Laguerre-Gauss Quadrature
Lab Report for Assignment No. 4

SHASHVAT JAIN
(2020PHY1114)

HARSH SAXENA
(2020PHY1162)

S.G.T.B. Khalsa College, University of Delhi, Delhi-110007, India.

March 10, 2022

Submitted to Dr. Mamta
"32221401 - MATHEMATICAL PHYSICS III"

Contents

1 Theory

The theoretical aspect of Gaussian Quadrature is not limited to the use of the roots of Legendre polynomials, instead it is a generalized method which is useful in calculating larger variety of integrals.

There is a underlying theory which connects all the Gaussian quadrature rules.

While performing Gaussian Quadrature, the integral approximated for an integrand obtained by multiplying a weighting function($w(x)$) to some other function $f(x)$.

$$\int_a^b w(x)f(x)dx \tag{1}$$

We use weighting functions to ensure the convergence of inner products of the family of orthogonal polynomial functions in their respective intervals, when dealing with Legendre polynomials $W(x) = 1$ because the family of polynomials is orthogonal in the interval $[-1, 1]$ and the inner-product converges for all pairs of Legendre polynomials in this interval. Associated with each Weighting function and integration interval pair there is a special family of polynomials.

1.1 Laguerre-Gauss Quadrature

One such family of Polynomials is Laguerre Polynomials which are known to be the solution of laguerre's equation which is a second order linear differential equation.

$$xy" + (1 - x)y' + ny = 0 \qquad \text{where n} \in \mathbb{Z}_+ \tag{2}$$

The weighting function and integration interval of these polynomials is -

$$w(x) = e^{-x} \qquad \text{on} \quad [0, \infty) \tag{3}$$

It is evident by seeing the limit of integration that this gaussian quadrature rule is used in calculating the improper of integral of type.

$$\int_0^\infty f(x)dx \tag{4}$$

Specifically when we include the weighting function e^{-x} then.

$$\int_0^\infty f(x)e^{-x}dx = \sum_1^n w_i f(x_i) \tag{5}$$

where n is the n point quadrature used.We can write

$$\int_0^\infty f(x)dx = \int_0^\infty f(x)e^x e^{-x}dx = \int_0^\infty g(x)e^{-x}dx \tag{6}$$

$$g(x) := f(x)e^x \tag{7}$$

1.2 Laguerre polynomials

The Solution of above differential equation results in the following recurrence relation.

$$L_{n+1}(x) = (1 - x + 2n)L_n - n^2 L_{n-1}(x) \tag{8}$$

If we know the polynomials $L_0 and L_1$ we can determine all the other polynomials

$$L_n(x) = \sum_{k=0}^{n} \frac{(-1)^{n-k}(n!)^2}{(n-k)!^2 k!} x^{n-r} \tag{9}$$

From these relations we can derive first five Laguerre polynomials

$$\begin{aligned}
L_0 &= 1 \qquad\qquad L_1 = -x + 1 \\
L_2 &= \frac{1}{2}(x^2 - 4x + 2) \\
L_3 &= \frac{1}{6}(-x^3 + 9x^2 - 18x + 6 \\
L_4 &= \frac{1}{24}(x^4 - 16x^3 + 72x^2 - 96x + 24 \\
L_5 &= \frac{1}{120}(-x^5 + 25x^4 - 200x^3 + 600x^2 - 600x + 120
\end{aligned}$$

Orthogonality relation

$$\int_0^\infty L_m(x) L_n(x) e^{-x} dx = (n!)^2 \delta_{mn} \tag{10}$$

Where δ_{mn} is the kronecker delta function

1.3 Construction of 2-point Laguerre gauss quadrature

Since this formula is to have degree of precision equal to 3,Then the weights and abcissas must satisfy-that for constant,linear,quadratic and cubic functions the integral must be exatly equal to the weights multiplied by the function evaluation at these decided abcissa.

1. $f(x) = 1$

$$\begin{gathered}
\int_0^\infty 1 e^{-x} dx = w_1 f(x_1) + w_2 f(x_2) \\
w_1 + w_2 = [-e^{-x}]_0^\infty
\end{gathered}$$

$$w_1 + w_2 = 1 \tag{11}$$

2. $f(x) = x$

$$\begin{gathered}
\int_0^\infty x e^{-x} dx = w_1 f(x_1) + w_2 f(x_2) \\
w_1 x_1 + w_2 x_2 = [-(x+1)e^{-x}]_0^\infty
\end{gathered}$$

$$w_1 x_1 + w_2 x_2 = 1 \tag{12}$$

3. $f(x) = x^2$

$$\int_0^\infty x^2 e^{-x} dx = w_1 f(x_1) + w_2 f(x_2)$$
$$w_1 x_1^2 + w_2 x_2^2 = [-(x^2 + 2x + 2)e^{-x}]_0^\infty$$
$$w_1 x_1^2 + w_2 x_2^2 = 2 \tag{13}$$

4. $f(x) = x^3$

$$\int_0^\infty x^3 e^{-x} dx = w_1 f(x_1) + w_2 f(x_2)$$
$$w_1 x_1^3 + w_2 x_2^3 = [-(x^3 + 3x^2 + 6x + 6)e^{-x}]_0^\infty$$
$$w_1 x_1^3 + w_2 x_2^3 = 6 \tag{14}$$

After solving the following equations we get the following weights-

x_i	w_i
$2 - \sqrt{2}$	$\frac{1}{4}(2 + \sqrt{2})$
$2 + \sqrt{2}$	$\frac{1}{4}(2 - \sqrt{2})$

Table 1: Caption

2 Programming

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar  6 15:40:50 2022

@author: harsh
"""

from Myintegration import *
import matplotlib.pyplot as plt
from scipy import stats
from sympy import Symbol


# define function
def MyLeguQuad(func,n):
    x,w = roots_laguerre(n)
    return np.dot(w,func(x)*np.exp(x))

MyLeguQuad = np.vectorize(MyLeguQuad)

#
# Validation
#
'''
listp = ['one','two','three']
```

```
func = []
P = []
K = []
for i in range(len(listp)):
    k = input(f"function {listp[i]}: ")
    K.append(k)
    func.append(lambda x,i = i: eval(K[i],{'x':x,'np':np}))

print(func[0](0))

for j in range(len(func)):
    p1 = lambda x,j=j : func[j](x)* np.exp(-x)
    P.append(p1)

mat = np.zeros((2,4))
n_r = [2,4]
mat[:,0] = n_r
for i in range(1,4):
   mat[:,i] = MyLeguQuad(P[i-1],n_r)

np.savetxt("validate-lag-1162.out",mat,fmt="%.7g",delimiter=",",header="n,$I_1$,$I_2$
    ")
print(mat)
'''
######

I1 = lambda x : np.exp(-x)/(1+x**2)
I2 = lambda x : 1/(1+x**2)


dat = np.zeros((7,3))
dat[:,0] = 2**np.arange(1,8)
n_arr = dat[:,0]
dat[:,1] = MyLeguQuad(I1,n_arr)
dat[:,2] = MyLeguQuad(I2,n_arr)
print(dat)

np.savetxt("quad-lag-1162.out",dat,fmt="%.7g",delimiter=",",header="n,$I_1$,$I_2$")


#simpson
def laguerre_tol(func,n_max,rtol):
    n_arr = np.arange(5,n_max,5)
    I = np.zeros(n_arr.shape)
    r_err = np.zeros(len(n_arr)-1)

    for i in range(len(n_arr)):
        I[i] =  MyLeguQuad(func, n_arr[i])
        if i==0:
           continue
        r_err[i-1] = abs((I[i] - I[i-1])/(I[i]))
        if r_err[i-1] <= rtol:
           return r_err[:i],I[:i],n_arr[:i]


def simpsonLagu(f,max_b= int(2e16),rtol = 0.5*10**(-4)):
    b = 2**np.arange(1,np.floor(np.log2(max_b)))
    print(b)
    I = np.zeros(b.shape)
    r_err = np.zeros(len(b)-1)
    for i,bi in enumerate(b):
        I[i] = MySimp(f,0,bi,m = int(1e7),d =5)[0]
```

```
        if i == 0:
            continue
        r_err[i-1] = abs((I[i] - I[i-1])/(I[i]))

        if r_err[i-1]<=rtol:
            return  r_err[:i],I[:i],b[:i]
    return I,b




rl_l, inte_l,n_ar = laguerre_tol(I2,500,0.5*10**(-4))
rl_s, inte_s,b_ar = simpsonLagu(I2,int(2e16),0.5*10**(-4))

slope, intercept, r_value, p_value, std_err = stats.linregress(rl_s,inte_s)
Y = slope*rl_s + intercept

print('Slope of relative tol vs Integral line for function 1/(1+x^2)',slope)

#Plotting and comparison

fig,ax = plt.subplots()
ax.plot(rl_l,inte_l,'--o',label ='improper integral by Laguerre')
ax.plot(rl_s,inte_s,'* ',label = 'improper integral by simpson')
ax.plot(rl_s,Y,label = 'Regression line')
ax.set_xlabel('relative tolerence')
ax.set_ylabel('Integral')
ax.set_title('Comparsion between integraal calculated by simpson and laguerre')
ax.legend()

fig,(ax1,ax2) = plt.subplots(1,2)
ax1.plot(n_ar,inte_l,'--v')
ax1.set_xlabel('n point formula used')
ax1.set_ylabel('Integral')
ax1.set_title('Convergence of Laguerre integral with n upto to an accuracy of 4
    significant digits')
ax2.plot(b_ar,inte_s,'r--x')
ax2.set_xlabel('Upper limit on the integral')
ax2.set_ylabel('Integral')
ax2.set_title('Convergence of simpson integral with b(upper limit) upto to an
    accuracy of 4 significant digits')
```

3 Results and Analysis

From the data obtained, shown below, we notice that both the integrals I_1 and I_2 converge as the Gaussian Quadrature is calculated at greater number of absiccas.(n in the n-point formuala, which is also what we expected to happen as the Gaussian quadrature is able to calculate functions differing from the usual polynomial functions more accurately with an increase in n-points.

n	I_1	I_2
2	0.6470588	1.493257
4	0.636427	1.50119
8	0.620075	1.53376
16	0.6215065	1.553738
32	0.6214493	1.562483
64	0.6214496	1.566725
128	0.6214496	1.568789

Table 2: Data from quad-lag-1162.out

To compare the two methods, we tested how well do they approximate the integral with their individual tolerances, with a decrease in tolerance we expect the methods to converge to a value by performing more computation, that is more n-points for Lageurre and an higher upper bound for Simpson.

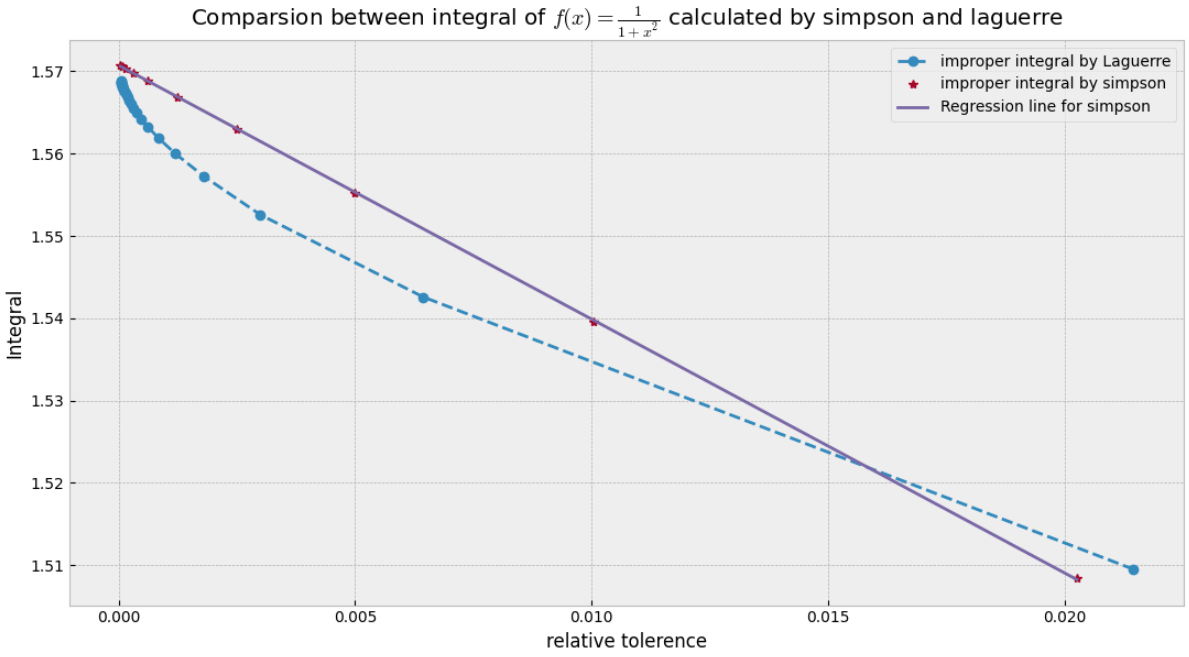

Figure 1: Graph of the integral values produced by Lageurre and simpson methods against the tolerance required to achieve those.

However we also expect this change in integral to die out which is explained by the second figure. The simpson method seems to converge much more sharply, but this is really not the case, do not forget that in simpson's case we are increasing the upper-bound thus introducing many more sub-intervals to compute the formula over whereas in the case of n-points, we are varying n-points not so much in comparison to Simpson.

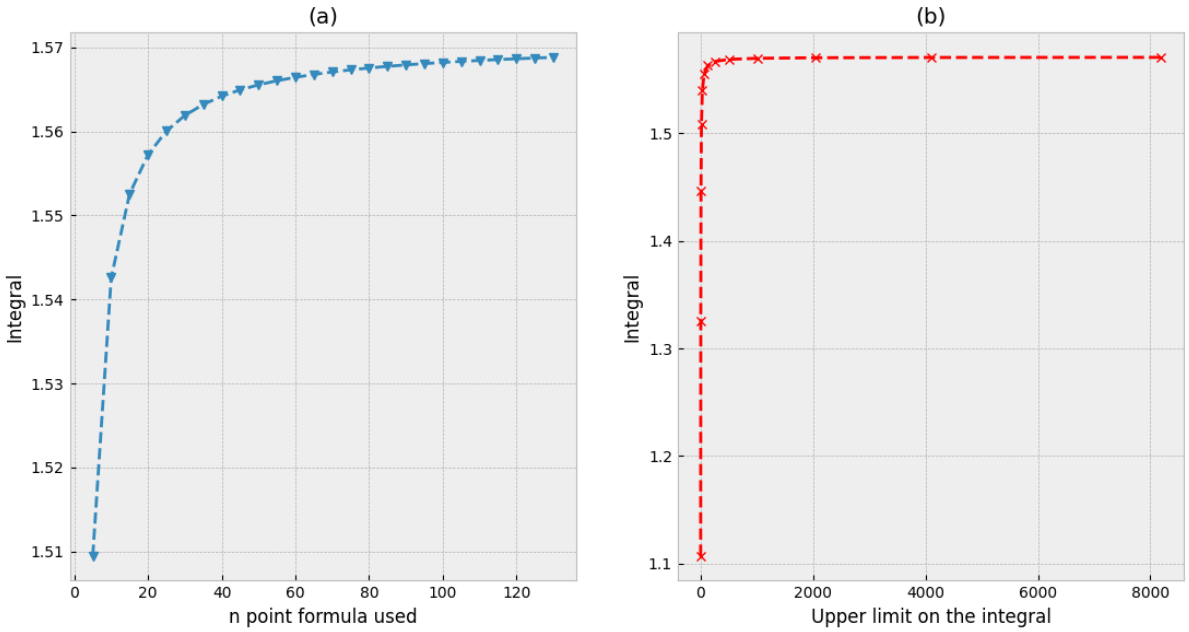

Figure 2: (a) Convergence of Laguerre integral with n upto to an accuracy of 4 significant digits (b) Convergence of Laguerre integral with upper limit(b) upto to an accuracy of 4 significant digits