```
clc
clf
clear all

%*********************************************************************
%
% INPUTS
%
% Click the run button and refer to the command window
% These are the inputs that can be modified by the user
%
% f(x), the function to integrate

  f= @(x) 2000*log(1400/21./x)-9.8*x ;

% a, the lower limit of integration

  a=8 ;

% b, the upper limit of integration

  b=10 ;

% n, the maximum number of segments

  n=10 ;

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

disp(sprintf('\n\nSimulation of the Trapezoidal Method'))
disp(sprintf('University of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu\n'))

disp(sprintf↙
('\n*****************************Introduction*******************************'))
disp('Trapezoidal rule is based on the Newton-Cotes formula that if one approximates the')
disp('integrand by an nth order polynomial, then the integral of the function is↙
approximated')
disp('by the integral of that nth order polynomial. Integrating polynomials is simple and↙
is')
disp('based on calculus. Trapezoidal rule is the area under the curve for a first order')
disp('polynomial (straight line).')

disp(sprintf('\n\n*******************************Input↙
Data*******************************\n'))
disp(sprintf('     f(x), integrand function'))
disp(sprintf('     a = %g, lower limit of integration ',a))
disp(sprintf('     b = %g, upper limit of integration ',b))
disp(sprintf('     n = %g, number of subdivisions',n))
format short g
```

```
% Calculate the spacing parameter
disp(sprintf('\nFor this simulation, the following parameter is constant.\n'))
h=(b-a)/n ;
disp(sprintf('     h = ( b - a ) / n '))
disp(sprintf('       = ( %g - %g ) / %g ',b,a,n))
disp(sprintf('       = %g',h))

disp(sprintf↙
('\n******************************Simulation*********************************\n'))
sum=0 ;
disp('The approximation is expressed as')
disp(' ')
disp('     approx = h * ( 0.5*f(a) + Sum (i=1,n-1) f(a+i*h) + 0.5*f(b) )')
disp(' ')


disp('1) Begin summing all function values at points between a and b not')
disp('   including a and b.')
disp(' ')
disp('       Sum (i=1,n-1) f(a+i*h)')
disp(' ')
for i=1:n-2
  disp(sprintf('          f(%g)',a+i*h))
end
disp(sprintf('       +  f(%g)',a+(n-1)*h))
disp(sprintf('       ------------'))
for i=1:n-2
  sum=sum+f(a+i*h) ;
  disp(sprintf('          %g',f(a+i*h)))
end
sum=sum+f(a+(n-1)*h) ;
disp(sprintf('       +  %g',f(a+(n-1)*h)))
disp(sprintf('       ------------'))
disp(sprintf('          %g\n',sum))


disp('2) Add to this 0.5*(f(a) + f(b))')
disp(' ')
disp(sprintf('     %g + 0.5*(%g + %g) =',sum,f(a),f(b)))
disp(sprintf('     %g + %g = %g',sum,0.5*(f(a)+f(b)),sum+0.5*(f(a)+f(b))))
sum=sum+0.5*(f(a)+f(b)) ;
disp(' ')


disp('3) Multiply this by h to get the approximation for the integral.')
disp(' ')
disp(sprintf('     approx = h * %g',sum))
disp(sprintf('     approx = %g * %g',h,sum))
approx=h*sum ;
disp(sprintf('     approx = %g',approx))

disp(sprintf('\n\n**************************Results****************************'))
```

```
% The following finds what is called the 'Exact' solution
exact = quad(f,a,b) ;

disp(sprintf('\n   Approximate = %g',approx))
disp(sprintf('   Exact       = %g',exact))
disp(sprintf('\n   True Error = Exact - Approximate'))
disp(sprintf('              = %g - %g',exact,approx))
disp(sprintf('              = %g',exact-approx))
disp(sprintf('\n   Absolute Relative True Error Percentage'))
disp(sprintf('              = | ( Exact - Approximate ) / Exact | * 100'))
disp(sprintf('              = | %g / %g | * 100',exact-approx,exact))
disp(sprintf('              = %g',abs( (exact-approx)/exact )*100))

disp(sprintf('\nThe trapezoidal approximation can be more accurate if we made our'))
disp(sprintf('segment size smaller (that is, increasing the number of segments).\n\n'))

% The following code is needed to produce the trapezoidal method
% visualization.
x(1)=a ;
y(1)=f(a) ;
hold on
for i=1:n
   x(i+1)=a+i*h ;
   y(i+1)=f(x(i+1)) ;
   fill([x(i) x(i) x(i+1) x(i+1)], [0 y(i) y(i+1) 0], 'y')
end
xrange=a:(b-a)/1000:b;
plot(xrange,f(xrange),'k','Linewidth',2)
title('Integrand function and Graphical Depiction of Trapezoidal Method')
```