```
function mtl_aae_sim_quadratic

% % Mfile name
%   mtl_aae_sim_quadratic.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 use the solution of quadratic equations as a way to show the effect
%   of significant digits on round-off errors.

% Keywords
%   Quadratic Formula
%   Significant Digits

% 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.
    % The quadratic formula is derived from the standard form of a
    % quadratic equation: ax^2 + bx + c.

    % Enter coefficient "a"
    a = 0.001;

    % Enter coefficient "b"
    b = -4.946268;

    % Enter coefficient "c".
    c = 0.002;

    % Enter range of significant digits to be used.  Enter both a low and a
    % high value to be taken.
    sig_low = 7;
    sig_high = 10;


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

disp(sprintf('\n\nThe Quadratic Formula as a Way to Show the Subtraction '))
```

```
disp(sprintf('of Small Numbers'))
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 uses Matlab to use the solution of quadratic'));
disp('equations as a way to show the effect of significant digits on round-off');
disp('errors.');


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

disp(sprintf('\nThe following worksheet illustrates the use of a quadratic equation '))
disp(sprintf('solution for showing the effect of significant digits on round-off '))
disp(sprintf('errors The user will enter the a, b and c values as given by the '))
disp(sprintf('equation for the standard form of a quadratic equation: '))
disp(sprintf('ax^2 + bx + c = 0, as well as the number of significant digits to'))
disp(sprintf('be displayed in a table that will be created at the end of the program.'))
disp(sprintf('Two variations of the quadratic equation solution will be used:'))

disp(sprintf('  A)  x1 = (-b + sqrt(b^2 - 4ac))/2a, x2 = (-b - sqrt(b^2 - 4ac))/2a'))
disp(sprintf('  B)  x1 = 2c/(-b + sqrt(b^2 - 4ac)), x2 = 2c/(-b - sqrt(b^2 - 4ac))'))


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

disp(sprintf('\nThe coefficient corresponding with the quadratic term is, a = %g',a));
disp(sprintf('\nThe coefficient corresponding with the linear term is, b = %g',b));
disp(sprintf('\nThe coefficient corresponding with the constant term is, c = %g',c));
disp(sprintf('\nTherefore, the quadratic equation to be used is %gx^2 + %gx + %g',a,b,c));
disp(sprintf('\nNumber of significant digits to be taken will range from: %g to %g',↙
sig_low,sig_high));


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

% The following calculations are done using specific subfunctions which are
% at the end of the m-file.  The purpose of these functions is to perform
% basic arithmetic operations based strictly upon the number of significant
% digits specified.  All operations inside each quadratic equation
% variation are done with these functions so that the number of significant
% digits can be controlled throughout the process.

disp(sprintf('\nQuadratic formula variation A) '))
disp('x1 = (-b + sqrt(b^2 - 4ac)/2a, x2 = (-b - sqrt(b^2 - 4ac))/2a');

% The following calculations will  be performed inside a loop so that the
% number of significant digits used can be varied as specified by the user.
j = 1;
for i=sig_low:1:sig_high

    % Calculating b^2:
    b_squared = sdmul(i,b,b);
```

```
    % Calculating 4ac:
    a_c_4 = sdmul(i,(sdmul(i,a,c)),4);

    % Calculating 2a:
    a_2 = sdmul(i,2,a);

    % Calculating b^2 - 4ac:
    in_rad = sdsub(i,b_squared,a_c_4);

    % Calculating discriminate
    descrim = sdsqrt(i,in_rad);

    % Calculating first root "x1" = (-b + discriminate)/2a:
    x1(j) = sddiv(i,(sdadd(i,-b,descrim)),a_2);

    % Calculating second root "x2" = (-b - discriminate)/2a:
    x2(j) = sddiv(i,(sdsub(i,-b,descrim)),a_2);

    j = j+1;
end


disp(sprintf('\nQuadratic formula variation B)'));
disp('x1 = 2c/(-b + sqrt(b^2 - 4ac)), x2 = 2c/(-b - sqrt(b^2 - 4ac))');


j = 1;
for i=sig_low:1:sig_high

    % Calculating 2c:
    c_2 = sdmul(i,2,c);

    % Calculating b^2:
    b_squared2 = sdmul(i,b,b);

    % Calculating 4ac:
    a_c_4_2 = sdmul(i,(sdmul(i,a,c)),4);

    % Calculating b^2 - 4ac:
    in_rad2 = sdsub(i,b_squared2,a_c_4_2);

    % Calculating discriminate:
    descrim2 = sdsqrt(i,in_rad2);

    % Calculating first root "x1_2" 2c/(-b - discriminate):
    x1_2(j) = sddiv(i,c_2,(sdsub(i,-b,descrim2)));

    % Calculating second root "x1_2" 2c/(-b + discriminate):
    x2_2(j) = sddiv(i,c_2,(sdadd(i,-b,descrim2)));

    j = j+1;
end
```

```
% Inputting these values in a table for comparison.  A loop is used to show
% the entire data array.

disp(sprintf('\n\n**********************************Table of↙
Values********************************'));
disp('Sig           First Root (x1):                         Second Root (x2):      ')
disp↙
('-------------------------------------------------------------------------------------')
disp('Digits    varA               varB                    varA               varB↙
' )
disp↙
('-------------------------------------------------------------------------------------')
j=1;
for i=sig_low:1:sig_high
    disp(sprintf('%g         %1.9e   %1.9e       %1.9e    %1.9e',i,x1(j),x1_2(j),x2(j),↙
x2_2(j)));
    j=j+1;
end

% Creating plots: x1:variation1 vs. significant digits, x1:variation2, vs
% significant digits.
fig1 = figure(1);

% Using Matlab to position the plots where the programmer specifies.  The
% first bar graph will show values of x1 for both variations of the
% quadratic function.
scnsize = get(0,'ScreenSize');
set(fig1,'Position',[0.01*scnsize(3),0.25*scnsize(3),0.50*scnsize(3),0.45*scnsize(4)]);
bar1 = bar(sig_low:sig_high,x1,'FaceColor','b','EdgeColor','b');
title('\bfValue of First Root (x1) as a Function of Significant Digits');
xlabel('\bfNumber of Significant Digits')
ylabel('\bfValue of Quadratic Root (x1)')
set(bar1,'BarWidth',0.55);
hold on;
bar2 = bar(sig_low:sig_high,x1_2,'FaceColor','r','EdgeColor','r');
set(bar2,'BarWidth',0.3);
legend('Variation 1','Variation 2')

hold off;

% This bar graph will show values of x2 for both variations of the
% quadratic function.
fig2 = figure(2);
set(fig2,'Position',[0.52*scnsize(3),0.25*scnsize(3),0.50*scnsize(3),0.45*scnsize(4)]);
bar1 = bar(sig_low:sig_high,x2,'FaceColor','g','EdgeColor','g');
title('\bfValue of Second Root (x2) as a Function of Significant Digits');
xlabel('\bfNumber of Significant Digits');
ylabel('\bfValue of Quadratic Root (x2)');
set(bar1,'BarWidth',0.55);
hold on;
bar2 = bar(sig_low:sig_high,x2_2','FaceColor','k','EdgeColor','k');
set(bar2,'BarWidth',0.3)
legend('Variation 1','Variation 2')
```

```
disp(sprintf('\n\n*************************Conclusion*****************************'))
disp(sprintf('Subtraction of numbers that are nearly equal can result in '))
disp(sprintf('unwanted inaccuracies. The number of significant digits used in '))
disp(sprintf('calculations plays a large role in the creation of these inaccuracies '))
disp(sprintf('and the magnitude of the round-off errors. Hence, when the '))
disp(sprintf('accuracy of calculations is critical, it is necessary to understand '))
disp(sprintf('possible sources of error and how they are best avoided.'))


disp(sprintf('\n\n*************************Refrences*****************************'))
disp('See: <a href = "http://numericalmethods.eng.usf.↙
edu/mtl/gen/01aae/mtl_gen_aae_txt_sourcesoferror.pdf">Sources of Error</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.'))



hold off;

%   The following functions modify standard arithmetic operators allowing
%   computation with the appropriate number of significant digits. These
%   redefined operators are then used in the Forward Difference
%   Approximation method to generate a solution that was computed with the
%   number of significant digits specified.

    function q=sdscale(sd,k)
        if k==0;
           m=sd;
        else
           m=sd-floor(log10(abs(k))+1);
           q=k*10^m;
           q=floor(q)*10^(-m);
        end
    end
%--------------------------------------------------------------------------
    function d_sig=sdadd(sd,a,b)
        a_sig = sdscale(sd,a);
        b_sig = sdscale(sd,b);
        d = a_sig + b_sig;
        d_sig = sdscale(sd,d);
    end

    function d_sig=sdsub(sd,a,b)
        a_sig = sdscale(sd,a);
        b_sig = sdscale(sd,b);
        d = a_sig - b_sig;
        d_sig = sdscale(sd,d);
    end
```

```
    function d_sig=sdmul(sd,a,b)
        a_sig = sdscale(sd,a);
        b_sig = sdscale(sd,b);
        d = a_sig * b_sig;
        d_sig = sdscale(sd,d);
    end

    function d_sig=sddiv(sd,a,b)
        a_sig = sdscale(sd,a);
        b_sig = sdscale(sd,b);
        d = a_sig / b_sig;
        d_sig = sdscale(sd,d);
    end

    function a_sq_sig=sdsqrt(sd,a)
        a_sig = sdscale(sd,a);
        a_sq = sqrt(a_sig);
        a_sq_sig = sdscale(sd,a_sq);
    end


end
```