**Numerical Methods for Science and Engineering**

**Lecture Note 4**

**Nonlinear System of Equations**

**4.1 Newton-Raphson Method**

Recall that the Newton-Raphson formula to find the root of the equation can be written as

For a system of nonlinear equations, the Newton-Raphson formula in matrix form can be expressed as

where *X* is a variables matrix, *F*(*X*) is a function matrix and *J(X)* is a Jacobian matrix.

The variable, function and Jacobian matrices for a system of two variables can be written as

**This method can be extended for equations with more variables**.

**Example 4.1** Consider the nonlinear system .

1. Find the Jacobian matrix for the above system.
2. Evaluate the inverse of the Jacobian matrix at .
3. Write down the iterative formula for the above system based on the Newton-Raphson method.
4. Estimate the root to 2 d.p. using the using the above iterative formula once starting with .
5. Write down MATLAB commands to execute the iteration four times.
6. Use MATLAB function “**fsolve(fun,x0)**” to find the above root.

**Solution**:

(a) The function matrix and Jacobian matrix are

(b)

**Note:** The inverse of a matrix can be calculated by

At the point . we have

and

(c) The iterative formula is

where

(d) Using , we have

where

and

Thus

root to 2 d.p. is

Further iterations can be done by updating .

**(e)** Define the functions as

we have the Jacobian

**>> clear**

**>> F=@(x,y) [x.^2+y-1; y.^3-2\*x+4];**

**>> J=@(x,y)[2\*x, 1; -2, 3\*y.^2];**

**>> x(1)=1.4;**

**>> y(1)=-1;**

**>> for n=1:50**

**Xn=[x(n); y(n)];**

**Fn=F(x(n), y(n));**

**Jn=J(x(n), y(n));**

**Xn1=Xn-Jn\Fn;**

**x(n+1)=Xn1(1);**

**y(n+1)=Xn1(2);**

**end**

>> Roots=[x',y']

Roots =

1.4000 -1.0000

1.4308 -1.0462

1.4299 -1.0447

1.4299 -1.0447

1.4299 -1.0447

(f) >> clear

% Left\_hand side of equations as vector

>> F=@(x) [x(1)^2+x(2)-1; x(2)^3-2\*x(1)+4];

% Guess solution

>> x0=[1; -1];

% Solve the system

>> x=fsolve(F, x0)

x =

1.4299

-1.0447

**4.2 Fixed Point Iteration Method for Nonlinear System**

Consider the nonlinear system of two variables of the form

By rearrangements the system can be expressed as

If there is a point such that

then the point is a **fixed point** of the system and it is a root of the system.

With the arrangement (2), we may assume an iterative formula

where

Without further details it may be mentioned that if is close to the fixed point and if

and

then the iterative formula (3) will converge to the fixed point .

If the conditions (4) are not satisfied, the iterative process might diverge.

The above method can be extended for more than two variables.

**Seidel Iteration**

An improvement of the iteration process is the Seidel iteration similar to Gauss-Seidel iteration for linear system. In Seidel iteration process the iterative formulas are

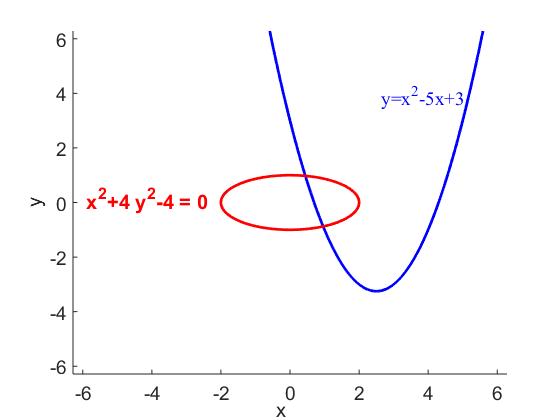
**Example 4.2**

Consider the system of equations

1. Plot the graphs of the above system using MATLAB function “**ezplot(fun)**”.

From your plot find the number of real roots and estimate the roots.

1. A fixed point iteration formula is suggested to estimate root at
2. Verify whether the above iterative formula will converge to the root near . If converges, perform one iteration otherwise suggest another fixed point iterative formula which converge to root.
3. Verify whether the above iterative formula will converge to the root near . If converges, perform one iteration otherwise suggest another fixed point iterative formula which converge to the root.
4. Write MATLAB commands to execute the iterative formula in b(ii) five times.

(a) MATAB codes for ezplot

>> clear

>> figure

>> hold on

>> ezplot('y-x.^2+5\*x-3')

>> ezplot('x.^2+4\*y.^2-4')

>> hold off

From graph it can be seen that the system has two real roots near and .

(b) (i) From the iterative formula let us define

and

Here

and

Near , we have

and

The partial derivative tests are satisfied and hence the iterations converges to the root near .

We can use iterative formulas for the root near as

In Seidel iteration the second iterative equation will be of the form

Calculations with the above iterative formulas with are given below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Jacobi iteration | | |  | Seidel iteration | | |
| *n* | *xn* | *yn* |  | *n* | *xn* | *yn* |
| 0 | 0.4 | 1.0 |  | 0 | 0.4 | 1.0 |
| 1 | 0.432 | 0.984 |  | 1 | 0.432 | 0.9813 |

(ii) Near (0.9, -1.0) we have

satisfy the partial derivative test.

But

Test fails and the convergence is not guaranteed.

Rearranging the second equation by

we have

Corresponding iterative function can be of the form

and

At ,

satisfies the partial derivative test.

Thus a fixed point iterative formula which converge to root near is

(e) >> clear

>> x(1)=0.9;

>> y(1)=-1;

>> for n=1:5

x(n+1)=(x(n)^2-y(n)+3)/5;

y(n+1)=(x(n)^2+4\*y(n)^2-4+10\*y(n))/10;

end

>> Iterative\_Roots =[x',y']

Iterative\_Roots =

0.9000 -1.0000

0.9620 -0.9190

0.9689 -0.8886

0.9655 -0.8789

0.9622 -0.8767

0.9605 -0.8767

**Exercise 4**

1. Consider the following nonlinear systems:

(a) starting with

(b) starting with .

(c) starting with .

(d) starting with .

(e) starting with .

(f) starting with .

Now using Newton-Raphson method answer the following questions. For each of the above nonlinear system of equations:

(i) Find the Jacobian matrix for the above system.

(ii)Evaluate the inverse of the Jacobian matrix at the given point.

(iii) Write down the iterative formula for Newton-Raphson method.

(iv) Estimate the root to 2 d.p. using the using the above iterative formula **once** starting with the given point.

(v)Write down MATLAB commands to execute the iteration four times.

(vi) Use MATLAB function “**fsolve (fun,x0)**” to find the root.

.

2. Sketch the curves represented by the following non-linear equations,

1. Solve using Newton’s method with initial estimate
2. Repeat (a) but with an initial estimate
3. Use graphical approach to validate the results in (a) and (b).

3. Sketch the curves represented by the following non-linear equations,

1. Solve using Newton’s method with initial estimate
2. Repeat (a) but with an initial estimate
3. Use graphical approach to validate the results in (a) and (b).

|  |
| --- |
| 4. The system of non-linear equations  has a solution near .  The following iterative formula is suggested to estimate the roots. |
| i. Verify whether the above iterative formula will converge to the root near . If converges, perform one iteration otherwise write ‘Not Convergent’. |
| ii. Write MATLAB commands to execute the iterative formula five times. |

.

5. Determine the roots of the following simultaneous non linear equations using (a) Newton-Raphson method, (b) fixed-point iteration method.

,

Employ initial guesses of and discuss the results.

6. The following system has a root near .

Estimate the root correct to 3 decimal places using fixed point iterative method.

7. A planar, two-link robot arm is shown in the figure. The coordinate system is the tool frame and attached to the end-effector. The coordinates of the end-effector relative to the base frame are expressed as

Suppose the lengths, in consistent physical units, of the two links are and and that . Find the joint angles and (in radians) using Newton’s method with initial estimate of (0.8,0.9)

