



第七章 常微分方程的数值解法

肖 明

微电子科学与技术学院

邮箱: xiaom37@mail.sysu.edu.cn



- 1 欧拉法
- 2 改进的欧拉法
- 3 龙格—库塔方法
- 4 **线性多步法**
- 5 **微分方程组的数值解法**



回顾-常微分方程的数值解法

研究一阶常微分方程的初值问题的数值解

已知初值问题的一般形式为：

$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = y_0 \end{cases} \quad a \leq x \leq b \quad (1)$$



回顾-Euler法

Euler方法

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) \\ y_0 = y(x_0) \end{cases}$$

显式算法

向后Euler方法

$$\begin{cases} y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \\ y(x_0) = y_0 \end{cases}$$

隐式算法

梯形法

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

隐式算法

改进Euler法

$$\tilde{y}_{k+1} = y_k + hf(x_k, y_k)$$

$$y_{k+1} = y_k + \frac{h}{2} (f(x_k, y_k) + f(x_{k+1}, \tilde{y}_{k+1}))$$

显式算法

可以从数值积分的角度理解几种方法的几何意义



回顾-龙格-库塔法

$$\begin{cases} y_{n+1} = y_n + h \sum_{i=1}^r c_i K_i \\ K_1 = f(x_n, y_n) \\ K_i = f(x_n + \lambda_i h, y_n + h \sum_{j=1}^{i-1} \mu_{ij} K_j) \end{cases}$$

这里 c_i, λ_i, μ_{ij} 均为常数.

上式称为 r 级Runge - Kutta法, 简称R - K方法.

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 4K_2 + K_3), \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right), \\ K_3 = f(x_n + h, y_n - hK_1 + 2hK_2). \end{cases}$$

三阶R-K方法

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ K_1 = f(x_n, y_n), \\ K_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right), \\ K_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2\right), \\ K_4 = f(x_n + h, y_n + hK_3). \end{cases}$$

四阶R-K方法

理解平均斜率的意义以理解R-K公式的含义



线性多步法



问题引入

单步法优点：只要给定初值，就可进行计算。

单步法缺点：计算 y_{n+1} 只用到前一步的信息 y_n 。

提高精度需通过增加非节点处函数值的计算，如4阶R-K公式需计算4个函数值。

线性多步法基本思想

充分利用前面已求得的多步信息，如 $y_0, y_1, \dots, y_n$ 来预测 y_{n+1} ，期望会获得较高的精度。



问题引入

在逐步推进的求解过程中，计算 y_{n+1} 之前事实上已经求出了一系列的近似值 $y_0, y_1, \dots, y_n$ ，如果充分利用前面多步的信息来预测 y_{n+1} ，则可以期望会获得较高的精度。这就是构造所得线性多步法的基本思想。

构造多步法的主要途径基于数值积分方法和基于泰勒展开方法，前者可直接由方程两端积分后利用插值求积公式得到。

定义： 求 y_{n+1} 用到前面 $y_n, y_{n-1}, \dots, y_{n-r}$ 等 $r+1$ 个值，且关于 y_{n-j} 和 f_{n-j} 都是线性的，因此称为线性 $r+1$ 步方法。



线性多步法的一般公式

如果计算 y_{n+k} 时，除用 y_{n+k-1} 的值，还要用到 y_{n+i} ($i=0,1,\cdots,k-2$)的值，则称此方法为**线性多步法**，其中步数为用到的函数值的个数。一般的线性多步法公式可表示为

$$y_{n+k} = \sum_{i=0}^{k-1} \alpha_i y_{n+i} + h \sum_{i=0}^k \beta_i f_{n+i} \quad \text{or} \quad y_{n+1} = \sum_{i=0}^k \alpha_i y_{n-i} + h \sum_{i=-1}^k \beta_i f_{n-i}$$

其中 y_{n+1} 为 $y(x_{n+1})$ 的近似， $f_{n+i}=f(x_{n+i}, y_{n+i})$ ，这里 $x_{n+i}=x_n+ih$ ， α_i, β_i 为常数， α_0 及 β_0 不全为零，则称之为**线性 k 步法**，计算时需先给出前面 k 个近似值 $y_0, y_1, \cdots, y_{k-1}$ ，再逐次求出 $y_k, y_{k+1}, \cdots$ 。



线性多步法的一般公式

如果 $\beta_k=0$ ，则称为**显式 k 步法**，这时 y_{n+k} 可直接算出；如果 $\beta_k\neq 0$ ，则称为**隐式 k 步法**，求解时与梯形法相同，要用**迭代法**方可算出 y_{n+k} 。式中系数 α_i 及 β_i 可根据方法的局部截断误差及阶确定。

定义： 设 $y(x)$ 是初值问题的准确解，线性多步法在 x_{n+k} 上局部截断误差为

$$\begin{aligned} R_{n+k} &= L[y(x_n); h] \\ &= y(x_{n+k}) - \sum_{i=0}^{k-1} \alpha_i y(x_{n+i}) - h \sum_{i=0}^k \beta_i y'(x_{n+i}) \end{aligned}$$

若 $R_{n+k}=O(h^{p+1})$ ，则称方法是 **p 阶的**。



线性多步法的一般公式

对 R_{n+k} 在 x_n 处泰勒展开, 由于

$$y(x_n + ih) = y(x_n) + ihy'(x_n) + \frac{(ih)^2}{2!} y''(x_n) + \frac{(ih)^3}{3!} y'''(x_n) + \cdots,$$

$$y'(x_n + ih) = y'(x_n) + ihy''(x_n) + \frac{(ih)^2}{2!} y'''(x_n) + \cdots.$$

代入 $R_{n+k} = L[y(x_n); h] = y(x_{n+k}) - \sum_{i=0}^{k-1} \alpha_i y(x_{n+i}) - h \sum_{i=0}^k \beta_i y'(x_{n+i})$

得 $R_{n+k} = c_0 y(x_n) + c_1 h y'(x_n) + c_2 h^2 y''(x_n) + \cdots + c_p h^p y^{(p)}(x_n) + \cdots$

其中 $c_0 = 1 - (\alpha_0 + \alpha_1 + \cdots + \alpha_{k-1}),$

$$c_1 = k - [\alpha_1 + 2\alpha_2 + \cdots + (k-1)\alpha_{k-1}] - (\beta_0 + \beta_1 + \cdots + \beta_k),$$



线性多步法的一般公式

$$R_{n+k} = c_0 y(x_n) + c_1 h y'(x_n) + c_2 h^2 y''(x_n) + \cdots + c_p h^p y^{(p)}(x_n) + \cdots$$

$$c_0 = 1 - (\alpha_0 + \alpha_1 + \cdots + \alpha_{k-1}),$$

其中 $c_1 = k - [\alpha_1 + 2\alpha_2 + \cdots + (k-1)\alpha_{k-1}] - (\beta_0 + \beta_1 + \cdots + \beta_k),$

$$c_q = \frac{1}{q!} [k^q - (\alpha_1 + 2^q \alpha_2 + \cdots + (k-1)^q \alpha_{k-1})] - \frac{1}{(q-1)!} (\beta_0 + 2^{q-1} \beta_1 + \cdots + k^{q-1} \beta_k)$$

若在公式中选择系数 α_i 及 β_i , 使它满足

$$c_0 = c_1 = \cdots = c_p = 0, \quad c_{p+1} \neq 0.$$

由定义可知此时所构造的多步法是 p 阶的, 且

$$R_{n+k} = c_{p+1} h^{p+1} y^{(p+1)}(x_n) + O(h^{p+2})$$

称右端第一项为局部截断误差主项, c_{p+1} 称为误差常数.



线性多步法的一般公式

若使得 $p \geq 1$, 即 $c_0 = c_1 = 0$, 可得

$$\begin{cases} \alpha_0 + \alpha_1 + \cdots + \alpha_{k-1} = 1, \\ \sum_{i=1}^{k-1} i \alpha_i + \sum_{i=0}^k \beta_i = k. \end{cases}$$

显然, 当 $k=1$ 时, 若 $\beta_1=0$, 则可求得 $\alpha_0=1, \beta_0=1$.

此时公式为欧拉公式 $y_{n+1} = y_n + hf_n$,

而 $c_2=1/2 \neq 0$, 故方法为1阶精度, 且局部截断误差为

$$R_{n+1} = \frac{1}{2} h^2 y''(x_n) + O(h^3).$$

这和上节给出的定义及结果是一致的.



线性多步法的一般公式

$$\begin{cases} \alpha_0 + \alpha_1 + \cdots + \alpha_{k-1} = 1, \\ \sum_{i=1}^{k-1} i\alpha_i + \sum_{i=0}^k \beta_i = k. \end{cases}$$

$$R_{n+k} = c_0 y(x_n) + c_1 h y'(x_n) + c_2 h^2 y''(x_n) + \cdots + c_p h^p y^{(p)}(x_n) + \cdots$$

对 $k=1$, 若 $\beta_1 \neq 0$, 此时方法为隐式公式, 为了确定系数 $\alpha_0, \beta_0, \beta_1$, 可由 $c_0=c_1=c_2=0$ 解得 $\alpha_0=1, \beta_0=\beta_1=1/2$. 于是得到公式

$$y_{n+1} = y_n + \frac{h}{2} (f_n + f_{n+1}). \text{ 即为梯形公式.}$$

可求得 $c_2=-1/12$, 故 $p=2$, 所以梯形法是二阶方法, 其局部截断误差主项是 $-\frac{h^3}{12} y'''(x_n)$. 这与上节中讨论是一致的.



常用的线性多步法公式

对 $k \geq 2$ 的多步法公式都可利用公式确定系数 α_i, β_i 和给出局部截断误差，下面只就若干常用的多步法导出具体公式。

(一) 阿达姆斯(Adams)公式

取 $r = 3$ ，并令 $\alpha_1 = \alpha_2 = \alpha_3 = \beta_{-1} = 0$ ，由方程组

$$\begin{cases} \sum_{i=0}^r \alpha_i = 1 \\ \sum_{i=0}^r (-i)^k \alpha_i + k \sum_{i=-1}^r (-i)^{k-1} \beta_i = 1 \quad (k = 1, 2, 3, 4) \end{cases}$$

可解得 $\alpha_0 = 1, \beta_0 = \frac{55}{24}, \beta_1 = -\frac{59}{24}, \beta_2 = \frac{37}{24}, \beta_3 = -\frac{9}{24}$ 。



常用的线性多步法公式

相应的线性多步公式为

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

因为 $\beta_{-1} = 0$ ，此式称为**Adams**显式公式，是四阶公式。

局部截断误差为

$$\begin{aligned} R_{n+1} &= \frac{h^5}{5!} [1 - \sum_{i=1}^3 (-i)^5 \alpha_i - 5 \sum_{i=-1}^3 (-i)^4 \beta_i] y_n^{(5)} + O(h^6) \\ &= \frac{251}{720} h^5 y_n^{(5)} + O(h^6) \end{aligned}$$



常用的线性多步法公式

如果令 $\alpha_1 = \alpha_2 = \alpha_3 = \beta_3 = 0$ ，由方程组可解得

$$\alpha_0 = 1, \beta_{-1} = \frac{9}{24}, \beta_0 = \frac{19}{24}, \beta_1 = -\frac{5}{24}, \beta_2 = \frac{1}{24}$$

相应的线性多步公式为

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$$

称其为四阶**Adams**隐式公式，其局部截断误差为

$$R_{n+1} = -\frac{19}{720}h^5 y_n^{(5)} + O(h^6)$$



Adams公式的典型例题

例 1

分别用四阶Adams显式和隐式公式求初值问题的数值解.

$$\begin{cases} y' = x - y & 0 \leq x \leq 1 \\ y(0) = 0 \end{cases} \quad \text{取 } h = 0.1$$

解

根据题意, $x_n = nh = 0.1n$, $f_n = x_n - y_n$,

由四阶Adams显示公式有

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{24} (55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}) \\ &= \frac{1}{24} (18.5 y_n + 5.9 y_{n-1} - 3.7 y_{n-2} + 0.9 y_{n-3} + 0.24n + 0.12) \end{aligned}$$



Adams公式的典型例题

由四阶Adams隐示公式有

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{24} (9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) \\ &= \frac{1}{24} (-0.9y_{n+1} + 22.1y_n + 0.5y_{n-1} - 0.1y_{n-2} + 0.24n + 0.12) \end{aligned}$$

由上式可解出

$$y_{n+1} = \frac{1}{24.9} (22.1y_n + 0.5y_{n-1} - 0.1y_{n-2} + 0.24n + 0.12)$$

利用精确解求出初始值(一般可用四阶R-K公式计算)后, 按上面的公式计算。



常用的线性多步法公式---Milne公式

取 $r=3$ ，并令 $\alpha_0 = \alpha_1 = \alpha_2 = \beta_{-1} = 0$ ，由方程组可解出

$$\alpha_3 = 1, \beta_0 = \frac{8}{3}, \beta_1 = -\frac{4}{3}, \beta_2 = \frac{8}{3}, \beta_3 = 0$$

相应的线性多步公式为

$$y_{n+1} = y_{n-3} + \frac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2})$$

称为Milne公式，其局部截断误差为

$$R_{n+1} = \frac{14}{45}h^5 y_n^{(5)} + O(h^6)$$

Milne公式是四阶四步显式公式。



常用的线性多步法公式---Hamming公式

取 $r = 2$ ，并令 $\alpha_1 = \beta_2 = 0$ ，可得到Hamming公式

$$y_{n+1} = \frac{1}{8}(9y_n - y_{n-2}) + \frac{3}{8}h(f_{n+1} + 2f_n - f_{n-1})$$

局部截断误差为

$$R_{n+1} = -\frac{1}{40}h^5 y_n^{(5)} + O(h^6)$$

Hamming公式是四阶三步隐式公式。

隐式法与显式法的比较

一般地，同阶的隐式法比显式法精确，而且数值稳定性也好。但在隐式公式中，通常很难解出 y_{n+1} ，需要用迭代法求解，这样又增加了计算量。

在实际计算中，很少单独用显式公式或隐式公式，而是将它们联合使用：先用显式公式求出 $y(x_{n+1})$ 的预测值，记作 $\bar{y}_{n+1}$ ，再用隐式公式对预测值进行校正，求出 $y(x_{n+1})$ 的近似值 y_{n+1} 。



预测-校正系统

用显式公式计算预测值，然后用隐式公式进行校正，得到近似值 y_{n+1} 这样一组计算公式称为预测—校正系统。

一般采用同阶的隐式公式与显式公式。常用的预测—校正系统有两种：

Adams预测——校正公式

$$\begin{cases} \bar{y}_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) & \text{预测} \\ y_{n+1} = y_n + \frac{h}{24}[9f(x_{n+1}, \bar{y}_{n+1}) + 19f_n - 5f_{n-1} + f_{n-2}] & \text{校正} \end{cases}$$

Milne—Hamming预测——校正公式

$$\begin{cases} \bar{y}_{n+1} = y_{n-3} + \frac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2}) \\ y_{n+1} = \frac{1}{8}(9y_n - y_{n-2}) + \frac{3}{8}h[f(x_{n+1}, \bar{y}_{n+1}) + 2f_n - f_{n-1}] \end{cases}$$

说明:

(1)以上两种预测—校正公式均为四阶公式，其起步值通常用四阶PK公式计算。

(2)有时为提高精度，校正公式可迭代进行多次，当迭代次数一般不超过3次。

(3)为减少一次迭代所产生的误差，常常用局部截断误差进一步修正预测值与校正值，得到更为精确的预测-校正公式。

预测-校正系统的程序实现



```
function [x,y]=adams_pre_corr(func,x0,y0,xn,h)
```

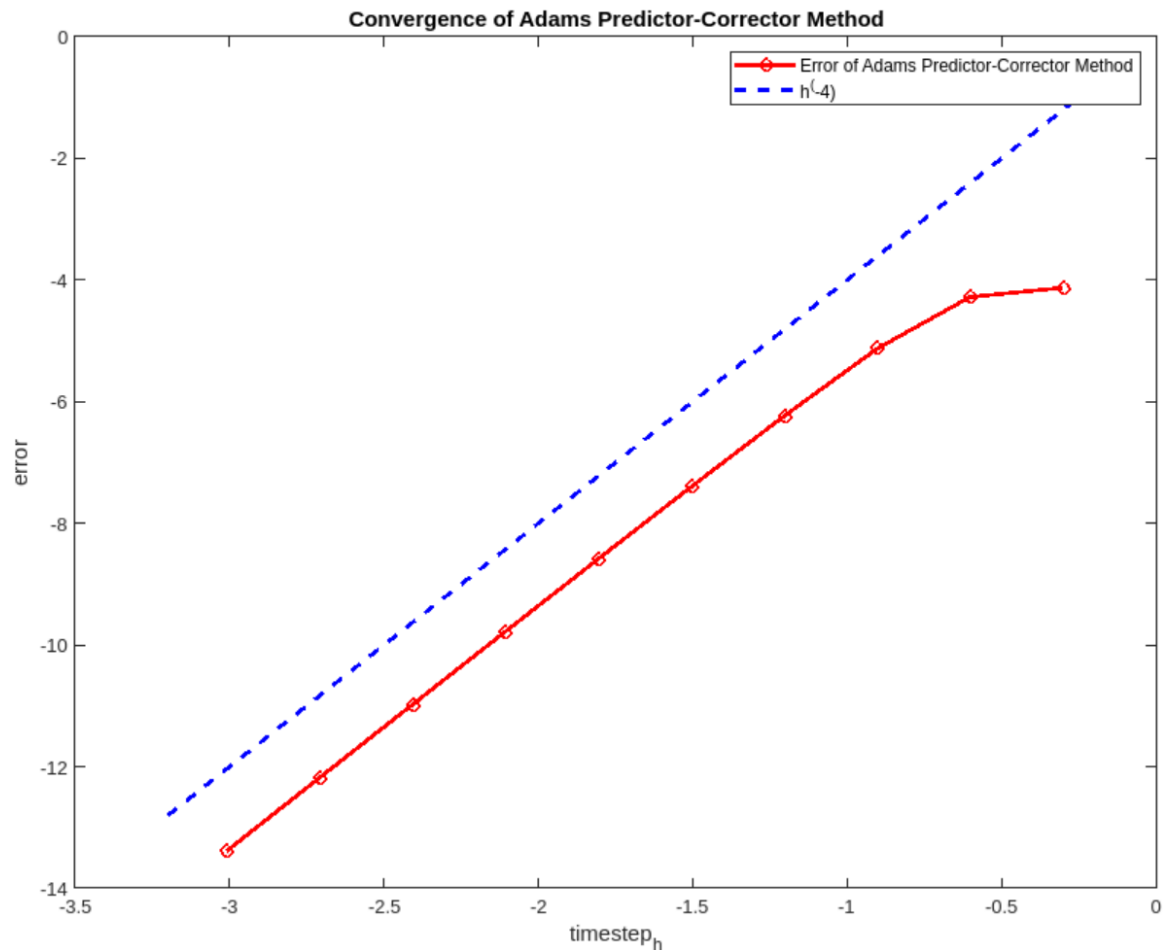
```
f=func;  
x(1)=x0;  
y(1)=y0;  
n=(xn-x0)/h;
```

```
%Runge_Kutta to generate the starting estimates
```

```
for i=1:3  
    k1 = f(x(i),y(i));  
    k2 = f(x(i)+h/2,y(i)+h/2*k1);  
    k3 = f(x(i)+h/2,y(i)+h/2*k2);  
    k4 = f(x(i)+h,y(i)+h*k3);  
    y(i+1) = y(i)+h/6*(k1+2*k2+2*k3+k4);  
    x(i+1) = x(i)+h;  
end
```

```
%iteration
```

```
for i=4:n  
    %by Adams-Bashforth method to 'predict'  
    y(i+1) = y(i)+h/24*(55*f(x(i),y(i))-59*f(x(i-1),y(i-1))...  
        +37*f(x(i-2),y(i-2))-9*f(x(i-3),y(i-3)));  
    x(i+1) = x(i)+h;  
    %by Adams-Moulton method to 'correct'  
    y(i+1) = y(i)+h/24*(9*f(x(i+1),y(i+1))+19*f(x(i),y(i))...  
        -5*f(x(i-1),y(i-1))+f(x(i-2),y(i-2)));  
end  
end
```





Matlab微分方程的求解器

求解函数	采用方法	适用场合
ode23	2阶或3阶Runge-Kutta算法, 低精度	非刚性
ode45	4阶或5阶Runge-Kutta算法, 中精度	非刚性
ode113	Adams算法, 精度可到10 ⁻³ ~ 10 ⁻⁶	非刚性, 计算时间比ode45短
ode23t	梯形算法	适度刚性
ode15s	Gear's反向数值微分算法, 中精度	刚性
ode23s	2阶Rosebrock算法, 低精度	刚性, 当精度较低时, 计算时间比ode15s短
ode23tb	梯形算法, 低精度	刚性, 当精度较低时, 计算时间比ode15s短

ode45 ➢ 解非刚性问题 ➢ 中等精度 ➢ 使用龙格—库塔法的四、五阶算法 ➢ 大部分情况下适用, 是解决问题的首选	ode23 ➢ 解非刚性问题 ➢ 低精度 ➢ 使用龙格—库塔法的二、三阶算法 ➢ 适用于容差较大或适度刚性问题	ode113 ➢ 解非刚性问题 ➢ 从低到高精度 ➢ 使用Adams-Bashforth-Moulton PECE法 ➢ 适用于容差要求严格, 或者常微分方程函数计算代价较大的情况。	ode15s ➢ 解刚性问题 ➢ 从低到中等精度 ➢ 使用可变阶次的数值微分(NDFs)算法 ➢ 在ode45计算很慢(由于刚性)或有质量矩阵时使用。	ode23s ➢ 解刚性问题 ➢ 低精度 ➢ 修正的Rosenbrock公式 ➢ 适用于对误差要求不高的情况或者有质量矩阵的情况	ode23t ➢ 解刚性问题 ➢ 低精度 ➢ 自由内插法的梯形法则 ➢ 适用于要求结果无数值衰减的适度刚性问题	ode23tb ➢ 解刚性问题 ➢ 低精度 ➢ TR-BDF2方法 ➢ 适用于对误差要求不高的刚性问题或者质量矩阵情况
---	--	--	--	---	--	--



微分方程组的数值解法



前面我们研究了单个方程 $y'=f$ 的数值解，只要把 y 和 f 理解为向量，那么，所提供的各种计算公式即可应用到一阶方程组的情形。

考察一阶方程组

$$y'_i = f_i(x, y_1, y_2, \dots, y_N), \quad (i = 1, 2, \dots, N)$$

的初值问题，初始条件为

$$y_i(x_0) = y_i^0, \quad (i = 1, 2, \dots, N).$$

若采用向量的记号，记(向量)

$$y = (y_1, y_2, \cdots, y_N)^T,$$

$$y_0 = (y_1^0, y_2^0, \cdots, y_N^0)^T,$$

$$f = (f_1, f_2, \cdots, f_N)^T.$$

则上述方程组的初值问题可表示为

$$\begin{cases} y' = f(x, y), \\ y(x_0) = y_0. \end{cases} \quad (5.1)$$

求解这一初值问题的四阶龙格-库塔公式为(向量)

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

式中(向量)

$$\begin{cases} k_1 = f(x_n, y_n), \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \\ k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \\ k_4 = f(x_n + h, y_n + hk_3). \end{cases}$$



一阶方程组

或表示为(分量)

$$y_{i,n+1} = y_{in} + \frac{h}{6}(K_{i1} + 2K_{i2} + 2K_{i3} + K_{i4})$$

$$(i = 1, 2, \dots, N),$$

其中(分量)

$$\begin{cases} K_{i1} = f_i(x_n, y_{1n}, y_{2n}, \dots, y_{Nn}), \\ K_{i2} = f_i(x_n + \frac{h}{2}, y_{1n} + \frac{h}{2}K_{11}, y_{2n} + \frac{h}{2}K_{21}, \dots, y_{Nn} + \frac{h}{2}K_{N1}), \\ K_{i3} = f_i(x_n + \frac{h}{2}, y_{1n} + \frac{h}{2}K_{12}, y_{2n} + \frac{h}{2}K_{22}, \dots, y_{Nn} + \frac{h}{2}K_{N2}), \\ K_{i4} = f_i(x_n + h, y_{1n} + hK_{13}, y_{2n} + hK_{23}, \dots, y_{Nn} + hK_{N3}). \end{cases}$$

这里 y_{in} 是第 i 个因变量 $y_i(x)$ 在节点 $x_n = x_0 + nh$ 的近似值.

为了帮助理解这一公式的计算过程，我们再考察两个方程的特殊情形

$$\begin{cases} y' = f(x, y, z), \\ z' = g(x, y, z), \\ y(x_0) = y_0, \\ z(x_0) = z_0. \end{cases}$$

这时四阶龙格-库塔公式具有形式

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ z_{n+1} = z_n + \frac{h}{6}(L_1 + 2L_2 + 2L_3 + L_4). \end{cases} \quad (5.2)$$



一阶方程组

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ z_{n+1} = z_n + \frac{h}{6}(L_1 + 2L_2 + 2L_3 + L_4). \end{cases}$$

其中

这是一步法，利用节点 x_n 上的值 y_n, z_n ，由(5.3)式顺序计算 $K_1, L_1, K_2, L_2, K_3, L_3, K_4, L_4$ ，然后代入(5.2)式即可求得节点 x_{n+1} 上的 y_{n+1}, z_{n+1} 。

$$\begin{cases} K_1 = f(x_n, y_n, z_n), \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1, z_n + \frac{h}{2}L_1), \\ K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2, z_n + \frac{h}{2}L_2), \\ K_4 = f(x_n + h, y_n + hK_3, z_n + hL_3), \\ L_1 = g(x_n, y_n, z_n), \\ L_2 = g(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1, z_n + \frac{h}{2}L_1), \\ L_3 = g(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2, z_n + \frac{h}{2}L_2), \\ L_4 = g(x_n + h, y_n + hK_3, z_n + hL_3). \end{cases} \quad (5.3)$$



化高阶方程为一阶方程组

关于高阶微分方程(或方程组)的初值问题, 原则上总可以归结为一阶方程组来求解. 例如, 考察下列 m 阶微分方程

$$y^{(m)} = f(x, y, y', \cdots, y^{(m-1)}), \quad (5.4)$$

初始条件为

$$y(x_0) = y_0, y'(x_0) = y'_0, \cdots, y^{(m-1)}(x_0) = y_0^{(m-1)}. \quad (5.5)$$

只要引进新的变量

$$y_1 = y, y_2 = y', \cdots, y_m = y^{(m-1)}.$$

即可将 m 阶方程(5.4)化为如下的一阶方程组:



化高阶方程为一阶方程组

$$\begin{cases} y_1' = y_2, \\ y_2' = y_3, \\ \dots\dots\dots \\ y_{m-1}' = y_m \\ y_m' = f(x, y_1, y_2, \dots, y_m). \end{cases} \quad (5.6)$$

初始条件(5.4)则相应地化为

$$y_1(x_0) = y_0, y_2(x_0) = y_0', \dots, y_m(x_0) = y_0^{(m-1)}. \quad (5.7)$$

不难证明初始条件(5.4),(5.5)和(5.6),(5.7)是彼此等价的.



化高阶方程为一阶方程组

特别地，对于下列二阶方程的问题

$$\begin{cases} y'' = f(x, y, y'), \\ y(x_0) = y_0, \\ y'(x_0) = y'_0. \end{cases}$$

引进新变量 $z=y'$,即可化为下列一阶方程组的初值问题:

$$\begin{cases} y' = z, \\ z' = f(x, y, z), \\ y(x_0) = y_0, \\ z(x_0) = y'_0. \end{cases}$$



化高阶方程为一阶方程组

针对这个问题应用**四阶龙格-库塔公式**(5.2), 有

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ z_{n+1} = z_n + \frac{h}{6}(L_1 + 2L_2 + 2L_3 + L_4). \end{cases}$$

$$K_1 = z_n, \quad L_1 = f(x_n, y_n, z_n);$$

由(5.3)式可得 $K_2 = z_n + \frac{h}{2}L_1, \quad L_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1, z_n + \frac{h}{2}L_1);$

$$K_3 = z_n + \frac{h}{2}L_2, \quad L_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2, z_n + \frac{h}{2}L_2);$$

$$K_4 = z_n + hL_3, \quad L_4 = f(x_n + h, y_n + hK_3, z_n + hL_3).$$



化高阶方程为一阶方程组

如果消去 K_1, K_2, K_3, K_4 , 则上述格式可表示为

$$\begin{cases} y_{n+1} = y_n + h z_n + \frac{h^2}{6} (L_1 + L_2 + L_3), \\ z_{n+1} = z_n + \frac{h}{6} (L_1 + 2L_2 + 2L_3 + L_4). \end{cases}$$

$$L_1 = f(x_n, y_n, z_n),$$

$$L_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} z_n, z_n + \frac{h}{2} L_1\right),$$

$$L_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} z_n + \frac{h^2}{4} L_1, z_n + \frac{h}{2} L_2\right),$$

$$L_4 = f\left(x_n + h, y_n + h z_n + \frac{h^2}{2} L_2, z_n + h L_3\right).$$

这里

化高阶方程为一阶方程组

例 1

用二阶R-K方法解常微分方程组:

$$\begin{cases} x'(t) = -y(t) - x(t)^2 = f(t, x, y) \\ y'(t) = 2x(t) - y(t) = g(t, x, y) \\ x(0) = y(0) = 1 \end{cases} \quad t \in [0, 2]$$

解

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \frac{h}{2} \left[\begin{pmatrix} k_1^{(1)} \\ k_1^{(2)} \end{pmatrix} + \begin{pmatrix} k_2^{(1)} \\ k_2^{(2)} \end{pmatrix} \right]$$

$$\begin{pmatrix} k_1^{(1)} \\ k_1^{(2)} \end{pmatrix} = \begin{pmatrix} f(t_n, x_n, y_n) \\ g(t_n, x_n, y_n) \end{pmatrix}$$

$$\begin{pmatrix} k_2^{(1)} \\ k_2^{(2)} \end{pmatrix} = \begin{pmatrix} f(t_n + h, x_n + hk_1^{(1)}, y_n + hk_1^{(2)}) \\ g(t_n + h, x_n + hk_1^{(1)}, y_n + hk_1^{(2)}) \end{pmatrix}$$

化高阶方程为一阶方程组



运行结果为:

1	0.1	0.7995	1.105
2	0.2	0.625306	1.16193
3	0.3	0.471024	1.18035
4	0.4	0.332603	1.16772
5	0.5	0.207335	1.12998
6	0.6	0.0933295	1.07202
7	0.7	-0.0107734	0.997913
8	0.8	-0.105978	0.911065
9	0.9	-0.193037	0.814378
10	1.	-0.272512	0.710335



算法稳定性和收敛性分析

数值解法的基本思想是，通过某种离散化手段将微分方程转化为差分方程，如单步法

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h)$$

它在点 x_n 处的解为 y_n ，而初值问题在点 x_n 处的精确解为 $y(x_n)$ ，记 $e_n = y(x_n) - y_n$ 称为整体截断误差. 收敛性就是讨论当 $x=x_n$ 固定且 $h = \frac{x_n - x_0}{n} \rightarrow 0$ 时 $e_n \rightarrow 0$ 的问题.

定义 若一种数值方法对于固定的 $x_n = x_0 + nh$, 当 $h \rightarrow 0$ 时有 $y_n \rightarrow y(x_n)$, 其中 $y(x)$ 是初值问题的准确解, 则称该方法是**收敛**的.

前面关于收敛性的讨论有个前提, 必须假定数值方法本身的计算是准确的. 实际情形并不是这样, 差分方程的求解还会有**计算误差**. 譬如由于数字舍入而引起的小扰动. 这类小扰动在传播过程中会不会恶性增长, 以至于“**淹没**”了差分方程的“**真解**”呢? 这就是**差分方程的稳定性问题**. 在实际计算时, 我们希望某一步产生的扰动值, 在后面的计算中**能够被控制**, 甚至是**逐步衰减**的.



算法的稳定性和收敛性

定义 若一种数值方法在节点值 y_n 上大小为 δ 的扰动，于以后各节点值 $y_m(m>n)$ 上产生的偏差均不超过 δ ，则称该方法是稳定的。

下面以欧拉法为例考察计算稳定性。

例 用欧拉公式求解初值问题

$$\begin{cases} y' = -100y, \\ y(0) = 1. \end{cases}$$

其准确解 $y(x) = e^{-100x}$ 是一个按指数曲线衰减很快的函数。

解 用欧拉法解方程 $y' = -100y$ 得

$$y_{n+1} = (1 - 100h)y_n.$$

若取步长 $h=0.025$ ，则欧拉公式的具体形式为 $y_{n+1} = -1.5y_n$ 。



算法的稳定性和收敛性

计算结果见表, 明显计算过程不稳定, 但取 $h=0.005$, $y_{n+1}=-1.5y_n$, 则计算过程稳定.

对后退的欧拉公式, 取 $h=0.025$ 时, 则计算公式为 $y_{n+1}=-(1/3.5)y_n$. 计算结果见表, 这时计算过程是稳定的.

节点 x_n	欧拉方法 y_n	后退欧拉方法 y_n
0.025	-1.5	0.2857
0.050	2.25	0.0816
0.075	-3.375	0.0233
0.100	5.0625	0.0067

例题表明稳定性不但与方法有关, 也与步长 h 有关, 当然与方程中的 $f(x, y)$ 有关.

1. 绝对稳定性

若某种数值解方法对 $y' = \lambda y$ ($\lambda < 0$) 是稳定的, 则称该方法时绝对稳定的;

2. 绝对稳定性区间

使数值方法绝对稳定的所有 " $z \triangleq \lambda h$ " 的集合。

3. 模型方程 $y' = \lambda y$ ($\lambda < 0$)



算法的稳定性分析-举例

Euler 公式

$$y_{n+1} = y_n + hf(x_n, y_n)$$

应用于模型方程 $y' = \lambda y \ (\lambda < 0)$

得到:
$$y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)y_n$$

设 y_n 有误差 ε_n , 则由此引起的 y_{n+1} 有误差 ε_{n+1} 满足:

$$y_{n+1} + \varepsilon_{n+1} = (1 + h\lambda)(y_n + \varepsilon_n)$$

$$\varepsilon_{n+1} = (1 + h\lambda)\varepsilon_n$$

为使得Euler公式绝对稳定, 则 $|\varepsilon_{n+1}| < |\varepsilon_n|$

令: $z = \lambda h$ 则有, $|1 + z| < 1 \quad -2 < z < 0$

所以Euler公式的绝对稳定区间为 $-2 < z < 0$ 步长h满足 $0 < h < \left| \frac{2}{\lambda} \right|$



算法的稳定性分析-举例

向后Euler 公式

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

应用于模型方程 $y' = \lambda y \ (\lambda < 0)$

$$\text{得到: } y_{n+1} = y_n + h\lambda y_{n+1}$$

设 y_n 有误差 ε_n , 则由此引起的 y_{n+1} 有误差 ε_{n+1} 满足:

$$y_{n+1} + \varepsilon_{n+1} = y_n + \varepsilon_n + h\lambda(y_{n+1} + \varepsilon_{n+1})$$

$$\text{则有, } \varepsilon_{n+1} = \frac{1}{(1 - h\lambda)} \varepsilon_n$$

为使得Euler 公式绝对稳定, 则 $|\varepsilon_{n+1}| < |\varepsilon_n|$

$$\text{令: } z = \lambda h \text{ 则有, } \frac{1}{|1 - z|} < 1 \quad -\infty < z < 0$$

所以向后Euler公式的绝对稳定区间为 $-\infty < z < 0$

1. 线性多步法的构造思想
2. 显式和隐式Adams方法的公式
3. 微分方程组的数值解法（向量的使用）
4. 算法的稳定性和收敛性（步长、绝对稳定性区间）



肖明

中山大学



谢谢大家！

肖 明

微电子科学与技术学院

邮箱: xiaom37@mail.sysu.edu.cn

电话: 1781722706

办公室: 中山大学珠海校区公共实验楼2楼A214室