Bangladesh University of Engineering and Technology

Numerical Technique Laboratory

EEE 212

Experiment No.: 06

Name of the Experiment: Numerical Differentiation

Department: EEE

Section: C1

Group: 01

Student No.: 1406131

1406132

Date of Performance: November 5, 2016

Date of Submission: November 11, 2016

- **Problem:** Find the nth order derivative of a given function using Richardson's Extrapolation.

General formula for nth order derivative is:

$$f^n(x) = \frac{1}{(2h)^n} \sum_{k=0}^{n} \binom{n}{k} (-1)^k f(x + (n-2k)h)$$

General formula for Richardson's Extrapolation is:

$$d_{k-1}(h) + \frac{d_{k-1(h)} - d_{k-1}(2h)}{(2^{2n})^k - 1}$$

Using Matlab we will find the 1 to 5th order derivative and show the error in each step.

nCr function:

In, the above formula we need to calculate $\binom{n}{k}$ which is nothing but nCr where n=n and r=k. So, we write a function named nCr to calculate its value.

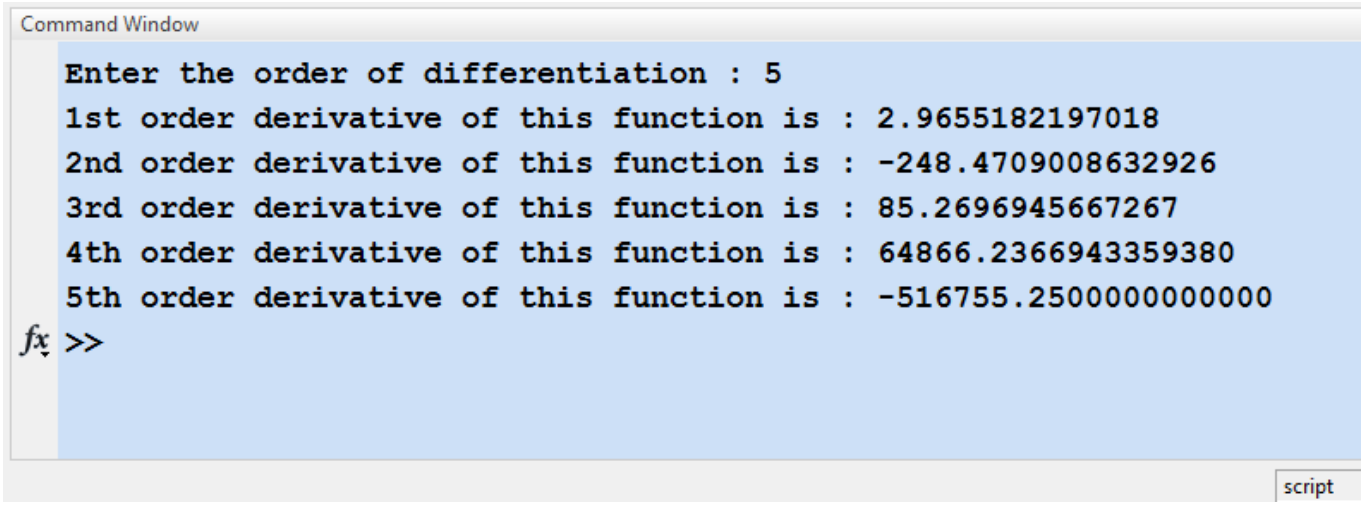
```
function ans=nCr(n,r)
    ans = factorial(n)/(factorial(r)*factorial(n-r)) ;
end
```

Code:

```
clc , clear all ;
clf , close all ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Richardson's Extrapolation for nth order derivative
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

order = input('Enter the order of differentiation : ') ;

syms y ;
f = @(y) sin(y^3-7*y^2+6*y+8) ;
x = (1-sqrt(5)) / 2 ;
n = 1 ;

while order > 0 ;
    err = 1 ;
    relerr = 1 ;
    h = 1 ;
    j = 1 ;
    toler = 10e-8 ;
    delta = 10e-10 ;
    error = [] ;
    iteration = [] ;
    sum = 0 ;
    p = 1 ;

    for k = 0 : n
       p = 1 ;
       p = p * nCr(n,k) * (-1)^k * f(x+(n-2*k)*h) ;
       sum = sum + p ;
    end

    sum = sum / (2*h)^n ;
    d(1,1) = sum ;

    while relerr>toler && err>delta && j<12
        h = h /2 ;
        sum = 0 ;
        for k = 0 : n
            p = 1 ;
            p = p*nCr(n,k)*(-1)^k*f(x+(n-2*k)*h) ;
            sum = sum + p ;
        end
        sum = sum / (2*h)^n ;
        d(j+1,1) = sum ;
        for k = 1 : j
            d(j+1,k+1) = d(j+1,k) + (d(j+1,k) - d(j,k)) /(2^(2*n) - 1) ;
        end
        err = abs(d(j+1,j+1)-d(j,j)) ;
        relerr = ( 2*err) / (abs(d(j+1,j+1))+abs(d(j,j))+eps ) ;
        original_value = eval(subs(diff(f,y,n),x)) ;
        error = [error abs(d(j+1,j+1)-original_value)] ;
        iteration = [iteration j] ;
        j = j + 1 ;
    end
```

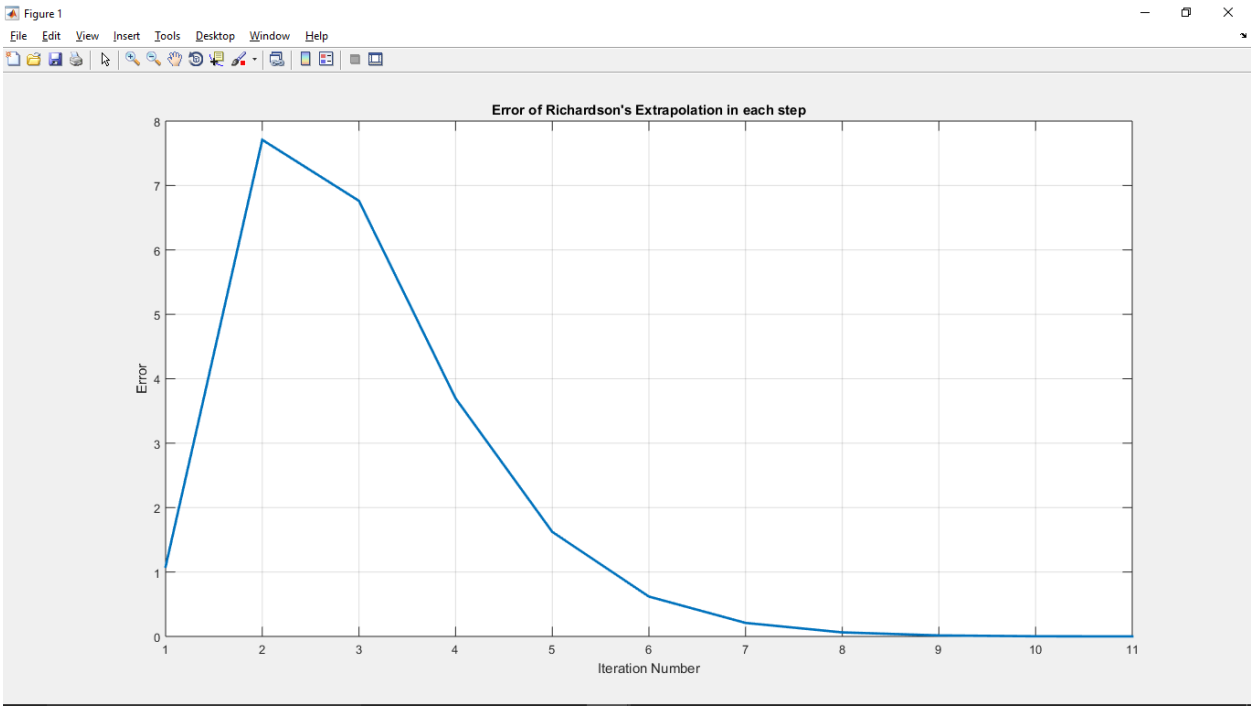
```
    plot(iteration,error,'LineWidth',2) ;
    grid on ;
    title('Error of Richardson''s Extrapolation in each step'...
     ,'LineWidth',2);
    xlabel('Iteration Number','LineWidth',2);
    ylabel('Error','LineWidth',2) ;
    figure ;
    l = length(d) ;
    d = d(l) ;

    if n == 1
        fprintf('1st order derivative of this function is : %.13f\n',d);
        elseif n==2
            fprintf('2nd order derivative of this function is : %.13f\n',d);
            elseif n==3
                fprintf('3rd order derivative of this function is :
%.13f\n',d);
                elseif n > 3
                    fprintf('%dth order derivative of this function is :
%.13f\n',n,d);
    end

    n = n + 1 ;
    order = order - 1 ;
end
```

Command Window Output:

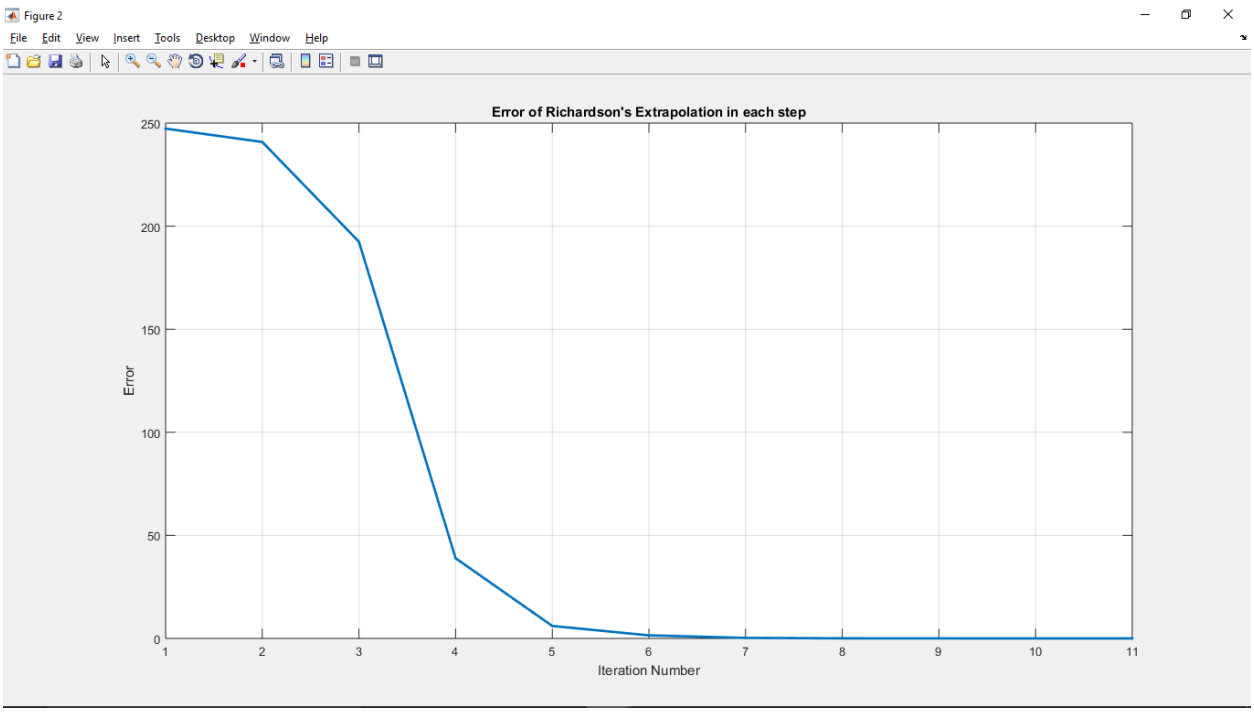


Error Plot:

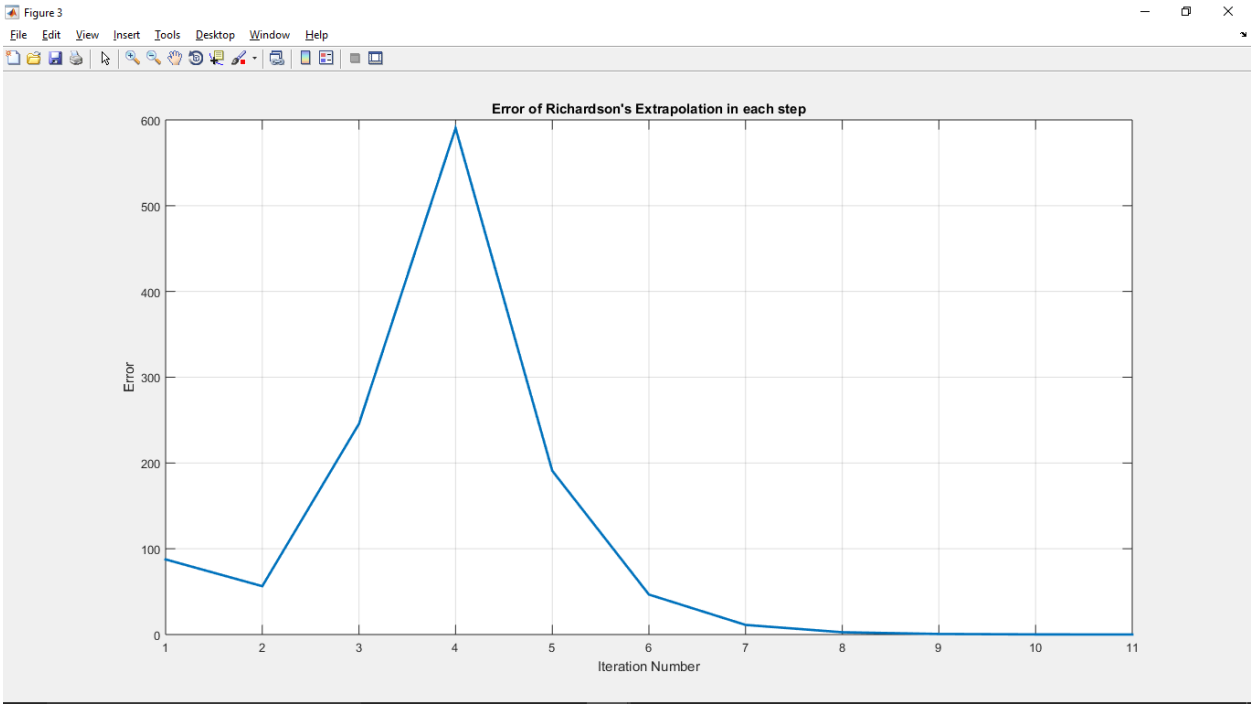
Error for 1st order differentiation:



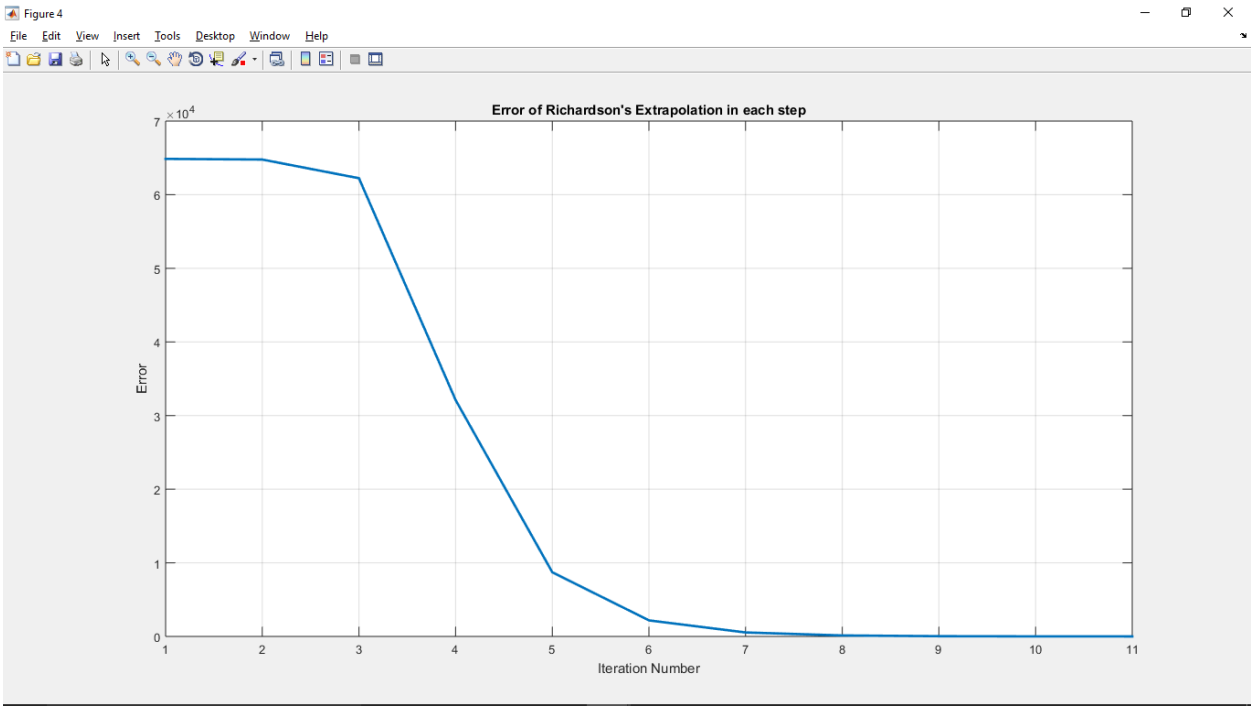
Error for 2nd order differentiation:



Error for 3rd order differentiation:



Error for 4th order differentiation:



Error for 5^{th} order differentiation: