```
% Mfile name
%   mtl_aae_sim_trueerror.m

% Version:
%   Matlab R2007a

% Revised:
%   August 28, 2008

% Authors:
%   Luke Snyder, Dr. Autar Kaw
%   University of South Florida
%   kaw@eng.usf.edu
%   Website: http://numericalmethods.eng.usf.edu

% Purpose
%   To illustrate the concept of true error, absolute true error,
%   relative true error and absolute relative true error via Maclaurin
%   series of transcendental and trigonometric functions.

% Keywords
%   True Error
%   Relative True Error

% Clearing all data, variable names, and files from any other source and
% clearing the command window after each successive run of the program.
clc
clear all

% Inputs:
%    This is the only place in the program where the user makes the changes
%    based on their wishes
%        Pick the function of your desire by choosing an integer:
%        1 for exp(x); 2 for sin(x); 3 for cos(x)
         func_choice=1;

%        Maximum number of terms to use
         n=15;

%        Value of x at which function is calculated
         x=1.6;

% **********************************************************************

disp(sprintf('\n\nConcepts of True Error: True Error, Absolute True Error,'))
disp(sprintf('Relative True Error, and Absolute Relative True Error'))
disp(sprintf('\nUniversity of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu'))
disp(sprintf('Website: http://numericalmethods.eng.usf.edu'))
disp(sprintf('\nNOTE: This worksheet illustrates the use of Matlab to find'))
disp('true error, absolute true error, relative true')
disp(' error, absolute relative true error, and the number of ')
disp('significant digits correct via Maclaurin series of trancendental and')
disp('trigonometric functions.')

disp(sprintf('\n**************************Introduction***************************'))

disp(sprintf('\nThe following worksheet demonstrates how to calculate different '))
disp(sprintf('definitions related to true error, such as true error, absolute true '))
disp(sprintf('error, relative true error, and absolute relative true error. The '))
disp(sprintf('concept is demonstrated using an example of a Maclaurin series. '))
disp(sprintf('The user will choose which function to perform the calculation for '))
disp(sprintf('in the Input section of the program. The choices are given as 1 for '))
disp(sprintf('e^x, 2 for sin(x) , and 3 for cos(x). The true value of these '))
disp(sprintf('functions will be assumed as given by the Matlab commands '))
disp(sprintf('for these functions.'))

disp(sprintf('\n\n***************************Input Data*******************************'))

disp(sprintf('     func_choice = %g',func_choice))
disp(sprintf('     Value of x at which to approximate, x = %g',x))
disp(sprintf('     Maximum number of terms, n = %g',n))

% Using a long format so that all the necessary digits can be shown
format long

disp(sprintf('\n\n***************************Procedure******************************'))

% Determining which function to use and displaying it in the command
% window. Once function is determined, value is calculated using a
% Maclaurin series in a repetitive loop.

sumprevious = 0;

for i=1:1:n
    if func_choice == 1
        func = ' exp(x)';

        %This function "f" will be used to find the "true value" in calculations.
        f = inline('exp(x)');

        % Using a repetitive loop to calculate the value of function after
        % adding a term.
        sumpresent(i) = sumprevious + (x^(i-1))/(factorial(i-1));
    end

    if func_choice == 2
        func = ' sin(x)';

        %This function "f" will be used to find the "true value" in calculations.
        f = inline('sin(x)');

        % Using a repetitive loop to calculate the value of function after
        % adding a term.
        sumpresent(i) = sumprevious + ((-1)^(i-1))*((x^(2*i-1))/(factorial(2*(i-1)+1)));
    end
    if func_choice == 3
        func = ' cos(x)';

        %This function "f" will be used to find the "true value" in calculations.
        f = inline('cos(x)');

        % Using a repetitive loop to calculate the value of function after
        % adding a term.
        sumpresent(i) = sumprevious + ((-1)^(i+1))*(x^(2*i-2))/(factorial(2*i-2));
    end

    % Using Matlab to calculate true, absolute true, relative true, and
    % absolute relative true error for each term.
    TrueError(i) = f(x) - sumpresent(i);
    AbsTrueError(i) = abs(f(x) - sumpresent(i));
    RelTrueError(i) = ((f(x) - sumpresent(i))/f(x))*100;
    AbsRelTrueError(i) = abs((f(x) - sumpresent(i))/f(x))*100;

    % Displaying values in the command window.
    disp(sprintf('\nStep %g',i));
    disp(sprintf('---------------------------------------------------------'));
    disp(sprintf('\nEvaluating the function using %g terms',i));
    disp(sprintf('\n     Approximate value of function = %10.10f',sumpresent(i)));
    disp(sprintf('     True Error = (%g - %g) = %g',f(x),sumpresent(i),TrueError(i)));
    disp(sprintf('     Abs True Error = |(%g - %g)| = %g',f(x), sumpresent(i),AbsTrueError↙
(i)));
    disp(sprintf('     Rel True Error = ((%g - %g)/%g) * 100 = %g',f(x),sumpresent(i),f↙
(x),RelTrueError(i)));
    disp(sprintf('     Abs Rel True Error = |((%g - %g)/%g)| * 100 = %g',f(x),sumpresent↙
(i),f(x),AbsRelTrueError(i)));

    %This line reinitializes xold as xnew which allows for the next
    %sequence of calculations to take place.
    sumprevious = sumpresent(i);
end

% Creating a table of values based on the error calculations previously
% performed.
disp(sprintf('\n\n****************************Table of↙
Values****************************'));
disp('Terms     True           True          Abs True          Relative         Abs Rel')
disp('Used      Value          Error         Error             True             True ');
for i=1:1:n
    string = '%g        %+1.3e    %+1.3e    %+1.3e      %+1.3e       %+1.3e';
    disp(sprintf(string,i,sumpresent(i),TrueError(i),AbsTrueError(i),RelTrueError(i),↙
AbsRelTrueError(i)))
    iteration(i) = i;
end

% Plotting the Approximate value of function as a function of the number of
% terms.  The variable previously defined as "f" will be used to input the
% true value for comparisons.

% Clearing the figure windows of any previous plots.
```

```
clf

% Graph 1: Maclaurin series f(x) vs True value as a function of number of terms.
figure(1)
plot(iteration,sumpresent,'r','LineWidth',1);
hold on
xx = 0:0.01:n;
plot(xx,f(x),'g','LineWidth',2);
hold off
if func_choice == 1
    axis([1 n (f(x) - f(x)) (f(x) + f(x))])
else
    axis([1 n -1.5  1.5]);
end
title('\bfCalculated Value of f(x) Using Maclaurin Series as a Function of Number of↙
Terms');
xlabel('\bfNumber of Terms Used');
ylabel('\bff(x)');
legend('Calculated Value','True Value');

% Using Matlabs ability to position plots exactly where the user specifies.
scnsize = get(0,'ScreenSize');

% Graph 2: True and Absolute True Errors
fig2 = figure(2);
% First Subplot: True Error vs. Number of Terms Calculated
set(fig2,'Position',[0.15*scnsize(3),0.45*scnsize(3),0.7*scnsize(3),0.35*scnsize(4)])
subplot(1,2,1);
plot(iteration,TrueError,'LineWidth',2)
title('\bfTrue Error vs. Number of Terms ');
xlabel('\bfNumber of Terms Used')
ylabel('\bfTrue Error');
axis([1 n min(TrueError) max(TrueError)])
% Second Subplot: Absolute Approximate Error vs. Number of Terms Calculated
subplot(1,2,2);
plot(iteration,AbsTrueError,'LineWidth',2);
title('\bfAbsolute True vs. Number of Terms Used');
xlabel('\bfNumber of Terms Used');
ylabel('\bfAbsolute True Error');
axis([1 n min(AbsTrueError) max(AbsTrueError)])


fig3 = figure(3);
% First Subplot: Relative True Error vs. Number of Terms Calculated
set(fig3,'Position',[0.15*scnsize(3),0.1*scnsize(3),0.7*scnsize(3),0.35*scnsize(4)]);
subplot(1,2,1);
plot(iteration,RelTrueError,'LineWidth',2);
title('\bfRel True Error vs. Number of Terms Used');
xlabel('\bfNumber of Terms Used');
ylabel('\bfRelative True Error');
axis([1 n min(RelTrueError) max(RelTrueError)])
% Second Subplot: Absolute Relative True Error vs. Number of Terms
% Calculated.
subplot(1,2,2);
```

```
plot(iteration,AbsRelTrueError,'LineWidth',2);
title('\bfAbs Rel True Error vs. Number of Terms Used');
xlabel('\bfNumber of Terms Used');
ylabel('\bfAbsolute Relative True Error');
axis([1 n min(AbsRelTrueError) max(AbsRelTrueError)])



disp(sprintf('\n\n***************************Conclusion******************************'))
disp(sprintf('This worksheet shows how the number of terms taken in a '))
disp(sprintf('Maclaurin series affects the accuracy of the calculated answer '))
disp(sprintf('through the analysis of error. Note that though true error shows '))
disp(sprintf('the magnitude of the error, it does not indicate how bad the error '))
disp(sprintf('really is. Hence, relative true error is used here to give a more '))
disp(sprintf('complete picture of the state of error.'))


disp(sprintf('\n\n***************************Refrences******************************'))
disp('See: <a href = "http://numericalmethods.eng.usf.↙
edu/nbm/gen/01aae/nbm_gen_aae_txt_measuringerror.pdf">Measuring Errors</a>')

disp(sprintf('\n\nLegal Notice: The copyright for this application is owned'))
disp(sprintf('by the author(s). Neither MathWorks nor the author(s)'))
disp(sprintf('are responsible for any errors contained within and are '))
disp(sprintf('not liable for any damages resulting from the use of this'))
disp(sprintf('material. This application is intended for non-commercial,'))
disp(sprintf('non-profit use only. Contact the author for permission if'))
disp(sprintf('you wish to use this application in for-profit activities.'))
```