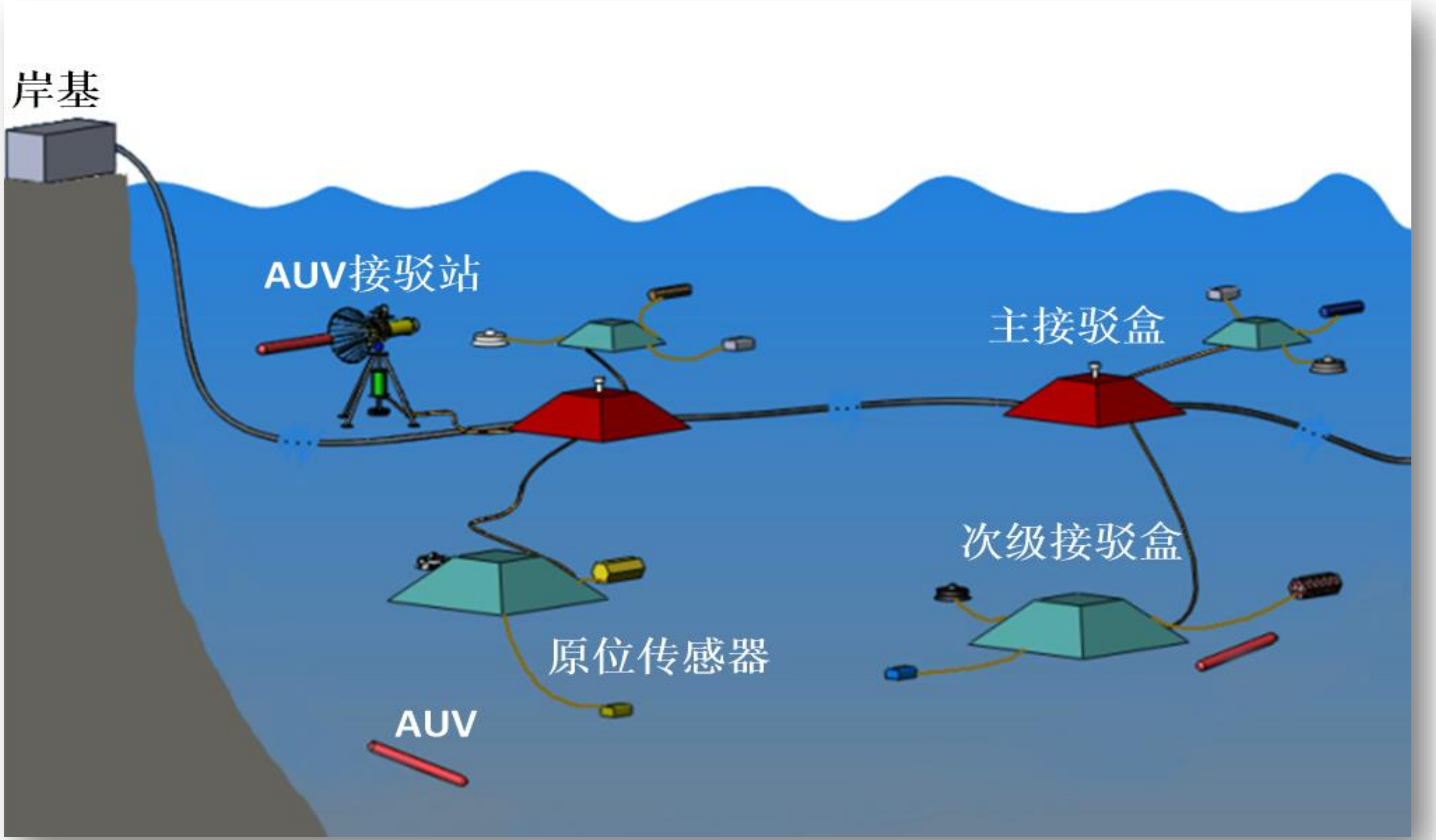
第6章 数值积分
Quadrature

苏 芮
srhello@zju.edu.cn
开物苑4-202

海底观测网移动接入拓展



目的：实现动/静结合，三维剖面/站位组合的海洋立体观测，拓展海洋观测范围。

海底观测网移动接入拓展

接驳基站与AUV艏部设计

- 接驳基站入口、转向结构与**AUV艏部外形**设计与优化
- 接驳基站**控制系统、电源**管理系统研发
- 接驳站**通讯系统**与上位机界面研发

AUV末端回坞视觉导航系统设计

- AUV末端回坞模式识别与**6自由度姿态求解**算法设计
- AUV末端**回坞路径规划**与控制方法设计

水下无线充电系统设计

- 充电线圈结构设计与仿真优化
- **海水环境**下系统最优**频率跟踪**算法与硬件电路设计
- 充电系统**散热**结构设计

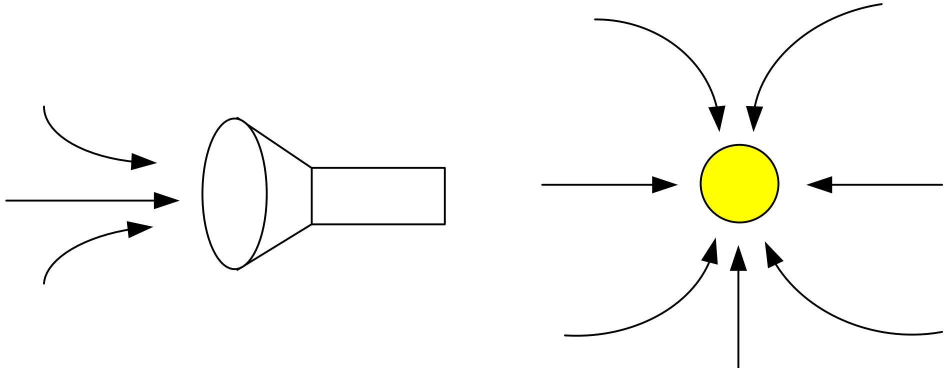
水下无线信号传输系统设计

- 天线结构与耐压封装设计
- 海水环境下信号传输衰减特性与**最优通讯距离**研究
- 无线信号传输系统链路与硬件设计

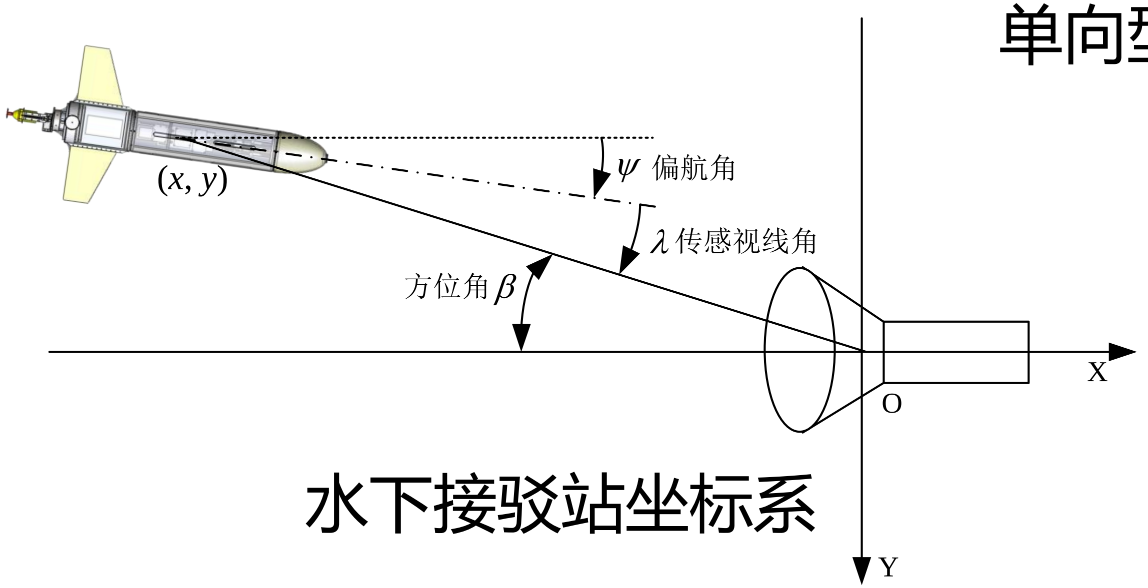
海底观测网移动接入拓展

水下接驳概述及回坞导航策略

水下接驳概述



单向型和全向型接驳站



水下接驳站坐标系

$$\beta = \psi + \lambda$$

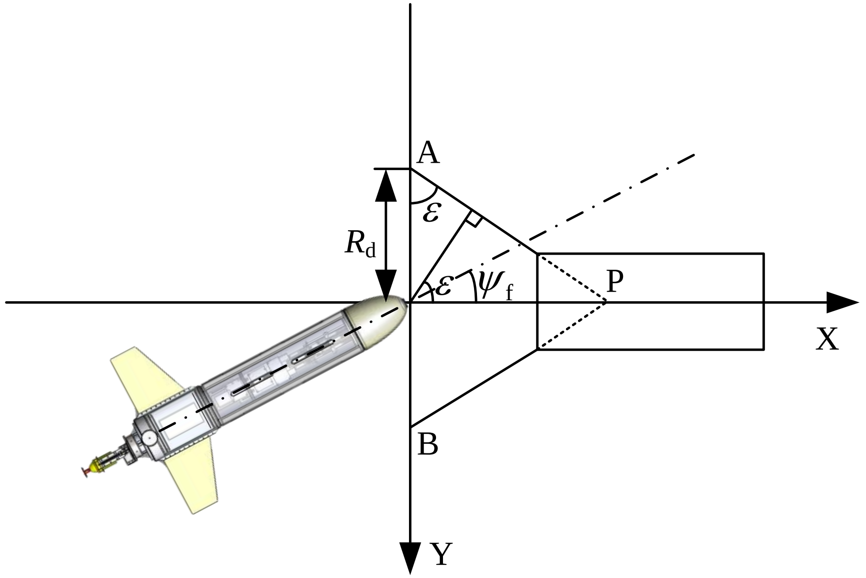
海底观测网移动接入拓展

水下接驳概述及回坞导航策略

入坞接驳问题数学描述

成功入坞的判定条件：

$$\begin{cases} \left|\psi_{\mathrm{f}}\right| < \varepsilon - \varepsilon_{\mathrm{t}} \\ \left|y_{\mathrm{f}}\right| < R_{\mathrm{d}} - R_{\mathrm{d,t}} \end{cases}$$

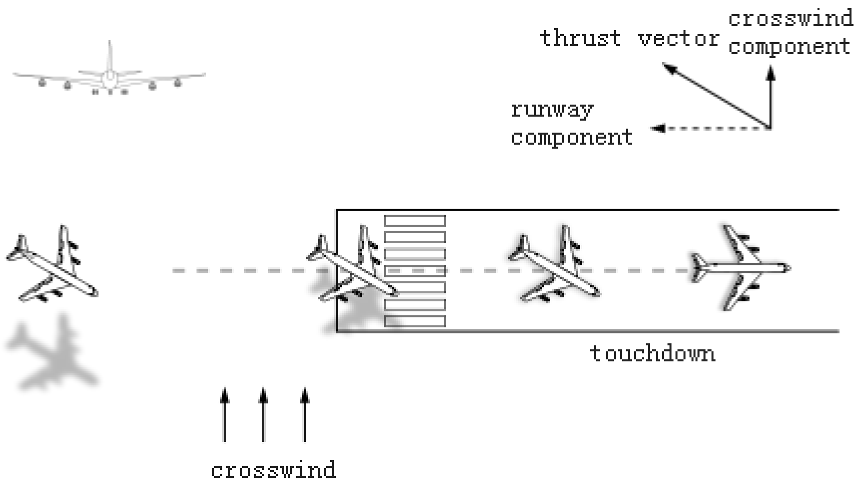


单向型接驳站的入坞问题

海底观测网移动接入拓展

基于可主动旋转接驳站的近端入坞引导算法分析

补偿洋流的跟踪算法

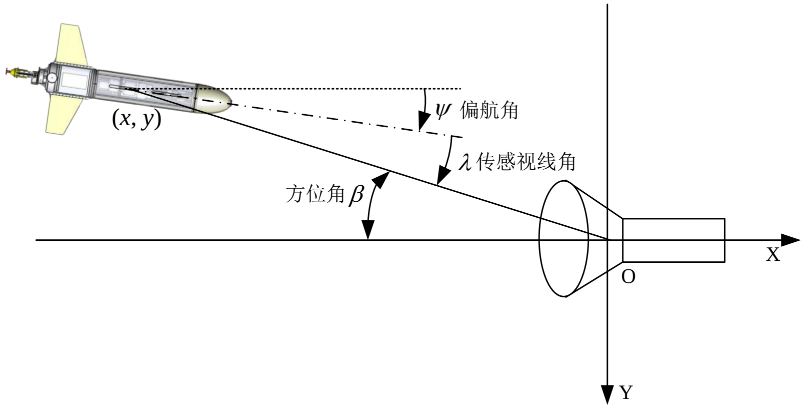


$$\psi_{crab} = \arcsin\frac{V_{fy}}{V_0}$$

$$\psi_c = k_c \times \psi_{crab}$$

$$\psi_d' = \begin{cases} \psi_d & \lambda > thresh \\ \psi_d' = \psi_d - k_c \times \arcsin\dfrac{V_{fy}}{V_0} & \lambda \le thresh \end{cases}$$

飞机在着陆时通过补偿偏航角补偿侧翼风



海底观测网移动接入拓展

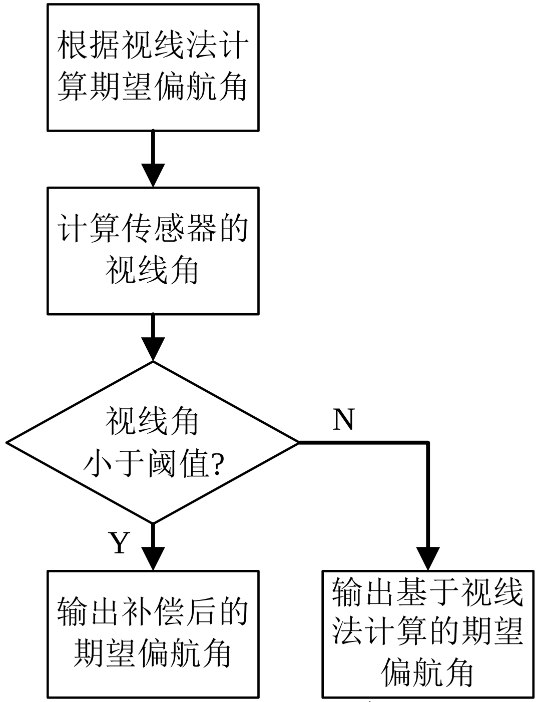
基于可主动旋转接驳站的近端入坞引导算法分析

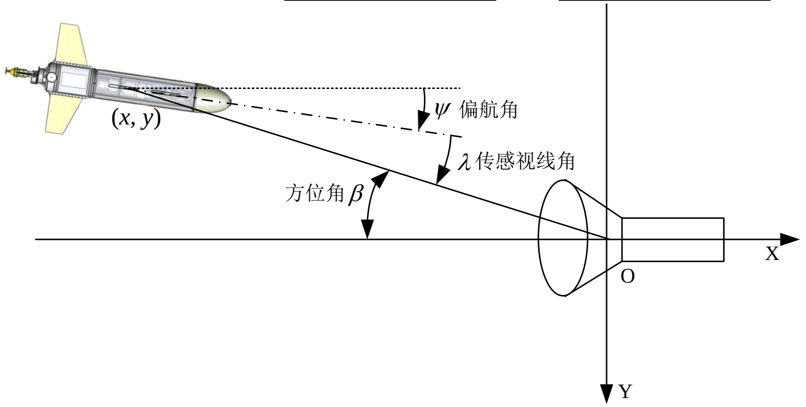
补偿洋流的跟踪算法

$$\psi_{\mathrm{crab}} = \arcsin \frac{V_{\mathrm{fy}}}{V_0}$$

$$\psi_{\mathrm{c}} = k_{\mathrm{c}} \times \psi_{\mathrm{crab}}$$

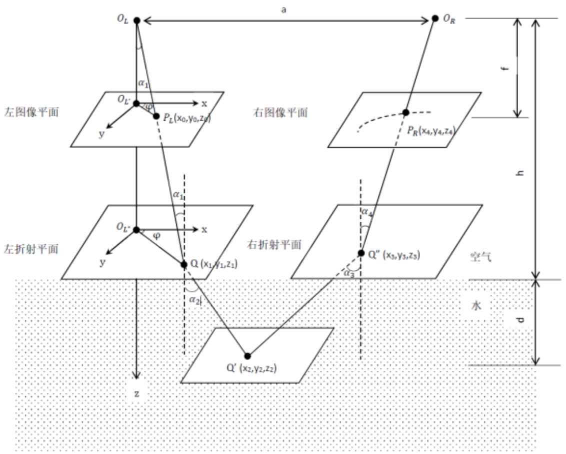
$$\psi_{\mathrm{d}}' = \begin{cases} \psi_{\mathrm{d}} & \lambda > \mathrm{thresh} \\ \psi_{\mathrm{d}}' = \psi_{\mathrm{d}} - k_{\mathrm{c}} \times \arcsin \dfrac{V_{\mathrm{fy}}}{V_0} & \lambda \le \mathrm{thresh} \end{cases}$$

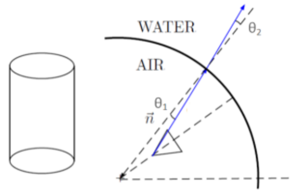


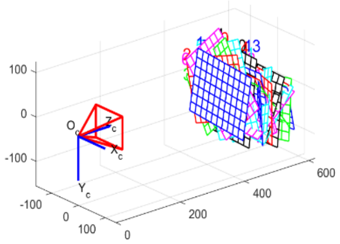


海底观测网移动接入拓展

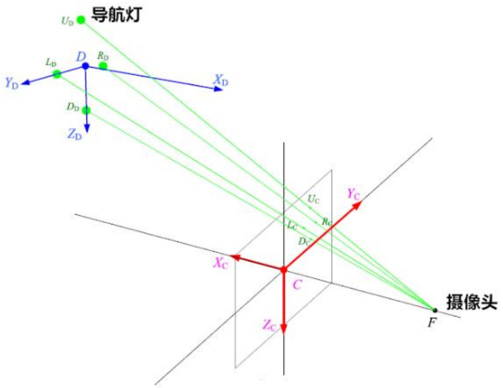
水下光学特性分析与校正

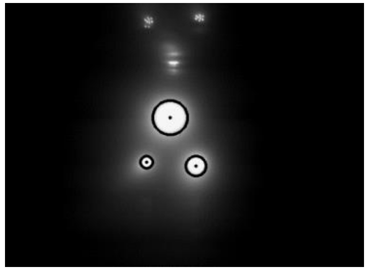






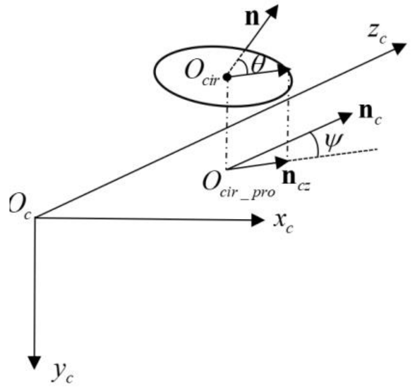
模式识别与特征提取

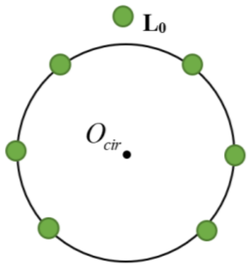




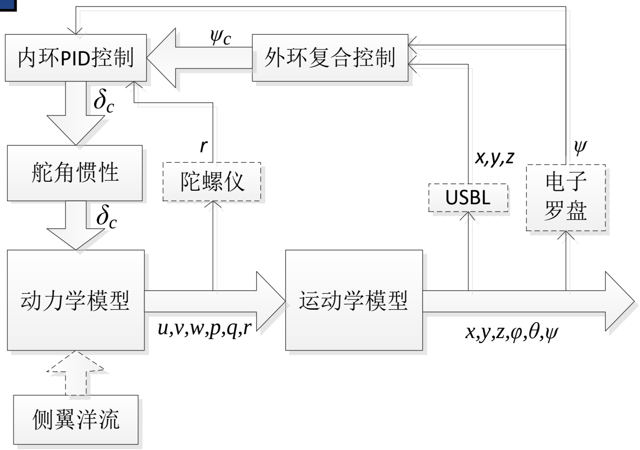
AUV回坞末端视觉导航技术

AUV姿态求解

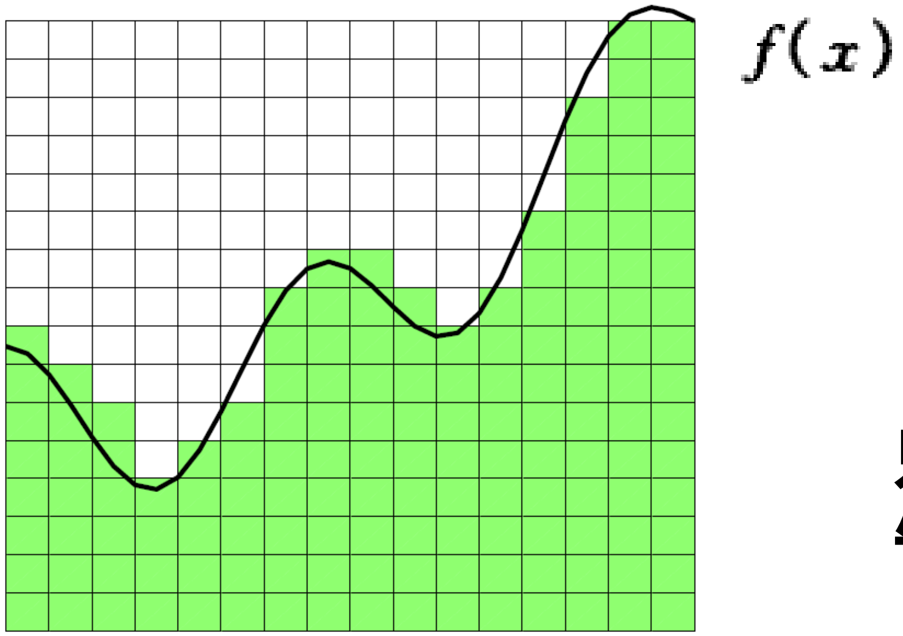




AUV路径规划



数值积分-问题来源



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

$$\int_a^b f(x)\mathrm{d}x$$

只要找到被积分函数f(x)的原函数F(x)，通过牛顿-莱布尼茨(Newton-Leibniz)公式：

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

但是实际使用这种求积方法往往很困难，原函数是不能使用初等函数来表达，所以不能使用上述公式。有时候即使求得被积函数的原函数，其公式非常复杂，导致积分的计算也很困难。

数值积分-问题来源

原函数是不能使用初等函数来表达:

$$\frac{\sin x}{x}\left(x \neq 0\right) \qquad e^{-x^2}$$

原函数公式非常复杂:

$$f(x)=\frac{1}{1+x^6}$$

$$F(x)=\frac{1}{3}\arctan x+\frac{1}{6}\arctan(x-\frac{1}{x})+\frac{1}{4\sqrt{3}}\ln\frac{x^2+x\sqrt{3}+1}{x^2-x\sqrt{3}+1}.$$

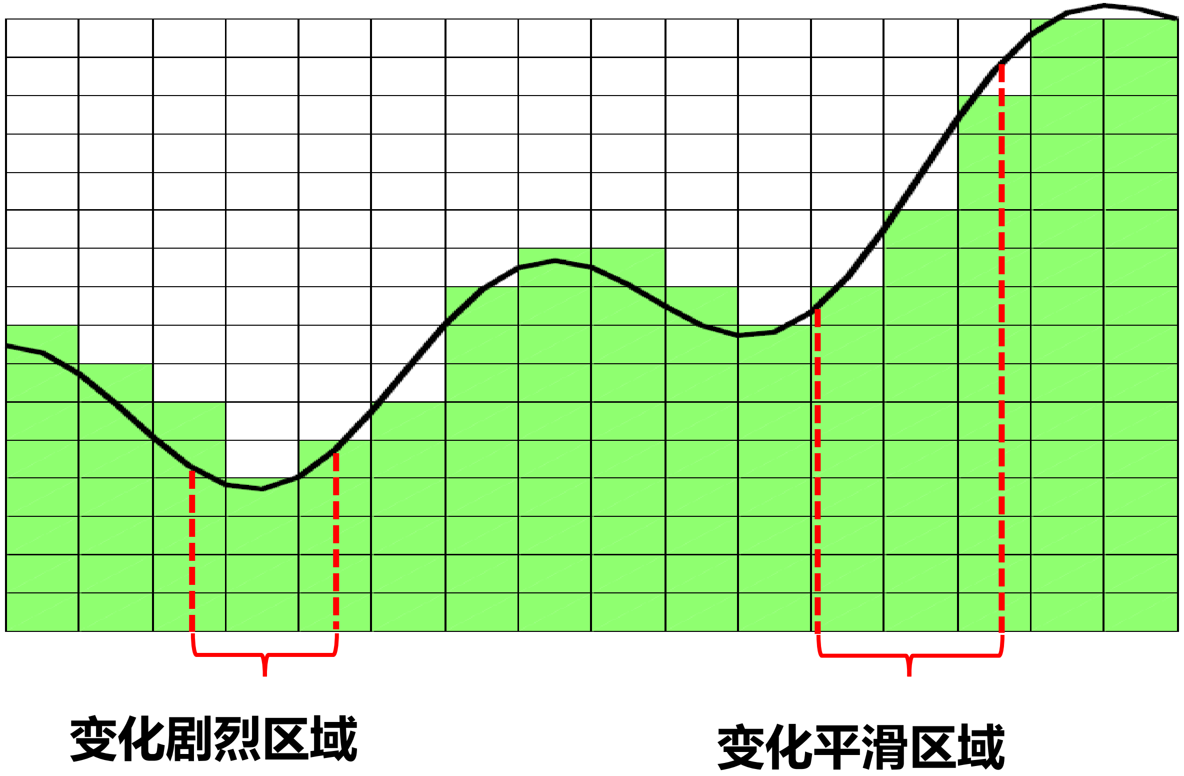
数值积分的基本思路:

通过求面积的方式得到来得到近似的积分值。

数值积分-问题来源

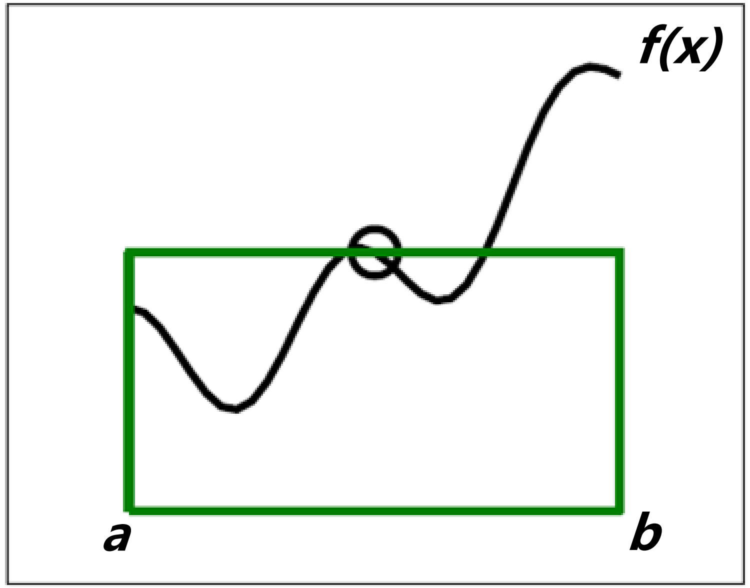
Adaptive Quadrature： 利用尽可能少的函数值，得到指定精度的积分近似值。

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx.$$

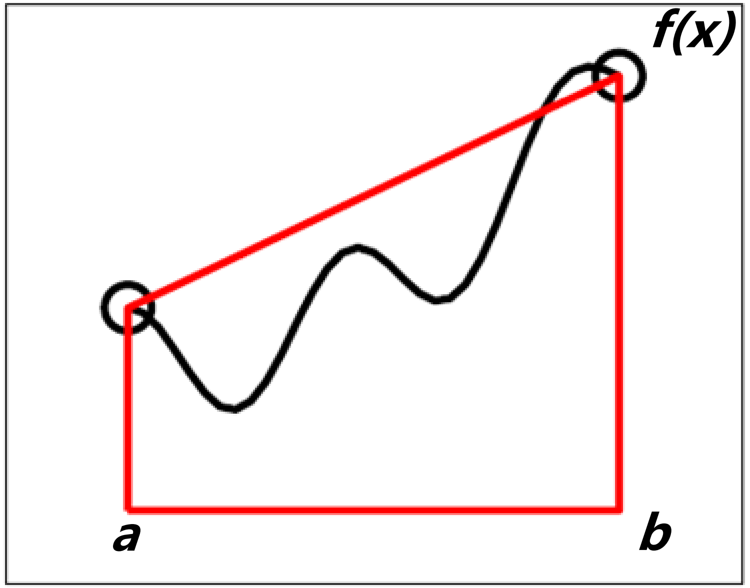


基本数值积分公式

中点公式



梯形公式



$$M = hf\left(\frac{a+b}{2}\right).$$

$$T = h\frac{f(a)+f(b)}{2}.$$

请注意没有计算/使用原函数F(x)

基本数值积分公式

$$\int_0^1 x^2 dx = \frac{1}{3}x^3\Big|_0^1 = \frac{1}{3}$$

$$M = hf(\frac{a+b}{2}) = 1\times(\frac{0+1}{2})^2 = \frac{1}{4}$$

误差为1/12

$$\boxed{M = hf\left(\frac{a+b}{2}\right).}$$

$$M = h(\frac{f(a)+f(b)}{2}) = 1\times(\frac{0+1}{2}) = \frac{1}{2}$$

误差为-1/6

$$\boxed{T = h\frac{f(a)+f(b)}{2}.}$$

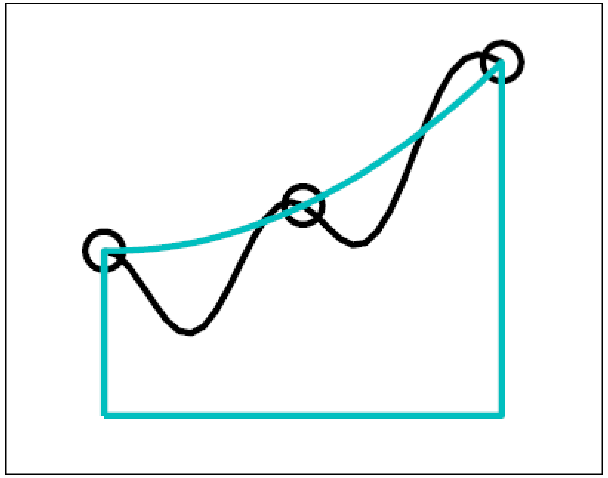
基本数值积分公式

辛普森法则：上述现象具有普遍性 - 中点法则积分结果M是梯形法则积分结果T准确性的-2倍

新的积分结果S
S-T=-2(S-M)

得到S=2/3M+1/3T（辛普森积分法则）

Simpson's rule



$$S = \frac{h}{6}(f(a) + 4f(c) + f(b)).$$

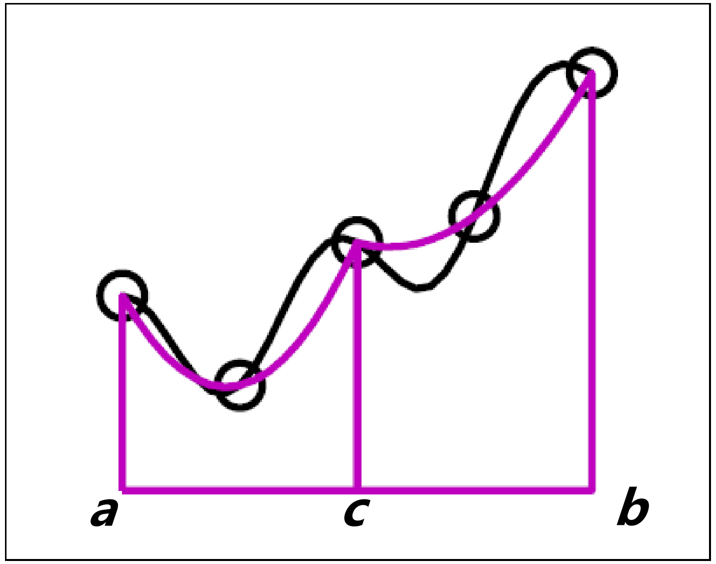
可证明：为对a, b, (a+b)/2 三点的插值积分（拉格朗日插值）

基本数值积分公式

复合辛普森法则: 将面积 S 再分成两半$[a,c],[c,b]$, 令 d 和 e 分别为两半的中点$d=(a+c)/2$, $e=(c+d)/2$,在两个区间上再使用辛普森法则，可得:

$$S_2=\frac{h}{12}(f(a)+4f(d)+2f(c)+4f(e)+f(b))$$

Composite Simpson's rule



基本数值积分公式

辛普森法则:S
复合辛普森法则:S_2

S和S_2都是四阶精度，S的步长是S_2的1/2，因此精度是1/16

$$Q - S = 16(Q - S_2).$$

$$Q = S_2 + (S_2 - S)/15.$$

六阶（5个插值点）的牛顿-科特斯法则
(Newton-Cotes Ruler)

上节课知识回顾

1. 方程求根

- 牛顿法的缺点？（需要计算fprim）
- 割线法的基本思路？（利用二个数值点，计算斜率，替代fprim）
- 逆二次插值法的基本思路？（利用三个数值点，进行多项式插值，计算和x轴的交点）
- Zeroin算法的基本思路？（将二分法的可靠性和割线法及IQI算法的收敛速度结合起来）

2. 数值积分

- 问题来源？基本思路？
- 中点公式？梯形公式？辛普森公式？复合辛普森公式？牛顿-科特斯公式？
- 如何构建辛普森公式和牛顿-科特斯公式？

数值积分公式的精度

辛普森法则： $S = \frac{h}{6}\left(f(a) + 4f(c) + f(b)\right).$

数值积分是通过离散点上函数值的线性组合近似计算积分

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i) = I_n(f)$$

称为求积节点，与f(x)无关

称为求积系数，与f(x)无关，与积分区间和求积节点有关

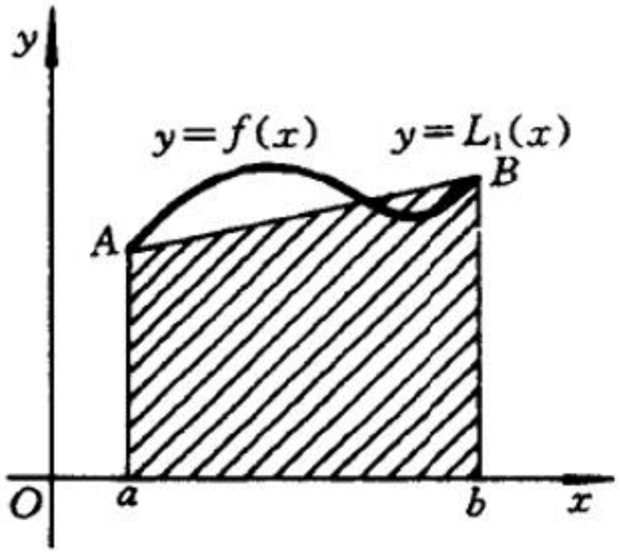
$$E_n(f) = I - I_n(f)$$

求积公式的余项

数值积分公式的精度

若求积公式 $\int_a^b f(x)dx \approx \sum_{k=1}^{n} A_k f(x_k)$

对任何次数不高于**m**次的代数多项式都准确成立，但对于x^{m+1}却不能准确成立，则求积公式的代数精度是m



$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a)+f(b)]$$

但是，当 $f(x)=x^2$ 时，

$$左端=\int_a^b x^2\mathrm{d}x=\frac{1}{3}(b^3-a^3)$$

$$右端=\frac{b-a}{2}[a^2+b^2]$$

$$左端\neq右端 \quad (a\neq b)$$

因此梯形法则的代数精度为 1

数值积分公式的精度

例 3 确定求积公式

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx A_{-1}f(-1) + A_0 f(0) + A_1 f(1)$$

中的系数，使其具有尽可能高的代数精度.

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i) = I_n(f)$$

称为求积节点，与$f(x)$无关

称为求积系数，与$f(x)$无关，与积分区间和求积节点有关

$$E_n(f) = I - I_n(f)$$

求积公式的余项

辛普森积分法则

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1)$$

定理 1 含有 $n+1$ 个节点的插值型数值积分公式的代数精度至少是 n.

数值积分公式的其他方法

•牛顿-科特斯（ Newton-Cotes)

•考虑到计算上的方便，常将积分区间等分之，并取分点为求积节点。这样构造出来的插值型求积公式称为Newton-Cotes积分。

•龙贝格（Romberg)

外推算法: 用若干个精度较低的积分近似值来推算更精确近似值的方法。这样构造出来的递进式求积公式称为Romber积分。

插值型积分公式

在积分区间$[a,b]$上取一组点

$$a \le x_0 \le x_1 \le \cdots \le x_n \le b$$

作$f(x)$的n次Lagrange插值多项式

$$L_n(x) = \sum_{i=0}^{n} l_i(x) f(x_i)$$

$$l_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} = \prod_{\substack{j=0\\ j\ne i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

其中$l_i(x)(k=0,1,\ldots,n)$为n次插值基函数

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{k=0}^{n} A_k f(x_k)$$

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

$$A_k = \int_a^b l_k(x)\mathrm{d}x = \int_a^b \frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}\mathrm{d}x$$

插值型积分公式

数值积分余项

$$R[f]=\int_a^b f(x)\mathrm{d}x-\sum_{k=0}^{n}A_k f(x_k)$$

$$R_n[f]=\int_a^b f(x)\mathrm{d}x-\int_a^b L_n(x)\mathrm{d}x=\int_a^b [f(x)-L_n(x)]\mathrm{d}x$$

$$R_n[f]=\int_a^b \frac{f^{(n+1)}(\zeta)}{(n+1)!}\omega_{n+1}(x)\mathrm{d}x$$

定理 2 若 $f(x)$在区间$[a,b]$上有直到 $n+1$ 阶导数，$p_n(x)$为 $f(x)$在 $n+1$ 个节点 $x_i\in[a,b](i=0,1,\cdots,n)$上的 n 次插值多项式，则对任何 $x\in[a,b]$有

$$R_n(x)=\frac{f^{(n+1)}(\zeta)}{(n+1)!}\omega_{n+1}(x) \tag{4.6}$$

其中 $\omega_{n+1}(x)=\prod_{i=0}^{n}(x-x_i)$，$\zeta\in(a,b)$且依赖于 x.

Newton-Cotes 积分

- 将积分区间[a, b]n等分，取分点

$$x_i = a + ih \quad \left(h = \frac{b-a}{n}, i = 0, \cdots, n \right)$$

作为求积节点，并作变量替换 $x = a + th$ ，则求积系数

$$A_i = \int_a^b l_i(x)dx = \int_0^n \frac{t(t-1)\cdots(t-i+1)(t-i-1)\cdots(t-n)}{i!(n-i)!(-1)^{n-i}} hdt$$

$$= nh \cdot \frac{(-1)^{n-i}}{i!(n-i)!n} \int_0^n t(t-1)\cdots(t-i+1)(t-i-1)\cdots(t-n)dt$$

$$\int_a^b f(x)\mathrm{d}x \approx (b-a) \sum_{k=0}^{n} C_k^{(n)} f(x_k)$$

Cotes系数

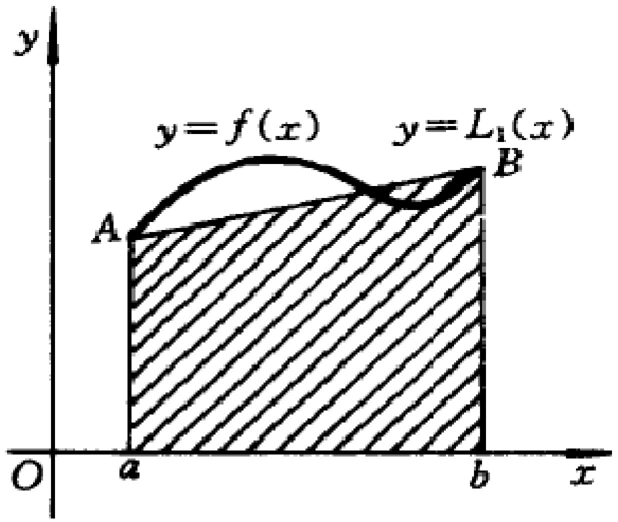
$$\boxed{C_k^{(n)} = \frac{(-1)^{n-k}}{n \cdot k! \cdot (n-k)!} \int_0^n t(t-1)\cdots(t-k+1)(t-k-1)\cdots(t-n)\mathrm{d}t}$$

Newton-Cotes 积分

$$C_k^{(n)}=\frac{(-1)^{n-k}}{n\cdot k!\ \cdot(n-k)!}\int_0^n t(t-1)\cdots(t-k+1)(t-k-1)\cdots(t-n)\mathrm{d}t$$

梯形rule n=1

当 $n=1$ 时有



$$C_0^{(1)}=-\int_0^1(t-1)\mathrm{d}t=\frac{1}{2}$$

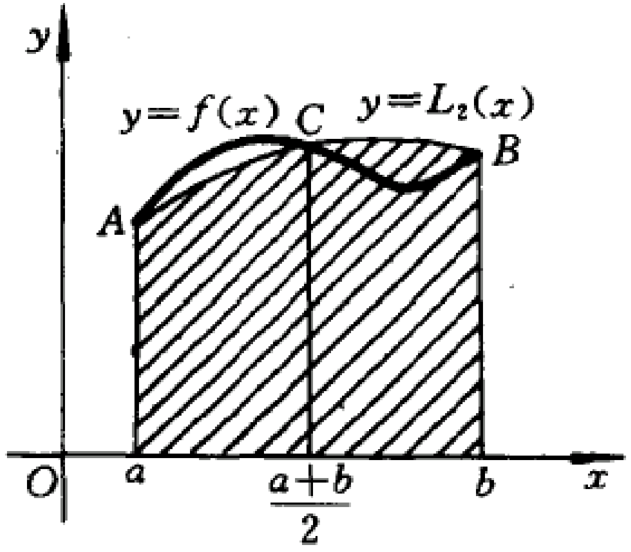
$$C_1^{(1)}=\int_0^1 t\,\mathrm{d}t=\frac{1}{2}$$

$$\int_a^b f(x)\mathrm{d}x\approx\frac{b-a}{2}[f(a)+f(b)]$$

Newton-Cotes 积分

$$C_k^{(n)} = \frac{(-1)^{n-k}}{n \cdot k! \cdot (n-k)!} \int_0^n t(t-1)\cdots(t-k+1)(t-k-1)\cdots(t-n)\mathrm{d}t$$

辛普森rule n=2



$$C_0^{(2)} = \frac{1}{4}\int_0^2 (t-1)(t-2)\mathrm{d}t = \frac{1}{6}$$

$$C_1^{(2)} = -\frac{1}{2}\int_0^2 t(t-2)\mathrm{d}t = \frac{4}{6}$$

$$C_2^{(2)} = \frac{1}{4}\int_0^2 t(t-1)\mathrm{d}t = \frac{1}{6}$$

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Newton-Cotes 积分

$$C_k^{(n)} = \frac{(-1)^{n-k}}{n \cdot k! \cdot (n-k)!} \int_0^n t(t-1)\cdots(t-k+1)(t-k-1)\cdots(t-n)\mathrm{d}t$$

3/8辛普森rule n=3

$$\int_a^b f(x)dx \approx \frac{(b-a)}{8}(f(x_0)+3f(x_1)+3f(x_2)+f(x_3))$$

保尔（Boole’s） rule n=4

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{90}[7f(x_0)+32f(x_1)$$

$$+12f(x_2)+32f(x_3)+7f(x_4)]$$

Newton-Cotes 积分

Cotes系数

n	$C_k^{(n)} \quad k=0,\ldots,n$									
1	1	1								/2
2	1	4	1							/6
3	1	3	3	1						/8
4	7	32	12	32	7					/90
5	19	75	50	50	75	19				/288
6	41	216	27	272	27	216	41			/840
7	751	3577	1323	2989	2989	1323	3577	751		/17280
8	989	5888	-928	10496	-4540	10496	-928	5888	989	/28350

Cotes公式

Newton-Cotes 积分

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

利用Newton-Cotes公式计算函数在[0, 0.8]上的积分（精确值为**1.640533**）

梯形rule n=1

f(0)=0.2
f(0.8)=0.232

$$I_1 \cong (0.8-0)\frac{1}{2}(0.2+0.232) = 0.1728$$

辛普森rule n=2

f(0)=0.2
f(0.4)=2.456
f(0 8)=0.232

$$I \cong \frac{0.8}{6}(0.2+4(2.456)+0.232) = 1.367467$$

Newton-Cotes 积分

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

利用Newton-Cotes公式计算函数在[0, 0.8]上的积分（精确值为**1.640533**）

Simpson 3/8 rule n=3

f(0)=0.2;
f(0.2667)=1.432724;
f(0.5333)=3.487177;
f(0.8)=0.232

$$I \cong \frac{0.8}{8}(0.2 + 3(1.432724 + 3.487177) + 0.232) = 1.519170$$

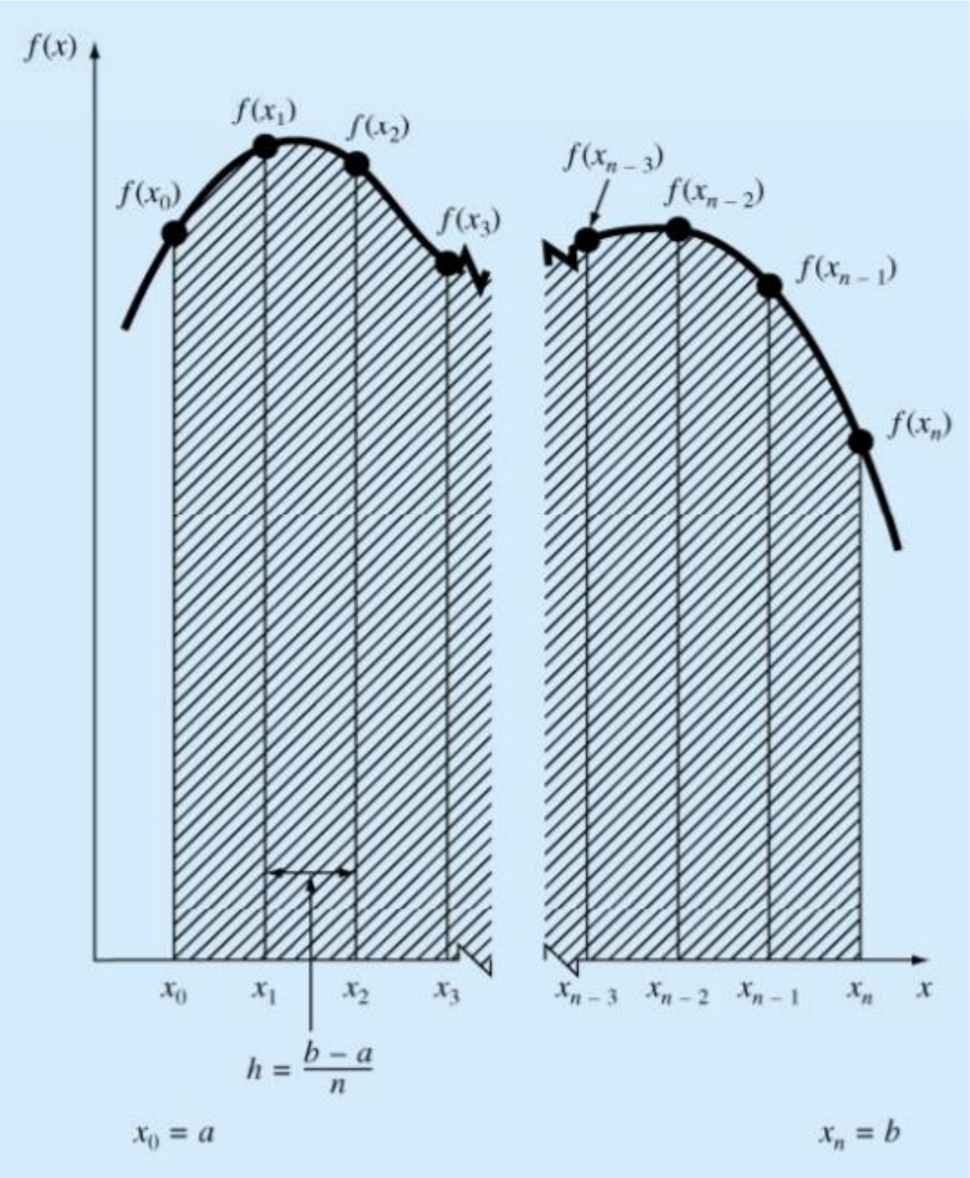
Newton-Cotes 积分

公式余项 $$R_n[f]=\int_a^b \frac{f^{(n+1)}(\zeta)}{(n+1)!}\omega_{n+1}(x)\mathrm{d}x$$

- 梯形公式余项：$R_1[f]=-\frac{(b-a)^3}{12}f''(\zeta_1)$
- 辛普森公式余项：$R_2[f]=-\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\zeta_2)$
- 科茨公式余项：$R_4[f]=-\frac{8}{945}\left(\frac{b-a}{4}\right)^7 f^{(6)}(\zeta_4)$

代数精度？ **当积分区间比较大时，精度？**

复合Newton-Cotes 积分



当积分区间[a,b]较大时，直接使用Newton-Cotes公式所得积分近似值的精度很难得到保证。

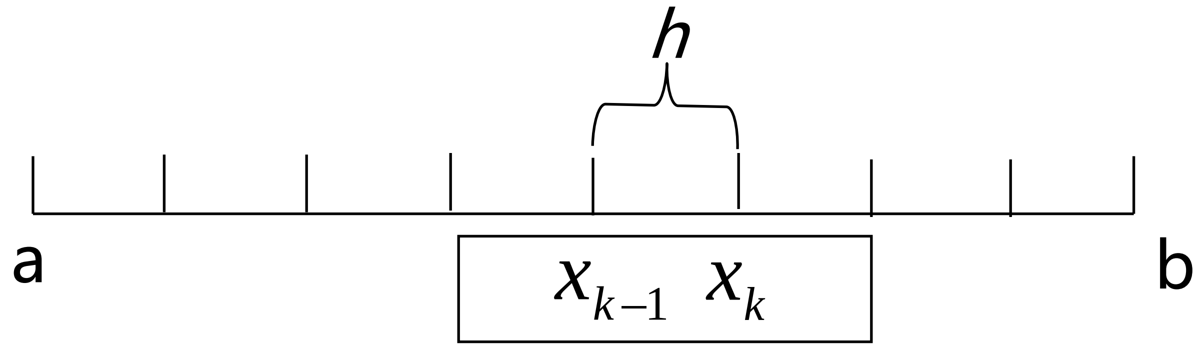
为了保证精度，采用复合求积的方法：先将积分区间分成几个小区间，并在每个小区间上用低阶Newton-Cotes公式计算积分的近似值，然后对这些近似值求和，从而得到所求积分的近似值。

复合Newton-Cotes 积分

Composite Trapezoidal Rule

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a)+f(b)] \qquad I_k = \int_{x_{k-1}}^{x_k} f(x)\mathrm{d}x$$

$$I_k = \frac{x_k - x_{k-1}}{2}[f(x_{k-1})+f(x_k)] = \frac{h}{2}[f(x_{k-1})+f(x_k)]$$

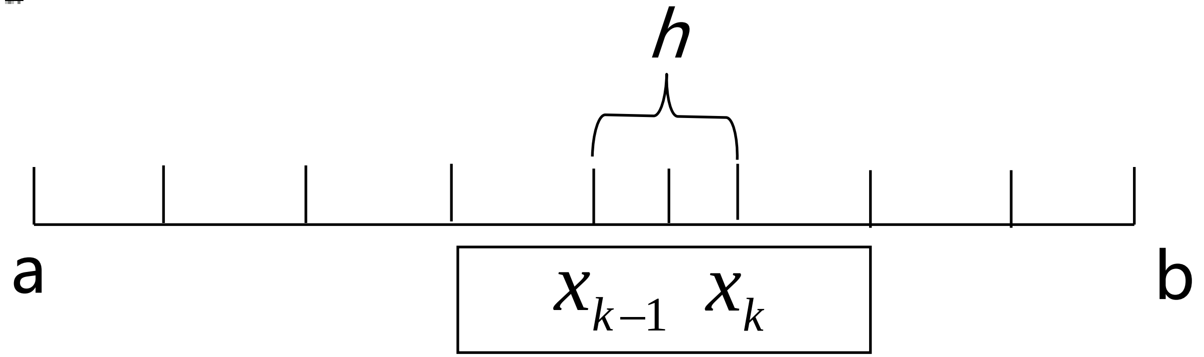


$$\int_a^b f(x)\mathrm{d}x \approx T_n = \frac{h}{2}\left[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)\right]$$

复合Newton-Cotes 积分

Composite Simpson Rule

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right]$$

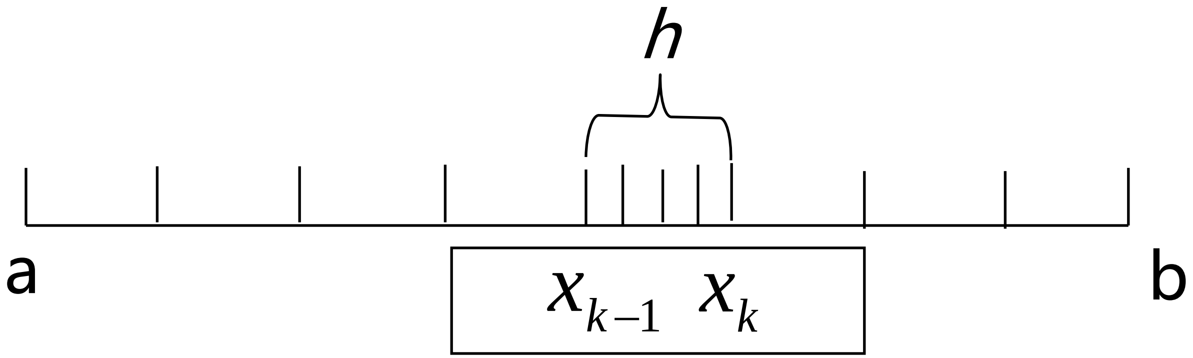


$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{6}\left[f(a)+4\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}})+2\sum_{k=1}^{n-1} f(x_k)+f(b)\right] \xlongequal{(记)} S_n$$

复合Newton-Cotes 积分

Composite Bool’s Rule

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{90}[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)]$$



$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{90}\Big[7f(a)+32\sum_{k=0}^{n-1} f(x_{k+\frac{1}{4}})+12\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}})$$

$$+32\sum_{k=0}^{n-1} f(x_{k+\frac{3}{4}})+14\sum_{k=1}^{n-1} f(x_k)+7f(b)\Big] \overset{(记)}{=\!=} C_n$$

复合Newton-Cotes 积分

- 复合梯形公式余项

$$\int_a^b f(x)\mathrm{d}x - T_n = -\frac{b-a}{12}h^2 f''(\eta_1)$$

  - 梯形公式余项

$$R_1[f] = -\frac{(b-a)^3}{12}f''(\zeta_1)$$

- 复合辛普森公式余项

$$\int_a^b f(x)\mathrm{d}x - S_n = -\frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\eta_2)$$

  - 辛普森公式余项

$$R_2[f] = -\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\zeta_2)$$

- 复合布尔公式余项

$$\int_a^b f(x)\mathrm{d}x - C_n = -\frac{2(b-a)}{945}\left(\frac{h}{4}\right)^6 f^{(6)}(\eta_4)$$

  - 布尔公式余项

$$R_4[f] = -\frac{8}{945}\left(\frac{b-a}{4}\right)^7 f^{(6)}(\zeta_4)$$

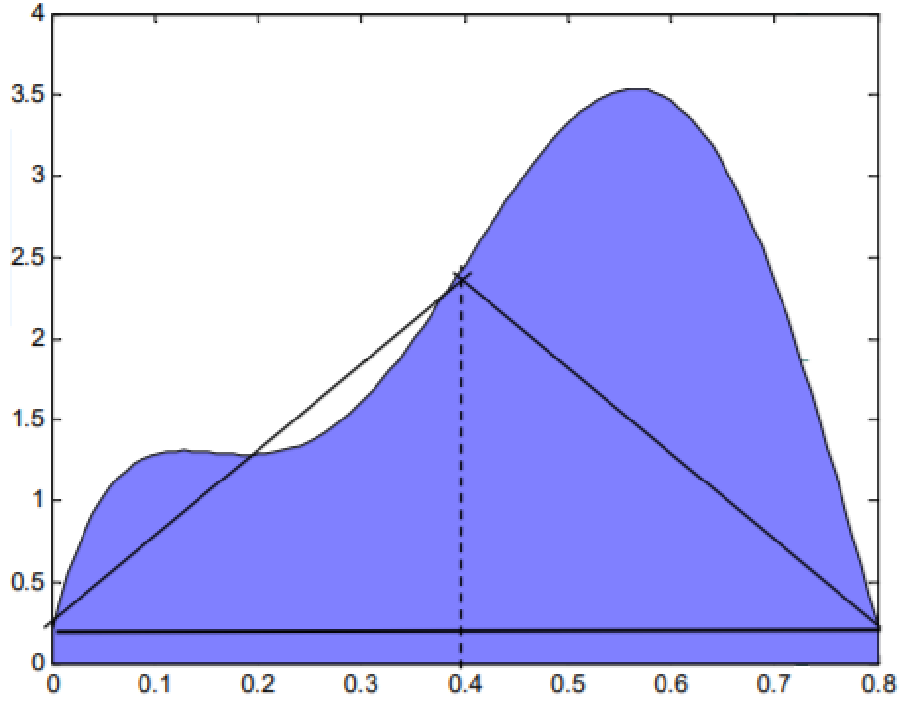
复合Newton-Cotes 积分

$$f(x)=0.2+25x-200x^2+675x^3-900x^4+400x^5$$

利用复合梯形公式计算函数在[0, 0.8]上的积分（精确值为**1.640533**）

$f(0)=0.2$　$f(0.4)=2.456$　$f(0.8)=0.232$

$$I_1 \cong 0.8\frac{(0.2+2(2.456)+0.232)}{2*2}=1.0688$$



$$\int_a^b f(x)\mathrm{d}x \approx T_n=\frac{h}{2}[f(a)+2\sum_{k=1}^{n-1}f(x_k)+f(b)]$$

复合Newton-Cotes 积分

$$f(x)=0.2+25x-200x^2+675x^3-900x^4+400x^5$$

（精确值为**1.640533**）

- 复合梯形公式的计算结果

n	h	I	$\varepsilon_t(\%)$
2	0.4	1.0688	34.9
3	0.2667	1.3695	16.5
4	0.2	1.4848	9.5
5	0.16	1.5399	6.1
6	0.1333	1.5703	4.3
7	0.1143	1.5887	3.2
8	0.1	1.6008	2.4
9	0.0889	1.6091	1.9
10	0.08	1.6150	1.6

子区间数目增加时，误差随之减小。误差与n^2成反比。

$$E_n(f)=-\frac{(b-a)^3}{12n^2}f''(\xi)$$

复合Newton-Cotes 积分

例 4 利用复合辛普森公式计算积分 $I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$ 的近似值，使截断误差不超过 $\frac{1}{2}\times 10^{-6}$，并用同样点按复合梯形公式和复合科茨公式重新计算近似值.

解 首先应根据精度要求，确定区间 $[a,b]$ 的等分数 n. 由于

$$f(x)=\frac{\sin x}{x}=\int_0^1 \cos tx\,\mathrm{d}t$$

故

$$f^{(k)}(x)=\int_0^1 \frac{\mathrm{d}^k}{\mathrm{d}x^k}(\cos tx)\mathrm{d}t=\int_0^1 t^k\cos\left(tx+\frac{k\pi}{2}\right)\mathrm{d}t$$

$$|f^{(k)}(x)|\leqslant\int_0^1 t^k\mathrm{d}t=\frac{1}{k+1}$$

根据复合辛普森公式的余项表达式(5.31)，为满足精度要求，需 n 满足

$$\boxed{\left|\frac{b-a}{2880}h^4f^{(4)}(\zeta)\right|=\frac{1}{2880}\times\frac{1}{n^4}\times|f^{(4)}(\zeta)|<\frac{1}{2}\times 10^{-6}}$$

这只需

$$\frac{1}{2880}\times\frac{1}{n^4}\times\frac{1}{5}<\frac{1}{2}\times 10^{-6}$$

即 $n\geqslant 4$. 取 $n=4$，可得

$$I\approx S_4=\frac{1}{6}\times\frac{1}{4}\left\{f(0)+4\left[f\left(\frac{1}{8}\right)+f\left(\frac{3}{8}\right)+f\left(\frac{5}{8}\right)+f\left(\frac{7}{8}\right)\right]\right.$$

$$\left.+2\left[f\left(\frac{1}{4}\right)+f\left(\frac{1}{2}\right)+f\left(\frac{3}{4}\right)\right]+f(1)\right\}$$

$$=0.9460832$$

复合Newton-Cotes 积分

对同样九个点上函数值(见表 5－3),若用复合梯形公式与复合科茨公式进行计算,则所得近似值分别为

$$I \approx T_8 = \frac{1}{2} \times \frac{1}{8} \{ f(0) + 2[f(\frac{1}{8}) + f(\frac{1}{4}) + f(\frac{3}{8})$$

$$+ f(\frac{1}{2}) + f(\frac{5}{8}) + f(\frac{3}{4}) + f(\frac{7}{8})] + f(1) \} = 0.9456909$$

$$I \approx C_2 = \frac{1}{90} \times \frac{1}{2} \{ 7f(0) + 32[f(\frac{1}{8}) + f(\frac{3}{8}) + f(\frac{5}{8}) + f(\frac{7}{8})]$$

$$+ 12[f(\frac{1}{4}) + f(\frac{3}{4})] + 14f(\frac{1}{2}) + 7f(1) \} = 0.9460829$$

表 5－3

x	$f(x) = \frac{\sin x}{x}$	x	$f(x) = \frac{\sin x}{x}$
0	1.0000000	$\frac{5}{8}$	0.9361556
$\frac{1}{8}$	0.9973978	$\frac{3}{4}$	0.9088516
$\frac{1}{4}$	0.9896158	$\frac{7}{8}$	0.8771925
$\frac{3}{8}$	0.9767267	1	0.8414709
$\frac{1}{2}$	0.9588510		

复合公式的递推公式

梯形公式 $T(f,h)$ 和 $T(f,2h)$ 满足如下关系：

$$T(f,h)=\frac{T(f,2h)}{2}+h\sum_{k=1}^{M}f(x_{2k-1})$$

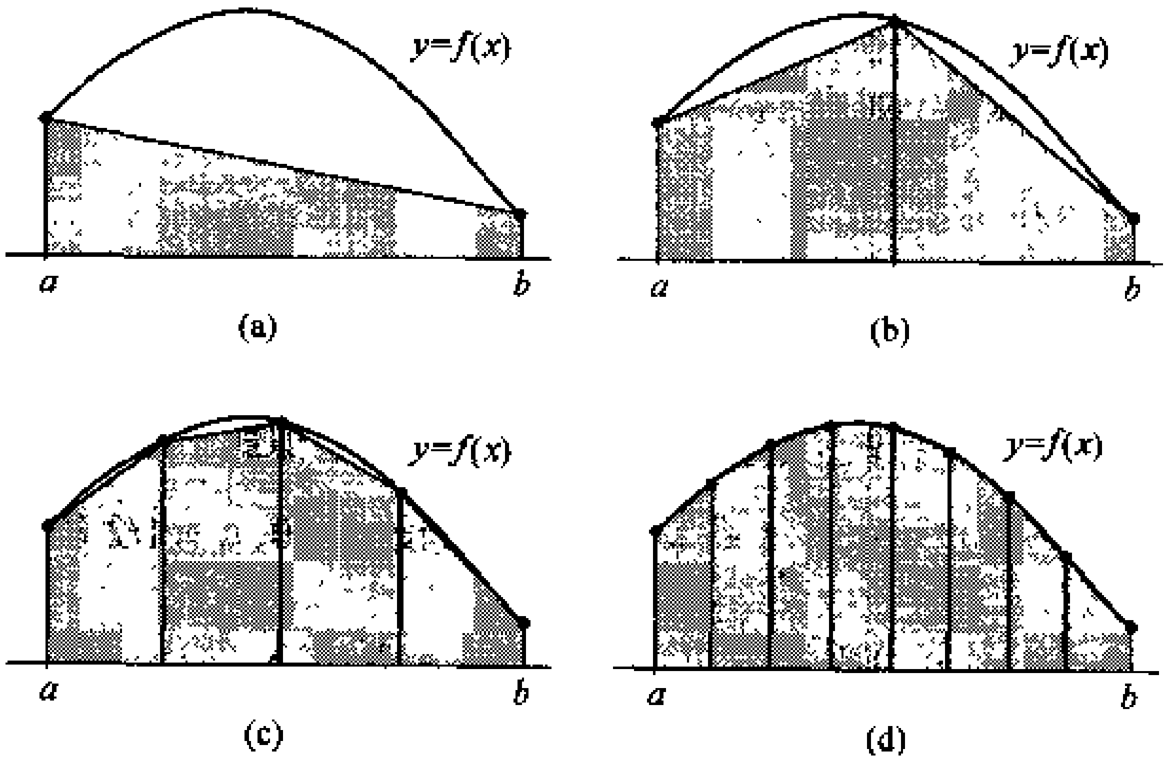


图 7.8　(a) $T(0)$ 为 $2^0=1$ 个梯形的面积
(b) $T(1)$ 为 $2^1=2$ 个梯形的面积
(c) $T(2)$ 为 $2^2=4$ 个梯形的面积
(d) $T(3)$ 为 $2^3=8$ 个梯形的面积

复合公式的递推公式

复合梯形公式的递推公式

$$T_n=\frac{h}{2}[f(a)+2\sum_{k=1}^{n-1}f(x_k)+f(b)]$$

$h=(b-a)/n$

$$T_{2n}=\frac{b-a}{4n}[f(a)+2\sum_{k=1}^{2n-1}f(a+k\frac{b-a}{2n})+f(b)]$$

$$x_k=a+k\frac{b-a}{2n}\quad(k=1,2\cdots,2n-1)$$

$$T_{2n}=\frac{b-a}{4n}\{f(a)+2[\sum_{k=1}^{n-1}f(a+k\frac{b-a}{n})+\sum_{k=1}^{n}f(a+(2k-1)\frac{b-a}{2n}]+f(b)\}$$

$$=\boxed{\frac{b-a}{4n}[f(a)+2\sum_{k=1}^{n-1}f(a+k\frac{b-a}{n})+f(b)]}+\frac{b-a}{2n}\sum_{k=1}^{n}f[a+(2k-1)\frac{b-a}{2n}]$$

已知

$$\boxed{T_{2n}=\frac{1}{2}T_n+\frac{b-a}{2n}\sum_{k=1}^{n}f[a+(2k-1)\frac{b-a}{2n}]}$$

复合公式的递推公式

例 7.11 用连续梯形公式计算积分 $\int_1^5 dx/x = \ln(5) - \ln(1) = 1.609437912$ 的逼近式 $T(0)$，$T(1)$，$T(2)$和 $T(3)$。

表 7.4 给出了计算 $T(3)$所需的 9 个值和计算 $T(1)$，$T(2)$和 $T(3)$所需的中点值。表值的详细过程如下：

$$T_n = \frac{h}{2}\left[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)\right]$$

$$T_{2n} = \frac{1}{2}T_n + \frac{b-a}{2n}\sum_{k=1}^{n} f\left[a + (2k-1)\frac{b-a}{2n}\right]$$

当 $h = 4$：$T(0) = \frac{4}{2}(1.000000 + 0.200000) = 2.400000$

当 $h = 2$：$T(1) = \frac{T(0)}{2} + 2(0.333333)$

$= 1.200000 + 0.666666 = 1.866666$

当 $h = 1$：$T(2) = \frac{T(1)}{2} + 1(0.500000 + 0.250000)$

$= 0.933333 + 0.750000 = 1.683333$

当 $h = \frac{1}{2}$：$T(3) = \frac{T(2)}{2} + \frac{1}{2}(0.666667 + 0.400000$

$+ 0.285714 + 0.222222)$

$= 0.841667 + 0.787302 = 1.628968$

复合公式的递推公式

表 7.4 用来计算 $T(3)$ 的 9 个点和计算 $T(1)$，$T(2)$和 $T(3)$需要的中点值

x	$f(x)=\frac{1}{x}$	计算 $T(0)$需要的端点值	计算 $T(1)$需要的中点值	计算 $T(2)$需要的中点值	计算 $T(3)$需要的中点值
1.0	1.000000	1.000000			
1.5	0.666667				0.666667
2.0	0.500000			0.500000	
2.5	0.400000				0.400000
3.0	0.333333		0.333333		
3.5	0.285714				0.285714
4.0	0.250000			0.250000	
4.5	0.222222				0.222222
5.0	0.200000	0.200000			

龙贝格（Romberg）积分

外推算法 - 用若干个精度较低的积分近似值来推算更精确近似值的方法

根据误差的事后估计

$$I - T_n = -\frac{b-a}{12}h^2 f''(\zeta_1) \qquad I - T_{2n} = -\frac{b-a}{12}\left(\frac{h}{2}\right)^2 f''(\zeta_2)$$

$$\Rightarrow \quad I - T_n \approx 4(I - T_{2n})$$

梯形公式误差分析

$$\Rightarrow \quad I \approx T_{2n} + \frac{1}{3}(T_{2n} - T_n)$$

$$\Rightarrow \quad S_n = \frac{4}{3}T_{2n} - \frac{1}{3}T_n$$

辛普森法则

龙贝格（Romberg）积分

下面的结果说明了梯形公式与辛普生公式之间的重要关系。用步长 $2h$ 和 h 来计算梯形公式的结果分别为 $T(f,2h)$和 $T(f,h)$。用这些值的组合可得辛普生公式：

$$S(f,h)=\frac{4T(f,h)-T(f,2h)}{3} \tag{6}$$

定理 7.5（递归辛普生公式） 设$\{T(J)\}$为由推论 7.4 产生的梯形公式序列，若 $J\geq 1$，且 $S(J)$为区间$[a,b]$的 2^J 个辛普生公式，则 $S(J)$和 $T(J-1)$，$T(J)$满足关系式：

$$S(J)=\frac{4T(J)-T(J-1)}{3},\quad J=1,2,\cdots \tag{7}$$

龙贝格（Romberg）积分

证明：由步长为 h 的梯形公式 $T(J)$ 得到逼近：

$$\int_a^b f(x)\,dx \approx \frac{h}{2}(f_0 + 2f_1 + 2f_2 + \cdots + 2f_{2M-2} + 2f_{2M-1} + f_{2M}) = T(J) \tag{8}$$

由步长为 $2h$ 的梯形公式 $T(J-1)$ 得到逼近：

$$\int_a^b f(x)\,dx \approx h(f_0 + 2f_2 + \cdots + 2f_{2M-2} + f_{2M}) = T(J-1) \tag{9}$$

将(8)式乘以 4，得：

$$4\int_a^b f(x)\,dx \approx h(2f_0 + 4f_1 + 4f_2 + \cdots + 4f_{2M-2} + 4f_{2M-1} + 2f_M) = 4T(J) \tag{10}$$

式(10)减去式(9)得：

$$3\int_a^b f(x)\,dx \approx h(f_0 + 4f_1 + 2f_2 + \cdots + 2f_{2M-2} + 4f_{2M-1} + f_{2M}) = 4T(J) - T(J-1) \tag{11}$$

该式重写为：

$$\int_a^b f(x)\,dx \approx \frac{h}{3}(f_0 + 4f_1 + 2f_2 + \cdots + 2f_{2M-2} + 4f_{2M-1} + f_{2M}) = \frac{4T(J) - T(J-1)}{3}$$

式(12)中的中项为辛普生公式 $S(J) = S(f,h)$，从而定理得证。

龙贝格（Romberg）积分

例 7.11 用连续梯形公式计算积分 $\int_1^5 dx/x = \ln(5) - \ln(1) = 1.609437912$ 的逼近式 $T(0)$，$T(1)$，$T(2)$和 $T(3)$。

例 7.12 用连续辛普生公式求例 7.11 中的积分逼近式 $S(1)$，$S(2)$和 $S(3)$。

利用例 7.11 中的结果和公式(7)及 $J=1,2,3$，计算得：

$$S(1)=\frac{4T(1)-T(0)}{3}=\frac{4(1.866666)-2.400000}{3}=1.688888$$

$$S(2)=\frac{4T(2)-T(1)}{3}=\frac{4(1.683333)-1.866666}{3}=1.622222$$

$$S(3)=\frac{4T(3)-T(2)}{3}=\frac{4(1.628968)-1.683333}{3}=1.610846$$

龙贝格（Romberg）积分

$$\int_a^b f(x)\mathrm{d}x - S_n = -\frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\eta_2)$$

$$I - S_n = -\frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\eta_2) \qquad I - S_{2n} = -\frac{b-a}{180}\left(\frac{h}{4}\right)^4 f^{(4)}(\eta_2{}')$$

$$\Rightarrow \quad I - S_n = 16(I - S_{2n}) \quad \Rightarrow \quad I = \frac{16S_{2n} - Sn}{15}$$

定理 7.6（递归布尔公式） 设$\{S(J)\}$为由定理 7.5 产生的辛普生公式序列，若 $J \geq 2$ 且 $B(J)$ 为区间$[a,b]$内 2^J 个子区间的布尔公式，则 $B(J)$与辛普生公式 $S(J-1)$和 $S(J)$ 满足关系：

$$\boxed{B(J) = \frac{16S(J) - S(J-1)}{15}, \quad J = 2,3,\cdots} \tag{14}$$

龙贝格（Romberg）积分

例 7.11 用连续梯形公式计算积分 $\int_1^5 dx/x = \ln(5) - \ln(1) = 1.609437912$ 的逼近式 $T(0)$，$T(1)$，$T(2)$和 $T(3)$。

例 7.13 用连续布尔公式求例 7.11 中积分的逼近 $B(2)$和 $B(3)$。

根据例 7.12 中的结果、式(14)及 $J=2$ 和 3，计算得：

$$B(2) = \frac{16S(2) - S(1)}{15} = \frac{16(1.622222) - 1.688888}{15} = 1.617778$$

$$B(3) = \frac{16S(3) - S(2)}{15} = \frac{16(1.610846) - 1.622222}{15} = 1.610088$$

龙贝格（Romberg）积分

$$\int_a^b f(x)\mathrm{d}x - C_n = -\frac{2(b-a)}{945}\left(\frac{h}{4}\right)^6 f^{(6)}(\eta_4)$$

$$I - C_n = -\frac{2(b-a)}{945}\left(\frac{h}{4}\right)^6 f^{(6)}(\eta_4) \qquad I - C_{2n} = -\frac{b-a}{180}\left(\frac{h}{8}\right)^6 f^{(6)}(\eta_4)$$

$$\Rightarrow \quad I - C_n = 64(I - C_{2n}) \quad \Rightarrow \quad I = \frac{64C_{2n} - Cn}{63}$$

例 7.11 用连续梯形公式计算积分 $\int_1^5 dx/x = \ln(5) - \ln(1) = 1.609437912$ 的逼近式 $T(0)$，$T(1)$，$T(2)$和 $T(3)$。

对例 7.11 积分的下一级逼近为：

$$\frac{64B(3) - B(2)}{63} = \frac{64(1.610088) - 1.617778}{63} = 1.609490$$

龙贝格（Romberg）积分

引理 7.1（龙贝格积分的理查逊改进） 给定两个 Q 的逼近 $R(2h,K-1)$和 $R(h,K-1)$，满足：

$$Q = R(h,K-1) + c_1 h^{2K} + c_2 h^{2K+2} + \cdots \tag{28}$$

和：

$$Q = R(2h,K-1) + c_1 4^K h^{2K} + c_2 4^{K+1} h^{2k+2} + \cdots \tag{29}$$

有改进的逼近，形如：

$$Q = \frac{4^K R(h,K-1) - R(2h,K-1)}{4^K - 1} + O(h^{2K+2}) \tag{30}$$

龙贝格（Romberg）积分

定义 7.4 定义$[a,b]$内$f(x)$的面积公式序列$\{R(J,K):J\geq K\}_{J=0}^{\infty}$如下：

$$\begin{aligned} R(J,0) &= T(J) \quad , \quad J\geq 0(\text{是连续梯形公式}) \\ R(J,1) &= S(J) \quad , \quad J\geq 1(\text{是连续辛普生公式}) \\ R(J,2) &= B(J) \quad , \quad J\geq 2(\text{是连续布尔公式}) \end{aligned} \tag{31}$$

第一个公式$\{R(J,0)\}$用来产生第一次改进$\{R(J,1)\}$，后者又用来产生第二次改进$\{R(J,2)\}$。我们已知道形式：

$$\begin{aligned} R(J,1) &= \frac{4^1 R(J,0)-R(J-1,0)}{4^1-1} \quad , \quad J\geq 1 \\ R(J,2) &= \frac{4^2 R(J,1)-R(J-1,1)}{4^2-1} \quad , \quad J\geq 2 \end{aligned} \tag{32}$$

在式(24)和式(27)中用式(31)中的符号来表示。构造改进的一般公式为：

$$R(J,K) = \frac{4^K R(J,K-1)-R(J-1,K-1)}{4^K-1} \quad , \quad J\geq K \tag{33}$$

龙贝格（Romberg）积分

表 7.5　龙贝格积分表

J	$R(J,0)$ 梯形公式	$R(J,1)$ 辛普生公式	$R(J,2)$ 布尔公式	$R(J,3)$ 第三次改进	$R(J,4)$ 第四次改进
0	$R(0,0)$				
1	$R(1,0)$	$R(1,1)$			
2	$R(2,0)$	$R(2,1)$	$R(2,2)$		
3	$R(3,0)$	$R(3,1)$	$R(3,2)$	$R(3,3)$	
4	$R(4,0)$	$R(4,1)$	$R(4,2)$	$R(4,3)$	$R(4,4)$

龙贝格（Romberg）积分

例 7.14 利用龙贝格积分计算定积分的近似值：

$$\int_0^{\pi/2}(x^2+x+1)\cos(x)\,dx=-2+\frac{\pi}{2}+\frac{\pi^2}{4}=2.038197427067\cdots$$

表 7.6 给出计算过程，每一列中的数都收敛到 2.038197427067…，辛普生公式的列比梯形公式的列收敛速度快。在本例中，相邻的两列中右边的列的速度快于左边的列。

表 7.6 例 7.14 的龙贝格积分表

J	$R(J,0)$ 梯形公式	$R(J,1)$ 辛普生公式	$R(J,2)$ 布尔公式	$R(J,3)$ 第三次改进
0	0.785398163397			
1	1.726812656758	2.040617487878		
2	1.960534166564	2.038441336499	2.038296259740	
3	2.018793948078	2.038213875249	2.038198711166	2.038197162776
4	2.033347341805	2.038198473047	2.038197446234	2.038197426156
5	2.036984954990	2.038197492719	2.038197427363	2.038197427064

$$S_n=\frac{4}{3}T_{2n}-\frac{1}{3}T_n \qquad I=\frac{16S_{2n}-Sn}{15} \qquad I=\frac{64C_{2n}-Cn}{63}$$

龙贝格（Romberg）积分

定理 7.7（龙贝格积分的精度） 设 $f\in C^{2k+2}[a,b]$，则龙贝格逼近的截断误差由公式：

$$\int_a^b f(x)\,dx = R(J,K) + b_K h^{2K+2} f^{(2K+2)}(C_{J,K})$$

$$= R(J,K) + O(h^{2K+2}) \qquad (34)$$

给出。其中，$h=(b-a)/2^J$，b_K 为依赖于 K 的常数，且 $C_{J,K}\in[a,b]$。

例 7.15 应用定理 7.7，并证明：

$$\int_0^2 10x^9\,dx = 1024 \equiv R(4,4)$$

证：

被积函数为 $f(x)=10x^9$，且 $f^{(10)}(x)=0$。故值 $K=4$ 可使误差项恒为 0，通过数值计算可得 $R(4,4)=1\,024$。

龙贝格（Romberg）积分

方法	需要的节点数	复合公式需要的节点数	精度	应用范围	编程难度
梯形公式	2	$n+1$	$h^3 f''(\xi)$	广泛	容易
Simpson1/3法则	3	$2n+1$	$h^5 f^{(4)}(\xi)$	广泛	容易
Simpson3/8法则	4			广泛	容易
高阶Newton-Cotes公式	≥5			很少	容易
Romberg积分	3			要求已知f(x)表达式	容易

数值积分在MATLAB中的应用

函数	描述
quad	Simpson公式
trapz	梯形公式
diff	数值微分（连续函数求导）
quadl	Lobatto求积（一种高斯求积公式，取代了quad8）
sum	求和
dblquad	二重积分

数值积分在MATLAB中的应用

辛普森积分公式的应用：

计算函数 x^2 在(0, 2)内的积分

$$\int_0^2 x^2 dx = \frac{1}{3}x^3\Big|_0^2 = \frac{1}{3}(2^3 - 0) = \quad 2.6667$$

Matlab 演示

将函数写成：
```
square1=@(x)x.*x;
y=quad(square1,0,2)

 y =
     2.6667
```

也可以
```
f=@(x)x.^2
y=quad(f,0,2)
```

```
aaa=@(x)1./(3*x-1);
```
对于[1,1.5]区间的积分，可以得到
```
y=quad(aaa,1,1.5)

y =

    0.1865
```

对于[0,1]区间的积分，中间出现了无穷大(x=1/3)的时候
```
y=quad(aaa,0,1)
Warning: Infinite or Not-a-Number function value encountered.
> In quad at 109

y =

   Inf
```

示教M文件 quadtx 和 quadgui

#内置函数

可以不采用调用函数，而采用内置函数的形式。

1）inline内置函数指令

```
f=inline('1./sqrt(1+x.^6)')
f =
     Inline function:
     f(x) = 1/sqrt(1+x^6)

quadtx(f,0,1)
ans =
   0.9270
```

2）匿名函数指令

```
f=@(x) 1/sqrt(1+x^4)
f =
   @(x)1/sqrt(1+x^4)

Q=quadtx(f,0,1)
Q =
   0.9270
```

```
quad(f,0,1)
Error using ==> inlineeval at 15
Error in inline expression ==> 1/sqrt(1+x^6)
Matrix must be square.
```

对 quadtx 函数的解释：

help quadtx

QUADTX Evaluate definite integral numerically.

Q = QUADTX(F,A,B) approximates the integral of F(x) from A to B to within a tolerance of 1.e-6.

Q = QUADTX(F,A,B,tol) uses the given tolerance instead of 1.e6.

The first argument, F, is a function handle or an anonymous functionthat defines F(x).

Arguments beyond the first four, Q = QUADTX(F,a,b,tol,p1,p2,...), are passed on to the integrand, F(x,p1,p2,..).

QUAD Numerically evaluate integral, adaptive Simpson quadrature.

Q = QUAD(FUN,A,B) tries to approximate the integral of scalar-valued function FUN from A to B to within an error of 1.e -6 using recursive adaptive Simpson quadrature. FUN is a function handle. The function Y=FUN(X) should accept a vector argument X and return a vector result Y, the integrand evaluated at each element of X.

Q = QUAD(FUN,A,B,TOL) uses an absolute error tolerance of TOLinstead of the default, which is 1.e-6. Larger values of TOL result in fewer function evaluations and faster computation, but less accurate results. The QUAD function in MATLAB 5.3 used a less reliable algorithm and a default tolerance of 1.e-3.

函数quad, quadtx的区别:

Quad: Numerically evaluate integral, adaptive Simpson quadrature,自适应的Simpson积分

Quadtx：
Q = QUADTX(F,A,B) approximates the integral of F(x) from A to B to within a tolerance of 1.e-6.

是简单的计算从A到B间的定积分

示教M文件 quadtx 和 quadgui

#符号积分
直接推导出积分公式，并算出积分，
要计算函数 x^2 在（0，2）内的积分，
2.6667
可以直接输入积分公式int(‘fun’,积分下限，积分上限)

result=int('x^2',0,2)
result =
8/3
也可以得到积分公式：
syms a b
result=int('x^2',a,b)
result =
b^3/3 - a^3/3

但是，请注意，直接用积分公式的话，有的积分是很难有结果的。尤其当需要进行定积分的上下限要讨论的时候很复杂。
result=int('1/3*x',a,b)
b^2/6 - a^2/6
而下面
result=int('1/x')
ans =
log(x)
>>result=int('1/3*x-1',0,1)
result =
-5/6
另外，不定积分
>> int('1/3*x-1')
ans =
(x - 3)^2/6

示教M文件 quadtx 和 quadgui

#quadgui.m

```
function [Qout,fcount] = quadgui(F,a,b,tol,varargin)
%QUADGUI  Demonstrate numerical evaluation  of a definite  integral.
%   Q = QUADGUI(F,A,B) shows the steps  in approximating  the integral
%   of F(x) from A to B by adaptive  extrapolated  Simpson's quadrature.
%
%   The shaded area shows the integral  over the current subinterval.
%   The color switches  to green when the desired accuracy  is obtained.
%
%   Q = QUADGUI(F,A,B,tol) uses the given  tolerance  instead of 1.e-4.
%
%   The first  argument,  F, is  a function  handle or an anonymous  function
%   that defines  F(x).
%   Arguments  beyond the first  four, Q = QUADGUI(F,a,b,tol,p1,p2,...),
%   are passed on to the integrand,  F(x,p1,p2,..).
%
%   [Q,fcount] = QUADGUI(F,...) also  counts the number of evaluations
%   of F(x).
%
```

```
%   Examples:
%         F                      a     b       tol(optional)
%      humps(x)                  0     1
%      humps(x)                  0     1       1.e-6
%      humps(x)                 -1     2
%      sin(x)                    0     pi      1.e-8
%      cos(x)                    0     9*pi/2  1.e-6
%      sqrt(x)                   0     1       1.e-8
%      -sqrt(x)*log(x)           eps   1       1.e-8
%      1/(3*x-1)                 0     1
%      t^(8/3)*(1-t)^(10/3)      0     1       1.e-8
%      tan(sin(x))-sin(tan(x))   0     pi
%
%      quadgui(@(x)F,a,b)
%
%   See also  QUADTX, QUAD, QUADL, DBLQUAD.

%   Copyright  2014 Cleve  Moler
%   Copyright  2014 The MathWorks,  Inc.
```

示教M文件 quadtx 和 quadgui

```
shg                                        #quadgui.m
clf reset
set(gcf,'menubar','none','numbertitle','off','name','Quad  gui')

% Default tolerance
if nargin < 4 | isempty(tol)
   tol = 0.5e-6;
end

% Default function and interval.
if nargin < 3
   F = @humps;
   a = 0;
   b = 1;
end

% Initialization
c = (a + b)/2;
fa = F(a,varargin{:});
fc = F(c,varargin{:});
fb = F(b,varargin{:});
```

```
% Scale the plot
h = b - a;
x = [a c b];
y = [fa fc fb];
maxy = max(y);
miny = min(y);
for k = 1:63
   v = F(a+k*h/64,varargin{:});
   maxy = real(max(maxy,v));
   miny = real(min(miny,v));
end
set(gcf,'userdata',0)
hold on
p(1) = fill(a,fa,'k');
p(2) = fill(b,fb,'k');
p(3) = plot(x,y,'.','markersize',16);
hold off
```

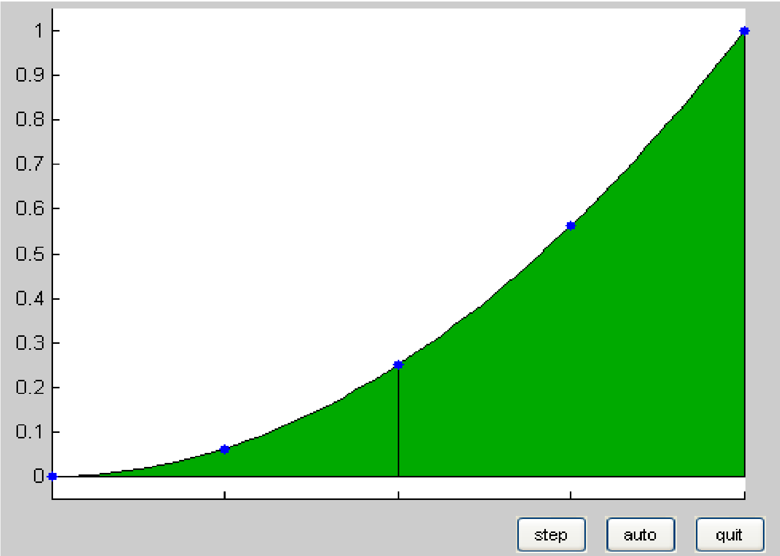
#quadgui.m

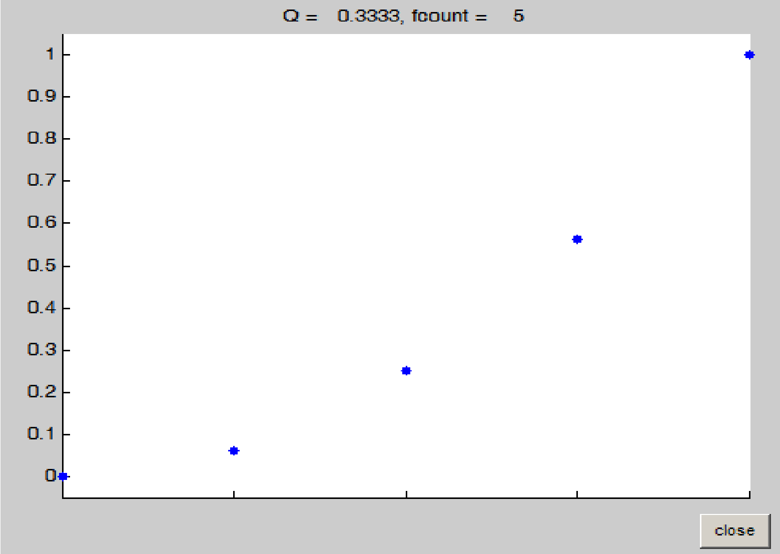
```
s = (maxy - miny)/20;
axis([a b miny-s maxy+s])
q(1) = uicontrol('string','step', ...
   'units','normal','pos',[.65 .02 .08 .06], ...
   'callback','set(gcf,"userdata",1)');
q(2) = uicontrol('string','auto', ...
   'units','normal','pos',[.75 .02 .08 .06], ...
   'callback','set(gcf,"userdata",2)');
q(3) = uicontrol('string','quit', ...
   'units','normal','pos',[.85 .02 .08 .06], ...
   'callback','set(gcf,"userdata",3)');

% Recursive call
[Q,k] = quadguistep(F, a, b, tol, fa, fc, fb, varargin{:});
fcount = k + 3;

% Finish
title(sprintf('Q = %8.4f, fcount = %4.0f',Q,fcount))
delete(p(1:2));
delete(q(1:2));
set(q(3),'string','close','callback','close(gcf)')
if nargout > 0, Qout = Q; end
```

```
% ------------------------------------------------------------
function [Q,fcount] = quadguistep(F,a,b,tol,fa,fc,fb,varargin)
% Recursive subfunction used by quadtx.
h = b - a;
c = (a + b)/2;
d = (a + c)/2;
e = (c + b)/2;
fd = F(d,varargin{:});
fe = F(e,varargin{:});
Q1 = h/6 * (fa + 4*fc + fb);
Q2 = h/12 * (fa + 4*fd + 2*fc + 4*fe + fb);
u1 = a:h/64:c;
v1 = polyinterp([a d c],[fa fd fc],u1);
u1 = [a u1 c];
v1 = [0 v1 0];
u2 = c:h/64:b;
v2 = polyinterp([c e b],[fc fe fb],u2);
u2 = [c u2 b];
v2 = [0 v2 0];
if (abs(Q2 - Q1) <= tol)
   color = [0 2/3 0];
else
   color = [.6 .6 .6];
end
```

#quadgui:
quadgui(@(x)x^2,0,1)

过程就是将其中取的区间显示出来



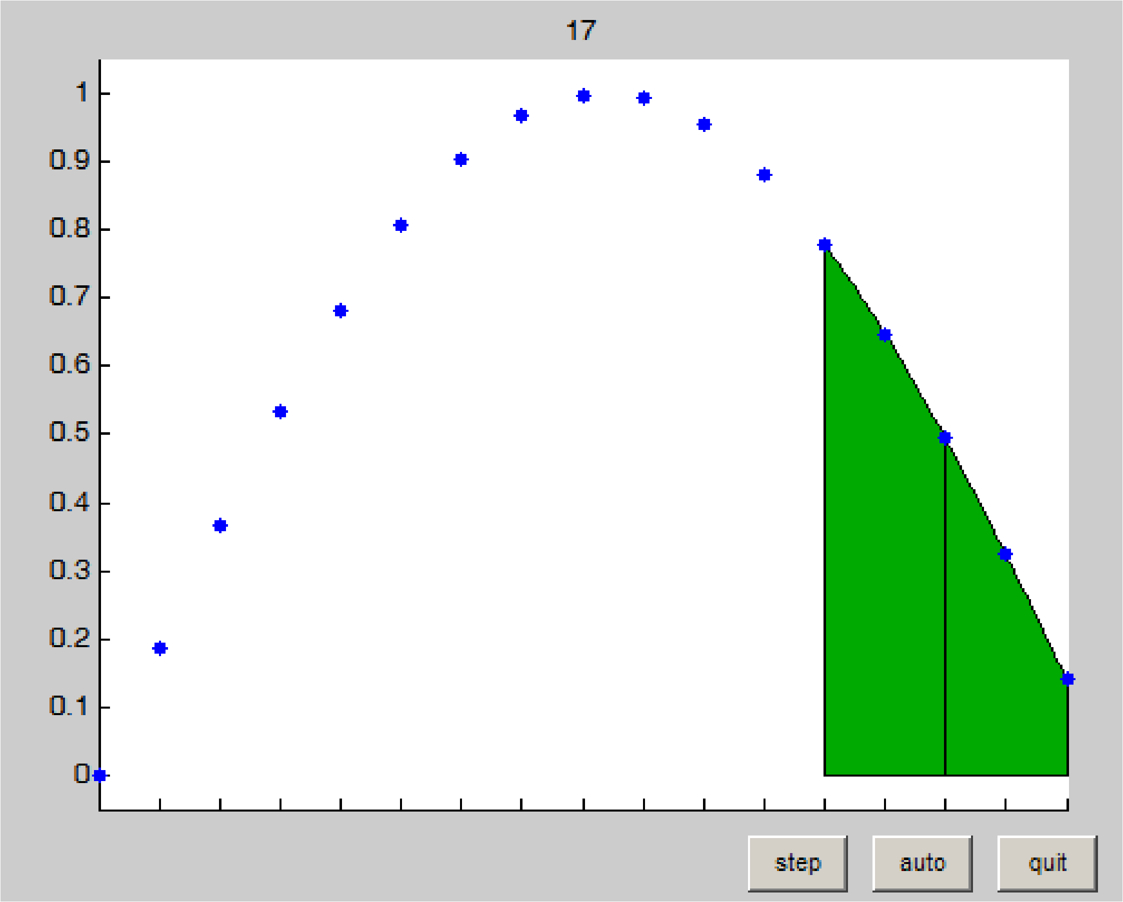



```
p = flipud(get(gca,'child'));
x = [get(p(3),'xdata') d e];
y = [get(p(3),'ydata') fd fe];
set(p(1),'xdata',u1,'ydata',v1,'facecolor',color)
set(p(2),'xdata',u2,'ydata',v2,'facecolor',color)
set(p(3),'xdata',x,'ydata',y)
set(gca,'xtick',sort(x),'xticklabel',[]);
title(num2str(length(x)))
pause(.25)
while get(gcf,'userdata') == 0
  pause(.25)
end
if get(gcf,'userdata') == 1
  set(gcf,'userdata',0)
end

if (abs(Q2 - Q1) <= tol) | (get(gcf,'userdata') == 3)
  Q  = Q2 + (Q2 - Q1)/15;
  fcount = 2;
else
  [Qa,ka] = quadguistep(F, a, c, tol, fa, fd, fc, varargin{:});
  [Qb,kb] = quadguistep(F, c, b, tol, fc, fe, fb, varargin{:});
  Q  = Qa + Qb;
  fcount = ka + kb + 2;
end
```

示教M文件 quadtx 和 quadgui

quadgui(@(x)sin(x),0,3)

过程就是将其中取的区间显示出来



被积函数的表述

MATLAB 有许多种不同方式表述求积程序所需的被积函数，对于简单的、表述长度不超过一行的公式，使用匿名函数最方便，例如：

$$\int_0^1 \frac{1}{\sqrt{1+x^4}}dx$$

可以用下面的语句就是：

```
f=@(x) 1/sqrt(1+x^4)
Q=quadtx(f,0,1)
```

如果我们想要计算：

$$\int_0^\pi \frac{\sin x}{x}dx$$

可以用下面的语句进行尝试：

```
f=@(x) sin(x)/x
Q=quadtx(f,0,pi)
```

习惯上，如果一个已给的连续函数的原函数能用初等函数表达出来，就说这函数是“积得出的函数”，否则就说它是“积不出”的函数。比如下面列出的几个积分都是属于“积不出”的函数，但是这些积分在概率论，数论，光学，傅里叶分析等领域起着重要作用。

（1）∫e^(-x²)dx；（2）∫(sinx)/xdx；
（3）∫1/(lnx)dx；（3）∫sinx²dx；

这个时候，由于有 0 作为除数，不能计算，改成 Matlab 能取的最小的浮点数 realmin,

```
>> Q=quadtx(f,realmin,pi)
Q =
     1.8519
```

但是当我们在MatLab中输入下面的值的话，得到的是:

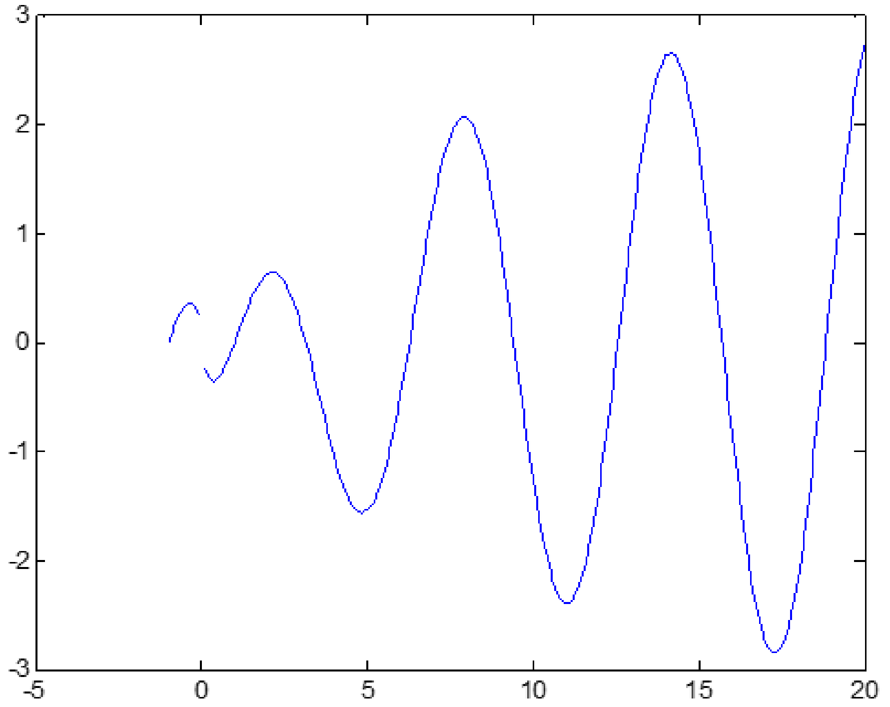
```
>>  result=int('sin(x)/x')
 result =
 sinint(x)
```

sinint是MATLAB自己定义的函数，其实就是sinint(x) = int(sin(t)/t,t,0,x)，断点问题其实没有体现出来。

对于简单的断点问题，我们可以看一下简单的例子，比如我们按照[0,pi]这样的区间定积分，这样同样就存在区间上不可取值的断点，比如对于函数sin(x)*log(x):

做出曲线为:

```
>> x=-1:0.1:20;
>> y=sin(x).*log(x);
>> plot(x,y)
```



这些不能求积分的函数其实就是在某些点存在严重的断点。

#几点说明:

1）首先作为一个函数的话，必须写成sin(x)而不能写成sinx，前者才是一个函数

2) 在多项式的表达中，当x本身已经定义了是一个数列或者向量的情况下，x后面要加点的，这就表示后面的计算是数列中的数一个一个计算的，这是数组的乘法，每一次是对应的值相乘，是一维的。如果不加点就变成了对x向量计算了，是高维的，其实是.^是运算符号表示数组的乘方，./表示数组除法等

求y=x^5-3.5*x^4+2.75*x^3+2.125*x^2-3.875*x+1.25的曲线，用x,y画出

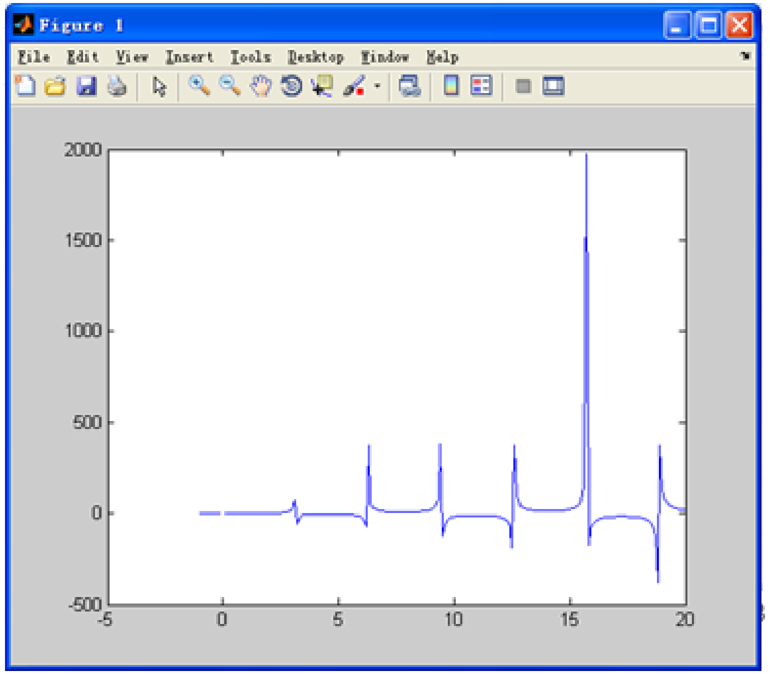
```
>> x=-1:0.05:3;
>> y=x.^5-3.5*x.^4+2.75*x.^3+2.125*x.^2-3.875*x+1.25;
>> plot(x,y)
```

而不能写成:

```
y=x^5-3.5*x^4+2.75*x^3+2.125*x^2-3.875*x+1.25;
```

3) 定义了

```
x=-1:0.1:20;
y=x./sin(x);
plot(x,y)
```



(可见画出的图在sin(x)=0的地方都是缺的，是无穷大)

被积函数的表述

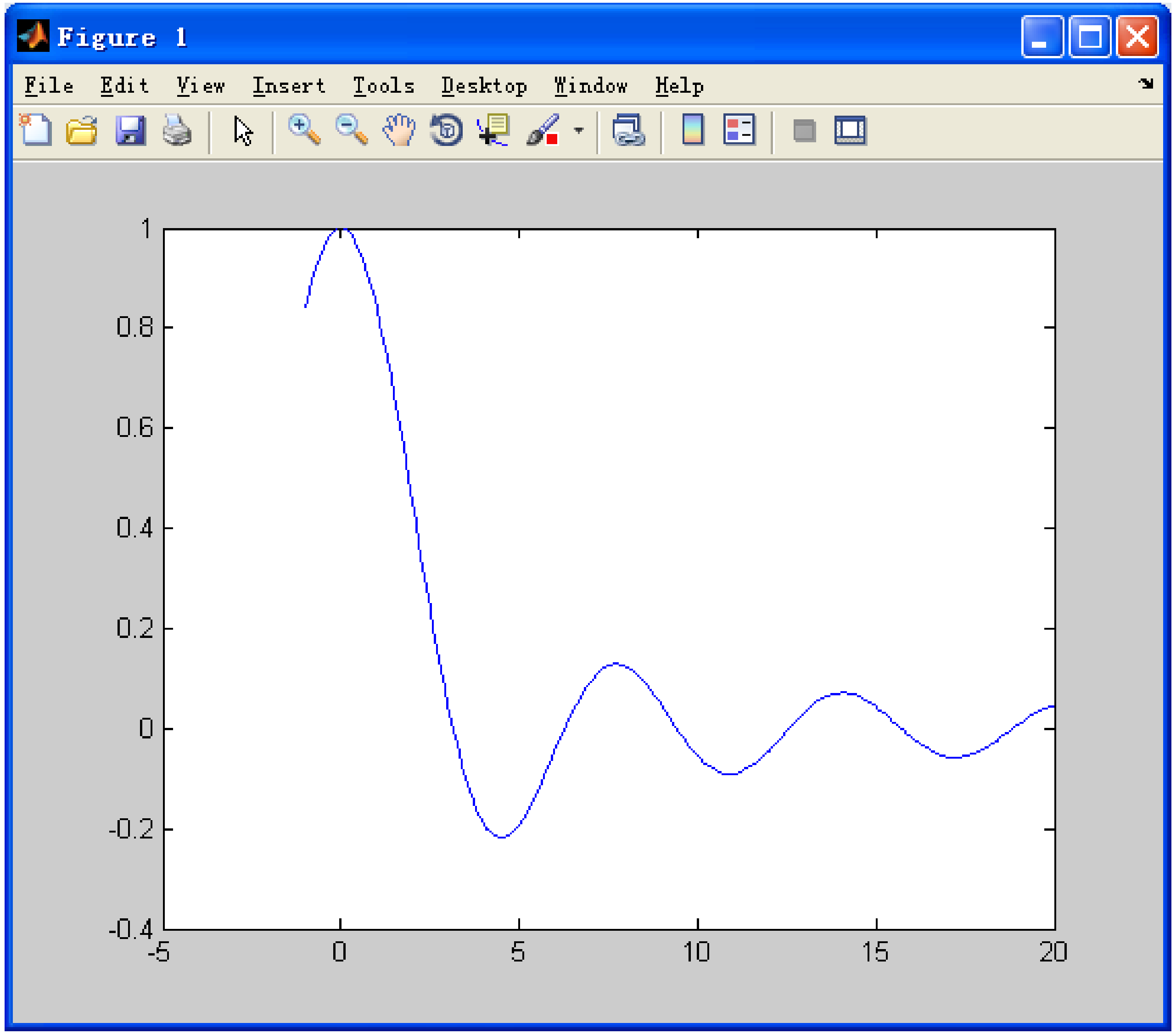
```
x=-1:0.1:20;
y= sin(x)./ x;
plot(x,y)
```

(可见画出的图在sin(x)=0的地方都是1，这是一个极值，符合情况。)

可以估算一下在(0,pi)间，面积为1*3.14*(1/2)=1.57

也可以直接使用内嵌函数来计算：

```
>> f=inline('sin(x)/x');
>> Q=quadtx(f, realmin,pi)
```



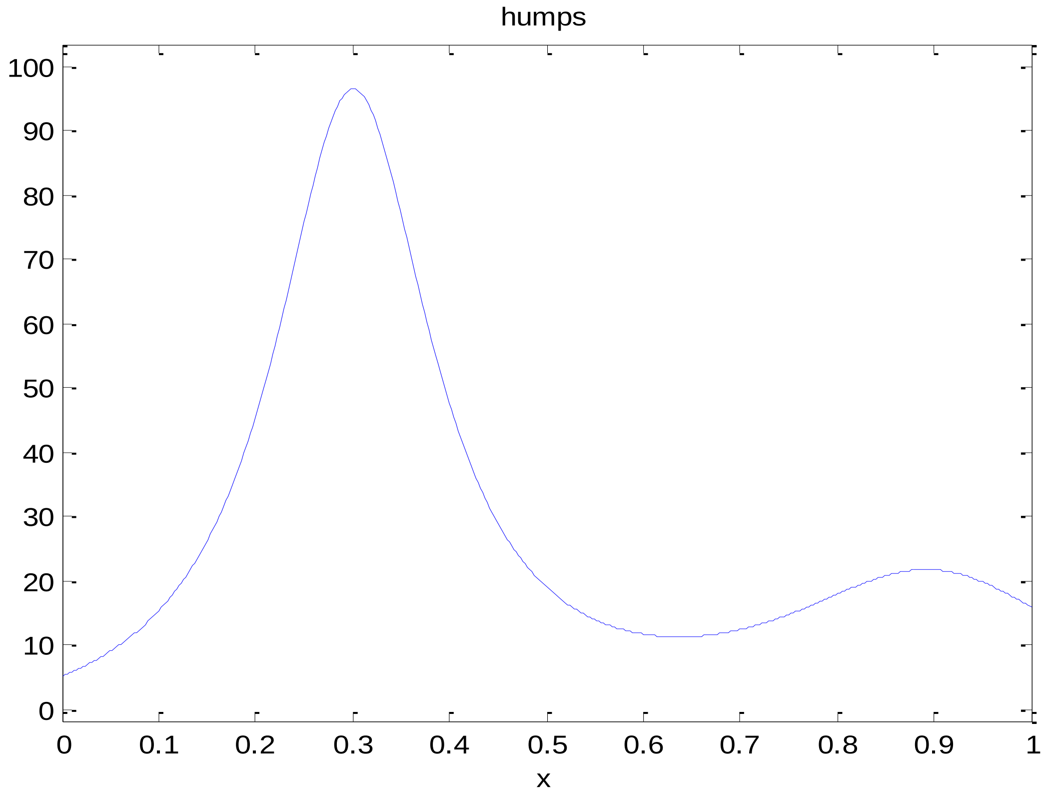
性能

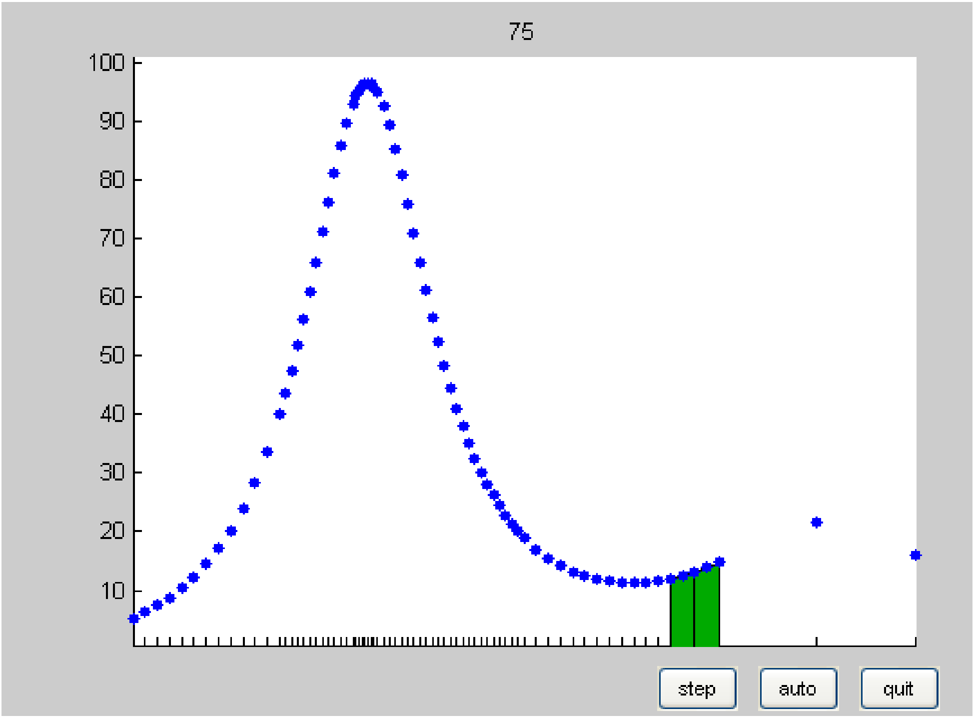
Matlab下的Demos目录中有一个名为humps的函数，它用于演示MATLAB中绘图、数值积分和方程求根的有关命令，这个函数为：

ezplot(@humps,0,1)

quadgui(@humps,0,1,1.e-4)

可以看到：按照所给的容差，自适应算法对被积函数计算了93次，在驼峰附近的点比较密集。





性能

借助符号工具包，可以对h(x)进行解析积分，运行下列语句：

```
syms x
h=1/((x-0.3)^2+0.01)+1/((x-0.9)^2+0.04)-6
I =int(h)
```

得到不定积分：

10*atan(10*x - 3) - 6*x + 5*atan(5*x - 9/2)

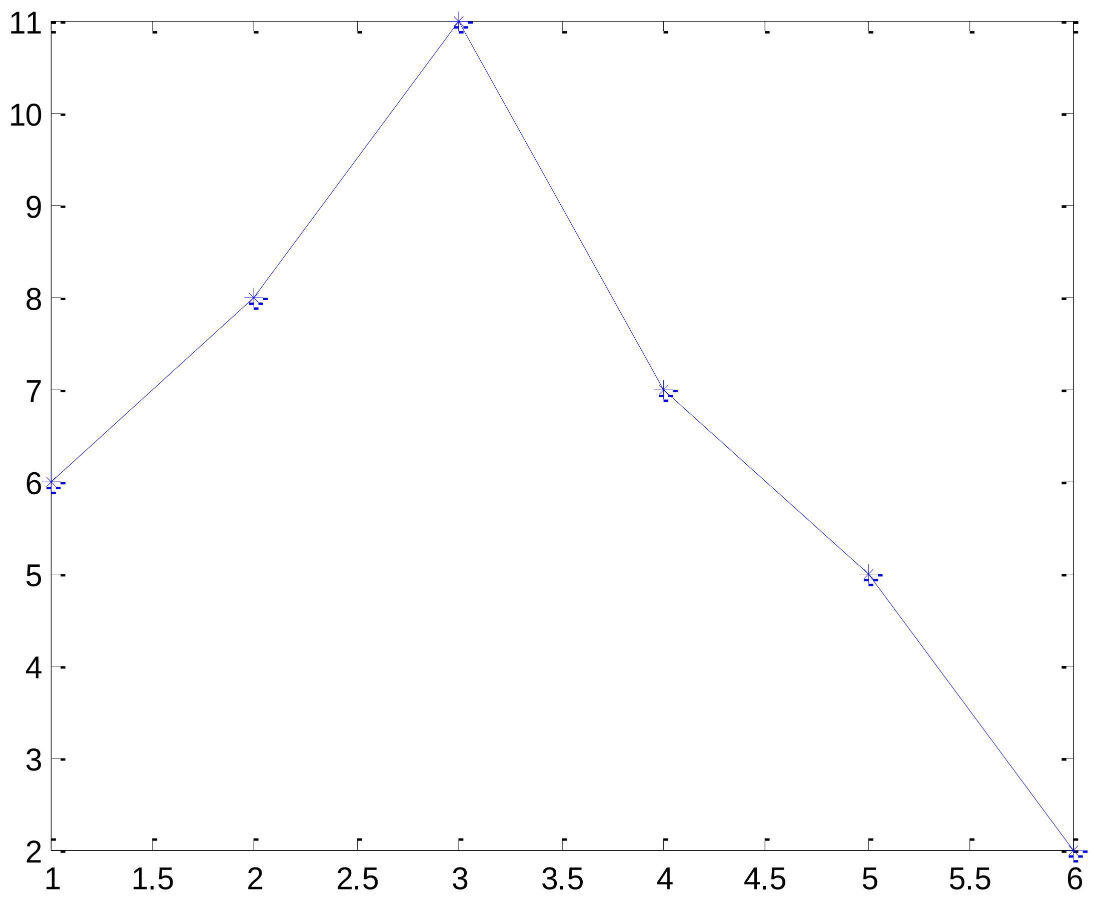
积分离散数据

离散型的积分，自适应方法无法使用，最显然的方式就是分段线性函数进行积分，这就是复合梯形积分法则(composite trapezoid rule).

离散数据的积分-复合梯形法则:

```
x=1:6
y=[6 8 11 7 5 2]
T=sum(diff(x).*(y(1:end-1)+y(2:end))/2)
35

plot(x,y)
```

一般而言，梯形积分已经足够满意，不需要考虑更复杂的方式，高阶积分也能给出其他近似值，如果没有原始数据的更深入的信息，就无法断言它们是否更精确。



MATLAB求积分

- quad 采用递推自适应Simpson法计算积分;精度较高，较常用
- quadl采用递推自适应Lobatto法计算积分;精度高，最常用
- trapz 采用梯形法计算积分;速度快，精度差
- cumtrapz 采用梯形法计算一个区间上的积分曲线;速度快，精度差
- Fnint 利用样条函数求不定积分；与spline、ppval配合使用；主要对付“表格”函数的积分

【例 题】 $$I = \int_0^1 e^{-x^2} dx$$

（1）符号解析法

```
syms x;IS=int('exp(-x*x)','x',0,1)
vpa(IS)
IS =
1/2*erf(1)*pi^(1/2)
ans =
.74682413281242702539946743613185
```

（2）MATLAB指令quad和quadl求积

```
fun=inline('exp(-x.*x)','x');
Isim=quad(fun,0,1),IL=quadl(fun,0,1)
Isim =0.7468
IL =0.7468
```

（3）10参数Gauss法

```
Ig=gauss10(fun,0,1)
Ig = 0.7463
```

（4）样条函数积分法

```
xx=0:0.1:1.5;ff=exp(-xx.^2);
pp=spline(xx,ff);
int_pp=fnint(pp);
Ssp=ppval(int_pp,[0,1])*[-1;1]
Ssp =
   0.7468
```

$$y = \sin x \qquad S(x) = \int_0^x \sin x dx = 1 - \cos x \qquad y' = \cos x$$

(1)不定积分样条函数、导数样条函数的求取和精度分析

```
x=(0:0.1:1)*2*pi;y=sin(x);
pp=spline(x,y);
int_pp=fnint(pp);
der_pp=fnder(pp);
%
xx=(0:0.01:1)*2*pi;
err_yy=max(abs(ppval(pp,xx)-sin(xx)))
err_int=max(abs(ppval(int_pp,xx)-(1-cos(xx))))
err_der=max(abs(ppval(der_pp,xx)-cos(xx)))
err_yy =0.0026
err_int =0.0010
err_der =0.0253
```

(2)不定积分样条函数、导数样条函数的使用

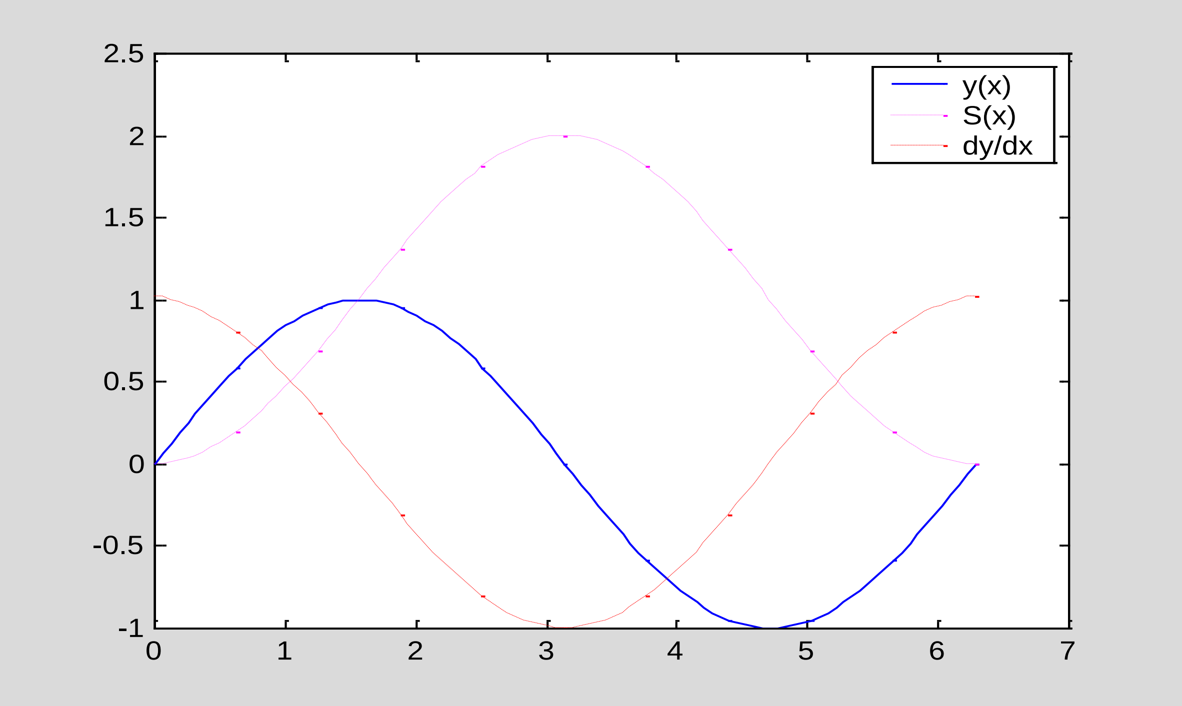
不定积分样条函数可用来计算基础区间中任何区间上的定积分。导数样条函数可以计算基础区间内任何一点的导数。

```
% 计算y(x)在区间[1，2]上的定积分
DefiniteIntegral.bySpline=ppval(int_pp,[1,2])*[-1;1];
DefiniteIntegral.byTheory=(1-cos(2))-(1-cos(1));
% 计算dy(3)/dx
Derivative.bySpline=fnval(der_pp,3);
Derivative.byTheory=cos(3);
Derivative.byDiference=(sin(3.01)-sin(3))/0.01;
DefiniteIntegral,Derivative
```

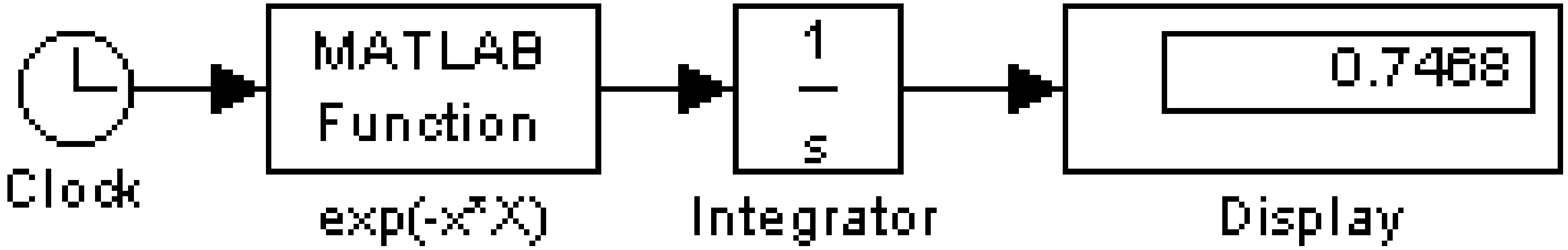
```
DefiniteIntegral =
    bySpline: 0.9563
    byTheory: 0.9564
Derivative =
      bySpline: -0.9895
      byTheory: -0.9900
    byDiference: -0.9907
```

(3)绘制3个样条函数的图形

```
fnplt(pp,'b-');hold on
fnplt(int_pp,'m:'),fnplt(der_pp,'r--');hold off
legend('y(x)','S(x)','dy/dx')
```



（4）SIMULINK 积分法



第六章 课后作业

Q1：分别用复合梯形公式和复合辛普森公式计算

$$\int_0^1 \frac{x}{4+x^2}dx, n = 8$$

Q2：若用复合梯形公式计算$\int_0^1 e^x\,dx$，问区间[0 ,1]应该分为多少等份才能使阶段误差不超$\frac{1}{2}\times 10^{-5}$？改用复合辛普森公式呢？

感谢聆听,欢迎讨论！