Mathematical Physics - III (Practical)
Paper: PHS-A-CC-4-8-P
Dirac-delta Function

2. Dirac-delta Function

(i). Evaluate $\frac{1}{\sqrt{2\pi\sigma^2}}\ e^{-\frac{(x-2)^2}{2\sigma^2}}(x+3)dx$ for σ=1.0, 0.1, 0.01 and show that it tends to 5.

```
import numpy as np

import scipy.integrate as sci

def f(x,mu,sig):

    return np.exp(-(x-mu)**2/(2.0*sig**2))*(x+3)/np.sqrt(2.0*np.pi*sig**2)

mu = 2.0; sig = 0.01; low = -np.inf; up = np.inf;

low = mu - 10*sig; up = mu + 10*sig;

I, err = sci.quad(f,low,up,args=(mu,sig))

print ('Sigma= ', sig, 'Integral value = ', I, ' error = ', err)
```

Output:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:\Python38-32\semIV-CC8\sem4\improp-dd.py
=============
Sigma=  1.0 Integral value =  5.000000000000001  error =  9.124333862032749e-09
>>>
======= RESTART: C:\Python38-32\semIV-CC8\sem4\improp-dd.py
=============
Sigma=  0.1 Integral value =  5.000000000000003  error =  4.368046350715512e-09
>>>
======= RESTART: C:\Python38-32\semIV-CC8\sem4\improp-dd.py
=============
Sigma=  0.01 Integral value =  5.000000000000027  error =  4.335842943253669e-09
>>>
```

(ii). Numerically verifying the Gaussian Integral result

$$\int \exp(-ax^2 + bx + c) = \sqrt{\frac{\pi}{a}}\exp(\frac{b^2}{4a} + c)$$

```
import numpy as np
import scipy.integrate as sci
def f(x,a,b,c): return np.exp(-a*x**2 + b*x + c)
a = 1; b = 2; c = 1; low = -np.inf; up = np.inf;
I_numerical, error = sci.quad(f, low, up, args=(a,b,c))
I_theoretical     = np.sqrt(np.pi/a)*np.exp(b**2/(4.0*a)+c)
print ('Integral_',low,'^',up,' e^(-',a,'x^2+',b,'x+',c,') dx = ', I_numerical)
print ('Theoretical value of the Integral = ', I_theoretical)
print ('Absolute error = ', error, ', Relative error = ', I_numerical - I_theoretical)
```

Output:

```
====== RESTART: C:\Python38-32\semIV-CC8\sem4\gaussian-dd.py
============
Integral_ -inf ^ inf  e^(- 1 x^2+ 2 x+ 1 ) dx =  13.09676093710652
Theoretical value of the Integral =  13.09676093710652
Absolute error =  2.710567613556251e-10 , Relative error =  0.0
>>>
======= RESTART: C:\Python38-32\semIV-CC8\sem4\gaussian-dd.py
============
Integral_ -inf ^ inf  e^(- 2 x^2+ 1 x+ 5 ) dx =  210.7750292631574
Theoretical value of the Integral =  210.7750292631574
Absolute error =  1.85425096410708e-08 , Relative error =  0.0
>>>
```

(iii). Verifying that the convolution of two Gaussian function is a Gaussian

Gaussian Probability Density Function:

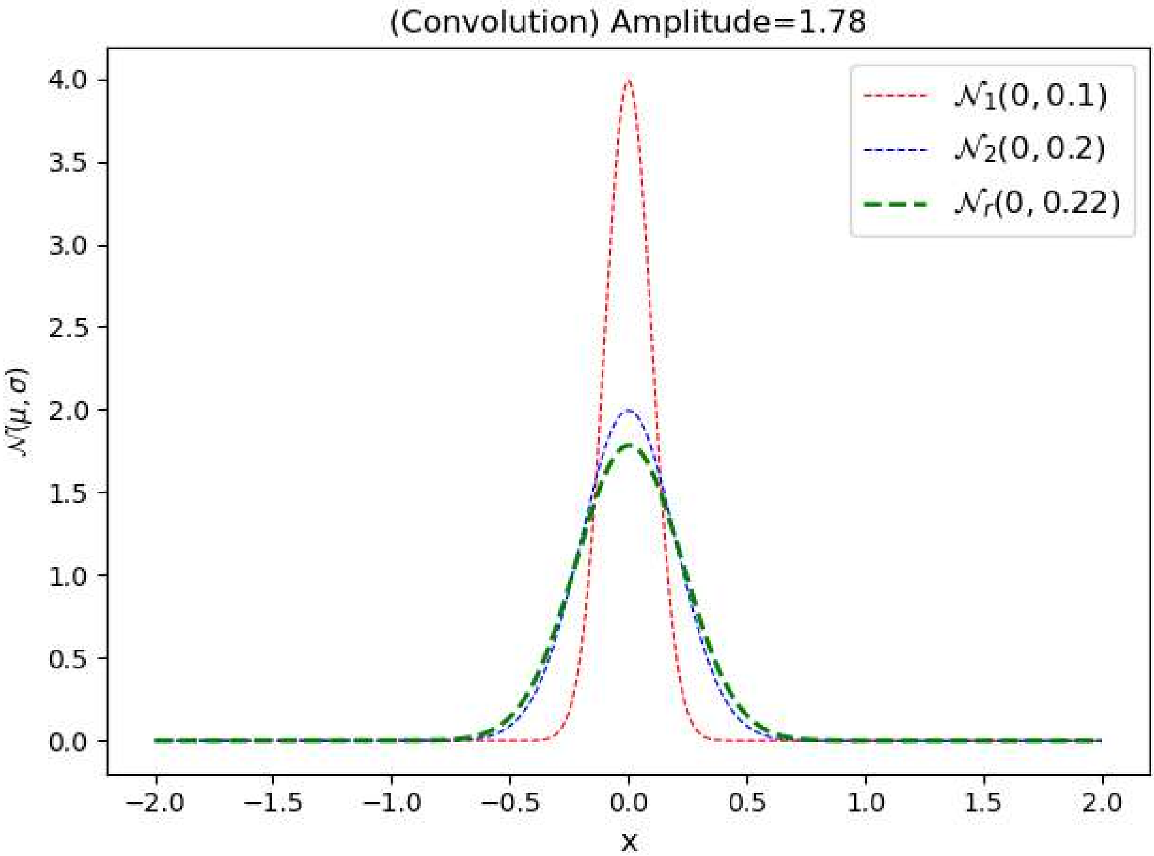
$$P(x) = \frac{1}{\sqrt{2\pi\sigma}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

```
import numpy as np
import scipy.integrate as sci
import matplotlib.pyplot as plt
from scipy.signal import gaussian

def gauss(x,mu,sig):
    return np.exp(-((x-mu)**2.0)/(2.0*sig**2.0))/np.sqrt(2.0*np.pi)/sig
mu1 = 0; sig1 = 0.1
mu2 = 0; sig2 = 0.2;
x = np.linspace(-2, 2, 500)
dx = x[1] - x[0]
convolution = np.convolve(gauss(x,mu1,sig1), gauss(x,mu2,sig2), mode="same")*dx
  #sigc = np.sqrt(sig1**2 * sig2**2/(sig1**2 + sig2**2)) # product std
  #ampc = sigc/(np.sqrt(2*np.pi)*sig1*sig2)              # product amplitude
sigc = np.sqrt(sig1**2 + sig2**2)             # convolution std
ampc = 1.0/np.sqrt(2*np.pi*(sig1**2 + sig2**2)) # convolution amplitude

plt.figure()
```

```
plt.plot(x, gauss(x,mu1,sig1), '--r', lw=.8,
        label=r"$\mathcal{N}_1("+str(mu1)+","+str(sig1)+")$")
plt.plot(x, gauss(x,mu2,sig2), '--b', lw=.8,
        label=r"$\mathcal{N}_2("+str(mu2)+","+str(sig2)+")$")
plt.plot(x, convolution, 'g--', lw=1.8,
        label=r"$\mathcal{N}_r("+str(mu1+mu2)+","+str(round(sigc,2))+")$")
plt.title("(Convolution) Amplitude="+str(round(ampc,2)))
plt.legend(loc='best', prop={'size':12})
plt.xlabel("x", size=12)
plt.xticks(size = 10)
plt.ylabel(r"$\mathcal{N}(\mu,\sigma)$", size=10)
plt.yticks(size = 10)
plt.tight_layout()
plt.show()
```



(iv). Verifying that $\int_{a-x1}^{a+x2} \delta(x-a)f(x)dx = f(a)$ using different limiting representation of $\delta(x)$.

```
import numpy as np

import scipy.integrate as sci
```

```
import matplotlib.pyplot as plt
from scipy.signal import gaussian

from sympy import *
def f(x) :
    #return sin(x)
    return x**2
   #return exp(-x**2+x+1)
x1 = 1.0; x2 = 1.5; a = 2.0;
x = Symbol('x')
I = integrate(f(x)*DiracDelta(x-a), (x, a-x1, a+x2))
rhs=f(a)
print ('f(a)=',rhs)
##def ddelta(x,a):
##    eps = 1.0;
##
##    return exp(-(x-a)**2/(4.0*eps))/(sqrt(4.0*pi*eps))
##x1 = 1.0; x2 = 1.5; a = 5.0;
print ('Integral_(',a,'-',x1,')^(',a,'+',x2,') f(x) ddelta(x-',a,')dx = ', I.evalf())

============== RESTART: C:\Python38-32\semIV-CC8\sem4\dd-prob4.py
==============
f(a)= 4.0
Integral_( 2.0 - 1.0 )^( 2.0 + 1.5 ) f(x) ddelta(x- 2.0 )dx =  4.00000000000000
>>>
```

-----------*****------------