{(4)} \right\| = 0$$

(Compare results with Fixed Point solution solved before)

MATLAB M-File (Newton’s for nonlinear Systems)

```
function X=Newtonsys(F,JF, x0 ,tol, maxit)
%           FAUSETT 5.1.2
%          Solve the nonlinear system F(x)=0 using Newton's
Method
%          vectors x and x0 are rowvectors
%           function F returns a column vector
% stop      if norm of change in solutişon vector is less than tol
value
% solve JF(x) y=-F(x) using Matlab's "backslash operator"
%     y=-feval(JF, x.old) \ feval(F, x.old) ;
% the next approximate solution is x.new=x.old+y';P is the inital
approximation to the solution
x.old=x0;
disp([0 x.old]);
iter=1;
while (iter <=maxit)
       y=-feval(JF, x.old) \ feval(F, x.old) ;
       x.new=x.old+y';
        diff=norm(x.new-x.old);
        disp('Newton method has converged)')
        return;
    else
        x.old=x.new;
    end
    iter=iter+1;
 end
 disp('Newron method did not Converge')
 x=x.new;
```

ANOTHER FILE

```
function [P,iter,err]=newdim(F,JF,P,delta,epsilon,maxit)
%Input    -F is the system saved as the M-file F.m
%          -JF is the Jacobian of F saved as the M-file JF.M
%          -P is the inital approximation to the solution
%          -delta is the tolerance for P
%          -epsilon is the tolerance for F(P)
%          -maxit is the maximum number of iterations
%Output -P is the approximation to the solution
%          -iter is the number of iterations required
%          -err is the error estimate for P
%Use the @ notation call
%[P,iter,err]=newdim(@F, @JF, P, delta, epsilon, maxit).
Y=F(P);
for k=1:maxit
  J=JF(P);
  Q=P-(J\Y')';
  Z=F(Q);
  err=norm(Q-P);
  relerr=err/(norm(Q)+eps);
  P=Q;
  Y=Z;
  iter=k;
  if (err<delta)|(relerr<delta)|(abs(Y)<epsilon)
    break
  end
end
```