

# 数值计算方法

## 常微分方程的数值解法

---

张晓平

2019 年 12 月 2 日

武汉大学数学与统计学院

# Table of contents

1. 简介
2. 欧拉方法
3. 龙格 - 库塔 (Runge-Kutta) 方法

## 简介

1. 常微分方程的求解问题在实践中经常遇到，但我们只知道一些特殊类型的常微分方程的解析解。
2. 在科学与工程问题中遇到的常微分方程往往很复杂，许多情况下不可能求出解的表达式。
3. 很多实际问题中，并不需要方程解的表达式，而仅仅需要获得解在若干点上的近似值即可

因此，研究常微分方程的数值解法就很有必要。

## 问题

本章讨论一阶常微分方程初值问题

$$\begin{cases} \frac{dy}{dx} = f(x, y), & x > x_0 \\ y(x_0) = y_0 \end{cases} \quad (1)$$

的数值解法。

## 问题

本章讨论一阶常微分方程初值问题

$$\begin{cases} \frac{dy}{dx} = f(x, y), & x > x_0 \\ y(x_0) = y_0 \end{cases} \quad (1)$$

的数值解法。

理论上,  $f(x, y)$  对于  $y$  满足 Lipschitz 条件:

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

则以上初值问题存在惟一解。

所谓数值解法，就是寻找  $y(x)$  在一系列离散节点

$$a \leq x_0 < x_1 < \cdots < x_n < \cdots \leq b$$

上的近似值  $y_0, y_1, \cdots, y_n, \cdots$ ，其相邻两个节点的距离  $h_n = x_{n+1} - x_n$  称为步长，我们总假定节点等距，即  $h_n \equiv h$ ，此时

$$x_n = x_0 + nh, \quad n = 0, 1, 2, \cdots$$

此时节点  $x_n$  对应的函数值为

$$y(x_n) = y(x_0 + nh), \quad n = 0, 1, 2, \cdots$$

## 欧拉方法



**欧拉方法**

**欧拉公式**

# 欧拉公式

在方程 (1) 中，用向前差商代替导数，即

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

有

$$y(x_{n+1}) \approx y(x_n) + hf(x_n, y(x_n)),$$

再用  $y_n$  近似代替  $y(x_n)$ ，便导出**显式欧拉公式**

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n), \\ y_0 = y(x_0) \end{cases} \quad n = 0, 1, 2, \dots, \quad (2)$$

# 欧拉公式

在方程 (1) 中，用向后差商代替导数，即

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

便可导出隐式欧拉公式

$$\begin{cases} y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \\ y_0 = y(x_0) \end{cases} \quad n = 0, 1, 2, \dots, \quad (3)$$

# 欧拉公式

在方程 (1) 中，用向后差商代替导数，即

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

便可导出**隐式欧拉公式**

$$\begin{cases} y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \\ y_0 = y(x_0) \end{cases} \quad n = 0, 1, 2, \dots, \quad (3)$$

这类隐式格式的计算远比显式格式困难！

# 欧拉公式

在方程 (1) 中，用中心差商代替导数，即

$$y'(x_n) \approx \frac{y(x_{n+1}) - y(x_{n-1}))}{2h}$$

便可导出**两点欧拉公式**

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n = 0, 1, 2, \dots, \quad (4)$$

# 欧拉公式

在方程 (1) 中，用中心差商代替导数，即

$$y'(x_n) \approx \frac{y(x_{n+1}) - y(x_{n-1}))}{2h}$$

便可导出**两点欧拉公式**

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n = 0, 1, 2, \dots, \quad (4)$$

在计算  $y_{n+1}$  时，需利用前两步的信息  $y_n, y_{n-1}$ 。

# 欧拉方法

## 欧拉预估 - 校正方法

## 欧拉预估 - 校正方法

对方程  $y' = f(x, y)$  的两端从  $x_n$  到  $x_{n+1}$  积分, 得

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx.$$

利用梯形公式计算积分得

$$y(x_{n+1}) \approx y(x_n) + \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

再用  $y_n$  代替  $y(x_n)$  的近似值, 便可导出**梯形公式**

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



## 欧拉预估 - 校正方法

对方程  $y' = f(x, y)$  的两端从  $x_n$  到  $x_{n+1}$  积分, 得

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx.$$

利用梯形公式计算积分得

$$y(x_{n+1}) \approx y(x_n) + \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

再用  $y_n$  代替  $y(x_n)$  的近似值, 便可导出**梯形公式**

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

梯形公式可视为显示欧拉公式与隐式欧拉公式的算术平均, 它仍为隐式, 不便于直接计算。

## 欧拉预估 - 校正方法

实际计算中，可将欧拉公式与梯形公式相结合，先由显式欧拉求得一个初步的近似值，记为  $y_{n+1}^*$ ，称之为**预报值**；再将该值代入梯形公式算得  $y_{n+1}$ ，这一步称为**校正**。流程如下

$$\left\{ \begin{array}{ll} \text{预估} & y_{n+1}^* = y_n + hf(x_n, y_n), \\ \text{校正} & y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1}^*)] \end{array} \right. \quad n = 0, 1, 2, \dots \quad (5)$$

该公式称为**欧拉预估 - 校正公式或改进的欧拉公式**，它是一种显示公式，是对隐式梯形公式的改进，可以直接计算。

为便于上机编程，可将上述步骤改为

$$\begin{cases} y_p = y_n + hf(x_n, y_n), \\ y_c = y_n + hf(x_{n+1}, y_p), \\ y_{n+1} = \frac{1}{2}(y_p + y_c) \end{cases}$$

### 例

利用欧拉公式和预估 - 校正公式求初值问题

$$\begin{cases} y' = y - \frac{2x}{y} \\ y(0) = 1 \end{cases}$$

在  $[0, 1]$  上的数值解 (取  $h = 0.1$ ), 并与精确解  $y = \sqrt{2x+1}$  进行比较。

### 例

利用欧拉公式和预估 - 校正公式求初值问题

$$\begin{cases} y' = y - \frac{2x}{y} \\ y(0) = 1 \end{cases}$$

在  $[0, 1]$  上的数值解 (取  $h = 0.1$ ), 并与精确解  $y = \sqrt{2x+1}$  进行比较。

### 解

- 欧拉公式

$$\begin{cases} y_{n+1} = y_n + h \left( y_n - \frac{2x_n}{y_n} \right) \\ y_0 = 1, \quad h = 0.1 \end{cases} \quad n = 0, 1, 2, \dots, 9$$

解

- 预估 - 校正公式

$$\begin{cases} y_{n+1}^* = y_n + h \left( y_n - \frac{2x_n}{y_n} \right) \\ y_{n+1} = y_n + \frac{h}{2} \left( y_n - \frac{2x_n}{y_n} + y_{n+1}^* - \frac{2x_{n+1}}{y_{n+1}} \right) \\ y_0 = 1, \quad h = 0.1 \end{cases} \quad n = 0, 1, 2, \dots, 9$$

表 1: 计算结果

$x_n$	欧拉公式 $y_n$	预估 - 校正公式 $y_n$	精确解 $y(x_n) = \sqrt{2x_n + 1}$
0.0	1	1	1
0.1	1.1	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	1.784771	1.737867	1.732051

## 欧拉预估 - 校正方法

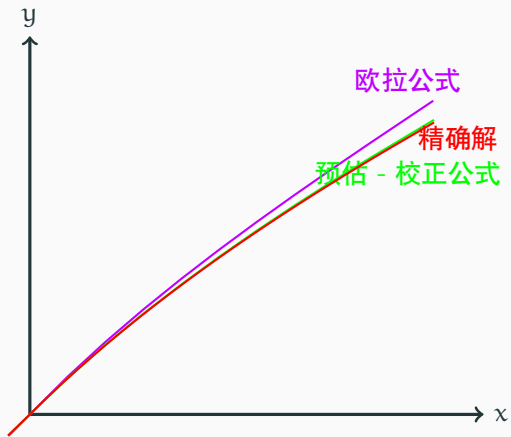


图 1: 计算结果



# 欧拉方法

## 欧拉方法的误差估计

# 欧拉方法的误差估计

## 定义

设  $y_n$  为精确解, 即  $y_n = y(x_n)$ , 在此前提下, 用某种数值方法计算  $y_{n+1}$  的误差  $R_n = y(x_{n+1}) - y_{n+1}$  称为该数值方法计算  $y_{n+1}$  的局部截断误差。

# 欧拉方法的误差估计

## 定义

设  $y_n$  为精确解, 即  $y_n = y(x_n)$ , 在此前提下, 用某种数值方法计算  $y_{n+1}$  的误差  $R_n = y(x_{n+1}) - y_{n+1}$  称为该数值方法计算  $y_{n+1}$  的局部截断误差。

## 定义

若某一数值方法的局部截断误差为  $R_n = O(h^{p+1})$ ,  $p$  为正整数, 则称这种数值方法为  $p$  阶方法, 或者说该方法有  $p$  阶精度。

# 欧拉方法的误差估计

## 定理

显式欧拉方法的局部截断误差为  $O(h^2)$ ，它为一阶方法。

# 欧拉方法的误差估计

## 定理

显式欧拉方法的局部截断误差为  $O(h^2)$ ，它为一阶方法。

## 证明

由泰勒公式

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

# 欧拉方法的误差估计

## 定理

显式欧拉方法的局部截断误差为  $O(h^2)$ ，它为一阶方法。

## 证明

由泰勒公式

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

对于显式欧拉公式，

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

# 欧拉方法的误差估计

## 定理

显式欧拉方法的局部截断误差为  $O(h^2)$ ，它为一阶方法。

## 证明

由泰勒公式

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

对于显式欧拉公式，

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

则其局部截断误差为

$$y(x_{n+1}) - y_{n+1} = \frac{h^2}{2!} y''(x_n) + \cdots = O(h^2)$$

## 定理

梯形公式为二阶方法。



# 欧拉方法的误差估计

## 定理

梯形公式为二阶方法。

## 证明

由梯形求积公式的误差知

$$|R_T(f)| \leq \frac{h^3}{12} \max |y''(x)|$$

其局部截断误差为  $O(h^3)$

# 欧拉方法的误差估计

## 定理

预估 - 校正公式为二阶方法。

# 欧拉方法的误差估计

## 定理

预估 - 校正公式为二阶方法。

## 证明

预估 - 校正公式可改写为

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

# 欧拉方法的误差估计

## 定理

预估 - 校正公式为二阶方法。

## 证明

预估 - 校正公式可改写为

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

在  $y_n = y(x_n)$  的前提下,

$$k_1 = hf(x_n, y_n) = hy'(x_n),$$

$$\begin{aligned} k_2 &= hf(x_n + h, y(x_n) + k_1) \\ &= h[f(x_n, y(x_n)) + hf_x(x_n, y(x_n)) + k_1 f_y(x_n, y(x_n)) + O(h^2)] \\ &= hf(x_n, y(x_n)) + h^2[f_x(x_n, y(x_n)) + f(x_n, y(x_n))f_y(x_n, y(x_n)) + O(h)] \\ &= hy'(x_n) + h^2 y''(x_n) + O(h^3) \end{aligned}$$

# 欧拉方法的误差估计

## 证明

将  $k_1, k_2$  代入原方程得,

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

与泰勒公式比较, 其截断误差为

$$y(x_{n+1}) - y_{n+1} = \frac{h^3}{3!}y'''(x_n) + \cdots$$

## 龙格 - 库塔 (Runge-Kutta) 方法

# 龙格 - 库塔 (Runge-Kutta) 方法

## 龙格 - 库塔方法的基本思想

## 龙格 - 库塔方法的基本思想

考察差商  $\frac{y(x_{n+1}) - y(x_n)}{h}$ , 由微分中值定理, 存在  $\xi$ , 使得

$$\frac{y(x_{n+1}) - y(x_n)}{h} = y'(\xi), \quad \xi \in (x_n, x_{n+1}).$$

由方程

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

得

$$y(x_{n+1}) = y(x_n) + hf(\xi, y(\xi)),$$

称  $k^* = f(\xi, y(\xi))$  为  $[x_n, x_{n+1}]$  的平均斜率。



只要对平均斜率提供一种近似算法，便相应导出一种计算格式。

只要对平均斜率提供一种近似算法，便相应导出一种计算格式。

## 1 显式欧拉公式

以  $k_1 = f(x_n, y_n)$  作为  $k^*$  的近似

只要对平均斜率提供一种近似算法，便相应导出一种计算格式。

## 1 显式欧拉公式

以  $k_1 = f(x_n, y_n)$  作为  $k^*$  的近似

## 2 欧拉预估 - 校正公式

以  $x_n$  与  $x_{n+1}$  两个点的斜率值  $k_1$  和  $k_2$  取算术平均作为  $k^*$  的近似

# 龙格 - 库塔方法的基本思想

## 龙格 - 库塔方法的基本思想

设法在  $[x_n, x_{n+1}]$  内多预报几个点的斜率值, 然后将它们加权平均作为  $k^*$  的近似, 则有可能构造出更高精度的计算格式。

# 龙格 - 库塔 (Runge-Kutta) 方法

## 二阶龙格 - 库塔方法

推广欧拉预估 - 校正方法，考察  $[x_n, x_{n+1}]$  内任一点

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

用  $x_n$  与  $x_{n+p}$  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

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

## 二阶龙格 - 库塔方法

类似于欧拉预估 - 校正，取

$$k_1 = f(x_n, y_n), \quad y_{n+p} = y_n + phk_1, \quad k_2 = f(x_{n+p}, y_{n+p})$$

便得如下格式

$$\begin{cases} y_{n+1} = y_n + h[(1 - \lambda)k_1 + \lambda k_2], \\ k_1 = f(x_n, y_n), \\ k_2 = f(x_n + ph, y_n + phk_1) \end{cases} \quad (6)$$

## 二阶龙格 - 库塔方法

类似于欧拉预估 - 校正，取

$$k_1 = f(x_n, y_n), \quad y_{n+p} = y_n + phk_1, \quad k_2 = f(x_{n+p}, y_{n+p})$$

便得如下格式

$$\begin{cases} y_{n+1} = y_n + h[(1 - \lambda)k_1 + \lambda k_2], \\ k_1 = f(x_n, y_n), \\ k_2 = f(x_n + ph, y_n + phk_1) \end{cases} \quad (6)$$

希望适当选取  $\lambda$ ,  $p$ , 使上述格式具有更高精度。



## 二阶龙格 - 库塔方法

设  $y_n = y(x_n)$ ，分别将  $k_1$  和  $k_2$  泰勒展开，

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

$$\begin{aligned} k_2 &= f(x_n + h, y(x_n) + phk_1) \\ &= f(x_n, y(x_n)) + phf_x(x_n, y(x_n)) + phk_1f_y(x_n, y(x_n)) + O(h^2) \\ &= f(x_n, y(x_n)) + ph[f_x(x_n, y(x_n)) + f(x_n, y(x_n))f_y(x_n, y(x_n)) + O(h)] \\ &= y'(x_n) + phy''(x_n) + O(h^2) \end{aligned}$$

代入计算格式 (6) 得

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

与泰勒公式

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

要使计算格式具有二阶精度，须使  $\lambda p = \frac{1}{2}$ 。

## 二阶龙格 - 库塔方法

把满足  $\lambda p = \frac{1}{2}$  的公式 (6) 统称为二阶龙格 - 库塔公式。

## 二阶龙格 - 库塔方法

把满足  $\lambda p = \frac{1}{2}$  的公式 (6) 统称为**二阶龙格 - 库塔公式**。

1 当  $p = 1, \lambda = \frac{1}{2}$  时, (6) 就是欧拉预估 - 校正公式

2 当  $p = \frac{1}{2}, \lambda = 1$  时, (6) 形式为

$$\begin{cases} y_{n+1} = y_n + hk_2, \\ k_1 = f(x_n, y_n), \\ k_2 = f(x_{n+\frac{1}{2}}, y_n + \frac{h}{2}k_1). \end{cases}$$

该公式可看作是用中点斜率值  $k_2$  作为平均斜率  $k^*$  的近似, 它也可称为**中点公式**, 具有二阶精度。

# 龙格 - 库塔 (Runge-Kutta) 方法

高阶龙格 - 库塔公式

常用的三阶龙格 - 库塔公式为

$$\left\{ \begin{array}{l} y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3), \\ k_1 = f(x_n, y_n), \\ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ k_3 = f(x_n + h, y_n + h(-k_1 + 2k_2)) \end{array} \right. \quad (7)$$

经典的四阶龙格 - 库塔公式为

$$\left\{ \begin{array}{l} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 = f(x_n, y_n), \\ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right), \\ k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right), \\ k_4 = f(x_n + h, y_n + hk_3) \end{array} \right. \quad (8)$$

例

用经典四阶龙格 - 库塔公式求解初值问题

$$\begin{cases} y' = y - \frac{2x}{y}, \\ y(0) = 1 \end{cases}$$

在  $[0, 1]$  上的数值解 (取  $h = 0.2$ )。

解

$$\left\{ \begin{array}{l} k_1 = y_n - \frac{2x_n}{y_n}, \\ k_2 = y_n + 0.1k_1 - \frac{2(x_n + 0.1)}{y_n + 0.1k_1}, \\ k_3 = y_n + 0.1k_2 - \frac{2(x_n + 0.1)}{y_n + 0.1k_2}, \\ k_4 = y_n + 0.2k_3 - \frac{2(x_n + 0.2)}{y_n + 0.2k_3}, \\ y_{n+1} = y_n + \frac{0.1}{3}(k_1 + 2k_2 + 2k_3 + k_4), \end{array} \right.$$



表 2: 计算结果

$x_n$	0	0.2	0.4	0.6	0.8	1.0
$y_n$	1	1.18323	1.34167	1.48324	1.61251	1.73214
$y(x_n)$	1	1.18322	1.34164	1.48324	1.61245	1.73205

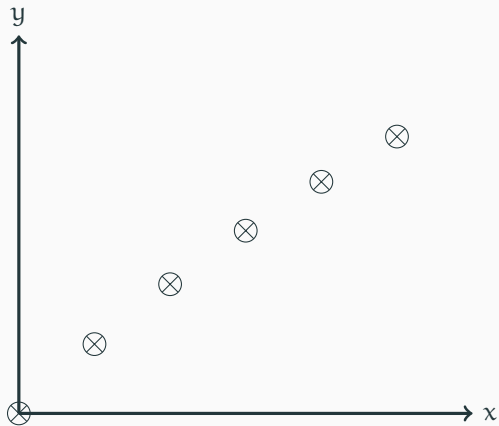


图 2: 计算结果