

## 第3 节

# 龙格—库塔法

# 龙格-库塔法的基本思想 (I)

考察差商  $\frac{y(x_{n+1}) - y(x_n)}{h}$

由微分中值定理, 可得

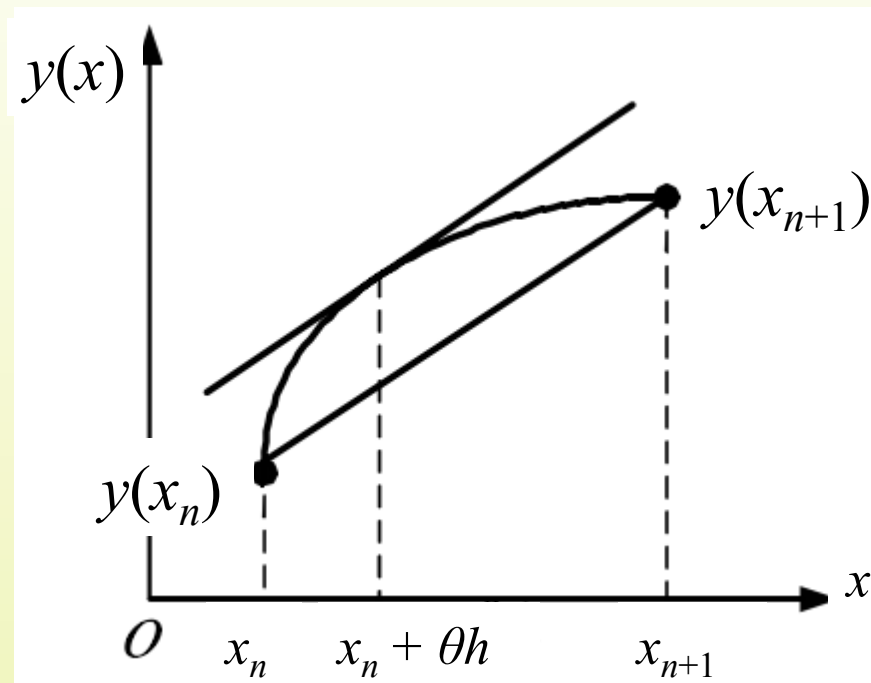
$$\frac{y(x_{n+1}) - y(x_n)}{h} = y'(x_n + \theta h)$$

$(0 < \theta < 1)$

因此微分方程  
的数值解为

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

$$y(x_{n+1}) = y(x_n) + h \cdot f[(x_n + \theta h), y(x_n + \theta h)]$$



## 龙格—库塔法的基本思想 (2)

这里的  $f[(x_n + \theta h), y(x_n + \theta h)]$  称为区间  $(x_n, x_{n+1})$  上的平均斜率, 记作  $k^*$ , 即

$$k^* = f[(x_n + \theta h), y(x_n + \theta h)]$$

因此只要对平均斜率  $k^*$  提供一种算法, 便相应地得到一种微分方程数值计算公式。

用这种观点很容易看出欧拉公式、后退欧拉公式、改进欧拉公式的特点。

$$y(x_{n+1}) = y(x_n) + h \cdot k^*$$

▣ 欧拉公式

$$k^* \approx f(x_n, y_n) \rightarrow y_{n+1} = y_n + hf(x_n, y_n)$$

▣ 后退欧拉公式

$$k^* \approx f(x_{n+1}, y_{n+1}) \rightarrow y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

▣ 梯形公式

$$k^* \approx \frac{f(x_n, y_n) + f(x_{n+1}, y_{n+1})}{2}$$
$$\rightarrow y_{n+1} = y_n + h \frac{f(x_n, y_n) + f(x_{n+1}, y_{n+1})}{2}$$

## 龙格-库塔法的基本思想 (3)

改进欧拉公式利用了 $(x_n, x_{n+1})$ 两个点的斜率值

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_{n+1}, y_n + hk_1)$$

取算术平均作为平均斜率  $k^*$  的近似值

$$k^* \approx \frac{k_1 + k_2}{2}$$

其中 $k_2$ 是通过已知信息  $y_n$  利用欧拉公式预报的。

## § 2 龙格 - 库塔法 /\* Runge-Kutta Method \*/



建立高精度的单步递推格式。



单步递推法的**基本思想**是从  $(x_n, y_n)$  点出发，以**某一斜率**沿直线达到  $(x_{n+1}, y_{n+1})$  点。欧拉法及其各种变形所能达到的最高精度为**2阶**。

🔗 考察改进的欧拉法，可以将其改写为：

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h, y_n + hk_1) \\ y_{n+1} = y_n + h \left[ \frac{k_1}{2} + \frac{k_2}{2} \right] \end{cases}$$

步长一定是一个 **$h$** 吗？

**斜率**  
一定取  $k_1$   $k_2$   
的**平均值**吗？

## 二阶龙格-库塔公式的推导 (1)

首先推广改进欧拉公式。随意考察区间  $(x_n, x_{n+1})$  内一点

$$x_{n+p} = x_n + ph \quad (0 < p \leq 1)$$

我们希望用  $x_n, x_{n+p}$  两个点的斜率值  $k_1$  和  $k_2$  加权平均得到平均斜率  $k^*$ , 即令

$$y_{n+1} = y_n + h[(1-\lambda)k_1 + \lambda k_2]$$

这里的  $\lambda$  为待定常数。  $(0 \leq \lambda \leq 1)$

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h, y_n + hk_1) \\ y_{n+1} = y_n + h \left[ \frac{k_1}{2} + \frac{k_2}{2} \right] \end{cases}$$



$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + ph, y_n + phk_1) \\ y_{n+1} &= y_n + h[(1-\lambda)k_1 + \lambda k_2] \end{aligned}$$

## 二阶龙格-库塔公式的推导 (3)

分别将  $k_1$  和  $k_2$  Taylor展开, 可得

$$k_1 = y'(x_n)$$

$$k_2 = y'(x_n) + p h y''(x_n) + O(h^2)$$

$$y_{n+1} = y(x_n) + h y'(x_n) + \lambda p h^2 y''(x_n) + O(h^3)$$

$y(x)$ 在  $x = x_{n+1}$  处二阶Taylor展开式为

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

比较系数可以发现, 要使得两者具有同样的局部截断误差, 需要满足

$$\lambda p = \frac{1}{2}$$

称满足这一条件的所有数值计算格式为  
**二阶龙格-库塔公式**



## 二阶龙格-库塔公式的推导 (4)

当取 $p=1, \lambda=1/2$ 时, 所得的公式即为改进欧拉公式。

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + ph, y_n + phk_1)$$

$$y_{n+1} = y_n + h[(1-\lambda)k_1 + \lambda k_2]$$



$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + h, y_n + hk_1)$$

$$y_{n+1} = y_n + h\left[\frac{k_1 + k_2}{2}\right]$$

## 二阶龙格-库塔公式的推导 (4)

当取 $p=1/2, \lambda=1$ 时, 所得的公式称为变型的欧拉公式或中点格式

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + ph, y_n + phk_1)$$

$$y_{n+1} = y_n + h[(1-\lambda)k_1 + \lambda k_2]$$



$$k_1 = f(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right)$$

$$y_{n+1} = y_n + hk_2$$

含义: 用一般欧拉公式预测 $[x_n, x_{n+1}]$ 区间的中点的斜率, 并利用此预测值计算 $y_{n+1}$

欧拉公式

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

后退欧拉公式

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

每一步仅计算  $f(x,y)$  一次，精度为一阶。

梯形公式

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

改进Euler公式可改写成

$$y_{n+1} = y_n + h \left[ \frac{f(x_n, y_n) + f(x_{n+1}, y_n + hk_1)}{2} \right]$$

每一步计算  $f(x,y)$  二次，精度为二阶。

上述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  $y(x_{n+1})$  的近似值  $y_{n+1}$
- 增加计算的次数  $f(x,y)$  的次数, 可提高精度。

可考虑用函数 $f(x,y)$ 在若干点上的函数值的线性组合来构造近似公式。

要求近似公式在 $(x_n, y_n)$ 处的Taylor展开式与解 $y(x)$ 在 $x_n$ 处的Taylor展开式的前面几项重合，从而使近似公式达到所需要的阶数。

或者说，在 $[x_n, x_{n+1}]$ 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。

*Runge-Kutta* 方法是一种高精度的单步法, 简称R-K法

## 二阶龙格—库塔法

在 $[x_n, x_{n+1}]$ 区间内取两个点。

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + ph, y_n + phk_1)$$

$$y_{n+1} = y_n + h[(1-\lambda)k_1 + \lambda k_2]$$

满足如下条件

$$\lambda p = \frac{1}{2}$$

存在无穷多个解。每一步计算 $f(x,y)$ 二次，精度为二阶。

所有满足上式的格式统称为**2阶龙格 - 库塔格式**。

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。

## 三阶龙格—库塔法

在 $[x_n, x_{n+1}]$ 区间内取三个点。

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + h, y_n - hk_1 + 2hk_2)$$

$$y_{n+1} = y_n + h[\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3]$$

常用的三阶龙格—库塔法。

每一步计算  $f(x,y)$  三次，精度为三阶。

## 四阶(经典)龙格—库塔法

在 $[x_n, x_{n+1}]$ 区间内取四个点。

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

$$y_{n+1} = y_n + h[\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4]$$

经典的四阶龙格-库塔法。

每一步计算 $f(x,y)$ 四次，精度为四阶。



注:

👉 龙格-库塔法的主要运算在于计算  $k_i$  的值, 即计算  $f$  的值。**Butcher** 于**1965**年给出了计算量与可达到的最高精度阶数的关系:

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	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^4)$	$O(h^5)$	$O(h^6)$	$O(h^{n-2})$

👉 由于龙格-库塔法的导出基于泰勒展开, 故精度主要受解函数的光滑性影响。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。

# 例1. 使用三阶和四阶R-K方法 计算初值问题

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

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + h, y_n - hk_1 + 2hk_2)$$

$$y_{n+1} = y_n + h[\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3]$$

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

$$y_{n+1} = y_n + h[\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4]$$

例1. 使用三阶和四阶R-K方法  
计算初值问题

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

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(x_n + h, y_n - hk_1 + 2hk_2)$$

$$y_{n+1} = y_n + h[\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3]$$

解: (1) 使用三阶R-K方法

$$k_1 = y_0^2 = 1$$

$$k_2 = (y_0 + \frac{1}{2}hk_1)^2 = 1.103$$

$$k_3 = (y_0 - hk_1 + 2hk_2)^2 = 1.256$$

$$y_1 = y_0 + \frac{h}{6}(k_1 + 4k_2 + k_3) = 1.111$$

其余结果如下：

$i$	$x_i$	$k_1$	$k_2$	$k_3$	$y_i$
1.000	0.100	1.000	1.103	1.256	1.111
2.000	0.200	1.235	1.376	1.595	1.250
3.000	0.300	1.562	1.764	2.092	1.428
4.000	0.400	2.040	2.342	2.866	1.666
5.000	0.500	2.777	3.259	4.163	1.999

## (2) 如果使用 四阶R-K方法

$$\begin{cases} y' = y^2 & 0 \leq x \leq 0.5 \\ y(0) = 1 \end{cases}$$

$$k_1 = y_0^2 = 1$$

$$k_2 = (y_0 + \frac{1}{2} h k_1)^2 = 1.103$$

$$k_3 = (y_0 + \frac{1}{2} h k_2)^2 = 1.113$$

$$k_4 = (y_0 + h k_3)^2 = 1.235$$

$$y_1 = y_0 + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) = 1.1111$$

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_2)$$

$$k_4 = f(x_n + h, y_n + h k_3)$$

$$y_{n+1} = y_n + h [\frac{1}{6} k_1 + \frac{2}{6} k_2 + \frac{2}{6} k_3 + \frac{1}{6} k_4]$$

其余结果如下：

$i$	$x_i$	$k_1$	$k_2$	$k_3$	$k_4$	$y_i$
1.000	0.100	1.000	1.103	1.113	1.235	1.111
2.000	0.200	1.235	1.376	1.392	1.563	1.250
3.000	0.300	1.562	1.764	1.791	2.042	1.429
4.000	0.400	2.040	2.342	2.389	2.781	1.667
5.000	0.500	2.777	3.259	3.348	4.006	2.000

## 四阶龙格-库塔公式算例

试用四阶龙格-库塔公式计算下列初值问题。

$$y'(x) = 1 - \frac{2xy}{1+x^2}, y(0) = 0, 0 \leq x \leq 2$$

取步长 $h=1$ ，并与精确值

$$y(x) = \frac{x(3+x^2)}{3(1+x^2)} \quad \text{比较。}$$

$$k_1 = f(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

$$y_{n+1} = y_n + h\left[\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4\right]$$

解：由四阶龙格-库塔公式得：

$n$	$x_n=nh$	$y_n$	$y(x_n)$ 精确值
0	0	0	0
1	1.0	0.666667	0.666667
2	2.0	0.933333	0.933333



# 第5 节

## 多步法

# 基本思路 (I)

多步法是指  $y_{n+1}$  近似地用  $y_n, y_{n-1}, \dots$  和  $f(x_{n+1}, y_{n+1}), f(x_n, y_n), f(x_{n-1}, y_{n-1}), \dots$  来表达, 而不像单步法只用到前一步长的数据。

多步法的通式为

$$y_{n+1} = a_0 y_n + a_1 y_{n-1} + \dots + a_p y_{n-p} \\ + h[b_{-1} f(x_{n+1}, y_{n+1}) + b_0 f(x_n, y_n) + b_1 f(x_{n-1}, y_{n-1}) + \dots + b_p f(x_{n-p}, y_{n-p})]$$

$$= \sum_{i=0}^p a_i y_{n-i} + h \sum_{i=-1}^p b_i f(x_{n-i}, y_{n-i})$$

在  $x_n$  点作泰勒展开

$$y_{n+1} = \left( \sum_{i=0}^p a_i \right) y_n + \sum_{j=1}^m \left[ \frac{h^j}{j!} \left( \sum_{i=0}^p a_i (-i)^j + j \sum_{i=-1}^p b_i (-i)^{j-1} \right) y_n^{(j)} \right] + \dots$$

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

比较两式，对应系数相等的话，则有

$$\sum_{i=0}^p a_i = 1$$

$$\sum_{i=0}^p a_i (-i)^j + j \sum_{i=-1}^p b_i (-i)^{j-1} = 1 \quad j = 1, 2, \dots, m$$

# 第6 节

## Adam法

## 基本思路 (I)

上面曾讨论过多步法的通用表达式为：

$$y_{n+1} = \sum_{i=0}^p a_i y_{n-i} + h \sum_{i=-1}^p b_i f(x_{n-i}, y_{n-i})$$

且满足

$$\sum_{i=0}^p a_i = 1$$

$$\sum_{i=0}^p a_i (-i)^j + j \sum_{i=-1}^p b_i (-i)^{j-1} = 1 \quad j = 1, 2, \dots, m \quad (9)$$

$$\sum_{i=0}^p a_i (-i)^j + j \sum_{i=-1}^p b_i (-i)^{j-1} = 1 \quad j = 1, 2, \dots, m$$

## 显式Adam方法 (I)

Adam方法是令系数 $a_1=a_2=\dots=a_p=0$ ，得 $a_0=1$ ，则多步法的公式变为：

$$y_{n+1} = y_n + h \sum_{i=-1}^p b_i f(x_{n-i}, y_{n-i})$$

可选择显式法还是隐式法。

显式Adam方法，又称“Adam's-Bashforth”法，是令 $b_{-1}=0$ ，由式（9）得

$$\sum_{i=0}^p (-i)^{(j-1)} b_i = \frac{1}{j} \quad (j = 1, 2, \dots, m) \quad (10)$$

$p+1$ 个未知数, $m$ 个方程，所以 $m$ 应大于等于 $p+1$

## 显式Adam方法 (2)

$$\sum_{i=0}^p (-i)^{(j-1)} b_i = \frac{1}{j} (j=1, 2, \dots, m)$$

式 (10) 也可写成矩阵形式:

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & -1 & -2 & \cdots & -p \\ 0 & 1 & 4 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+1} \end{pmatrix}$$

选择所希望的阶数, 即可通过式 (11) 计算系数 $b_i$ 。

$$\sum_{i=0}^p (-i)^{(j-1)} b_i = \frac{1}{j} (j=1, 2, \dots, m) \quad (10)$$

$$\begin{aligned}
 & \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & -1 & -2 & \cdots & -p \\ 0 & 1 & 4 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+1} \end{pmatrix} \\
 & = \begin{pmatrix} (0)^0 & (-1)^0 & (-2)^0 & \cdots & (-p)^0 \\ (0)^1 & (-1)^1 & (-2)^1 & \cdots & (-p)^1 \\ (0)^2 & (-1)^2 & (-2)^2 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (0)^p & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} \quad (11)
 \end{aligned}$$



## 显式Adam方法一例子 (I)

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & -1 & -2 & \cdots & -p \\ 0 & 1 & 4 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+1} \end{pmatrix}$$

试导出三阶Adam's-Bashforth法

解：令 $p=2$ ，有

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{pmatrix}$$

得到

$$b_0 = \frac{23}{12}, b_1 = -\frac{16}{12}, b_2 = \frac{5}{12}$$

从而，得到三阶Adam's-Bashforth法为：

$$y_{n+1} = y_n + \frac{1}{12} h [23f(x_n, y_n) - 16f(x_{n-1}, y_{n-1}) + 5f(x_{n-2}, y_{n-2})]$$

$$\sum_{i=0}^p a_i (-i)^j + j \sum_{i=-1}^p b_i (-i)^{j-1} = 1 \quad j = 1, 2, \dots, m$$

## 隐式Adam方法 (I)

当令  $b_{-1} \neq 0$ , Adam方法称为隐式Adam方法, 又称  
“Adam's-moulton”法。此时:

$$y_{n+1} = y_n + h \sum_{i=-1}^p b_i f(x_{n-i}, y_{n-i})$$

由式 (9) 得:

$$\sum_{i=-1}^p (-i)^{(j-1)} b_i = \frac{1}{j} \quad (j = 1, 2, \dots, m)$$

$p+2$ 个未知数,  $m$ 个方程, 所以  $m$  应大于等于  $p+2$

## 隐式Adam方法 (2)

$$\sum_{i=-1}^p (-i)^{(j-1)} b_i = \frac{1}{j} (j = 1, 2, \dots, m)$$

写成矩阵形式为：

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & -1 & -2 & \cdots & -p \\ 1 & 0 & 1 & 4 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_{-1} \\ b_0 \\ b_1 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+2} \end{pmatrix} \quad (14)$$

$$\sum_{i=-1}^p (-i)^{(j-1)} b_i = \frac{1}{j} (j = 1, 2, \dots, m)$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & -1 & -2 & \cdots & -p \\ 1 & 0 & 1 & 4 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_{-1} \\ b_0 \\ b_1 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+2} \end{pmatrix}$$

$$= \begin{pmatrix} (1)^0 & (0)^0 & (-1)^0 & (-2)^0 & \cdots & (-p)^0 \\ (1)^1 & (0)^1 & (-1)^1 & (-2)^1 & \cdots & (-p)^1 \\ (1)^2 & (0)^2 & (-1)^2 & (-2)^2 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (1)^p & (0)^p & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_{-1} \\ b_0 \\ b_1 \\ \vdots \\ b_p \end{pmatrix}$$

## 隐式Adam方法一例子 (I)

试推导三阶Adam's-moulton法

解：令 $p=1$ ，有

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_{-1} \\ b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \end{pmatrix}$$

计算得到

$$b_{-1} = \frac{5}{12}, b_0 = \frac{8}{12}, b_1 = -\frac{1}{12}$$

## 隐式Adam方法一例子（2）

从而，三阶Adam's-moulton法为：

$$x_{n+1} = x_n + \frac{1}{12} h [5f(x_{n+1}, t_{n+1}) + 8f(x_n, t_n) - f(x_{n-1}, t_{n-1})]$$

注意在算法的执行过程中需要采用迭代解法，因为是隐式的。

避免迭代的一般性做法是采用**预测—校正**方法，即用显式Adam做预测，用隐式Adam方法做校正。

# Adam法：显示Vs 隐式

$$y_{n+1} = \sum_{i=0}^p a_i y_{n-i} + h \sum_{i=-1}^p b_i f(x_{n-i}, y_{n-i})$$

$$a_0=1, \quad a_1=a_2=\dots=a_p=0 \quad \text{显示和隐式}$$

显示

$$\begin{pmatrix} (0)^0 & (-1)^0 & (-2)^0 & \cdots & (-p)^0 \\ (0)^1 & (-1)^1 & (-2)^1 & \cdots & (-p)^1 \\ (0)^2 & (-1)^2 & (-2)^2 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (0)^p & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+1} \end{pmatrix}$$

隐式

$$\begin{pmatrix} (1)^0 & (0)^0 & (-1)^0 & (-2)^0 & \cdots & (-p)^0 \\ (1)^1 & (0)^1 & (-1)^1 & (-2)^1 & \cdots & (-p)^1 \\ (1)^2 & (0)^2 & (-1)^2 & (-2)^2 & \cdots & (-p)^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (1)^p & (0)^p & (-1)^p & (-2)^p & \cdots & (-p)^p \end{pmatrix} \begin{pmatrix} b_{-1} \\ b_0 \\ b_1 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{p+2} \end{pmatrix}$$

# PSCAD-EMTDC基础课程公开课

主讲老师：乐健副教授（武汉大学）

时 间：**2015年5月24日（周日）**

地 点：浙江大学玉泉校区教四**306**

费 用：免费

报名方式：

[http://www.sciencesoftware.com.cn/Registers\\_ADD.aspx?trid=E8009E6723C59597](http://www.sciencesoftware.com.cn/Registers_ADD.aspx?trid=E8009E6723C59597)

注意事项：提前网上报名。

上课时自带电量充足的电脑，便于同步上机操作。



# PSCAD-EMTDC基础课程公开课

课程内容:

## 1、PSCAD基本操作和介绍

- ① PSCAD的主要设置：软件设置、Workspace设置、Project设置、Canvas设置；
- ② PSCAD基本仿真流程、程序结构；
- ③ PSCAD主元件库元件介绍；

## 2、PSCAD自定义元件方法

- ① 元件和组件的图形外观定义、输入输出端口设置；
- ② 元件的代码编制；
- ③ 调用外部源代码程序；
- ④ 元件和组件的管理和调用；
- ⑤ 元件和组件的信号传递方法；

## 3、PSCAD中高级操作

- ① PSCAD的数据输入输出方法；
- ② PSCAD与MATLAB的接口；
- ③ PSCAD的多重运行方法；

## 4、PSCAD在新能源仿真中的应用

- ① 太阳能光伏发电原理及仿真；
- ② 风力发电原理及仿真；