Chap. 5

Integrals and Derivatives

Soon-Hyung Yook

April 18, 2019

Table of Contents I

Taylor Expansion

- One basic tool that we will use in this class.

Taylor expansion of a function $f(x)$ around a point x_0

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \cdots \tag{1}$$

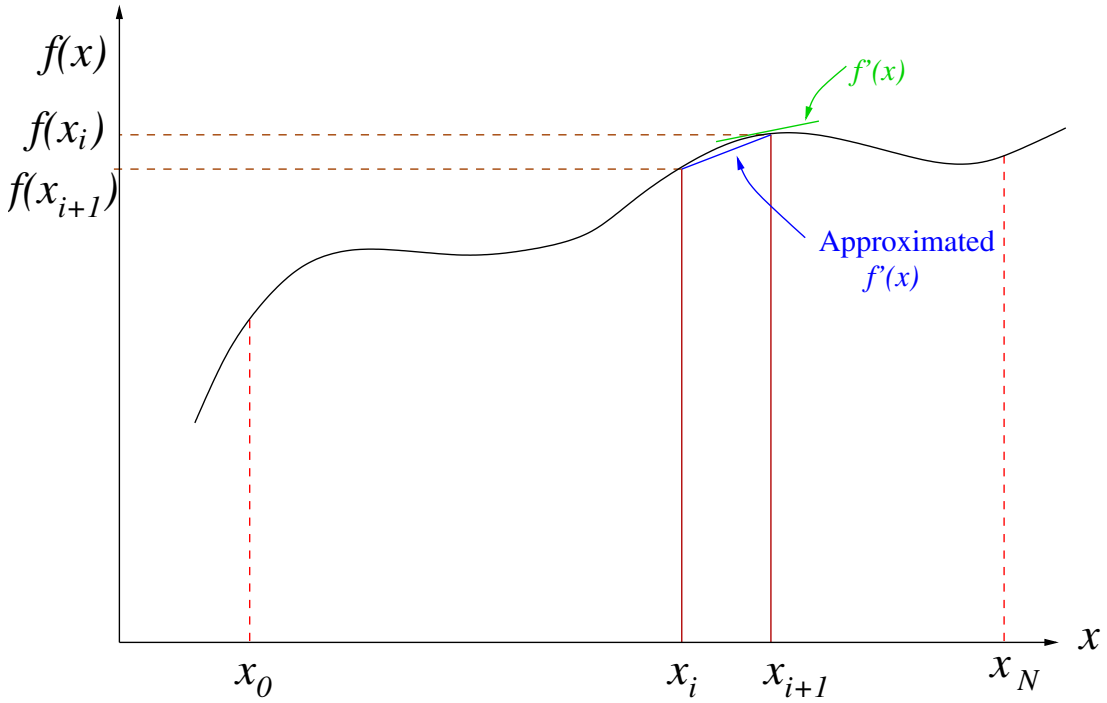
Taylor expansion of a mutivariable function $f(x, y, \cdots)$ around a point $(x_0, y_0, \cdots)$

$$\begin{aligned} f(x, y, \cdots) = \quad & f(x_0, y_0, \cdots) + (x - x_0)f_x(x_0, y_0, \cdots) \\ & + (y - y_0)f_y(x_0, y_0, \cdots) + \cdots \\ & + \frac{(x-x_0)^2}{2!}f_{xx}(x_0, y_0, \cdots) \\ & + \frac{(y-y_0)^2}{2!}f_{yy}(x_0, y_0, \cdots) \\ & + \frac{(x-x_0)(y-y_0)}{2!}f_{xy}(x_0, y_0, \cdots) + \cdots \end{aligned}$$

First-Order Derivative: Single Variable, Two-Point Formula

- From the high school definition of the first-oder derivative.

$$f'(x_i) = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} = \lim_{\Delta x \to 0} \frac{\Delta f_i}{\Delta x} \tag{2}$$



First-Order Derivative: Single Variable, Two-Point Formula

- Or more exactly, from the Taylor series (Eq. (1))
- Let $\Delta x = h$.

First order derivative: Two-Point Definition (forward difference)

$$\frac{df_i}{dx} = f_i' = \frac{f_{i+1} - f_i}{h} + \mathcal{O}(h^2) \tag{3}$$

First order derivative: Two-Point Definition (backward difference)

$$\frac{df_i}{dx} = f_i' = \frac{f_i - f_{i-1}}{h} + \mathcal{O}(h^2) \tag{4}$$

Accumulated Error

Accumulated Error

- For each step the largest error is proportional to h^2.
- At the end of the interval $[x_0, x_N]$, m steps of derivative has been made.
- The accumulated error becomes

$$\sum_i^m h^2 = mh^2 = \frac{x_N - x_0}{h} h^2 = (x_N - x_0)h = \mathcal{O}(h)$$

Example: $d\sin(x)/dx$ I

```
from math import pi,sin,cos
from numpy import linspace
from pylab import plot,show,xlim,ylim,bar

def f(x):
    y=sin(x)
    return y

def df_dx_forward(x0,xe,n):
  x=x0
  h=(xe-x0)/n
  xi=[]
  yi=[]
  for i in range(0,n):
    yi.append((f(x+h)-f(x))/h)
    xi.append(x)
    x+=h
  plot(xi,yi,'ro')
  return(1)

def df_dx_backward(x0,xe,n):
  x=x0
  h=(xe-x0)/n
  xi=[]
  yi=[]
```

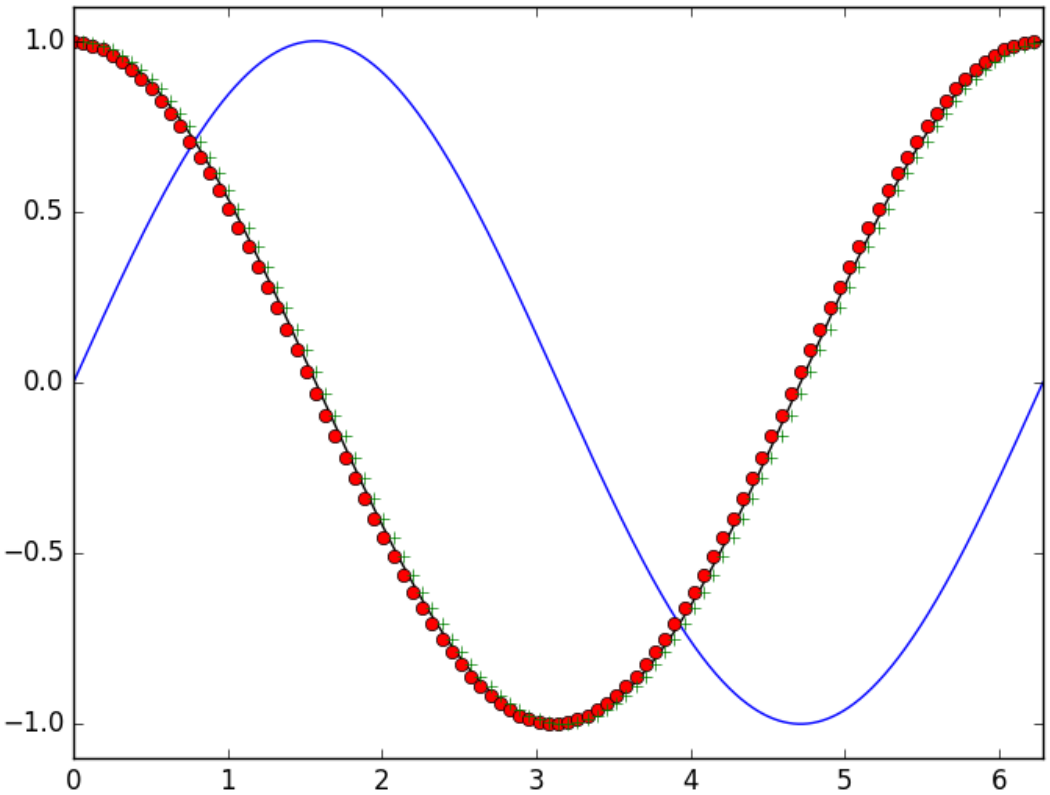
Example: $d\sin(x)/dx$ II

```
    for i in range(0,n):
        x+=h
        yi.append((f(x)-f(x-h))/h)
        xi.append(x)
    plot(xi,yi,'g+')
    return(1)

# ########################################
#   Plot sin(x)
# ########################################
Np=1000
px=linspace(0.0,2*pi,Np)
py=[]
dpy=[]
for xx in px:
    py.append(f(xx))
    dpy.append(cos(xx))
plot(px,py)
plot(px,dpy,'k')

# ########################################
#   Plot dsin(x)/dx
# ########################################
Ndx=100
df_dx_forward(0.0,2*pi,Ndx)
```

Example: $d\sin(x)/dx$ III

```
df_dx_backward(0.0,2*pi,Ndx)

ylim(-1.1,1.1)
xlim(0,2*pi)
show()
```

Example: $d\sin(x)/dx$ IV



Improved Method: Three-Point Definition

Again from the Taylor series:

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(x)h^3 + \cdots \tag{5}$$

$$f(x-h) = f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 - \frac{1}{6}f'''(x)h^3 + \cdots \tag{6}$$

From Eq. (5) and Eq. (6)

$$f(x+h) - f(x-h) = 2f'(x)h + \mathcal{O}(h^3) \tag{7}$$

Therefore,

Three-Point Definition

$$f'(x_i) = \frac{f(x_i+h) - f(x_i-h)}{2h} + \mathcal{O}(h^3) \tag{8}$$

Accumulated error: $\mathcal{O}(h^2)$

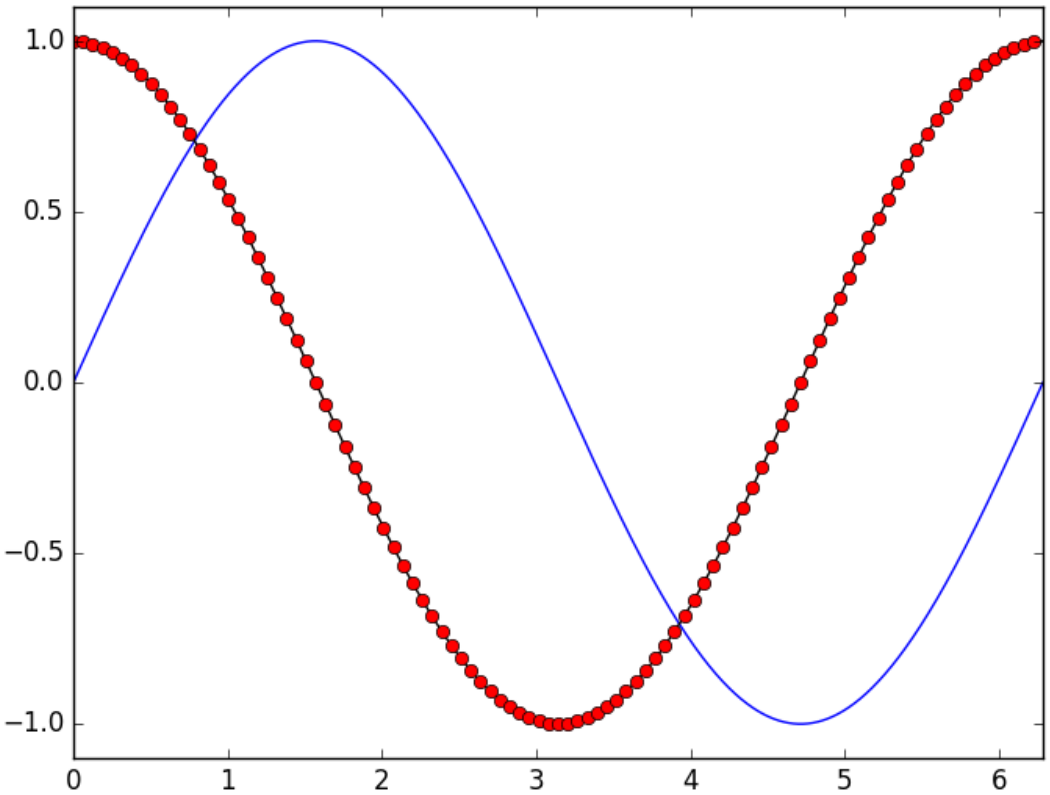
Example: $d\sin(x)/dx$ I

```
from math import pi,sin,cos
from numpy import linspace
from pylab import plot,show,xlim,ylim,bar

def f(x):
    y=sin(x)
    return y

def df_dx(x0,xe,n):
   x=x0
   h=(xe-x0)/n
   xi=[]
   yi=[]
   for i in range(0,n):
      yi.append((f(x+h)-f(x-h))/(2*h))
      xi.append(x)
      x+=h
   plot(xi,yi,'ro')
   return(1)

# ########################################
#  Plot sin(x)
# ########################################
Np=1000
px=linspace(0.0,2*pi,Np)
```

Example: $d\sin(x)/dx$ II

```
py=[]
dpy=[]
for xx in px:
  py.append(f(xx))
  dpy.append(cos(xx))
plot(px,py)
plot(px,dpy,'k')

# ########################################
#  Plot dsin(x)/dx
# ########################################
Ndx=100
df_dx(0.0,2*pi,Ndx)

ylim(-1.1,1.1)
xlim(0,2*pi)
show()
```

Example: $d\sin(x)/dx$ III



Homework

Find the first order derivative numerically

$$f(x) = x^2$$

in the interval $x \in [-2, 2]$ using both two-point and three-point definitions.

Second Order Derivative

Again from the Taylor series for $f(x)$, add Eq. (5) and Eq. (6)

$$f(x+h) - 2f(x) + f(x-h) = h^2 f''(x) + \mathcal{O}(h^4) \tag{9}$$

Therefore,

Second Order Derivative:Three-Point Definition

$$f''(x_i) = \frac{f(x_i+h) - 2f(x) + f(x_i-h)}{h^2} + \mathcal{O}(h^2) \tag{10}$$

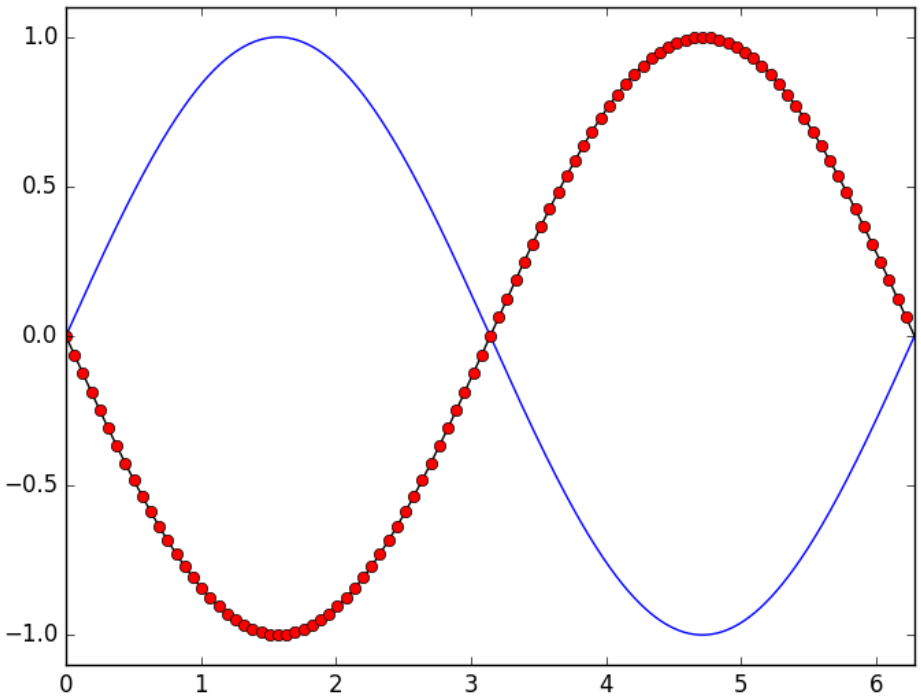
Example: Find $\frac{d^2 \sin(x)}{dx^2}$ I

```
from math import pi,sin,cos
from numpy import linspace
from pylab import plot,show,xlim,ylim,bar

def f(x):
    y=sin(x)
    return y

def d2f_dx2(x0,xe,n):
   x=x0
   h=(xe-x0)/n
   xi=[]
   yi=[]
   for i in range(0,n):
      yi.append((f(x+h)-2*f(x)+f(x-h))/h**2)
      xi.append(x)
      x+=h
   plot(xi,yi,'ro')
   return(1)

# ########################################
#  Plot sin(x)
# ########################################
Np=1000
px=linspace(0.0,2*pi,Np)
```

Example: Find $\frac{d^2 \sin(x)}{dx^2}$ II

```
py=[]
dpy=[]
for xx in px:
    py.append(f(xx))
    dpy.append(-f(xx))
plot(px,py)
plot(px,dpy,'k')

# ########################################
#   Plot d^2 sin(x)/dx^2
# ########################################
Ndx=100
d2f_dx2(0.0,2*pi,Ndx)

ylim(-1.1,1.1)
xlim(0,2*pi)
show()
```

Example: Find $\frac{d^2 \sin(x)}{dx^2}$ III



Homework

Find the second order derivative numerically

$$f(x) = x^2$$

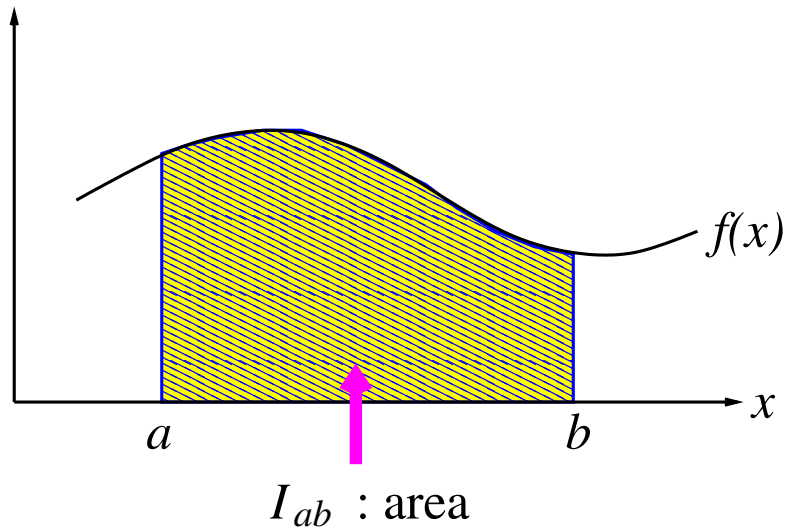
in the interval $x \in [-2, 2]$.

Numerical Integration

- Not all integrations are carried out analytically.
  - Ex. integrals including erf(x), $\Gamma(x)$, etc.
  - $\Rightarrow$ the results should be found numerically.
- Definite Integral

$$I_{[a,b]} = \int_a^b f(x)dx \tag{11}$$

for simply assume that $f(x) > 0$ for $x \in [a, b]$ then $I_{[a,b]}$ is just the area enclosed by $f(x)$ [high school definition].



Basic Idea

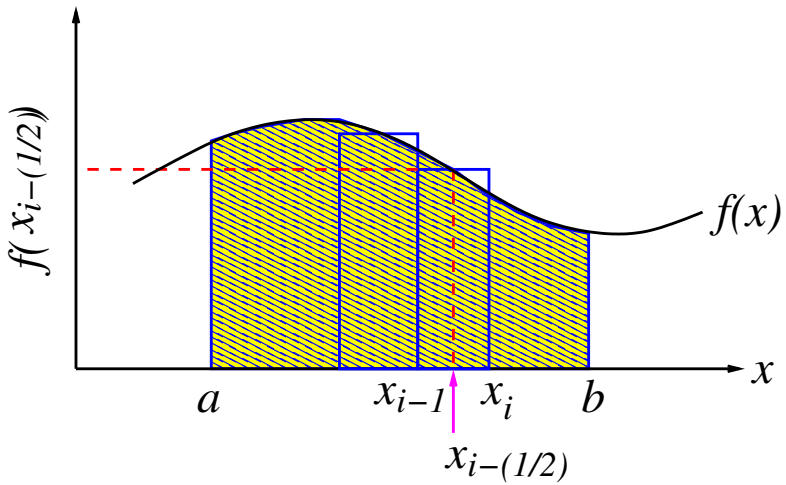
- Devide the interval $[a, b]$ into N sclices.
- For convenience, the width of each slice is identical, i.e. evenly spaced with intherval h.
- If we label the position (or the data points) as x_i, with $i = 1, 2, \cdots, N$, the integral Eq. (11) can be expressed as s summation of integrals over each slice.

Basic Idea

$$\int_a^b f(x)dx = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)dx \tag{12}$$

- Find a numerical scheme that evaluates the summation over each slice.

Rectangular Method



- The most intuitive method
- For simplicity, let all subinterval has equal size, $h = x_i - x_{i-1}$.
- Divide the intreval $[a, b]$ into N subintervals: $Nh = b - a$.
- Let $\bar{f}_i \equiv \frac{1}{h}\int_{x_{i-1}}^{x_i} f(x)dx$.

Then

Rectangular Method

$$I_{[a,b]} = \int_a^b f(x)dx = \sum_{i=1}^{N}\int_{x_{i-1}}^{x_i} f(x)dx = h\sum_{i=1}^{N}\bar{f}_i. \tag{13}$$

For slow varying function $\bar{f}_i \approx f(x_{i-1/2})$ where $x_{i-1/2} = (x_{i-1} + x_i)/2$.

Rectangular Method–Error Estimation

Contribution of each interval

$$I_{[x_{i-1},x_i]} = \int_{x_{i-1}}^{x_i} f(x)dx \approx hf_{i-1/2}$$

$$f(x) = f_{i-\frac{1}{2}} + f'_{i-\frac{1}{2}}(x - x_{i-\frac{1}{2}}) + \frac{1}{2}f''_{i-\frac{1}{2}}(x - x_{i-\frac{1}{2}})^2 + \frac{1}{3!}f'''_{i-\frac{1}{2}}(x - x_{i-\frac{1}{2}})^3 + \cdots$$

$$\begin{aligned}\int_{x_{i-1}}^{x_i} f(x)dx = f_{i-\frac{1}{2}} \int_{x_{i-1}}^{x_i} dx &+ f'_{i-\frac{1}{2}} \int_{x_{i-1}}^{x_i} (x - x_{i-\frac{1}{2}})dx \\ &+ \frac{1}{2}f''_{i-\frac{1}{2}} \int_{x_{i-1}}^{x_i} (x - x_{i-\frac{1}{2}})^2 dx \\ &+ \frac{1}{3!}f'''_{i-\frac{1}{2}} \int_{x_{i-1}}^{x_i} (x - x_{i-\frac{1}{2}})^3 dx + \cdots\end{aligned}$$

Rectangular Method–Error Estimation

Thus,

$$\begin{aligned}\Delta I_{[x_{i-1},x_i]} &= \int_{x_{i-1}}^{x_i} f(x)dx - \int_{x_{i-1}}^{x_i} f_{i-\frac{1}{2}}dx \\ &= \int_{x_{i-1}}^{x_i} f(x)dx - hf_{i-\frac{1}{2}} \\ &\approx \frac{1}{2}f''_{i-\frac{1}{2}}\int_{x_{i-1}}^{x_i}(x - x_{i-\frac{1}{2}})^2dx\end{aligned}$$

For over all interval $[a, b]$

$$\Delta I_{[a,b]} = \int_a^b f(x)dx - I_{[a,b]} \approx \frac{b-a}{24}h^2 f''(\xi) = \frac{(b-a)^3}{24N^2}f''(\xi)$$

where $f''(\xi)$ is the average value of the second derivative of $f(x)$

Rectangular Method–Example I

$$\int_0^2 (x^4 - 2x + 1)dx = \left[\frac{1}{5}x^5 - x^2 + x\right]_0^2 = 4.4$$

```
from numpy import linspace
from pylab import plot,show,ylim,xlabel,ylabel,bar

def f(x):
    y=x**4-2*x+1
    return y

N=10
a=0.0
b=2.0
h=(b-a)/N

s=0.0
for i in range(0,N):
   s+=f(a+h/2)
   a+=h

print(s*h)
```

Rectangular Method–Example II

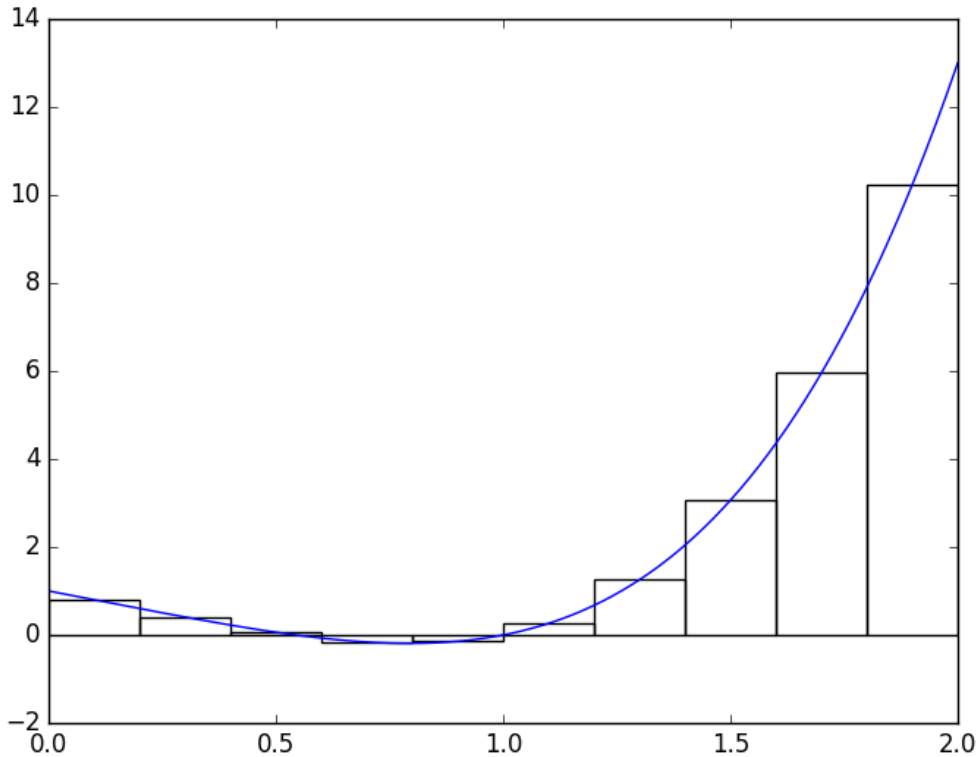
With plot

```
from numpy import linspace
from pylab import plot,show,xlim,xlabel,ylabel,bar

def f(x):
    y=x**4-2*x+1
    return y

def Integrate(x0,xe,n):
  s=0.0
  x=x0
  h=(xe-x0)/n
  for i in range(0,n):
    tmp=f(x+h/2.0)
    s+=tmp
    fbar.append(tmp)
    xi.append(x+h/2.0)
    x+=h
  return(h*s,h)

fbar=[]
xi=[]

N_f=1000
```

Rectangular Method–Example III

```
px=[]
py=[]
for xx in linspace(0.0,2.0,N_f):
  py.append(f(xx))
  px.append(xx)
Int_res,dx=Integrate(0,2.0,10)
print(Int_res)
plot(px,py)
bar(xi,fbar,dx,facecolor='None',edgecolor='k')
xlim(0,2)
show()

del fbar[:]
del xi[:]
Int_res,dx=Integrate(0,2.0,100)
print(Int_res)
plot(px,py)
bar(xi,fbar,dx,facecolor='None',edgecolor='k')
xlim(0,2)
show()
```

Rectangular Method–Example IV



Rectangular Method–Example2 I

$$\int_0^{\pi} \sin(x)dx$$

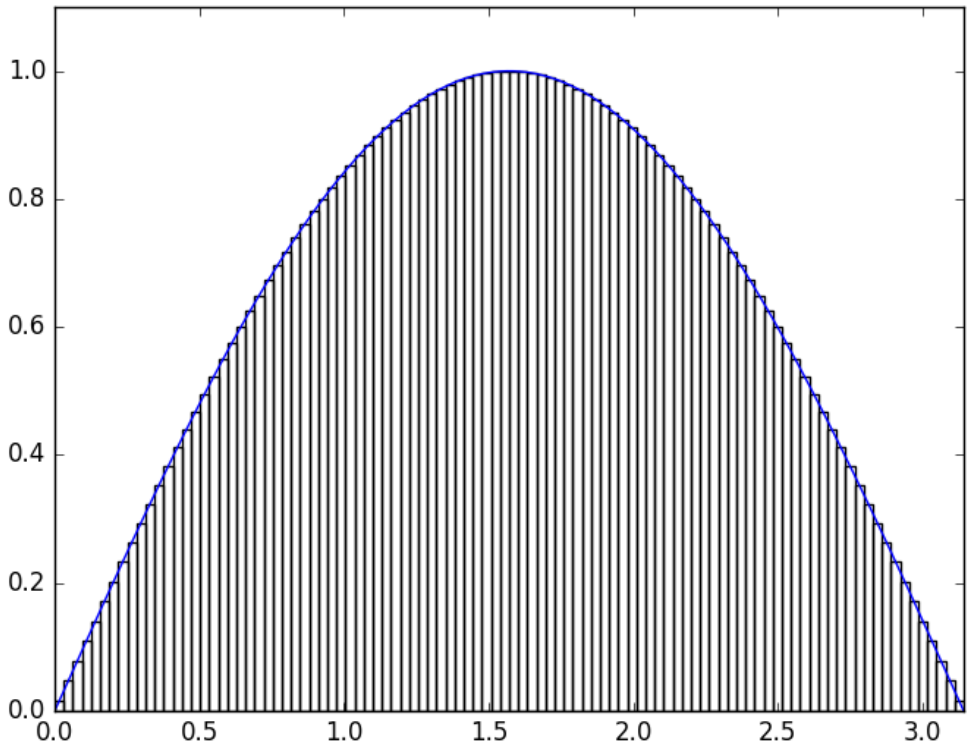
```
from math import pi,sin
from numpy import linspace
from pylab import plot,show,xlim,ylim,bar

def f(x):
    y=sin(x)
    return y

def Integrate(x0,xe,n):
  s=0.0
  x=x0
  h=(xe-x0)/n
  for i in range(0,n):
    tmp=f(x+h/2.0)
    s+=tmp
    fbar.append(tmp)
    xi.append(x+h/2)
    x+=h
  return(h*s,h)
```

Rectangular Method–Example2 II

```
fbar=[]
xi=[]

N_f=1000
px=[]
py=[]
for xx in linspace(0.0,pi,N_f):
  py.append(f(xx))
  px.append(xx)
Int_res,dx=Integrate(0,pi,10)
print(Int_res)
plot(px,py)
bar(xi,fbar,dx,facecolor='None',edgecolor='k')
ylim(0,1.1)
xlim(0,pi)
show()

del fbar[:]
del xi[:]
Int_res,dx=Integrate(0,pi,100)
print(Int_res)
plot(px,py)
bar(xi,fbar,dx,facecolor='None',edgecolor='k')
ylim(0,1.1)
xlim(0,pi)
```

Rectangular Method–Example2 III

```
show()
```



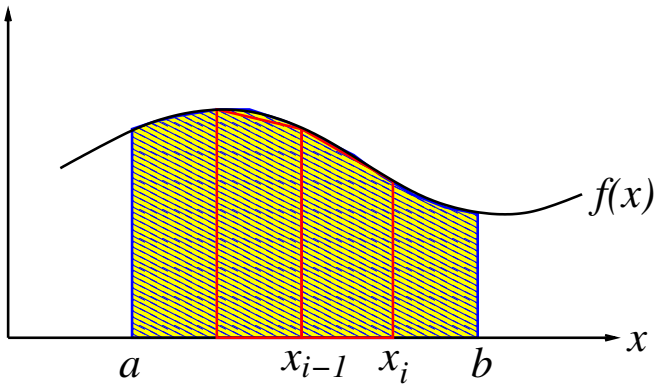
Homework

Integrate

$$f(x) = \exp(x)$$

over the interval $x \in [0, 2.5]$ using the rectangular method.

Trapezoidal Method



- Simply replace the rectangles by trapezoids.
- Or simply from the Taylor series ,Eq. (1),

$$f(x) = f(x_i) + (x - x_i)f'(x_i) + \mathcal{O}(h^2) \tag{14}$$

- Replace $f'(x_i)$ by $\frac{f(x_{i+1})-f(x_i)}{h}$, then

$$f(x) \simeq f(x_i) + (x - x_i)\frac{f(x_{i+1}) - f(x_i)}{h} \tag{15}$$

Integrate over every interval with this linear function to obtain

Trapezoidal Method

$$I_{[a,b]} = \frac{h}{2}\sum_{i=0}^{N-1}(f(x_i) + f(x_{i+1})) + \mathcal{O}(h^2) \tag{16}$$

Trapezoidal Method–Example I

$$\int_0^2 (x^4 - 2x + 1)dx = \left[\frac{1}{5}x^5 - x^2 + x\right]_0^2 = 4.4$$

```
from numpy import linspace
from pylab import plot,show,ylim,xlabel,ylabel,bar

def f(x):
    y=x**4-2*x+1
    return y

N=10
a=0.0
b=2.0
h=(b-a)/N

s=0.5*(f(a)+f(b))
for i in range(1,N):
   s+=f(a+h)
   a+=h

print(s*h)
```

Trapezoidal Method–Example II

With plot

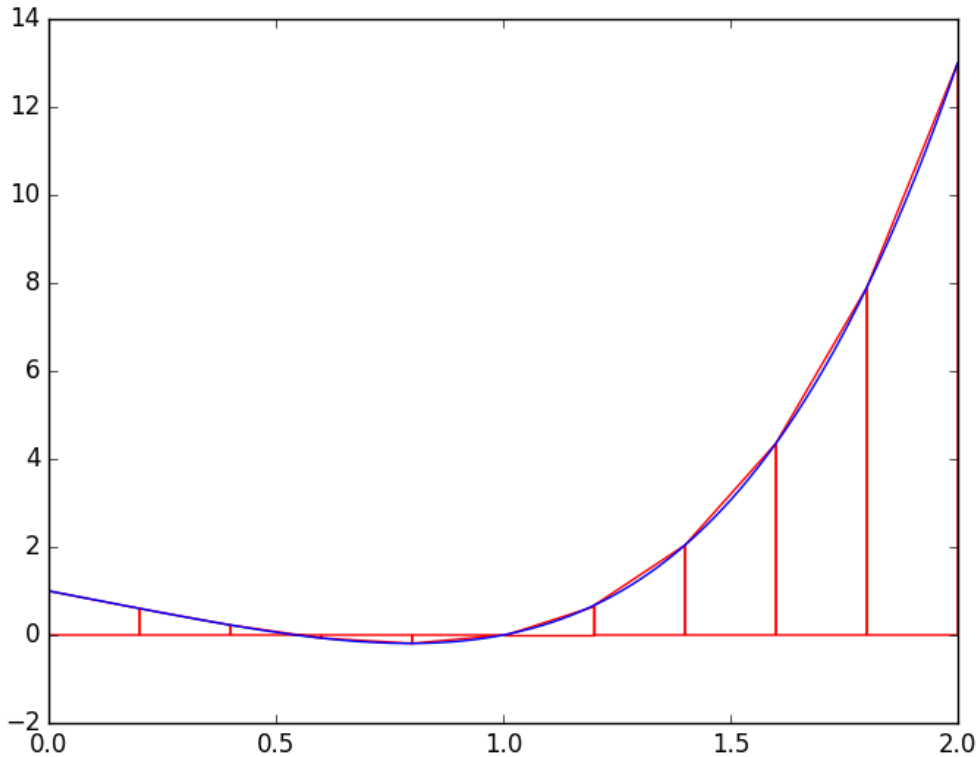
```
from numpy import linspace,array
from matplotlib import pyplot as plt
from matplotlib.patches import Polygon

def f(x):
    y=x**4-2*x+1
    return y

def Integrate(x0,xe,n):
  s=0.0
  x=x0
  h=(xe-x0)/n
  for i in range(0,n):
    tmp=f(x+h/2.0)
    ##############################
    #  To draw a trapezoid
    ##############################
    points=[[x,0.0],[x+h,0.0],[x+h,f(x+h)],[x,f(x)]]
    ax1.add_patch(Polygon(points,fill=False,edgecolor='r'))
    s+=tmp
    x+=h
  return(h*s,h)
```

Trapezoidal Method–Example III

```
# ##########################
# To draw a trapezoid
# ##########################
fig=plt.figure()
ax1=fig.add_subplot(111)
# ##########################

N_f=1000
px=[]
py=[]
for xx in linspace(0.0,2.0,N_f):
   py.append(f(xx))
   px.append(xx)
Int_res,dx=Integrate(0,2.0,10)
print(Int_res)
plt.plot(px,py)
plt.xlim(0,2)
plt.show()

fig=plt.figure()
ax1=fig.add_subplot(111)
Int_res,dx=Integrate(0,2.0,100)
print(Int_res)
plt.plot(px,py)
plt.xlim(0,2)
```

Trapezoidal Method–Example IV

```
plt.show()
```



Trapezoidal Method–Example2 I

$$\int_0^{\pi} \sin(x)dx$$

```
from math import sin,pi
from numpy import linspace,array
from matplotlib import pyplot as plt
from matplotlib.patches import Polygon

def f(x):
    y=sin(x)
    return y

def Integrate(x0,xe,n):
  s=0.0
  x=x0
  h=(xe-x0)/n
  for i in range(0,n):
    tmp=f(x+h/2.0)
    ##############################
    #  To draw a trapezoid
    ##############################
    points=[[x,0.0],[x+h,0.0],[x+h,f(x+h)],[x,f(x)]]
    ax1.add_patch(Polygon(points,fill=False,edgecolor='r'))
```

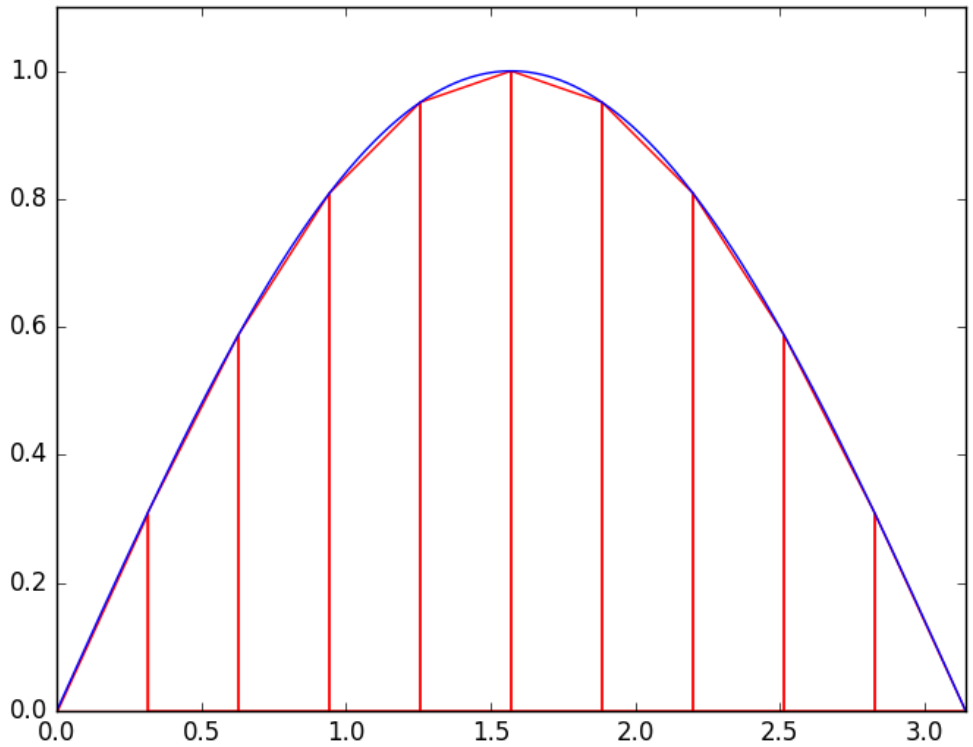
Trapezoidal Method–Example2 II

```
        s+=tmp
        x+=h
    return(h*s,h)

# ############################
# To draw a trapezoid
# ############################
fig=plt.figure()
ax1=fig.add_subplot(111)
# ############################

N_f=1000
px=[]
py=[]
for xx in linspace(0.0,pi,N_f):
    py.append(f(xx))
    px.append(xx)
Int_res,dx=Integrate(0,pi,10)
print(Int_res)
plt.plot(px,py)
plt.ylim(0.0,1.1)
plt.xlim(0,pi)
plt.show()

fig=plt.figure()
```

Trapezoidal Method–Example2 III

```
ax1=fig.add_subplot(111)
Int_res,dx=Integrate(0,pi,100)
print(Int_res)
plt.plot(px,py)
plt.ylim(0.0,1.1)
plt.xlim(0,pi)
plt.show()
```

Trapezoidal Method–Example2 IV



Homework

Integrate

$$f(x) = \exp(x)$$

over the interval $x \in [0, 2.5]$ using the trapezoidal method.

Simplson's Rule

If all interval have the same length, i.e.,

$$x_{i+1} - x_i = h = const., \qquad \forall i,$$

to improve the accuracy (from the Taylor series (Eq. (1)))

$$\begin{aligned} I_{[x_{i-1},x_{i+1}]} &= \int_{x_{i-1}}^{x_{i+1}} f(x)dx \\ &= f_i \int_{x_{i-1}}^{x_{i+1}} dx + f' \int_{x_{i-1}}^{x_{i+1}} (x - x_i)dx + \frac{1}{2} f_i'' \int_{x_{i-1}}^{x_{i+1}} (x - x_i)^2 dx \\ &\quad + \frac{1}{3} f_i''' \int_{x_{i-1}}^{x_{i+1}} (x - x_i)^3 dx \\ &= 2hf_i + 0 + \frac{1}{2} f_i'' \frac{2}{3} h^3 + 0 + \mathcal{O}(h^5) \end{aligned} \tag{17}$$

Simplson's Rule

Now use the discretized second order derivative (Eq. (10))

$$f_i'' = \left.\frac{d^2 f(x)}{dx^2}\right|_{x=x_i} = \frac{1}{h^2}\left(f_{i+1} - 2f_i + f_{i-1}\right).$$

Then Eq. (17) can be rewritten as

$$\begin{aligned} I_{[x_{i-1},x_{i+1}]} &= 2hf_i + \frac{1}{3}h\left(f_{i+1} - 2f_i + f_{i-1}\right) + \mathcal{O}(h^5) \\ &= \frac{h}{3}\left(f_{i+1} + 4f_i + f_{i-1}\right) + \mathcal{O}(h^5) \end{aligned} \tag{18}$$

Simpson's Rule

$$I_{[a,b]} = \sum_{i=0}^{N/2-1} \frac{h}{3}\left(f_{2i} + 4f_{2i+1} + f_{2i+2}\right) \tag{19}$$

Simpson Rule–Example

$$\int_0^{\pi} \sin(x)dx$$

```
from math import sin,pi
from numpy import linspace

def f(x):
  y=sin(x)
  return y

def simpson13(x0,xe,num_int):
  x=linspace(x0,xe,num_int)
  h=x[1]-x[0]
  i=0
  integral=0.0
  i_max=x.size-2
  while i<i_max:
    integral+=(f(x[i])+4.0*f(x[i+1])+f(x[i+2]))
    i+=2
  return (integral*h/3.0)

integ=simpson13(0.0,pi,1000)
print("Result=",integ)
```

Simpson's $3/8$ Rule

Simpson's $3/8$ Rule (Newton-Cotes formula with $n = 3$)

$$\int_a^b f(x)dx = \sum_{i=1}^{N/3} \frac{3h}{8} \left(f_i + 3f_{i+1} + 3f_{i+2} + f_{i+3}\right) \tag{20}$$

Verification of Simpson's $3/8$ rule: Let an integral of any function $f(x)$ over an interval $[a, a+3h]$ can be approximated as

$$\int_a^{a+3h} f(x)dx \approx c_0 f(a) + c_1 f(a+h) + c_2 f(a+2h) + c_3 f(a+3h)$$

Replace x by $x - a - \frac{3h}{2}$ then

$$\int_{-\frac{3h}{2}}^{\frac{3h}{2}} f(x)dx = c_0 f\left(-\frac{3h}{2}\right) + c_1 f\left(-\frac{h}{2}\right) + c_2 f\left(\frac{h}{2}\right) + c_3 f\left(\frac{3h}{2}\right) \tag{21}$$

Simpson's 3/8 Rule

Since there are four unknowns, c_0, c_1, c_2, c_3, we need four equations to obtain the unknown parameters.
Since Eq. (21) is satisfied by any function, we consider four different types of $f(x)$: $f(x) = 1$, $f(x) = x$, $f(x) = x^2$, and $f(x) = x^3$ for simplicity.

$$\int_{-\frac{3h}{2}}^{\frac{3h}{2}} dx = 3h = c_0 + c_1 + c_2 + c_3 \tag{22}$$

$$\int_{-\frac{3h}{2}}^{\frac{3h}{2}} x dx = 0 = c_0\left(-\frac{3h}{2}\right) + c_1\left(-\frac{h}{2}\right) + c_2\left(\frac{h}{2}\right) + c_3\left(\frac{3h}{2}\right) \tag{23}$$

$$\int_{-\frac{3h}{2}}^{\frac{3h}{2}} x^2 dx = \frac{9h^3}{4} = c_0\left(\frac{9h^2}{4}\right) + c_1\left(\frac{h^2}{4}\right) + c_2\left(\frac{h^2}{4}\right) + c_3\left(\frac{9h^2}{4}\right) \tag{24}$$

$$\int_{-\frac{3h}{2}}^{\frac{3h}{2}} x^3 dx = 0 = c_0\left(-\frac{27h^3}{8}\right) + c_1\left(-\frac{h^3}{8}\right) + c_2\left(\frac{h^3}{8}\right) + c_3\left(\frac{27h^3}{8}\right) \tag{25}$$

Simpson's 3/8 Rule

From Eqs. (22)-(25), we obtain

$$c_0 = c_3 = \frac{3h}{8}$$

and

$$c_1 = c_2 = \frac{9h}{8}$$

Thus,

$$\int_{-\frac{3h}{2}}^{\frac{3h}{2}} f(x)dx \approx \frac{3h}{8}\left[f\left(-\frac{3h}{2}\right) + 3f\left(-\frac{h}{2}\right) + 3f\left(\frac{h}{2}\right) + f\left(\frac{3h}{2}\right)\right]$$

or equivalently

$$\int_{a}^{a+3h} f(x)dx \approx \frac{3h}{8}\left[f\left(a\right) + 3f\left(a+h\right) + 3f\left(a+2h\right) + f\left(a+3h\right)\right] \qquad (26)$$

By summing over all interval, we obtain Eq. (20).

Simpson 3/8 Rule–Example

$$\int_0^{\pi} \sin(x)dx$$

```
from math import sin,pi
from numpy import linspace

def f(x):
  y=sin(x)
  return y

def simpson38(x0,xe,num_int):
  x=linspace(x0,xe,num_int)
  h=x[1]-x[0]
  i=0
  integral=0.0
  i_max=x.size-3
  while i<i_max:
    integral+=(f(x[i])+3.0*f(x[i+1])+3.0*f(x[i+2])+f(x[i+3]))
    i+=3
  return (integral*h*3.0/8.0)

integ=simpson38(0.0,pi,999)
print("Result=",integ)
```

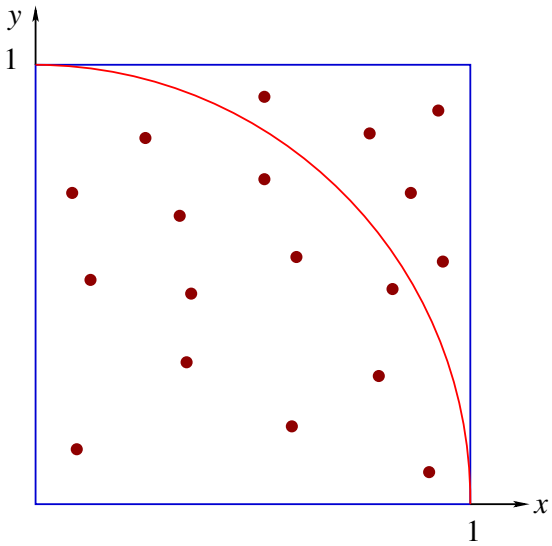
Homework

Integrate

$$f(x) = \exp(x)$$

over the interval $x \in [0, 2.5]$ using the Simpson's two rules.

Monte Carlo Integration



- Find π.
- Use the area of the unit quarter circle.
- Area of the blue square: $A = 1$.
- Area of the quarter circle: $A' = \pi/4$
- The probability that a dart lands in any particular region is proportional to the area of that region.

1. Generate a pair of random numbers x and y $(0 \le x \le 1$ and $0 \le y \le 1)$
2. If $y \le \sqrt{1 - x^2}$ then increase N_{circle} by one.
3. Repeat process (1) and (2) for N times.
4. Calculate the probability $p = N_{circle}/N$.
5. From the relation $p = \frac{\frac{\pi}{4}}{1} = \frac{N_{circle}}{N}$, we obtain $\pi = 4N_{circle}/N$.

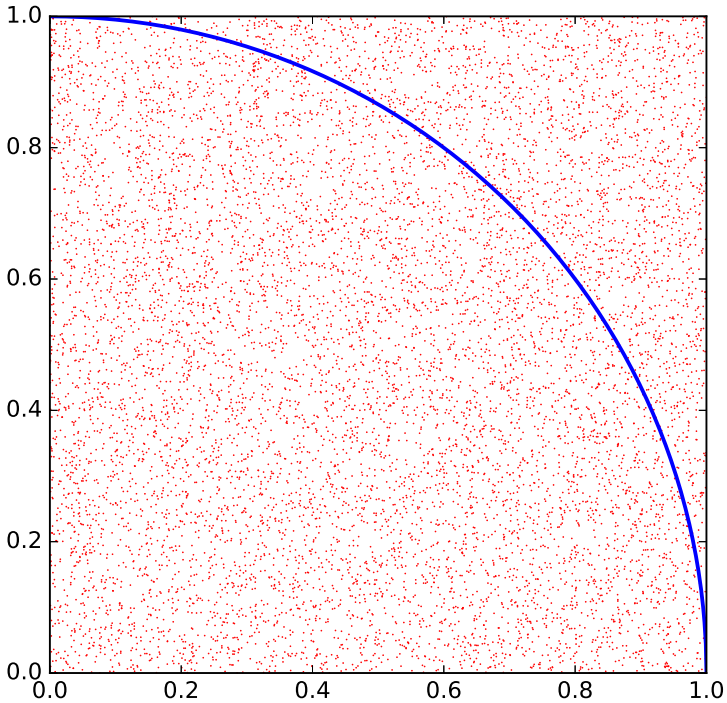
Finding π I

```
from random import random,seed
from numpy import empty,linspace
from math import sqrt
from matplotlib import pyplot as plt

np=10000      #number of random ponts
seed()
x=empty(np,float)
y=empty(np,float)
i=0
ncircle=0.0
while i<np:
   x[i]=random()
   y[i]=random()
   if(y[i]<sqrt(1.0-x[i]**2)):
      ncircle+=1.0
   i+=1

print("pi=",4.0*ncircle/float(np))

# ############################
```

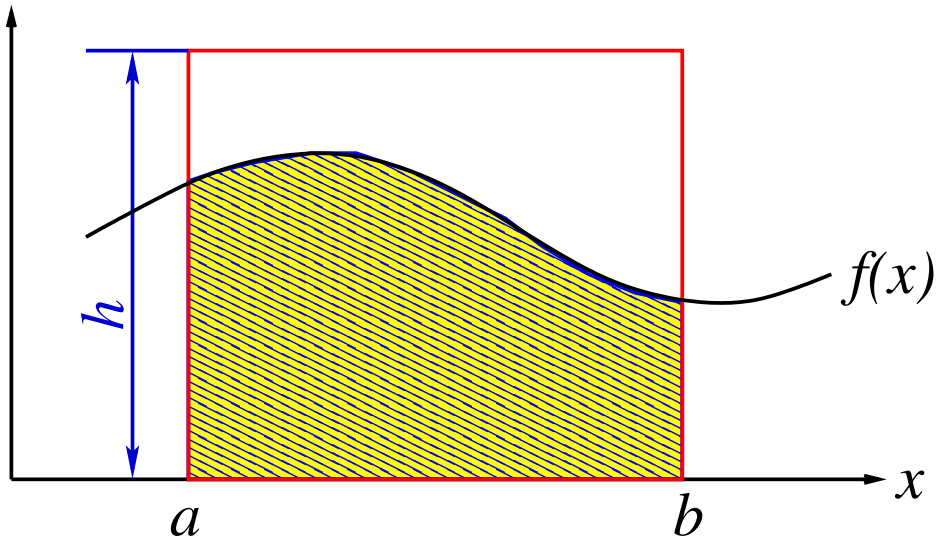
Finding π II

```
# Just for plotting
# ##########################
np=1000
px=linspace(0,1,np)
py=empty(np,float)
i=0
for xx in px:
   py[i]=sqrt(1.0-xx**2)
   i+=1

plt.plot(x,y,'r.',markersize=0.7)
plt.plot(px,py,linewidth=2)
ax=plt.gca()
ax.set_aspect('equal')

plt.show()
```

Finding π III



Monte Carlo Integral



- The most basic concept of Monte Carlo method is the important sampling.
- sample the value of $f(x)$

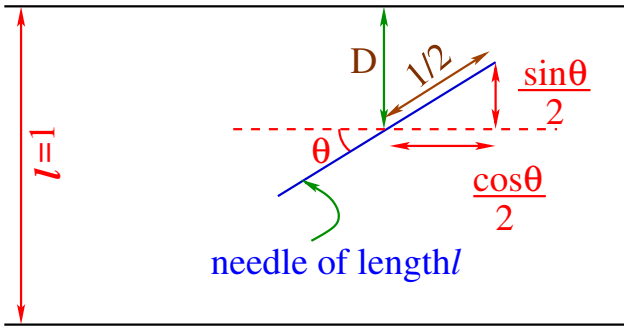
$$\int_a^b f(x)dx = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \quad \text{for large } N \tag{27}$$

MC Integral

$$\int_0^{\pi} \sin(x)dx$$

```
from random import random,seed
from math import sin,pi

np=10000      #number of random ponts
seed()
i=0
inte=0.0
xi=0.0
xf=pi
while i<np:
   x=pi*random()
   inte+=sin(x)
   i+=1

print("Int_0^pi sin(x)dx=",inte*(xf-xi)/np)
```

Monte Carlo Method – Buffon's Needle



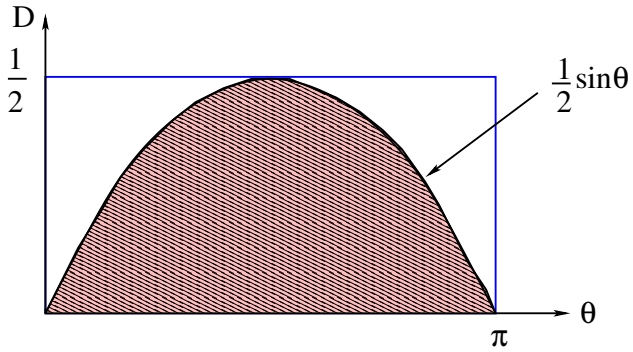
- Throw a needle to locate at a random position.
- Use the probability that the needle touch or cross the lines.
- Two random variables: D and θ.

$$0 \le D \le \frac{1}{2}, \qquad 0 \le \theta \le \pi \tag{28}$$

Monte Carlo Method – Buffon's Needle

Condition that the needle touch or cross the black lines:

$$D \leq \frac{1}{2}\sin\theta \tag{29}$$



Then we just obtain the area of the shaded region.

1. Generate two random numbers $0 \leq D \leq 1/2$ and $0 \leq \theta \leq \pi$.
2. If $D \leq \frac{1}{2}\sin\theta$ then increase N_{count} by 1.
3. Repeat (1) and (2) by N times.

Then the area of the shaded region becomes $A = \int_0^\pi \frac{1}{2}\sin\theta = 1$. Thus,

$$p = \frac{N_{count}}{N} = \frac{1}{\frac{\pi}{2}} \Rightarrow \pi = \frac{2}{p}$$

Simulation of Buffon's Needle I

```
from random import random,seed
from numpy import empty
from math import sin,cos,pi
from matplotlib import pyplot as plt

def plot_needles(n_needle):
    plt.figure(figsize=(14,7))
    Lmin=0.0
    Lmax=3.0
    topline_x=[Lmin,Lmax]
    topline_y=[1.0,1.0]
    bottomline_x=[Lmin,Lmax]
    bottomline_y=[0,0]
    i=0
    while i<2:
        plt.plot(topline_x,topline_y,'b',linewidth=3)
        plt.plot(bottomline_x,bottomline_y,'b',linewidth=3)
        i+=1

    center_x=empty(n_needle,float)
    i=0
    while i<n_needle:
        center_x[i]=random()*Lmax
        x=[center_x[i]-cos(theta[i])*0.5,center_x[i]+cos(theta[i])*0.5]
        y=[center_y[i]-sin(theta[i])*0.5,center_y[i]+sin(theta[i])*0.5]
```

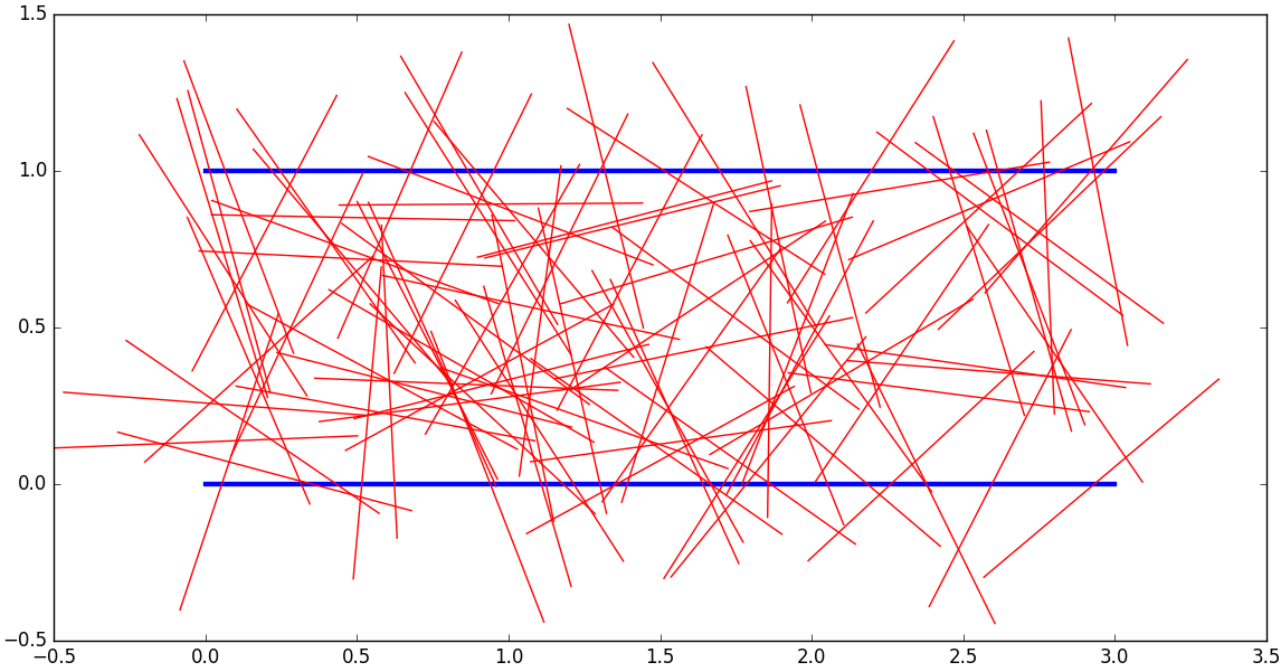
Simulation of Buffon's Needle II

```
      plt.plot(x,y,'r')
      i+=1
   plt.xlim(-0.5,3.5)
   plt.ylim(-0.5,1.5)

   ax=plt.gca()
   plt.show()

n=[10,100,500,1000] # array for the number of needles
#n=[10]
for np in n:
   seed()
   theta=empty(np,float)
   center_y=empty(np,float)
   D=empty(np,float)
   i=0

   ncount=0.0
   while i<np:
      theta[i]=pi*random()
      center_y[i]=random()
      if center_y[i]>0.5:
        D[i]=1.0-center_y[i]
      else:
        D[i]=center_y[i]
```

Simulation of Buffon's Needle III

```
    if D[i]<=sin(theta[i])/2.0:
      ncount+=1.0
    i+=1
  p=ncount/np
  print("pi=",2.0/p)

##################################
#   For plotting needles
##################################
  plot_needles(np)
```

Simulation of Buffon's Needle IV



Multidimensional Integral

- Basic idea: extension of the algorithms for single variable functions
- Monte Carlo Method: more efficient!

$$\int_a^b \int_c^d \int_e^f F(x,y,z)dxdydz = \sum_{i=1}^{N} F(x_i,y_i,z_i)\Delta v$$

where $\Delta v = \frac{(b-a)(d-c)(f-e)}{N}$ for large N.

Homework

Integrate

$$f(x, y, z) = x + y + z$$

over the interval $x \in [0, 1]$, $y \in [0, 1]$, and $z \in [0, 1]$ using the Monte Carlo method.

Improper Integral

- Types of improper integral
  - range of integral $\rightarrow$ infinite
  - Integrand contains a singularity within the integration range
    - integrable singularity: $\int_0^1 \frac{1}{\sqrt{1-x^2}} dx$
    - nonitegrable singularity: $\int_0^1 \frac{dx}{x}$
- Some special case:

$$\lim_{x \to 0} \frac{\sin x}{x} = 1$$

Take small ϵ then

$$\int_0^\pi \frac{\sin x}{x} dx = \epsilon + \int_\epsilon^\pi \frac{\sin x}{x} dx$$

Infinite range integration: in some cases

$$I = \int_0^b \exp(-x^2) \rightarrow \frac{\sqrt{pi}}{2} = 0.886227 \text{ as } b \rightarrow \infty$$

for $b = 3$, $I = 0.886207 \Rightarrow$ error might be reasonably small.