$ Supplement Note 15: Further Topics on Numerical Integration

Numerical integration is also referred to as *quadrature*. The *trapezoidal rule* (1) and the *Simpson rule* (2) for numerically evaluating definite integral were discussed in Supplement Note 11.

$$\int_a^b f(x)dx \approx h\left(\frac{f(x_0)}{2} + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + \frac{f(x_n)}{2}\right) \tag{1}$$

$$\begin{aligned}\int_a^b f(x)dx \approx \frac{h}{3}\Big(&f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots \\ &+ 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_1)\Big)\end{aligned} \tag{2}$$

$1 Numerical Integration Using Scipy.integrate.quad Sub-package

In practice, it is convenient to use the *scipy.integrate* sub-package for performing such jobs. In addition to both rules above, this sub-package contains several more advanced methods. The following code is an example for the numerical quadrature of the definite integral:

$$I(a,b) = \int_0^1 (ax^2 + bx)\,dx\ .$$

```
# modified from an example on the official site of SciPy.org
from scipy.integrate import quad
def f(x,a,b):
    return a*x**2 + b*x
a,b = 2,1
I = quad(f, 0,1, args = (a,b))
print('I=', I)
print('real part of I=',I[0],'imaginary part of I=',I[1])
```

Notice that the outputs contain both the *real part* and the *imaginary part*. The values of the parameters a and b in the function f can be passed in calling function *quad* by using the *args* argument. The bound of integration can also be ∞, which is represented by *numpy.inf*. For example, the following codes perform the Gaussian integral and check with the analytic result Eq.(8.36) in Lecture Note:

$$\int_0^\infty e^{-x^2}\, dx = \frac{\sqrt{\pi}}{2} \tag{3}$$

```
import numpy as np
from scipy.integrate import quad
def ff(x,c): return np.exp(- c*x**2)
c = 1
I_2 = quad(ff, 0,np.inf, args = (c))
print('\int_0^\infty e^{-x**2}dx =', I_2)
ss =np.sqrt(np.pi)/2
print('sqrt(pi)/2=', ss)
```

The relation $\mathrm{erf}(\infty) = 1$ follows directly from (3), where the *error function* $\mathrm{erf}(x)$ is a monotonically increasing function and is defined in Eq.(8.41) in Lecture Note:

$$\mathrm{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt\ . \tag{4}$$

The error function is available in the sub-package *scipy.special*. In the following codes, the result $\mathrm{erf}(\infty) = 1$ is checked explicitly and the error function is plotted.

```
import numpy as np
from scipy.special import erf
from scipy.integrate import quad
print('erf(\infty)=',erf(np.inf))
import matplotlib.pyplot as plt
xx= np.linspace(0,5)
plt.plot(xx,erf(xx))
plt.title('error function')
plt.legend()
fig = plt.gcf()
fig.savefig('errorfunc.eps', format='eps')
plt.show()
```

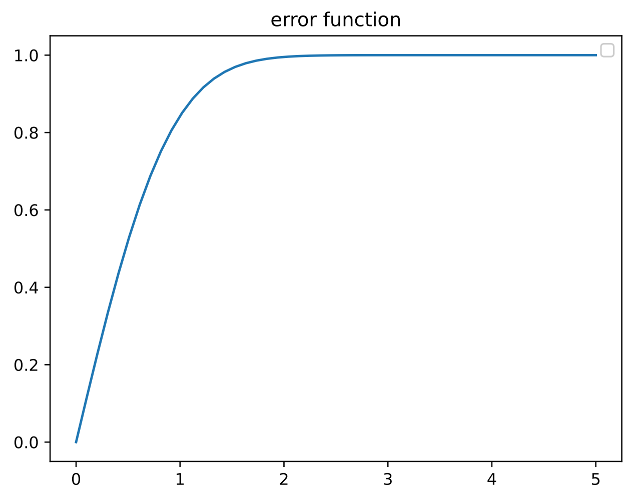

Figure 1: Error Function

$2 Magnetic Field Produced by a Steady Current

As a further application, let us calculate the magnetic field produced by a steady current using the *Biot-Savart law* for the magnetic field at a position $\mathbf{r}$ produced by a stead current I flowing in a wire segment $d\mathbf{l}$

$$d\mathbf{B}(\mathbf{r}) = \frac{\mu_0 I}{4\pi} \frac{d\mathbf{l} \times \mathbf{r}_1}{r_1^3} \,. \tag{5}$$

Here $\mathbf{r}_1$ is the vector from the current segment $d\mathbf{l}$ to the point $\mathbf{r}$. The value of μ_0 in SI units is equal to

$$\frac{\mu_0}{4\pi} = 10^{-7} \,. \tag{6}$$

In particular, let us calculate the magnetic field at a position $\mathbf{r} = (x, 0, 0)$ produced by a steady current I running along the z-axis. The current segment $d\mathbf{l}$ and the radial vector $\mathbf{r}_1$ are given explicitly by

$$d\mathbf{l} = (0, 0, dz) \quad ; \quad \mathbf{r}_1 = \mathbf{r} - (0, 0, z) = (x, 0, -z)$$

The magnetic field $\mathbf{B}$ may then be obtained from (5):

$$\mathbf{B}(\mathbf{r}) \equiv (B_x, B_y, B_z) = \left(0 \,,\; \frac{\mu_0 I}{4\pi} \int_{-\infty}^{\infty} \frac{x}{\sqrt{x^2 + z^2}^3} dz \,,\; 0\right) . \tag{7}$$

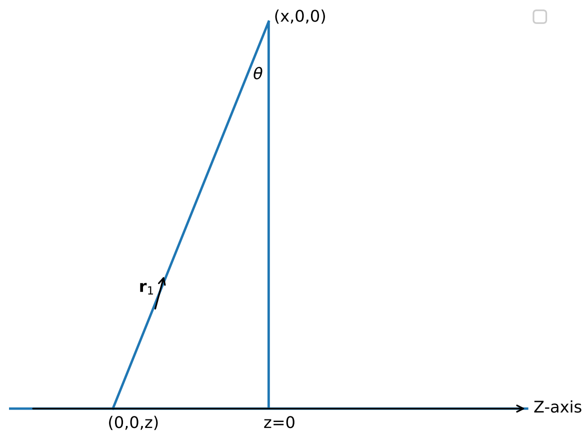

Figure 2: Magnetic field from a straight wire

They are explicit coded in *BiotSavart.py* and the results are plotted in Figure 3.

```
# BiotSavart.py
# numerical integration: \int^\infty_{-\infty} dx x/(x**2+ z**2)^{3/2} dz
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import quad
def dby(z,x):
    r1= np.sqrt(x**2+ z**2)
    return x/r1**3
By, xx = [], np.linspace(0.05,1,100)
for x in xx:
    I = quad(dby, -np.inf, np.inf, args = (x,))
    By.append(I[0])
xx2 = 2/xx- By  # differences of calculated and analytic results
error = np.sum(xx2**2) # = \sum_i (2/x_i - By_i)**2
print('total sum of the difference (2/x_i - By_i)**2=', error)
plt.plot(xx,By,label=r'$B_y$ versus x')
plt.title('Magnetic field from a wire')
plt.legend()
```

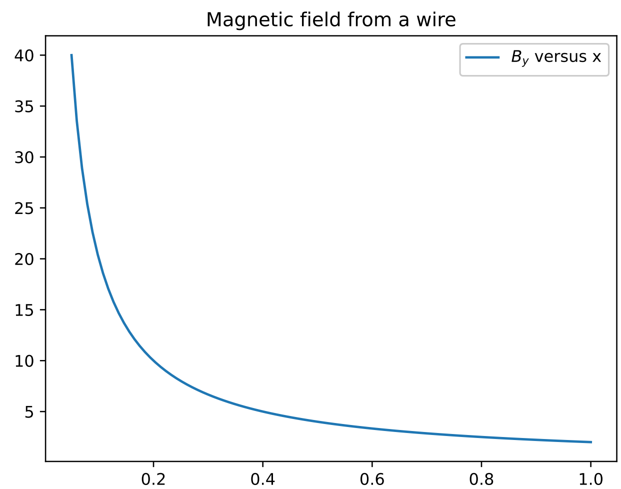

Figure 3: Magnetic field versus Distance

```
fig = plt.gcf()
fig.savefig('BfromWire.eps', format='eps')
plt.show()
```

The integration (7) may be explicitly integrated by a change of variable $z = x \tan\theta$:

$$\int_{-\infty}^{\infty} \frac{x\,dz}{\sqrt{x^2+z^2}^3} = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{x \sec^2\theta d\theta}{\sqrt{x^2+x^2\tan^2\theta}^3} = \frac{1}{x}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos\theta d\theta = \frac{2}{x}$$

This leads to the result for the magnetic field $\mathbf{B}(\mathbf{r}) \equiv (0, B_y, 0)$ with B_y given by

$$B_y(x,0,0) = \frac{\mu_0 I}{2\pi}\,\frac{1}{x}\,. \tag{8}$$

In other words, the direction of the magnetic field produced by an infinite wire is always perpendicular to the plane consisted of the field point $\mathbf{r}$ and the wire. *Its magnitude is inversely proportional to the distance of the point from the wire.*

As a simple exercise, obtain the value of π by calculating the area inside a unit circle, $x^2 + y^2 = 1$, i.e., by numerically evaluating the following integral:

$$\pi = 2\int_{-1}^{1} \sqrt{1-x^2}\,dx$$