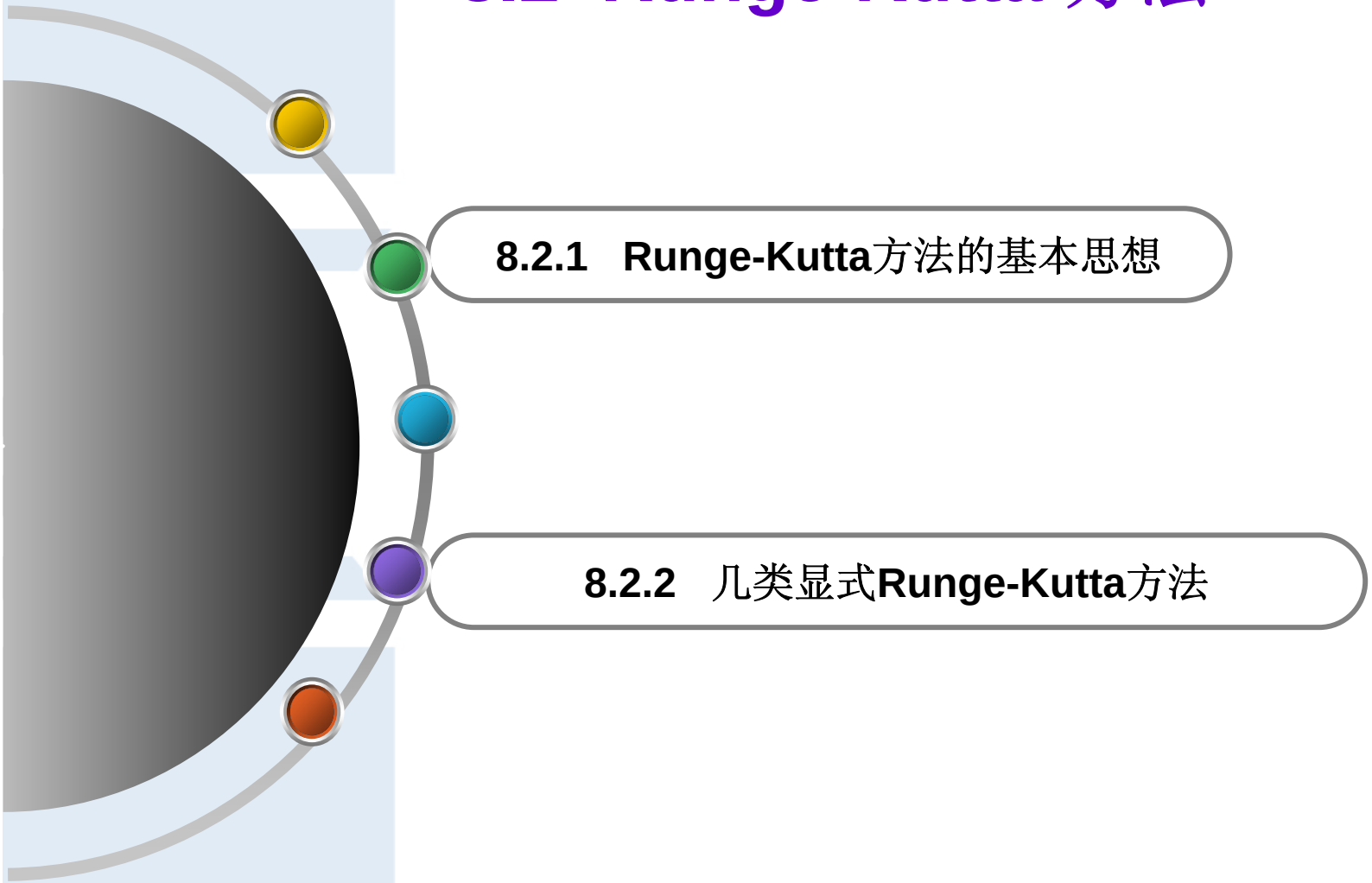
8.2 Runge-Kutta 方法



8.2.1 Runge-Kutta方法的基本思想

8.2.2 几类显式Runge-Kutta方法

8.2.1 Runge-Kutta方法的基本思想

显式**Euler**方法是最简单的单步法，它是一阶的，它可以看作**Talylor**展开后取前两项。因此，得到高阶方法的一个直接想法是用**Talylor**展开，如果能计算$y(x)$的高阶导数，则可写出**p**阶方法的计算方法

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}y''_n + \cdots + \frac{h^p}{p!}y_n^{(p)},$$

其中$y_n^{(j)}$是 $y^{(j)}(x_n)$ 的近似值，$j = 0,1,2,\cdots p$。若将 $f(x,\ y)$，$\partial f/\partial x, \partial f/\partial y, \cdots,$ 分别记成 f，f_x，$f_y, \cdots$，则对于二阶和三阶导数可表示为

$$y'' = f_x + f_y f,$$

$$y''' = f_{xx} + 2f_{xy}f + f_x f_y + f_{yy}f^2 + f_y^2\ f。$$

这个方法并不实用，因为一般情况下，求 $f(x，y)$的导数相当麻烦。从计算高阶导数的公式知道，方法的截断误差提高一阶，需要增加的计算量很大。但是由此启发我们用区间上若干个点的导数 f ，而不是高阶导数，将它们作线性组合得到平均斜率，将其与解的**Taylor**展开相比较，使前面若干项吻合，从而得到具有一定阶的方法。这就是**Runge-Kutta**方法的基本思想，其一般形式为

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{L}\lambda_i K_i, \\ K_1 &= f(x_n,\ y_n), \\ K_i &= f\left(x_n + c_i h, y_n + c_i h\sum_{j=1}^{i-1} a_{ij}K_j\right),\ \ i = 2,3,\cdots,\ L, \end{aligned} \tag{8.2.1}$$

其中，$c_i \le 1, \sum_{i=1}^{L} \lambda_i = 1, \sum_{j=1}^{i-1} a_{ij} = 1$ 。它的局部截断误差是

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\sum_{i=1}^{L} \lambda_i K_i^* \quad , \qquad (8.2.2)$$

其中，K_i^* 与 K_i 的区别在于：用微分方程准确解 $y(x_n)$ 代替 K_i 中的 y_n 就得到 K_i^* 。参数λ_i， c_i 和 a_{ij} 待定，确定它们的原则和方法是：将（8.2.2）式中的 $y(x_{n+1})$ 在 x_n 处作Taylor展开，将 K_i^* 在$(x_n, y(x_n))$ 处作二元Taylor展开，将展开式按H的幂次整理后，令 T_{n+1}中h的低次幂的系数为零，使 T_{n+1} 首项中h的幂次尽量高，比如使$T_{n+1} = O(h^{p+1})$，则称（8.2.1）式为L级p阶**Runge-Kutta方法**（简称**R-K法**）。

类似于显式R-K公式（8.2.1），稍加改变，就得到隐式R-K公式

$$y_{n+1} = y_n + h\sum_{i=1}^{L} \lambda_i K_i,$$

$$K_i = f\left(x_n + c_i h,\ y_n + c_i h\sum_{j=1}^{L} a_{ij} K_j\right),\ i = 1,2,\cdots,\ L。$$

它与显式R-K公式的区别在于：显式公式中，对系数 a_{ij} 求和的上限是 $i-1$，从而 a_{ij}构成的矩阵是一个严格下三角阵。而在隐式公式中，对系数 a_{ij} 求和的上限是L，从而 a_{ij} 构成的矩阵是方阵，需要用迭代法求出近似斜率 $K_i\ (i = 1,2,\cdots,\ L)$ 推导隐式公式的思路和方法与显式公式法类似。

8.2.2 几类显式Runge-Kutta方法

对于**L=2**，则

$$y_{n+1} = y_n + h(\lambda_1 K_1 + \lambda_2 K_2),$$
$$K_1 = f(x_n, \ y_n),$$
$$K_2 = f(x_n + c_2 h, \ y_n + c_2 h K_1).$$

其局部截断误差是

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h(\lambda_1 K_1^* + \lambda_2 K_2^*) \qquad (8.2.3)$$

将T_{n+1}中的各项作**Taylor**展开，并利用$y'(x_n)=f(x_n,\ y(x_n))$, $y''=f_x+f_y f$, 则有

$$y(x_{n+1})=y(x_n)+hy'(x_n)+\frac{h^2}{2}y''(x_n)+\frac{h^3}{6}y'''(x_n)+O(h^4),$$

$$K_1^*=f(x_n,\ y(x_n))=y'(x_n),$$

$$K_2^*=f(x_n+c_2h,\ y(x_n)+c_2hy'(x_n)),$$

$$=y'(x_n)+c_2hy''(x_n)+\frac{c_2^2h^2}{2}(f_{xx}+2f_{xy}f+f_{yy}f^2)+O(h^3),$$

将它们代入（**8.2.3**）式，整理后得

$$T_{n+1}=(1-\lambda_1-\lambda_2)hy'(x_n)+\left(\frac{1}{2}-\lambda_2c_2\right)h^2y''(x_n)$$

$$+h^3\left[\frac{1}{6}y'''(x_n)-\frac{\lambda_2c_2^2}{2}(f_{xx}+2f_{xy}f+f_{yy}f^2)\right]+O(h^4)$$

选取λ_1，λ_2和 c_2，使方法的阶尽可能高，就是使 h 和 h^2 的系数为零，因为h^3的系数一般不为零。于是得到方程组

$$\begin{cases} \lambda_1 + \lambda_2 = 1, \\ \lambda_2 c_2 = \dfrac{1}{2}。 \end{cases}$$

显然，该方程组有无穷多组解，从而得到一族二级二阶R-K方法。

若以c_2为自由参数，取$c_2 = 1/2$得中点公式

$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2},\ y_n + \frac{h}{2} f(x_n,\ y_n)\right), \qquad (8.2.4)$$

取c=2/3得Heun公式

$$y_{n+1} = y_n + \frac{h}{4}\left[f(x_n,\ y_n) + 3f\left(x_n + \frac{2}{3}h,\ y_n + \frac{2}{3}hf(x_n,\ y_n)\right)\right], \qquad (8.2.5)$$

取c=1得改进的Euler公式（8.1.6）。

对于L=3的情形，要计算三个斜率的近似值：

$$K_1 = f(x_n,\ y_n),$$
$$K_2 = f(x_n + c_2 h,\ y_n + c_2 h K_1),$$
$$K_3 = f(x_n + c_3 h,\ y_n + c_3 h(a_{31}K_1 + a_{32}K_2)).$$

类似于二阶方法的推导，可以得三阶的方法，所得系数应满足的方程组是

$$\begin{cases} \lambda_1 + \lambda_2 + \lambda_3 = 1, \quad a_{21} = 1, \\ \lambda_2 c_2 + \lambda_3 c_3 = \dfrac{1}{2}, \quad \lambda_2 c_2^2 + \lambda_3 c_3^2 = \dfrac{1}{3}, \\ \lambda_3 c_2 c_3 a_{32} = \dfrac{1}{6}, \quad a_{31} + a_{32} = 1。 \end{cases}$$

该方程组的解也是 不唯一的。常见的一种三级三阶方法是

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 4K_2 + K_3),$$
$$K_1 = f(x_n, y_n),$$
$$K_2 = f\left(x_n + \frac{h}{2}, y_2 + \frac{h}{2}K_1\right),$$
$$K_3 = f(x_n + h, y_n - hK_1 + 2hK_2).$$

对于**L=4**的情形，可进行类似推导。最常用的四级四阶方法是如下经典**R-K**方法

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \qquad (8.2.6)$$
$$K_1 = f(x_n, y_n),$$
$$K_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right),$$
$$K_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2\right),$$
$$K_4 = f(x_n + h, y_n + hK_3).$$

为了分析经典R-K公式的计算量和计算精度，将四阶经典R-K公式（8.2.6）与一阶显式Euler公式（8.1.2）及二阶改进的Euler公式相比较。一般说来，公式的级数越大，计算右端项 f 的次数越多，计算量越大。在同样步长的情况下，Euler方法每步只计算一个函数值，而经典方法要计算4个函数值。四阶R-K法的计算量差不多是改进的Euler公式的2倍，是显式Euler公式的4倍。下面的例子中Euler方法用步长 h_1 ，二阶改进的Euler法用步长 $2h_1$，而四阶经典公式用步长 $4h_1$。这样，从 x_n 到 x_n+4h_1 三种方法都计算了4个函数制，计算量大体相当。

例8.3　考虑初值问题

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

其解析解为 $y(x)=1-e^{-x}$。分别用h=0.025的显式Euler方法，h=0.05改进Euler法和h=0.1的经典R-K方法计算到x=0.5。三种方法在x方向每前进0.1都要计算4个右端函数值，计算量相当。计算结果列于表8-3。从计算结果看，在工作量大致相同的情况下，还是经典方法比其他两种方法的结果好得多。在x=0.5处，三种方法的误差分别是 $3.8\times10^{-3},1.3\times10^{-4},2.8\times10^{-7}$ 。经典R-K法对多数好条件问题（$f_y<0$，参见下节单步法的稳定性），能获得好的效果。

表 8-3

x_n	Euler法 h=0.025	改进Euler法 h=0.05	经典R-K法 h=0.1	准确解 $y(x_n)$
0.1	0.096312	0.095123	0.09516250	0.09516258
0.2	0.183348	0.181193	0.18126910	0.18126925
0.3	0.262001	0.259085	0.25918158	0.25918178
0.4	0.333079	0.329563	0.32967971	0.32967995
0.5	0.397312	0.393337	0.39346906	0.39346934

在微分方程数值解法的实际计算中，有个如何选择步长的问题。因为单从每一步看，步长越小，截断误差越小。但随着步长的缩小，在一定求解范围内所要完成的步数就增加了。步数的增加不但引起计算量的增大，而且可能导致舍入误差的严重积累。

在选择步长时，我们需要衡量和检验计算结果的精度，并依据所获得的精度处理步长。下面以经典**R-K**方法为例进行说明。

从节点 x_n 出发，先以**h**为步长求出一个近似值 $y_{n+1}^{(h)}$，由于公式的局部截断误差为 $O(h^5)$，故有

$$y(x_{n+1}) - y_{n+1}^{(h)} \approx ch^5$$

然后将步长折半，既 **h/2** 为步长，从 x_n 跨两步到 x_{n+1}，再求得一个近似值 $h_{n+1}^{(h/2)}$，每跨一步的截断误差约为 $c(h/2)^5$，因此有

$$y(x_{n+1})-y_{n+1}^{(h/2)}\approx 2c\left(\frac{h}{2}\right)^5$$

比较上述二式，有

$$\frac{y(x_{n+1})-y_{n+1}^{(h/2)}}{y(x_{n+1})-y_{n+1}^{(h)}}\approx\frac{1}{16}。$$

由此易得下列事后估计式

$$y(x_{n+1})-y_{n+1}^{(h/2)}\approx\frac{1}{15}\left(y_{n+1}^{(h/2)}-y_{n+1}^{(h)}\right)。$$

这样，我们可以通过检查步长折半前后两次计算结果的偏差

$$\Delta=\left|y_{n+1}^{(h/2)}-y_{n+1}^{(h)}\right|$$

来判定所选的步长是否合适。

具体地说，对于给定的精度 ε ，将按两种情况处理。如果 $\Delta > \varepsilon$ ，我们反复将步长折半进行计算，直到 $\Delta < \varepsilon$ 为止，这时取最终得到的 $y_{n+1}^{(h/2)}$ 作为结果。如果 $\Delta < \varepsilon$，我们反复将步长加倍，直到 $\Delta > \varepsilon$ 为止，这时再将前一次步长折半的结果作为所要的结果。这种通过加倍或折半处理步长的方法称作变步长方法。虽然为了选择步长，每一步的计算量有所增加，但总体考虑是值得的。