

回顾

第9章 微分方程求解

9.1 引言

9.2 Euler方法

9.3 显式欧拉格式的改进

9.4 龙格-库塔方法

9.5 预报-矫正方法

9.6 微分方程组

9.7 边值问题

9.2 Euler方法

回顾

一、初值问题及其数值解的概念

一阶常微分方程初值问题：

$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = y_0, \quad x \geq x_0 \end{cases} \quad (*)$$

定理9.1 (解的存在唯一性)

对于初值问题 $(*)$ ，如果 $f(x, y)$ 在下列区域内连续：

$$G = \{a \leq x \leq b; |y| < \infty\}$$

且关于 y 满足Lipschitz条件，即存在常数 $L > 0$ ，使

$$|f(x, y_1) - f(x, y_2)| \leq L |y_1 - y_2|; \quad \forall x, y \in G$$

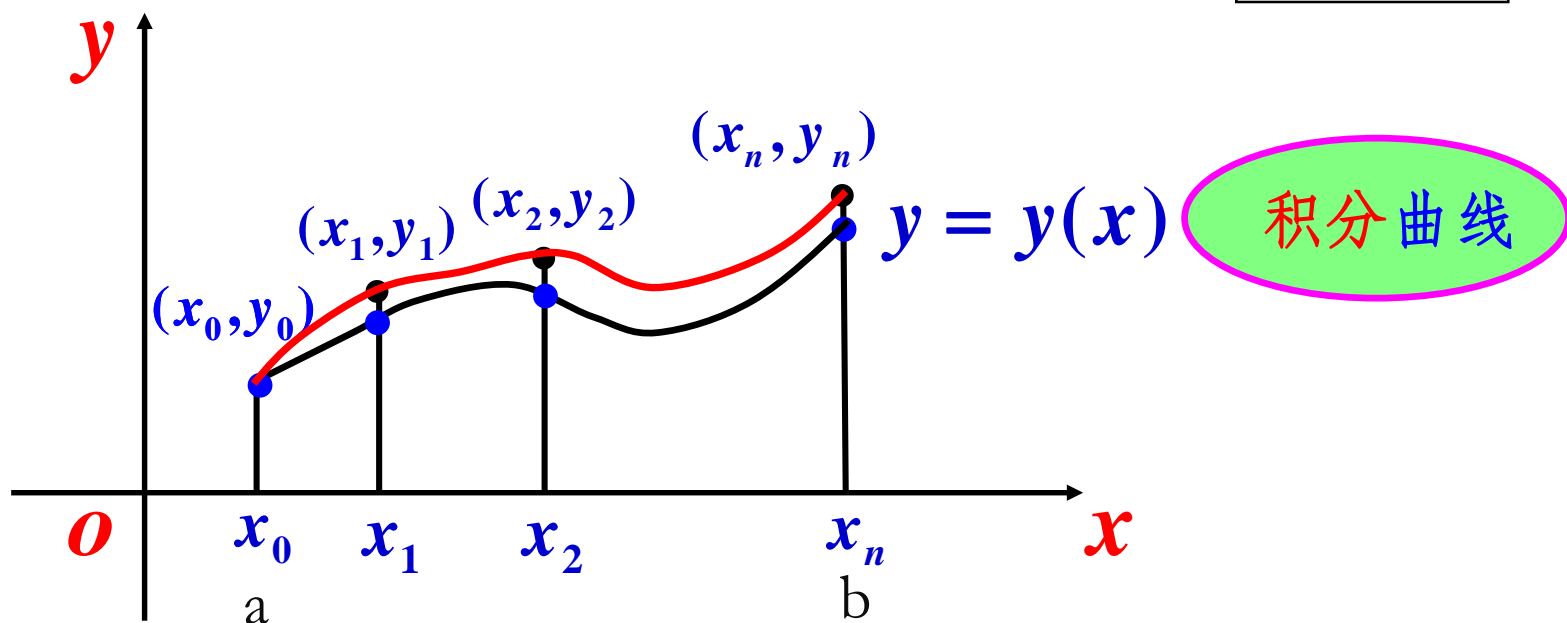
则初值问题 $(*)$ 存在唯一解，且解是连续可微的。

设 $f(t, y)$ 定义在区域 R 上，如果存在一个常数 $L > 0$ ，使得

$$|f_y(t, y)| \leq L, \quad (t, y) \in R$$

► 初值问题 (*) 的解析解及其数值解的几何意义:

回顾



初值问题 (*) 的解表示过点 (x_0, y_0) 的一条曲线

初值问题 (*) 的数值解表示一组离散点列 (x_i, y_i)

可用插值或拟合方法求该组数据 (x_i, y_i) 的近似曲线

9.2 Euler方法

回顾

考察初值问题：

$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = y_0; x \geq x_0 \end{cases}$$

将区间 $[a, b]$ 划分为 M 个等距子区间，并选择网格点

$$x_k = a + kh, \quad k = 0, 1, 2, \dots, M \text{ with } h = \frac{b-a}{M}$$

值 h 称为步长。

三类Euler方法的推导公式：

(1) 向前差商近似微分法

(2) Taylor展开法

(3) 数值积分法

欧拉格式

$$\begin{cases} y_0 = y(x_0) \\ y_{k+1} = y_k + hf(x_k, y_k), \end{cases} \quad k = 0, 1, \dots, n-1 \quad (9.1)$$

Euler格式的步长与误差

回顾

定义9.2: 若某算法的局部截断误差为 $O(h^{p+1})$, 则称该算法有 p 阶精度 (整体截断误差)。

局部离散误差 R_k 定义为

$$R_{k+1} = y(x_{k+1}) - hf(x_k, y_k) - y_k, \quad k=0,1,\dots,M-1$$

它是从 x_k 到 x_{k+1} 这一步计算的误差。

☞ 欧拉法的局部截断误差:

$$\begin{aligned} R_{i+1} &= y(x_{i+1}) - y_{i+1} = [\cancel{y(x_i)} + \cancel{hy'(x_i)} + \frac{h^2}{2} y''(x_i) + O(h^3)] - [\cancel{y_i} + \cancel{hf(x_i, y_i)}] \\ &= \frac{h^2}{2} y''(x_i) + O(h^3) = O(h^2) \end{aligned}$$

欧拉法具有 1 阶精度。

2). 梯形格式 — 显式、隐式两种算法的平均 又称为休恩方法

在区间 $[x_n, x_{n+1}]$ 上对初值问题两端积分：

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y) dx$$

回顾

$$\approx \frac{h}{2} [f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

然后用 y_n 代替 $y(x_n)$ ，得

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

用 $y_n + hf(x_n, y_n)$ 替代，休恩方法

称上述公式为改进的Euler 公式（梯形公式）。

$$\text{梯形公式} \int_a^b f(x) dx \approx \frac{b-a}{2} [f(a) + f(b)]$$

注：它有局部截断误差 $R_{i+1} = y(x_{i+1}) - y_{i+1} = O(h^3)$ ，即梯形格式具有2阶精度，比欧拉方法有了进步。但注意到该格式是隐式格式，计算时不得不用到迭代法，其迭代收敛性与欧拉格式相似。

3). 两步欧拉格式

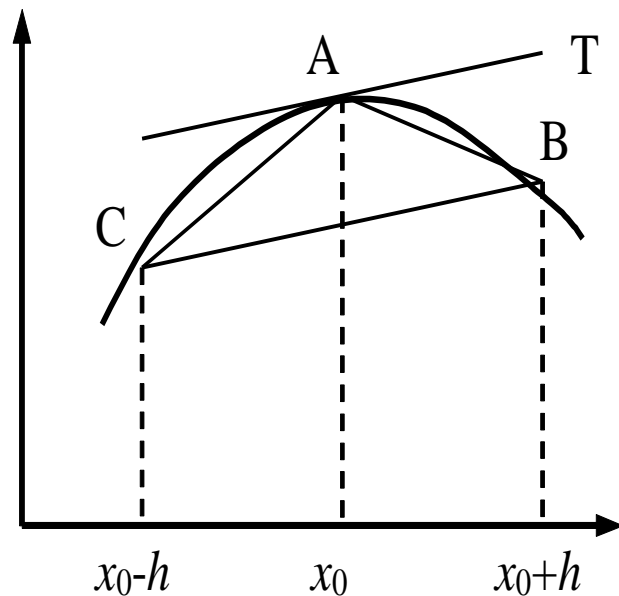
中心差商近似导数 $\rightarrow y'(x_1) \approx \frac{y(x_2) - y(x_0)}{2h}$

$\rightarrow y(x_2) \approx y(x_0) + 2h f(x_1, y(x_1))$

$$y_{i+1} = y_{i-1} + 2h f(x_i, y_i) \quad i = 1, \dots, n-1$$

假设 $y_{i-1} = y(x_{i-1})$, $y_i = y(x_i)$ ，则可以导出
 $R_{i+1} = y(x_{i+1}) - y_{i+1} = O(h^3)$

即两步Euler格式具有 2 阶精度。



需要2个初值 y_0 和 y_1 来启动递推过程，这样的算法称为两步法，而前面的三种算法都是单步法

方 法	优点	缺点
显式欧拉格式	简单	精度低
隐式欧拉格式	稳定性最好	精度低, 计算量大
梯形格式	精度提高	计算量大
两步欧拉格式	精度提高, 显式	多一个初值, 可能影响精度

能找到一个具有上面
所有优点的格式吗?

9.4 龙格-库塔 (Runge-Kutta) 法

前面所讨论的欧拉法与改进的欧拉法都是一步法，即计算 y_{n+1} 时，只用到前一步的值，但精度却不一样，欧拉法是1阶的，而改进的欧拉法却是2阶的。

龙格-库塔(Runge-Kutta)法(简称R-K方法)是一类高精度的一步法，这类方法与泰勒展开级数法有着密切的关系。

9.4 龙格-库塔 (Runge-Kutta) 法

一、泰勒级数法

设有初值问题
$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

由泰勒展开式

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!} y''(x_n) + \cdots + \frac{h^k}{k!} y^{(k)}(x_n) + O(h^{k+1})$$

若令 $y_{n+1} = y(x_n) + hy'(x_n) + \frac{h^2}{2!} y''(x_n) + \cdots + \frac{h^k}{k!} y^{(k)}(x_n)$ (9.2)

则 $y(x_{n+1}) - y_{n+1} = O(h^{k+1})$

即公式(9.2)为 k 阶龙格-库塔方法.

$$y_{n+1} = y(x_n) + hy'(x_n) + \frac{h^2}{2!} y''(x_n) + \cdots + \frac{h^k}{k!} y^{(k)}(x_n) \quad (9.2)$$

考虑k=1的情形

对于常微分方程的初值问题

$$\begin{cases} y' = f(x, y) & a \leq x \leq b \\ y(a) = y_0 \end{cases}$$

的解 $y=y(x)$ ，在区间 $[x_i, x_{i+1}]$ 上使用微分中值定理，有

$$y(x_{i+1}) - y(x_i) = y'(\xi_i) \underline{(x_{i+1} - x_i)}$$

其中 $\xi_i \in (x_i, x_{i+1})$

$$\text{即} \quad y(x_{i+1}) = y(x_i) + hy'(\xi_i)$$

引入记号

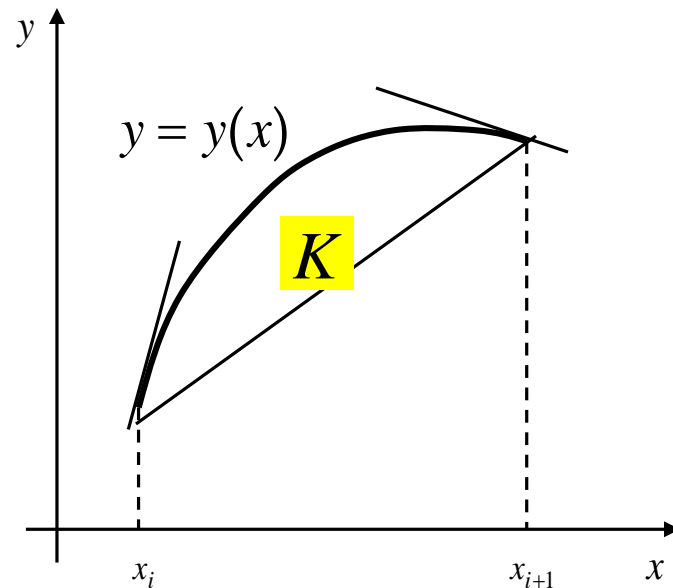
$$y(x_{i+1}) = y(x_i) + K \qquad K = hy'(\xi_i) = hf[\xi_i, y(\xi_i)]$$

$$y_{i+1} = y_i + K$$

K 可以认为是 $y = y(x)$ 在区间 $[x_i, x_{i+1}]$ 上的平均斜率

只要使用适当的方法求出 $y(x)$ 在区间 $[x_i, x_{i+1}]$ 上平均斜率的近似值 K

就可得到相应的Runge-Kutta方法



如果以 $y(x)$ 在 x_i 处的斜率作为 $y(x)$ 在 $[x_i, x_{i+1}]$ 上的平均斜率 K

即
$$K = hy'(x_i) = hf[x_i, y(x_i)]$$
$$= hf(x_i, y_i)$$

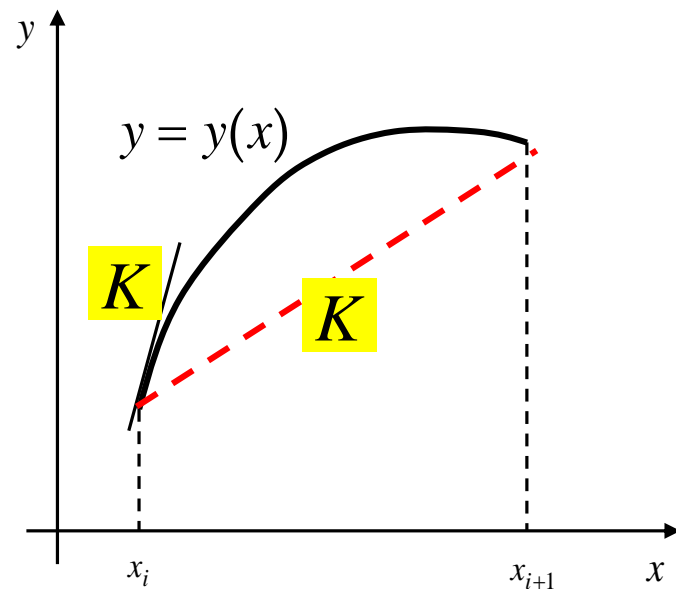
如下图

则上式化为

$$y_{i+1} = y_i + hf(x_i, y_i)$$

即Euler方法

$$\begin{cases} y_{n+1} = y_n + k_1 \\ k_1 = hf(x_n, y_n) \end{cases}$$



每步计算 f 的值一次，其截断误差为 $O(h^2)$.

Euler方法也称为一阶Runge-Kutta方法

当 $k=2$ 时

在 $[x_i, x_{i+1}]$ 上取两点 x_i 和 $x_{i+a_2} = x_i + a_2h$, 以该两点处的斜率值 K_1 和 K_2 的加权平均(或称为线性组合)来求取平均斜率 k^* 的近似值 K , 即

$$K = c_1 K_1 + c_2 K_2$$

式中: K_1 为 x_i 点处的切线斜率值 $K_1 = hf(x_i, y_i) = hy'(x_i)$

K_2 为 $x_i + a_2h$ 点处的切线斜率值, 比照改进的欧拉法, 将 x_{i+a_2} 视为 x_{i+1} , 即可得

$$K_2 = hf(x_i + a_2h, y_i + b_{21}K_1)$$

确定系数 c_1 、 c_2 、 a_2 、 b_{21} , 可得到有2阶精度的算法格式

因此 $y(x_{i+1}) = y(x_i) + K = y(x_i) + (c_1 K_1 + c_2 K_2)$

$$K_1 = hf(x_i, y_i) \quad K_2 = hf(x_i + a_2 h, y_i + b_{21} K_1)$$

将 $y(x_{i+1})$ **在** $x=x_i$ **处进行** *Taylor* **展开：**

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + hy'(x_i) + \frac{h^2}{2!} y''(x_i) + O(h^3) \\ &= y(x_i) + hf(x_i, y_i) + \frac{h^2}{2!} [f'_x + ff'_y] + O(h^3) \end{aligned}$$

将 $K_2 = hy'(x_i + a_2 h) = hf(x_i + a_2 h, y_i + b_{21} K_1)$

在 $x=x_i$ **处进行** *Taylor* **展开：**

$$K_2 = h \left[f(x_i, y_i) + a_2 h f'_x + b_{21} K_1 f'_y + O(h^2) \right]$$

$$= h \left[f(x_i, y_i) + a_2 h f'_x + b_{21} h f f'_y \right] + O(h^3)$$

$$K_1 = h f(x_i, y_i)$$

$$\begin{aligned} \therefore y_{i+1} &= y_i + (c_1 K_1 + c_2 K_2) \\ &= y(x_i) + c_1 h f(x_i, y_i) \\ &\quad + c_2 h \left[f(x_i, y_i) + a_2 h f'_x + b_{21} h f f'_y \right] + O(h^3) \\ &= y(x_i) + (c_1 + c_2) h f(x_i, y_i) \\ &\quad + a_2 c_2 h^2 f'_x + b_{21} c_2 h^2 f f'_y + O(h^3) \end{aligned}$$

与Taylor展开逐项系数比对：

$$y(x_{i+1}) = y(x_i) + h f(x_i, y_i) + \frac{h^2}{2!} \left[f'_x + f f'_y \right] + O(h^3)$$

令 $y(x_{i+1}) = y_{i+1}$ 对应项的系数相等，得到

$$c_1 + c_2 = 1, \quad a_2 c_2 = \frac{1}{2}, \quad b_{21} c_2 = \frac{1}{2}$$

这里有 4 个
未知数，3 个
方程。

存在无穷多个解。所有满足上式的格式统称为2阶
龙格 - 库塔格式。

$$y(x_{i+1}) = y(x_i) + K = y(x_i) + (c_1 K_1 + c_2 K_2)$$

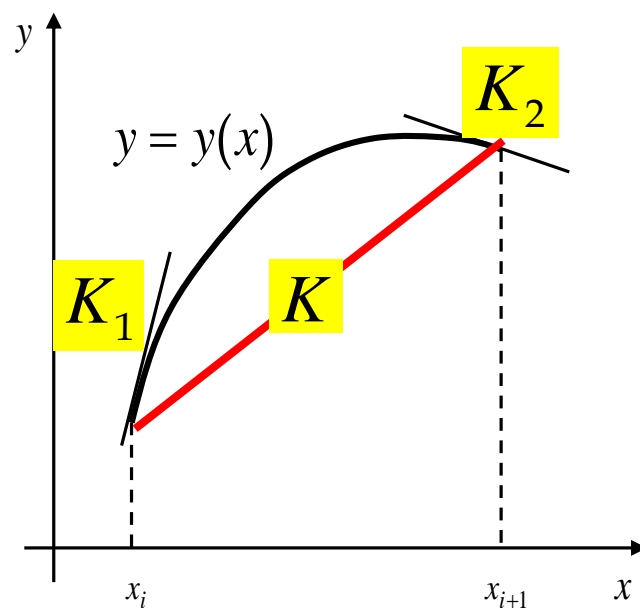
$$K_1 = hf(x_i, y_i) \quad K_2 = hf(x_i + a_2 h, y_i + b_{21} K_1)$$

$$y(x_{i+1}) = y(x_i) + K = y(x_i) + (c_1 K_1 + c_2 K_2)$$

$$K_1 = hf(x_i, y_i) \quad K_2 = hf(x_i + a_2 h, y_i + b_{21} K_1)$$

注意到， $a_2 = b_{21} = 1$ ， $c_1 = c_2 = \frac{1}{2}$ 就是二阶龙格 - 库塔公式，也就是改进的欧拉法。

$$\begin{cases} y_{i+1} &= y_i + \frac{1}{2}(K_1 + K_2) \\ K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + h, y_i + K_1) \end{cases}$$



因此，凡满足条件式有一簇形如上式的计算格式，这些格式统称为二阶龙格—库塔格式。因此改进的欧拉格式是众多的二阶龙格—库塔法中的一种特殊格式。

若取 $a_2 = b_{21} = \frac{1}{2}$, $c_1 = 0$, $c_2 = 1$, 就是另一种形式的二阶龙格 - 库塔公式。

$$\begin{cases} y_{i+1} &= y_i + K_2 \\ K_1 &= hf(x_i, y_i) \\ K_2 &= hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1) \end{cases}$$

此计算公式称为变形的二阶龙格—库塔法。式中 $x_i + \frac{1}{2}h$ 为区间 $[x_i, x_{i+1}]$ 的中点。也称中点公式。

Q: 为获得更高的精度，应该如何进一步推广？

二级 $R-K$ 方法是显式单步式,每前进一步需要**计算两个函数值**。由上面的讨论可知,适当选择四个参数 c_1, c_2, a_2, b_{21} ,可使每步计算两次函数值的二阶 $R-K$ 方法达到二阶精度。

能否在计算函数值次数不变的情况下,通过选择不同的参数值,使得二阶 $R-K$ 方法的精度再提高呢?

答案是否定的! 无论四个参数怎样选择,都不能使公式的局部截断误差提高到三阶。

这说明每一步计算两个函数值的二阶 $R-K$ 方法最高阶为二阶。若要获得更高阶得数值方法,就必须增加计算函数值的次数。

分析改进的欧拉法（休恩法）

即Euler方法

$$\begin{cases} y_{n+1} = y_n + k_1 \\ k_1 = hf(x_n, y_n) \end{cases}$$

每步计算 f 的值一次，其截断误差为 $O(h^2)$ 。

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + h, y_n + k_1) \end{cases}$$

每步计算 f 的值两次，其截断误差为 $O(h^3)$ 。

上述两组公式在形式上共同点：

都是用 $f(x, y)$ 在某些点上值的线性组合得出 $y(x_{i+1})$ 的近似值 y_{i+1} ，且增加计算的次数 $f(x, y)$ 的次数，可提高截断误差的阶。

如欧拉法：每步计算一次 $f(x, y)$ 的值，为一阶方法。

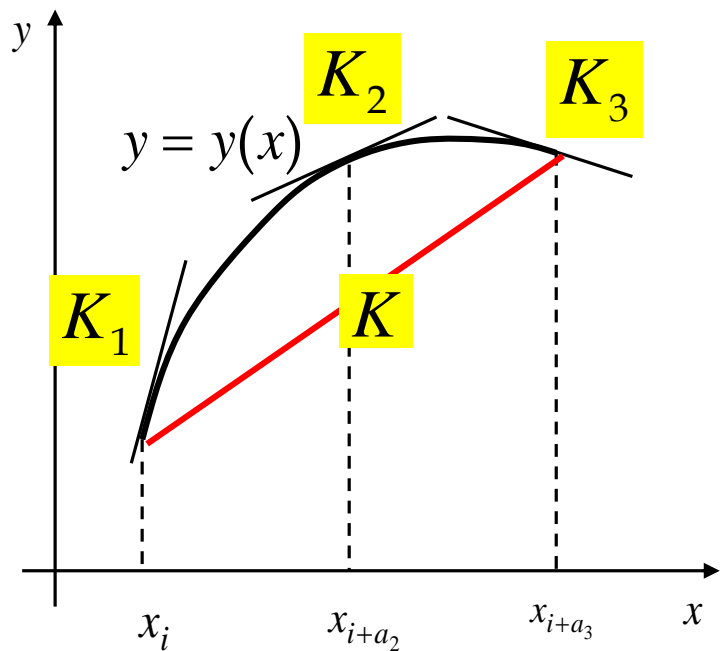
改进欧拉法需计算两次 $f(x, y)$ 的值，为二阶方法。

一般龙格-库塔方法的形式为

$$\left\{ \begin{array}{l} y_{i+1} = y_i + c_1 K_1 + c_2 K_2 + \cdots + c_p K_p \\ K_1 = hf(x_i, y_i) \\ K_2 = hf(x_i + a_2 h, y_i + b_{21} K_1) \\ \bullet \dots \bullet \bullet \dots \bullet \bullet \dots \bullet \bullet \dots \bullet \bullet \dots \bullet \bullet \dots \\ K_p = hf(x_i + a_p h, y_i + b_{p1} K_1 + \cdots + b_{p,p-1} K_{p-1}) \end{array} \right.$$

称为 P 阶龙格-库塔方法。

其中 a_i, b_{ij}, c_i 为待定参数，要求上式 y_{i+1} 在点 (x_i, y_i) 处作 *Taylor* 展开，通过相同项的系数确定参数。



9.4 龙格-库塔 (Runge-Kutta) 法

$R-K$ 方法不是通过求导数的方法构造近似公式,而是通过计算不同点上的函数值,并对这些函数值作线性组合,构造近似公式,再把近似公式与解的泰勒展开式进行比较,使前面的若干项相同,从而使近似公式达到一定的阶数.

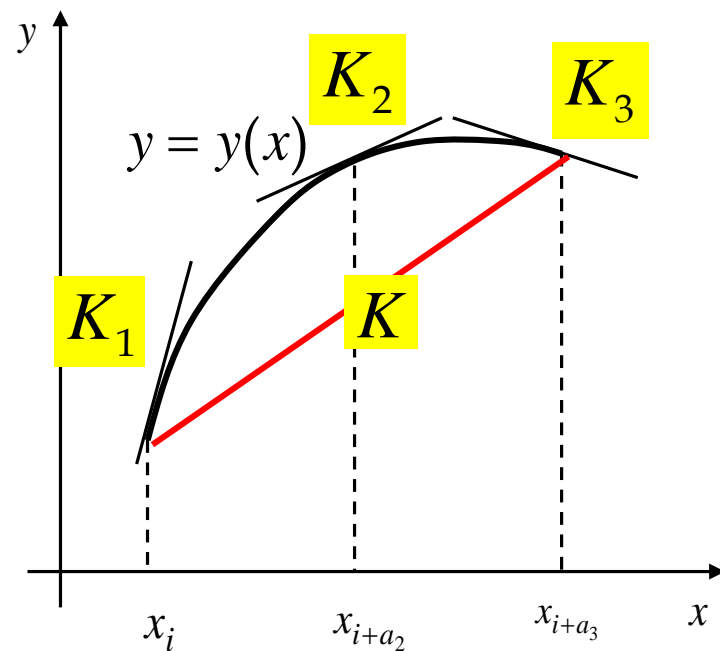
既避免求高阶导数,又提高了精度的阶数。或者说,在 $[x_i, x_{i+1}]$ 这一步内多计算几个点的斜率值,然后将其进行加权平均作为平均斜率,则可构造出更高精度的计算格式,这就是龙格—库塔 ($Runge-Kutta$) 法的基本思想。

R-K方法不是直接使用Taylor级数,而是利用它的思想

三阶龙格—库塔法

为进一步提高精度，在区间 $[x_i, x_{i+1}]$ 上除两点 x_i 和 $x_{i+a_2} = x_i + a_2h$ ，以外，再增加一点 $x_{i+a_3} = x_i + a_3h$ ，用这三点处的斜率值 K_1 、 K_2 和 K_3 的加权平均得出平均斜率 K^* 的近似值 K ，这时计算格式具有形式：

$$\begin{cases} y_{i+1} = y_i + c_1K_1 + c_2K_2 + c_3K_3 \\ K_1 = hf(x_i, y_i) \\ K_2 = hf(x_i + a_2h, y_i + b_{21}K_1) \\ K_3 = hf(x_i + a_3h, y_i + b_{31}K_1 + b_{32}K_2) \end{cases}$$



同理推导二阶公式，将 $y(x_{i+1})$ 和 y_{i+1} 在 $x=x_i$ 处进行Taylor展开，使局部截断误差达到 $O(h^4)$ ，使对应项的系数相等，得到系数方程组：

$$\left\{ \begin{array}{ll} c_1 + c_2 + c_3 = 1 & \\ a_2 c_2 + a_3 c_3 = \frac{1}{2}, & b_{21} c_2 + (b_{31} + b_{32}) c_3 = \frac{1}{2} \\ a_2^2 c_2 + a_3^2 c_3 = \frac{1}{3}, & a_2 b_{21} c_2 + a_3 c_3 (b_{31} + b_{32}) = \frac{1}{3} \\ b_{21}^2 c_2 + (b_{31} + b_{32})^2 c_3 = \frac{1}{3} & \\ a_2 b_{32} c_2 = \frac{1}{6}, & b_{21} b_{32} c_3 = \frac{1}{6} \end{array} \right.$$

参数的选择不唯一，从而构成一类不同的三阶 $R-K$ 公式，
下面给出一种常用的三阶 $R-K$ 公式，形似 $Simpson$ 公式：

$$\left\{ \begin{array}{l} y_{i+1} = y_i + \frac{1}{6} (K_1 + 4K_2 + K_3) \\ K_1 = hf(x_i, y_i) \\ K_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}K_1\right) \\ K_3 = hf(x_i + h, y_i - K_1 + 2K_2) \end{array} \right.$$

四阶龙格—库塔法 (经典)

如果需要再提高精度，用类似上述的处理方法，只需在区间 $[x_i, x_{i+1}]$ 上用四个点处的斜率加权平均作为平均斜率 K^* 的近似值，构成一系列四阶龙格—库塔公式。具有四阶精度，即局部截断误差是 $O(h^5)$ 。

推导过程与前面类似，由于过程复杂，这里从略，只介绍最常用的一种**四阶经典龙格—库塔公式**。

设

$$y_{i+1}=y_i+c_1K_1+c_2K_2+c_3K_3+c_4K_4$$

这里 K_1 、 K_2 、 K_3 、 K_4 为四个不同点上的函数值，分别设其为

$$\begin{cases} K_1=hf(x_i, y_i) \\ K_2=hf(x_i+a_2h, y_i+b_{21}K_1) \\ K_3=hf(x_i+a_3h, y_i+b_{31}K_1+b_{32}K_2) \\ K_4=hf(x_i+a_4h, y_i+b_{41}K_1+b_{42}K_2+b_{43}K_3) \end{cases}$$

其中 c_1 、 c_2 、 c_3 、 c_4 、 a_2 、 a_3 、 a_4 、 b_{21} 、 b_{31} 、 b_{32} 、 b_{41} 、 b_{42} 、 b_{43} 均为待定系数。

类似于前面的讨论，把 K_2 、 K_3 、 K_4 分别在 x_i 点展成 h 的幂级数，代入线性组合式中，将得到的公式与 $y(x_{i+1})$ 在 x_i 点上的泰勒展开式比较，使其两式右端直到 h_4 的系数相等，经过较复杂的解方程过程便可得到关于 c_i ， a_i ， b_{ij} 的一组特解

$$a_2=a_3=b_{21}=b_{32}=1/2$$

$$b_{31}=b_{41}=b_{42}=0$$

$$a_4=b_{43}=1$$

$$c_1=c_4=1/6$$

$$c_2=c_3=1/3$$

➤ 四阶(经典)*Runge-Kutta*方法

$$\left\{ \begin{array}{l} y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = hf(x_i, y_i) \\ K_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}K_1\right) \\ K_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{1}{2}K_2\right) \\ K_4 = hf(x_i + h, y_i + K_3) \end{array} \right.$$

小 结

龙格—库塔方法的推导基于*Taylor*展开方法，因而它要求所求的解具有较好的光滑性。
如果解的光滑性差，那么，使用四阶龙格—库塔方法求得的数值解，其精度可能反而不如改进的欧拉方法。在实际计算时，应当针对问题的具体特点选择合适的算法。对于光滑性不太好的解，最好采用低阶算法而将步长 h 取小。

前面已经看到,二阶、四阶 $R-K$ 方法可分别达到最高阶数2阶、3阶和4阶,但是 N 阶 $R-K$ 方法的最高阶却不一定是 N 阶。 $R-K$ 方法的级数表示公式中计算函数值 f 的次数。 $Butcher$ 于1965年给出了 $R-K$ 方法计算函数值 f 的次数与可达到的最高精度阶数之间的关系表,如下:

每步须算 K_i 的个数	2	3	4	5	6	7	$n \geq 8$
可达到的最高精度	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^4)$	$O(h^5)$	$O(h^6)$	$O(h^{n-2})$

由表可见,四级以下 $R-K$ 的方法其最高阶数与计算 f 的次数一致,对 m 阶 $R-K$ 公式,当 $m > 4$,虽然计算 f 的次数增加,但是方法阶数不一定增加。

因此,四阶 $R-K$ 公式是应用最为广泛的公式。

例9.3 使用高阶 $R-K$ 方法计算初值问题

$$\begin{cases} y' = y^2 & 0 \leq x \leq 0.5 \\ y(0) = 1 \end{cases} \quad \text{取 } h = 0.1.$$

解: (1) 使用三阶 $R-K$ 方法

$$i = 0 \text{ 时} \quad K_1 = hy_0^2 = 0.1$$

$$K_2 = h(y_0 + \frac{1}{2}K_1)^2 = 0.1103$$

$$K_3 = h(y_0 - K_1 + 2K_2)^2 = 0.1256$$

$$y_1 = y_0 + \frac{1}{6}(K_1 + 4K_2 + K_3) = 1.1111$$

其余结果如下:

i	x_i	k_1	k_2	k_3	y_i
1.0000	0.1000	0.1000	0.1103	0.1256	1.1111
2.0000	0.2000	0.1235	0.1376	0.1595	1.2499
3.0000	0.3000	0.1562	0.1764	0.2092	1.4284
4.0000	0.4000	0.2040	0.2342	0.2866	1.6664
5.0000	0.5000	0.2777	0.3259	0.4163	1.9993

(2) 如果使用四阶R-K方法

$$i = 0 \text{ 时} \quad K_1 = hy_0^2 = 0.1$$

$$K_2 = h(y_0 + \frac{1}{2} K_1)^2 = 0.1103$$

$$K_3 = h(y_0 + \frac{1}{2} K_2)^2 = 0.1113$$

$$K_4 = h(y_0 + K_3)^2 = 0.1235$$

$$y_1 = y_0 + \frac{0.1}{6}(K_1 + 2K_2 + 2K_3 + K_4) = 1.1111$$

其余结果如下:

i	x_i	k_1	k_2	k_3	k_4	y_i
1.0000	0.1000	0.1000	0.1103	0.1113	0.1235	1.1111
2.0000	0.2000	0.1235	0.1376	0.1392	0.1563	1.2500
3.0000	0.3000	0.1562	0.1764	0.1791	0.2042	1.4286
4.0000	0.4000	0.2040	0.2342	0.2389	0.2781	1.6667
5.0000	0.5000	0.2777	0.3259	0.3348	0.4006	2.0000

例9.4 用标准四阶龙格-库塔方法求解初值问题

$$\begin{cases} \frac{dy}{dx} = y - \frac{2x}{y}, & 0 \leq x \leq 0.6, \quad h = 0.2 \\ y(0) = 1, \end{cases}$$

解 计算过程和结果如下表所示

$$y_{n+1} = y_n + \bar{k}$$

$$\bar{k} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

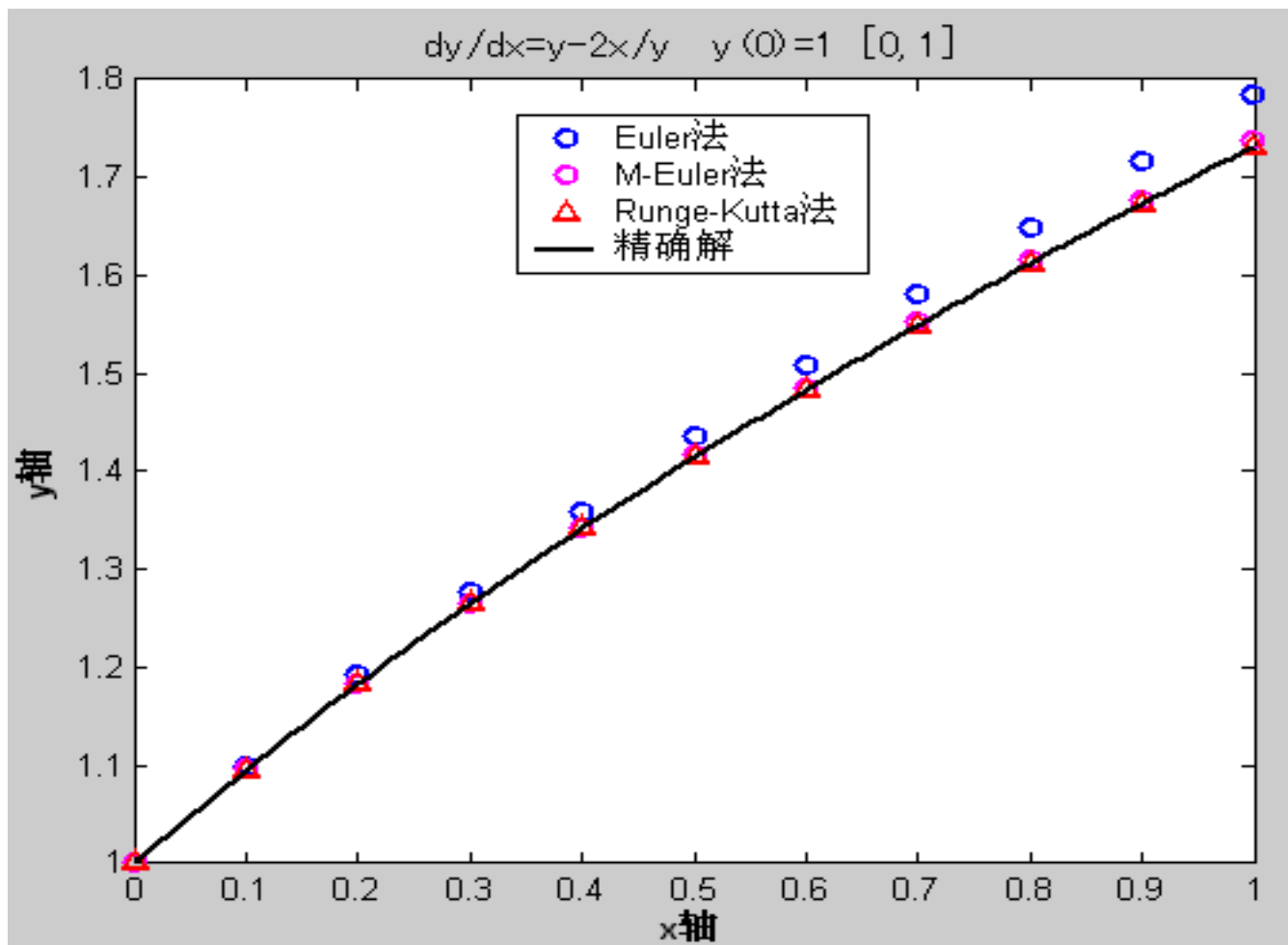
x_n	y_n	$\frac{k_i}{2} = 0.1 \left(y_n - \frac{2x_n}{y_n} \right)$	\overline{k}
0	1.000000	0.1000000	0.1832292
0.1	1.100000	0.0918182	
0.1	1.091818	0.0908637	
0.2	1.181727	0.0843239	
0.2	1.183229	0.0849171	0.1584376
0.3	1.267746	0.0794465	
0.3	1.262676	0.0787495	
0.4	1.340728	0.0744037	
0.4	1.341667	0.0745394	0.1416245
0.5	1.416026	0.0710094	
0.5	1.412676	0.0708400	
0.6	1.482627	0.0673253	
0.6	1.483281		

因此有 $y(0) = y_0 = 1$, $y(0.2) \approx y_1 = 1.183229$,

$y(0.4) \approx y_2 = 1.341667$, $y(0.6) \approx y_3 = 1.483281$.

对该例, 用几种不同的一步法计算的结果如下表

x_n	欧拉法 ($h=0.1$)	预估-校正法 ($h=0.1$)	四阶 $R-K$ 法 ($h=0.2$)	准确值 $y = \sqrt{1+2x}$
0	1	1	1	1
0.1	1.1	1.095909		1.095445
0.2	1.191818	1.184097	1.183229	1.183216
0.3	1.277438	1.266201		1.264911
0.4	1.358213	1.343360	1.341667	1.341641
0.5	1.435133	1.416402		1.414214
0.6	1.508966	1.485956	1.483281	1.483240
0.7	1.580388	1.552514		1.549193
0.8	1.649738	1.616475	1.612514	1.612452



由上表可见，虽然四阶龙格-库塔方法每步要计算四次 f 的值，但以 $h=0.2$ 为步长的计算结果就有 5 位有效数字，而欧拉法与预估-校正方法以 $h=0.1$ 为步长的计算结果才具有 2 位与 3 位有效数字。如果步长 h 也取 0.2，则结果的精度会更低。

程序 9.4(4 阶龙格 – 库塔方法) 用公式

$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

计算 $[a, b]$ 上的初值问题 $y' = f(t, y), y(a) = y_0$ 的近似解。

```
function R=rk4(f,a,b,ya,M)
```

```
%Input - f is the function
```

```
% - a and b are the left and right endpoints
```

```
% - ya is the initial condition y(a)
```

```
% - M is the number of steps
```

```
%Output - R = [T' Y'] where T is the vector of  
abscissas
```

```
% and Y is the vector of ordinates
```

```
h=(b-a)/M;
```

```
T=zeros(1,M+1);
```

```
Y=zeros(1,M+1);
```

```
T=a:h:b;
```

```
Y(1)=ya;
```

```
for j=1:M
```

```
    k1=h*f(T(j),Y(j));
```

```
    k2=h*f(T(j)+h/2,Y(j)+k1/2);
```

```
    k3=h*f(T(j)+h/2,Y(j)+k2/2);
```

```
    k4=h*f(T(j)+h,Y(j)+k3);
```

```
    Y(j+1)=Y(j)+(k1+2*k2+2*k3+k4)/6;
```

```
end
```

```
R=[T' Y'];
```


9.5 预报-矫正方法

欧拉方法、休恩方法、泰勒方法以及龙格-库塔方法都称为单步长方法,因为它们只利用前一个点的信息来计算下一个点,即计算 (t_1, y_1) 时只使用了初始点 (t_0, y_0) 。一般地,只有 y_k 用来计算 y_{k+1} 。当计算出若干个点之后,就可以利用几个已计算出的点来计算下一个点。以亚当斯-巴什福斯4步法的推导为例,计算 y_{k+1} 需要 $y_{k-3}, y_{k-2}, y_{k-1}$ 和 y_k 。该方法不是自启动的,要生成点 $\{(t_k, y_k): k \geq 4\}$,必须先给出其4个初始点 $(t_0, y_0), (t_1, y_1), (t_2, y_2)$ 和 (t_3, y_3) (可用前面各节中的方法完成)。

多步法的一个优点是,可以确定它的局部截断误差(local truncation error, 简称 L.T.E.), 并可以包含一个校正项,用于在每一步计算中改善解的精确度。该方法还可以确定步长是否小到能得到 y_{k+1} 的精确值,同时又大到能够免除不必要的和费时的计算。使用预报子和校正子的组合在每一步只需要进行两次函数 $f(t, y)$ 求值。

本节只介绍一种多步法---阿达姆斯 (Adams) 方法

阿达姆斯 (Adams) 方法

► 基本思想:

对微分方程 $\frac{dy}{dx} = f(x, y)$ 两边在区间 $[x_n, x_{n+1}]$ 上积分:

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y) dx$$

在积分区间 $[x_n, x_{n+1}]$ 上 $f(x, y)$ 用其插值多项式近似代替。

► Adams 四步显式公式:

选取下列4个等距节点作为插值节点:

$$x_k = x_0 + kh; \quad k = n-3, n-2, n-1, n$$

构造 $f(x, y)$ 的3次Lagrange插值多项式:

$$f(x, y) \approx L_3(x) = \sum_{k=0}^3 \left(\prod_{\substack{m=0 \\ m \neq k}}^3 \frac{x - x_{n-m}}{x_{n-k} - x_{n-m}} \right) f(x_{n-k}, y_{n-k})$$

$$y_{n+1} = y_n + \sum_{k=0}^3 f(x_{n-k}, y_{n-k}) \int_{x_n}^{x_{n+1}} \prod_{\substack{m=0 \\ m \neq k}}^3 \frac{x - x_{n-m}}{x_{n-k} - x_{n-m}} dx$$

$$y_{n+1} = y_n + \frac{h}{24} [55f(x_n, y_n) - 59f(x_{n-1}, y_{n-1}) \\ + 37f(x_{n-2}, y_{n-2}) - 9f(x_{n-3}, y_{n-3})]$$

称之为Adams四步显式公式（或Adams外插值公式）

► Adams 显式公式的局部截断误差：

4阶方法

由Lagrange插值余项知

$$f(x, y) - L_3(x) = \left[\frac{d^4}{dx^4} f(x, y(x)) \right]_{x=\xi} \frac{\omega(x)}{4!}$$

其中 $\omega(x) = (x - x_n)(x - x_{n-1})(x - x_{n-2})(x - x_{n-3})$

$$y(x_{n+1}) - y_{n+1} = \int_{x_n}^{x_{n+1}} [f(x, y) - L_3(x)] dx$$

$$= \int_{x_n}^{x_{n+1}} \left[\frac{d^4}{dx^4} f(x, y) \right]_{x=\xi} \frac{\omega(x)}{4!} dx \quad (\text{第二积分中值定理})$$

$$= \frac{1}{4!} \frac{d^5 y(x)}{dx^5} \Big|_{x=\eta} \int_{x_n}^{x_{n+1}} \omega(x) dx = \frac{251}{720} h^5 y^{(5)}(\eta) = O(h^5)$$

► **Adams** 三步隐式公式（或 **Adams** 内插值公式）：

选取下列4个等距节点作为插值节点：

$$x_k = x_0 + kh; \quad k = n-2, n-1, n, n+1$$

构造 $f(x, y)$ 的3次 **Lagrange** 插值多项式：

$$f(x, y) \approx L_3(x) = \sum_{k=-1}^2 \left(\prod_{\substack{m=-1 \\ m \neq k}}^2 \frac{x - x_{n-m}}{x_{n-k} - x_{n-m}} \right) f(x_{n-k}, y_{n-k})$$

$$y_{n+1} = y_n + \sum_{k=-1}^2 f(x_{n-k}, y_{n-k}) \int_{x_n}^{x_{n+1}} \prod_{\substack{m=-1 \\ m \neq k}}^2 \frac{x - x_{n-m}}{x_{n-k} - x_{n-m}} dx$$

$$y_{n+1} = y_n + \frac{h}{24} [9f(x_{n+1}, y_{n+1}) + 19f(x_n, y_n) \\ - 5f(x_{n-1}, y_{n-1}) + f(x_{n-2}, y_{n-2})]$$

Adams 三步隐式公式的局部截断误差：

$$y(x_{n+1}) - y_{n+1} = -\frac{19}{720}h^5 y^{(5)}(\eta) = O(h^5)$$

► Adams 预报-校正公式：

初值由4阶R-K方法计算

$$\left\{ \begin{array}{l} y_{n+1}^{(0)} = y_n + \frac{h}{24} [55f(x_n, y_n) - 59f(x_{n-1}, y_{n-1}) \\ \quad + 37f(x_{n-2}, y_{n-2}) - 9f(x_{n-3}, y_{n-3})] \quad (\text{显式, 预报}) \\ y_{n+1} = y_n + \frac{h}{24} [9f(x_{n+1}, y_{n+1}^{(0)}) + 19f(x_n, y_n) \\ \quad - 5f(x_{n-1}, y_{n-1}) + f(x_{n-2}, y_{n-2})] \quad (\text{隐式, 校正}) \end{array} \right.$$

例9.5: 用Adams预报-校正公式

求解下列初值问题 $h=0.1$ 。

$$\begin{cases} \frac{dy}{dx} = y - \frac{2x}{y} \\ y(0) = 1 \end{cases} \quad x \in [0,1]$$

解: Adams预报-校正公式:

$$\begin{cases} y_{n+1}^{(0)} = y_n + \frac{h}{24} \left[55 \left(y_n - \frac{2x_n}{y_n} \right) - 59 \left(y_{n-1} - \frac{2x_{n-1}}{y_{n-1}} \right) \right. \\ \quad \left. + 37 \left(y_{n-2} - \frac{2x_{n-2}}{y_{n-2}} \right) - 9 \left(y_{n-3} - \frac{2x_{n-3}}{y_{n-3}} \right) \right] \\ y_{n+1} = y_n + \frac{h}{24} \left[9 \left(y_{n+1}^{(0)} - \frac{2x_{n+1}}{y_{n+1}^{(0)}} \right) + 19 \left(y_n - \frac{2x_n}{y_n} \right) \right. \\ \quad \left. - 5 \left(y_{n-1} - \frac{2x_{n-1}}{y_{n-1}} \right) + \left(y_{n-2} - \frac{2x_{n-2}}{y_{n-2}} \right) \right] \end{cases}$$

x_i	R-K方法	Adams预-校法	精确解
0	1		1.00000000000
0.1	1.095446		1.0954451153
0.2	1.183217		1.1832159566
0.3	1.264912		1.2649110640
0.4		1.3416413571	1.3416407864
0.5		1.4142138334	1.4142135623
0.6		1.4832398242	1.4832396974
0.7		1.5491933804	1.5491933384
0.8		1.6124515364	1.6124515496
0.9		1.6733199993	1.6733200530
1.0		1.7320507198	1.7320508075

9.6 微分方程组

本节是对微分方程组的入门介绍。

为说明概念，考虑如下初值问题

$$\begin{aligned} \frac{dx}{dt} &= f(t, x, y) \\ \frac{dy}{dt} &= g(t, x, y), \end{aligned} \quad \begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases} \quad (9.3)$$

式中的解是一对可微函数 $x(t)$ 和 $y(t)$ ，当将 $t, x(t)$ 和与 $y(t)$ 代入 $f(t, x, y)$ 和 $g(t, x, y)$ 时，结果分别等于导数函数 $x'(t)$ 和 $y'(t)$ ，即

$$\begin{aligned} x'(t) &= f(t, x(t), y(t)) \\ y'(t) &= g(t, x(t), y(t)) \end{aligned} \quad \begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases}$$

例9.6 考虑微分方程组

$$\begin{cases} \frac{dx}{dt} = x + 2y \\ \frac{dy}{dt} = 3x + 2y, \end{cases} \quad \begin{cases} x(0) = 6 \\ y(0) = 4 \end{cases} \quad (9.4)$$

该初值问题的解为

$$\begin{aligned} x(t) &= 4e^{4t} + 2e^{-t} \\ y(t) &= 6e^{4t} - 2e^{-t} \end{aligned} \quad (9.5)$$

直接将 $x(t)$ 和 $y(t)$ 代入式 (9.4) 右端计算式 (9.5) 的导数函数，并将其代入式 (9.4) 右端，得

$$\begin{aligned} 16e^{4t} - 2e^{-t} &= (4e^{4t} + 2e^{-t}) + 2(6e^{4t} - 2e^{-t}) \\ 24e^{4t} + 2e^{-t} &= 3(4e^{4t} + 2e^{-t}) + 2(6e^{4t} - 2e^{-t}) \end{aligned}$$

考虑微分方程组

$$dx = f(t, x, y)dt, \quad dy = g(t, x, y)dt \quad (9.6)$$

可得出式 (9.3) 在区间[a, b]上的数值解。

易得求解该方程组的Eular方法，将差分式 $dt = t_{k+1} - t_k$,
 $dx = x_{k+1} - x_k$, $dy = y_{k+1} - y_k$ 代入式 (9.6) , 得

$$\begin{aligned} x_{k+1} - x_k &\approx f(t_k, x_k, y_k)(t_{k+1} - t_k) \\ y_{k+1} - y_k &\approx g(t_k, x_k, y_k)(t_{k+1} - t_k) \end{aligned} \quad (9.7)$$

将区间[a,b]分为M个子区间，宽度为 $h=(b-a)/M$, 网格点为
 $t_{k+1} = t_k + h$. 把它们代入 (9.7) 中可得Eular方法的递归公式：

$$\begin{aligned} t_{k+1} &= t_k + h, \\ x_{k+1} &= x_k + hf(t_k, x_k, y_k), \\ y_{k+1} &= y_k + hg(t_k, x_k, y_k), \quad k = 0, 1, \dots, M-1 \end{aligned}$$

要得到合理的精确度,应该使用更高阶的方法。例如,4阶龙格-库塔公式为

$$x_{k+1} = x_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4)$$

$$y_{k+1} = y_k + \frac{h}{6}(g_1 + 2g_2 + 2g_3 + g_4)$$

其中

$$f_1 = f(t_k, x_k, y_k),$$

$$g_1 = g(t_k, x_k, y_k)$$

$$f_2 = f\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_1, y_k + \frac{h}{2}g_1\right), \quad g_2 = g\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_1, y_k + \frac{h}{2}g_1\right)$$

$$f_3 = f\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_2, y_k + \frac{h}{2}g_2\right), \quad g_3 = g\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_2, y_k + \frac{h}{2}g_2\right)$$

$$f_4 = f(t_k + h, x_k + hf_3, y_k + hg_3), \quad g_4 = g(t_k + h, x_k + hf_3, y_k + hg_3)$$

例9.7 用4阶龙格-库塔方法计算区间[0, 0.2]上例9.6中微分方程组的数值解，采用10个子区间，步长为0.02.

$$\begin{cases} \frac{dx}{dt} = x + 2y \\ \frac{dy}{dt} = 3x + 2y, \end{cases} \quad \begin{cases} x(0) = 6 \\ y(0) = 4 \end{cases}$$

解:第一个点为 $t_1 = 0.02$, 计算 x_1 和 y_1 所需的中间计算为

$$f_1 = f(0.00, 6.0, 4.0) = 14.0$$

$$g_1 = g(0.00, 6.0, 4.0) = 26.0$$

$$x_0 + \frac{h}{2} f_1 = 6.14$$

$$y_0 + \frac{h}{2} g_1 = 4.26$$

$$f_2 = f(0.01, 6.14, 4.26) = 14.66$$

$$g_2 = g(0.01, 6.14, 4.26) = 26.94$$

$$x_0 + \frac{h}{2} f_2 = 6.1466$$

$$y_0 + \frac{h}{2} g_2 = 4.2694$$

$$f_3 = f(0.01, 6.1466, 4.2694) = 14.6854$$

$$g_3 = f(0.01, 6.1466, 4.2694) = 26.9786$$

$$x_0 + h f_3 = 6.293708 \quad y_0 + h g_3 = 4.539572$$

$$f_4 = f(0.02, 6.293708, 4.539572) = 15.372852$$

$$g_4 = f(0.02, 6.293708, 4.539572) = 27.960268$$

用它们得到最终的计算结果:

$$x_1 = 6 + \frac{0.02}{6} (14.0 + 2(14.66) + 2(14.6854) + 15.372852) = 6.29354551$$

$$y_1 = 4 + \frac{0.02}{6} (26.0 + 2(26.94) + 2(26.9786) + 27.960268) = 4.53932490$$

表 9.13 给出了所有的计算结果。

表 9.13 $x'(t) = x + 2y, y'(t) = 3x + 2y, x(0) = 6, y(0) = 4$ 的龙格 - 库塔解

k	t_k	x_k	y_k
0	0.00	6.00000000	4.00000000
1	0.02	6.29354551	4.53932490
2	0.04	6.61562213	5.11948599
3	0.06	6.96852528	5.74396525
4	0.08	7.35474319	6.41653305
5	0.10	7.77697287	7.14127221
6	0.12	8.23813750	7.92260406
7	0.14	8.74140523	8.76531667
8	0.16	9.29020955	9.67459538
9	0.18	9.88827138	10.6560560
10	0.20	10.5396230	11.7157807

高阶常微分方程的数值解法简介

高阶微分方程包含高阶导数 $x''(t)$ 和 $x'''(t)$ 等。它们出现在物理和工程问题的数学模型中,例如

$$mx''(t) + cx'(t) + kx(t) = g(t)$$

表示一个弹性系数为 k 的弹簧带着质量为 m 的物体的力学系统。设阻尼正比于速度,函数 $g(t)$ 为外力。通常的情况是,已知某一时刻 t_0 的位置 $x(t_0)$ 和速度 $x'(t_0)$ 。

通过求解 2 阶导数,可将问题描述为 2 阶初值问题

$$x''(t) = f(t, x(t), x'(t)), \quad x(t_0) = x_0, \quad x'(t_0) = y_0$$

使用变量替换

$$x'(t) = y(t)$$

可将 2 阶微分方程重新表示为两个 1 阶问题的方程组, 这样就有 $x''(t) = y'(t)$, 而高阶常微分方程变为方程组

$$\begin{aligned} \frac{dx}{dt} &= y \\ \frac{dy}{dt} &= f(t, x, y), \end{aligned} \quad \begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases}$$

可以用龙格-库塔之类的方法予以求解。

例9.8 考虑2阶初值问题

$$x''(t) + 4x'(t) + 5x(t) = 0, x(0) = 3, x'(0) = -5$$

- (a) 写出等价的两个1阶问题组成的方程组。
- (b) 用龙格-库塔方法求解区间 $[0, 5]$ 上重新描述的方程, 使用 $M = 50$ 个宽度为 $h = 0.1$ 的子区间。
- (c) 比较数值解与精确解:

$$x(t) = 3e^{-2t} \cos(t) + e^{-2t} \sin(t)$$

解:(a) 微分方程的形式为

$$x''(t) = f(t, x(t), x'(t)) = -4x'(t) - 5x(t)$$

(b) 用变量替换式(10), 得到问题的新的方程描述:

$$\begin{aligned} \frac{dx}{dt} &= y \\ \frac{dy}{dt} &= -5x - 4y, \end{aligned} \quad \begin{cases} x(0) = 3 \\ y(0) = -5 \end{cases}$$

(c) 表 9.14 给出了数值计算的解。表中没有列出值 $\{y_k\}$ (因为它是附加的), 而是列出了真解 $\{x(t_k)\}$ 以供比较。 ■

表 9.14 $x''(t) + 4x'(t) + 5x(t) = 0$ 的龙格-库塔解, 初值条件为 $x(0) = 3$ 和 $x'(0) = -5$

k	t_k	x_k	$x(t_k)$
0	0.0	3.00000000	3.00000000
1	0.1	2.52564583	2.52565822
2	0.2	2.10402783	2.10404686
3	0.3	1.73506269	1.73508427
4	0.4	1.41653369	1.41655509
5	0.5	1.14488509	1.14490455
10	1.0	0.33324302	0.33324661
20	2.0	-0.00620684	-0.00621162
30	3.0	-0.00701079	-0.00701204
40	4.0	-0.00091163	-0.00091170
48	4.8	-0.00004972	-0.00004969
49	4.9	-0.00002348	-0.00002345
50	5.0	-0.00000493	-0.00000490

作业 9.4

算法与程序

13. 捕食者 – 被捕食者模型。非线性微分方程的一个例子是捕食者 – 被捕食者模型。设 $x(t)$ 和 $y(t)$ 分别表示兔子和狐狸在时刻 t 的数量, 捕食者 – 被捕食者模型表明, $x(t)$ 和 $y(t)$ 满足

$$x'(t) = Ax(t) - Bx(t)y(t)$$

$$y'(t) = Cx(t)y(t) - Dy(t)$$

一个典型的计算机模拟可使用系数

$$A = 2, \quad B = 0.02, \quad C = 0.0002, \quad D = 0.8$$

如果

(a) $x(0) = 3000$ 只兔子, $y(0) = 120$ 只狐狸,

(b) $x(0) = 5000$ 只兔子, $y(0) = 100$ 只狐狸,

在区间 $[0, 5]$ 上用 $M = 50$ 步和 $h = 0.2$ 求解。

9.7 边值问题

9.7.1 基本概念

9.7.2 打靶法

9.7.3 有限差分法

9.7.1 常微分方程边值问题的基本概念

考察二阶微分方程

$$y''=f(x, y, y'), \quad x \in [a, b], \quad -\infty < y < +\infty$$

当 $f(x, y, y')$ 关于 y, y' 为线性时, 即

$$f(x, y, y')=p(x)y'+q(x)y+r(x)$$

方程为线性微分方程

$$y''-p(x)y'-q(x)y=r(x)$$

其边值条件有下面三类:

第一边值条件 $y(a)=\alpha, y(b)=\beta, \alpha=0$ 或 $\beta=0$ 为齐次, 否则为非齐次。

第二边值条件 $y'(a)=\alpha, y'(b)=\beta, \alpha=0$ 或 $\beta=0$ 为齐次, 否则为非齐次。

第三边值条件 $y'(a)-\alpha_0 y(a)=\alpha_1, y'(b)+\beta_0 y(b)=\beta_1, \alpha_0, \beta_0 \geq 0,$

$\alpha_0+\beta_0>0 \alpha_0=0$ 或 $\beta_0=0$ 为齐次, 否则为非齐次。

在进行任何数值方法之前,必须检查保证方程(1)的解存在的条件,否则可能产生无意义的结果。下面的定理给出了一般的条件。

定理 9.8(边值问题) 设 $f(t, x, y)$ 在区间 $R = \{(t, x, y): a \leq t \leq b, -\infty < x < \infty, -\infty < y < \infty\}$ 上连续,且 $\partial f / \partial x = f_x(t, x, y)$ 和 $\partial f / \partial y = f_y(t, x, y)$ 在 R 上也连续,如果存在常数 $M > 0$,使 f_x 和 f_y 对于所有 $(t, x, y) \in R$ 满足

$$f_x(t, x, y) > 0 \quad (3)$$

$$|f_y(t, x, y)| \leq M \quad (4)$$

则边值问题

$$x'' = f(t, x, x'), \quad x(a) = \alpha, \quad x(b) = \beta \quad (5)$$

在 $a \leq t \leq b$ 上有惟一解 $x = x(t)$ 。

其中,用记号 $y = x'(t)$ 来表示函数 $f(t, x, x')$ 中的第三个变量。最后值得一提的是,线性微分方程的特殊情况。

推论 9.1(线性边值问题) 设定理 9.8 中的 f 具有 $f(t, x, y) = p(t)y + q(t)x + r(t)$ 的形式,且 f 及其偏导数 $\partial f / \partial x = q(t)$ 和 $\partial f / \partial y = p(t)$ 在 R 上连续。如果存在常数 $M > 0$,使得 $p(t)$ 和 $q(t)$ 满足

$$q(t) > 0 \quad \text{对于所有 } t \in [a, b] \quad (6)$$

$$|p(t)| \leq M = \max_{a \leq t \leq b} \{|p(t)|\} \quad (7)$$

则线性边值问题

$$x'' = p(t)x'(t) + q(t)x(t) + r(t), \quad x(a) = \alpha, \quad x(b) = \beta \quad (8)$$

在 $a \leq t \leq b$ 上有惟一解 $x = x(t)$ 。

9.7.2 打靶法

实质：把边值问题转化为初值问题

以二阶系统为例，考虑边值问题：

$$y''(x) = f(x, y(x), y'(x)), \quad x \in [a, b]$$

$$y(a) = A, \quad y(b) = B$$

变换：

$$\begin{array}{l} y_1 = y \\ y_2 = y' \end{array} \quad \Longrightarrow \quad \begin{array}{l} y_1'(x) = y_2(x) \\ y_2'(x) = f(x, y_1, y_2), \quad x \in [a, b] \\ y_1(a) = A, \quad y_1(b) = B \end{array}$$

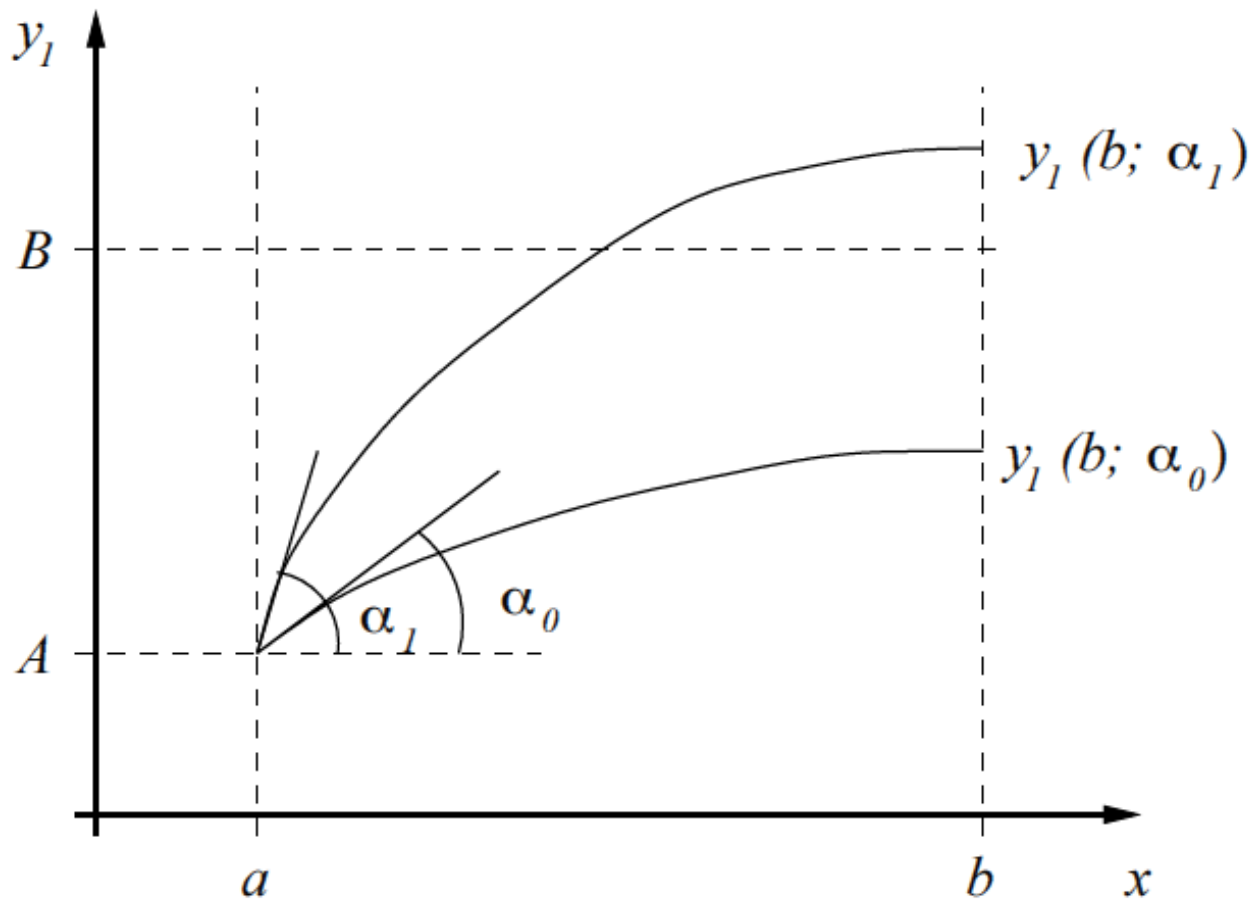
先猜测一个初始斜率 $y'(a) = y_2(a) = \alpha$ ，通过解初值问题

$$y_1' = y_2, \quad y_2' = f(x, y_1, y_2), \quad x \in [a, b]$$

$$y_1(a) = A, \quad y_2(a) = \alpha$$

初值问题的解为： $y_1(x; \alpha), y_2(x; \alpha)$ 如何求 α ?

找到 α 满足： $y_1(b; \alpha) = B$ 即把问题转化为求方程 $y_1(b, \alpha) = B$ 的根。



打靶法

1 割线法

以两个不同的 α 值求解初值问题，得到两个解：

$$y_1(x; \alpha_0), y_1(x; \alpha_1)$$

根据初值条件知： $y_1(a; \alpha_0) = y_1(a; \alpha_1) = A$

假设 $y_1(b; \alpha)$ 是 α 的线性函数，可取 α 为：

$$\alpha_2 = \alpha_0 + \frac{B - y_1(b; \alpha_0)}{y_1(b; \alpha_1) - y_1(b; \alpha_0)} (\alpha_1 - \alpha_0)$$

迭代求解公式：

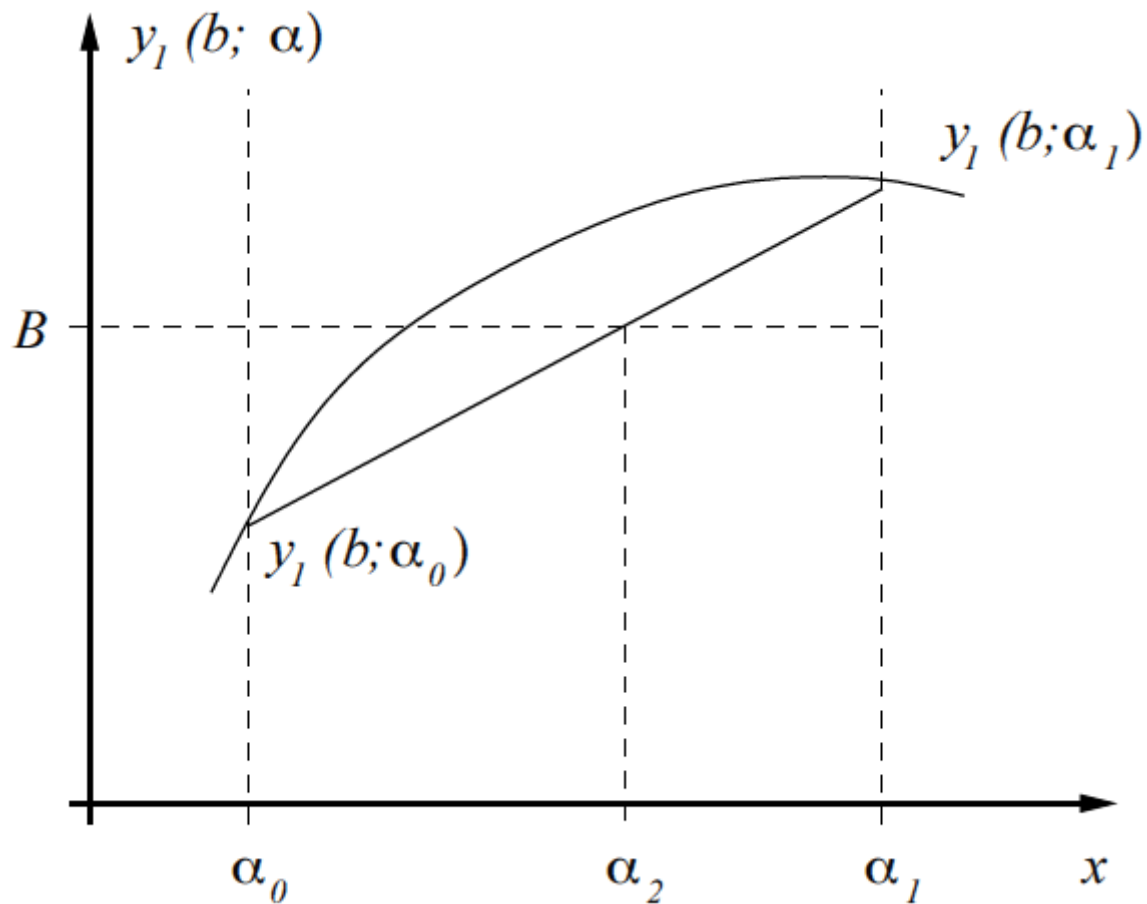
$$\alpha_{m+1} = \alpha_{m-1} + \frac{B - y_1(b; \alpha_{m-1})}{y_1(b; \alpha_m) - y_1(b; \alpha_{m-1})} (\alpha_m - \alpha_{m-1})$$

结束条件：

$$|1 - y_1(b; \alpha_{m-1})/B| \leq \varepsilon$$

9.7.2 打靶法

割线法的几何解释：



线性近似：按割线求根

2 牛顿法

求解非线性方程（组）： $y_1(b; \alpha) = B$

在已知初值 α_0 的处Taylor展开：

$$y_1(b; \alpha_1) = y_1(b; \alpha_0) + \frac{\partial y_1}{\partial \alpha}(b; \alpha_0)(\alpha_1 - \alpha_0) + O(\alpha_1 - \alpha_0)^2 = B$$

线性近似：

$$\alpha_1 = \alpha_0 + (B - y_1(b; \alpha_0)) / \frac{\partial y_1}{\partial \alpha}(b; \alpha_0)$$

迭代求解公式：

$$\alpha_{m+1} = \alpha_m + (B - y_1(b; \alpha_m)) / \frac{\partial y_1}{\partial \alpha}(b; \alpha_m) \quad \frac{\partial y_1}{\partial \alpha}(b; \alpha) ?$$

结束条件：

$$|1 - y_1(b; \alpha_{m+1}) / B| \leq \varepsilon$$

打靶法程序实现

```
function L=linsht(F1,F2,a,b,alpha,beta,M)
%Input   - F1 and F2 are the systems of first-order equations
%         representing the I.V.P.'s (9) and (10), respectively;
%         input as M-file functions
%         - a and b are the endpoints of the interval
%         - alpha = x(a) and beta = x(b); the boundary conditions
%         - M is the number of steps
%Output - L = [T' X]; where T' is the (M+1) x 1 vector of abscissas
%         and X is the (M+1) x 1 vector of ordinates

Za=[alpha,0];
[T,Z]=rks4(F1,a,b,Za,M);
U=Z(:,1);

%Solve the system F2
Za=[0,1];
[T,Z]=rks4(F2,a,b,Za,M);
V=Z(:,1);

%Calculate the solution to the boundary value problem
X=U+(beta-U(M+1))*V/V(M+1);
L=[T' X];
```



9.7.3 有限差分法

用差分近似代替微分，将微分方程化为代数方程求解

以二阶系统为例，边值问题：

$$y''(x) = f(x, y(x), y'(x)), \quad x \in [a, b]$$

$$y(a) = A, \quad y(b) = B$$

- 有限差分近似

将区间 $x \in [a, b]$ 等分为 N 个子区间

$$h = \frac{b-a}{N}, \quad x_i = a + ih, \quad (i = 0, 1, 2, \dots, N)$$

将 $y(x)$ 在 x_i 处 Taylor 展开：

$$y(x) = y(x_i) + y'(x_i)(x - x_i) + \frac{1}{2} y''(x_i)(x - x_i)^2 + \frac{1}{3!} y'''(x_i)(x - x_i)^3 + \dots$$

若取 $x=x_{i+1}=x+ih$:

$$y(x_{i+1}) = y(x_i) + y'(x_i)h + \frac{1}{2}h^2 y''(x_i) + \frac{1}{6}h^3 y'''(x_i) + \cdots$$

忽略二阶以上部分，得一阶导数的前向差分近似：

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} \quad \text{一阶精度}$$

若取 $x=x_{i-1}=x-ih$:

$$y(x_{i-1}) = y(x_i) - y'(x_i)h + \frac{1}{2}h^2 y''(x_i) - \frac{1}{6}h^3 y'''(x_i) + \cdots$$

忽略二阶以上部分，得一阶导数的后向差分近似：

$$y'(x_i) = \frac{y(x_i) - y(x_{i-1})}{h} \quad \text{一阶精度}$$

x_{i+1} 和 x_{i-1} 在 x_i 处的 Taylor 展开相减，忽略三阶以上部分，得一阶导数的中心差分近似：

$$y'(x_i) = \frac{y(x_{i+1}) - y(x_{i-1}))}{2h} \quad \text{二阶精度}$$

x_{i+1} 和 x_{i-1} 在 x_i 处的Taylor展开相加，忽略四阶以上部分，得二阶导数中心差分近似：

$$\begin{aligned} y''(x_i) &= \frac{y'(x_{i+1}) - y'(x_i)}{h} \\ &= \frac{\frac{y(x_{i+1}) - y(x_i)}{h} - \frac{y(x_i) - y(x_{i-1}))}{h}}{h} \quad \text{二阶精度} \\ &= \frac{y(x_{i+1}) + y(x_{i-1}) - 2y(x_i)}{h^2} \end{aligned}$$

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

$$x \in [a, b]$$

$$y(a) = A, \quad y(b) = B$$

离散化，将区间 $x \in [a, b]$ 等分为 N 个子区间：

$$h = \frac{b-a}{N}, \quad x_i = a + ih, \quad (i = 0, 1, 2, \dots, N)$$

在节点上应用中心差分公式，得到代数方程组：

$$\frac{y_{i+1} + y_{i-1} - 2y_i}{h^2} = f(x_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}), \quad (i = 1, 2, \dots, N-1)$$

$$y_0 = A, \quad y_N = B$$

考虑区间 $[a,b]$ 上线性微分方程的情况

$$x'' = p(t)x'(t) + q(t)x(t) + r(t) \quad (1)$$

边值条件为 $x(a) = \alpha, x(b) = \beta$ 。点 $a = t_0 < t_1 < \cdots < t_N = b$ 将区间 $[a, b]$ 划分为 N 个子区间, 其中 $h = (b - a)/N$, 且对 $j = 0, 1, \cdots, N, t_j = a + jh$ 。用第 6 章讨论的中心差分公式来近似导数

$$x'(t_j) = \frac{x(t_{j+1}) - x(t_{j-1}))}{2h} + O(h^2) \quad (2)$$

和

$$x''(t_j) = \frac{x(t_{j+1}) - 2x(t_j) + x(t_{j-1}))}{h^2} + O(h^2) \quad (3)$$

首先, 用 x_j 替换式(2)和式(3)右端项中的 $x(t_j)$ 项, 并将结果代入式(1), 得到

$$\begin{aligned} \frac{x_{j+1} - 2x_j + x_{j-1}}{h^2} + O(h^2) &= p(t_j) \left(\frac{x_{j+1} - x_{j-1}}{2h} + O(h^2) \right) \\ &\quad + q(t_j)x_j + r(t_j) \end{aligned} \quad (4)$$

然后略去式(4)中的项 $O(h^2)$, 并引入记号 $p_j = p(t_j), q_j = q(t_j)$ 和 $r_j = r(t_j)$, 得到差分方程

$$\frac{x_{j+1} - 2x_j + x_{j-1}}{h^2} = p_j \frac{x_{j+1} - x_{j-1}}{2h} + q_j x_j + r_j \quad (5)$$

用它来计算微分方程(1)的数值近似解。式(5)的两端同乘以 h^2 , 合并包含 x_{j-1} , x_j 和 x_{j+1} 的项, 并将它们组织为线性方程组:

$$\left(\frac{-h}{2}p_j - 1\right)x_{j-1} + (2 + h^2q_j)x_j + \left(\frac{h}{2}p_j - 1\right)x_{j+1} = -h^2r_j \quad (6)$$

$j=1, 2, \dots, N-1$, 其中 $x_0 = \alpha$ 和 $x_N = \beta$ 。方程组(6)是我们熟悉的三对角形式, 以矩阵方式表示能看得更清楚:

$$\begin{bmatrix} 2 + h^2q_1 & \frac{h}{2}p_1 - 1 & & & \\ \frac{-h}{2}p_2 - 1 & 2 + h^2q_2 & \frac{h}{2}p_2 - 1 & & \\ & \frac{-h}{2}p_j - 1 & 2 + h^2q_j & \frac{h}{2}p_j - 1 & \\ O & & \frac{-h}{2}p_{N-2} - 1 & 2 + h^2q_{N-2} & \frac{h}{2}p_{N-2} - 1 \\ & & \frac{-h}{2}p_{N-1} - 1 & 2 + h^2q_{N-1} & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_j \\ x_{N-2} \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} -h^2r_1 + e_0 \\ -h^2r_2 \\ -h^2r_j \\ -h^2r_{N-2} \\ -h^2r_{N-1} + e_N \end{bmatrix}$$

其中,

$$e_0 = \left(\frac{h}{2}p_1 + 1\right)\alpha, \quad e_N = \left(\frac{-h}{2}p_{N-1} + 1\right)\beta$$

当用步长 h 进行计算时, 数值近似解为离散点集 $\{(t_j, x_j)\}$, 如果解析解 $x(t_j)$ 已知, 则可将 x_j 和 $x(t_j)$ 进行比较。


```
function F=findiff(p,q,r,a,b,alpha,beta,N)
```

```
%Input   - p, q, and r are the coefficient  
functions of (1)
```

```
%         - a and b are the left and right  
endpoints
```

```
%         - alpha =x(a) and beta=x(b)
```

```
%         - N is the number of steps
```

```
%Output - F=[T' X']:where T' is the 1xN vector  
of abscissas and
```

```
%         X' is the 1xN vector of ordinates.
```

```
%Initialize vectors and h
```

```
T=zeros(1,N+1);
```

```
X=zeros(1,N-1);
```

```
Va=zeros(1,N-2);
```

```
Vb=zeros(1,N-1);
```

```
Vc=zeros(1,N-2);
```

```
Vd=zeros(1,N-1);
```

```
h=(b-a)/N;
```

```
%Calculate the constant vector B in AX=B
```

```
Vt=a+h:h:a+h*(N-1);
```

```
Vb=-h^2*r(Vt);
```

```
Vb(1)=Vb(1)+(1+h/2*p(Vt(1)))*alpha;
```

```
Vb(N-1)=Vb(N-1)+(1-h/2*p(Vt(N-1)))*beta;
```

有限差分方法数值实现

```
%Calculate the main diagonal of A in AX=B
```

```
Vd=2+h^2*q(Vt);
```

```
%Calculate the super diagonal of A in AX=B
```

```
Vta=Vt(1,2:N-1);
```

```
Va=-1-h/2*p(Vta);
```

```
%Calculate the sub diagonal of A in AX=B
```

```
Vtc=Vt(1,1:N-2);
```

```
Vc=-1+h/2*p(Vtc);
```

```
%Solve AX=B using trisys
```

```
X=trisys(Va,Vd,Vc,Vb);
```

```
T=[a,Vt,b];
```

```
X=[alpha,X,beta];
```

```
F=[T' X'];
```

```
function X=trisys (A,D,C,B)
```

```
%Input   - A is the sub diagonal of the coefficient matrix  
%         - D is the main diagonal of the coefficient matrix  
%         - C is the super diagonal of the coefficient matrix  
%         - B is the constant vector of the linear system  
%Output - X is the solution vector
```

```
N=length(B);  
for k=2:N  
    mult=A(k-1)/D(k-1);  
    D(k)=D(k)-mult*C(k-1);  
    B(k)=B(k)-mult*B(k-1);  
end
```

```
X(N)=B(N)/D(N);
```

```
for k= N-1:-1:1  
    X(k)=(B(k)-C(k)*X(k+1))/D(k);  
end
```

本章教学要求及重点难点

- 理解：微分方程求解的基本思想、方法和理论
- 熟练掌握：Eular方法、显式欧拉格式的改进、龙格-库塔方法以及微分方程组求解
- 重点：Eular方法、显式欧拉格式的改进、龙格-库塔方法以及微分方程组求解
- 了解：预报-矫正方法和二阶常微分方程边值问题求解