```
format long
f = @(x) sin(pi.*x)+(cos(pi.*x)).^2;
g = @(x) x.^(1/2);
F = @(x) [f(x); g(x)];
m0 = @(x) x.^0;
m1 = @(x) x.^1;
m2 = @(x) x.^2;
G = @(x) [m0(x); m1(x); m2(x)];

a0=0;
b0=1;
x=2:5;
ta1=maketable(a0,b0,x,@d1_quad_3,f,m0);
disp("Part A")
disp(ta1)
disp("This convergence rate is what I expected as the derivatives of f are "
+ ...
    "bounded so this quadrature should integrate exactly up to O(h^4)")
ta3=maketable(a0,b0,x,@d1_quad_5,g,m0);
disp(ta3)
disp("This convergence rate is what I expected as the derivatives of g are "
+ ...
    "unbounded at x=0 so it would not give as great of a convergence rate.")

N=32;

h=(b0-a0)/N;

%Integrate F(x)
l1=0;
for n = 0:N-1
    a=a0+n*h;
    b=a+h;
    l1=l1+d1_ip_fun([a b],@d1_quad_5,F,m0);
end

%Integrate f(x)
j1=0;
for n = 0:N-1
    a=a0+n*h;
    b=a+h;
    j1=j1+d1_ip_fun([a b],@d1_quad_5,f,m0);
end

%Integrate g(x)
j2=0;
for n = 0:N-1
    a=a0+n*h;
    b=a+h;
    j2=j2+d1_ip_fun([a b],@d1_quad_5,g,m0);
end
```

```
disp("Part B")
disp("Norm of integrals calculated together minus the integrals calculated
seperately.")
disp(norm(l1-[j1 j2]))

%Integrate F(x)*x
l2=0;
for n = 0:N-1
    a=a0+n*h;
    b=a+h;
    l2=l2+d1_ip_fun([a b],@d1_quad_5,F,m1);
end

%Integrate F(x)*x^2
l3=0;
for n = 0:N-1
    a=a0+n*h;
    b=a+h;
    l3=l3+d1_ip_fun([a b],@d1_quad_5,F,m2);
end

%Integrate f_i(x)g_j(x)
l4=0;
for n = 0:N-1
    a=a0+n*h;
    b=a+h;
    l4=l4+d1_ip_fun([a b],@d1_quad_5,F,G);
end

disp("Part C")
disp("Norm of integrals calculated together minus the integrals calculated
seperately.")
disp(norm(l4-[l1;l2;l3]))

function t=maketable(a0,b0,x,quadrule,func1,func2)
    N=2.^x;
    l=zeros(size(x));
    alph=zeros(size(x));
    h=zeros(size(x));

    for i = 1:size(x(:))
        h(i)=(b0-a0)./N(i);
        for n = 0:N(i)-1
            a=a0+n*h(i);
            b=a+h(i);
            l(i)=l(i)+d1_ip_fun([a b],quadrule,func1,func2);
        end
    end

    funcint=integral(func1,a0,b0);
    e=(funcint-l);

    for i=1:3
        alph(i+1)=log(e(i)/e(i+1))/log(h(i)/h(i+1));
    end

    t = table(N',e',h',alph');
    t.Properties.VariableNames=["N","Error","h","alpha"];
end

%I only included these scripts in the code to turn in as I could not figure
%out how to show the updates that I made to the scripts with the publish
%command in MatLab with seperate scipts being called in the folder.

function [val] = d1_ip_fun(ab, quadrule, func1, func2)
%
% This function approximates the integral of the function func
% over the interval (ab(1), ab(2)) using the quadrature rule quadrule.
%

    % Description of subinterval.
    xleft = ab(1)  ;
    xright = ab(2)  ;
    hsub = xright - xleft ;

    % Evaluation of quadrature points and quadrature weights.
    [quad_pts, quad_wghts] = feval(quadrule) ;
    nqpts = size(quad_pts,1) ;

    % Adjust points and weights to account for size of the interval.
    quad_pts = xleft + hsub* quad_pts ;
    quad_wghts = hsub * quad_wghts ;

    % Evaluate the function at the quadrature points.
    fun_vals1 = feval(func1, quad_pts) ;
    fun_vals1 = reshape(fun_vals1, [nqpts size(fun_vals1,1)/nqpts])';

    fun_vals2 = feval(func2, quad_pts) ;
    fun_vals2 = reshape(fun_vals2, [nqpts size(fun_vals2,1)/nqpts])';

    % Now to approximate the integral
    for i=1:size(quad_wghts,1)
        fun_vals2(:,i) = fun_vals2(:,i)*quad_wghts(i) ;
    end

    val = fun_vals2*fun_vals1';
end

function [quad_pts, quad_wghts] = d1_quad_3
%
% This function contains weights and quadrature points
% which are exact for polynomials of degree three for evaluting
% the line integral from (0,0) to (1,0).
%

    quad_pts = zeros(2,1) ;
    quad_wghts = zeros(2,1) ;
    quad_pts(1) = (1 - 1/sqrt(3))/2 ;
    quad_pts(2) = (1 + 1/sqrt(3))/2 ;

    quad_wghts(1:2) = 0.5* ones(1,2) ;
end

function [quad_pts, quad_wghts] = d1_quad_5
%
% This function contains weights and quadrature points
% which are exact for polynomials of degree five for evaluting
% the line integral from (0,0) to (1,0).
%

    quad_pts = zeros(3,1) ;
    quad_wghts = zeros(3,1) ;

    quad_pts(1) = (1 - sqrt(3/5))/2 ;
    quad_pts(2) = 0.5 ;
    quad_pts(3) = (1 + sqrt(3/5))/2 ;


    quad_wghts(1) = 5/18 ;
    quad_wghts(2) = 4/9 ;
    quad_wghts(3) = 5/18 ;
end
```

```
Part A
    N            Error                 h            alpha

    __     ____________________     _______     ________________

     4     5.71896105032721e-05       0.25                     0
     8     3.52180723028894e-06      0.125     4.02136523086816
    16     2.19304574944346e-07     0.0625     4.00530813023834
    32     1.36939535266833e-08    0.03125     4.00132498202058

This convergence rate is what I expected as the derivatives of f are bounded
so this quadrature should integrate exactly up to O(h^4)
    N             Error                 h            alpha

    __     _____________________     _______     ________________

     4     -0.000314239701989294       0.25                     0
     8     -0.000111100639606954      0.125     1.49999834422555
    16     -3.92800099003887e-05     0.0625     1.49999992396217
    32     -1.38875807155214e-05    0.03125     1.49999999660283

This convergence rate is what I expected as the derivatives of g are
unbounded at x=0 so it would not give as great of a convergence rate.
Part B
Norm of integrals calculated together minus the integrals calculated
seperately.
     0

Part C
Norm of integrals calculated together minus the integrals calculated
```

```
seperately.
      0
```

Published with MATLAB® R2023b