VINIT KUMAR SINGH: 16PH20036

Assignment – 3

Newton-Raphson for finding reciprocal of a number: The reciprocal of a real
number *a* is defined as a zero of the function: f(x)=1/x-a.

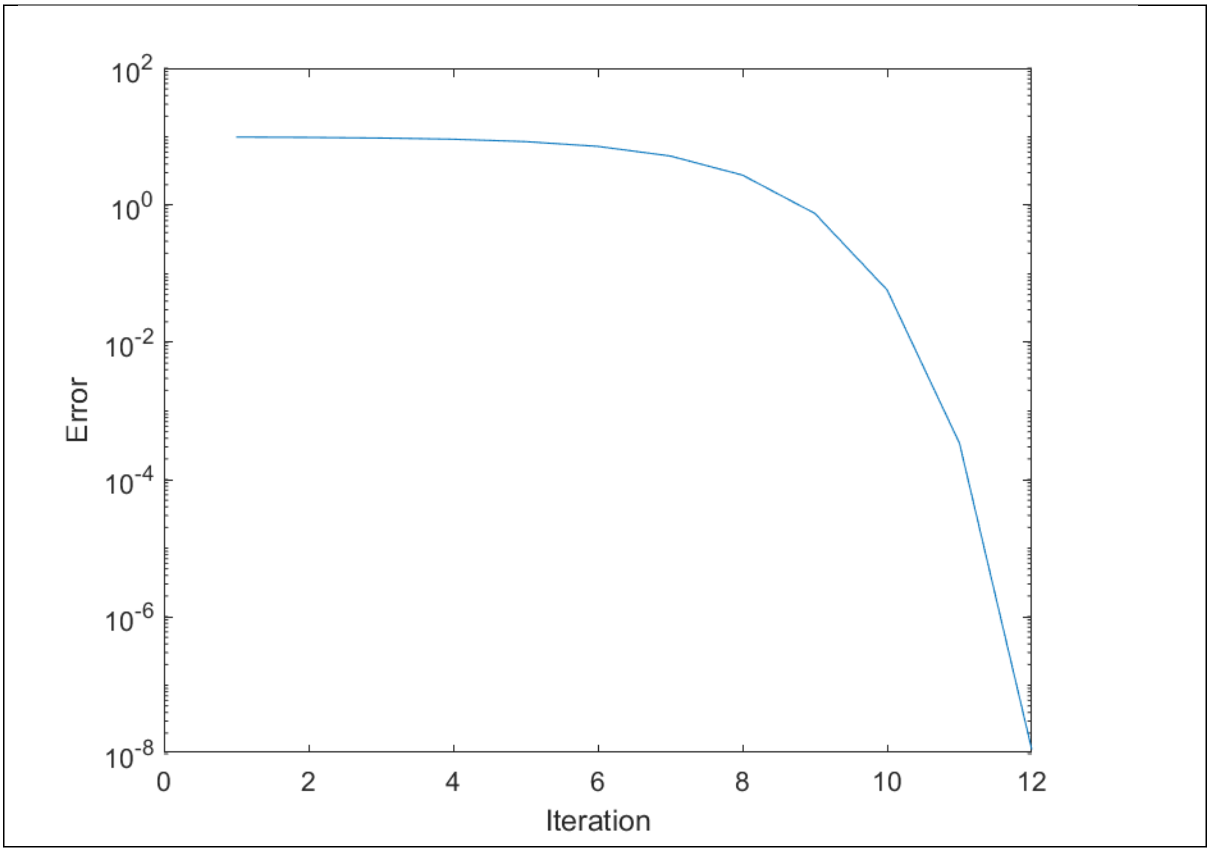
The function converges for an initial estimate in the range $0 < x_0 < 2/a$.
a) Write a Matlab code that will be able to find the reciprocal of any real number
using Newton-Raphson method. Do not set an error limit. Rather let the code run for a fixed number of 50 iterations
b) Plot the error propagation (by comparing the outcome of the code and 1/*a*) and plot is as a function of the iteration.

Solution:

```
a=1/10;
n=50;
X=zeros(1,n);
X(1)=0.1;

f=@(x,a) 1./x-a;
df=@(x) -1./x.^2;

for i=2:n
    X(i)=X(i-1)-f(X(i-1),a)/df(X(i-1));
end
semilogy(1:n,abs(X-1/a))
ylabel('Error')
xlabel('Iteration')
```



Newton-Raphson for simultaneous non-homogeneous equations: Consider the set of algebraic equations,

x+y+x=3

x^2+y^2+z^2=5

e^x+xy-xz=1

a) Use Newton-Raphson method to solve this system of equations with a starting guess of (x,y,z) = (1, 0, 0). See if the values converge to (1.2244, - 0.0931, 1.8687). If we use (1, 0,1) as the initial guess do we get the same root?

b) Check if (x,y,z)= (0,0,0) fails as a initial guess. Why?

c) Try plotting maximum absolute error of each iteration as a function of number of iteration in semi log scale to check error propagation.

```
clc
clear
N=10;

F1=@(x) x(1)+x(2)+x(3)-3;
F2=@(x) x(1)^2+x(2)^2+x(3)^2-5;
F3=@(x) exp(x(1))+x(1)*x(2)-x(1)*x(3)-1;
F=@(x) [F1(x);F2(x);F3(x)];

J=@(x) [1,1,1; 2*x(1), 2*x(2), 2*x(3); exp(x(1))+x(2)-
x(3),x(1),-x(1)];
```

```
x0=[1;0;1];
err=zeros(1,N);

for i=1:N
    if det(J(x0))==0
        fprintf('Determinant of Jacobian is 0')
        break

    end
    x=x0-(J(x0)\F(x0));
    err(i)=norm(abs(x-x0));
    x0=x;
end
disp(x)
semilogy(1:N,err)
ylabel('Error')
xlabel('Iteration')
```

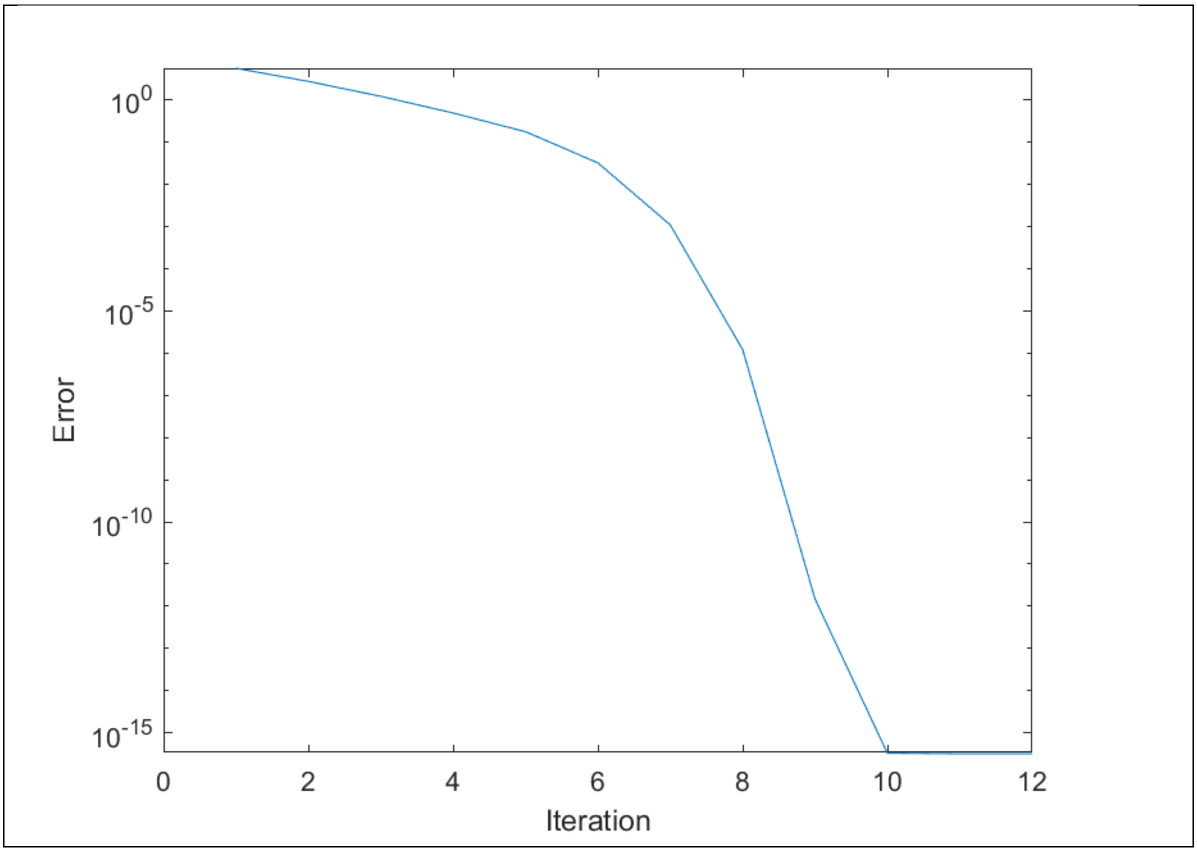
Execution:

Initial Guess: (1,0,0)
Result: (1.2244, -0.0931, 1.8687)

Initial Guess: (1,0,1)
Result: (1.2244, -0.0931, 1.8687)
Yes we get the same root.

Initial Guess: (2,3,2)
Result: (0,2,1)

Initial Guess: (0,0,0)
Result: Determinant of Jacobian is 0.



Diagonal dominance of matrix: Consider the square matrices:

A =[-6 2 1 2 1; 3 8 -4 1 0; -1 1 4 10 1; 3 -4 1 9 2; 2 0 1 3 10]

B= [18 3 6 -3; 9 13 -5 2; -3 -2 4 9; 6 0 11 3]

Write a code to see is the matrices A and B are diagonally dominant. In case if they are not, make the code display a message like "Not strictly diagonally dominant on row (row number)"

```
function [] = Di_mat(A)
disp(A);
for i=1:size(A,1)
    if abs(2*A(i,i))<sum(abs(A(i,:)))
        fprintf('Not Diagonally dominant at row
%d\n',i);
    end
end
end
```

Execution:

```
>> Di_mat([-6 2 1 2 1;3 8 -4 1 0;-1 1 4 10 1;3 -4 1 9 2;2 0 1 3 10])
    -6     2     1     2     1
     3     8    -4     1     0
    -1     1     4    10     1
     3    -4     1     9     2
     2     0     1     3    10

Not Diagonally dominant at row 3
Not Diagonally dominant at row 4

>> Di_mat([18 3 6 -3;9 13 -5 2;-3 -2 4 9;6 0 11 3])
    18     3     6    -3
     9    13    -5     2
    -3    -2     4     9
     6     0    11     3

Not Diagonally dominant at row 2
Not Diagonally dominant at row 3
Not Diagonally dominant at row 4
```