# MA202

# ASSIGNMENT 3

**NAME:**

ARCHIT AGRAWAL

**ROLL NO. :**

202051213

**SECTION:**

2

# Lab Assignment 3

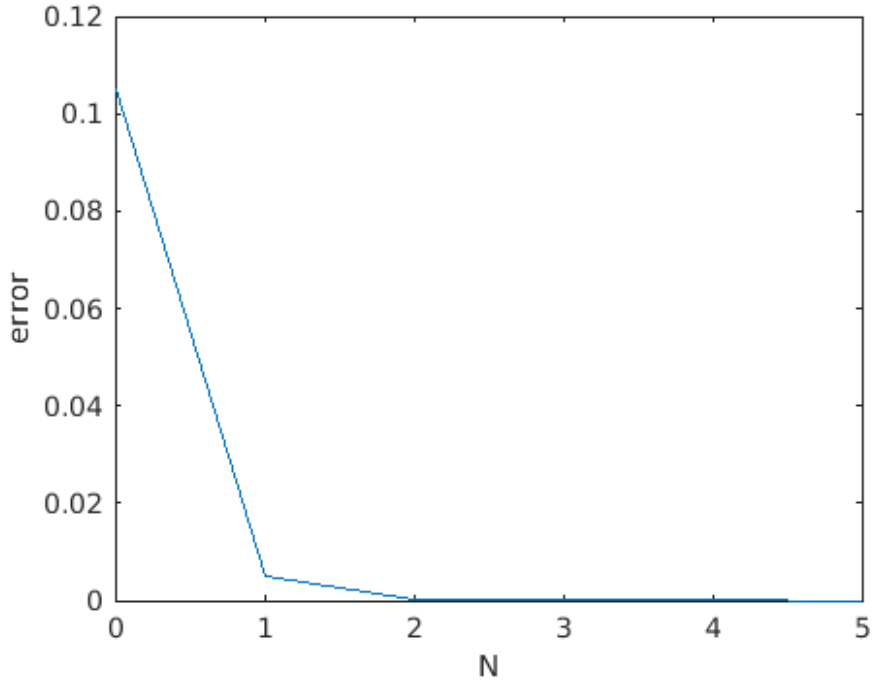
## Table of Contents



# Question 1 (a,b)

```
%(a,b)
clc
clear;
close all
n=5;
a=0.1;
expVal=1.0;
currentTerm=1.0;
for i = 1:n
    currentTerm=currentTerm*(a/i);
    expVal(i+1) = expVal(i)+currentTerm;
end
trueVal=exp(0.1);
Error = abs(expVal-trueVal)
N=0:1:5;
plot(N,Error);
xlabel("N");
ylabel("error");
```

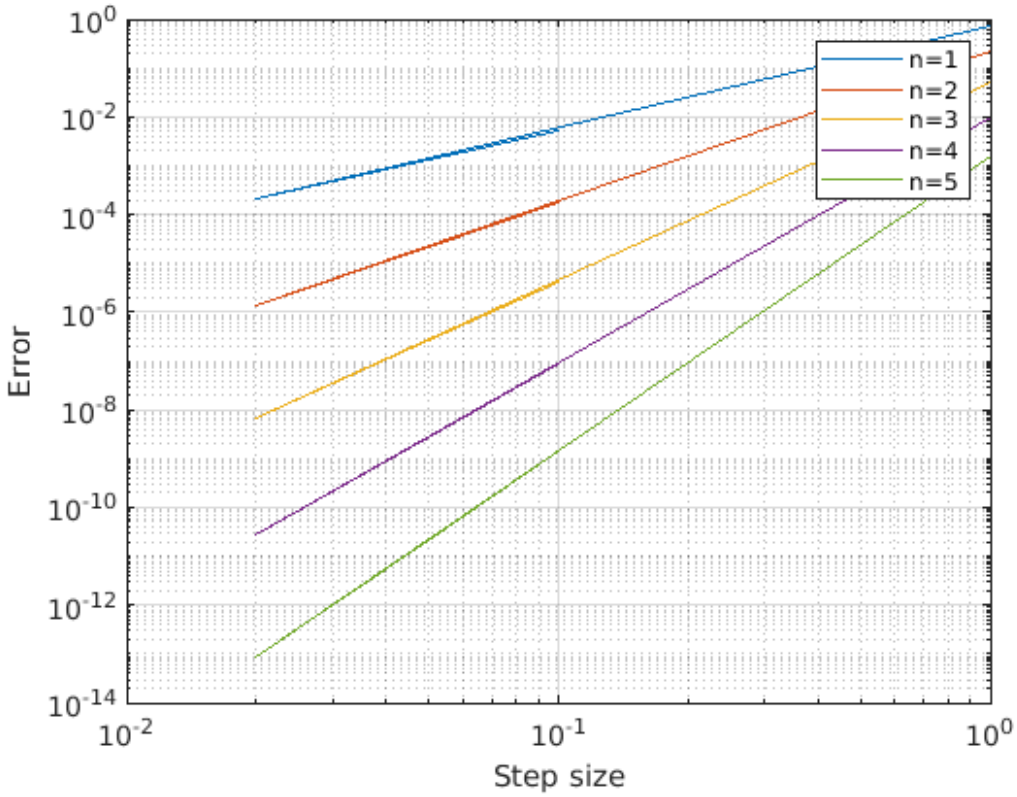
```
Error =

    0.1052    0.0052    0.0002    0.0000    0.0000    0.0000
```

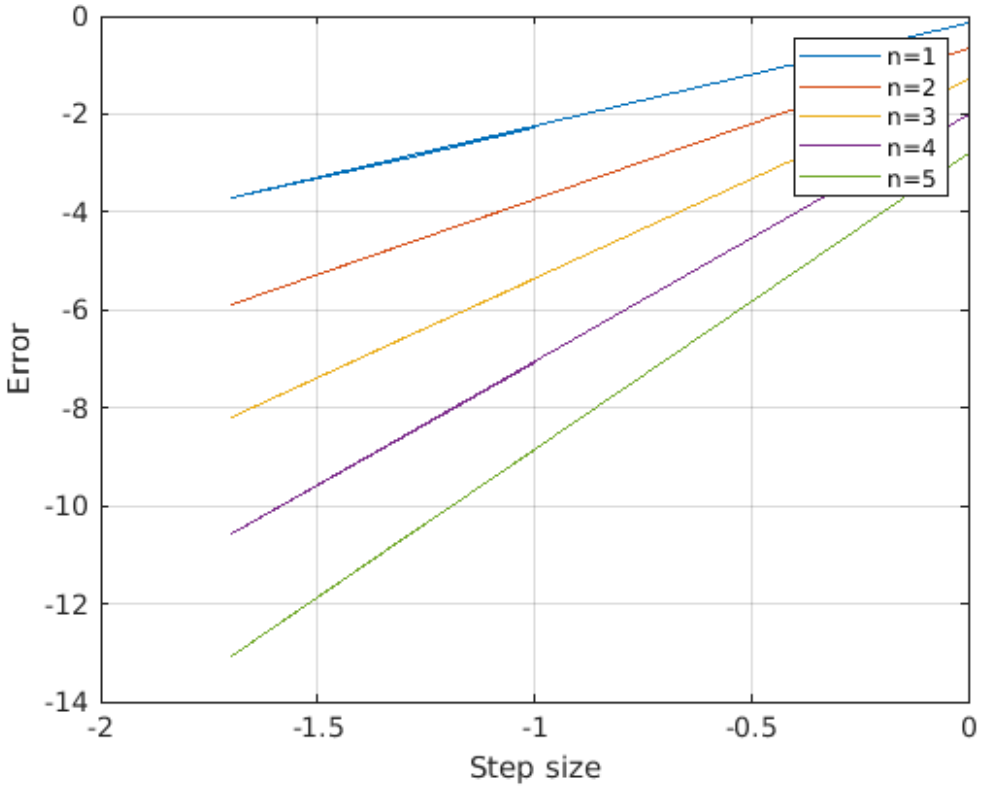


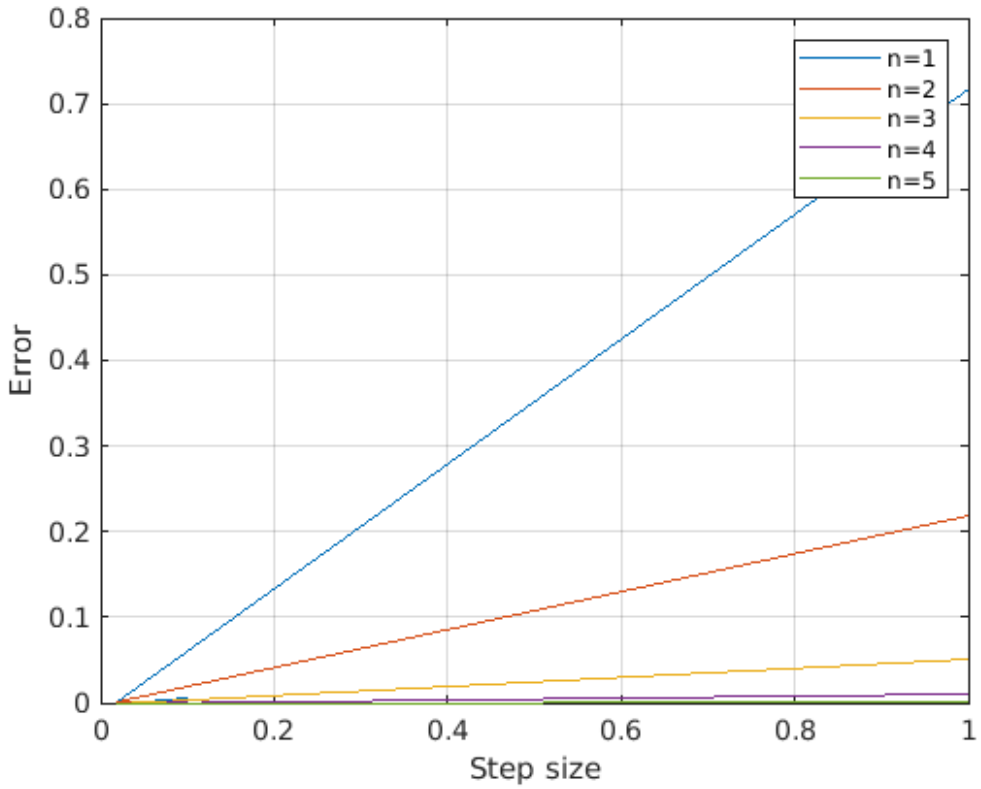

# Question 1 (c,d,e)

```
%maclaren series expansion for exp(x)
a_all = [0.1, 0.05, 0.02, 001];
Error = [];
expval = 1.0;
currentterm = 1.0;
trueVal=[];
for i=1:length(a_all)
 n=5;
 vec=[1:n];
 a=a_all(i);
 terms=a.^vec./cumprod(vec);
 expVal=1+cumsum(terms);
 trueVal=exp(a);
 Error=[Error;abs(trueVal-expVal)];
end
% displaying True Value
disp('True value : ')
trueVal=exp(a_all);
fprintf('%10.12e\n', trueVal)
% displaying Calculated Value
disp('Calculated value : ')
fprintf('%10.12e\n', expval)
disp('Errors')
disp(Error)

x2 = [-1,-1,-1,-1,-1];
x1 = [-1.699, -1.699, -1.699, -1.699, -1.699];
```

```
y2 = [-2.286, -3.767, -5.371, -7.072, -8.851];
y1 = [-3.696, -5.873, -8.174, -10.57, -13.05];
disp('Slopes :')
for i=1:5
 disp(slope(x1(i), y1(i), x2(i), y2(i)))
end
tan = [];


% plotting various plots
figure('Name','loglog plot')
loglog(a_all, Error)
xlabel('Step size')
ylabel('Error')
grid on
legend('n=1', 'n=2', 'n=3', 'n=4', 'n=5');
figure('Name','log 10 plot')
plot(log10(a_all), log10(Error))
xlabel('Step size')
ylabel('Error')
grid on
legend('n=1', 'n=2', 'n=3', 'n=4', 'n=5');
figure('Name','Plot')
plot(a_all, Error)
xlabel('Step size')
ylabel('Error')
grid on
legend('n=1', 'n=2', 'n=3', 'n=4', 'n=5');


% as the number of iterations increases erreo decreases
% lower the step size the error corresponding to the iteration is
 largest
% slope of the nth order curve is n+1 accuracy.

True value :
1.105170918076e+00
1.051271096376e+00
1.020201340027e+00
2.718281828459e+00
Calculated value :
1.000000000000e+00
Errors
    0.0052    0.0002    0.0000    0.0000    0.0000
    0.0013    0.0000    0.0000    0.0000    0.0000
    0.0002    0.0000    0.0000    0.0000    0.0000
    0.7183    0.2183    0.0516    0.0099    0.0016

Slopes :
```


Error
Step size
n=1
n=2
n=3
n=4
n=5


Error
Step size
n=1
n=2
n=3
n=4
n=5



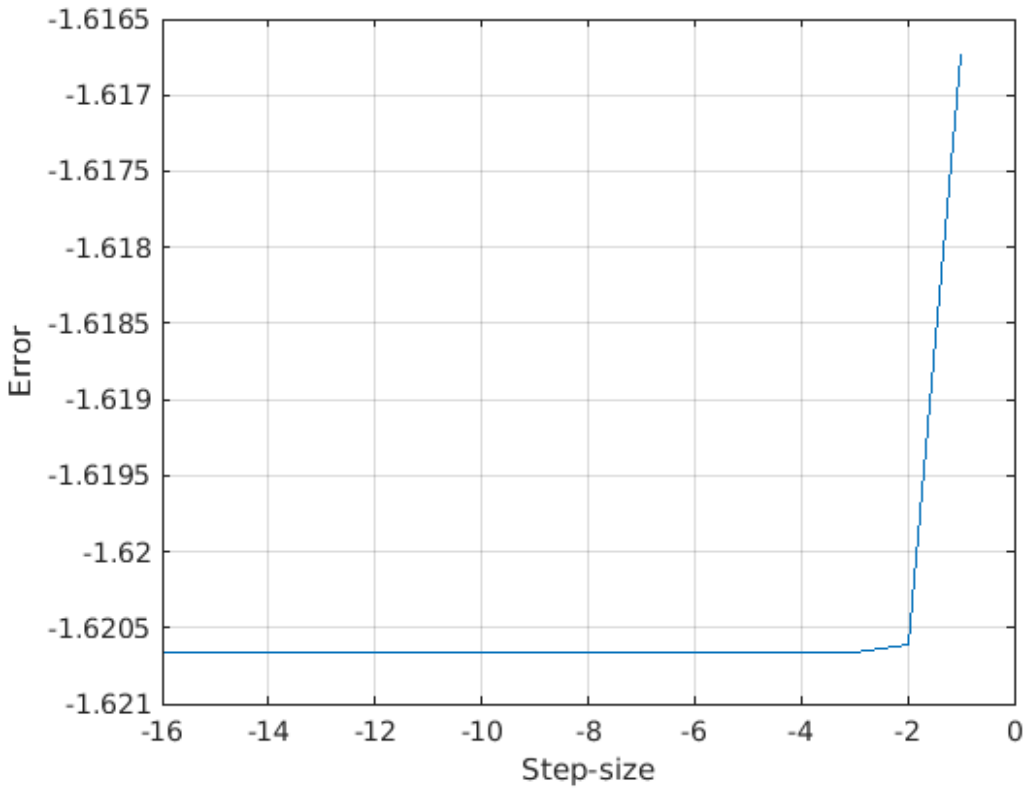

# Question 2

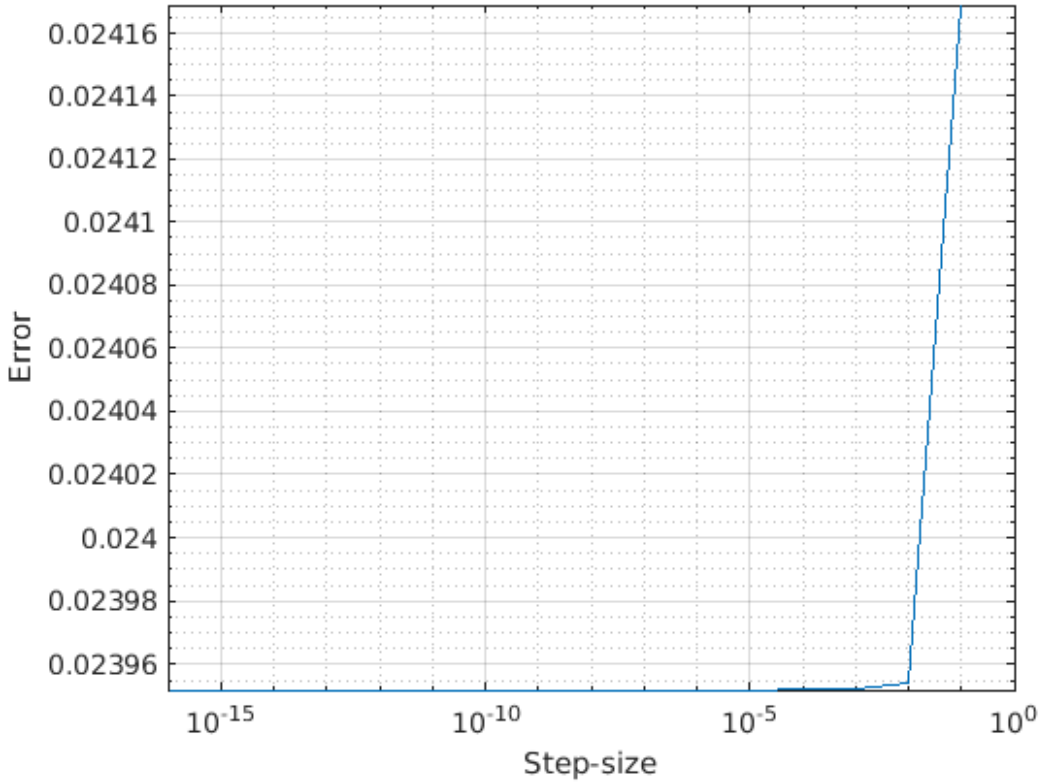
finding deravative of tan^-1(x)

```
a = 1;
trueval = 1./(1 + (a.^2));
error = [];
h = [];
for i=1:16
 h(i) = 10.^(-i);
 expval = (atan(a+h) - atan(a))/(h);
 error = [error, abs(trueval - expval)];
end
disp(error)
plot(log10(h), log10(error));
xlabel("Step-size")
ylabel("Error")
grid on
figure('Name','Log-Log Plot')
loglog(h,error);
xlabel("Step-size")
ylabel("Error")
grid on

%Interpretations:
%due to decrease in step size there is decrease in truncation error
```

```
%slope is order of accuracy n+1 th order
% due to decrease in step size there is increase in round off error
% machine presicision is step size(h) dedpendent
% smaller the h we use higher the machine precision
```

```
  Columns 1 through 7

    0.0242    0.0240    0.0240    0.0240    0.0240    0.0240    0.0240

  Columns 8 through 14

    0.0240    0.0240    0.0240    0.0240    0.0240    0.0240    0.0240

  Columns 15 through 16

    0.0240    0.0240
```






# Question 3

```
A=[1.01 0.99 ; 0.99 1.01];
%solving for equation 4
b=[2;2];
x=pinv(A)*b;
disp("x=");
disp(x);
%solving for equation 5
b=[2.02;1.98];
y=pinv(A)*b;
disp("y=");
disp(y);
% a small change in data the value of result changes drastically
%error
error=norm(y-x);
disp("error=");
disp(error);
```

```
x=
    1.0000
    1.0000

y=
     2
     0
```

```
error=
    1.4142
```

# Question 4

```
A=[1.01 0.99 ; 0.99 1.01];
%solving for equation 4(a)
b=[2;2];
b1=[2.02;1.98];
%error
error=norm(b1-b);
disp("error=");
disp(error);
%solving for equation 4(b)
x=pinv(A)*b
b_predicted=A*x
%error
error=norm(b_predicted-b);
disp("error=");
disp(error);
% Interpretation
% 1:As we can see from the process in equation 4(b) we can say that
 for
% x=x' i.e x_observed is equal to x for norm(b_predicted-b)=0
% 2:The equation4(a) is about finding backward error.

error=
    0.0283


x =

    1.0000
    1.0000


b_predicted =

     2
     2

error=
     0
```

# Functions

```
function [tan] = slope(x1,y1,x2,y2)
 tan = (y2-y1)./(x2-x1);
end
```

```
2.0172

3.0129

4.0100

5.0043

6.0072
```

*Published with MATLAB® R2020b*