1. Solution:

Julia code:

julia> using LinearAlgebra

julia> function gauss\_quad(n)

b=@. .5/sqrt(1-(2\*(1:n-1))^(-2.))

T=SymTridiagonal(zeros(n), b)

D, V = eigen(T)

i = sortperm(D); x = D[i]

w = 2\*view(V, 1, i).^2

x,w

end

gauss\_quad (generic function with 1 method)

julia> f(z)=(z+3)^-1

f (generic function with 1 method)

julia> x,w=gauss\_quad(2)

([-0.5773502691896258, 0.5773502691896258], [0.9999999999999998, 0.9999999999999998])

julia> w'f.(x)

0.6923076923076921

julia> x,w=gauss\_quad(3)

([-0.7745966692414824, 8.881784197001252e-16, 0.7745966692414834], [0.555555555555557, 0.8888888888888867, 0.5555555555555558])

julia> w'f.(x)

0.6931216931216929

julia> x,w=gauss\_quad(4)

([-0.8611363115940502, -0.3399810435848556, 0.33998104358485715, 0.8611363115940526], [0.3478548451374555, 0.652145154862545, 0.652145154862546, 0.3478548451374536])

julia> w'f.(x)

0.6931464174454828

the exact value for the integral:0.693147180559945

So:

N=2: error=0.0008394882522528846

N=3: error=2.5487438252036476e-5

N=4: error=7.63114462176695e-7

1. Solution:

We already know that

x1=-0.5773502691896258

x2=0.5773502691896258

a1=1

a2=1

Matlab code:

>> syms w x

>> s=(pi/8)\*cos((pi/8)\*w\*(x+1))

s =

(pi\*cos((pi\*w\*(x + 1))/8))/8

>> error=(int(s,x,-1,1)-a\*subs(s,x,z)')

error =

sin((pi\*w)/4)/w - (pi\*cos((pi\*conj(w)\*(3^(1/2)/3 + 1))/8))/8 - (pi\*cos((pi\*conj(w)\*(3^(1/2)/3 - 1))/8))/8

If w=0.1:

>> double(subs(error,w,0.1))

ans =

6.912006352198659e-09

If w=1:

>> double(subs(error,w,1))

ans =

6.352168618674107e-05

If w=5:

>> double(subs(error,w,5))

ans =

-0.014171081338654

When w is large,the error would be larger.

3.Solution:

Assume a=-1,b=1:

Matlab code:

roo.m:

function F=roo(p)

syms x s z

s=[exp(-x) x\*exp(-x) x^2\*exp(-x) x^3\*exp(-x)]

z=[x^0 x^1 x^2 x^3]

F(1)=double(subs(int(subs(z(1),x,p(1))\*p(2)+subs(z(1),x,p(3))\*p(4)-int(s(1),x,-1,1),x,-1,1)))

F(2)=double(subs(int(subs(z(2),x,p(1))\*p(2)+subs(z(2),x,p(3))\*p(4)-int(s(2),x,-1,1),x,-1,1)))

F(3)=double(subs(int(subs(z(3),x,p(1))\*p(2)+subs(z(3),x,p(3))\*p(4)-int(s(3),x,-1,1),x,-1,1)))

F(4)=double(subs(int(subs(z(4),x,p(1))\*p(2)+subs(z(4),x,p(3))\*p(4)-int(s(4),x,-1,1),x,-1,1)))

p=[0.1,1,0.1,1];

fun=@roo;

x=fsolve(fun,p)

x =

-0.689267954589970 1.553491399887313 0.420389431978486 0.796910987400290

x1=-0.689267954589970 x2=0.420389431978486

α1=1.553491399887313 α2=0.796910987400290

Test:

julia> x=[-0.689267954589970,0.420389431978486]

2-element Array{Float64,1}:

-0.68926795458997

0.420389431978486

julia> w=[1.553491399887313,0.796910987400290]

2-element Array{Float64,1}:

1.553491399887313

0.79691098740029

julia> f(z)=z^2

f (generic function with 1 method)

julia> w'f.(x)

0.8788846226019027

When f(x)=, the real result is 0.878884622601834

Error=-6.872280522429719e-14

So the results are correct.