§3.3 龙格—库塔方法

要进一步提高求解的精度，可用一种高精度的单步法—**龙格—库塔（Runge—Kutta）方法**，简称 **R—K法**。它采用了间接使用泰勒级数法的技术。

3.1 龙格—库塔公式的导出

对于一阶常微分方程**（1.1）**的解 $y=y(x)$，利用微分中值定理得

$$y(x_{i+1})-y(x_i)=y'(x_i+\xi_i h)(x_{i+1}-x_i)$$
$$(0<\xi_i<1)$$

即

$$y(x_{i+1})=y(x_i)+hy'(x_i+\xi_i h),(0<\xi_i<1)$$

也即

$$y(x_{i+1}) = y(x_i) + hK \qquad \textbf{(3.1)}$$

式中

$$\begin{aligned} K &= y'(x_i + \xi_i h) \\ &= f(x_i + \xi_i h, y(x_i + \xi_i h)) \end{aligned} \qquad \textbf{(3.2)}$$

K 可看作是$y = y(x)$ 在区间$[x_i, x_{i+1}]$ 上的**平均斜率**。这样，欧拉格式**(2.2)**相当于取(x_i, y_i) 点上斜率 $\left.\dfrac{dy}{dx}\right|_{x=x_i} = f(x_i, y_i)$ 作为平均斜率K 的近似值，当然

是十分粗糙的，因此精度必然很低。而改进的欧拉格式（2.12）可改写成

$$\begin{cases} y_{i+1} = y_i + \dfrac{h}{2}(K_1 + K_2) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_{i+1}, y_i + hK_1) \\ (i = 0,1,2,\cdots,\ n-1) \end{cases}$$

与（3.1）比较知，它相当于把(x_i, y_i) 和 (x_{i+1}, y_{i+1}) 两个点上的斜率K_1和K_2 的算术平均值作为（3.1）中的平均斜率K的近似值。其中K_2 是通过已知信息y_i来近似地预测的。

由此可以设想，取区间$[x_i, x_{i+1}]$ 内的某一个节点

$$x_{i+p} = x_i + ph(0 < p \leq 1)$$

上的斜率K_2 与$(x_i，y_i)$ 点上的斜率K_1作线性组合（即加权平均），作为平均斜率K的近似值，即

$$y_{i+1} = y_i + h(\lambda_1 K_1 + \lambda_2 K_2)$$

为了要得到$(x_{i+p}，y_{i+p})$ 点上的斜率 K_2， 需先预测

$$y_{i+p} = y_i + ph K_1$$

根据预测值 y_{i+p} 再来算出K_2：

$$K_2 = f(x_{i+p}, y_{i+p}) = f(x_i + ph, y_i + phK_1)$$

由此构成的计算格式为

$$\begin{cases} y_{i+1} = y_i + h(\lambda_1 K_1 + \lambda_2 K_2) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + ph, y_i + phK_1) \\ (i = 0,1,2, \quad n-1) \end{cases} \qquad \textbf{(3.3)}$$

上式含有三个待定参数: λ_1, λ_2 和 p。适当选定其值可使算法的局部截断误差为$O(h^3)$，即有二阶精度。

假定$y_i = y(x_i)$，分别将K_1和K_2作泰勒展开：

$$
\begin{aligned}
K_1 &= f(x_i, y_i) = y'(x_i) \\
K_2 &= f(x_i + ph, y_i + phK_1) \\
&= f(x_i, y_i) + ph[f_x(x_i, y_i) \\
&\quad + f(x_i, y_i) f_y(x_i, y_i)] + O(h^2) \\
&= y'(x_i) + phy''(x_i) + O(h^2)
\end{aligned}
$$

代入（3.3），得

$$
\begin{aligned}
y_{i+1} &= y(x_i) + h(\lambda_1 + \lambda_2) y'(x_i) \\
&\quad + \lambda_2 p h^2 y''(x_i) + O(h^3)
\end{aligned}
$$

将它与$y(x)$在x_{i+1}处的二阶泰勒展开式

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

进行比较系数后可知，只要

$$\begin{cases} \lambda_1 + \lambda_2 = 1 \\ \lambda_2 p = \dfrac{1}{2} \end{cases} \qquad (3.4)$$

成立，格式（3.3）的局部截断误差就等于$O(h^3)$，从而能具有二阶精度。

（3.4）中有三个待定系数: λ_1, λ_2 和 p，但却只有两个方程式，因此还有一个自由度。凡满足条件（3.4）的一族格式（3.3）统称为二阶龙格-库塔格式。

当 $p=1, \lambda_1=\lambda_2=1/2$时，二阶R—K格式（3.3）即为改进的欧拉公式（2.12）。

如取$p=1/2$，则 $\lambda_1=0, \lambda_2=1$，（3.3）就成为

$$\begin{cases} y_{i+1}=y_i+hK_2 \\ K_1=f(x_i, y_i) \\ K_2=f(x_i+\dfrac{h}{2}, y_i+\dfrac{h}{2}K_1) \\ (i=0,1,2,\quad,\ n-1) \end{cases} \qquad (3.5)$$

（3.5）称为**变形的欧拉格式**。

由于（3.5）中的

$$y_{i+\frac{1}{2}} = y_i + \frac{h}{2} K_1$$

是欧拉格式预测出来的区间$[x_i, x_{i+1}]$的中点$x_{i+1/2}$的近似解，K_2 就近似地等于此中点的斜率 $f(x_{i+\frac{1}{2}}, y(x_{i+\frac{1}{2}}))$，因此（3.5）就相当于用中点$x_{i+1/2}$的斜率作为（3.1）中的平均斜率K的近似值，故格式（3.5）也称为**中点格式**。粗看起来，$y_{i+1}=y_i+hK_2$中只含有一个斜率值K_2，但实际上K_2 是通过K_1 才能算出来的，因此，式

中还隐含着K_1。这样，每完成一步仍需计算函数f值两次，其计算工作量仍与改进的欧拉格式一样。

3.2 高阶龙格-库塔公式

要提高精度，可在区间$[x_i, x_{i+1}]$内取二个节点：

$$\begin{cases} x_{i+p} = x_i + ph \\ x_{i+q} = x_i + qh \end{cases} \quad (0 < p < q \le 1)$$

上的斜率K_2，K_3与点(x_i, y_i)上的斜率K_1加权平均，作为平均斜率K的近似值，即

$$y_{i+1} = y_i + h(\lambda_1 K_i + \lambda_2 K_2 + \lambda_3 K_3)$$

其中K_1和K_2仍如（3.3）。

利用区间$[x_i, x_{i+q}]$ 内的两个斜率K_1和K_2，加权平均作为其平均斜率K，来预测$y(x_{i+q})$：

$$y_{i+q} = y_i + qh(\alpha_1 K_1 + \alpha_2 K_2)$$

从而得到

$$\begin{aligned} K_3 &= f(x_{i+q}, y_{i+q}) \\ &= f(x_{i+q}, y_i + qh(\alpha_1 K_1 + \alpha_2 K_2)) \end{aligned}$$

由此构成的计算格式为

$$\begin{cases} y_{i+1} = y_i + h(\lambda_1 K_1 + \lambda_2 K_2 + \lambda_3 K_3) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + ph, y_i + phK_1) \\ K_3 = f(x_i + qh, y_i + qh(\alpha_1 K_1 + \alpha_2 K_2)) \end{cases} \qquad \textbf{(3.6)}$$

类似于二阶龙格-库塔格式的导出过程，运用泰勒展开的方法，可找出格式（3.6）的局部截断误差为$O(h^4)$，从而具有三阶精度所必须满足的条件为：

$$\begin{cases} \alpha_2+\alpha_1=1 \\ \lambda_1+\lambda_2+\lambda_3=1 \\ \lambda_2 p+\lambda_3 q=\dfrac{1}{2} \\ \lambda_2 p^2+\lambda_3 q^2=\dfrac{1}{3} \\ \lambda_3 pq\alpha_2=\dfrac{1}{6} \end{cases} \qquad (3.7)$$

其中共有七个待定系数：$\lambda_1,\lambda_2,\lambda_3,p,q,\alpha_1,\alpha_2$，但只有五个方程式，因此还有**两个自由度**。凡满足条件**（3.7）**的一族格式**（3.6）**统称为**三阶龙格-库塔格式**。

当待定系数取为

$$\lambda_1 = \lambda_3 = 1/6, \quad \lambda_2 = 2/3, \quad p = 1/2$$

$$q = 1, \quad \alpha_1 = -1, \quad \alpha_2 = 2$$

时的三阶龙格-库塔格式称为**库塔格式**，其具体形式为：

$$\begin{cases} y_{i+1} = y_i + \dfrac{h}{6}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + \dfrac{h}{2}, y_i + \dfrac{h}{2}K_1) \\ K_3 = f(x_i + h, y_i - hK_1 + 2hK_2) \end{cases} \quad \textbf{(3.8)}$$

继续推广这种处理过程，可得四阶龙格-库塔格式。最常用的一种经典龙格-库塔格式的具体形式如下：

$$\begin{cases} y_{i+1} = y_i + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(x_i, y_i) \\ K_2 = f(x_i + \dfrac{h}{2}, y_i + \dfrac{h}{2}K_1) \\ K_3 = f(x_i + \dfrac{h}{2}, y_i + \dfrac{h}{2}K_2) \\ K_4 = f(x_i + h, y_i + hK_3) \end{cases} \tag{3.9}$$

经典龙格—库塔方法的程序框图见图7-6。

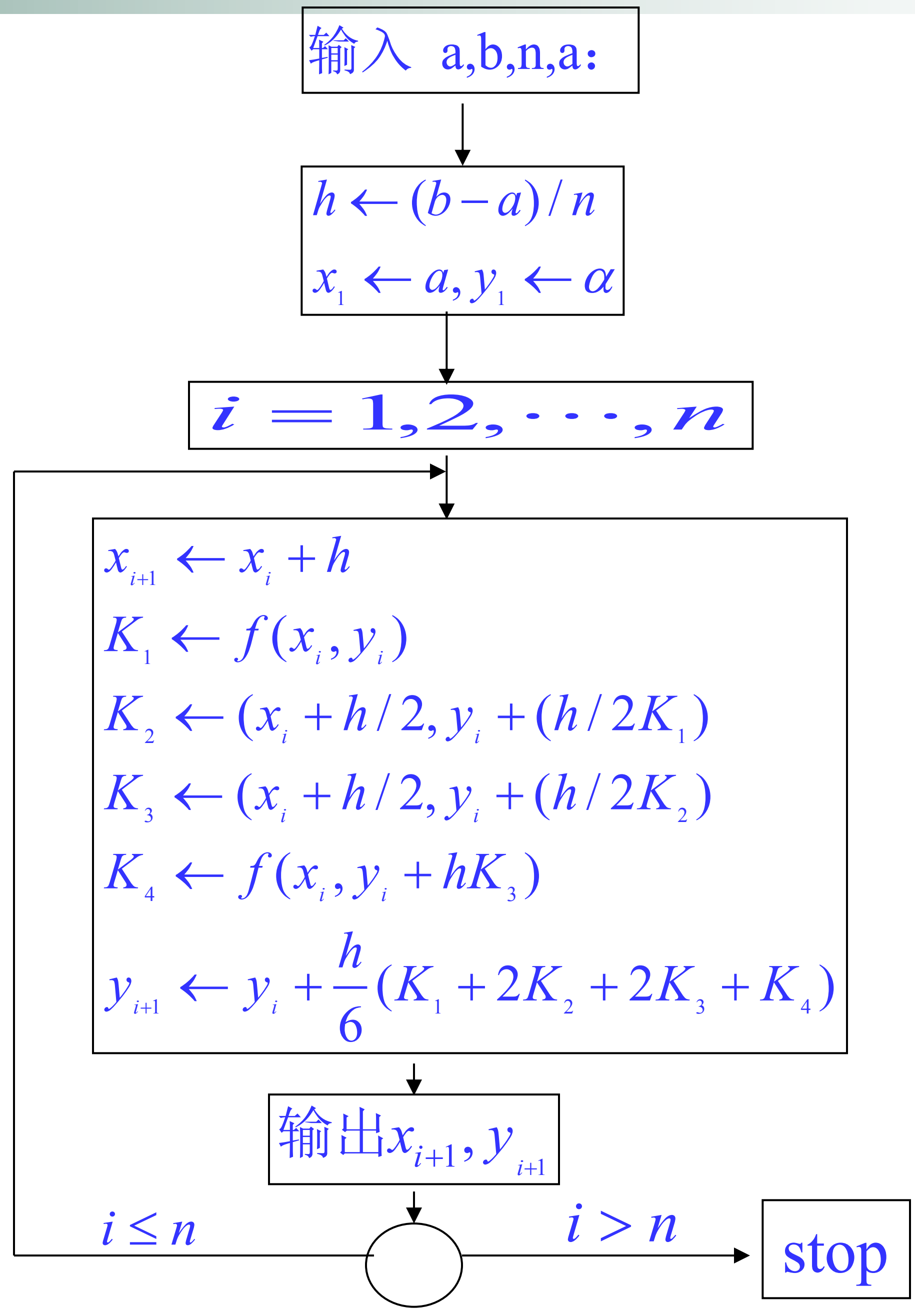
输入 a,b,n,a:
$h \leftarrow (b-a)/n$
$x_1 \leftarrow a, y_1 \leftarrow \alpha$
$i = 1,2,\cdots,n$
$x_{i+1} \leftarrow x_i + h$
$K_1 \leftarrow f(x_i, y_i)$
$K_2 \leftarrow (x_i + h/2, y_i + (h/2K_1)$
$K_3 \leftarrow (x_i + h/2, y_i + (h/2K_2)$
$K_4 \leftarrow f(x_i, y_i + hK_3)$
$y_{i+1} \leftarrow y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$
输出x_{i+1}, y_{i+1}
$i \le n$
$i > n$
stop

图7-6

例2 试分别用欧拉方法 $(h=0.025)$，改进的欧拉方法 $(h=0.05)$ 及经典R—K方法 $(h=0.1)$ ，求解下列初值问题，并比较三种方法所得结果的精度：

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

解 三种方法的具体算式分别如下：

欧拉格式： $y_{i+1} = y_i + 0.025(-y_i) = 0.975 y_i$

$$(h = 0.025)$$

改进的欧拉公式:

$$\begin{cases} y_p = y_i + 0.05(-y_i) = 0.95y_i \\ y_c = y_i + 0.05(-y_p) = 0.9525y_i \quad (h = 0.05) \\ y_{i+1} = 1/2(y_p + y_c) = 0.95125y_i \end{cases}$$

经典R—K格式:

$$\begin{cases} K_1 = -y_i \\ K_2 = -[y_i + \frac{0.1}{2}(-y_i)] = -0.95y_i \\ K_3 = -(y_i + \frac{0.1}{2}K_2) = -0.9525y_i \qquad (h = 0.1) \\ K_4 = -(y_i + 0.1K_3) = -0.90475y_i \\ y_{i+1} = y_i + \frac{0.1}{6}(K_1 + 2K_2 + 2K_3 + K_4) = 0.9048375y_i \end{cases}$$

表7-2

x_i	欧拉方法 $(h=0.025)$		改进的欧拉方法 $(h=0.05)$		经典$R-K$方法 $(h=0.1)$		精确解
	y_i	$\lvert y(x_i)-y_i\rvert$	y_i	$y(x_i)-y_i$	y_i	$y(x_i)-y_i$	$y(x_i)$
0	1		1		1		1
1	0.903688	1.15×10^{-3}	0.904877	3.91×10^{-5}	0.90484	8.2×10^{-8}	0.90484
0.2	0.816652	2.08×10^{-3}	0.818802	7.08×10^{-5}	0.81873	1.48×10^{-7}	0.81873
0.3	0.737998	2.82×10^{-3}	0.740914	9.62×10^{-5}	0.74082	2.02×10^{-7}	0.74082
0.4	0.666920	3.44×10^{-3}	0.670436	1.16×10^{-4}	0.67032	2.43×10^{-7}	0.67032
0.5	0.602688	3.84×10^{-3}	0.606662	1.31×10^{-4}	0.60653	2.75×10^{-7}	0.60653
0.6	0.544642	4.17×10^{-3}	0.548954	1.42×10^{-4}	0.54881	2.98×10^{-7}	0.54881
0.7	0.492186	4.40×10^{-3}	0.496736	1.50×10^{-4}	0.49659	3.15×10^{-7}	0.49659
0.8	0.444783	4.55×10^{-3}	0.449484	1.56×10^{-4}	0.44933	3.25×10^{-7}	0.44933
0.9	0.401945	4.63×10^{-3}	0.406728	1.58×10^{-4}	0.40657	3.32×10^{-7}	0.40657
1	0.363233	4.65×10^{-3}	0.368039	1.59×10^{-4}	0.36788	3.33×10^{-7}	0.36788

三种格式的 计算结果分别列于表7—2中。

上例中对三种方法采用了不同的步长h值，是为了使它们所耗的计算工作量大致相同，以便于比较。从表7—2可见，经典R—K方法的精度较改进的欧拉方法又有了很大的提高。关于这一结论也可以从理论上大致分析出来：

（1）欧拉方法的局部截断误差为

$$R'_E = C_1 h_E^2 = \frac{C_1}{16} \times 10^{-2}$$

计算四步后截断误差为

$$R_E = 4 \times \frac{C_1}{16} \times 10^{-2} = \frac{C_1}{4} \times 10^{-2}$$

（2）经典R—K方法的局部截断误差为

$$R_R = C_2 h_R^5 = C_2 \times 10^{-5}$$

C_1, C_2 为大致相同数量级下的常数，故有

$$R_R << R_E$$

注意： R—K方法的导出利用了泰勒展开，因此要求所求的解有较好的光滑性，如果解的光滑性差，则采用经典 R—K 法所得的数值解，其精度有可能反而不及改进的欧拉法。因此在实际计算中，应根据问题的具体情况来选择合适的算法。