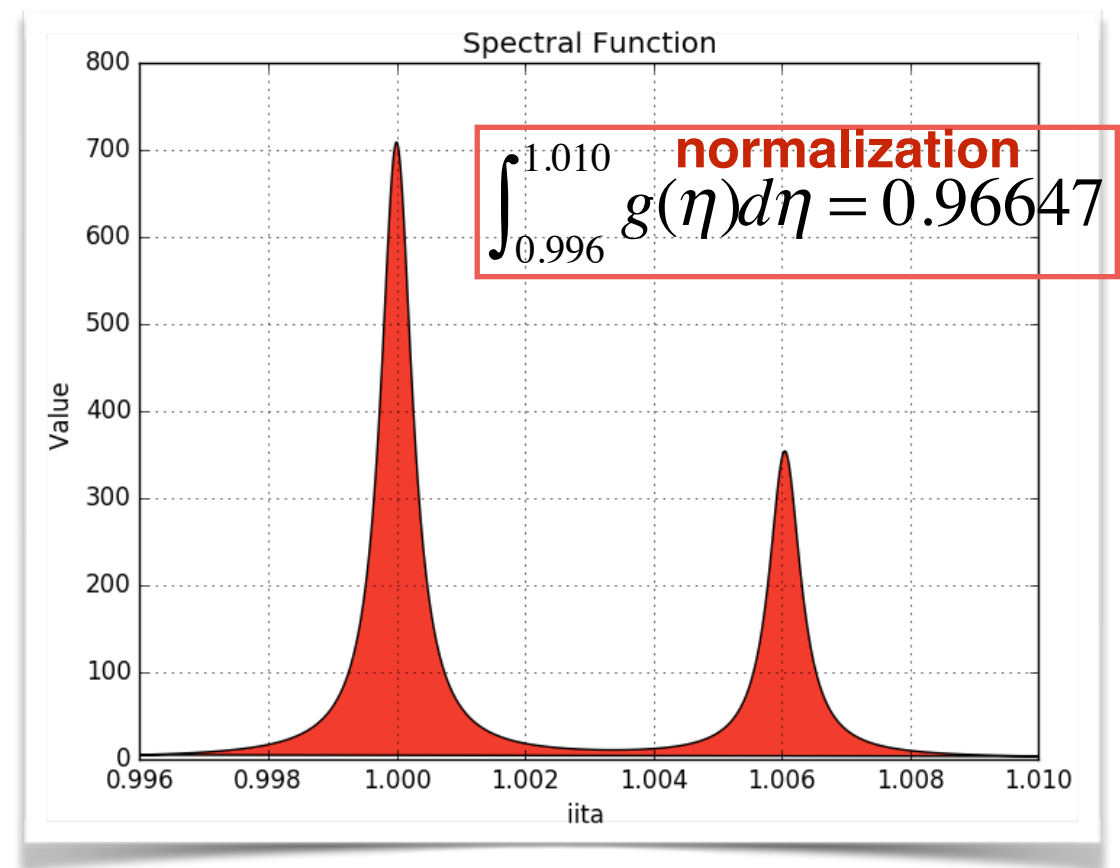


Спектральная функция

$$g(\eta) = \frac{2\pi}{3} \left[\frac{\frac{\delta\lambda_1}{\lambda_1}}{(\eta - 1)^2 + \left(\frac{\delta\lambda_1}{\lambda_1}\right)^2} + 0.5 \frac{\frac{\delta\lambda_2}{\lambda_1}}{(\eta - \frac{\lambda_2}{\lambda_1})^2 + \left(\frac{\delta\lambda_2}{\lambda_1}\right)^2} \right]$$



Реализация

```
wavelength_1 = 0.709300*1e-10
wavelength_2 = 0.713590*1e-10
path = os.path.realpath(os.path.dirname(sys.argv[0]))+'/'
def g_lambd(itta):# спектральная функция

    d_lambd1 = wavelength_1*3e-4
    d_lambd2 = wavelength_2*3e-4
    return 2/3/math.pi*((d_lambd1/wavelength_1)/(math.pow((itta-1),2)+
        math.pow(d_lambd1/wavelength_1,2)))+
        0.5*(d_lambd2/wavelength_1)/(math.pow((itta-wavelength_2/wavelength_1),2)+
        math.pow((d_lambd2/wavelength_1),2)))

gsx=[] # пустой массив x
gsy=[] # пустой массив y

itta_1 = 0.996 # предел интегрирования
itta_2 = 1.01 # предел интегрирования
shag = (itta_2 - itta_1)/1000

A = integrate.quad(g_lambd, itta_1, itta_2)# нормировка
itta=itt_1

while itta<itt_2:
    gsx.append(itt_1)
    gsy.append(g_lambd(itt_1)/A) # значение функции в массив
    itta = itta + shag
```

значение

ошибка

(0.9664760739789325, 2.7651858351461316e-11)

частный случай

Аппаратная функция

$$g_s(\theta) = \int_{x_1(\theta)}^{x_2(\theta)} e^{-x^2} dx$$

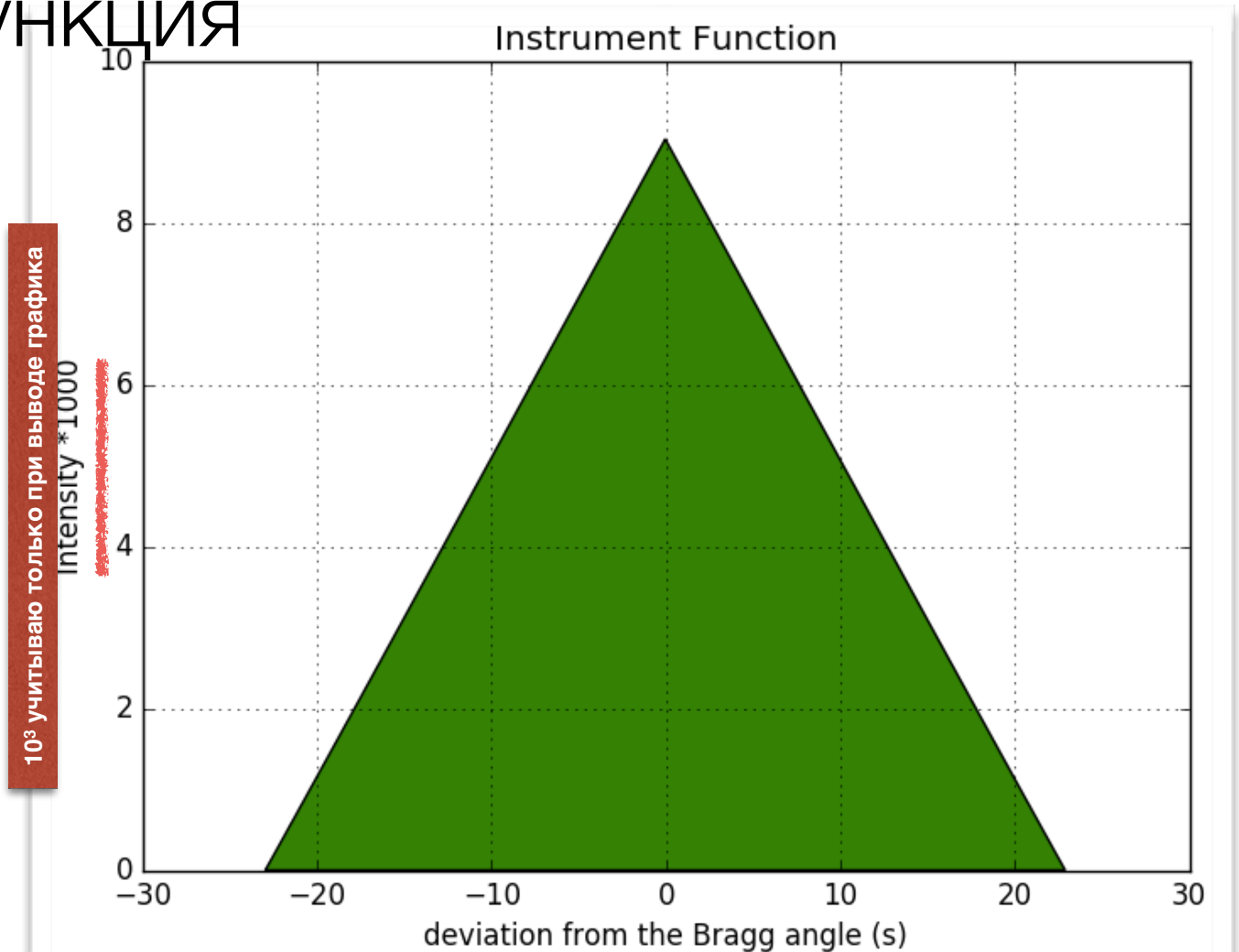
$$x_1(\theta) = -\frac{S_1/2 + |\theta|l_{1x}}{\sqrt{2}\sigma_x},$$

$$x_2(\theta) = \begin{cases} \frac{S_1/2 - |\theta|l_{1x}}{\sqrt{2}\sigma_x}, & |\theta| < \theta_1 \\ \frac{S_2/2 - |\theta|l_{2x}}{\sqrt{2}\sigma_x}, & \theta_1 < |\theta| < \theta_2, \end{cases}$$

$$\theta_{1,2} = \frac{S_2 \mp S_1}{2l_{12}}.$$

normalization

$$\int_{-30}^{30} g_s(\theta) = 7.83624 \cdot 10^{-6}$$



$L_1 = 0.54$ м; $L_2 = 0.99$ м.
 $S_1 = S_2 = 50$ мкм;

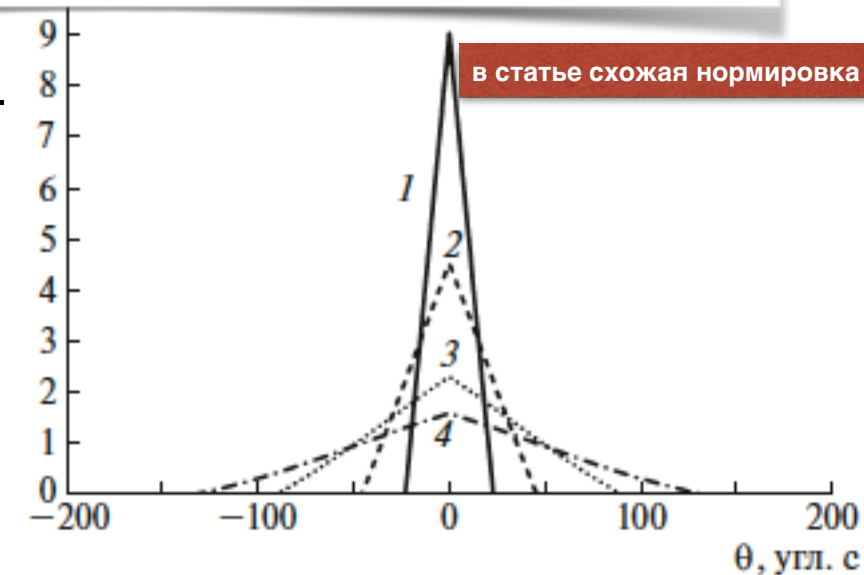


Рис. 2. Аппаратная функция углового распределения $g_s(\theta)$, рассчитанная по формулам (3) для различных значений ширины коллимирующих щелей $S = S_1 = S_2$: 1 – 50, 2 – 100, 3 – 200, 4 – 300 мкм. Оптические расстояния двухкристального дифрактометра l_1 , l_2 и l_{12} равны 0.54, 0.99 и 0.45 м соответственно.

Реализация

```
def f(teta):
    def x1(teta):
        return -(S1/2+abs(teta)*L1x)/(math.sqrt(2)*sigmaX)

    def x2(teta):
        if teta1 >= abs(teta):
            return (S1/2-abs(teta)*L1x)/(math.sqrt(2)*sigmaX)

        elif abs(teta)>=teta1 and abs(teta)<=teta2:
            return (S2/2-abs(teta)*L2x)/(math.sqrt(2)*sigmaX)

    return integrate.quad(lambda x: math.exp(-x*x) ,x1(teta),x2(teta))[0]

# -----Аппаратная функция-----
gsx=[]
gsy=[]
gsycor = []
A=0
S1 = 50*1e-6 # S1=S2 - ширина коимирующих щелей
S2 = 50*1e-6
L1x = 0.54 # L12, L1, L2 - оптические расстояния
L12 = 0.45
L2x = L1x+L12
sigmaX = 0.5*1e-3 # полуширина излучающего пятна рентгеновской трубки
teta1 = (S2-S1)/(2*L12)
teta2 = (S2+S1)/(2*L12)
teta = teta2
n_teta = 1000
shag = 2*teta/n_teta

while teta >= -teta2:
    g1 = f(teta)
    A=A+g1# нормировка
    gsx.append(math.degrees(teta)*3600)
    gsy.append(g1)
    teta=teta - shag

A=(A+(gsy[0]+gsy[-1])/2)*shag #нормировочный коэффициент по трапеции
print(A) 7.83624327593642e-06

A = integrate.quad(f,-teta2,teta2)[0]
print(A) 7.836243207337198e-06

for i in range(len(gsy)):
    gsycor.append(gsy[i]/A/1000)
```