2018-2019学年第一学期

计 算 方 法

第十三讲：常微分方程数值解法

第五章 §1-2

主讲人：张治国

zgzhang@szu.edu.cn

深圳大学 医学部 生物医学工程学院

SHENZHEN UNIVERSITY HEALTH SCIENCE CENTER SCHOOL OF BIOMEDICAL ENGINEERING

上节课回顾

- 为了避免使用高阶Newton-Cotes公式和提高求积的精度，通常采用复合求积法。
- 复合Simpson公式的复杂度居中，精度也令人满意，使用较普遍。
- 与数值积分类似，可以基于Lagrange插值发展插值型求导公式。

第四章回顾

- 第四章以插值多项式理论为基础，建立积分与微分的数值公式，因此，把建立的公式统称为插值型求积或求导公式。
- 基于等距节点的Newton-Cotes求积公式，可以设计出一系列自动取补偿的算法，其中以自动选择步长的Simpson公式最常用（逻辑简单、估计误差方便、收敛较快）。
- 其它常用数值积分微分公式还有Romberg求积算法、Gauss求积法、样条求导法等，它们有不同的特点和用途。

本章内容

第五章 常微分方程数值解法

§1 引言

§2 Runge-Kutta法

§3 线性多步法

§4 常微分方程数值解法的进一步讨论

本节课内容

第五章 常微分方程数值解法

§1 引言

1-1 基于数值微分的求解公式

1-2 截断误差

1-3 基于数值积分的求解公式

§2 Runge-Kutta法

2-1 Runge-Kutta法

2-2 四阶Runge-Kutta算法

§1 引言

- 微分方程定义
  1. 含有未知函数的导数（微分）的方程，称为微分方程
  2. 未知函数为一元函数的微分方程，称为常微分方程
  3. 未知函数为多元函数的微分方程，称为偏微分方程

- 问题：实际中需要求解常微分方程的初值问题

$$\begin{cases} y' = f(x, y) & a \le x \le b & (1.1) \\ y(a) = y_0 & & (1.2) \end{cases}$$

- 上述方程等价于积分方程

$$y(x) = y_0 + \int_a^x f(t, y(t))dt \qquad (1.3)$$

§ 1 引言

- 当 $f(x, y)$ 在矩形区域 $a \le x \le b,\ c \le y \le d$ 中连续，并且对变量 y 满足Lipschitz条件，即对任意 $x \in [a,b],\ y \in [c,d]$ 都有

$$| f(x, y_1) - f(x, y_2) | \le L | y_1 - y_2 |$$

其中 L 为常数，则初值问题(1.1)、(1.2)的解存在且唯一。

- 所谓常微分方程的数值解就是求 $y(x_k)$ 在区间 $[a, b]$ 中一系列离散点（或节点）

$$a < x_1 < x_2 < \cdots < x_n \le b$$

上 $y(x_k)$ 的近似值 $y_k (k = 1,2,\cdots,n)$，这些近似值就是(1.1)、(1.2)的数值解。

- 求解计算这些近似值的方法称为该方程数值方法。由数值方法计算所得的数值解统称为计算解。

§1 引言

- 在计算时，从初始条件 $y(a)=y_0$ 出发，先求 y_1，再从 y_0,y_1 求出 y_2，以此递推求出所有 $y_k(k=1,2,\cdots,n)$ 。这种顺着节点的排列顺序一步步地向前推进的求解方法，称为步进法。
  - 如果计算 y_k 时只利用了 y_{k-1}，这种方法称为单步法。
  - 如果计算 y_k 时利用了 $y_{k-1},y_{k-2},\cdots,y_{k-p}$，这种方法称为多步法。

§1 引言

- 下面重点介绍单步法。
- 将求解区间 $[a, b]$ 分成 n 等分，令

$$\frac{b-a}{n}=h,\quad x_i-x_{i-1}=h, i=1,2,\cdots,n,\quad x_0=a$$

即 $x_i=a+ih$ 。

- 因此数值解法就是求 $y(x_i)$ 的计算值 y_i， $i=1,2,\cdots,n$ 。

1-1 基于数值微分的求解公式

- 在(1.1)−(1.2)中，利用两点数值微分公式(6.5)（153页），得

$$[a,x_1]:\quad \frac{1}{h}(y_1-y_0)=y'(a)+\frac{h}{2}y''(\eta_0)$$

$$=y'(x_1)-\frac{h}{2}y''(\xi_0)$$

$$[x_1,x_2]:\quad \frac{1}{h}(y_2-y_1)=y'(x_1)+\frac{h}{2}y''(\eta_1)$$

$$=y'(x_2)-\frac{h}{2}y''(\xi_1)$$

$$\cdots\cdots$$

$$[x_j,x_{j+1}]:\quad \frac{1}{h}(y_{j+1}-y_j)=y'(x_j)+\frac{h}{2}y''(\eta_j)$$

$$=y'(x_{j+1})-\frac{h}{2}y''(\xi_j) \qquad (1.6)$$

$$\cdots\cdots$$

其中 $y_j=y(x_j),\ \ j=2,3,\cdots,n-1$。

1-1 基于数值微分的求解公式

- 由(1.6)式的前半部分的公式和(1.1)-(1.2)，得

$$\frac{1}{h}(y_{j+1}-y_j)=y'(x_j)+\frac{h}{2}y''(\eta_j)\approx f(x_j,y(x_j))$$

- 因此

$$\begin{cases} y_{j+1}=y_j+hf(x_j,y_j) & (1.7) \\ e_{j+1}(h)=\dfrac{h^2}{2}y''(\eta_j)\approx\dfrac{h^2}{2}y''(x_j) & (1.8) \end{cases}$$

- (1.7)称为前进Euler求解公式或Euler法。
- 注意：(1.7)中 $y_j=y(x_j)$, $y_{j+1}\approx y(x_{j+1})$。

1-1 基于数值微分的求解公式

- 后退Euler求解公式：由(1.6)后半部分公式得，

$$\begin{cases} y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}) & (1.9) \\ e_{j+1}(h) \approx -\dfrac{h^2}{2} y''(\xi_j) \approx -\dfrac{h^2}{2} y''(x_{j+1}) & (1.10) \end{cases}$$

其中 $y_j = y(x_j)$, $y_{j+1} \approx y(x_{j+1})$ 。

- 前进Euler求解公式(1.7)是关于 y_{j+1} 的一个直接计算公式，这类计算公式称为显式；后退Euler求解公式(1.9)的右端含有未知量 y_{j+1} ，这类计算公式称为隐式。

1-1 基于数值微分的求解公式

- 预测-校正系统（显式与隐式结合方法求解）

$$y_{j+1} = y_j + hf(x_j, y_j)$$ 前进Euler求解公式 (1.7)

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1})$$ 后退Euler求解公式 (1.8)

$$\begin{cases} \bar{y}_{j+1} = y_j + hf(x_j, y_j) & \text{预测值} \\ y_{j+1} = y_j + hf(x_{j+1}, \bar{y}_{j+1}) & \text{矫正值} \\ y(x_0) = y_0, \quad j = 0,1,2,\ldots \end{cases} \qquad (1.12)$$

1-1 基于数值微分的求解公式

- 例1 取 $h=0.1$，求解初值问题

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

- 解：已知 $x_0 = a = 0,\ y_0 = 1,\ n = 1/h = 10$,

$$f(x,y) = y - 2x/y$$

1-1 基于数值微分的求解公式

- 利用Euler法求解，公式为

$$y_{j+1} = y_j + hf(x_j, y_j) = y_j + h\left(y_j - \frac{2x_j}{y_j}\right), \quad j = 0,1,\cdots,9$$

- 因此，

$$y_1 = y_0 + hf(x_0, y_0) = y_0 + h\left(y_0 - \frac{2x_0}{y_0}\right) = 1.1000$$

$$y_2 = y_1 + hf(x_1, y_1) = y_1 + h\left(y_1 - \frac{2x_1}{y_1}\right) \approx 1.1918$$

……

$$y_{10} = y_9 + hf(x_9, y_9) = y_9 + h\left(y_9 - \frac{2x_9}{y_9}\right) \approx 1.7848$$

1-3 基于数值积分的求解公式

- 因为
$$\int_{x_{k-1}}^{x_k} f(x, y(x))dx = \int_{x_{k-1}}^{x_k} y'(x)dx = y(x_k) - y(x_{k-1})$$

  因此
$$y(x_k) = y(x_{k-1}) + \int_{x_{k-1}}^{x_k} f(x, y(x))dx \tag{1.13}$$

- 若 y_{k-1} 已知，则由(1.13)可求出 y_k。
- 令 $y(x_{k-1}) = y_{k-1}$ ，并利用矩形求积公式得
$$y(x_k) = y(x_{k-1}) + \int_{x_{k-1}}^{x_k} f(x, y(x))dx \approx y(x_{k-1}) + hf(x_{k-1}, y(x_{k-1}))$$
  或 $y_k = y_{k-1} + hf(x_{k-1}, y_{k-1})$ 。
- 上式即为Euler求解公式，又可称为矩形法。

1-3 基于数值积分的求解公式

一. 梯形公式

- 若在(1.13)中，利用梯形公式(1.10)（见123页）：

$$\int_{x_{k-1}}^{x_k} f(x, y(x))dx \approx \frac{h}{2}[f(x_{k-1}, y_{k-1}) + f(x_k, y_k)]$$

- 则 $$y(x_k) \approx y(x_{k-1}) + \frac{h}{2}[f(x_{k-1}, y_{k-1}) + f(x_k, y_k)]$$

- 令 $$y_k = y_{k-1} + \frac{h}{2}[f(x_{k-1}, y_{k-1}) + f(x_k, y_k)] \qquad (1.14)$$

- (1.14)称为梯形求解公式或梯形法。

1-3 基于数值积分的求解公式

- 梯形法为隐式，可将其显化，采用预测-校正系统

$$\begin{cases} \bar{y}_k = y_{k-1} + hf(x_{k-1}, y_{k-1}) \\ y_k = y_{k-1} + \dfrac{h}{2}[f(x_{k-1}, y_{k-1}) + f(x_k, \bar{y}_k)] \\ y_0 = x_0 \end{cases} \tag{1.15}$$

- (1.15)称为改进的Euler求解公式，或改进Euler法。

- 改进Euler法还可以写作

$$y_k = y_{k-1} + \frac{h}{2}[f(x_{k-1}, y_{k-1}) + f(x_k, y_{k-1} + hf(x_{k-1}, y_{k-1}))] \tag{1.16}$$

1-3 基于数值积分的求解公式

- 例3 用Euler法、梯形法以及改进Euler法求解

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

其中步长 $h = 0.1$，计算至 $x = 0.5$。

- 解：设 $f(x, y) = -y + x + 1,$

$$x_0 = a = 0, y_0 = 1, n = 5, b = 0.5$$

- 1) Euler法：计算公式为

$$y_k = y_{k-1} + h(x_{k-1} - y_{k-1} + 1) = hx_{k-1} + (1-h)y_{k-1} + h$$
$$= 0.1x_{k-1} + 0.9y_{k-1} + 0.1$$

1-3 基于数值积分的求解公式

- 2）梯形法：计算公式为

$$y_k = y_{k-1} + 0.5h[(x_{k-1} - y_{k-1} + 1) + (x_k - y_k + 1)]$$

解得

$$y_k = \frac{(2-h)y_{k-1} + h(x_{k-1} + x_k) + 2h}{2+h}$$

$$= \frac{1.9y_{k-1} + 0.1(x_{k-1} + x_k) + 0.2}{2.1}$$

- 3）改进Euler法：计算公式为

$$y_k = y_{k-1} + 0.5h[(x_{k-1} - y_{k-1} + 1)$$

$$+ (-y_{k-1} - h(x_{k-1} - y_{k-1} + 1) + x_k + 1]$$

$$= 0.905y_{k-1} + 0.045x_{k-1} + 0.05x_k + 0.095$$

- 计算结果见172页表5-1。可见梯形法和改进Euler法比Euler法精度高。

1-3 基于数值积分的求解公式

二. Simpson公式

$$
\begin{aligned}
&\int_{x_{k-1}}^{x_k} f(x, y(x))dx \\
&\approx \frac{h}{6}\left[f(x_{k-1}, y_{k-1}) + 4f\left(\frac{x_{k-1}+x_k}{2}, y\left(\frac{x_{k-1}+x_k}{2} \right) \right) + f(x_k, y(x_k)) \right] \\
&= \frac{h}{6}\left[f(x_{k-1}, y_{k-1}) + 4f\left(x_{k-1}+\frac{h}{2}, y\left(x_{k-1}+\frac{h}{2} \right) \right) + f(x_{k-1}+h, y(x_{k-1}+h)) \right]
\end{aligned}
$$

- 令

$$
\begin{aligned}
y_k = y_{k-1} + \frac{h}{6}\Bigg[& f(x_{k-1}, y_{k-1}) + 4f\left(x_{k-1}+\frac{h}{2}, \boxed{y\left(x_{k-1}+\frac{h}{2} \right)} \right) \\
& + f(x_{k-1}+h, \boxed{y(x_{k-1}+h)}) \Bigg]
\end{aligned}
\tag{1.17}
$$

- 上式是隐式，不便计算，需要显化。

§ 2 Runge-Kutta法

- 将改进Euler公式(1.15)改写为

$$\begin{cases} y_n = y_{n-1} + \dfrac{h}{2}[K_1 + K_2] \\ K_1 = f(x_{n-1}, y_{n-1}) \\ K_2 = f(x_n, y_{n-1} + hK_1) \\ y_0 = x_0 \end{cases} \tag{2.1}$$

- 也称为二阶Runge-Kutta法，简记为二阶R-K法。

- 利用不同的显化方法，对Simpson公式(1.17)进行各种显化，可得到其它R-K法。

2-1 Runge-Kutta法

- 由拉格朗日微分中值定理，

$$y'(\xi_{n-1}) = \{y(x_{n-1}+h) - y(x_{n-1})\}/h$$

- 故

$$y(x_{n-1}+h) = y(x_{n-1}) + hy'(\xi_{n-1})$$

- 因此，利用 x_{n-1}代替 ξ_{n-1}，得

$$y(x_{n-1}+h) \approx y(x_{n-1}) + hy'(x_{n-1}) = y(x_{n-1}) + hf(x_{n-1}, y(x_{n-1}))$$

- 令 $y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$，即为Euler公式(1.7)。

2-1 Runge-Kutta法

- 直观上，当 h 越小，x_{n-1}与 ξ_{n-1}越接近。故

$$y(x_{n-1}+\frac{h}{2}) \approx y_{n-\frac{1}{2}} = y_{n-1} + \frac{h}{2} f(x_{n-1}, y_{n-1}) \qquad (2.2)$$

- 又 (参见175页图5-1)

$$\begin{aligned} y'(\xi_{n-1}) &= \overline{HD} \approx \overline{FC} + \overline{EC} \\ &= y'(x_{n-1}+0.5h) + y'(x_{n-1}+0.5h) - y'(x_{n-1}) \\ &= 2y'(x_{n-1}+0.5h) - y'(x_{n-1}) \\ &= 2f(x_{n-1}+0.5h, y(x_{n-1}+0.5h)) - f(x_{n-1}, y(x_{n-1})) \end{aligned}$$

- 因此

$$\begin{aligned} y(x_{n-1}+h) &= y(x_{n-1}) + hy'(\xi_{n-1}) \\ &= y_{n-1} + hy'(\xi_{n-1}) \\ &= y_{n-1} + h[2f(x_{n-1}+0.5h, y_{n-0.5}) - f(x_{n-1}, y_{n-1})] \end{aligned} \qquad (2.3)$$

2-1 Runge-Kutta法

$$y_k = y_{k-1} + \frac{h}{6}\left[f(x_{k-1}, y_{k-1}) + 4f\left(x_{k-1} + \frac{h}{2}, y\left(x_{k-1} + \frac{h}{2} \right) \right) \right.$$

$$\left. + f(x_{k-1} + h, y(x_{k-1} + h)) \right] \quad (1.17)$$

- 利用(2.2)和(2.3)对(1.17)进行显化，得

$$\begin{cases} y_n = y_{n-1} + \dfrac{h}{6}[K_1 + 4K_2 + K_3] \\ K_1 = f(x_{n-1}, y_{n-1}) \\ K_2 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_1) \\ K_3 = f(x_{n-1} + h, y_{n-1} + h(2K_2 - K_1)) \end{cases} \quad (2.4)$$

- 称为三阶Runge-Kutta法或三阶R-K法。

2-1 Runge-Kutta法

- 为了再提高精度，若用

$$
\begin{cases}
K_1 = f(x_{n-1}, y_{n-1}) \\
K_2 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_1), & y(x_{n-1} + 0.5h) \approx y_{n-1} + 0.5K_1 \\
K_3 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_2), & y(x_{n-1} + 0.5h) \approx y_{n-1} + 0.5K_2 \\
K_4 = f(x_{n-1} + h, y_{n-1} + hK_3), & y(x_{n-1} + h) \approx y_{n-1} + hK_3
\end{cases}
$$

代入(1.17)式中的右端，得

$$
y_k = y_{k-1} + \frac{h}{6}\left[f(x_{k-1}, y_{k-1}) + 4f\left(x_{k-1} + \frac{h}{2}, y\left(x_{k-1} + \frac{h}{2} \right) \right) \right.
$$

$$
\left. + f(x_{k-1} + h, y(x_{k-1} + h)) \right]
$$

$$
\approx y_{n-1} + \frac{h}{6}[K_1 + 2K_2 + 2K_3 + K_4]
$$

2-1 Runge-Kutta法

- 最终可得如下求解公式：

$$\begin{cases} y_n = y_{n-1} + \dfrac{h}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right) \\ K_1 = f(x_{n-1}, y_{n-1}) \\ K_2 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_1), \\ K_3 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_2), \\ K_4 = f(x_{n-1} + h, y_{n-1} + hK_3), \end{cases} \tag{2.6}$$

以上称为四阶Runge-Kutta法或四阶R-K法，也称经典R-K法。

2-1 Runge-Kutta法

- 例：使用三阶、四阶R-K法计算初值问题：

$$\begin{cases} y' = y^2, x \in [0, 0.5] \\ y(0) = 1 \end{cases}$$

的部分计算解 y_1，y_2，y_3，其中 $h = 0.1$。

- 解：(1) 三阶R-K法

$$\begin{cases} y_n = y_{n-1} + \dfrac{h}{6}(K_1 + 4K_2 + K_3) \\ K_1 = f(x_{n-1}, y_{n-1}) = y_{n-1}^2 \\ K_2 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_1) = (y_{n-1} + 0.5hy_{n-1}^2)^2 \\ K_3 = f(x_{n-1} + h, y_{n-1} + h(2K_2 - K_1)) = \left(y_{n-1} + h(2K_2 - K_1)\right)^2 \end{cases}$$

2-1 Runge-Kutta法

- 当 $n=1$

$$\begin{cases} K_1 = f(x_0, y_0) = y_0^2 = 1 \\ K_2 = f(x_0+0.5h, y_0+0.5hK_1) = (y_0+0.5hy_0^2)^2 = 1.102500 \\ K_3 = f(x_0+h, y_0+h(2K_2-K_1)) = \left(y_{n-1}+h(2K_2-K_1)\right)^2 \approx 1.255520 \\ y_1 = y_0 + \dfrac{h}{6}(K_1+4K_2+K_3) \approx 1.111092 \end{cases}$$

- 当 $n=2$

$$\begin{cases} K_1 = f(x_1, y_1) = y_1^2 = 1.111092^2 \approx 1.234525 \\ K_2 = f(x_1+0.5h, y_1+0.5hK_1) = (y_1+0.5hy_1^2)^2 = 1.375502 \\ K_3 = f(x_1+h, y_1+h(2K_2-K_1)) = \left(y_1+h(2K_2-K_1)\right)^2 \approx 1.594512 \\ y_2 = y_1 + \dfrac{h}{6}(K_1+4K_2+K_3) \approx 1.249943 \end{cases}$$

- 当 $n=3$：略

2-1 Runge-Kutta法

- (2) 四阶R-K法

$$\begin{cases} K_1 = f(x_{n-1}, y_{n-1}) = y_{n-1}^2 \\ K_2 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_1) = (y_{n-1} + 0.5hK_1)^2 \\ K_3 = f(x_{n-1} + 0.5h, y_{n-1} + 0.5hK_2) = (y_{n-1} + 0.5hK_2)^2 \\ K_4 = f(x_{n-1} + h, y_{n-1} + hK_3) = (y_{n-1} + hK_3)^2 \\ y_n = y_{n-1} + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

- 当 $n = 1$

$$\begin{cases} K_1 = f(x_0, y_0) = y_0^2 = 1.000000 \\ K_2 = f(x_0 + 0.5h, y_0 + 0.5hK_1) = (y_0 + 0.5hK_1)^2 = 1.102500 \\ K_3 = f(x_0 + 0.5h, y_0 + 0.5hK_2) = (y_0 + 0.5hK_2)^2 \approx 1.113289 \\ K_4 = f(x_0 + h, y_0 + hK_3) = (y_0 + hK_3)^2 \approx 1.235052 \\ y_1 = y_0 + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \approx 1.111111 \end{cases}$$

2-1 Runge-Kutta法

- 当 $n=2$

$$\begin{cases} K_1 = f(x_1, y_1) = y_1^2 = 1.234568 \\ K_2 = f(x_1 + 0.5h, y_1 + 0.5hK_1) = (y_1 + 0.5hK_1)^2 = 1.375551 \\ K_3 = f(x_1 + 0.5h, y_1 + 0.5hK_2) = (y_1 + 0.5hK_2)^2 \approx 1.392138 \\ K_4 = f(x_1 + h, y_1 + hK_3) = (y_1 + hK_3)^2 \approx 1.563313 \\ y_2 = y_1 + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \approx 1.249999 \end{cases}$$

- 当 $n=3$

$$\begin{cases} K_1 = f(x_2, y_2) = y_2^2 = 1.562498 \\ K_2 = f(x_2 + 0.5h, y_2 + 0.5hK_1) = (y_2 + 0.5hK_1)^2 = 1.763913 \\ K_3 = f(x_2 + 0.5h, y_2 + 0.5hK_2) = (y_2 + 0.5hK_2)^2 \approx 1.790766 \\ K_4 = f(x_2 + h, y_2 + hK_3) = (y_2 + hK_3)^2 \approx 2.042258 \\ y_3 = y_2 + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \approx 1.428568 \end{cases}$$

本节课/本章小结

- 简单介绍了基于微积分数值方法的常微分方程数值解法。

- 实际计算中，当函数 $f(x, y)$ 不复杂时可采用单步法，其中使用较多的是四阶R-K法（稳定、精度高、步长易调，但计算量大且要求函数光滑）。

- 当函数 $f(x, y)$ 较复杂时，建议采用多步法（未介绍）。

作业

习题五：6（计算到4位小数，计算至 $x = 0.5$）

作业上交日期：2018年12月18日

下节课内容

第六章 逐次逼近法

§1 基本概念

§2 解线性方程组的迭代法