# 第三章 插值、微分与积分

苏湘宁　　邮箱：suxn@hainanu.edu.cn

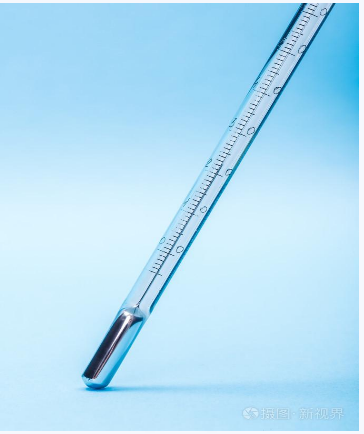
# 插值、微分与积分

- 物理问题：探测器刻度
- 拉格朗日插值
- 三次样条插值
- 数值微分与比热计算
- 定积分的近似计算

# 3.1物理问题：探测器刻度

任何介质,若其某一属性随待测物理量呈单调变化,则都可以用来制作探测仪器。使用前需要做仪器刻度。

酒精温度计：
-113.5℃～78.4℃



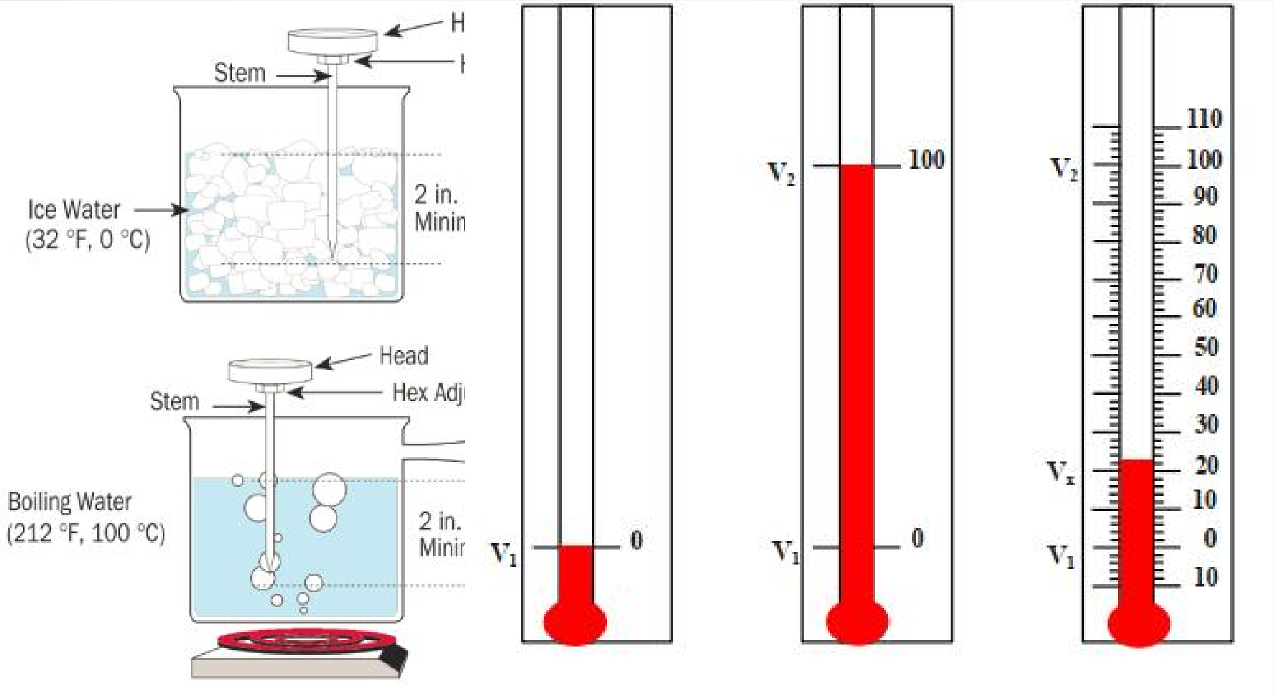
水银温度计：
-30℃～300℃


煤油温度计：
30℃～150℃

**数值插值**是仪器刻度中用到的最主要算法。

# 3.1物理问题：探测器刻度

将液面vl和v2之差等分为 100 份，每一份为 1.0 摄氏度,完成温度计的刻度定标。




数学语言:
通过(vl,t1)和(v2.t2)两个已知数据点的两点式直线方程为

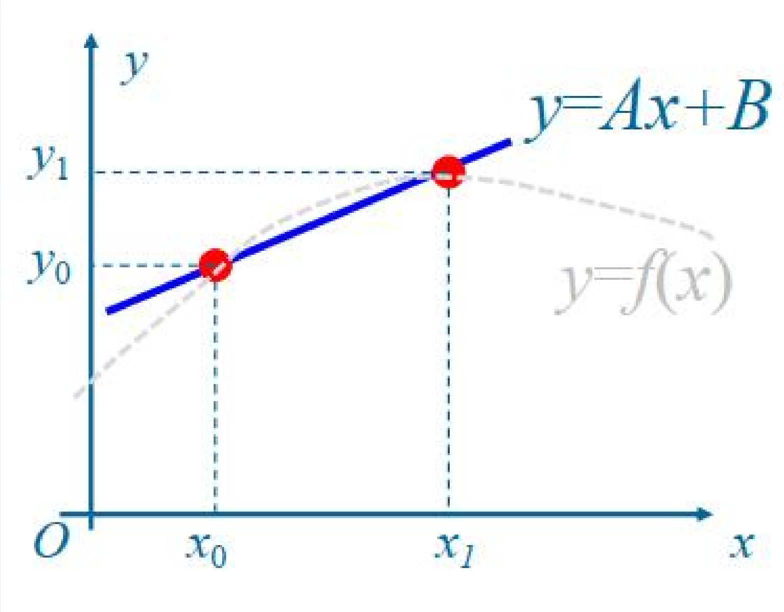
$$t(v) = av + b$$

$$t(v) = \frac{v - v1}{v2 - v1} \times 100 + \frac{v - v2}{v1 - v2} \times 0$$

$$t(v) = \frac{v - v1}{v2 - v1} \times t2 + \frac{v - v2}{v1 - v2} \times t1$$

**构建解析函数表达式，使其通过已知数据点的过程叫做插值，是基本的数值计算方法。**

## 3.2插值问题：拉格朗日插值

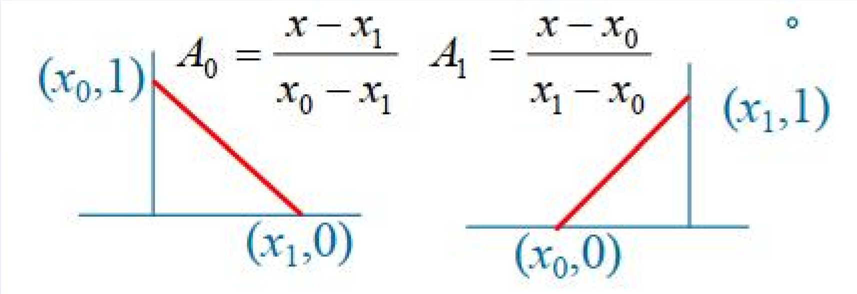



两点式直线方程:

$$y(x)=\frac{x-x1}{x0-x1}y0+\frac{x-x0}{x1-x0}y1$$

$$y(x)=A0(x)y0+A1(x)y1$$

**插值函数y(x)可以看作是A0(x)和A1(x)的线性组合**




x0，x1插值基函数：

$$A0=\frac{x-x1}{x0-x1},\quad A1\frac{x-x0}{x1-x0}$$

**将插值函数表示为基函数线形组合的插值方法称为拉格朗日插值法。**

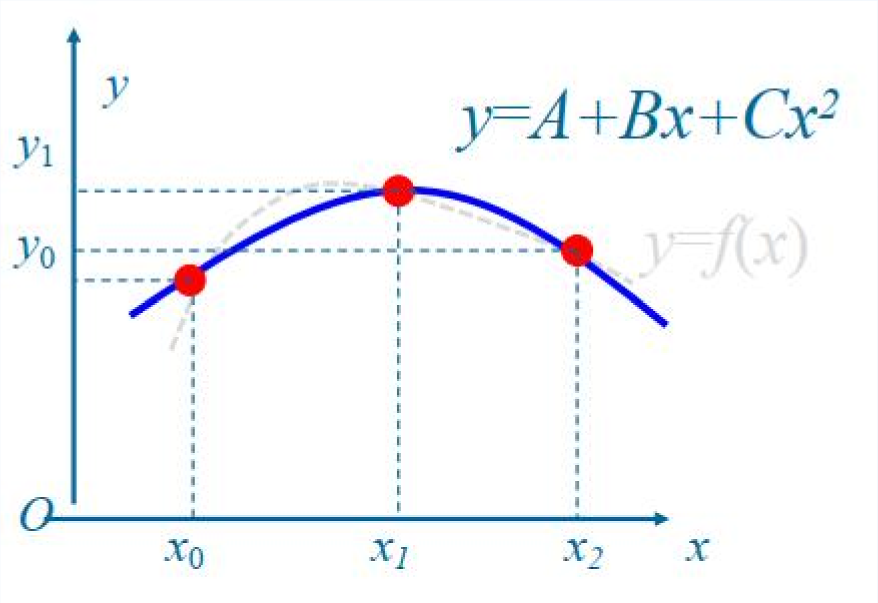
# 3.2插值问题：拉格朗日插值

已知（$x1, y1$），（$x2, y2$）和（$x2, y2$）三个点的二次插值函数:

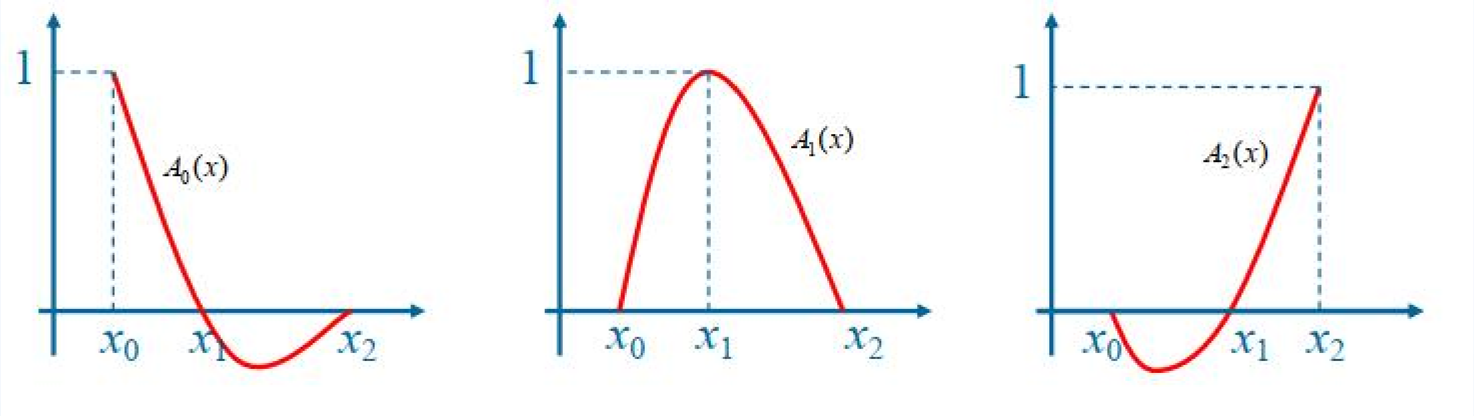
$$A_0=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$$

$$A_1=\frac{(x-x_2)(x-x_0)}{(x_1-x_2)(x_1-x_0)}$$

$$A_2=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$




$$\boldsymbol{y(x)=A0y0+A1y1+A2y2}$$



# 3.2插值问题：拉格朗日插值

## $n$次插值：

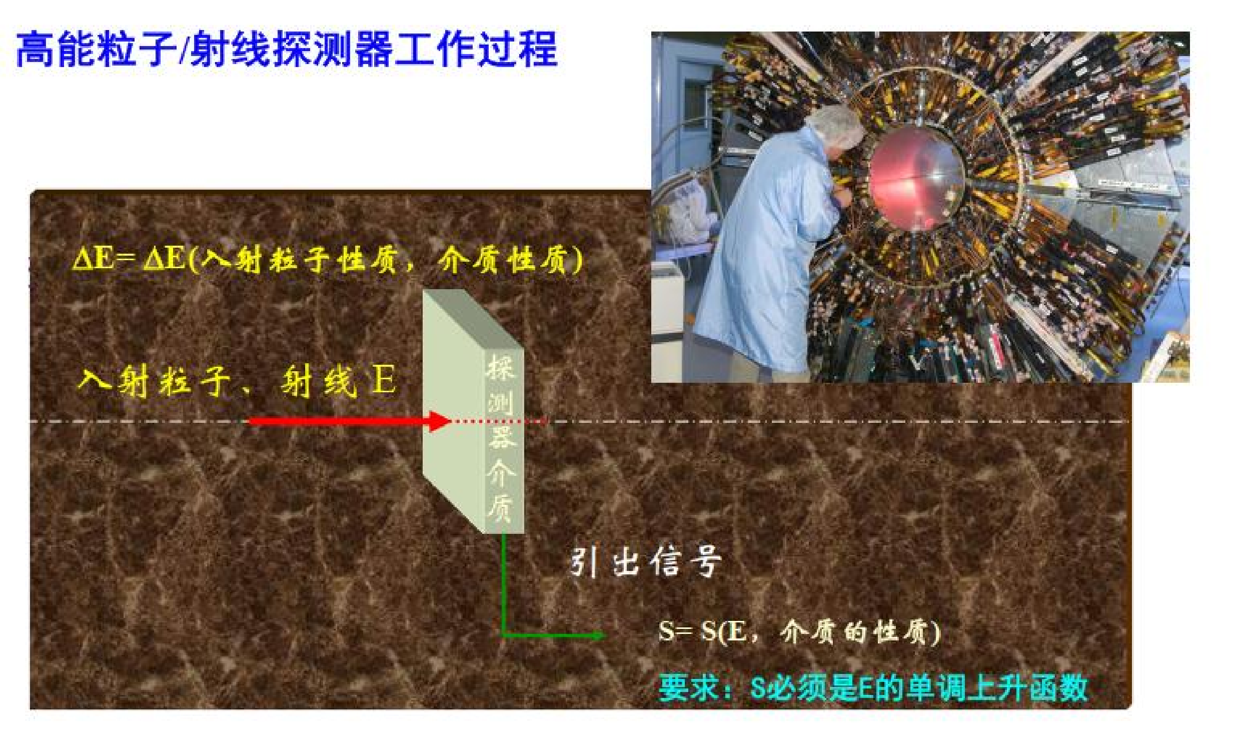
以上的拉格朗日插值可以推广到$n$次插值：

$$y(x)=\sum_{j=0}^{n}A_j(x)y_j$$

其中插值基函数可表示为：

$$A_j(x)=\prod_{\substack{i=0\\ i\neq j}}^{n}\frac{x-x_i}{x_j-x_i}$$

# 3.2插值问题：拉格朗日插值应用




基本原理：

当带电粒子射入探测器介质后，所经之处硅原子会发生电离，产生电子-空穴对。

在高压电源产生的强电场驱动下，电子向正电极移动并产生感应信号，被电子学元件收集并放大后，记录信号幅度。

能量越高，产生的电子-空穴对越多，信号幅度越大。

粒子探测问题中：测量所得是信号幅度,由信号幅度反推出粒子能量就需要对探测器进行刻度定标，构建信号幅度和粒子能量之间的解析函数关系式。

## 3.2插值问题：拉格朗日插值应用

问题2-1：

使用包括 Am241，Cm244，Pu239的三组份α源产生的5种能量的α粒子，对高纯硅探测器进行能量刻度，得到的信号幅度见表 2.1 所示。若用该探测器设备探测未知能量的带电粒子，测得的信号幅度为 1588mV, 请计算该粒子的能量为多少 kev?

表2.1.三组份a源产生的α粒子能量以及其沉积在探测器上所产生的信号幅度。

| 能量E(keV) | 5156 | 5440 | 5486 | 5763 | 5805 |
|---|---|---|---|---|---|
| 幅度S(mV) | 1390 | 1455 | 1561 | 1640 | 1705 |

## 问题2-1编程实操：

### （1）导入模块，读取数据

```python
import numpy as np
import matplotlib.pylab as pl

#读取数据'calibrate.dat'
data=np.loadtxt('calibrate.dat')
energy_alpha_source=[5156,5440,5486,5763,5805]#data[:,0]
signal_alpha_source=[1390,1455,1561,1640,1705]#data[:,1]
####
```

# 3.2插值问题：拉格朗日插值应用

问题2-1编程实操：

（2）定义了一个子函数 lagr_poly，构建出拉格朗日插值函数，并返回函数值

```
#定义了一个子函数 lagr_poly，构建出拉格朗日插值函数，并返回函数值
def lagr_poly(x0,y0,n,x):
    y=0.0
    for i in range(0,n):
        p=1.0
        for j in range(0,n):
            if(i!=j):
                p=p*(x-x0[j])/(x0[i]-x0[j])
        y=y+p*y0[i]
    return y
######
```

# 3.2插值问题：拉格朗日插值应用

问题2-1编程实操：

（3）调用子函数 lagr poly，计算出拉格朗日插值函数在(1300，1800)区间的数值。调用子函数 lagr poly，计算出信号幅度为1588mV的待测粒子能量。

```
#调用子函数 lagr poly，计算出拉格朗日插值函数在(1300，1800)区间的数值。
signal=np.arange(1300,1800,1)
energy=[]

for x in signal:
    xenergy=lagr_poly(signal_alpha_source,energy_alpha_source,5,x)
    energy.append(xenergy)
######

#调用子函数 lagr poly，计算出信号幅度为1588mV的待测粒子能量。
print('enegy of particle:%8.3fkeV'%lagr_poly(signal_alpha_source,\
              energy_alpha_source,5,1588))
```

# 3.2插值问题：拉格朗日插值应用

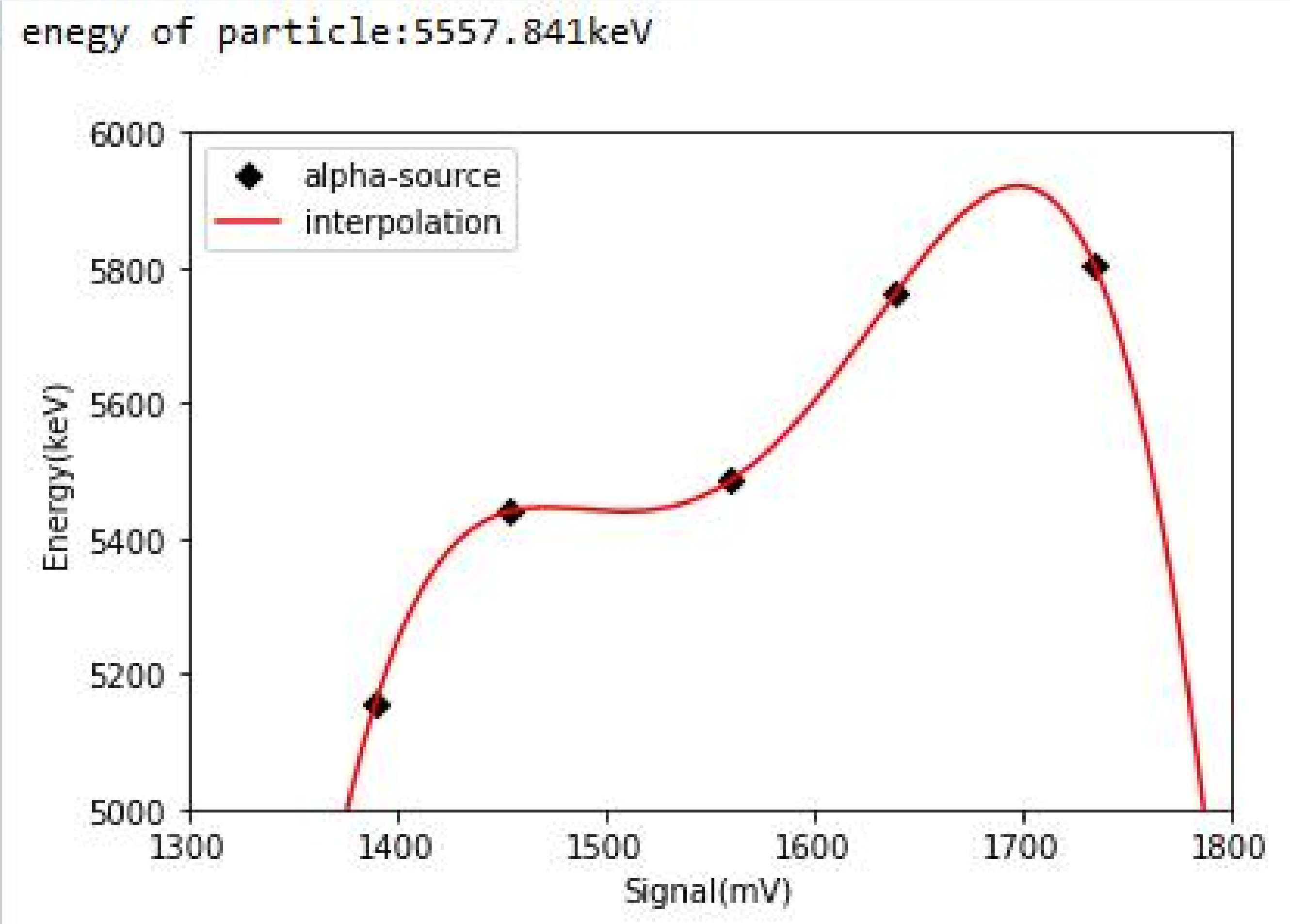
问题2-1编程实操：

（4）将结果用图形表现出来：

```
# 将结果用图形表现出来
pl.plot(signal_alpha_source,energy_alpha_source,'kD',label='alpha-source')
pl.plot(signal,energy,'r-',label='interpolation')
pl.xlabel('Signal(mV)')
pl.ylabel('Energy(keV)')
pl.xlim(1300,1800)
pl.ylim(5000,6000)
pl.legend(loc='upper left')
pl.show()
####
```

# 3.2插值问题：拉格朗日插值应用

问题2-1结果展示：



## 3.2插值问题：拉格朗日插值应用

问题2-1编程实操2.0：

（2）定义了一个子函数 lagr_poly，构建出拉格朗日插值函数，并返回函数值

直接调用 scipy 程序库中的 interpolate 模块来实现拉格朗日插值函数的构建。

```
#定义了一个子函数 lagr_poly，构建出拉格朗日插值函数，并返回函数值
import numpy as np
import matplotlib.pylab as pl
from scipy import interpolate


#调用 scipy 程序库中的 interpolate 模块
poly = interpolate.lagrange(signal_alpha_source, energy_alpha_source) #
#####
```

# 3.2插值问题：拉格朗日插值应用

问题2-1编程实操2.0：

（3）调用子函数 lagr poly，计算出拉格朗日插值函数在(1300，1800)区间的数值。调用子函数 lagr poly，计算出信号幅度为1588mV的待测粒子能量。

```python
#调用子函数 lagr poly，计算出拉格朗日插值函数在(1300，1800)区间的数值。
signal=np.arange(1300,1800,1)
energy=[]

for x in signal:
    xenergy=poly(x)
    energy.append(xenergy)
######

#调用子函数 lagr poly，计算出信号幅度为1588mV的待测粒子能量。
print('enegy of particle:%8.3fkeV'%poly(1588))
```

# 3.3插值问题：三次样条插值

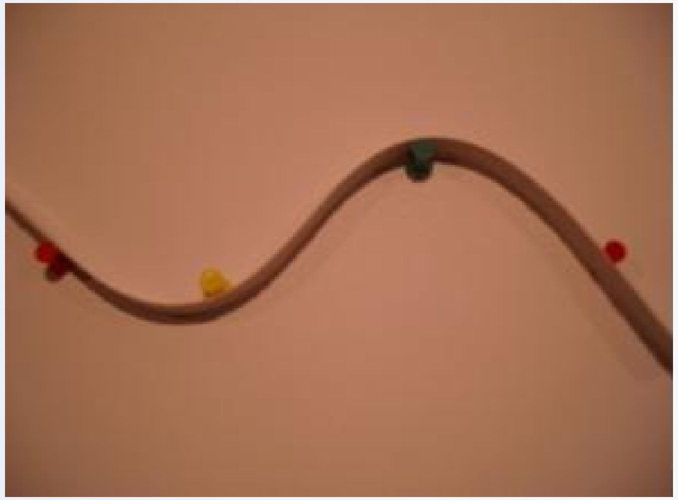
拉格朗日多项式插值法,n个数据点可以构造出n-1 次插值多项式。当n较大时，插值函数为高阶多项式，通常会出现震荡行为，从而引入非物理特征。

解决思路：

（1）采用分段拉格朗日插值，即把n个数据点划分为不同的区间，每个区间分别作拉格朗日插值，将不同区间的插值函数整合起来极为原数据的插值函数。缺点是不同区间构建的插值函数不连续。

**解决思路：**

（2）采用**样条插值法**，使得不同区间的插值函数光滑连接，得到整体光滑的插值函数。



特征：

一阶导数处处光滑，二阶导数处处连续。

$$A=\frac{x-x_{j+1}}{x_j-x_{j+1}} \qquad B=\frac{x-x_j}{x_{j+1}-x_j}$$

$$C=\frac{1}{6}(A^3-A)(x_{j+1}-x_j)^2$$

$$D=\frac{1}{6}(B^3-B)(x_{j+1}-x_j)^2$$

$$y=Ay_j+By_{j+1}+\boxed{Cy_j''+Dy_{j+1}''} \quad *$$

在左右端点 $x_j$, $x_{j+1}$ 为零

从左端点到右端点，二阶导数由 $y_j''$ 线性地变为 $y_{j+1}''$ → 二阶导连续

# 3.3插值问题：三次样条插值

对*求一阶导：

由 $y = Ay_j + By_{j+1} + Cy_j^{''} + Dy_{j+1}^{''}$ 得：

$$\frac{dy}{dx} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{3A^2 - 1}{6}(x_{j+1} - x_j)y_j^{''} + \frac{3B^2 - 1}{6}(x_{j+1} - x_j)y_{j+1}^{''}$$

$$\frac{d^2y}{dx^2} = Ay_j^{''} + By_{j+1}^{''}$$

根据一阶导数连续的要求可得： **N-2**个方程

$$\frac{x_j - x_{j-1}}{6}y_{j-1}^{''} + \frac{x_{j+1} - x_{j-1}}{3}y_j^{''} + \frac{x_{j+1} - x_j}{6}y_{j+1}^{''} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$$

自然边界条件：$y_1^{''}=0$； $y_N^{''}=0$

可确定所有 $y_j^{''}$

# 3.3插值问题：三次样条插值

问题2-1编程实操3.0：

（2）定义了一个子函数 lagr_poly，构建出拉格朗日插值函数，并返回函数值

直接调用 scipy 程序库中的 interpolate 模块来实现三次样条插值函数的构建。

```
#定义了一个子函数 lagr_poly，构建出拉格朗日插值函数，并返回函数值
import numpy as np
import matplotlib.pylab as pl
from scipy import interpolate

#调用 scipy 程序库中的 interpolate 模块
# cubic spline interpolation
tck = interpolate.splrep(signal_alpha_source, energy_alpha_source,k=3,s=1.2)
print(tck)
```

## 3.3 插值问题：三次样条插值

问题2-1编程实操3.0：

（2）调用splrep() ,返回值是三元组tck，包括节点向量、系数以及插值函数阶数等信息,定义了*给出的插值函数。

k=3:花键拟合的程度。建议使用三次样条。甚至应避免使用k值，尤其是在s值小的情况下。1 <= k <= 5

s=1.2:平滑条件。较大的s表示更平滑，而较小的s表示较不平滑。

```
#调用 scipy 程序库中的 interpolate 模块
# cubic spline interpolation
tck = interpolate.splrep(signal_alpha_source, energy_alpha_source,k=3,s=1.2)
print(tck)

```

# 3.3 插值问题：三次样条插值

问题2-1编程实操3.0：

（3）调用splev() ,计算出(1300,1800)范围内的信号幅度对应的粒子能量值.

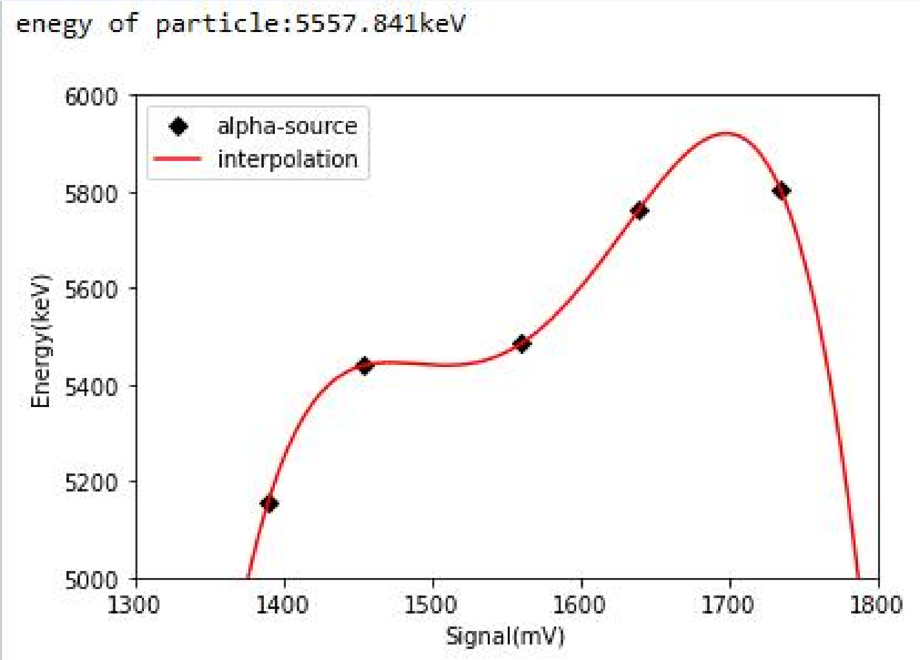
der:要计算的样条导数的阶数(必须小于或等于k，样条的阶数)。

```
signal=np.arange(1300,1800,1)
energy = interpolate.splev(signal, tck, der=0)
```

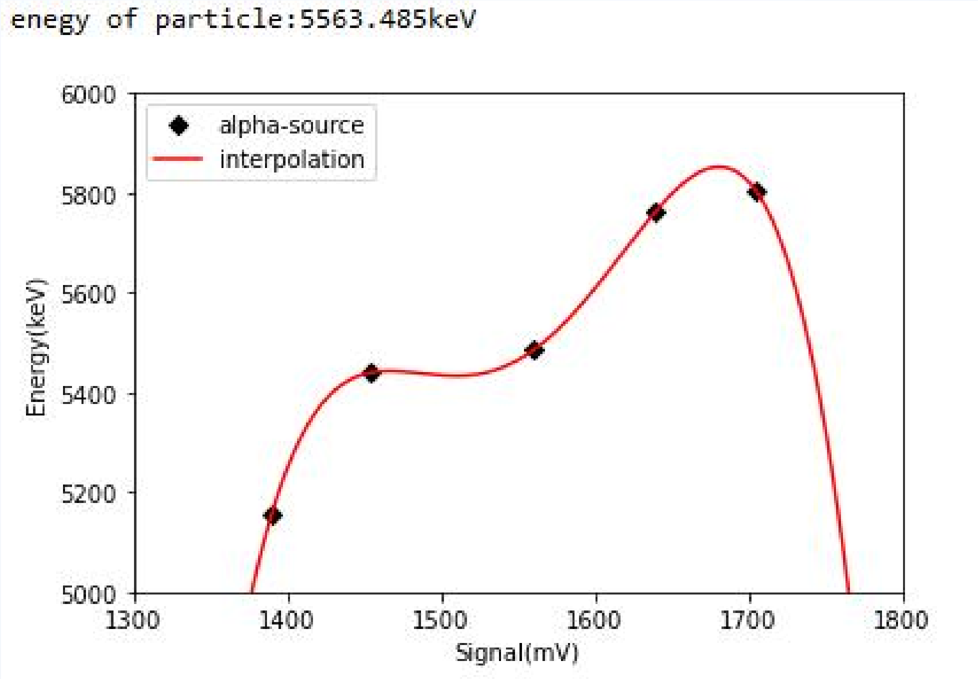
# 3.3 插值问题：三次样条插值

结果对比：

**拉格朗日插值：**




**三次样条插值：**



## 课后练习：

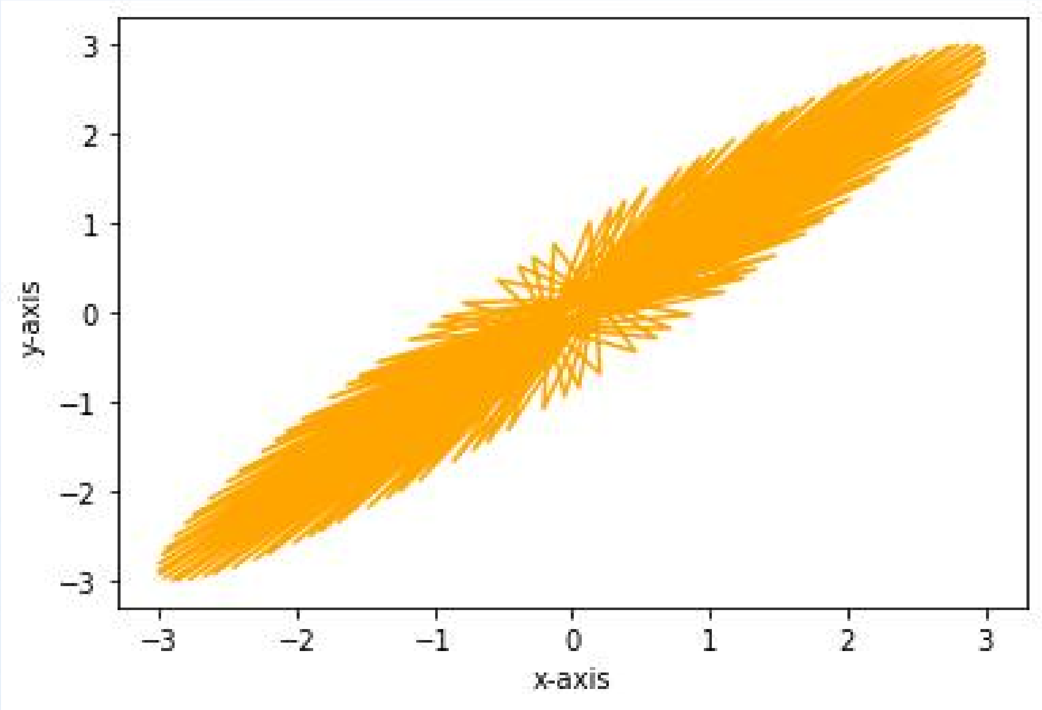
恒星耀发能量-频次分布：

数据文件“flare-frequency.dat”给出的是某恒星的耀发能量和对应的发生次数的数据。1)请编写 python 程序，分别画出能量-频次散点图及其三次样条插值函数曲线，并估计$2.5*10^{32}$erg耀发的发生次数，2)改用拉格朗日插值法估计$2.5*10^{32}$erg耀发的发生次数:3)对使用以上两种方法进行插值和预测的结果进行讨论。

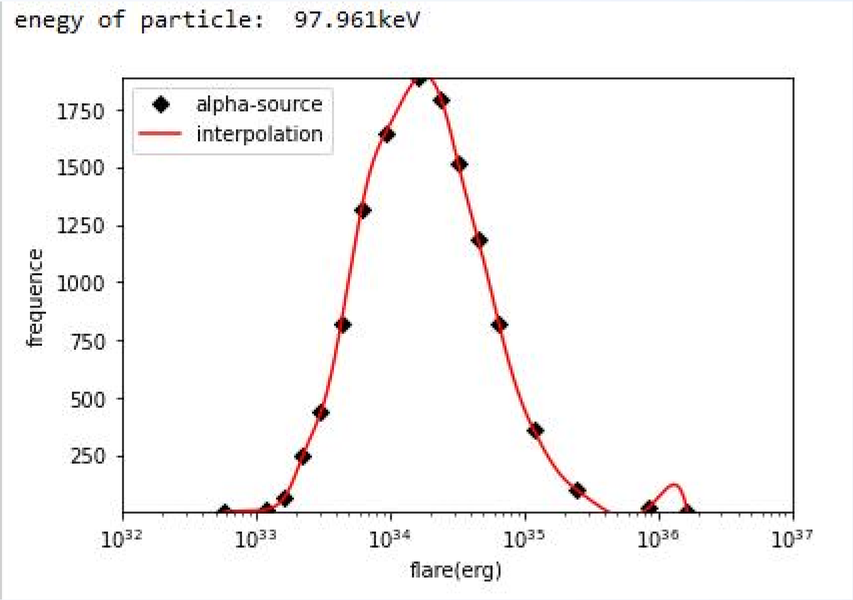
# 课后练习总结：

结果对比：

## 作业一：




## 作业二：



# 3.4 数值微分：蛋白质折叠比热计算

蛋白质分子在高温时会发生变性

$$Cv = \frac{d < E >}{dT}$$

表 2.2 分子动力学模拟得到的不同温度（T）下蛋白质分子 CI2 的内能（E）。

| T(K) | E(kcal/mol) | T(K) | E(kcal/mol) | T(K) | E(kcal/mol) |
|---|---|---|---|---|---|
| 300.0 | -100.5 | 335.0 | -81.4 | 370.0 | -15.0 |
| 305.0 | -100.3 | 340.0 | -69.3 | 375.0 | -12.2 |
| 310.0 | -100.0 | 345.0 | -54.5 | 380.0 | -11.7 |
| 315.0 | -99.1 | 350.0 | -34.2 | 385.0 | -10.5 |
| 320.0 | -97.4 | 355.0 | -25.8 | 390.0 | -10.3 |
| 325.0 | -94.2 | 360.0 | -20.4 | 395.0 | -10.2 |
| 330.0 | -89.7 | 365.0 | -18.9 | 400.0 | -10.1 |

## 由插值函数计算数值微分：

思路：用插值函数$y(x)$的微分代替原始函数$f(x)$的微分

$$f'(x) = y'(x)$$

由线性插值 $$y(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1$$

两边求导，并利用$x_1$-$x_0$=$h$, $h$为步长

$$y'(x) = \frac{y_1 - y_0}{h}$$

两点式微分

# 3.4 数值微分

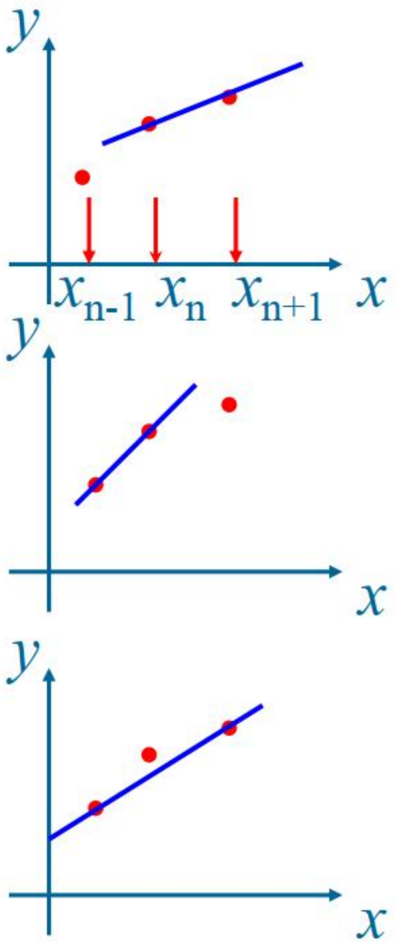
根据所使用的插值数据点，可通过如下三种差分格式：

## 更一般地：

$$f'(x_n)=\frac{y_{n+1}-y_n}{h}$$ 向前差分

$$f'(x_n)=\frac{y_n-y_{n-1}}{h}$$ 向后差分

$$f'(x_n)=\frac{y_{n+1}-y_{n-1}}{2h}$$ 中心差分



# 3.4 数值微分

## 三点公式:

由二次插值函数

$$y(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0+\frac{(x-x_2)(x-x_0)}{(x_1-x_2)(x_1-x_0)}y_1+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2$$

两边求导，并利用$x_2$-$x_1$ =$x_1$-$x_0$=$h$, $h$为步长

$$y'(x)=\frac{2x-x_1-x_2}{2h^2}y_0-\frac{2x-x_2-x_0}{h^2}y_1+\frac{2x-x_0-x_1}{2h^2}y_2$$

得三点公式

$$f'(x_0)\approx y'(x_0)=\frac{-3y_0+4y_1-y_2}{2h}$$

$$f'(x_1)\approx y'(x_1)=\frac{y_2-y_0}{2h}$$

$$f'(x_2)\approx y'(x_2)=\frac{y_0-4y_1+3y_2}{2h}$$

三点微分

# 3.4 数值微分

二次微分：

由三点的二次插值多项式，可进一步求二次微商，得到二阶数值微分公式

$$y'(x)=\frac{2x-x_1-x_2}{2h^2}y_0-\frac{2x-x_2-x_0}{h^2}y_1+\frac{2x-x_0-x_1}{2h^2}y_2$$

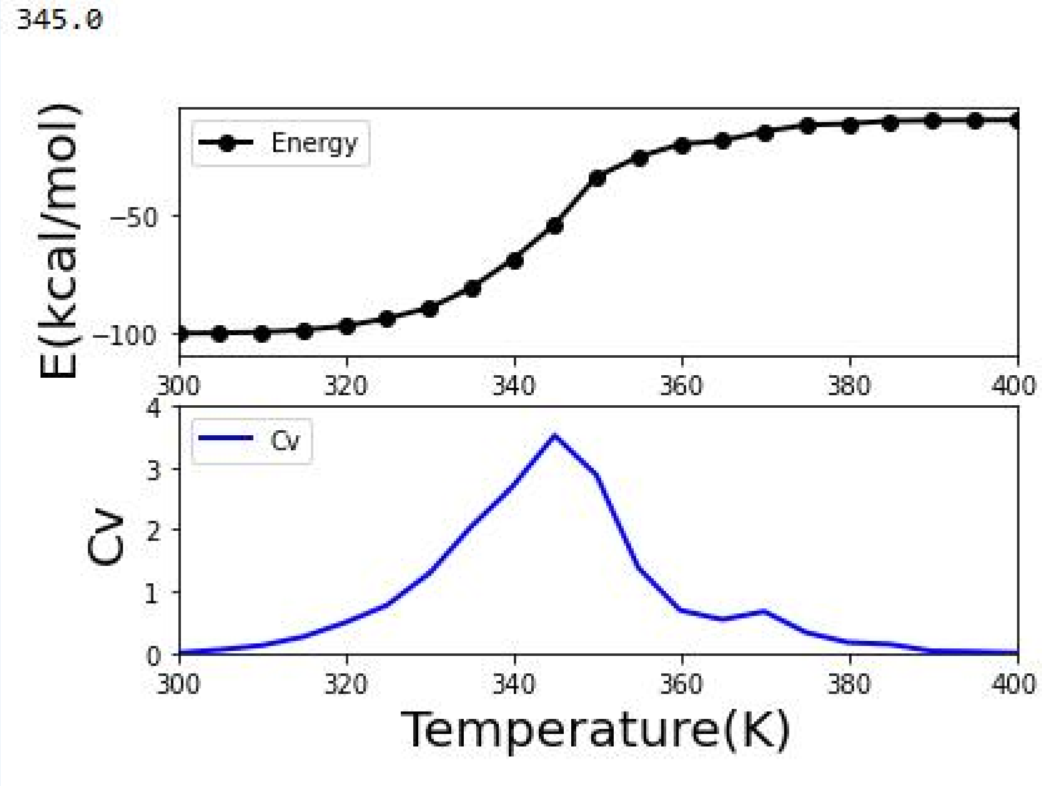
$$y''(x)=\frac{1}{h^2}y_0-\frac{2}{h^2}y_1+\frac{x_1}{h^2}y_2=\frac{y_0-2y_1+y_2}{h^2}$$

得：

$$f''(x_0)=\underline{f''(x_1)}=f''(x_2)=\frac{y_0-2y_1+y_2}{h^2}$$

二阶微分

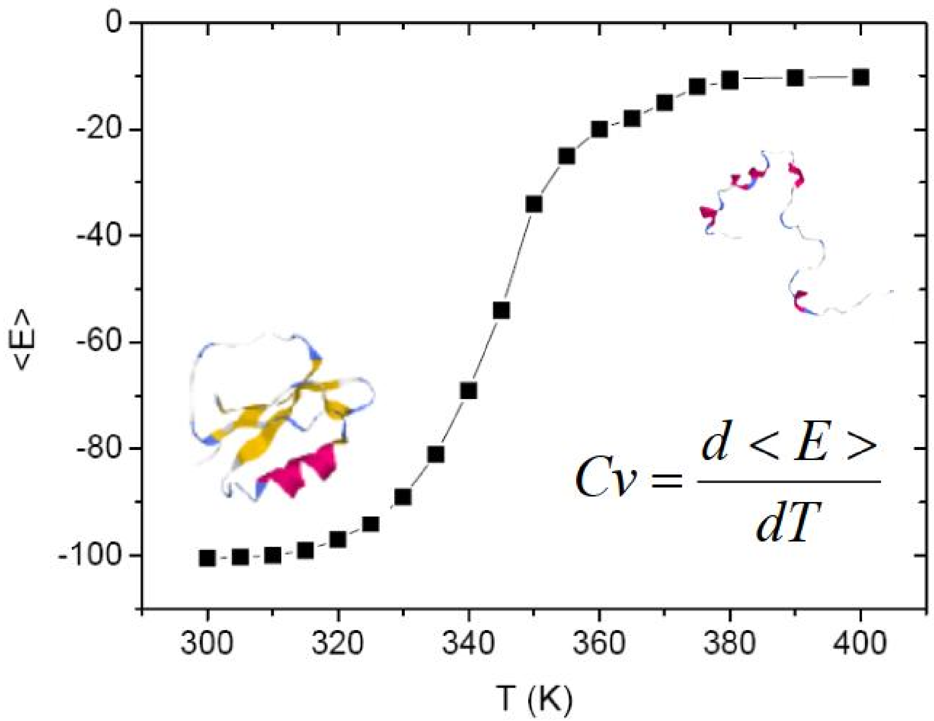
# 3.4 数值微分：应用




举例：

## 蛋白质折叠比热计算

| T | <E> |
|---|---|
| 300 | -100.5 |
| 305 | -100.3 |
| 310 | -100 |
| 315 | -99 |
| 320 | -97 |
| 325 | -94 |
| 330 | -89 |
| 335 | -81 |
| 340 | -69 |
| 345 | -54 |
| 350 | -34 |
| 355 | -25 |
| 360 | -20 |
| 365 | -18 |
| 370 | -15 |
| 375 | -12 |
| 380 | -11 |
| 385 | -10.5 |
| 390 | -10.3 |
| 395 | -10.2 |




$$Cv = \frac{d<E>}{dT}$$

# 3.4 数值微分：应用

第一步：导入模块，读取数据

```python
import matplotlib.pylab as pl
import numpy as np
# ------------------- read in temperature and energy--
-----------------


data=np.loadtxt('Etot-Temp.dat')
temperature =data[:,0]
energy=data[:,1]
```

# 3.4 数值微分：应用

第二步：定义了一个包含n个元素的列表 Cv，为后续记录比热数据做准备，且列表元素都设为 0。

利用中心差分格式，计算出能量对温度的一阶微分，即比热，并更新列表 Cv中的元素值

```
n = len(temperature)
Cv=[0]*n
for i in range(1,n-1):
    Cv[i] = (energy[i+1] - energy[i-1])/(temperature[i+1] - temperature[i-1])
for i in range(len(temperature)):
    if Cv[i]==max(Cv):
        print(temperature[i])
```

第三步：可视化，在图中的两个面板画出能量随温度的变化以及比热随温度的变化

```
fig, ax = pl.subplots(nrows = 2, ncols = 1)

ax[0].plot(temperature, energy, marker='o',color='k',
linewidth=2.0, label='Energy')
ax[1].plot(temperature, Cv, color='b',
linewidth=2.0,label='Cv')
```

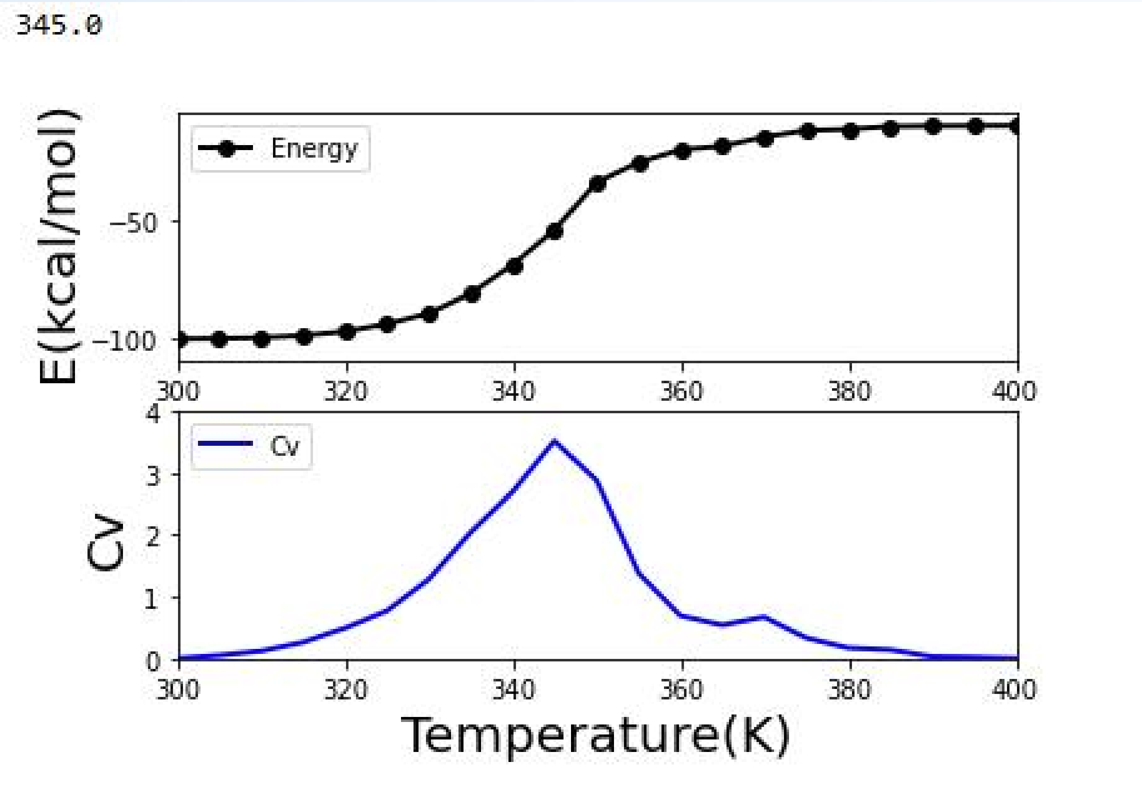
# 3.4 数值微分：应用

第三步：可视化，设置横纵坐标、图例，取值范围

```
ax[0].set_xlim((300, 400))
ax[0].set_ylim((-110., -5.))
ax[1].set_xlim((300, 400))
ax[1].set_ylim((0., 4.))
ax[0].legend(loc='upper left')
ax[1].legend(loc='upper left')
ax[1].set_xlabel('Temperature(K)', fontsize=20)
ax[0].set_ylabel('E(kcal/mol)', fontsize=20)
ax[1].set_ylabel('Cv', fontsize=20)
pl.show()
```

# 3.4 数值微分：应用

结果展示：




在温度较低时(<330K)：

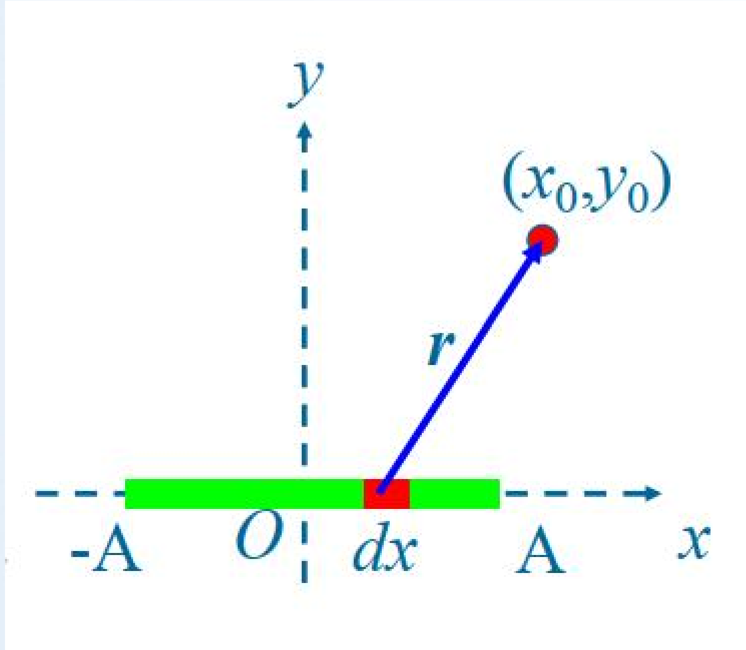
能量随温度变化较缓慢，蛋白质维持在折叠状态。

当温度由 340K 升高至 360K 的过程中：

蛋白质分子的能量显著升高，蛋白质解折叠。

解折叠相变温度T约为 345K

# 3.5 定积分的近似计算：带电细杆产生的静电势




$$V = \frac{1}{4\pi\varepsilon}\int_l \frac{dq}{r} \qquad dq = \lambda(x)dx$$

$$V(x_0, y_0) = \frac{1}{4\pi\varepsilon}\int_{-A}^{A} \frac{\lambda(x)dx}{\left[(x-x_0)^2 + y_0^{\ 2}\right]^{1/2}}$$

$$\lambda(x) = e^{-x^2}$$

# 3.5 定积分的近似计算：带电细杆产生的静电势

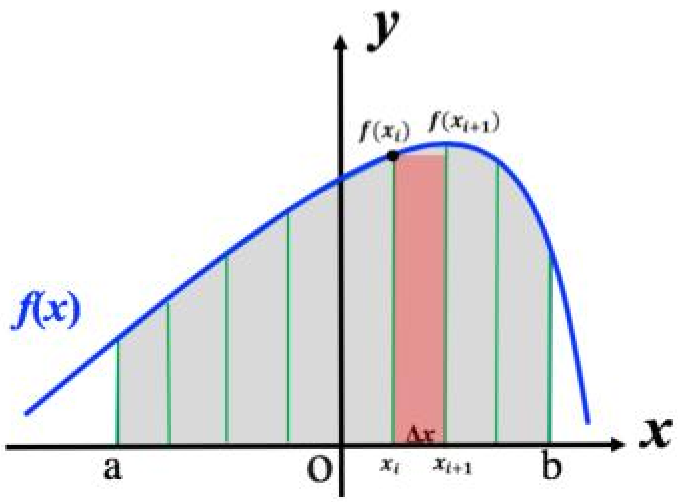
引入符号$f(x)$，令：

$$f(x)=\frac{1}{4\pi\varepsilon}\frac{\lambda(x)}{\left[(x-x_0)^2+{y_0}^2\right]^{1/2}}$$

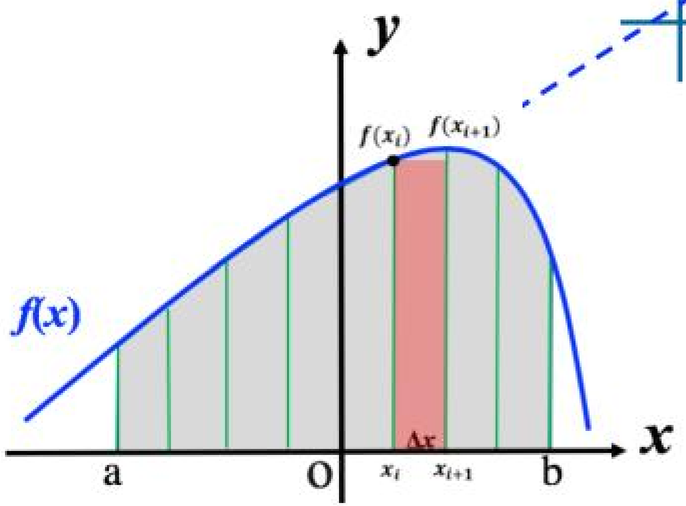
问题变为求解标准的定积分

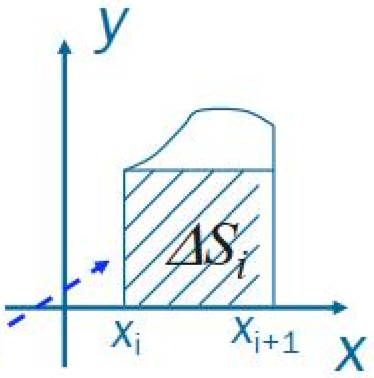
$$V=\int_{-A}^{A}f(x)dx$$

几何意义:求曲线下的面积



# 3.5 定积分的近似计算




$$\Delta S_i = f(x_i)\Delta x \quad i = 1,2,\cdots,n-1$$

$$V = \sum_{i=1}^{n-1} \Delta S_i = \sum_{i=1}^{n-1} f(x_i)\Delta x$$

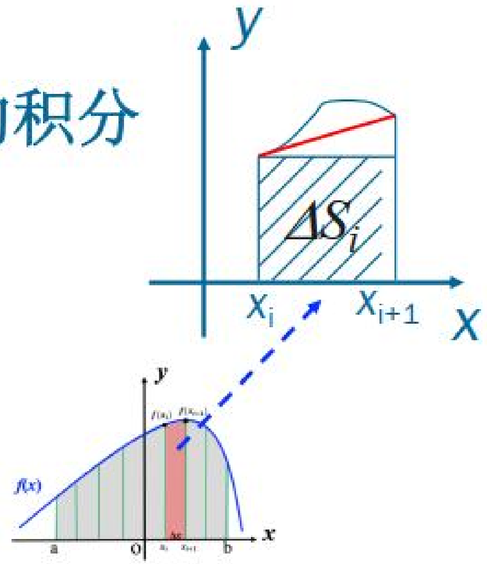
$$V = \sum_{i=1}^{n-1} \Delta S_i = f(x_1)\Delta x + f(x_2)\Delta x + \cdots + f(x_{n-1})\Delta x$$

矩形公式

# 3.5 定积分的近似计算

对于矩形公式，当积分步长较大时，会导致较大的误差。

想法：用插值函数的积分代替原函数的积分



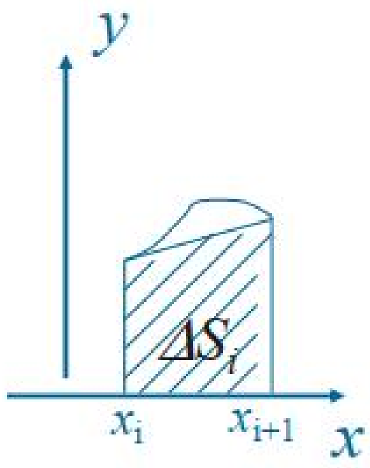

$$\int_a^b f(x)dx = \int_a^b y(x)dx$$

其中：$y(x)$为插值多项式，可以解析积分

# 3.5 定积分的近似计算

线性插值公式：

$$y(x)=\frac{x-x_{i+1}}{x_i-x_{i+1}}y_i+\frac{x-x_i}{x_{i+1}-x_i}y_{i+1}$$




$$\Delta S_i=\int_{x_i}^{x_{i+1}} f(x)dx$$

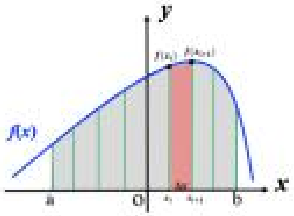
$$\approx\int_{x_i}^{x_{i+1}} y(x)dx=\frac{1}{2}[f(x_i)+f(x_{i+1})]\Delta x \quad i=1,2,\cdots,n-1$$

存在截断误差

$$R=\int_{x_i}^{x_{i+1}} f(x)dx-\frac{\Delta x}{2}(f(x_i)+f(x_{i+1}))$$

$$=-\frac{(\Delta x)^3}{12}f''(\eta)\quad (x_i<\eta<x_{i+1})$$

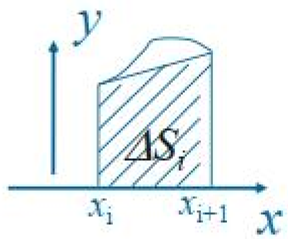
$$V=\sum_{i=1}^{n-1}\Delta S_i=\sum_{i=1}^{n-1}[f(x_i)+f(x_{i+1})]\frac{\Delta x}{2}$$




梯形公式

# 3.5 定积分的近似计算

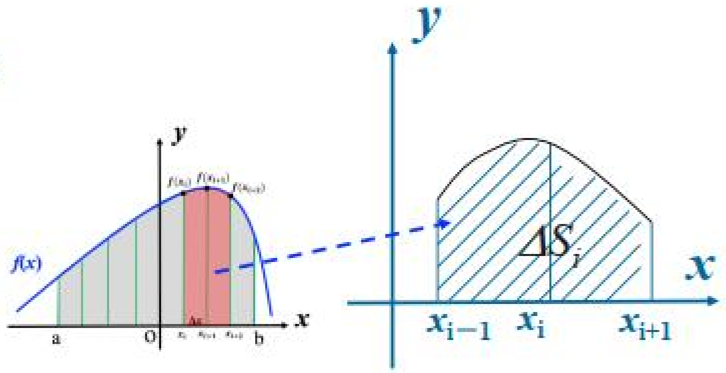
由二次插值函数




$$y(x)=\frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})}y_{i-1}+\frac{(x-x_{i+1})(x-x_{i-1})}{(x_i-x_{i+1})(x_i-x_{i-1})}y_i+\frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_{i-1})}y_{i+1}$$

抛物线形公式
辛普森公式

对$y(x)$由$x_{i-1}$到$x_{i+1}$积分:




$$\Delta S=\int_{x_{i-1}}^{x_{i+1}}f(x)dx$$

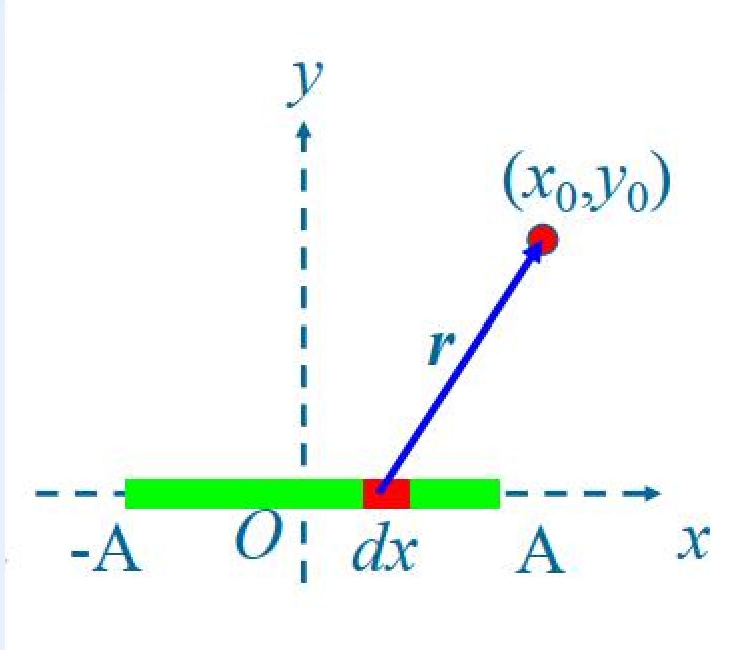
$$\approx\int_{x_{i-1}}^{x_{i+1}}y(x)dx=\frac{1}{3}[y_{i-1}+4y_i+y_{i+1}]\Delta x$$

$$R=\int_{x_i}^{x_{i+1}}f(x)dx-\frac{\Delta x}{2}(f(x_i)+f(x_{i+1}))$$

$$=-\frac{(\Delta x)^5}{90}f^{(4)}(\eta)\quad(x_i<\eta<x_{i+1})$$

$$V=\sum_{i=2,4\ldots}^{n-1}\Delta S_i=\frac{1}{3}[f(x_1)+4f(x_2)+2f(x_3)+4f(x_4)+\cdots+f(x_n)]\Delta x$$

# 3.5 定积分的近似计算：带电细杆产生的静电势




$$V=\frac{1}{4\pi\varepsilon}\int_l \frac{dq}{r} \qquad dq=\lambda(x)dx$$

$$V(x_0,y_0)=\frac{1}{4\pi\varepsilon}\int_{-A}^{A}\frac{\lambda(x)dx}{\left[(x-x_0)^2+y_0{}^2\right]^{1/2}}$$

$$\lambda(x)=e^{-x^2}$$

选取 $\frac{1}{4\pi\varepsilon}$ 为电势值的单位

# 3.5 定积分的近似计算：应用

第一、二步：导入模块，定义被积函数，给出积分区间的划分，确定了带点细杆的长度以及计算得到的电势值的单位，初始化x坐标

```
import numpy as np
import matplotlib.pyplot as pl
#--------defining a function----------
def f(x,x0,y0):
    return np.exp(-x**2)/np.sqrt((x-x0)**2+y0**2)
N=101  # number of points
A=1.0  # half-length of the stem
c=1.0  # determine the unit of filed. c= 1/4piepsilon
xx=np.zeros(N)  #生成100零元素的列表
```

# 3.5 定积分的近似计算：应用

第三步：得到积分步长、计算出积分区间内的节点x坐标值，给定待计算电势的坐标点($x_0$，$y_0$)

```
h = 2.0*A/float(N-1)  # integration step

for i in np.arange(0,N,1):
    xx[i] = i*h - A    # coordinate alining the x-axis

xaxis = np.arange(-5.0,5.0,0.2) # the the (x0.y0) points
yaxis = np.arange(-5.0,5.0,0.2)
```

## 3.5 定积分的近似计算：应用

第四步：初始化电势值储存列表，对所有待计算电势的坐标点($x_0$, $y_0$)循环，利用梯形积分法计算出每个坐标点的电势值

```
field=[] # list to store the calculated field.
for x0 in xaxis:   #calculate the fields for a number of
(x0,y0) pionts
    for y0 in yaxis:
        v = 0.0
#trapezoid integration
        for i in np.arange(0,N-1,1):
            v = v + 0.5*(f(xx[i],x0,y0)+f(xx[i+1],x0,y0))*h
        field.append(v)
```

# 3.5 定积分的近似计算：应用

第五步：将列表 field 转化为数组，由于 afield 的默认横坐标为 y，纵坐标为 x，为了作图方便，将 afield 进行转置，得到横坐标为 x，纵坐标为y的数组

```
afield = np.array(field) # convert list to array
afield.shape = len(xaxis), len(yaxis) #setup the dimension of array
afield_xy = afield.T #so that X --> x axis
```

## 3.5 定积分的近似计算：应用

第六步：可视化，子图1，确定了待展示的电势值的范围画出电势值的等高线，并进行数字标记。

```
extent = [-5.0, 5.0, -5.0, 5.0] # the range of points to be plotted.
fig = pl.figure(figsize=(9,4)) # create a figure
#contour line
ax1 =fig.add_subplot(1,2,1) # create a subplot
levels = np.arange(0.0,5.0,0.05) # setup the level of field to show.
cs=ax1.contour(afield_xy,levels,origin='lower',linewidths=2,
extent=extent) # contour line
ax1.clabel(cs)
```

## 3.5 定积分的近似计算：应用

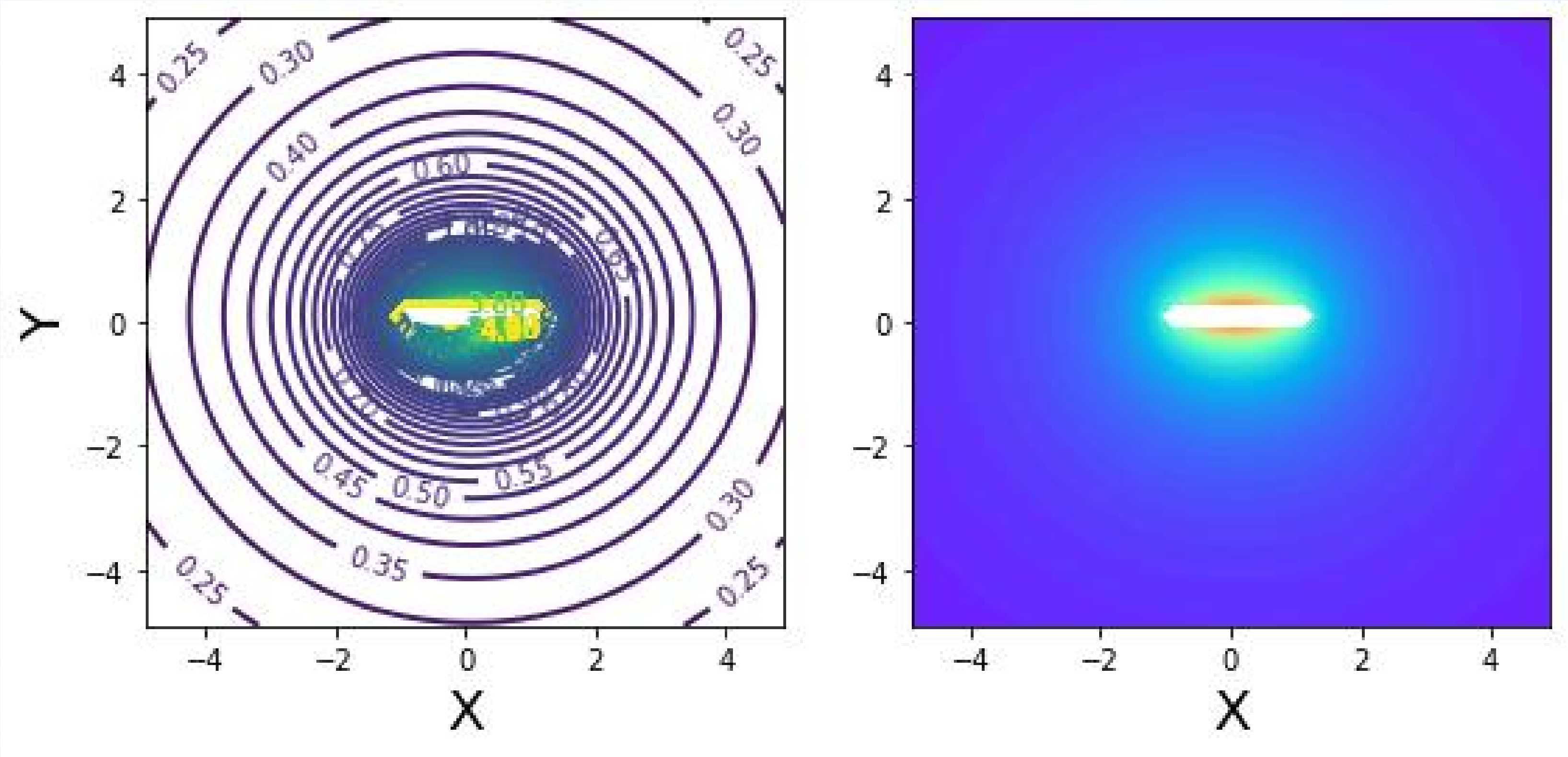
第六步：可视化，子图2，画出电势值的填充模式等高图。

```
# color bar
ax2=fig.add_subplot(1,2,2)
cs=ax2.contourf(afield_xy,levels,origin='lower',extent=extent,cmap=pl.cm.rainbow)

ax1.set_xlabel('X', fontsize=20)
ax1.set_ylabel('Y', fontsize=20)
ax2.set_xlabel('X', fontsize=20)
pl.show()
```

# 3.5 定积分的近似计算：应用

结果展示：




左图用等高线来表示电势值分布，右图用填充模式的等高图来展示电势分布，其中从红色到蓝色，电势值由高变低。从图中可以看出，在靠近细杆的位置电势最高，随着远离细杆位置，电势逐渐变弱。

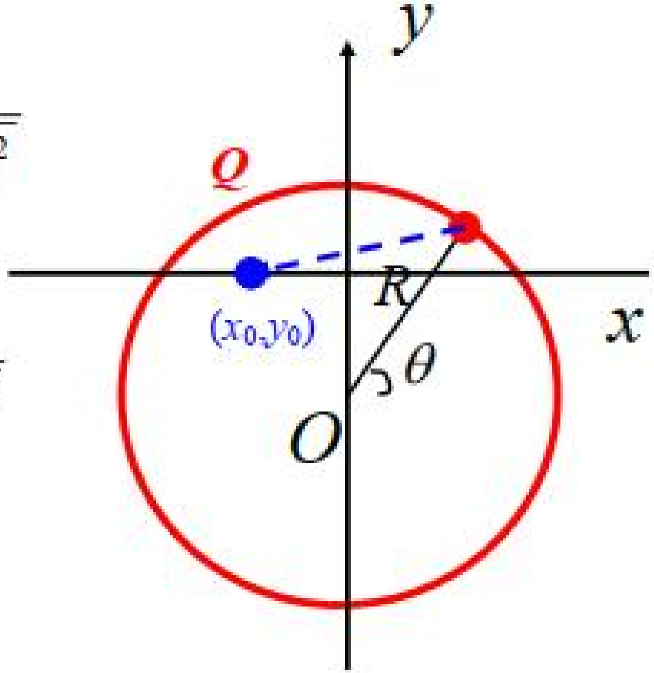
## 课后练习:

问题:二维平面半径为R的圆环均匀带电，总电量为Q，求带电圆环在二维平面产生的电势分布，写出python代码并将计算结果用图形表示出来(物理常数、Q、R可设为1.0)。

$$V(x_0,y_0)=\frac{1}{4\pi\varepsilon}\int_0^{2\pi}\frac{Q}{2\pi R}\frac{Rd\theta}{\sqrt{(x_0-R\cos\theta)^2+(y_0-R\sin\theta)^2}}$$

令 $$f(\theta)=\frac{1}{4\pi\varepsilon}\frac{Q}{2\pi R}\frac{R}{\sqrt{(x_0-R\cos\theta)^2+(y_0-R\sin\theta)^2}}$$

$$V(x_0,y_0)=\int_0^{2\pi}f(\theta)d\theta$$




辛普森公式:

$$V=\sum_{i=2,4,\ldots}^{N-1}\frac{1}{3}[f(\theta_{i-1})+4f(\theta_i)+f(\theta_{i+1})]\Delta\theta \qquad \Delta\theta=\frac{2\pi}{N-1}$$

# 勤动手，多思考！

苏湘宁　　邮箱：suxn@hainanu.edu.cn