```
clc
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=12 ;

% n, the maximum number of Gauss Points (1-10)

  n=10 ;

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

disp(sprintf('\n\nConvergence of Gaussian Quadrature'))
disp(sprintf('University of South Florida'))
disp(sprintf('United States of America'))
disp(sprintf('kaw@eng.usf.edu\n'))

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

disp('The following simulation illustrates the convergence of Gaussian Quadrature')
disp('applied to numerically integrate functions. This section is the')
disp('only section where the user interacts with the program.  The user ')
disp('enters a function in the form f(x), the lower and upper limit of integration,')
disp('a and b, and the number of subdivisions to take. By entering this data, the')
disp('program will calculate the exact (Matlab numerical value if it is not exact)')
disp('value of the solution, followed by the results using Gaussian Quarature')
disp('with 1-10 Gauss Points. The program will also display the true error,')
disp('the absolute relative percentage true error, the approximate error, the absolute')
disp('relative aprroximate percentage error, and the least number of significant ')
disp('digits correct all as a function of number of Gauss Points used.')


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

```
disp(sprintf('     f(x), function which defines the integrand'))
disp(sprintf('     a = %g, lower limit of integration',a))
disp(sprintf('     b = %g, upper limit of integration',b))
disp(sprintf('     n = %g, number of Gauss Points to use',n))
format short g

% Setup Gaussian Coefficients and Evaluation Points
x_values = zeros(10,10) ;
c_values = zeros(10,10) ;

i=1 ;
x_values(i,1) = 0.0 ;
c_values(i,1) = 2.0 ;

i=2 ;
x_values(i,1) = -0.5773502691896260 ;
x_values(i,2) = -x_values(i,1) ;
c_values(i,1) = 1.0 ;
c_values(i,2) = 1.0 ;

i=3 ;
x_values(i,1) = -0.7745966692414830 ;
x_values(i,2) = 0.0 ;
x_values(i,3) = -x_values(i,1) ;
c_values(i,1) = 0.5555555555555560 ;
c_values(i,2) = 0.8888888888888890 ;
c_values(i,3) =c_values(i,1) ;

i=4 ;
x_values(i,1) = -0.8611363115940530 ;
x_values(i,2) = -0.3399810435848560 ;
x_values(i,3) = -x_values(i,2) ;
x_values(i,4) = -x_values(i,1) ;
c_values(i,1) = 0.3478548451374540 ;
c_values(i,2) = 0.6521451548625460 ;
c_values(i,3) =c_values(i,2) ;
c_values(i,4) =c_values(i,1) ;

i=5 ;
x_values(i,1) = -0.9061798459386640 ;
x_values(i,2) = -0.5384693101056830 ;
x_values(i,3) = 0.0 ;
x_values(i,4) = -x_values(i,2) ;
x_values(i,5) = -x_values(i,1) ;
c_values(i,1) = 0.2369368850561890 ;
c_values(i,2) = 0.4786386704993660 ;
c_values(i,3) = 0.5688888888888890 ;
c_values(i,4) =c_values(i,2) ;
c_values(i,5) =c_values(i,1) ;

i=6 ;
x_values(i,1) = -.9324695142032520 ;
```

```
x_values(i,2) = -.6612093864662650 ;
x_values(i,3) = -.2386191860831970 ;
x_values(i,4) = -x_values(i,3) ;
x_values(i,5) = -x_values(i,2) ;
x_values(i,6) = -x_values(i,1) ;
c_values(i,1) = 0.1713244923791700 ;
c_values(i,2) = 0.3607615730481390 ;
c_values(i,3) = 0.4679139345726910 ;
c_values(i,4) =c_values(i,3) ;
c_values(i,5) =c_values(i,2) ;
c_values(i,6) =c_values(i,1) ;

i=7 ;
x_values(i,1) = -0.9491079123427590 ;
x_values(i,2) = -0.7415311855993940 ;
x_values(i,3) = -0.4058451513773970 ;
x_values(i,4) = 0.0 ;
x_values(i,5) = -x_values(i,3) ;
x_values(i,6) = -x_values(i,2) ;
x_values(i,7) = -x_values(i,1) ;
c_values(i,1) = 0.1294849661688700 ;
c_values(i,2) = 0.2797053914892770 ;
c_values(i,3) = 0.3818300505051190 ;
c_values(i,4) = 0.4179591836734690 ;
c_values(i,5) =c_values(i,3) ;
c_values(i,6) =c_values(i,2) ;
c_values(i,7) =c_values(i,1) ;

i=8 ;
x_values(i,1) = -0.9602898564975360 ;
x_values(i,2) = -0.7966664774136270 ;
x_values(i,3) = -0.5255324099163290 ;
x_values(i,4) = -0.1834346424956500 ;
x_values(i,5) = -x_values(i,4) ;
x_values(i,6) = -x_values(i,3) ;
x_values(i,7) = -x_values(i,2) ;
x_values(i,8) = -x_values(i,1) ;
c_values(i,1) = 0.1012285362903760 ;
c_values(i,2) = 0.2223810344533740 ;
c_values(i,3) = 0.3137066458778870 ;
c_values(i,4) = 0.3626837833783620 ;
c_values(i,5) =c_values(i,4) ;
c_values(i,6) =c_values(i,3) ;
c_values(i,7) =c_values(i,2) ;
c_values(i,8) =c_values(i,1) ;

i=9 ;
x_values(i,1) = -0.9681602395076260 ;
x_values(i,2) = -0.8360311073266360 ;
x_values(i,4) = -0.6133714327005900 ;
x_values(i,4) = -0.3242534234038090 ;
x_values(i,5) = 0.0 ;
```

```
x_values(i,6) = -x_values(i,4) ;
x_values(i,7) = -x_values(i,3) ;
x_values(i,8) = -x_values(i,2) ;
x_values(i,9) = -x_values(i,1) ;
c_values(i,1) = 0.0812743883615740 ;
c_values(i,2) = 0.1806481606948570 ;
c_values(i,3) = 0.2606106964029350 ;
c_values(i,4) = 0.3123470770400030 ;
c_values(i,5) = 0.3302393550012600 ;
c_values(i,6) =c_values(i,4) ;
c_values(i,7) =c_values(i,3) ;
c_values(i,8) =c_values(i,2) ;
c_values(i,9) =c_values(i,1) ;

i=10 ;
x_values(i,1)  = -0.9739065285171720 ;
x_values(i,2)  = -0.8650633666889850 ;
x_values(i,3)  = -0.6794095682990240 ;
x_values(i,4)  = -0.4333953941292470 ;
x_values(i,5)  = -0.1488743389816310 ;
x_values(i,6)  = -x_values(i,5) ;
x_values(i,7)  = -x_values(i,4) ;
x_values(i,8)  = -x_values(i,3) ;
x_values(i,9)  = -x_values(i,2) ;
x_values(i,10) = -x_values(i,1) ;
c_values(i,1)  = 0.0666713443086880 ;
c_values(i,2)  = 0.1494513491505810 ;
c_values(i,3)  = 0.2190863625159820 ;
c_values(i,4)  = 0.2692667193099960 ;
c_values(i,5)  = 0.2955242247147530 ;
c_values(i,6)  = c_values(i,5) ;
c_values(i,7)  = c_values(i,4) ;
c_values(i,8)  = c_values(i,3) ;
c_values(i,9)  = c_values(i,2) ;
c_values(i,10) = c_values(i,1) ;

% Exact Solution
exact = quad(f,a,b) ;

for i=0:n-1

  NN(i+1)=i+1 ;

  % Apply the approximation
  integral = 0 ;
  for j=1:NN(i+1)
    tempx = x_values(i+1,j)*(b-a)/2+(b+a)/2 ;
    integral = integral + f(tempx)*c_values(i+1,j) ;
  end
  integral =(b-a)/2*integral ;
  YY(i+1)=integral ;
```

```
  % Compute Errors
  Et(i+1)=exact-integral ;
  if(exact > 0)
    Etabs(i+1)=abs((integral-exact)/exact) ;
  else
    Etabs(i+1)=0 ;
  end
  if(i > 0)
    Ea(i+1)=YY(i+1)-YY(i) ;
  end
  if(i>0 && YY(i)>0)
    Eaabs(i+1)=abs((YY(i+1)-YY(i))/YY(i)) ;
    SD(i+1)=floor((2-log10(Eaabs(i+1)/0.5))) ;
    if(SD(i+1)<0)
      SD(i+1)=0
    end
  else
    Ea(1)=0 ;
    Eaabs(1)=0 ;
    SD(1)=0 ;
  end

end


disp(sprintf('\n\n***********************Table of 
Values*****************************\n'))

disp('        Approx        True     Relative      Approx   Rel Appr   Sig   ')
disp(' n     Integral       Error   True Error     Error     Error   Digits ')
disp('-----------------------------------------------------------------------')

% The following displays the results in a table. Conditional statements are
% used to avoid the printing of 'inf' in cases where the approximation is
% zero.
for i=1:n
if(i > 1)
  if(exact || YY(i) > 0)
    disp(sprintf('%4i  %+1.3e  %+1.3e  %+1.3e  %+1.3e  %+1.3e  %2i',NN(i),YY(i),Et(i),
Etabs(i),Ea(i),Eaabs(i),SD(i) ))
  else
    disp(sprintf('%4i  %+1.3e  %+1.3e     n/a      %+1.3e      n/a     n/a',NN(i),YY(i),
Etabs(i),Ea(i)))
  end
else
  disp(sprintf('%4i  %+1.3e  %+1.3e  %+1.3e      n/a         n/a     n/a',NN(i),YY(i),Et
(i),Etabs(i)))
end

end
disp('-----------------------------------------------------------------------')
```

```
% The following generates 3 plots. This function detects information about your
% screensize and tries to then place/size the graphs accordingly.
scnsize = get(0,'ScreenSize');

% Graph 1: Approximation and True Errors
fig2=figure ;
set(fig2,'Position',[0.2*scnsize(3),0.2*scnsize(3),0.6*scnsize(3),0.2*scnsize(4)]) ;
subplot(1,3,1); plot(NN,YY,'-O','LineWidth',2,'Color',[1 0 0]);
title('Appr. Integral vs No. of Gauss Points')

subplot(1,3,2); plot(NN,Et,'-O','LineWidth',2,'Color',[0 0 1]);
title('Et vs No. of Gauss Points')

subplot(1,3,3); plot(NN,Etabs,'-O','LineWidth',2,'Color',[0 0 1]);
title('Abs et vs No. of Gauss Points')

% Graph 2: Relative Errors and Significant Digits
fig = figure ;
set(fig,'Position',[0.2*scnsize(3),0,0.6*scnsize(3),0.2*scnsize(4)]) ;
subplot(1,3,1); plot(NN(2:n),Ea(2:n),'-O','LineWidth',2,'Color',[0 1 0]);
title('Ea vs No. of Gauss Points')

subplot(1,3,2); plot(NN(2:n),Eaabs(2:n),'-O','LineWidth',2,'Color',[0 1 0]);
title('Abs ea vs No. of Gauss Points')

subplot(1,3,3); plot(NN(2:n),SD(2:n),'-O','LineWidth',2,'Color',[1 0.5 0.5]);
title('Significant Digits Correct vs No. of Gauss Points')
```