第6章
常微分方程数值解法

绪论

在工程和科学计算中，所建立的各种常微分方程的初值或边值问题，除很少几类的特殊方程能给出解析解，绝大多数的方程是很难甚至不可能给出解析解的，其主要原因在于积分工具的局限性。因此，人们转向用数值方法去解常微分方程，并获得相当大的成功，讨论和研究常微分方程的数值解法是有重要意义的。

6.1 初值问题的Euler方法

设一阶常微分方程初值问题

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

记 $x_n = a + nh \ (n = 0,1,2,\ldots), h > 0$为步长，一般

假定为常数。该式的数值解是指通过某种方

去获得解$y(x)$在点x_n上的近似值y_n，即

$$y(x_n) \approx y_n \quad (n = 0,1,2,\ldots)$$

初值问题的Euler方法

为实现这一目标,Euler方法首先将微分算子离散化，并用x_n代替x_0,于是该式可离散为:

$$\frac{y(x_n+h)-y(x_n)}{h}\approx f(x_n,y(x_n))$$

以y_n表示$y(x_n)$的近似值，则有

$$y_{n+1}=y_n+hf(x_n,y_n)\quad (n=0,1,2,\ldots)\qquad (1)$$

这就是显式的Euler公式，它可以从y_0出发，逐次算出$y_1,y_2,y_3\ldots$。

初值问题的Euler方法

如果用x_{n+1}代替x_0，于是该式可离散为：

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

以y_n表示$y(x_n)$的近似值，则有

$$y_{n+1}=y_n+hf(x_{n+1},y_{n+1})\quad (n=0,1,2,\ldots)\qquad (2)$$

这就是隐式的Euler公式或向后Euler方法，它与显式的不同在于，它每算步要解函数方程(2)才能得到y_{n+1}。

初值问题的Euler方法

如果取以上两式的算术均值的结果，则得

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] \quad (n = 0,1,2,\dots)$$

称为梯形公式。

计算y_n时常用以下迭代式：

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n) \\ y_{n+1}^{(k+1)} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})] \quad (k = 0,1,2,\dots) \end{cases} \quad (3)$$

当$|y_{n+1}^{(k+1)} - y_n^{(k)}| < \varepsilon$ 时，取$y_{n+1} \approx y_{n+1}^{(k+1)}$

初值问题的Euler方法

定理6.1.1　设函数$f(x,y)$对变量y满足Lipschitz条件，L为Lipschitz常数。如果步长h满足$0 \le \frac{hL}{2} < 1$，即$h < \frac{2}{L}$时，则由（3）产生的序列$\{y_{n+1}^{(k)}\}(k = 0,1,2\ldots)$收敛。

初值问题的Euler方法

证明：由式（2）和（3）有

$$|y_{n+1}-y_{n+1}^{(k+1)}|=\frac{h}{2}|f(x_{n+1},y_{n+1})-f(x_{n+1},y_{n+1}^{(k)})|$$

$$\leq\frac{hL}{2}|y_{n+1}-y_{n+1}^{(k)}|$$

$$\cdots\cdots$$

$$\leq(\frac{hL}{2})^{k+1}|y_{n+1}-y_{n+1}^{(0)}|$$

由假设知 $\lim\limits_{k\to\infty}(\frac{hL}{2})^{k+1}=0$，故有 $\lim\limits_{k\to\infty}y_{n+1}^{(k+1)}=y_{n+1}$。

初值问题的Euler方法

对于(2)计算y_{n+1}，由于迭代工作量较大，一般只迭代一次构成一类预估-校正算法，即

$$\begin{cases} y_{n+1}^{(p)} = y_n + hf(x_n, y_n) \\ y_{n+1}^{(c)} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(p)})] \end{cases}$$

并取$y_{n+1} = y_{n+1}^{(c)}$。

初值问题的Euler方法

上式还常写成

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{2} f(k_1 + k_2) \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + h, y_n + k_1) \quad (n = 0,1,2,\ldots) \end{cases}$$

该式称为改进Euler方法,亦可写成

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

初值问题的Euler方法

例6.1.1　设初值问题

$$\begin{cases} \dfrac{dy}{dx} = y - \dfrac{2x}{y} \\ y(0) = 1 \end{cases}$$

试分别用Euler法和改进Euler法求解，并与精确解$y = \sqrt{1+2x}$进行比较。

初值问题的Euler方法

解：取$h=0.1$, 计算$x\in[0,1]$上结果，此时

Euler法：$y_{n+1}=y_n+0.1(y_n-\dfrac{2x_n}{y_n})\quad(n=0,1,2,\ldots)$

改进的Euler法：
$$\begin{cases} y_{n+1}=y_n+\dfrac{1}{2}(k_1+k_2) \\ k_1=0.1(y_n-\dfrac{2x_n}{y_n}) \\ k_2=0.1(y_n+k_1-\dfrac{2(x_n+0.1)}{y_n+k_1}) \quad (n=0,1,2,\ldots) \end{cases}$$

计算结果如下表所示：

x	Euler法y	改进的Euler法y	精确解
0	1.000000	1.000000	1.000000
0.1	1.000000	1.095909	1.095445
0.2	1.191818	1.184097	1.183216
0.3	1.277438	1.266201	1.264911
0.4	1.358213	1.343360	1.341641
0.5	1.435133	1.416402	1.414214
0.6	1.508966	1.485956	1.483240
0.7	1.580338	1.552514	1.549193
0.8	1.649783	1.616475	1.612452
0.9	1.717779	1.678166	1.673320
1.0	1.784770	1.737867	1.732051

误差概述

显式单步法一般形式为

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

而隐式单步法一般形式为

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

函数φ与$f(x, y)$有关，称为增量函数。

误差概述

定义6.1.1 从初值 $y(x_0) = y_0$ 出发，由单步法显式或隐式逐步计算，得 x_{n+1} 的值 y_{n+1}，则 $e_{n+1} = y(x_{n+1}) - y_{n+1}$ 称为在点 x_{n+1} 上的整体截断误差。如果第 n 步在点 x_n 的值计算没有误差，即 $y_n = y(x_n)$，由单步法计算出 $\tilde{y}_{n+1}$，则 $T_{n+1} = y(x_{n+1}) - \tilde{y}_{n+1}$，称为点 x_{n+1} 上的局部截断误差。

误差概述

定义6.1.2　如果给果给定的算法的截断

误差为

$$T_{n+1}=O(h^{p+1})$$

则称该算法具有p阶精度。如果

$$T_{n+1}=g(x_n,y(x_n))h^{p+1}+O(h^{p+2})$$

则非零项$g(x_n,y(x_n))h^{p+1}$称为为局部截断误差主

误差概述

一般说来，一个算法，局部截断误差阶 p 越大，则精度相对的越高。

对于显式Euler法：$T_{n+1}=\dfrac{h^2}{2}y''(x_n)+O(h^3)$

对于隐式Euler法：$T_{n+1}=-\dfrac{h^2}{2}y''(x_n)+O(h^3)$

对于梯形公式：$T_{n+1}=-\dfrac{h^3}{2}y'''(x_n)+O(h^4)$

数值稳定性分析

假设计算y_n时有一舍入误差ρ_n，则实际计算结果为$\tilde{y}_n = y_n + \rho_n$，如果通过某种数值方法又算得$\tilde{y}_{n+1} = y_{n+1} + \rho_{n+1}$，且

$$|\rho_{n+1}| \le |\rho_n|$$

则称该算法是数值稳定也成绝对稳定 如果算法的舍入误的舍入误差大，则，则称该算数值不稳定。

数值稳定性分析

- 定义若某数值算法的绝对稳定性区域包含hλ平面上的左半平面Re(hλ)＜0，则称该方法是A稳定的。
- 隐式Euler法是A稳定的。

6.2 Runge-Kutta方法

受改进的Euler方法启发，更一般算式设为

$$\begin{cases} y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2 \\ k_1 = hf(x_n, y_n) \qquad\qquad (n = 0,1,2,\ldots) \\ k_2 = hf(x_n + \alpha h, y_n + \beta k_1) \end{cases}$$

适当选择参数ω_1，ω_2，α，β，使局部截断误差

$T_{n+1} = y(x_{n+1}) - y_{n+1} = O(h^3)$,这里仍假定$y_n = y(x_n)$。

Runge-Kutta方法

由二元函数Taylor展开式:

$$k_2 = hf(x_n, y_n) + \alpha h^2 f_x'(x_n, y_n) + \beta h k_1 f_y'(x_n, y_n) + O(h^3)$$

$$= hy'(x_n) + h^2(\alpha f_x'(x_n, y_n) + \beta f(x_n, y_n) f_y'(x_n, y_n)) + O(h^3)$$

于是

$$y_{n+1} = y(x_n) + (\omega_1 + \omega_2) h y'(x_n) + [\alpha \omega_2 f_x'(x_n, y_n)$$

$$+ \beta \omega_2 f(x_n, y_n) f_y'(x_n, y_n)] h^2 + O(h^3)$$

Runge-Kutta方法

与Taylor展式相比较得

$$\begin{cases} \omega_1 + \omega_2 = 1 \\ \alpha\omega_2 = \dfrac{1}{2} \\ \beta\omega_2 = \dfrac{1}{2} \end{cases}$$

由于四个参数，三个程，因此有一个自由参数，即解答不唯。

(1) 取$\omega_1 = \frac{1}{2}$,可得$\omega_2 = \frac{1}{2}$,$\alpha = \beta = 1$,此时算式为

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

这是改进的Euler方法。

Runge-Kutta方法

(2) 取$\omega_1=0$,可得$\omega_2=1, \alpha=\beta=\dfrac{1}{2}$,此时算式为

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

这是二阶R-K方法

(3) 取$\omega_1=\frac{1}{4}$,可得$\omega_2=\frac{3}{4}$,$\alpha=\beta=\frac{2}{3}$,又有算式

$$
\begin{cases}
y_{n+1}=y_n+\frac{1}{4}(k_1+3k_2) \\
k_1=hf(x_n,y_n) \\
k_2=hf(x_n+\frac{2}{3}h,y_n+\frac{2}{3}k_1)
\end{cases}
$$

这也是二阶R - K方法。

四阶Runge-Kutta方法

与二阶R－K方法类似，解一阶常微分方程初值问题可设

$$\begin{cases} y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3 + \omega_4 k_4 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \alpha_2 h, y_n + \beta_{21} k_1) \\ k_3 = hf(x_n + \alpha_3 h, y_n + \beta_{31} k_1 + \beta_{32} k_2) \\ k_3 = hf(x_n + \alpha_4 h, y_n + \beta_{41} k_1 + \beta_{42} k_2 + \beta_{43} k_3) \end{cases} \quad (n = 0,1,2,\ldots)$$

约定$\alpha_2 = \beta_{21}$，$\alpha_3 = \beta_{31} + \beta_{32}$，$\alpha_4 = \beta_{41} + \beta_{42} + \beta_{43}$

四阶Runge-Kutta方法

类似以上推导过程，可求得经典四阶R－K算式

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 = hf(x_n, y_n) \\
k_2 = hf(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}k_1) \\
k_3 = hf(x_n + \dfrac{1}{2}h, y_n + \dfrac{1}{2}k_2) \\
k_3 = hf(x_n + h, y_n + k_3)
\end{cases}
$$

该公式结构整齐，但不是A稳定。

法的稳定性

以经典R - K法为例应用于实验方程$\frac{\mathrm{d}y}{\mathrm{d}x}=\lambda y$有

$$\begin{cases} k_1 = h\lambda y_n \\ k_2 = (h\lambda + \frac{1}{2}(h\lambda)^2)y_n \\ k_3 = (h\lambda + \frac{1}{2}(h\lambda)^2 + \frac{1}{4}(h\lambda)^3)y_n \\ k_4 = (h\lambda + (h\lambda)^2 + \frac{1}{2}(h\lambda)^3 + \frac{1}{4}(h\lambda)^4)y_n \end{cases}$$

R-K法的稳定性

于是

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

$$= [1 + (h\lambda) + \frac{(h\lambda)^2}{2!} + \frac{(h\lambda)^3}{3!} + \frac{(h\lambda)^4}{4!}]y_n$$

相应的误差为

$$\rho_{n+1} = [1 + (h\lambda) + \frac{(h\lambda)^2}{2!} + \frac{(h\lambda)^3}{3!} + \frac{(h\lambda)^4}{4!}]\rho_n$$

R-K法的稳定性

由稳定性要求$|\rho_{n+1}| \le |\rho_n|$，于是得稳定性区域为

$$|1+(h\lambda)+\frac{(h\lambda)^2}{2!}+\frac{(h\lambda)^3}{3!}+\frac{(h\lambda)^4}{4!}| \le 1$$

当λ为负实数时可得稳定性区间$-2.78 < \lambda h < 0$,

因此，当$\lambda = \frac{\partial f}{\partial y}$的绝对值较大时，步长$h$限制很大，即

h必须很小才能保证算法稳定性

隐式R-K法

显式R - K法的优点是从y_n可以直接计算出y_{n+1}，使用方便，

但从稳定性分析知，当$|\frac{\partial f}{\partial y}|$较大时，对步长$h$有较大的限制，

否则数值不稳定。

隐式格式的向后Euler法和梯形方法计算y_{n+1}需要多次

迭代求解是它们的缺点，但从稳定性分析知它们均是A稳

定的，即对步长h限制较小。因此，构造高阶隐式方法，既可以

精度提高又能保证数值稳定。

隐式R-K法

设二阶隐式R－K方法形式为：

$$\begin{cases} y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2 \\ k_1 = hf(x_n + \alpha_1 h, y_n + \beta_{11} k_1 + \beta_{12} k_2) \\ k_2 = hf(x_n + \alpha_2 h, y_n + \beta_{21} k_1 + \beta_{22} k_2) \end{cases}$$

约定

$$\begin{cases} \alpha_1 = \beta_{11} + \beta_{12} \\ \alpha_2 = \beta_{21} + \beta_{22} \end{cases}$$

隐式R-K法

将k_1，k_2在点(x_n, y_n)做二元函数Taylor展开求得的k_j，再反复带入上式，并与y_{n+1}在(x_n, y_n)的Taylor展式比较h的同次幂系数得：

$$\begin{cases} \omega_1 + \omega_2 = 1 \\ \omega_1\alpha_1 + \omega_2\alpha_2 = \dfrac{1}{2} \\ \omega_1\alpha_1^{\ 2} + \omega_2\alpha_2^{\ 2} = \dfrac{1}{3} \\ \omega_1(\alpha_1\beta_{11} + \alpha_2\beta_{12}) + \omega_2(\alpha_1\beta_{21} + \alpha_2\beta_{22}) = \dfrac{1}{6} \end{cases}$$

隐式R-K法

(1) 令$\alpha_1 = 0$，$\alpha_2 = \frac{2}{3}$，得二阶三级隐式R-K方法

$$\begin{cases} y_{n+1} = y_n + \frac{1}{4}(k_1 + 3k_2) \\ k_1 = hf(x_n, y_n - k_1 + k_2) \\ k_2 = hf(x_n + \frac{2}{3}h, y_n + \frac{2}{3}k_2) \end{cases}$$

隐式R-K法

(2) 令$\alpha_1=\dfrac{1}{2}-\dfrac{\sqrt{3}}{6}$，$\alpha_2=\dfrac{1}{2}+\dfrac{\sqrt{3}}{6}$得二阶四级隐式R-K方法

$$\begin{cases} y_{n+1}=y_n+\dfrac{1}{2}(k_1+k_2) \\ k_1=hf\left(x_n+\left(\dfrac{1}{2}-\dfrac{\sqrt{3}}{6}\right)h,\ y_n+\dfrac{k_1}{4}+\left(\dfrac{1}{4}-\dfrac{\sqrt{3}}{6}\right)k_2\right) \\ k_2=hf\left(x_n+\left(\dfrac{1}{2}+\dfrac{\sqrt{3}}{6}\right)h,\ y_n+\left(\dfrac{1}{4}+\dfrac{\sqrt{3}}{6}\right)k_1+\dfrac{1}{4}k_2\right) \end{cases}$$

6.3 线形多步法

- 单步法主要依据y_n的信息去计算y_{n+1}[redacted]性多步法是想依据y_n，y_{n-1}…，y_{n-r}[redacted]1)的信息去计算y_{n+1}。
- 考虑到线性组合较为方便，因此，线性多步法一般形式可设为

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \ldots + \alpha_r y_{n-r} + h(\beta_{-1} f_{n+1} + \beta_0 f_n + \ldots + \beta_r f_{n-r})$$

$$= \sum_{k=0}^{r} \alpha_k y_{n-k} + h \sum_{k=-1}^{r} \beta_h f_{n-k}$$

其中, $f_{n-k} = f(x_{n-k}, y_{n-k})(k = -1,0,1,\ldots,r)$.

当$\beta_{-1} = 0$时,称为显式否则称为隐式

基于数值积分的方法

初值问题

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

的等价积分方程形式为

$$y(x) = y(x_0) + \int_{x_0}^{x} f(t, y(t))dt$$

取$x = x_1$得

$$y(x_1) = y(x_0) + \int_{x_0}^{x_1} f(t, y(t))dt$$