

第五章作业

1 问题叙述

火箭发射过程的速度可由如下公式计算：

$$v = u \ln \left(\frac{m_0}{m_0 - qt} \right) - gt \quad (1)$$

其中， v 是向上的速度， u 是燃料相对于火箭喷出的速度， m_0 是火箭在 $t = 0$ 时的初始质量， q 是燃料消耗速度， g 是重力加速度。假设 $u = 1800\text{m/s}$ ， $m_0 = 160000\text{kg}$ ， $q = 2500\text{kg/s}$ ， $g = 9.8\text{m/s}^2$ ，请：

- 采用不同的数值积分方法计算火箭在 30s 时能上升多高，并分析误差。
- 利用数值微分方法画出火箭加速度与时间的关系图。

2 问题分析

本问题可以用解析法算出精确值：

$$\int_0^{30} u \ln \left(\frac{m_0}{m_0 - qt} \right) - gt \, dt = -61200 \ln \left(\frac{32}{17} \right) + 49590 = 10879.6194048972 \quad (2)$$

$$\frac{dv}{dt} = \frac{1800}{64 - t} - 9.8 \quad (3)$$

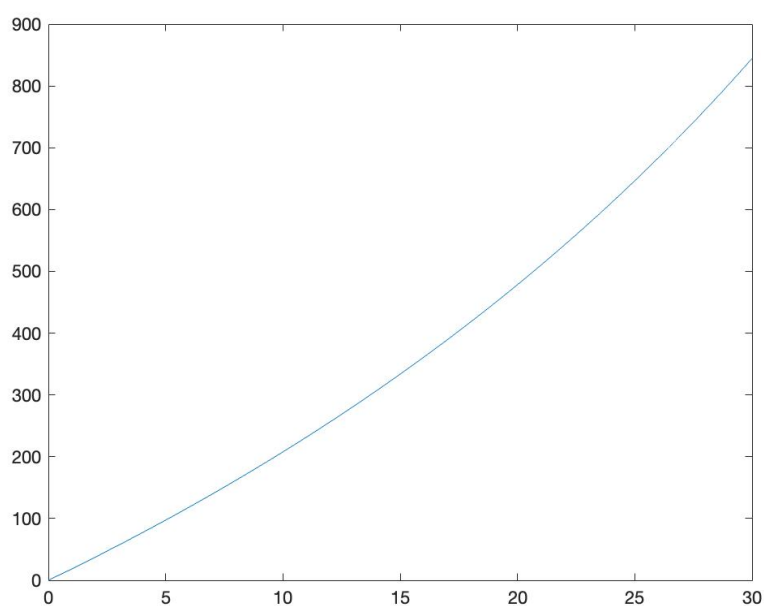


图 1: 原函数

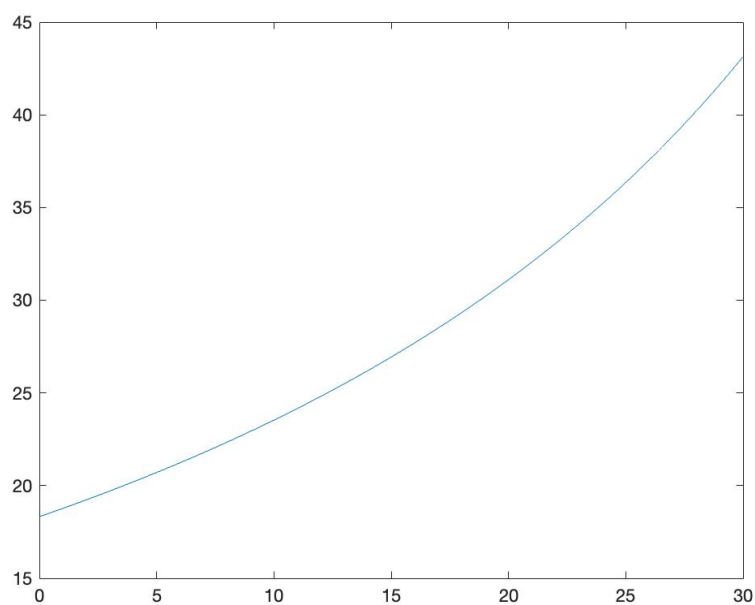


图 2: 导函数

对于第一问,使用 Newton-Cotes 积分中的梯形法、Simpson1/3 法、Simpson3/8 法、复合梯形法、复合 Simpson 法、非等距积分, Romberg 法, Guass 法分别来估计计算积分值, 并分析误差。

对于第二问, 使用差商近似法、插值型数值微分法来估计计算微分值。

目录

1 问题叙述	1
2 问题分析	1
3 数值积分	4
3.1 Newton-Cotes 积分	5
3.1.1 梯形公式	6
3.1.2 Simpson 公式 (Simpson1/3 法则)	7
3.1.3 Simpson3/8 法则	8
3.1.4 复合梯形公式	9
3.1.5 复合 Simpson 公式	11
3.1.6 非等距积分 (步长自适应)	13
3.2 Romberg 积分	14
3.2.1 普通 Romberg 积分	15
3.2.2 改进的 Romberg 积分	16
3.3 Guass 求积公式	17
3.3.1 单段 Guass 求积	18
3.3.2 复合 Guass 求积	19
3.4 数值积分总结	20
4 数值微分	20
4.1 差商近似	20
4.1.1 有限差商近似	20
4.1.2 高精度微分公式	23
4.2 插值型数值微分	25
5 小结	28

3 数值积分

首先将 v 转换为函数文件并保存在 $v.m$ 中：

```
1 function out = v(t)
2 u=1800;m0=160000;q=2500;g=9.8;
3 out=u.*log(m0./(m0-q.*t))-g.*t;
4 end
```

- 定义数值积分是离散点上的函数值的线性组合

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

$$I = I_n(f) + R_n(f)$$

求积公式的余项

称为求积系数，与 $f(x)$ 无关，与积分区间和积分点有关

称为求积节点，与 $f(x)$ 无关

- 两个问题：
 - 求积系数如何选取
 - 节点可以自由选取，取什么点好

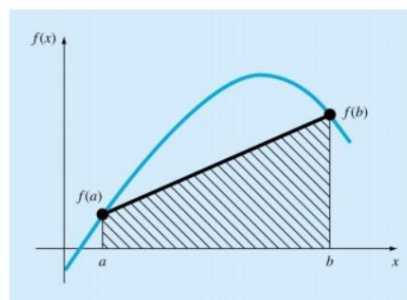
代数精度

- 如果一个求积公式对任何次数不超过 m 次的多项式 $P_m(x)$ 都准确成立

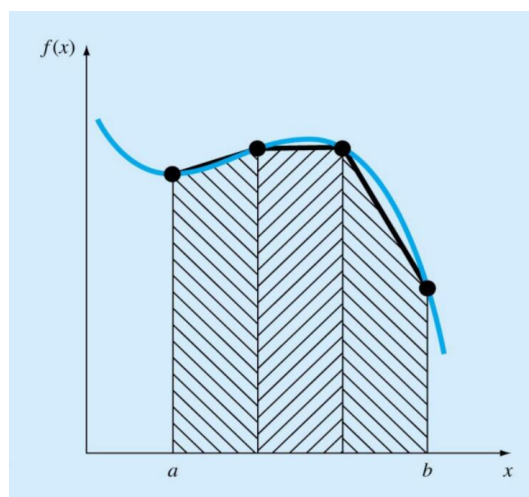
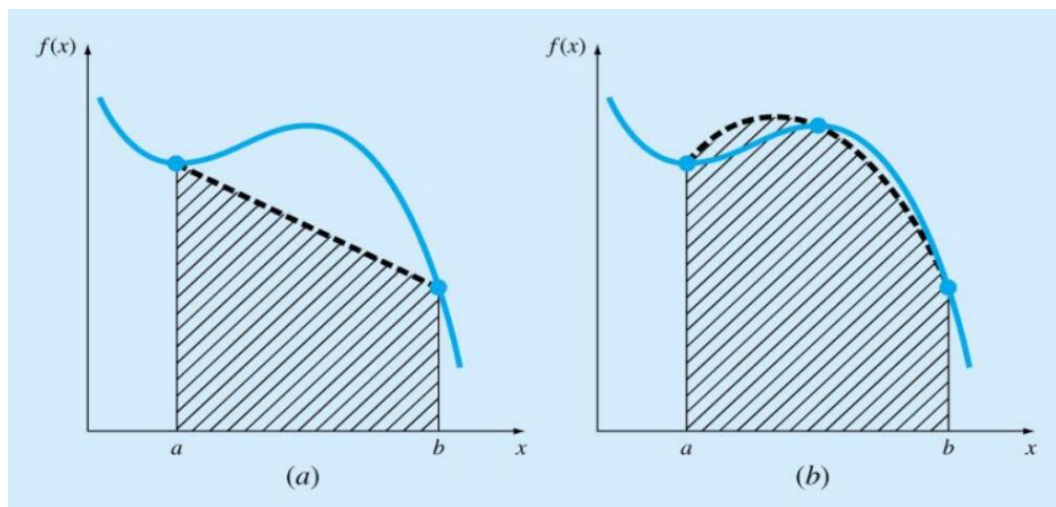
$$\int_a^b P_m(x) dx = \sum_{k=1}^n A_k P_m(x_k)$$

阶次为 $m+1$ 时， $R(f) \neq 0$ ，则称求积公式 $I_n(f)$ 的代数精度是 m

- 梯形公式的代数精度 $m=1$
 - 对不高于1次的代数多项式都准确成立
 - 对2次以上的代数多项式存在误差
- 一个求积公式的代数精度越高，就能对更多的被积函数准确或较准确地成立



3.1 Newton-Cotes 积分



Newton-Cotes 积分

- 采用拉格朗日插值多项式 $P_n(x)$ 来逼近 $f(x)$, 将积分区间 n 等分, 取分点为求积节点, 并作变量替换

$$h = \frac{b-a}{n}, x_i = a + ih, i = 0, \dots, n \quad x = a + th,$$

$$\begin{aligned} A_i &= \int_a^b l_i(x) dx = \int_0^n \frac{t(t-1)\cdots(t-i+1)(t-i-1)\cdots(t-n)}{i!(n-i)!(-1)^{n-i}} h dt \\ &= nh \cdot \frac{(-1)^{n-i}}{i!(n-i)!n} \int_0^n t(t-1)\cdots(t-i+1)(t-i-1)\cdots(t-n) dt \end{aligned}$$

$b-a$

与步长 h 无关, 可以预先求出 $C_i^{(n)}$ —— Cotes 系数

$$A_i = (b-a)C_i^{(n)}$$

3.1.1 梯形公式

梯形公式

$I = (b-a) \times \text{平均高度}$

- $n=1$

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

$$C_1^{(1)} = \int_0^1 t dt = \frac{1}{2}$$

$$E_1(f) = \int_a^b \frac{f''(\xi)}{2!} (x-a)(x-b) dx$$

$$= \frac{f''(\xi)}{2!} \int_a^b (x-a)(x-b) dx$$

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

$$I_1(f) = (b-a) \frac{1}{2} f(a) + (b-a) \frac{1}{2} f(b)$$

$$= (b-a) \frac{f(a) + f(b)}{2}$$

```

1 %% Trapezoidal formula
2 clc, clear
3 I_real = 10879.6194048972;
4 a = 0;
5 b = 30;
6 I = (b-a) * (v(a) + v(b)) / 2;
7 et = abs((I - I_real) / I_real);
8 e = abs(I - I_real);

```

积分结果	与真值相对误差	与真值绝对误差
1.266810908607478e+04	0.164388993274208	1.788489681177583e+03

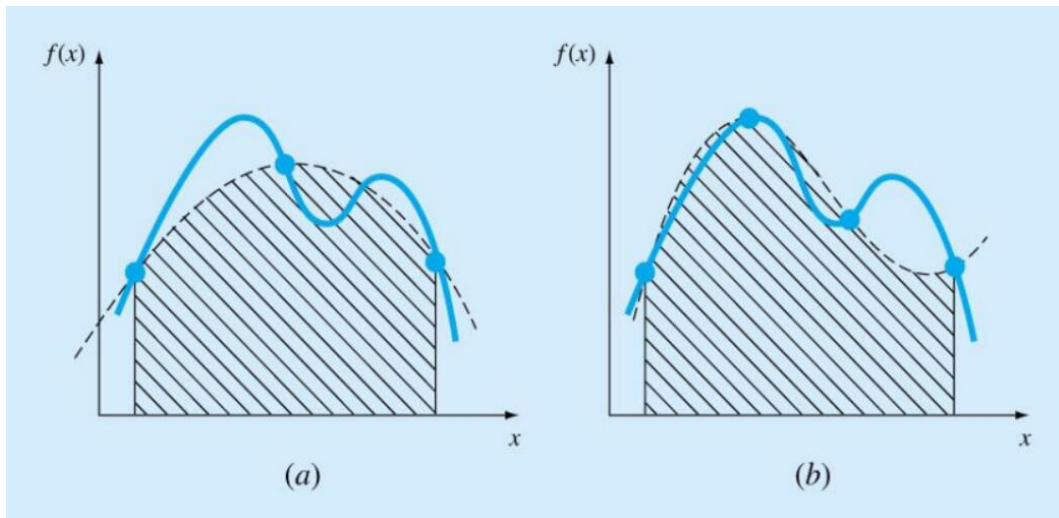
估计误差：用原函数二阶导的积分平均值代替 $f''(\zeta)$ ：

$$f''(x) = \frac{\int_0^{30} \left(\frac{1800}{(64-t)^2} \right) dt}{30-0} = 0.8272 \quad (4)$$

$$E_a = \left| -\frac{1}{12} * 0.8272 * 30^3 \right| = -1.8612e + 03 \quad (5)$$

可以看出，直接使用梯形公式的误差较大，且估计误差的绝对值略大于真实绝对误差的绝对值，误差估计有效。

3.1.2 Simpson 公式 (Simpson1/3 法则)



Simpson公式

还可由扩展的4阶牛顿插值多项式证明

$$\begin{aligned} n=2 \\ C_0^{(2)} &= \frac{1}{4} \int_0^2 (t-1)(t-2) dt = \frac{1}{6} \\ C_1^{(2)} &= -\frac{1}{2} \int_0^2 t(t-2) dt = \frac{4}{6} \\ C_2^{(2)} &= \frac{1}{4} \int_0^2 t(t-1) dt = \frac{1}{6} \end{aligned}$$

$$\begin{aligned} E_2(f) &= \int_a^b \frac{f^{(4)}(\xi)}{4!} (x-a) \left(x - \frac{a+b}{2}\right)^2 (x-b) dx \\ &= \frac{f^{(4)}(\xi)}{4!} \int_a^b (x-a) \left(x - \frac{a+b}{2}\right)^2 (x-b) dx \\ &= -\frac{1}{90} \left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi) = -\frac{1}{90} h^5 f^{(4)}(\xi) \end{aligned}$$

$$\begin{aligned} I_2(f) &= (b-a) \frac{1}{6} f(a) + (b-a) \frac{4}{6} f\left(\frac{b+a}{2}\right) + (b-a) \frac{1}{6} f(b) \\ &= \frac{h}{3} \left[f(a) + 4f\left(\frac{b+a}{2}\right) + f(b) \right] \end{aligned}$$

Simpson1/3
法则

$I=(b-a) \times \text{平均高度}$

数值计算方法

34

```
1 %% Simpson 1/3
2 clc , clear
3 I_real=10879.6194048972;
4 a=0;
5 b=30;
6 I=(b-a)/6*v(a)+(b-a)*4/6*v((a+b)/2)+(b-a)/6*v(b);
7 et=abs((I-I_real)/I_real);
```



```
8 e=abs(I-I_real);
```

积分结果	与真值相对误差	与真值绝对误差
1.089696329765722e+04	0.001594163556145	17.343892760020026

估计误差：用原函数四阶导的积分平均值代替 $f^{(4)}(\zeta)$ ：

$$f^{(4)}(x) = \frac{\int_0^{30} \left(\frac{10800}{(64-t)^4}\right) dt}{30-0} = 0.0025953607 \quad (6)$$

$$E_a = \left| -\frac{1}{90} * 0.0025953607 * 15^5 \right| = 21.8983 \quad (7)$$

可以看出，误差比直接使用梯形公式的小了一些，且估计误差的绝对值略大于真实绝对误差的绝对值，误差估计有效。

3.1.3 Simpson3/8 法则

Simpson 3/8 法则

- $n=3$ ，构造区间上的四点三次Lagrange插值多项式，并逼近被积函数

$$\begin{aligned} I(f) &= \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] \\ &= (b-a) \frac{[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]}{8} \end{aligned}$$

$$E(f) = -\frac{(b-a)^5}{6480} f^{(4)}(\xi) = -\frac{3}{80} h^5 f^{(4)}(\xi)$$

$I=(b-a) \times \text{平均高度}$

具有与Simpson公式相同阶次的代数精度，比Simpson公式精确一点。一般情况下，优先使用Simpson公式。

```
1 %% Simpson 3/8
2 clc , clear
3 I_real=10879.6194048972;
4 a=0;
5 b=30;
6 x0=0;x1=10;x2=20;x3=30;
```



```

7 I=(b-a)*(v(x0)+3*v(x1)+3*v(x2)+v(x3))/8;
8 et=abs((I-I_real)/I_real);
9 e=abs(I-I_real);

```

积分结果	与真值相对误差	与真值绝对误差
1.088752511781406e+04	7.266534446327055e-04	7.905712916861376

估计误差：用原函数四阶导的积分平均值代替 $f^{(4)}(\zeta)$ ：

$$f^{(4)}(x) = \frac{\int_0^{30} (\frac{10800}{(64-t)^4}) dt}{30-0} = 0.0025953607 \quad (8)$$

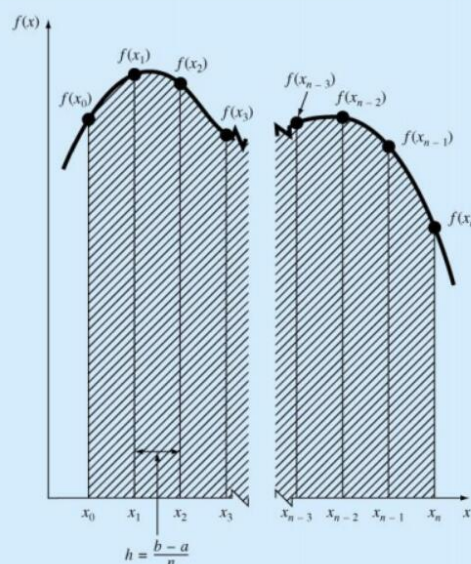
$$E_a = | -\frac{3}{80} * 0.0025953607 * 10^5 | = 9.7326 \quad (9)$$

可以看出，Simpson3/8 法则比 Simpson1/3 法则的误差又的小了一些，且估计误差的绝对值略大于真实绝对误差的绝对值，误差估计有效。

3.1.4 复合梯形公式

复合Newton-Cotes公式

- 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。



复合梯形公式

$$h = \frac{b-a}{n}, x_i = a + ih, i = 0, \dots, n$$

$$f \in C^2[a, b] \Rightarrow \exists \xi \in [a, b], \text{s.t.}, \sum_{i=0}^{n-1} f''(\xi_i) = nf''(\xi)$$

$$\int_{x_i}^{x_{i+1}} f(x) dx = \frac{h}{2} (f(x_i) + f(x_{i+1})) - \frac{h^3}{12} f''(\xi_i)$$

$$\begin{aligned} T_n(f) &= \sum_{i=0}^{n-1} \left\{ \frac{h}{2} (f(x_i) + f(x_{i+1})) - \frac{h^3}{12} f''(\xi_i) \right\} \\ &= h \left(\frac{1}{2} f(a) + \sum_{i=1}^{n-1} f(x_i) + \frac{1}{2} f(b) \right) - \sum_{i=0}^{n-1} \frac{h^3}{12} f''(\xi_i) \end{aligned}$$

$$\begin{aligned} E_n(f) &= -\frac{nh^3}{12} f''(\xi) \\ &= -\frac{h^2}{12} (b-a) f''(\xi) \\ &= -\frac{(b-a)^3}{12n^2} f''(\xi) \end{aligned}$$

$$I = (b-a) \frac{\left(f(a) + 2 \sum_{i=1}^{n-1} f(x_i) + f(b) \right)}{2n}$$

$I = (b-a) \times \text{平均高度}$

```

1 %% combined trapz
2 clc, clear
3 I_real = 10879.6194048972;
4 a = 0;
5 b = 30;
6 n = 100;
7 x = linspace(a, b, n+1);
8 h = (b-a)/n;
9 I = (v(x(1)) + v(x(n+1))) / 2 + sum(v(x(2:n))) * h;
10 I = I * h;
11 et = abs((I - I_real) / I_real);
12 e = abs(I - I_real);

```

复合梯形法的误差可以被控制。在计算中，由于 $f''(x)$ 是一个下凸函数，因此用原函数二阶导的积分平均值代替 $f''(\zeta)$ 来估计误差是可以保证不等号方向的。

$$f''(x) = \frac{\int_0^{30} \left(\frac{1800}{(64-t)^2} \right) dt}{30-0} = 0.8272 \quad (10)$$

若要使绝对误差限 $E = 5e - 05$ ，可以由公式算出：

$$n \geq \sqrt{\frac{30^3}{12} * \frac{f''(\zeta)}{E}} = 6101.15 \quad (11)$$

因此，理论上 n 大于等于 6102 即可。

n	积分结果	与真值相对误差	与真值绝对误差
1	1.266810908607478e+04	0.164388993274208	1.788489681177583e+03
100	1.087980552534478e+04	1.710725721633231e-05	0.186120447575377
1000	1.087962126611031e+04	1.710733657619924e-07	0.001861213109805
5000	1.087961947934569e+04	6.842931433761493e-09	7.444848961313255e-05
6101	1.087961945489993e+04	4.596000174000550e-09	5.000273267796729e-05
6102	1.087961945488355e+04	4.594494606666866e-09	4.998635267838836e-05
10000	1.087961942350929e+04	1.710730266958566e-09	1.861209420894738e-05

经过实践验证,我惊讶地发现理论结果与实际结果恰好吻合,正好是在 $n=6102$ 处绝对误差降到 $5e-05$ 以下,因此用原函数二阶导的积分平均值代替 $f''(\zeta)$ 来估计误差非常的有效。

3.1.5 复合 Simpson 公式

复合Simpson公式

$$h = \frac{b-a}{n}, x_i = a + ih, i = 0, \dots, n$$

$$S_n = \int_a^b f(x) dx \approx \sum_{k=1}^n \int_{x_{k-1}}^{x_k} f(x) dx = \frac{h}{6} \sum_{k=1}^n (f(x_{k-1}) + 4f(x_{k-\frac{1}{2}}) + f(x_k))$$

$$= \frac{h}{6} \left[f(a) + f(b) + 4 \sum_{k=1}^n f(x_{k-\frac{1}{2}}) + 2 \sum_{k=1}^{n-1} f(x_k) \right]$$

$$n = 2m \quad \text{偶数个子区间}$$

$$\int_{x_{2i}}^{x_{2i+2}} f(x) dx = \frac{2h}{6} (f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})) - \frac{(2h)^5}{2880} f^{(4)}(\xi_i)$$

$$S_n(f) = \sum_{i=0}^{n-1} \left\{ \frac{2h}{6} (f(x_{2i}) + f(x_{2i+1}) + f(x_{2i+2})) - \frac{(2h)^5}{2880} f^{(4)}(\xi_i) \right\}$$

$$= \frac{h}{3} \left(f(a) + 4 \sum_{i=0}^{m-1} f(x_{2i+1}) + 2 \sum_{i=1}^{m-1} f(x_{2i}) + f(b) \right) - \sum_{i=0}^{m-1} \frac{(2h)^5}{2880} f^{(4)}(\xi_i)$$

$$E_n(f) = -\frac{(2h)^5 m}{2880} f^{(4)}(\xi)$$

$$= -\frac{(b-a)^5}{2880 m^4} f^{(4)}(\xi)$$

$$= -\frac{(b-a)^5}{180 n^4} f^{(4)}(\xi)$$

误差 **48**

```

1 %% combined simpson
2 clc , clear
3 I_real=10879.6194048972;
4 a=0;
5 b=30;

```

```

6  n=100;
7  m=round(n/2);
8  x=linspace(a,b,2*m+1);
9  h=(b-a)/2/m;
10 I=(v(a)+v(b)+4*sum(v(x(2:2:2*m)))+2*sum(v(x(3:2:2*m-1))
    ));
11 I=I*h/3;
12 et=abs((I-I_real)/I_real);
13 e=abs(I-I_real);

```

积分结果	与真值相对误差	与真值绝对误差
1.087961940840046e+04	3.220018132555801e-10	3.503257175907493e-06

复合 Simpson 的误差可以被控制。在计算中, 由于 $f^{(4)}(x)$ 是一个下凸函数, 因此用原函数二阶导的积分平均值代替 $f^{(4)}(\zeta)$ 来估计误差是可以保证不等号方向的。

$$f^{(4)}(x) = \frac{\int_0^{30} \left(\frac{10800}{(64-t)^4} \right) dt}{30-0} = 0.0025953607 \quad (12)$$

$$n \geq \sqrt[4]{\frac{30^5}{180} * \frac{f^{(4)}(\zeta)}{E}} = 51.45 \quad (13)$$

因此, 理论上 n 大于等于 52 即可 (注意这里的 n 必须是偶数)。

n	积分结果	与真值相对误差	与真值绝对误差
2	1.089696329765722e+04	0.001594163556145	17.343892760020026
10	1.087965401388441e+04	3.181084367287731e-06	0.034608987210959
50	1.087961946092888e+04	5.150150772952511e-09	5.603168028756045e-05
52	1.087961945279512e+04	4.402536192757309e-09	4.789791819348466e-05
100	1.087961940840046e+04	3.220018132555801e-10	3.503257175907493e-06

经过实践验证, 理论结果与实际结果也是恰好吻合, 正好是在 $n=52$ 处绝对误差降到 $5e-05$ 以下, 因此用原函数四阶导的积分平均值代替 $f^{(4)}(\zeta)$ 来估计误差非常的有效。

3.1.6 非等距积分（步长自适应）

误差的事后估计与步长自适应

● 事后误差估计

● 复合梯形公式

步长自适应：利用 Simpson 公式，在对半分的区间上递归应用公式并估计误差，决定步长是否减半

$$I(f) - T_n(f) = -\frac{(b-a)}{12} h^2 f''(\xi) \quad n \text{ 等分区间}$$

$$I(f) - T_{2n}(f) = -\frac{(b-a)}{12} \left(\frac{h}{2}\right)^2 f''(\eta) \quad 2n \text{ 等分区间}$$

近似有

$$f''(\eta) \approx f''(\xi) \Rightarrow$$

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

● 复合 Simpson 公式

$$I(f) - S_{2n}(f) \approx \frac{1}{15} (S_{2n}(f) - S_n(f))$$

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

计算停止条件：

$$\frac{|T_{2n}(f) - T_n(f)|}{3} < \varepsilon$$

在区间逐步分半进行计算的过程中可以用前后两次计算结果来估计误差

2022/3/27

数值计算

```

1 %% Step size adaptation
2 clc, clear
3 I_real = 10879.6194048972;
4 tol = 5e-5;
5 a = 0;
6 b = 30;
7 c = (a+b)/2;
8 fa = v(a); fb = v(b); fc = v(c);
9 I = quadstep(a, b, tol, fa, fc, fb);
10 et = abs((I - I_real)/I_real);
11 e = abs(I - I_real);
12
13 function q = quadstep(a, b, tol, fa, fc, fb)
14 % Recursive subfunction used by quadadapt.
15 h = b - a; c = (a+b)/2;
16 fd = v((a+c)/2); fe = v((c+b)/2);
17 q1 = h/6 * (fa + 4*fc + fb);
18 q2 = h/12 * (fa + 4*fd + 2*fc + 4*fe + fb);
19 % 满足精度，跳出函数

```

```

20 if abs(q2-q1)<=tol
21     q=q2+(q2-q1)/15;
22 % 不满足精度, 继续迭代递归
23 else
24     qa=quadstep(a,c,tol,fa,fd,fc);
25     qb=quadstep(c,b,tol,fc,fe,fb);
26     q=qa+qb;
27 end
28 end

```

积分结果	与真值相对误差	与真值绝对误差
1.087961940491261e+04	1.416788182747394e-12	1.541411620564759e-08

从运行结果中可以看出, 实际与真值绝对误差在给定误差限以下, 步长自适应的非等距积分可以完成误差控制。

3.2 Romberg 积分

龙贝格积分

- 外推算法

- 用若干个积分近似值来推算更精确的新的近似值的方法

- 根据误差的事后估计

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

$$\bar{T}_n(f) = T_{2n}(f) + \frac{1}{3}(T_{2n}(f) - T_n(f)) = \frac{4}{3}T_{2n}(f) - \frac{1}{3}T_n(f)$$

比 T_{2n} 更好地接近积分真值

- P56例: $T_4 = 3.13117647$ $T_8 = 3.13898849$

$$\bar{T}_4 = \frac{4}{3} \times 3.13898849 - \frac{1}{3} \times 3.13117647 = 3.14159250$$

准确程度比 T_{512} 高, 却仅需九点!

- 验证可得 $I(f) \approx T_{2n}(f) + \frac{1}{3}(T_{2n}(f) - T_n(f)) = S_n(f)$

$$I(f) \approx S_{2n}(f) + \frac{1}{15}(S_{2n}(f) - S_n(f)) = C_n(f)$$

用低阶的公式组合后成为一个高阶的公式

$$R_n = C_{2n} + \frac{1}{63}(C_{2n} - C_n) = \frac{1}{63}(64C_{2n} - C_n)$$

龙贝格 (Romberg) 积分

Richardson外推法

$$I - G_{2n} = \frac{1}{4^m - 1}(G_{2n} - G_n) \quad m = 1, 2, 3, \dots$$

当 $m > 4$ 时上式趋近于0, 再进行外推已无必要。故在实际中只作到 R 为止。

计算过程

积分区间的二分数

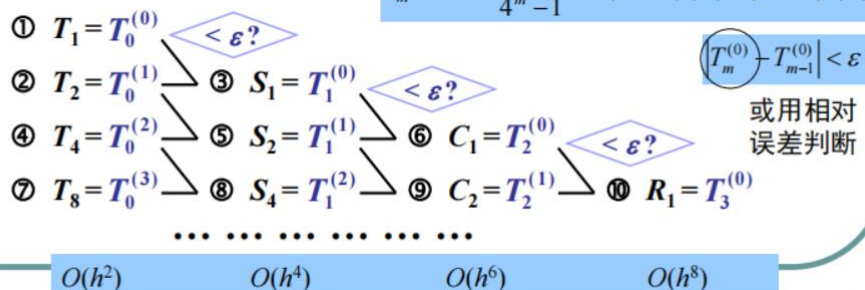
- 记号 $T_m^{(k)}$

近似值所在序列的性质, $m=0$ 在梯形序列中; $m=1$ 在Simpson序列中; $m=2$ 在Cotes序列中; $m=3$ 在Romberg序列中

$$T_0^{(0)} = \frac{b-a}{2}[f(a) + f(b)]$$

$$T_0^{(l)} = \frac{1}{2}T_0^{(l-1)} + \frac{b-a}{2^l} \sum_{i=1}^{2^{l-1}} f\left[a + (2i-1)\frac{b-a}{2^l}\right], l = 1, 2, 3, \dots$$

$$T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1}, k = 0, 1, 2, \dots; m = 1, 2, 3, \dots$$



3.2.1 普通 Romberg 积分

```
1 %% Romberg
2 clc, clear
3 I_real = 10879.6194048972;
4 a = 0;
5 b = 30;
6 j = 0; t = 5; % 迭代次数
7 M = [];
8 h = (b - a) / 2^j;
9 M(1, 1) = h * (v(a) + v(b)) / 2;
10 for j = 2:t
```

```

11     h=h/2;
12     M(j,1)=0.5*M(j-1,1)+h*sum(v(a+h*(1:2:2^(j-1)))));
13 end
14 for k=2:t
15     M(k:t,k)=M(k:t,k-1)+(M(k:t,k-1)-M(k-1:t-1,k-1))
        /(4^(k-1)-1); % 龙贝格公式
16 end
17 et=abs((M(5,5)-I_real)/I_real);
18 e=abs(M(5,5)-I_real);

```

积分结果	与真值相对误差	与真值绝对误差
1.087961940894866e+04	3.723895823567930e-10	4.051456926390529e-06

然而，传统的龙贝格积分难以控制误差，因此有下面改进后的龙贝格积分。

3.2.2 改进的 Romberg 积分

```

1 %% Romberg step adaptation
2 clc, clear
3 I_real=10879.6194048972;
4 a=0;
5 b=30;
6 tol=5e-10;% 给定误差
7 j=1;
8 M=[];
9 h=b-a;
10 M(1)=h*(v(a)+v(b))/2;
11 M(2)=inf;
12 while (abs(M(1)-M(2))>tol)% 估计相对误差
13     j=j+1;
14     h=h/2;
15     M(j)=0.5*M(j-1)+h*sum(v(a+h*(1:2:2^(j-1)))));
16     for k=j-1:-1:1
17         M(k)=M(k+1)+(M(k+1)-M(k))/(4^(j-k)-1);
18     end
19 end
20 I=M(1);

```

```

21 et=abs((M(1)-I_real)/I_real);
22 e=abs(M(1)-I_real);

```

积分结果	与真值相对误差	与真值绝对误差
1.087961940489716e+04	3.511039867559036e-15	3.819877747446299e-11

可以看出，利用改进后的龙贝格公式，成功将实际绝对误差降到了给定误差限以下。

3.3 Guass 求积公式

高斯求积公式

- 梯形公式
 - 限定节点为两个端点，代数精度为1
- 对节点不加限制，可以适当选取 x_0, x_1 ，使得积分公式具有更高的代数精度。

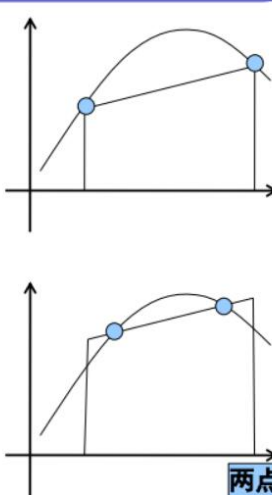
$$\int_a^b f(x)dx = A_0 f(x_0) + A_1 f(x_1)$$

- 假设积分区间为 $[-1, 1]$ ，若要求代数精度为3

$$\begin{aligned} A_0 &= A_1 = 1 \\ x_0 &= -\frac{\sqrt{3}}{3} \\ x_1 &= \frac{\sqrt{3}}{3} \end{aligned}$$

←

$$\begin{cases} A_0 \cdot 1 + A_1 \cdot 1 = \int_{-1}^1 1 dx = 2 \\ A_0 \cdot x_0 + A_1 \cdot x_1 = \int_{-1}^1 x dx = 0 \\ A_0 \cdot x_0^2 + A_1 \cdot x_1^2 = \int_{-1}^1 x^2 dx = \frac{2}{3} \\ A_0 \cdot x_0^3 + A_1 \cdot x_1^3 = \int_{-1}^1 x^3 dx = 0 \end{cases}$$



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

两点Gauss-Legendre公式

2022/3/27
66

两点Gauss-Legendre公式

- 积分区间变换 $[a,b] \rightarrow [-1,1]$

- 定义一个新变量 x_d

$$x = a_0 + a_1 x_d$$

- 下限 $x=a$, 对应于 $x_d=-1$

$$a = a_0 + a_1(-1)$$

- 上限 $x=b$, 对应于 $x_d=1$

$$b = a_0 + a_1(1)$$

- 可得

$$a_0 = \frac{b+a}{2}$$

$$a_1 = \frac{b-a}{2}$$

$$x = \frac{(b+a) + (b-a)x_d}{2} \Rightarrow dx = \frac{(b-a)}{2} dx_d$$

- 定理

- $n+1$ 个积分点的数值积分公式, 最高 $2n+1$ 阶

- 高斯求积公式

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

- 若代数精度达到 $2n+1$, 则称为**高斯求积公式**, 相应的求积节点称为**高斯点**

- 构造

- 待定系数法, x_k, A_k , 令 $f(x)=1, x, x^2, \dots, x^{2n+1}$ 使求积公式精确成立
- 首先利用 $[a,b]$ 上的 $n+1$ 次正交多项式确定高斯点, 再利用高斯点确定求积系数
 - $[a,b]$ 上的 $n+1$ 次正交多项式的零点是Gauss点

3.3.1 单段 Gauss 求积

```

1 %% Guass
2 clc, clear
3 I_real=10879.6194048972;
4 a=0;
5 b=30;
6 a0=(a+b)/2;
7 a1=(b-a)/2;
```

```

8 % three points
9 I=5/9*v_(-sqrt(15)/5,a0,a1)+8/9*v_(0,a0,a1)+5/9*v_(sqrt
    (15)/5,a0,a1);
10 et=abs((I-I_real)/I_real);
11 e=abs(I-I_real);
12
13 function out=v_(t,a0,a1)
14 out=v(a0+a1*t)*a1;
15 end

```

积分结果	与真值相对误差	与真值绝对误差
1.087942772294797e+04	1.761844253008763e-05	0.191681949234408

同样是取 3 个点，相比于 Simpson1/3 法则的误差 17.343892760020026，Simpson3/8 法则的误差 7.905712916861376，可以看出使用高斯求积公式可以将误差大大减小。

3.3.2 复合 Guass 求积

利用 Guass 求积公式时，较多采用复合求积的方法，将积分区间分成 m 个等长的小区间，在每个小区间上使用同一低阶 Gauss 求积公式算出积分的近似值，再相加。

```

1 %% combined guass
2 clc , clear
3 I_real=10879.6194048972;
4 a=0;
5 I=0;
6 n=100;% 分割段数
7 for b=30/n:30/n:30
8     a=b-30/n;
9     a0=(a+b)/2;
10    a1=(b-a)/2;
11    % three points
12    I=I+5/9*v_(-sqrt(15)/5,a0,a1)+8/9*v_(0,a0,a1)+5/9*
        v_(sqrt(15)/5,a0,a1);
13 end
14 et=abs((I-I_real)/I_real);

```

```

15 e=abs(I-I_real);
16
17 % 自变量转换
18 function out=v_(t,a0,a1)
19 out=v(a0+a1*t)*a1;
20 end

```

积分结果	与真值相对误差	与真值绝对误差
1.087961940489718e+04	1.671923746456684e-15	1.818989403545857e-11

同样相较 Simpson 公式，相同采样条件下误差大大减小。

3.4 数值积分总结

4 数值微分

4.1 差商近似

4.1.1 有限差商近似

有限差商近似

● 向前差商
$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}$$

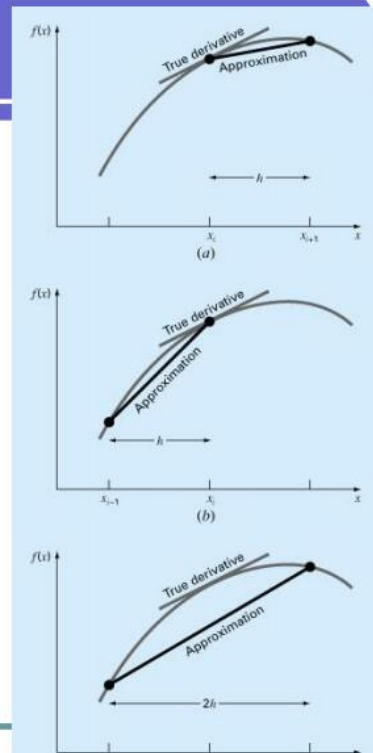
$$R(x) = f'(x_0) - \frac{f(x_0+h) - f(x_0)}{h} = -\frac{h}{2!} f''(\xi) = O(h)$$

● 向后差商
$$f'(x_0) \approx \frac{f(x_0) - f(x_0-h)}{h}$$

$$R(x) = f'(x_0) - \frac{f(x_0) - f(x_0-h)}{h} = \frac{h}{2!} f''(\xi) = O(h)$$

● 中心差商
$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h}$$

$$R(x) = f'(x_0) - \frac{f(x_0+h) - f(x_0-h)}{2h} = -\frac{h^2}{12} [f'''(\xi_1) + f'''(\xi_2)] = -\frac{h^2}{6} f'''(\xi) = O(h^2)$$



```

1 %% Difference quotient approximation

```



```
2  clc , clear
3  a=0;
4  b=30;
5  step=1e-4;
6  x=a:step:b;
7  dv=1800./(64-x)-9.8;
8  subplot(4,1,1);
9  plot(x,dv,'r');% 精确值
10 title('精确值');
11 h=1e-4;
12 for i=1:b/step+1
13     dv1(i)=(v(x(i)+h)-v(x(i)))/h;
14 end
15 subplot(4,1,2);
16 plot(x,dv1,'b');% 向前差商
17 title('向前差商');
18 for i=1:b/step+1
19     dv2(i)=(v(x(i))-v(x(i)-h))/h;
20 end
21 subplot(4,1,3);
22 plot(x,dv2,'b');% 向后差商
23 title('向后差商');
24 for i=1:b/step+1
25     dv3(i)=(v(x(i)+h)-v(x(i)-h))/2/h;
26 end
27 subplot(4,1,4);
28 plot(x,dv3,'b');% 中心差商
29 title('中心差商');
```

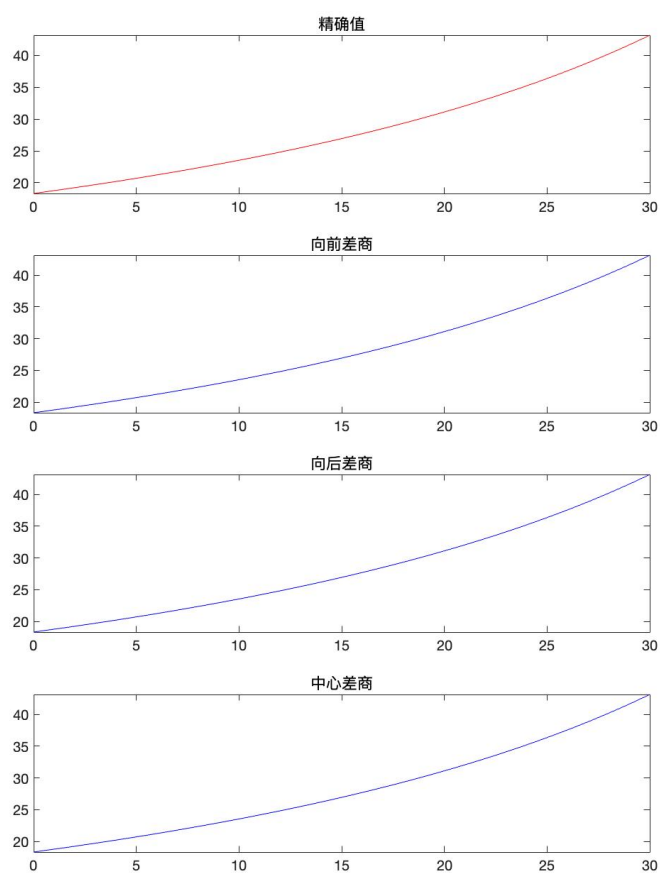
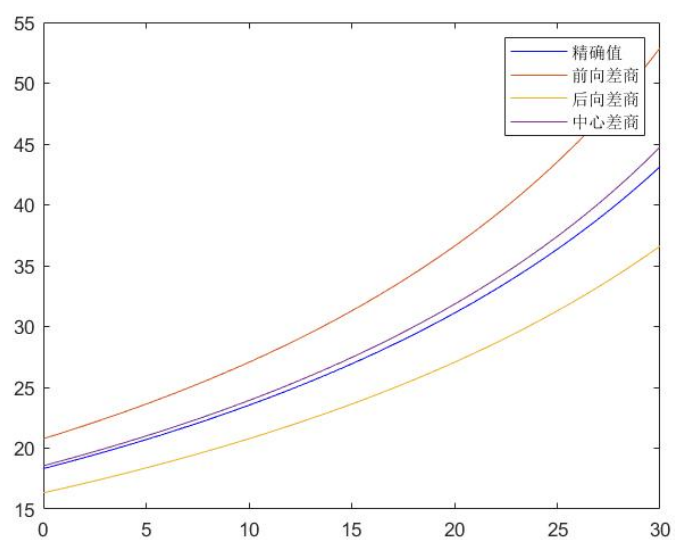
图 3: 有限差商近似 (步长 $1e-4$)

图 4: 有限差商近似 (步长 10)

4.1.2 高精度微分公式

高精度微分公式

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2}h^2 + \dots$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2}h + O(h^2)$$

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i))}{h^2} + O(h)$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i))}{2h^2}h + O(h^2)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i))}{2h} + O(h^2)$$



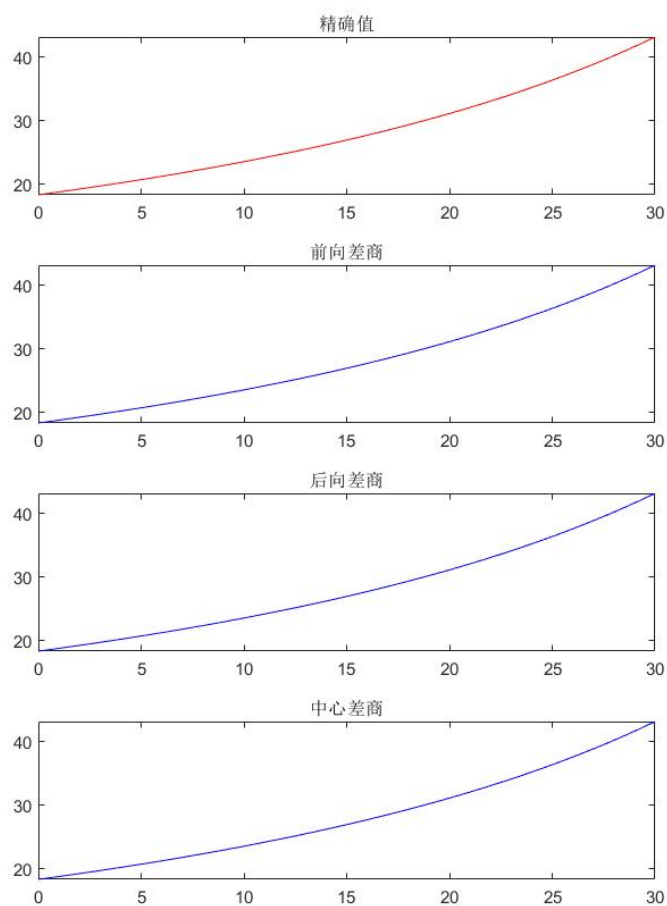
增加二阶导数项
可以将精度提高到
 $O(h^2)$

```

1 %% High precision difference quotient
2 clc , clear
3 a=0;
4 b=30;
5 step=1e-4;
6 x=a:step:b;
7 dv=1800./(64-x) -9.8;
8 subplot(4,1,1);
9 plot(x,dv,'r');% 精确值
10 title('精确值');
11 h=1e-4;
12 for i=1:b/step+1
13     dv1(i)=(-v(x(i)+2*h)+4*v(x(i)+h)-3*v(x(i)))/2/h;
14 end
15 subplot(4,1,2);
16 plot(x,dv1,'b');% 前向差商
17 title('前向差商');
18 for i=1:b/step+1
19     dv2(i)=(3*v(x(i))-4*v(x(i)-h)+v(x(i)-2*h))/2/h;
20 end

```

```
21 subplot(4,1,3);
22 plot(x,dv2,'b');% 后向差商
23 title('后向差商');
24 for i=1:b/step+1
25     dv3(i)=(-v(x(i)+2*h)+8*v(x(i)+h)-8*v(x(i)-h)+v(x(i)
        -2*h))/2/h;
26 end
27 subplot(4,1,4);
28 plot(x,dv3,'b');% 中心差商
29 title('中心差商');
```

图 5: 高精度微分公式 (步长 $1e-4$)

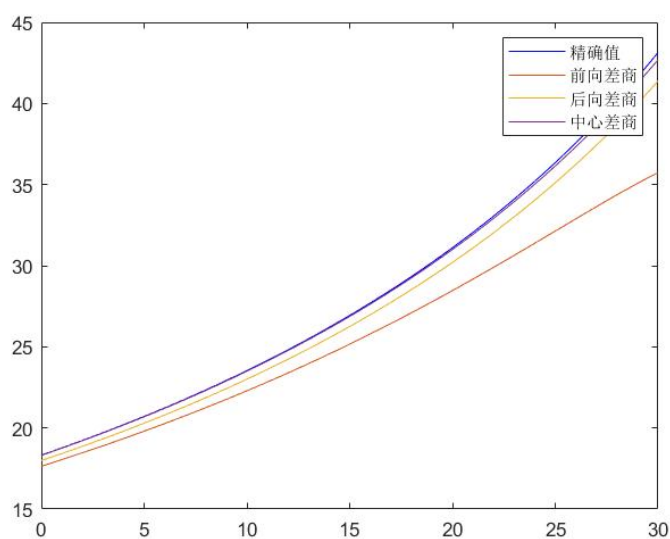


图 6: 高精度微分公式 (步长 10)

可以看出，步长较小时利用数值微分和使用解析法绘制出的加速度曲线基本相同。然而，当步长取的较大时（这里以 10 为例），高精度微分公式的优势就明显体现出来了。并且在同一精度的公式下，中心差商的误差明显小于前向差商和后向差商。

4.2 插值型数值微分

插值型数值微分

- 用插值函数的导数近似为原函数的导数

$$(x_j, f(x_j)) (j=0, 1, 2, \dots, n)$$

$$f(x) = L_n(x) + R_n(x)$$

$$f(x) = N_n(x) + R_n(x)$$

$$f^{(k)}(x) \approx P_n^{(k)}(x)$$

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

$$R_n^{(k)}(x) = \frac{d^k}{dx^k} \left[\frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x) \right]$$

```
2  clc , clear
3  step=1;
4  x0=0:step:30;
5  y0=v(x0);
6  % 三弯矩法方程组构造
7  for i=1:30/step
8      h(i)=x0(i+1)-x0(i);
9  end
10 for i=1:30/step-1
11     u(i)=h(i)/(h(i)+h(i+1));
12     ld(i)=1-u(i);
13     g(i)=6*((y0(i+2)-y0(i+1))/(x0(i+2)-x0(i+1))-(y0(i+1)-y0(i))/(x0(i+1)-x0(i)))/(h(i)+h(i+1));
14 end
15 A=zeros(30/step+1,30/step+1);
16 b=zeros(30/step+1,1);
17 A(1,1)=2;A(1,2)=1;
18 for i=2:30/step
19     A(i,i-1)=u(i-1);A(i,i)=2;A(i,i+1)=ld(i-1);
20 end
21 A(30/step+1,30/step)=1;
22 A(30/step+1,30/step+1)=2;
23 b(1)=6*((y0(2)-y0(1))/(x0(2)-x0(1))-0)/h(1);
24 for i=2:30/step
25     b(i)=g(i-1);
26 end
27 b(30/step+1)=6*(0-(y0(30/step+1)-y0(30/step))/(x0(30/step+1)-x0(30/step)))/h(30/step);
28 X=A\b;
29 % 绘图
30 step0=1e-5;
31 x=0:step0:30-step0;
32 count=1;
33 for t=0:step0:30-step0
34     i=floor(t/step)+1;
35     y(count)=1/(6*h(i))*(X(i)*(x0(i+1)-t)^3+X(i+1)*(t-x0(i))^3)+(y0(i)-X(i)*h(i)^2/6)*(x0(i+1)-t)/h(i)
```



```
        +(y0(i+1)-X(i+1)*h(i)^2/6)*(t-x0(i))/h(i);  
36    dy(count)=1/(6*h(i))*(X(i)*3*(x0(i+1)-t)^2*(-1)+X(i  
        +1)*3*(t-x0(i))^2)+(y0(i)-X(i)*h(i)^2/6)*(-1)/h(i  
        )+(y0(i+1)-X(i+1)*h(i)^2/6)*(1)/h(i);  
37    dv(count)=1800/(64-t)-9.8;  
38    count=count+1;  
39    end  
40    plot(x0,y0,'o');  
41    hold on;  
42    plot(x,y);  
43  
44    subplot(2,1,1)  
45    plot(x,dv,'r')  
46    subplot(2,1,2)  
47    plot(x,dy,'b')
```

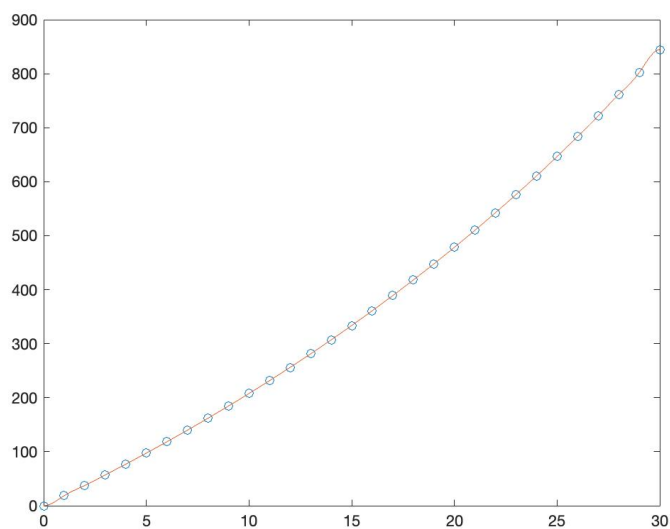


图 7: 三次样条插值结果

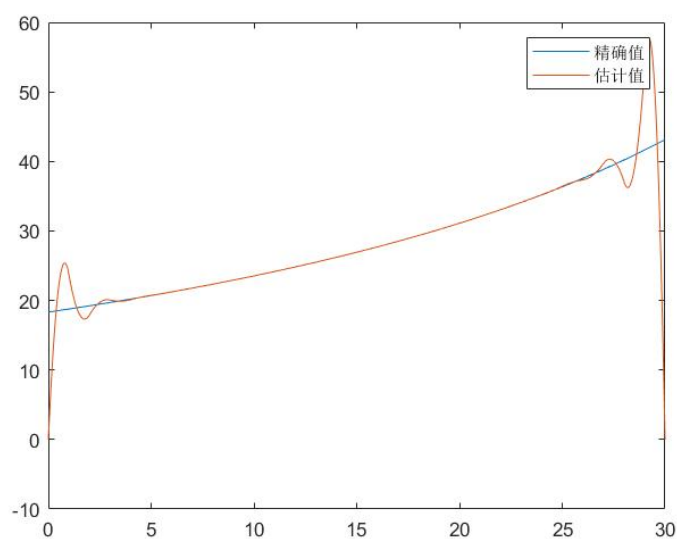


图 8: 利用三次样条插值估计微分的结果

可以看出，在中间部分用插值函数的导数来近似求原函数微分效果相当好，然而在一头一尾会出现类似龙格现象，并且边界条件确定也存在问题，求得的微分值偏差较大。

5 小结

本文从火箭发射过程的速度公式出发，分别使用数值积分和数值微分方法得到了火箭发射过程中的高度和加速度与时间的关系。本文对数值积分各种方法的优劣程度进行了讨论，并且分析、估计了相应的误差；同时，本文使用多种方法利用数值微分方法画出了加速度与时间的关系图，取得了良好的效果。