## Contents



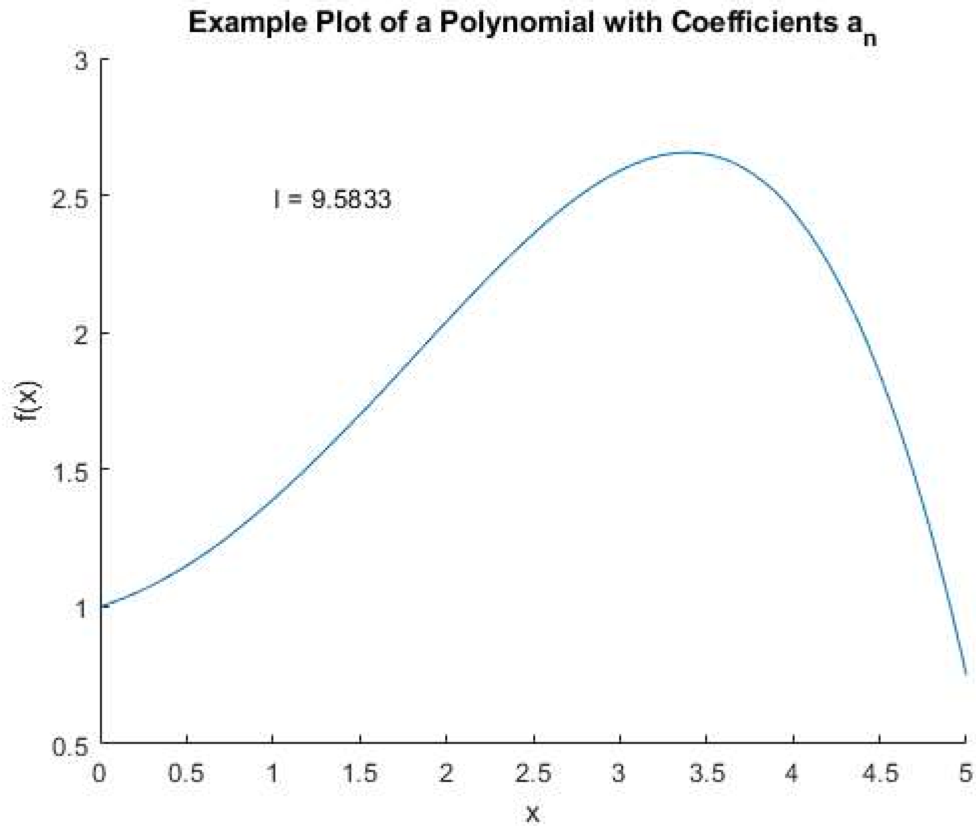
```
close all
clear
clc
```

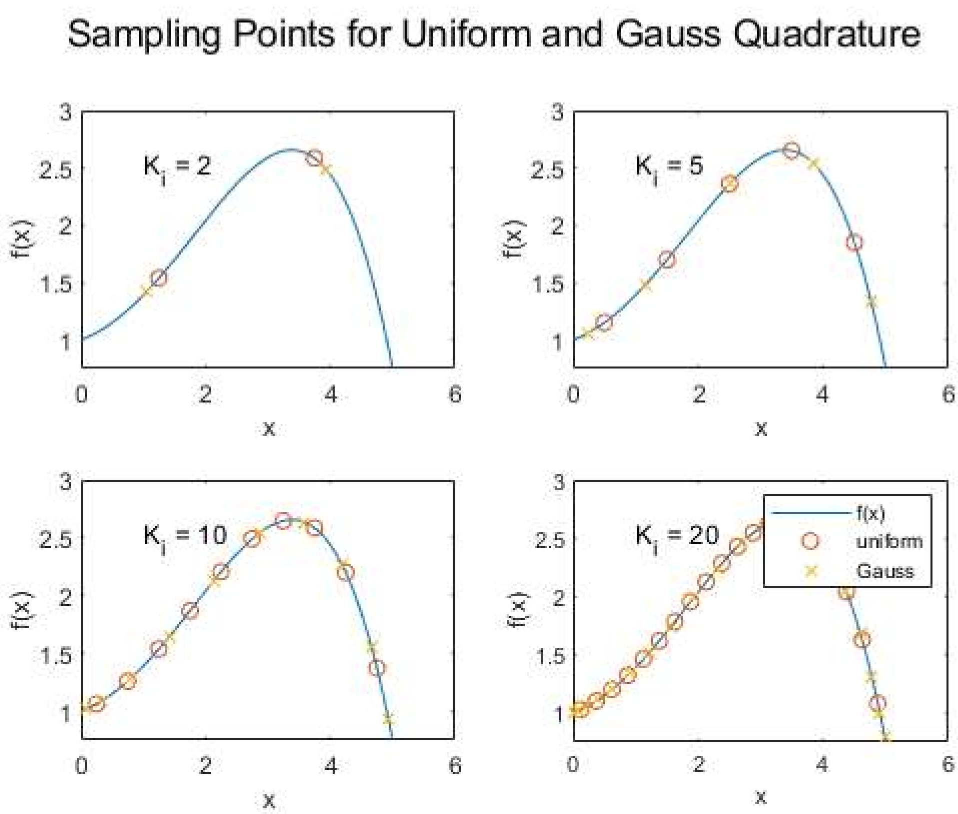
## Numerical Integration of One-Dimensional Integrals

```
%user defined: coefficients, vars array, and linearly spaced vector
%xa and xb bounds from 5 to 0
an   = [1 0.2 0.2 0 -0.01]; %vector coefficients
x    = linspace(0 , 5 , 100);
I    = 0;
fxk  = I;
f    = fxk;

%create function to integrate and verification of analytic results
for n = 1 : length(an)
    I = I + (an(n)/n)*5^n;
    f = f +  an(n)*x.^(n-1);
end
io = I; %approx I_o = 9.583

%first figure of Example plot of a polynomial with coefficients a_n
figure
hold on
plot(x , f)     %plot the f(x) over range from xa to xb with k points
title('Example Plot of a Polynomial with Coefficients a_n')
text(1 , 2.5 , 'I = 9.5833')
xlabel('x')
ylabel('f(x)')
hold off

%Uniform Sampling
p  = 0;
for k1 = [2, 5, 10, 20]

    dx = 5/k1;      %=(x2 - x1)/k1
    [xw , i] = lgwt(k1 , 0 , 5);   %call the lgwt function for weights and sums

    for n = 1 : k1
        xk(n) = (dx*n) - (dx/2);
    end
    %re-set zeroes vars
    fxk = 0;
    fxw = 0;

    for i = 1 : length(an)
        fxk = fxk + an(i)*xk.^(i-1);      %create the approximation by x with coefficients
        fxw = fxw + an(i)*xw.^(i-1);      %create the approximation by w with coefficients
    end

    % Sampling Points for Uniform and Gauss Quadrature
```

```
    figure(2)
    p  = p + 1;     %increment p for subplot
    subplot(2 , 2 , p)
    plot(x , f)             %plot the f(x)
    hold on
    scatter(xk,fxk,'o')        %scatter plot for uniform sampling
    scatter(xw,fxw,'x')
    xlim([0,6])
    %scatter plot for gaussian quad
    xlabel('x')
    ylabel('f(x)')
    txt = ["K_i = 2","K_i = 5","K_i = 10","K_i = 20"];  %place text in each graph
    text(1,2.5,txt(p))          %text location
end
L = legend('f(x)','uniform','Gauss')
suptitle('Sampling Points for Uniform and Gauss Quadrature')
hold off
```

```
L =

  Legend (f(x), uniform, Gauss) with properties:

         String: {'f(x)'  'uniform'  'Gauss'}
       Location: 'northeast'
    Orientation: 'vertical'
       FontSize: 8.1000
       Position: [0.7365 0.3206 0.1554 0.1119]
          Units: 'normalized'

  Use GET to show all properties
```

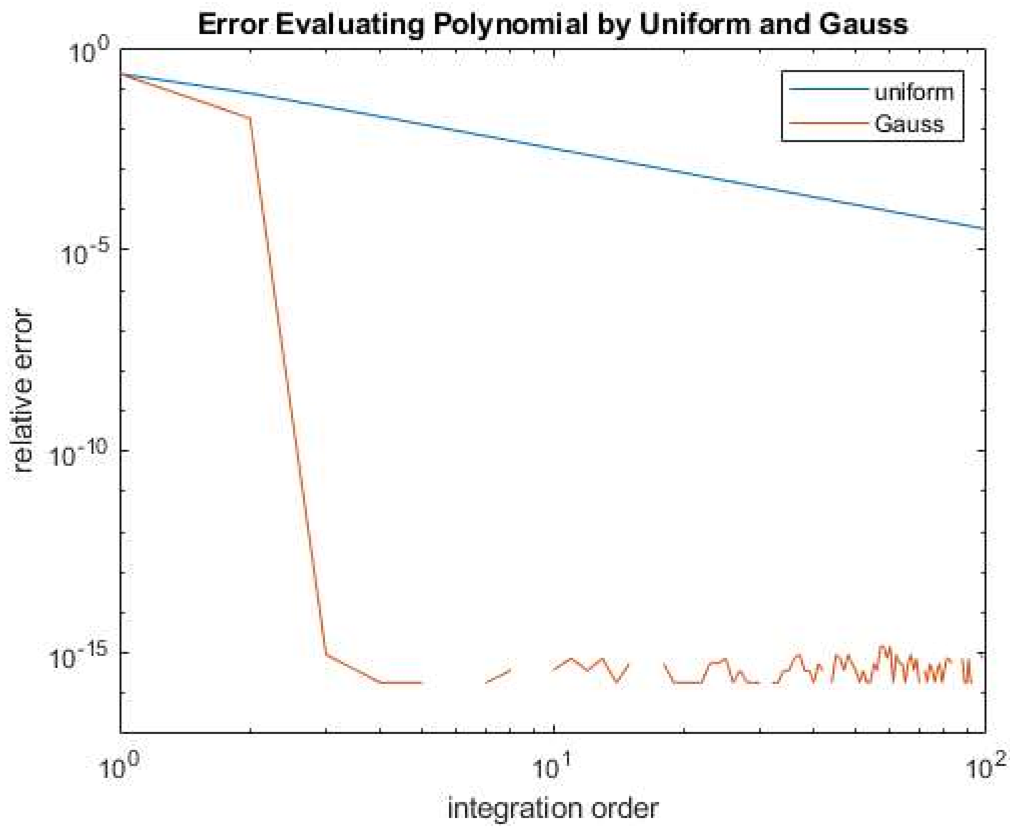



## Uniform Sampling

```
an   = [1 0.2 0.2 0 -0.01];
x  = linspace(0 , 5 , 100);
o = 1 : 100;
```

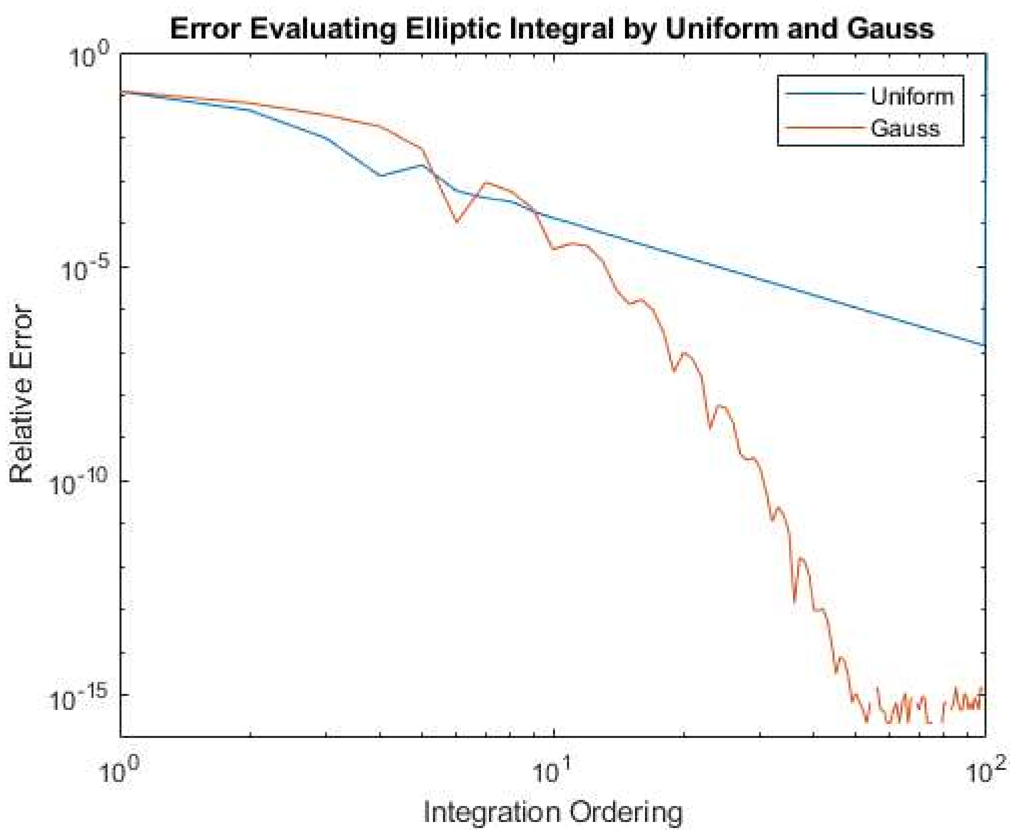
```
I = 0;
xk = 0;
dx = 0;
% fxk = 0;

for y = 1 : 100              % 1st nested for loop
    dx(y) = (5)/(o(y));
    itj = 0;
    for n = 1 : o(y)          % 2nd nested for loop
        xk(n) = 0.5*dx(y) + dx(y)*(n-1);
        fxn = 0;        %reset fxn
        for g = 1 : length(an)             % 3rd nested for loop
            fxn = fxn +(an(g)*xk(n).^(g-1));
        end
        itj = itj + fxn*dx(y);
    end
    I(y) = itj;
    polynInt(y) = (I(y) - io)./(io); %calculate the error

end
figure
loglog(o , polynInt)        %plot
hold on

%Gaussian Quadrature
for u = 1 : length(o)       %1st nested loop
    fxn   = 0;
    sumIntg = 0;
    [xi , wi] = lgwt(o(u), 0 , 5); %calling our lgwt function
    for a = 1 : length(xi)      %2nd for loop that iterates to create polynomials of degree
        fxn = 0;        %reset var
        for y = 1 : length(an)        %3rd nested loop to add up the approximation
            fxn = fxn + (an(y)*xi(a).^(y-1));
        end
        sumIntg = sumIntg + fxn*wi(a);
    end
    sumIntg(u) = sumIntg;
    error(u)  = abs( (sumIntg(u) - io)/ io);
end

loglog (o , error)      % plot
xlim([1 , 100])
ylim ([10e-18 , 1])
title('Error Evaluating Polynomial by Uniform and Gauss')
xlabel('integration order')
ylabel('relative error')
legend('uniform','Gauss')
hold off
```




### Convergence Analysis

```
kj = 0 : 100;
o  = kj;
k  = 0.8;
sumIq = 0*x;

for y = 1 : 101
    iq = 0;
    dX(y) = 5./kj(y);
    for n = 1 : (kj(y))
        xk(n) = 0.5.*dX(y) + dX(y).*(n - 1);
        E = sqrt(1 - 0.64.*(sin(xk(n)).^2));
        iq = iq + E*dX(y);
    end
    sumIq(y) = iq;
end
Iz = [sumIq 0];
for y = 1 : 101
    elp(y) = abs((Iz(y + 1) - Iz(y)) / Iz(y));
end
total = elp(1 : length(elp) );
op    = o( 1 : length(o) );

figure
loglog(op , total)
hold on


for j = 1 : length(kj)
    Iy = 0;
    [xi,wi] = lgwt(kj(j) , 0 , 5);
    for n = 1 : length(xi)
```

```
        E = sqrt(1 - k.^ 2.*(sin(xi(n)).^2));
        Iy = Iy + E*wi(n);
    end
    sumIy(j) = Iy;
end
Ig2 = [sumIy 0];

for j = 1 : length(sumIy)
    elp(j) = abs((Ig2(j+1) - Ig2(j))/Ig2(j));
end

elp = elp(1 : (length(elp)));

loglog(op,elp)
xlim([1 , 100])
ylim([10e-17 , 1])
title('Error Evaluating Elliptic Integral by Uniform and Gauss')
xlabel('Integration Ordering')
ylabel('Relative Error')
legend('Uniform','Gauss')
hold off
```




---

*Published with MATLAB® R2020a*