Homework n.1

Evaluate numerically and plot graphically the convolution integral of the energy spectrum f(E) with a gaussian resolution g(E) defined below.

$$f(E)=a_1 f_1(E)+a_2 f_2(E)+a_3 f_3(E)$$

where:

$$f_1(E)=1/E \text{ for } 0.1<E<0.92\ MeV$$

$$f_1(E)=0 \text{ for } E<0.1\ or\ E>0.92\ MeV$$

$$f_2(E)=G(\mu=1.17\ MeV, \sigma=0.001\ MeV)$$

$$f_3(E)=G(\mu=1.33\ MeV, \sigma=0.001\ MeV)$$

$$a_1=1;\ a_2=0.9;\ a_3=0.8$$

Consider the following cases:

1. g(E)=G(E,s) with s/E = 5%/√E(MeV)
2. g(E)=G(E,s) with s/E = 10%/√E(MeV)
3. g(E)=G(E,s) with s/E = 30%/√E(MeV)
4. g(E)=G(E,s) with s/E = 1%/√E(MeV)

(optional) Invent yourself an f(E) distribution with sharp edges or peaks and repeat the previous exercise to point out the effect of different resolutions on f(E).

Solution

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from numpy import vectorize
```

We define the functions Gaussian and f(E) and vectorize them in order to pass a vector as an argument.

```
def Gaussian(x,mean, sigma):
    return (1/np.sqrt(2*np.pi*sigma**2))*np.exp(-(x-
mean)**2/(2*sigma**2))


def f1(x):
    if (x < 0.1 or x > 0.92) :
        return 0.
    else :
        return float(1/x)
vf1 = vectorize(f1)
```

```
def f(x):
    return 1*Gaussian(x,1.17,0.01) + 0.9*Gaussian(x,1.3,0.01) +
0.8*vf1(x)

vf = vectorize(f)
```

Then we define the function we want to integrate and a function to make the convolution.

```
def integrand(t, x, percent):
    return f(t)*Gaussian(t,x,percent*np.sqrt(x))



def convolve(step, x0, percent):
    dx = step
    t = np.arange(0,100,step)
    return np.sum(dx*integrand(t,x0,percent))

vconvolve = vectorize(convolve)
```

Then we need a vector for the coordinates on the x axis

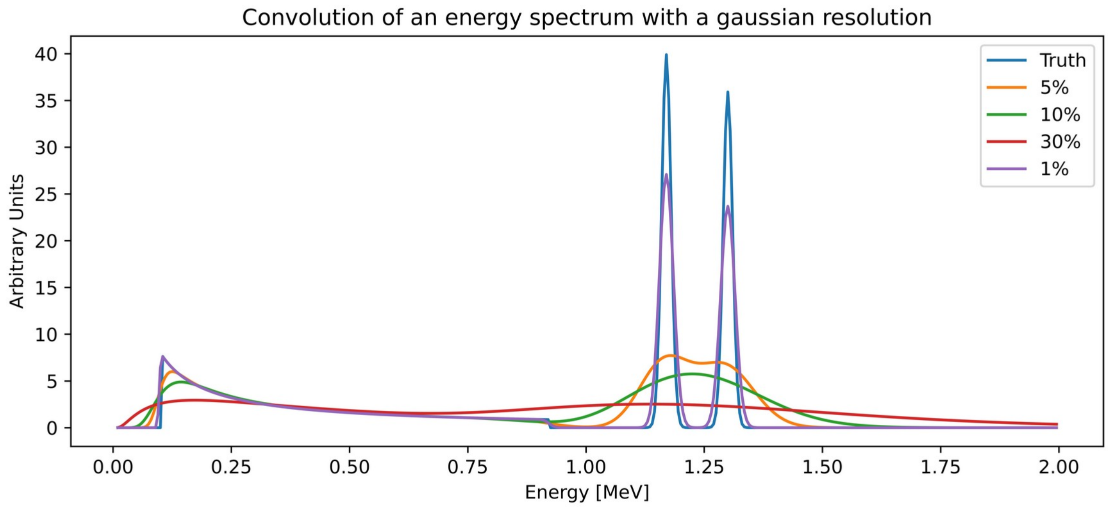
```
a = 0.01
b = 2
dx = 0.005
x = np.arange(a,b,dx)
```

And finally we convolve f(E) with the resolution for the requested cases. The variable step characterizes the width of the increment dE.

```
step = 0.005
y = vconvolve(step,x,0.05)
y1=vconvolve(step, x,0.1)
y2=vconvolve(step,x,0.3)
y3=vconvolve(step,x,0.01)

import matplotlib_inline
matplotlib_inline.backend_inline.set_matplotlib_formats('jpg')
plt.figure(figsize = (10,4), dpi = 1200)
plt.title('Convolution of an energy spectrum with a gaussian
resolution')
plt.plot(x, vf(x), label = 'Truth')
plt.plot(x, y, label = '5%')
plt.plot(x,y1, label = '10%')
plt.plot(x,y2, label = '30%')
plt.plot(x,y3, label = '1%')
plt.xlabel('Energy [MeV]')
plt.ylabel('Arbitrary Units')
```

```
plt.legend()
plt.show()
```



As a double check we assure that the integral under the curves is more or less the same. Obviously the areas will agree better as the integration step decreases.

```
from numpy import trapz
area = trapz(y, dx=dx)
print("area (5%) =", area)
```

```
area (5%) = 3.697701356218631
```

```
area = trapz(y1, dx=dx)
print("area (10%)  =", area)
```

```
area (10%)  = 3.697700848359407
```

```
area = trapz(y2, dx=dx)
print("area (30%) =", area)
```

```
area (30%) = 3.605465187734962
```

```
area = trapz(y3, dx=dx)
print("area (1%) =", area)
```

```
area (1%) = 3.697758914232729
```