
```

% Q1

% a.
a = 0.1;
trueval = exp(a);
expval = 1;
currentterm = 1;
n = 5;
% loop to calculate the expected value of a
for i = 1 : n
    % ith term of the exp is  $a^i/i! = a \cdot a^{(i-1)}/i \cdot (i-1)! = (a/i) \cdot \text{currentterm}$ 
    currentterm = currentterm*(a/i);
    expval(i + 1) = expval(i) + currentterm;
end
% error in calculating = trueval-expval
Error = abs(trueval - expval(n + 1));
fprintf('Error in calculating e^0.1 for n = %d is %1.20f', n, Error);

% b.
a = 0.1;
trueval = exp(a);
expval = 1;
currentterm = 1;
%loop to calculate the expected value of exp(a)
for i = 1 : 5
    % ith term of the exp is  $a^i/i! = a \cdot a^{(i-1)}/i \cdot (i-1)! = (a/i) \cdot \text{currentterm}$ 
    currentterm = currentterm*(a/i);
    expval(i+1)=expval(i)+currentterm;
end
% error in calculating = trueval-expval
Error = abs(trueval - expval);
n = [0:1:5];
plot(n, Error)

% c.
% %array of all the step sizes
aAll = [0.1 0.05 0.02 0.01];
%array of all the values of n
vec = [1:5];
Error = [];
%loop for calculating error of all the step sizes
for i = 1:length(aAll)
    %selecting of individual step sizes as a
    a = aAll(i);
    %makes as array of the value of nth term
    terms = a.^vec./cumprod(vec);
    %makes an array of expected value at each value of n

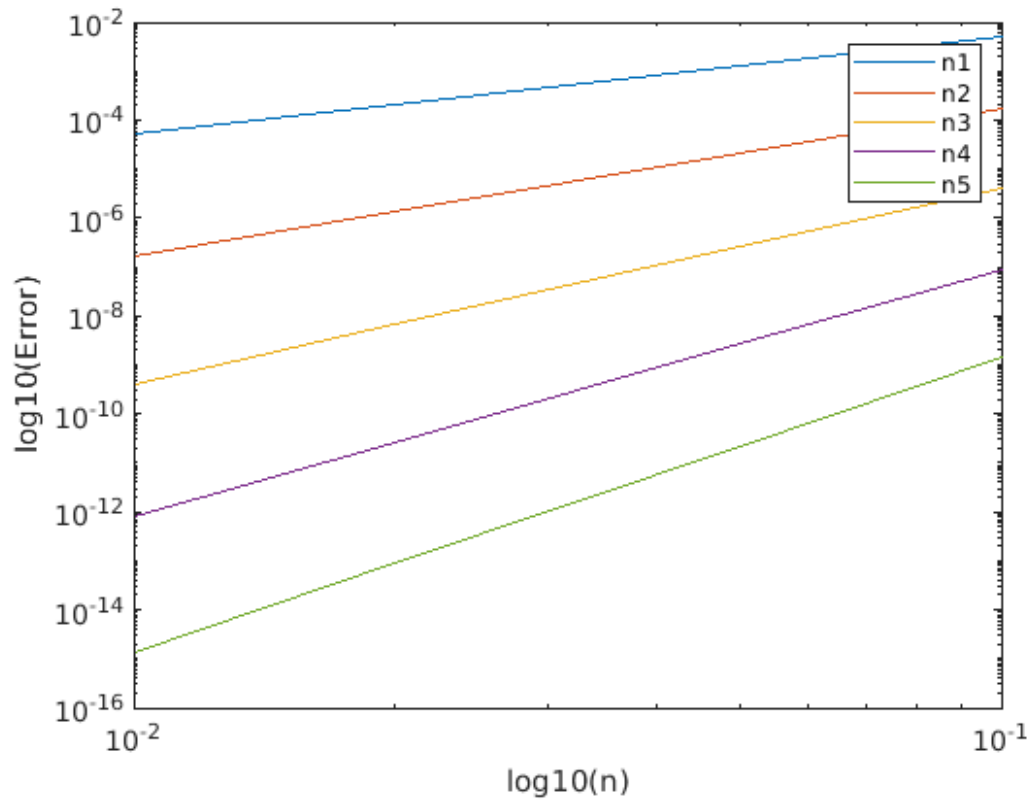
```

```

    expval = 1 + cumsum(terms);
    trueval = exp(a);
    %this makes an matrix of size 4X5 where each column tells the
    error associated with that stepsize
    Error = [Error; abs(trueval - expval)];
end
loglog(aAll,Error);
xlabel('log10(n)');
ylabel('log10(Error)');
legend('n1','n2','n3','n4','n5');

```

Error in calculating $e^{0.1}$ for $n = 5$ is 0.00000000140898115397



Published with MATLAB® R2020b