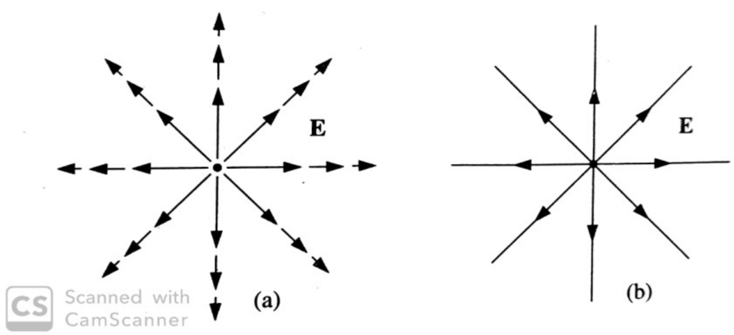
Python code: Dirac-delta function and different Limit representation of Dirac-delta function

Shyamal Bhar
Assistant Professor
Vidyasagar College for Women
Kolkata 700 006

Dirac delta function: To know the basics of the Dirac Delta function let us first calculate the divergence of $\frac{\hat{r}}{r^2}$.

Consider the vector function $\vec{E} = \frac{\hat{r}}{r^2}$.



This vector function is directed radially outward and has a large positive divergence. This can be seen from the figure below.

What happens, if we want to calculate the divergence of $\vec{E}$? Let us calculate it,

$$\nabla \cdot \vec{E} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{1}{r^2}\right)$$
$$= \frac{1}{r^2}\frac{\partial}{\partial r}(1)$$
$$= 0 \qquad \qquad ....(1)$$

Here we have used spherical polar co-ordinate system in calculating the divergence. Now, we shall want to apply the divergence theorem to this function. For this, first calculate the surface integration of the vector function over a sphere of radius R, centered at the origin. The integral is,

$$\oint \vec{E}\cdot d\vec{S} = \int\left(\frac{\hat{r}}{R^2}\right)\cdot\left(R^2 \sin(\theta)\, d\theta d\phi\, \hat{r}\right)$$
$$= \int_0^{\pi} \sin(\theta)\, d\theta \int_0^{2\pi} d\phi$$
$$= 4\pi \qquad \qquad ....(2)$$

And the volume integral of the divergence of the vector function is,

$$\int \nabla \cdot \vec{E}\, dv = 0 \qquad ....(3)$$ [According to equation (1)]

But according to the divergence theorem this volume integral should be 4π. Now, the question is, what is the problem in equation (3). The source of the problem is the point $r = 0$. At this point the vector function $\vec{E}$ blows up. It is clear that equation (1) i.e. $\nabla \cdot \vec{E} = 0$ everywhere except at the point $r = 0$. The surface integral is independent of R. So the surface integral will be 4π for any sphere centered at the origin, whatever be the radius of the sphere. It is clear that the entire contribution of the surface integral comes from the point $r = 0$. So, equation (3) is true for any point except $r = 0$ and it should be 4π if we include the point $r = 0$ for the consistence of the divergence theorem. So, $\nabla \cdot \vec{E}$ has a bizarre property that it vanishes everywhere except at one point and its integral over any volume containing that point is 4π. This type of function is called Dirac Delta function.

One dimensional Dirac Delta function: The One dimensional Dirac Delta function $\delta(x)$ is defined as a function which has an infinitely high, infinitesimally narrow spike with unit area. Mathematically,

$$\delta(x) = \begin{cases} 0, & \text{if } x \neq 0 \\ \infty, & \text{if } x = 0 \end{cases}$$

and

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1$$

An important property of a Delta function:

Now, if $f(x)$ is an ordinary continuous function, then the product $f(x)\delta(x)$ is zero everywhere except at $x = 0$. So, it can be written as

$$f(x)\delta(x) = f(0)\delta(x)$$

Hence,

$$\int_{-\infty}^{\infty} f(x)\delta(x)\, dx = f(0)\int_{-\infty}^{\infty} \delta(x)\, dx$$
$$= f(0)$$

One can take the limits of integration from $-\varepsilon$ to $+\varepsilon$ instead of $-\infty$ to $+\infty$, as the $\delta(x)$ is non-zero at $x = 0$. If we consider that the peak of the delta function is at $x = a$ then we can write

$$\delta(x-a)=\begin{cases}0, & \text{if } x\neq a\\ \infty, & \text{if } x=a\end{cases}$$

and

$$\int_{-\infty}^{\infty}\delta(x-a)\,dx=1$$

The product of this delta function with an arbitrary function $f(x)$ is

$$f(x)\delta(x-a)=f(a)\delta(x)$$

and the integration of this product is

$$\int_{-\infty}^{\infty}f(x)\delta(x-a)\,dx=f(a)$$

Generalizing it in three dimension it can be written as

$$\int_{-\infty}^{\infty}f(\vec{r})\delta^3(\vec{r}-\vec{a})\,d\tau=f(\vec{a})$$

Now, we can in the position to write the expression of the divergence of $\frac{\hat{r}}{r^2}$. It is zero except at the origin and its integral over any volume containing the origin is a constant i.e. 4π. This is basically the definition of Dirac delta function, so

$$\nabla\cdot\left(\frac{\hat{r}}{r^2}\right)=4\pi\delta^3(\vec{r})\ .$$

Limiting representation of Dirac Delta function:

Technically $\delta(x)$ is not a function, since its value is not finite at $x=0$. In mathematical literature it is known as a generalized function or distribution. Dirac delta function basically is the limit of sequence of functions, such as rectangles $R_\varepsilon(x)$ of width ε and height $\frac{1}{\varepsilon}$, or isosceles triangles $T_\varepsilon(x)$ of base 2ε and height $\frac{1}{\varepsilon}$. There are many limit representation of Dirac delta functions, few are listed below:

1. **Limit of sequence of Rectangles $R_\varepsilon(x)$:**

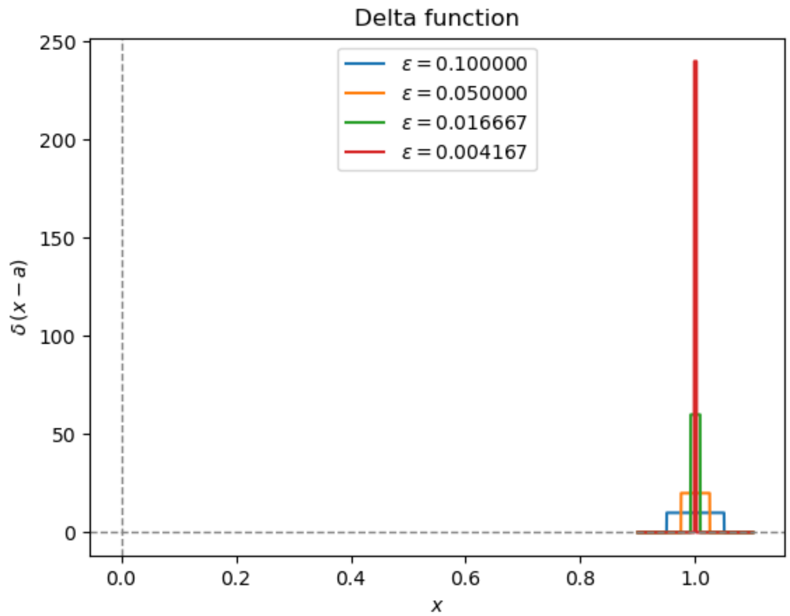
$$\delta(x)=\lim_{\varepsilon\to 0}R_\varepsilon(x)$$

$$\text{where,}\quad R_\varepsilon(x)=\begin{cases}0 & \text{for } x<-\frac{\varepsilon}{2}\\ \frac{1}{\varepsilon} & \text{for } -\frac{\varepsilon}{2}<x<\frac{\varepsilon}{2}\\ 0 & \text{for } x>\frac{\varepsilon}{2}\end{cases}$$

Here the width of the rectangle $R_{\varepsilon}(x)$ is ε and height $\frac{1}{\varepsilon}$ so that the area is unity.

```
# Dirac Delta function
# From Rectangles
# eps-->0
# d(x)=0 for x>eps/2
#      =1/eps for (-eps/2))<x<(eps/2))
#      =0 for x<-eps/2
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function

x1,x2=-0.1+a,0.1+a  # x-range for plotting about the peak of
delta function


# Defining delta function by rectangles

delta=lambda x,eps: 1.0/eps if -eps/2.0<x and x<eps/2.0 else 0
delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=0.1
x=np.linspace(x1,x2,1000)
n_val=np.arange(1,5)
for i in n_val:
         eps=eps/(i)
         plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()

#######################################
# verification of integration of del(x-a)g(x) for limit -inf to
inf =g(a)

# Given function
g=lambda x: x**2+3

# evaluation of integration
xp=np.linspace(a-eps,a+eps,1000)
```

```
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```



----

2. **Limit of sequence of isosceles triangles** $T_\varepsilon(x)$:

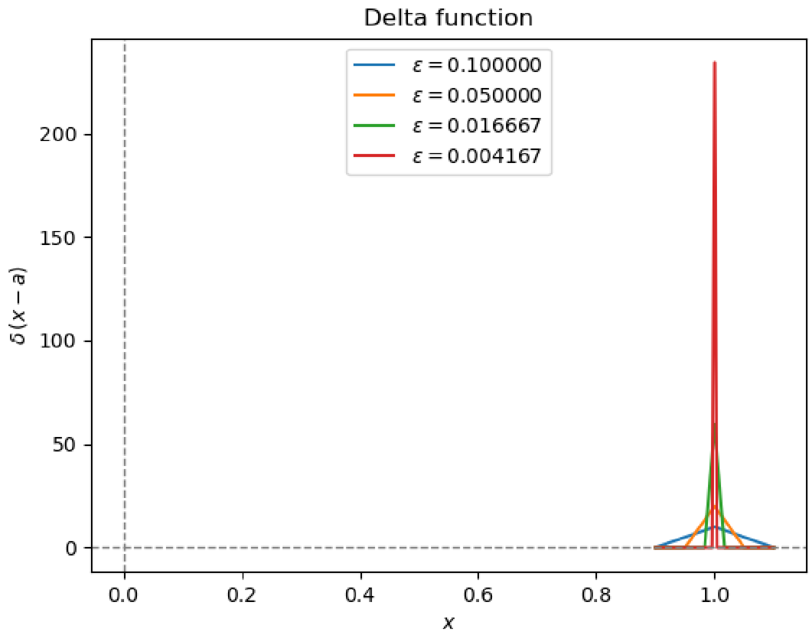
$$\delta(x) = \lim_{\varepsilon\to 0} T_\varepsilon(x)$$

where, $$T_\varepsilon(x) = \begin{cases} 0 & \text{for } |x| > \varepsilon \\ \dfrac{\left(1-\dfrac{|x|}{\varepsilon}\right)}{\varepsilon} & \text{for } |x| < \varepsilon \end{cases}$$

Here the base of the triangle $T_\varepsilon(x)$ is 2ε and height $\dfrac{1}{\varepsilon}$ so that the area of the triangle is unity.

```
# Dirac Delta function
# From Isosceles Triangles
# eps-->0
# d(x)=0 for x<-eps
#      =(1+x/eps)/eps for -eps<x<0
#      =(1-x/eps)/eps for 0<x<eps
#      =0   for x>eps
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function
```

```
x1,x2=-0.1+a,0.1+a   # x-range for plotting about the peak of delta function


# Defining delta function by Isosceles Triangles

def delta(x,eps):
         if x<-eps or x>eps:
                  return 0.
         else:
                  return (1-np.abs(x)/eps)/eps
delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=0.1
x=np.linspace(x1,x2,1000)
n_val=np.arange(1,5)
for i in n_val:
         eps=eps/(i)
         plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()

######################################
# verification of integration of del(x-a)g(x) for limit -inf to inf =g(a)

# Given function
g=lambda x: x**2+3

# evaluation of integration
xp=np.linspace(a-eps,a+eps,1000)
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```



---

3. **Limit of sequence of Gaussian functions** $G_\varepsilon(x)$:

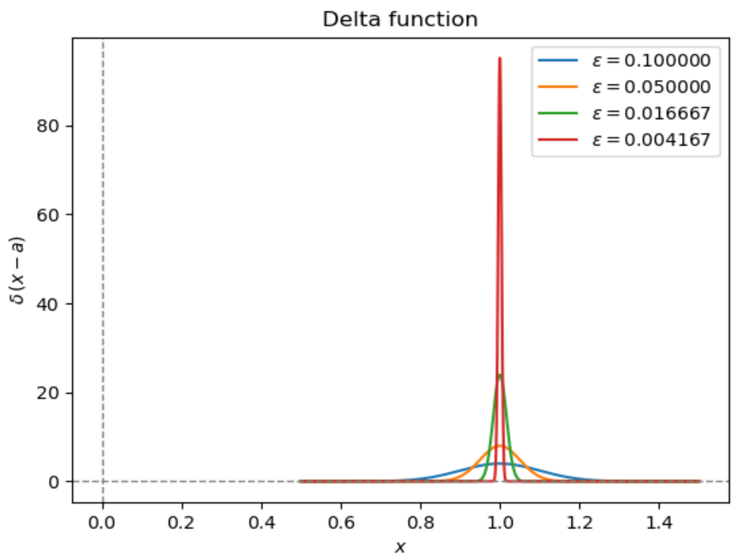
$$\delta(x) = \lim_{\varepsilon \to 0} G_\varepsilon(x)$$

$$\text{where,} \qquad G_\varepsilon(x) = \frac{1}{\varepsilon\sqrt{\pi}} e^{-\frac{x^2}{\varepsilon^2}}$$

This is the normalized Gaussian distribution function. The area under the curve is unity and the peak value $x = 0$ is $\frac{1}{\varepsilon\sqrt{\pi}}$.

```
# Dirac Delta function
# From Gaussian function
# eps-->0
# d(x)=(1/(eps*sqrt(pi)))*exp(-x**2/eps**2)

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function

x1,x2=-0.5+a,0.5+a  # x-range for plotting about the peak of
delta function


# Defining delta function by Gaussian

def delta(x,eps):
```

```
        return 1.0/(eps*np.sqrt(2.*np.pi))*np.exp(-x**2/(2.*eps**2))
delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=0.1
x=np.linspace(x1,x2,1000)
n_val=np.arange(1,5)
for i in n_val:
        eps=eps/(i)
        plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()

######################################
# verification of integration of del(x-a)g(x) for limit -inf to inf =g(a)

# Given function
g=lambda x: x**2+3

# evaluation of integration
xp=np.linspace(a-5.0*eps,a+5.0*eps,1000)
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```



---

4. **Limit of sequence of Exponential functions** $E_\varepsilon(x)$:

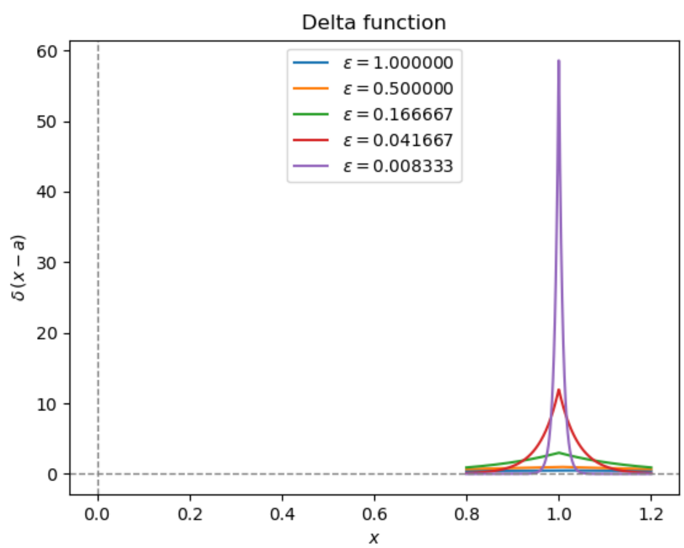
$$\delta(x) = \lim_{\varepsilon \to 0} E_\varepsilon(x)$$

$$\text{where,} \qquad E_\varepsilon(x) = \frac{1}{2\varepsilon} e^{-\frac{|x|}{\varepsilon}}$$

The area under the curve is unity and the peak value $x = 0$ is $\frac{1}{2\varepsilon}$.

```
# Dirac Delta function
# From Exponentials
# eps-->0
# d(x)=(n/2)*(exp(-abs(x)/eps)

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function

x1,x2=-0.2+a,0.2+a  # x-range for plotting about the peak of
delta function


# Defining delta function by Exponential

delta=lambda x,eps: (1.0/(2.0*eps))*np.exp(-1.0*np.abs(x)/eps)
delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=1.0
```

```
x=np.linspace(x1,x2,1000)
n_val=np.arange(1,6)
for i in n_val:
        eps=eps/i
        plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)
plt.show()

######################################
# verification of integration of del(x-a)g(x) for limit -inf to
inf =g(a)

# Given function
g=lambda x: x**2+3

# evaluation of integration
xp=np.linspace(-15*eps+a,15*eps+a,1000)
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```



---

5. **Limit of sequence of sinc functions** $S_\varepsilon(x)$:

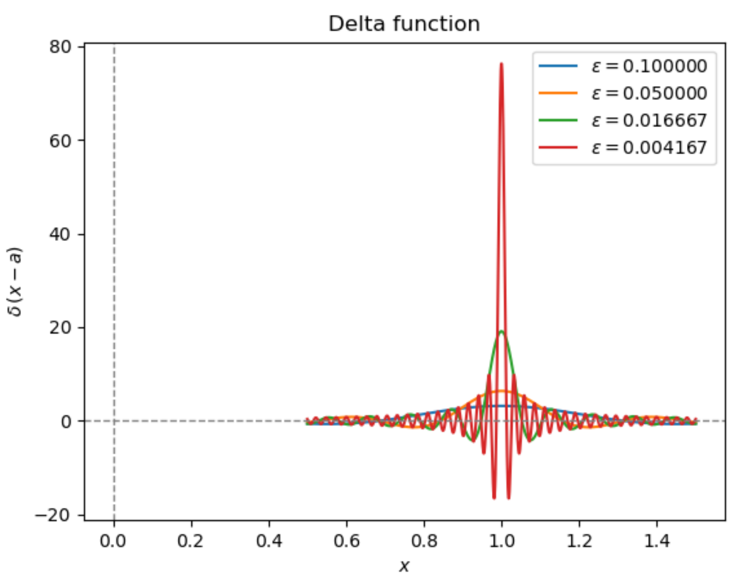
$$\delta(x) = \lim_{\varepsilon \to 0} S_\varepsilon(x)$$

$$\text{where,} \quad S_\varepsilon(x) = \frac{\sin(x/\varepsilon)}{\pi x}$$

This function arises frequently in signal processing and the theory of Fourier transforms. The full name of this function is since cardinal but it is commonly referred to "sinc" function.

```
# Dirac Delta function
# From sine cardinal function
# eps-->0
# d(x)=sin(x/eps)/(pi*x)

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function

x1,x2=-.5+a,.5+a  # x-range for plotting about the peak of delta
function


# Defining delta function by sine cardinal function

def delta(x,eps):
         if x!=0.0:
                  return np.sin(x/eps)/(np.pi*x)
         else:
                  return 1./(eps*np.pi)

delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=0.1
x=np.linspace(x1,x2,1000)
n_val=np.arange(1,5)
for i in n_val:
         eps=eps/(i)
         plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()
```

```
########################################
# verification of integration of del(x-a)g(x) for limit -inf to
inf =g(a)

# Given function
g=lambda x: x**2+3

# evaluation of integration
xp=np.linspace(a-60*eps,a+60*eps,1000)
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#######################################
```



----

6. Limit of sequence of modified sinc function $Sm_{\varepsilon}(x)$:

$$\delta(x) = \lim_{\varepsilon \to 0} Sm_{\varepsilon}(x)$$

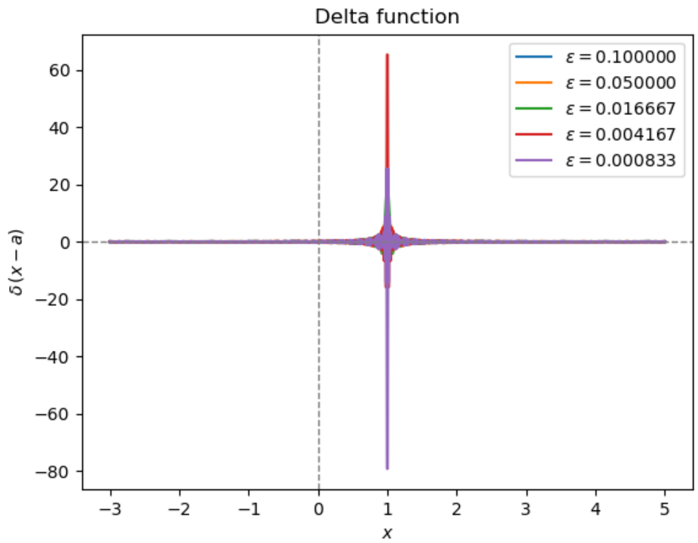
$$\text{where,} \quad Sm_{\varepsilon}(x) = \frac{1}{2\pi} \frac{\sin\left[\frac{\left(1+\frac{2}{\varepsilon}\right)x}{2}\right]}{\sin\left(\frac{x}{2}\right)}$$

The peak value $x = 0$ is $\left(\frac{1}{2}+\frac{1}{\varepsilon}\right)/\pi$.

```
# Dirac Delta function
```

```
# From modified sine cardinal function
# eps-->0
# d(x)=1.0/(2*pi)*(sin((1+2/eps)*x/2)/sin(x/2)
#      = (1/2+1/eps)/pi  for x=0

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function

x1,x2=-4+a,4.+a  # x-range for plotting about the peak of delta
function


# Defining delta function by modified sine function

def delta(x,eps):
         if x!=0:
                  return
1.0/(2.*np.pi)*(np.sin((1.+2./eps)*x/2.)/np.sin(x/2.))
         else:
                  return (0.5+1./eps)/np.pi
delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=0.1

x=np.linspace(x1,x2,1000)
n_val=np.arange(1,6)
for i in n_val:
         eps=eps/(i)
         plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()

######################################
# verification of integration of del(x-a)g(x) for limit -inf to
inf =g(a)

# Given function
g=lambda x: x**2+3.0

# evaluation of integration
xp=np.linspace(a-5,a+5,10000)
```

```
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```



----

7. **Limit of sequence of Lorentzian $L_\varepsilon(x)$:**

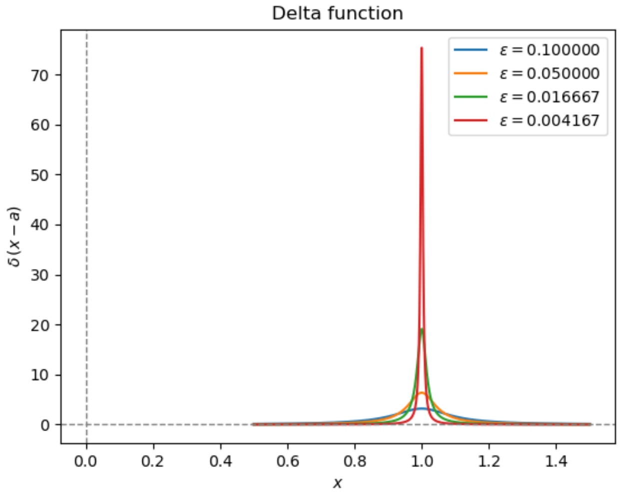
$$\delta(x) = \lim_{\varepsilon \to 0} L_\varepsilon(x)$$

$$\text{where,} \quad L_\varepsilon(x) = \frac{1}{\pi\varepsilon}\frac{1}{1+\frac{x^2}{\varepsilon^2}}$$

The peak value $x=0$ is $\frac{1}{\pi\varepsilon}$.

```
# Dirac Delta function
# From Lorentzian function
# eps-->0
# d(x)=1.0/(pi*eps*(1+x**2/eps**2)

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0   # position of peak of delta function

x1,x2=-0.5+a,0.5+a   # x-range for plotting about the peak of
delta function
```

```
# Defining delta function by Lorentzian function

def delta(x,eps):
         return 1.0/(np.pi*eps*(1+x**2/eps**2))
delta=np.vectorize(delta)


# plotting delta functions for different eps values

eps=0.1

x=np.linspace(x1,x2,1000)
n_val=np.arange(1,5)
for i in n_val:
         eps=eps/(i)
         plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$" %(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()

######################################
# verification of integration of del(x-a)g(x) for limit -inf to
inf =g(a)

# Given function
g=lambda x: x**2+3.0

# evaluation of integration
xp=np.linspace(a-10,a+10,10000)
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```



---

8. Limit of sequence of Inverse Cosh Square functions $C_\varepsilon(x)$:

$$\delta(x) = \lim_{\varepsilon\to 0} C_\varepsilon(x)$$

$$\text{where,} \quad C_\varepsilon(x) = \frac{1}{2\varepsilon}\frac{1}{\cosh^2\left(\frac{x}{\varepsilon}\right)}$$

The peak value $x=0$ **is** $\frac{1}{2\varepsilon}$**.**

```
# Dirac Delta function
# From Inverse cosine hyperbolic square function
# eps-->0
# d(x)=1.0/(2*eps*cosh(x/eps)**2)

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps


a=1.0  # position of peak of delta function

x1,x2=-0.5+a,0.5+a   # x-range for plotting about the peak of delta function


# Defining delta function by Inverse cosine hyperbolic square

def delta(x,eps):
        return 1.0/(2.0*eps*np.cosh(x/eps)**2)
delta=np.vectorize(delta)


# plotting delta functions for different eps values
```

```
eps=0.1
x=np.linspace(x1,x2,1000)
n_val=np.arange(1,5)
for i in n_val:
         eps=eps/(i)
         plt.plot(x,delta(x-a,eps),label="$\epsilon=%f$"
%(eps))

plt.xlabel("$x$")
plt.ylabel("$\delta\,(x-a)$")
plt.title("Delta function")
plt.axvline(0, c='gray',ls='--', lw=1)
plt.axhline(0, c='gray',ls='--', lw=1)
plt.legend(loc='best', fontsize=10)

plt.show()

######################################
# verification of integration of del(x-a)g(x) for limit -
inf to inf =g(a)

# Given function
g=lambda x: x**2+3

# evaluation of integration
xp=np.linspace(a-5.0*eps,a+5.0*eps,1000)
func=simps(g(xp)*delta(xp-a,eps),xp)

# Exact result
exact=g(a)
# Printing results
print 'calculated=',func, 'exact=',exact
#####################################
```