```
function Backward
clc
clear all

% Revised:
% February 10, 2008

% Authors:
% Ana Catalina Torres, Dr. Autar Kaw

% Purpose

%   To illustrate the concept of approximate error, absolute approximate
%   error, relative approximate error and absolute relative approximate
%   error, number of significant digits correct when using Difference
%   Approximation of the first derivative of continuous functions method.

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

%    This is the only place in the program where the user makes changes to
%    the data
%        Function f(x)

    function k=f(x)
     k=exp(2*x);
    end

%   Declaring 'x' as a variable

x = sym('x','real');

%        Value of x at which f '(x) is desired, xv

         xv=4;

%        Starting step size, h

         h=0.2;

%        Number of times starting step size is halved

         n=12;

%---------------------------------------------------------------------------
disp(sprintf('                 Differentiation of Continuous Functions'))
disp(sprintf('          Backward Difference Approximation of the First Derivative'))
disp(sprintf('                          Ana Catalina Torres, Autar Kaw'))
disp(sprintf('                            University of South Florida'))
disp(sprintf('                              United States of America'))
disp(sprintf('                                  kaw@eng.usf.edu'))
```

```
%-----------------------------------------------------------------------------
disp(sprintf('\n\n**************************** Introduction
*****************************'))

disp(sprintf('\nThis worksheet demonstrates the use of Matlab to illustrate Backward '))
disp(sprintf('Difference Approximation of the first derivative of continuous functions.'))
disp(sprintf('Backward Difference Approximation of the first derivative uses a point h'))
disp(sprintf('behind of the given value of x at which the derivative of f(x) is to
be\nfound.'))

disp(sprintf('\n\n************************** Section 1: Input
***************************'))
format short g
disp(sprintf('\nThe following simulation approximates the first derivative of a'))
disp(sprintf('function using Backward Difference Approximation. \n\nThe user inputs are'))
disp(sprintf('      a) function,\nf(x)=%g'))
disp(f(x))
disp(sprintf('      b) point at which the derivative is to be found, xv = %g',xv))
disp(sprintf('      c) starting step size, h = %g',h))
disp(sprintf('      d) number of times user wants to halve the starting step size, n = %
g',n))

disp(sprintf('\nThe outputs include'))
disp(sprintf('      a) approximate value of the derivative at the point and given '))
disp(sprintf('         initial step size'))
disp(sprintf('      b) exact value'))
disp(sprintf('      c) true error, absolute relative true error, approximate error '))
disp(sprintf('         and absolute relative approximate error, least number of '))
disp(sprintf('         correct significant digits in the solution as a function of '))
disp(sprintf('         step size.'))

disp(sprintf('\nAll the information must be entered at the beginning of the M-File.'))


disp(sprintf('\n\n************************ Section 2: Simulation
*************************'))
disp(sprintf('\nThe exact value EV of the first derivative of the equation:'))
disp(sprintf('\nFirst, using the derivative command the solution is found. '))
Soln=diff(f(x))
disp(sprintf('In a second step, the exact value of the derivative is shown'))
disp(sprintf('The exact solution of the first derivative is:'))
Ev=subs(Soln,x,xv)

disp(sprintf('\nAn internal loop calculates the following:'))
disp(sprintf('Av: Approximate value of the first derivative using Backward Difference
\nApproximation'))
disp(sprintf('Ev: Exact value of the first derivative'))
disp(sprintf('Et: True error'))
disp(sprintf('et: Absolute relative true percentage error'))
disp(sprintf('Ea: Approximate error'))
disp(sprintf('ea: Absolute relative approximate percentage error'))
disp(sprintf('Sig: Least number of correct significant digits in an approximation'))
```

```
j=zeros(1,n);
N=zeros(1,n);
H=zeros(1,n);
Av=zeros(1,n);
Et=zeros(1,n);
et=zeros(1,n);
Ea=zeros(1,n);
ea=zeros(1,n);
Sig=zeros(1,n);

%   The next loop calculates the previously mentioned values:
for i=1:n
    j(i)=i;
    N(i)=2^(i-1);
    H(i)=h/(N(i));
    Av(i)=(f(xv)-f(xv-H(i)))/(H(i));
    Et(i)=Ev-Av(i);
    et(i)=abs(Et(i)/Ev*100);
    if i>1
        Ea(i)=Av(i)-Av(i-1);
        ea(i)=abs(Ea(i)/Av(i)*100);
        if 0<ea(i)<5
            Sig(i)=floor((2-log10(ea(i)/0.5)));
        else
           Sig(i)=0;
        end
    end
   end
%   The loop halves the value of the starting step size n times. Each time
%   the approximate value of the derivative is calculated and saved in a
%   vector. The approximate error is calculated after at least two
%   approximate values of the derivative have been saved. The number of
%   significant digits is calculated. If the number of significant digits
%   calculated is less than zero, it is shown as zero.

disp(sprintf('\n\n******************** Section 3: Table of Values↙
***********************'))

disp(sprintf('\nThe next tables show the step size value, approximate value, true↙
error,'))
disp(sprintf('the absolute relative true percentage error, the approximate error, the'))
disp(sprintf('absolute relative approximate percentage error and the least number of '))
disp(sprintf('correct significant digits in an approximation as a function of the↙
step\nsize value.\n\n'))
disp('       H            Av            Et             et')
Results=[H' Av' Et' et'];
disp(sprintf('\n'))
disp(Results)
disp(sprintf('\n'))
disp('       H            Av            Ea             ea              Sig')
disp(sprintf('\n'))
ResultsCont=[H' Av' Ea' ea' Sig'];
```

```
disp(ResultsCont)

disp(sprintf('\n\n**************************** Section 4: Graphs↙
***************************'))

disp(sprintf('\nThe attached graphs show the approximate solution, absolute relative↙
true'))
disp(sprintf('error, absolute relative approximate error and least number of↙
significant'))
disp(sprintf('digits as a function of the number of iterations.\n'))

set(0,'Units','pixels')
  scnsize=get(0,'ScreenSize');
  wid=round(scnsize(3));
  hei=round(0.9*scnsize(4));
  wind=[1, 1, wid, hei];
  figure('Position',wind)

%   Approximate Solutions vs. Step size:

subplot(2,2,1); plot(H,Av,'LineWidth',2,'Color','g')
xlabel('Step Size')
ylabel('Approximate Value')
title({'Approximate Solution of the First Derivative using'; 'Backward Difference↙
Approximation as a Function of Step Size'})

%   Absolute relative true error vs. Step size:

subplot(2,2,2); plot(H,et,'LineWidth',2,'Color','y')
xlabel('Step Size')
ylabel('Absolute Relative True Error')
title('Absolute Relative True Percentage Error as a Function of Step Size')

%   Absolute relative approximate error vs. Step size:

subplot(2,2,3); plot(H(2:n),ea(2:n),'LineWidth',2,'Color','m')
xlabel('Step Size')
ylabel('Absolute Relative Approximate Error')
title('Absolute Relative Approximate Percentage Error as a Function of Step Size')

%   Number of significant digits vs. the number of iterations.
subplot(2,2,4);
bar(j,Sig);
xlabel('Number of iterations');
ylabel('Number of Significant digits');
title('Number of Significant Digits as function of Number of Iterations');


disp(sprintf('\n\n****************************** References↙
*******************************'))

disp(sprintf('\nNumerical Differentiation of Continuous Functions. See'))
```

```
disp(sprintf('http://numericalmethods.eng.usf.
edu/mws/gen/02dif/mws_gen_dif_txt_\ncontinuous.pdf'))

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

disp(sprintf('\n1. The velocity of a rocket is given by\n\n              v(t)=2000*ln
(140000/(140000-2100t))-9.8*t'))

disp(sprintf('\nUse Backward Divided Difference method with a step size of 0.25 to find'))
disp(sprintf('the acceleration at t=5s. Compare with the exact answer and study the
effect\nof the step size.'))

disp(sprintf('\n\n2. Look at the true error vs. step size data for problem # 1. Do you
see'))
disp(sprintf('a relationship between the value of the true error and step size ? \nIs this
concidential? '))
disp(sprintf('\n3. Choose a step size of h=10^-10 in problem #1. Keep halving the step'))
disp(sprintf('size. Does the approximate value get closer to the exact result or does\nthe
result seem odd?'))

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

disp(sprintf('\nTo obtain more accurate values of the first derivative using Backward '))
disp(sprintf('Difference Approximation, the step size needs to be small. As the '))
disp(sprintf('spreadsheet shows, the smaller the step size value is, the approximation'))
disp(sprintf('is closest to the exact value. However, too small a step size can result'))
disp(sprintf('in noticeable round-off errors, and hence giving highly inaccurate
results.'))

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

end
```