第5章 数值微分和积分

- 数值微分函数
  - diff
  - fnder
  - polyder
- 数值积分函数
  - trapz
  - quad
  - quadl

上讲内容

1) 函数的功能是什么？
2) 函数要求的输入变量是什么？这些输入变量有什么特殊要求？
3) 函数返回的输出变量是什么？

p=polyfit(x,y,n)

[b,bint,r,rint,stats] =regress(y,x)

[b,r,J] = nlinfit(X,y,fun,b0,options)

数值微积分方法

- ✓ 微积分的数值方法，适合求解没有或很难求出微分或积分表达式的实际化工问题的计算，例如：列表函数求微分或积分。
- ✓ 数值微分和数值积分与插值和拟合往往是密不可分的。
- ✓ 进行数值微分时，常针对离散的数据点，利用插值和拟合可以减少误差；
- ✓ 数值积分的基本思路也来自于插值法。可通过构造一个插值多项式来代替原函数，从而使问题简化。

建立数值微分公式的三种思路

常用三种思路建立数值微分公式：

1. 从微分定义出发，通过近似处理，得到数值微分的近似公式；

2. 从插值近似公式出发，对插值公式的近似求导可得到数值微分的近似公式；

3. 先用最小二乘拟合方法根据已知数据或得近似函数(如样条函数)，再对此近似函数求微分可得到数值微分的近似公式。

差分近似微分

在微积分中，一阶微分的计算可以取下列极限求得：

$$f'(x)=\frac{df(x)}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}=\lim_{h\to 0}\frac{f(x)-f(x-h)}{h}=\lim_{h\to 0}\frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h}$$

取其达到极限前的形式，就得到以下微分的差分近似式：

$$f'(x)=\frac{df(x)}{dx}\approx\frac{f(x+h)-f(x)}{h}\approx\frac{f(x)-f(x-h)}{h}\approx\frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h}$$

上式中三种不同表示形式依次是一阶前向差分、一阶后向差分和一阶中心差分来近似表示微分。其中一阶中心差分的精度较高。

注：高阶微分项可以利用低阶微分项来计算，如二阶微分与对应的差分式有：

$$f''(x)\approx\frac{f'(x+h)-f'(x)}{h}\approx\frac{f'(x)-f'(x-h)}{h}\approx\frac{f'(x+\frac{h}{2})-f'(x-\frac{h}{2})}{h}$$

差分的MATLAB实现

在MATLAB中，可用diff求向量相邻元素的差值，diff(y)./diff(x)则表示一阶差分。

diff函数调用形式：

Y = diff(X,n)

输入变量：

- X表示待求差值的变量，可以是向量或矩阵。
- n表示函数循环运算n次；
- 当X为矩阵时，可使用diff(X,n,dim)指定求差的维数，dim为2时，表示对行元素求差值。

例题

丁二烯的气相二聚反应方程式如下

$$2C_4H_6 \longrightarrow (C_4H_6)_2$$

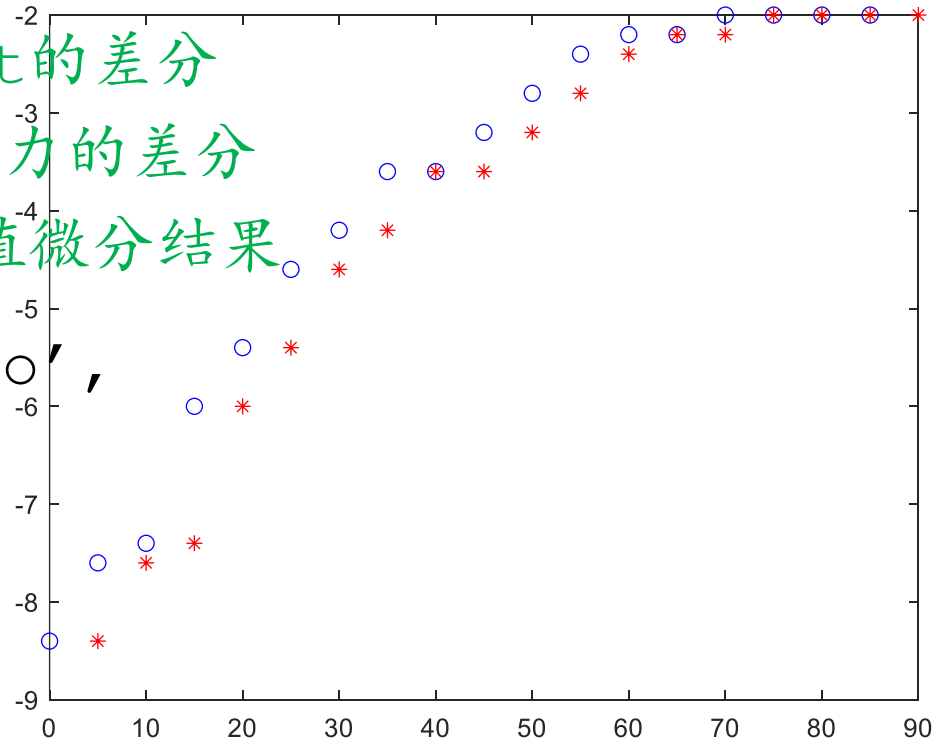
实验在一定容器的反应器中进行，326^{0}C时，测得物系中丁二烯的分压P_A(mmHg)与时间的关系如表所示。用数值微分法计算所列时刻每一瞬间的反应速率：$\frac{dp_A}{dt}$

t (min)	P_A (mmHg)	t (min)	P_A (mmHg)
0	632.0	50	362.0
5	590.0	55	348.0
10	552.0	60	336.0
15	515.0	65	325.0
20	485.0	70	314.0
25	458.0	75	304.0
30	435.0	80	294.0
35	414.0	85	284.0
40	396.0	90	274.0
45	378.0		

例题

$$f'(x)=\frac{df(x)}{dx}\approx\frac{f(x+h)-f(x)}{h}\approx\frac{f(x)-f(x-h)}{h}\approx\frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h}$$

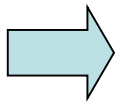
```
function ReactionRate
t=0:5:90;
pA=[632.0 590.0 552.0 515.0 485.0 458.0
435.0 414.0 396.0 378.0 362.0 348.0 336.0
325.0 314.0 304.0 294.0 284.0 274.0];
dt=diff(t); % 求时间t的差分
dpA=diff(pA); % 求压力的差分
q=dpA./dt    % q为数值微分结果
plot(t(1:end-1),q,'bo',
t(2:end),q,'r*')
```



差分的误差

计算y=sin(x)在0:0.1π:2π 的导数值及其误差

```
h=0.1*pi;
x=0:h:2*pi;
y=sin(x);
dy1=diff(y)/h;
dy0=cos(x);
err1=norm(dy1-dy0(1:20))
```



err1 = 0.4954

- 如果把步长改为0.01*pi，则误差可以降低至0.1571，可见减小步长有利于减小数值微分的误差，但是否步长越小越好呢？
- 差分代替微分计算简便，在一些数值方法的中间过程中仍然大量被运用，但应注意采用合适的步长。
- 在化工实用过程中，一般最好将数据利用插值或拟合得到多项式，然后对近似多项式进行微分。

三次样条插值函数求微分

若三次样条插值函数$S(x)$收敛于$f(x)$，那么导数$S'(x)$收敛于$f'(x)$，因此用样条插值函数$S(x)$作为$f(x)$的近似函数，不但彼此的函数值非常接近，而且导数值也很接近。

用三次样条插值函数建立的数值微分公式为：

$$f'(x) \approx S'(x)$$

$$S(x)=\frac{1}{6h_i}[(x_i-x)^3M_{i-1}+(x-x_{i-1})^3M_i]+\left(y_{i-1}-\frac{h_i^2}{6}M_{i-1}\right)\frac{x_i-x}{h_i}+\left(y_i-\frac{h_i^2}{6}M_i\right)\frac{x-x_{i-1}}{h_i}$$

求导

$$f'(x)\approx S'(x)=-\frac{(x_i-x)^2}{2h_i}M_{i-1}+\frac{(x-x_{i-1})^2}{2h_i}M_i+\frac{(y_i-y_{i-1})}{h_i}-\frac{h_i}{6}(M_i-M_{i-1})$$

其中，$i=1,2,\cdots,n$ ；$x\in[x_{i-1},x_i]$

上式不但适用于求节点处的导数，而且可求非节点处的导数。

三次样条插值求微分的MATLAB函数

步骤1：对离散数据用spline/pchip得到其三次样条插值函数

调用形式　　pp = spline(x,y)或pp = pchip(x,y)或pp=interp1(x,y,'method','pp')

其中：　x,y分别为离散数据对的自变量和因变量；

　　pp为得到的三次样条插值函数。

步骤2：对用fnder函数求三次样条插值函数的导数

调用形式　　fprime = fnder(pp,dorder)

其中：pp为三次样条插值函数；

　　dorder为三次样条插值函数的求导阶数；

　　fprime为得到的三次样条插值函数的导函数。

步骤3：用fnval函数求导函数在未知点处的导数值

调用形式　　v = fnval(fprime,x)

其中：fprime为三次样条插值函数导函数；

　　x为未知点处自变量值；　v为未知点处的导数值。

例题

某液体冷却时，温度随时间的变化数据如下表所示：

t	0	1	2	3	4	5
T	92.0	85.3	79.5	74.5	70.2	67.0

试编写一个MATLAB函数分别计算t＝2，3，4min及t＝1.5，2.5，4.5min时的降温速率。

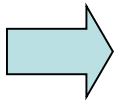
例题

```
function Demo2
t=0:5;
T=92,85.3,79.5,74.5,70.2,67;
cs=spline(t,T); % 生成三次样条插值函数
plot(t,T,'bo',0:0.1:5,fnval(cs,0:0.1:5),'k-
')
pp=fnder(cs); % 生成三次样条插值函数的导函数
t1=[2,3,4,1.5,2.5,4.5];
dT=fnval(pp,t1);% 计算导函数在t1处的导数值
disp('相应时间时的降温速率:')
disp([t1;dT])
```

插值求微分误差

计算y=sin(x)在0:0.1π:2π 的导数值及其误差

```
h=0.1*pi;
x=0:h:2*pi;
y=sin(x);
pp=spline(x,y);
dp=fnder(pp);
dy1=fnval(dp,x);
dy0=cos(x);
err1=norm(dy1-dy0(1:end))
```



err1 = 0.0025

与差分相比，插值后微分显著降低了误差。

最小二乘法拟合函数求微分

当离散数据不可避免地含有较大随机误差时，可采用最小二乘法样条拟合实验数据，获得一个函数模型，然后再对其求导数。多项式拟合和样条拟合都可以用于求微分

多项式拟合求微分

离散数据 —polyfit()→ 向量p表示的多项式拟合函数 —polyder()→

导函数pp —polyval()→ pp在xi的导数值。

样条拟合求微分

离散数据 —csaps()或spap2()或spaps()→ 样条拟合函数sp —fnder()→

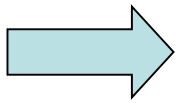
sp的导数pp —fnval()→ pp在xi的导数值。

多项式拟合求微分

导函数polyder( )的调用格式为：pp=polyder(p)
该函数对向量p表示的多项式函数进行求导，返回导函数pp

利用多项式拟合计算y=sin(x)在0:0.1π:2π 的导数值及其误差

```
h=0.1*pi;
x=0:h:2*pi;
y=sin(x);
pp=polyfit(x,y,5);
dp=polyder(pp);
dy1=polyval(dp,x);
dy0=cos(x);
err1=norm(dy1-dy0(1:end))
```



```
err1 =    0.1579
```

样条拟合求微分的MATLAB实现

最小二乘法样条拟合函数求微分共三个步骤：

Step 1:对离散数据用cspas/spaps/spap2函数得到最小二乘样条拟合函数。

调用格式：pp = csaps(x,y)

其中： x,y——要处理的离散数据（）

Step 2:可用fnder函数求样条拟合函数的导函数;

sp=fnder(pp)

Step3: 可用fnval函数求导函数在未知点处的导数值。

fnder( )和fnval（）调用形式以前已经介绍过。

小结

由离散数据求数值微分的四种方法及有关MATLAB函数：

1）差分法 用差分函数diff()近似计算导数，即

dy=diff(y)./diff(x)。

2）多项式拟合方法

离散数据 —polyfit()→ 向量p表示的多项式拟合函数 —polyder()→

导函数pp —polyval()→ pp在xi的导数值。

3）三次样条插值方法

离散数据 —spline(),pchip()→ 三次样条插值函数cs —fnder()→

cs的导数pp —fnval()→ pp在xi的导数值

4）样条拟合方法（最小二乘法）

离散数据 —csaps()或spap2()或spaps()→ 样条拟合函数sp —fnder()→

sp的导数pp —fnval()→ pp在xi的导数值。

例题

反应物A在一等温间歇反应器中发生的反应为：A→产物

测量得到的反应器中不同时间下反应物A的浓度C_A如下表所示

t(s)	0	20	40	60	120	180	300
C_A (mol/L)	10	8	6	5	3	2	1

系统的动力学模型为：

$$-\frac{dC_A}{dt} = k\,C_A{}^m$$

试根据表中数据确定其反应速率方程。

例题

首先根据表中数据采用数值微分计算反应速率$\frac{dC_A}{dt}$

其次，将动力学模型线性化，方程两边取对数：

$$\ln(-\frac{dC_A}{dt}) = m\ln C_A + \ln k$$

令 $y = \ln(-\frac{dC_A}{dt}), x = \ln C_A$

则原模型变为：

$$y = \ln k + mx$$

最后采用线性拟合方法确定模型参数

```
function Cha5demo3
t=[0 20 40  60 120  180 300];
CA=[10  8   6   5   3   2   1];
sp=csaps(t,CA);
pp=fnder(sp);
dCAdt=fnval(pp,t);
ti=linspace(t(1),t(end),200);
CAi=fnval(sp,ti);
plot(t,CA,'bo',ti,CAi,'r-
'),xlabel('t'),ylabel('CA')
y=log(-dCAdt);x=log(CA);
p=polyfit(x,y,1);
k=exp(p(2)),m=p(1)
```

执行结果：

k =0.0059

m =1.2904

$$-\frac{dC_A}{dt} = 0.0059C_A^{1.2904}$$

数值积分

对于积分：$$I(f)=\int_a^b f(x)dx$$

解析法

Newton-Leibniz公式

$$\int_a^b f(x)dx = F(x)\Big|_a^b = F(b)-F(a)$$ F(x)为f(x)的原函数

1. 有的原函数十分复杂难以计算；
2. 被积函数过于特殊或原函数不能用初等函数表示；
3. 被积函数以一组数据形式表示；

- ✓ 当Newton-Leibniz方法不适用时，可以采用积分的近似计算方法：数值积分
- ✓ 数值积分在化学化工领域应用甚广，如反应热效应计算、热容计算、熵的计算、反应活化能的计算等

数值积分的基本思路和方法

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

- 常用的数值积分的基本思路来自于插值法
- 通过构造一个插值多项式Pn(x)作为f(x)的近似表达式，用Pn(x)的积分值作为f(x)的近似积分值。

- 数值积分的方法很丰富，常用的插值型求积公式有两类：一类是等距节点的牛顿—柯特斯(Newton-Cotes)求积公式；另一类是不等距节点的高斯型求积公式。

Newton-Cotes求积公式

Newton-Cotes公式是指等距节点下使用Lagrange插值多项式建立的数值求积公式

设：$f(x) \in C[a,b]$，将积分区间$[a,b]$分割为n等份，各节点为：$x_k = a + kh,\ k = 0,1,2,\cdots,n$，其中$h = (b-a)/n$为步长，可建立$f(x)$的Lagrange插值多项式：

$$L_n = \sum_{k=0}^{n} f(x_k) l_k(x)$$

则$f(x)$的积分近似等于Lagrange插值多项式的积分

$$I = \int_a^b f(x)dx \approx \int_a^b L_n(x)dx = \int_a^b \sum_{k=0}^{n} f(x_k) l_k(x)dx = \sum_{k=0}^{n} f(x_k) \int_a^b l_k(x)dx$$

Newton-Cotes求积公式

$$A_k = \int_a^b l_k(x)dx = \int_a^b \prod_{\substack{0\le j\le n \\ j\ne k}} \frac{x-x_j}{x_k-x_j}dx$$

令： $x = a + th$ ⇨ $dx = hdt,\ x_i - x_j = (i-j)h,\ x - x_j = (t-j)h$

$$A_k = h\int_0^n \prod_{\substack{j=0 \\ j\ne k}} \frac{t-j}{k-j}dt = (b-a)\frac{1}{n}\int_0^n \prod_{\substack{j=0 \\ j\ne k}} \frac{t-j}{k-j}dt$$

令： $C_i = A_i/(b-a)$

$$C_i = \frac{(-1)^{n-k}}{n\cdot k!\cdot(n-k)!}\int_0^n \prod_{\substack{0\le j\le n \\ j\ne k}} (t-j)dt$$

牛顿—柯特斯求积公式 $I = \int_a^b f(x)dx \approx (b-a)\sum_{i=0}^{n} C_i f(x_i)$

C_i是既不依赖于被积函数，也不依赖于积分区间的常数，称为柯特斯系数。

梯形求积公式

在Newton-Cotes公式中,n=1,2,4时的公式是最常用也是最重要三个公式,称为低阶公式

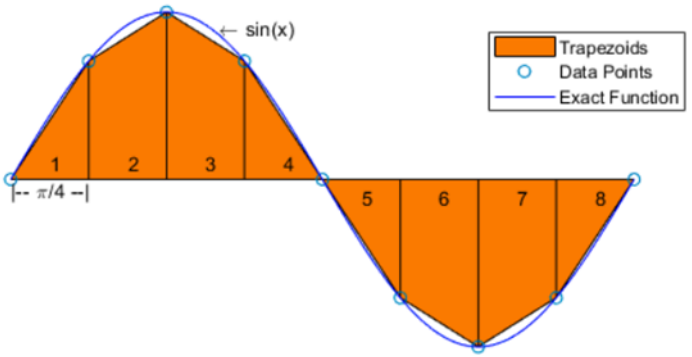
1）梯形(trapezoid)公式

n=1，则$x_0 = a, x_1 = b, h = b - a$

Cotes系数为 $C_0^{(1)} = -\int_0^1 (t-1)dt = \frac{1}{2}$ $C_1^{(1)} = \int_0^1 t dt = \frac{1}{2}$

求积公式为 $I_1(f) = (b-a)\sum_{k=0}^{1} C_k^{(1)} f(x_k) = \frac{b-a}{2}[f(x_0) + f(x_1)]$

上式称为梯形求积公式



Simpson公式

n=2，则$x_0 = a, x_1 = \frac{a+b}{2}, h = \frac{b-a}{2}$

Cotes系数为 $C_0^{(2)} = \frac{1}{4}\int_0^2 (t-1)(t-2)dt = \frac{1}{6}$

$$C_1^{(2)} = \frac{-1}{2}\int_0^2 t(t-2)dt = \frac{4}{6} \qquad C_2^{(2)} = \frac{1}{4}\int_0^2 (t-1)tdt = \frac{1}{6}$$

求积公式为 $I_2 = (b-a)\sum_{k=0}^{2} C_k^{(2)} f(x_k)$

$$I_2(f) = (b-a)[\frac{1}{6}f(x_0) + \frac{4}{6}f(x_1) + \frac{1}{6}f(x_2)]$$

$$= \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]$$

Simpson求积公式

复化法求积公式

Newton-Cotes公式当n大于7时，公式的稳定性将无法保证，因此,在实际应用中一般不使用高阶公式而是采用低阶复合求积法

复合求积法：将积分区间 $[a,b]$ 分成n个相等的子区间，而后对每个子区间再应用梯形公式或Simpson公式积分：

复化梯形公式：
$$T_n = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{k=1}^{n-1} f(a+kh) \right]$$

复化Simpson公式：
$$S_n = \frac{h}{6}\sum_{k=0}^{n-1}((f(x_k) + 4f(x_{k+\frac{1}{2}}) + f(x_{k+1}))$$

自适应求积公式

复化法求积的缺点：

- 采等步长方法，从而限制了它的效率
- 无法直接计算误差

⇒ 自适应求积（自动确定步长）

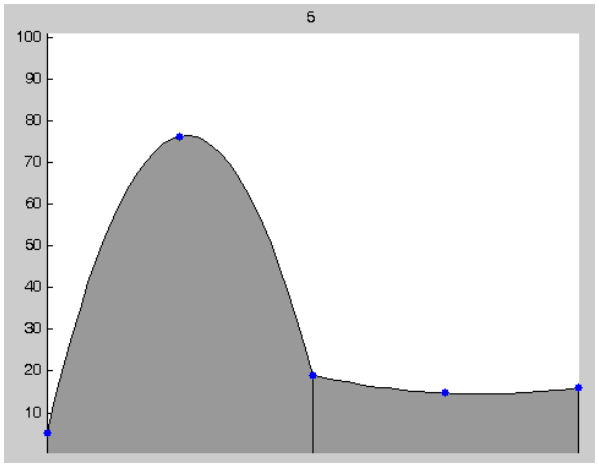
以Simpson积分法为例，某区间$[a_k, b_k]$，记$h_k = b_k - a_k$，考虑该区间上的Simpson积分和二等分以后的两个Simpson积分和：

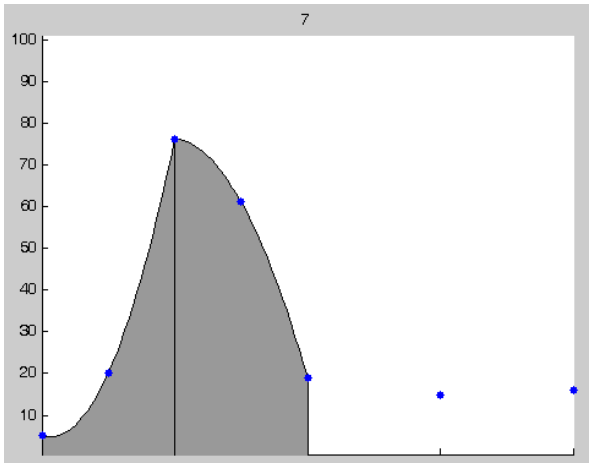
$$S_1 = \frac{h_k}{6}[f(a_k) + 4f(a_k + \frac{1}{2}h_k) + f(b_k)]$$

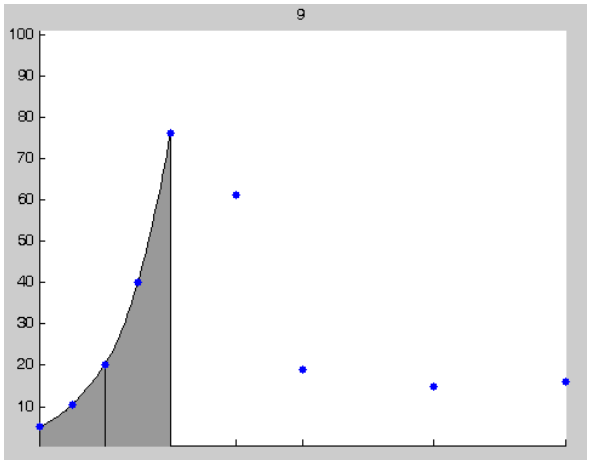
$$S_2 = \frac{h_k}{12}[f(a_k) + 4f(a_k + \frac{1}{4}h_k) + 2f(a_k + \frac{1}{2}h_k) + 4f(a_k + \frac{3}{4}h_k) + f(b_k)]$$

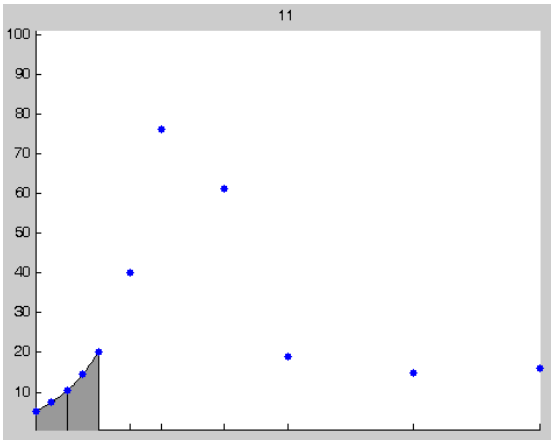
取计算需满足的精度为ε，令$\Delta = |0.1(S_2 - S_1)|$，当$\Delta \leq \varepsilon$时，可认为区间$[a_k, b_k]$上的Simpson积分$S_2$达到精度$\varepsilon$

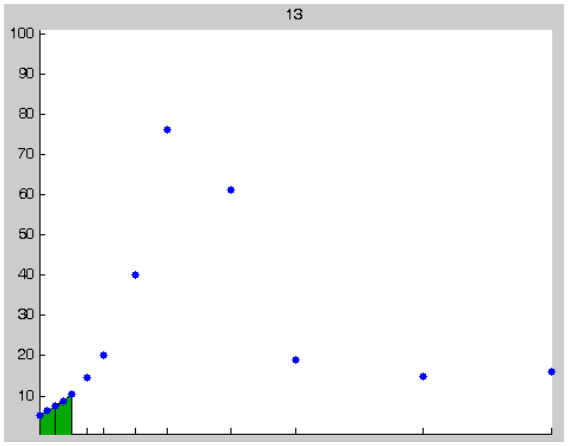
自适应积分原理示意

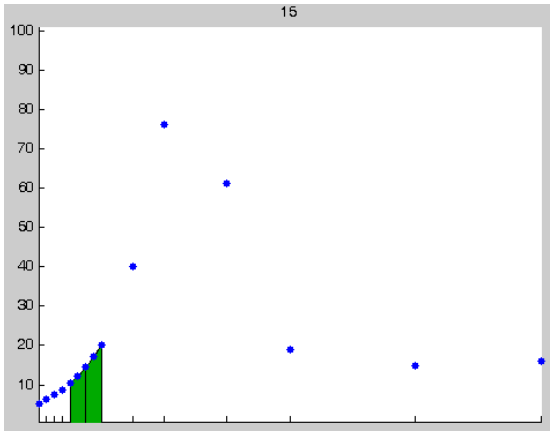

5
100
90
80
70
60
50
40
30
20
10


7
100
90
80
70
60
50
40
30
20
10


9
100
90
80
70
60
50
40
30
20
10


11
100
90
80
70
60
50
40
30
20
10


13
100
90
80
70
60
50
40
30
20
10


15
100
90
80
70
60
50
40
30
20
10

高斯求积方法

$$\int_a^b f(x)dx \approx \sum_{j=1}^{n} \omega_j f(x_j)$$

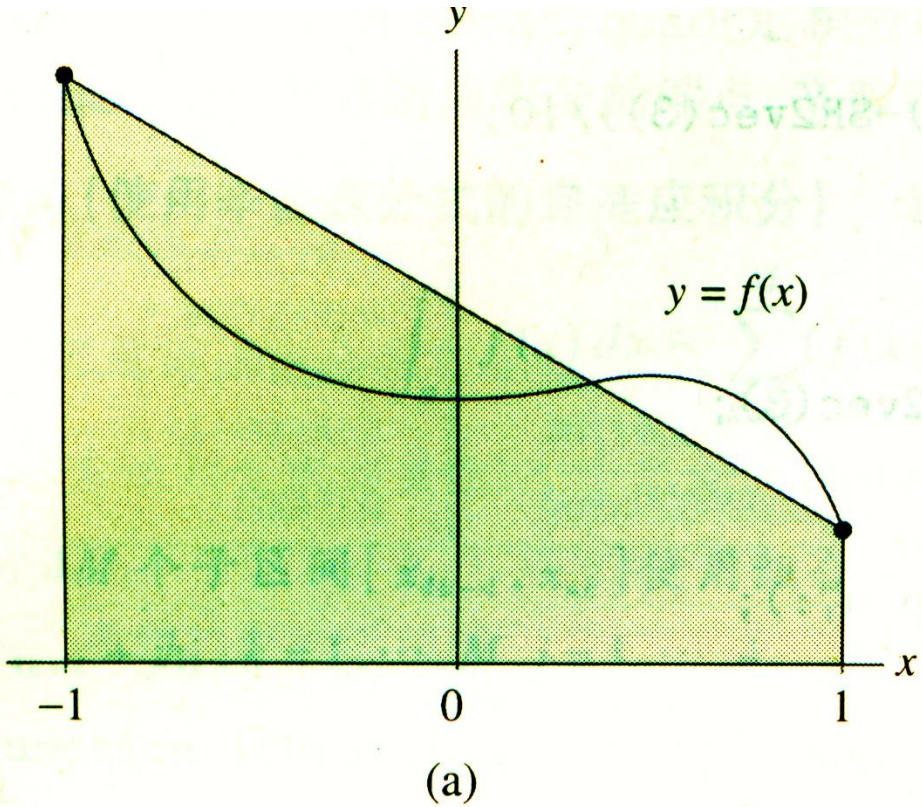
对于Newton-Cotes方法，求积公式的余项为：

$$R[f] = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (x - x_j)dx$$

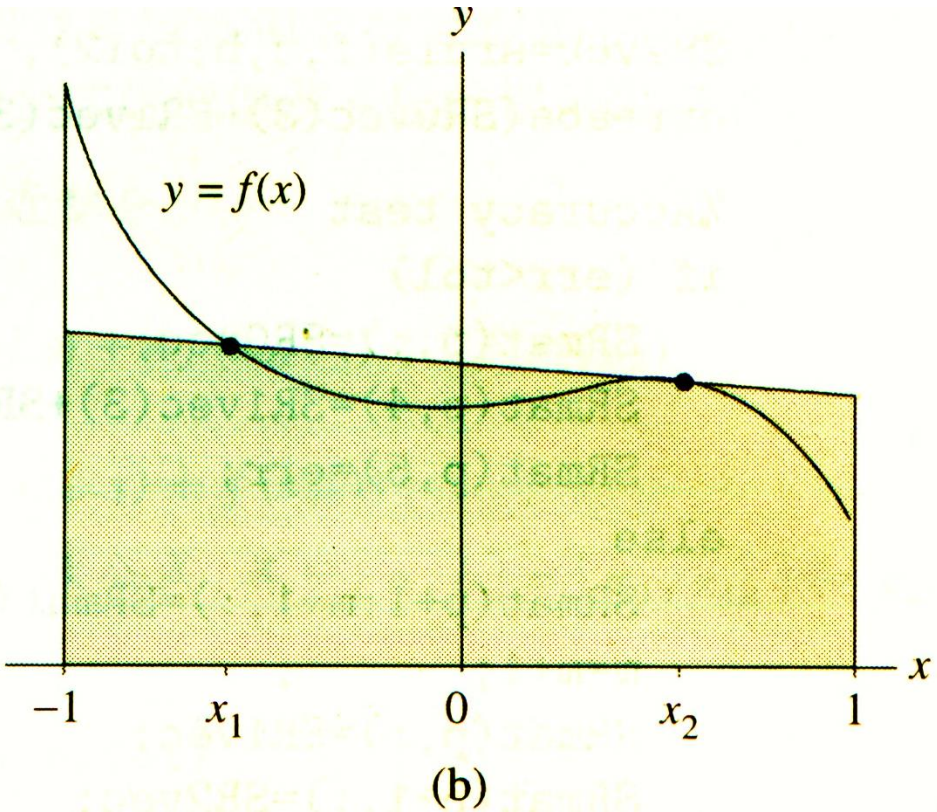
因此，当插值多项式不超过n次时，求积公式具有最少n次代数精度。

问题：对于以上求积公式，它可能具有的最高代数精度是多少？如何构造？

高斯求积方法



(a)



(b)

高斯求积法

- 过n+1个节点的插值型求积公式最高具有2n+1次代数精度。
- 高斯求积公式具有2n+1次代数精度，构造高斯求积公式关键是求高斯点

推导两点高斯公式 $\int_{-1}^{1} f(x)dx \approx A_1 f(x_1) + A_2 f(x_2)$

令以上公式对于f(x)=1,x,x^2,x^3准确成立

$$\begin{cases} A_1 + A_2 = 2 \\ A_1 x_1 + A_2 x_2 = 0 \\ A_1 x_1^2 + A_2 x_2^2 = \frac{2}{3} \\ A_1 x_1^3 + A_2 x_2^3 = 0 \end{cases}$$

求解非线性方程组

A_1=1,A_2=1

x_2=-x_1=1/sqrt(3)

$$\int_{-1}^{1} f(x)dx \approx f(\frac{1}{\sqrt{3}}) + f(-\frac{1}{\sqrt{3}})$$

高斯－勒让德公式

以高斯点x_k为零点的n次多项式称作勒让德多项式：

$$P_n(x) = (x - x_1)(x - x_2)\cdots(x - x_n)$$

一个n点高斯—勒让德求积公式具有如下形式：

$$\int_{-1}^{1} f(x)dx \approx \sum_{j=1}^{n} A_j f(x_j)$$

右边的$f(x_j)$是函数$f(x)$在节点x_j处的值，节点x_j是勒让德多项式$Pn(x)$的根。A_j为系数，其值为：

$$A_j = \frac{2}{[p_n'(x_j)]^2(1-x_j^2)}$$

式中P’n(x)是勒让德多项式Pn(x)的一阶导数

高斯－勒让德公式

$p_n(x)$的前几项表达式为

$$p_0(x)=1$$

$$p_1(x)=x$$

$$p_2(x)=\frac{1}{2}(3x^2-1)$$

$$p_3(x)=\frac{1}{2}(5x^3-3x)$$

$$p_4(x)=\frac{1}{8}(35x^4-30x^2+3)$$

.....

$$p_n(x)=(\frac{2n-1}{n})xp_{n-1}(x)-(\frac{n-1}{n})p_{n-2}(x)$$

由高斯—勒让德多项式得出的2～6点的根和系数见表5.5

高斯－勒让德公式

区间[a,b]内的高斯—勒让德求积公式

高斯—勒让德求积公式

$$\int_{-1}^{1} f(x)dx \approx \sum_{j=1}^{n} A_j f(x_j)$$

做变量代换：

$$x = \frac{1}{2}[(a+b)+(b-a)t]$$

也可转换成求区间[a,b]内的积分公式，得到如下结果：

$$\int_{a}^{b} f(x)dx \approx \frac{b-a}{2} \sum_{j=1}^{n} A_j f(\frac{a+b}{2} + \frac{b-a}{2} t_j)$$

MATLAB数值积分函数

MATLAB函数	公式
quad	自适应Simpson求积公式（低阶）
quadl	自适应Lobatto求积公式；精度高，最常用
trapz	梯形求积公式；速度快，精度差
cumtrapz	梯形法求一个区间上的积分曲线
cumsum	等宽距矩形法求一个区间上的积分曲线，精度很差
fnint	利用样条函数求不定积分；与spline，ppval配合使用，主要应用于表格“函数”积分

quad和quadl函数在新版本的MATLAB中被integral函数代替

cumsum函数

累积求和函数：cumsum( )

调用形式： Z = cumsum(A)

Z = cumsum(A,dim)

其中： 当A为向量时，返回对应A中各元素的累积和的向量

当A为矩阵时，按dim指定的维数返回各元素的累积和

```
A=cumsum(1:5)
```

A=1 3 6 10 15

累积求和值乘以步长则为矩形法求得的数值积分值

trapz函数

梯形法数值积分：trapz( )

调用形式：Z = trapz(X,Y)

其中:

X,Y—分别为长度相同的向量或数组，而Y与X的关系可以是函数型态（如$y=\sin(x)$）或是不以函数描述的离散型态；

Z—代表返回的积分值；

例题

采用trapz函数计算求下式积分：$\int_0^{\pi} \sin x dx$

```
x=0:0.1*pi:pi;
I=trapz(x,sin(x))
```

已知x=0:5;y=0:2:10,采用trapz函数计算积分 $\int_0^5 y dx$

```
x=0:5;
y=0:2:10;
I=trapz(x,y)
```

quad函数

自适应Simpson法数值积分：quad

基本调用格式：　　q=quad(fun,a,b)

或　q=quad(fun,a,b,tol,trace,p1,p2,…)

其中：　fun—被积函数。可以是匿名函数、m文件或函数句柄，它返回被积函数函数在指定点的函数值；函数表达式 中的必须使用点运算符号。

a, b—分别是积分的下限和上限；

q—积分结果。

tol—默认误差限，默认值为1.e-6.

trace-取0表示不用图形显示积分过程，非0表示用图形显示积分过程

p1,p2,…直接传递给函数fun的参数。

quadl函数

- 对于Gauss型积分公式，在实际计算中也常用复合法则和自适应法则。
- MATLAB基于Lobatto公式（高斯—勒让德公式）和自适应方法提供了quadl函数。
- quadl函数与quad函数的使用完全一致。

基本调用格式：

```
q=quadl(fun,a,b)
q=quadl(fun,a,b,tol,trace,p1,p2,…)
```

例题

求积分：$\int_2^5 \frac{\ln(x)}{x^2} dx$

1.采用匿名函数

```
>>f=@(x) log(x)./x.^2;
>> I1=quad(f,2,5)
```

2.采用匿名函数,调节求解精度为1e-2并显示积分过程：

```
>>f=@(x) log(x)./x.^2;
>>tol=1e-2
>> I1=quad(f,2,5,tol,1)
```

3.采用函数句柄：

```
function quadexp
I3=quad(@fun,2,5)
function y=fun(x)
y=log(x)./x.^2;
```

注意：被积函数一定要支持数组运算！

例题

求下式积分：

$$\int_1^2 \frac{dx}{\sin(x)-\ln(x)}$$

1.采用匿名函数

```
f=@(x) 1./(sin(x)-log(x));
I=quad(f,1,2)
```

2.采用函数句柄：

```
function quadexp
I=quad(@quadfun,1,2)
function y=quadfun(x)
y=1./(sin(x)-log(x));
```

例题

计算椭圆积分：

$$E(0.8,2\pi)=\int_0^{2\pi}\sqrt{(1-0,8\sin^2 t}dt$$

```
fun=@(x) sqrt(1-0.8*sin(x).^2);
format long
v1=quad(fun,0,2*pi,[],1)
v2=quadl(fun,0,2*pi,[],1)
```

```
v1 =   4.713959326612933
v2 =   4.713959697312330
```

通常quadl函数计算次数更多，但结果更精确。当计算任务中积分次数较少时，优选quadl函数。

例题－表格型(离散)函数的积分

真实气体的逸度f可用下式计算：

$$\lg f = \lg P - \frac{A}{2.303RT} \qquad A = \int_0^P \alpha dP, -\alpha = V - \frac{RT}{P}$$

α：真实气体的实测体积和按理想气体定律计算得到的体积之间的差值。

现测得0°C下氢气的有关数值如下表所示，试求1000 atm下的逸度。

P（atm）	$V \times 10^6 m^3$	$-\alpha = V - \frac{RT}{P}$	P（atm）	$V \times 10^6 m^3$	$-\alpha = V - \frac{RT}{P}$
0		15.46	600	53.43	16.09
100	239.51	15.46	700	48.14	16.13
200	127.49	15.46	800	44.17	16.16
300	90.29	15.61	900	41.06	16.16
400	71.86	15.85	1000	38.55	16.14
500	60.76	15.93			

例题

采用trapz函数求解

```
function Cha5demo5_1
P=0:100:1000;
a1=[15.46 15.46 15.46 15.61...
 15.85 15.93 16.09 16.13 16.16 16.16 16.14];
a1=-a1;
A=trapz(P,a1);
A=-A;
lf=log10(1000)+A./(2.303.*82.06.*273.2);
f=10.^lf
```

执行结果：f = 2.0290e+003

例题

采用quad函数求解

```
function Cha5demo5_2
P=0:100:1000;
a1=[15.46 15.46 15.46 15.61 15.85 15.93
16.09 16.13 16.16 16.16 16.14];
a1=-a1;
pp=pchip(P,a1);
plot(P,a1,'o',0:1000,ppval(pp,0:1000),'-')
A=quad(@ppval,0,1000,[],[],pp)
lf=log10(1000)-A./(2.303*82.06*273.2)
f=10.^lf
```

执行结果：f = 2.0290e+003

例题

氯仿-苯双组分精馏系统的气液平衡数据如下表所示。规定进料和塔顶的组成分别是$x_f = 0.4$，$x_d = 0.9$，精馏段的回流比为$R = 5$，精馏段理论板数的模型为

$$N = \int_{x_f}^{x_d} \frac{dx}{y - x - (x_d - y)/R}$$

试用MATLAB计算所需的精馏段理论板数。

x	0.178	0.275	0.372	0.456	0.650	0.844
y	0.243	0.382	0.518	0.616	0.795	0.931

例题

```
function Cha5demo6
xi=[0.178 0.275 0.372 0.456 0.650 0.844];
yi=[0.243 0.382 0.518 0.616 0.795 0.931];
plot(xi,yi,'*'),hold on
sp=pchip(xi,yi);
xplot=linspace(xi(1),xi(end),100);
yplot=ppval(sp,xplot);
plot(xplot,yplot,'-');
N=quad(@func1,0.4,0.9,[],[],sp);
N=ceil(N)
function f=func1(x,sp)
y=ppval(sp,x);
f=1./(y-x-(0.9-y)./5);
```

$$N=\int_{x_f}^{x_d}\frac{dx}{y-x-(x_d-y)/R}$$

例题

等压过程中使乙炔体系温度由t_1加热到t_2所需的热量Qp可按下式计算：

$$Q_p = \int_{t_1}^{t_2} C_P dt = \int_{t_1}^{t_2} (44.16 + 0.047t - 0.00002t^2) dt$$

用quadl函数计算从25℃加热到100℃所需的热量。

```
function QpCal
t1=25,t2=100;
Qp=quadl(@fun,t1,t2)
function y=fun(x)
y=44.16+0.047*x-0.00002*x.^2;
```

例题

等压过程中加热1 mol 乙炔，假定输入的热量为3000 J， 求乙炔可以从25℃上升到多少度。体系输入热量与温升的关系如下：

$$Q_p = \int_{t_1}^{t_2} C_P dt = \int_{t_1}^{t_2} (44.16 + 0.047t - 0.00002t^2) dt$$

本例实际求如下方程：

$$3000 - \int_{t1}^{t2} \left(44.16 + 0.047t - 0.00002t^2\right) dt = 0$$

这可视为一个非线性方程的求解问题。

```
function IsoPHeatT
f=@(t) 44.16+0.047*t-0.00002*t.^2;
fun=@(t) 3000-quad(f,25,t);
T=fzero(fun,25)
```

广义积分

1. 奇点积分

$$\int_0^1 \frac{dx}{\sqrt{x}(\exp(x)+1)}$$

```
fun=@(x) 1./(sqrt(x).*(exp(x)+1));
quad(fun,0,1)
quad(fun,eps,1)
```

2. 无穷积分

$$\int_0^{+\infty} \exp(\sin x - x^2/100)dx$$

可先选取一个有限的积分区间，如[0,100]计算；在选择一个较大的积分区间，如[0 200]计算，如两次计算结果的差满足一定的精度要求，则可认为此值即为无穷积分的值

本讲小结

1) quad quadl

   积分函数使用支持矩阵运算的运算符！

2) 表格型函数的数值积分方法

   a) 使用trapz函数

   b) 先使用插值生成样条函数，然后对样条函数进行积分

作业

公共邮箱下载文档：work10.pdf，直接打印、完成后上交