Text, letter

Description automatically generated

We need to prove thatis continuous on  so that the integral is calculable, and also this integral needs to converge so that its value is finite. C([0,1]) is the set of all continuous functions on the closed interval x in [0,1]

- First, we can see that both f(x), g(x) and 1/sqrt(x) are continuous in (0, 1]. At x = 0, 1/sqrt(x) is undefined, so this is an improper integral => f(x)g(x)/sqrt(x) is continuous on (0, 1] and thus this improper integral is defined.   
- Second, we need to prove that . Let (Since f(x) and g(x) is continuous) =>  (Trapezoid area)

(Trapezoidal area of equal bases)

M is a finite number => is well defined

Text, letter

Description automatically generated

We need to prove the following properties:

1) 

(proven)

2) 





3) 



4) 

We have from (2) andfrom (3)

=> (proven)

5) 



Text, letter

Description automatically generated

We will use the Gram’s Schmidt method to construct the 2nd order orthonormal basis by the Kronecker Delta, where the basis is :

and we have six equations to solve it.

1) => 

2) 



3) 



From (2) and (3) => 

4) 



5) 



6) 



From (4),(5) and (6) => 

=> The second order orthonormal basis is:

(answer). The roots are: 

=> The normalized polynomials:



Text, letter

Description automatically generated

Let ,….,be the roots of an orthonormal polynomial of degree n. Then:





In this case, we have . We have:



Now andbe the roots of an orthonormal polynomial of degree 2. We have:



=> (answer)

The two-point Gauss quadrature is:

Observation: we see that 

Text, letter

Description automatically generated

First, means that f to the 4th derivative is continuous on . We need to prove the error bound . The quadrature error is given by:



We haveL 2n = 4 => n = 2. Thus the quadrature error in this case is:

, where

and (proven)

The Matlab code for errorBound.m is

clc;

format long

% The two roots of the polynomial p2

root0 = (15 + 2\*sqrt(30))/35;

root1 = (15 - 2\*sqrt(30))/35;

l1w = @(x) (1./sqrt(x)).\*(x - root0)/(root1 - root0);

% A1

alpha1=quadl(l1w,0,1);

disp("The weighted term A1 is: " + alpha1);

l2w = @(x) (1./sqrt(x)).\*(x - root1)/(root0 - root1);

alpha2=quadl(l2w,0,1);

disp("The weighted term A2 is: " + alpha2);

pi2w = @(x)(1./sqrt(x)).\*((x.^2)-(6/7).\*x + 3/35).\*((x.^2)-(6/7).\*x + 3/35);

% Estimate for c2

c2 = quadl(pi2w,0,1)/24;

disp("The estimate for c2 is: " + c2);

Text, letter

Description automatically generated

The estimate of computed by Matlab is therefore around 4.837711261601187e-04

Graphical user interface, text

Description automatically generated with medium confidence

Text, letter

Description automatically generated

First, we have =>So we need to show that the solid ellipsoid is contained inside this stretched box C.







Similarly, we can prove that The stretched box of the cube has the dimension of  and we have proof from above that

=> All points in the solid ellipsoid are contained in the hypercube C (proven)

Text, letter

Description automatically generated

First, we can write the volume of K is as:

Considering the ratio of the volume of unit ball over volume of cube with sides 2

  
Let N be the number of uniformly distributed points generated inside the cube of side length 2 and be the number of points of those generated points that lie inside the unit sphere. By the central limit theorem, as the sample size N approaches to infinity, the ratio starts to approach the ratio =>. Therefore, the volume of the ellipsoid can be approximated by the formula .  
This formula can also be correct even if the points are not uniformly distributed.

Text, letter

Description automatically generated

The Matlab code for MonteCarlo.m is

clc;

format long

a = 1; b = 2; c = 3;

volC = 8\*a\*b\*c;

N = input("Enter the number of uniformly distributed points: ");

X = rand(N,3).\*2 - 1 ;

[numPoints, vectorSize] = size(X);

countInsideSphere = 0;

lie\_inside\_sphere = @(u, v, w) u^2 + v ^2 + w^2 <= 1;

for i=1:numPoints

point = X(i,:);

if lie\_inside\_sphere(point(1),point(2),point(3))

countInsideSphere = countInsideSphere + 1;

end

end

disp("Number of uniformly generated points inside the cube of side 2: " + numPoints);

disp("Number of points inside the unit sphere: " + countInsideSphere);

ratio = countInsideSphere/numPoints;

disp("The ratio of '(4/3pi)/8'(0.523599) approximated by the Monte Carlo method: " + ratio);

volKtrue = volC \* (4/3 \* pi)/8;

disp("True volume of ellipsoid K: " + volKtrue)

volKapprox = volC \* ratio;

disp("Approximated volume of ellipsoid K by Monte Carlo method: " + volKapprox)

piApprox = ratio\*8/(4/3);

disp("Approximation of pi: " + piApprox)

Inputting 100000 uniformaly distributed points, we get the results as follows:

Text

Description automatically generated