



第6章 常微分方程

主讲：纪庆革

中山大学数据科学与计算机学院

E-Mail: 1024180018@qq.com



第6章 常微分方程



实际中，很多问题的数学模型都是微分方程。我们可以研究它们的一些性质。但是，只有极少数特殊的方程有**解析解**。

绝大部分的微分方程没有**解析解**。



第6章 常微分方程



常微分方程作为微分方程的基本类型之一，在自然界与工程界有很**广泛的应用**。

很多问题的数学表述都可以归结为常微分方程的**定解问题**。

很多偏微分方程问题，也可以**化为常微分方程问题**来近似求解。

下面讨论常微分方程的**数值解法**。



本章内容

- 欧拉法
- 龙格-库塔法
- 线性多步法
- 方程组与高阶方程的数值解法
- 边值问题的数值解法



对于一个常微分方程：

$$y' = \frac{dy}{dx} = f(x, y) \quad , \quad x \in [a, b]$$

通常会有无穷个解。如：

$$\frac{dy}{dx} = \cos(x) \Rightarrow y = \sin(x) + a, \forall a \in R$$

因此，我们要加入一个限定条件。通常会在端点处给出，如下面的初值问题：

$$\begin{cases} \frac{dy}{dx} = f(x, y) \quad , \quad x \in [a, b] \\ y(a) = y_0 \end{cases} \quad (1)$$

为了使解存在唯一，一般要加限制条件在 f 上，要求 f 对 y 满足 **Lipschitz条件**：

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

定理 设函数 $f(x, y)$ 在区域

$$D: a \leq x \leq b, -\infty \leq y \leq +\infty$$

上连续, 且在区域 D 内满足李普希兹 (Lipschitz) 条件, 即存在正数 L , 使得对于 D 内任意两点 (x, y_2) 与 (x, y_1) , 恒有

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

则初值问题 (1) 的解 $y(x)$ 存在并且唯一。

常微分方程的解是一个函数，但是，计算机没有办法对函数进行运算。因此，常微分方程的数值解并不是求函数的近似，而是求解函数在某些节点的近似值。

例：我们对区间做等距分割： $x_i = h i$ ， $h = (b - a) / m$
设解函数在节点的近似为 $\{y_i\}$ ，则：

$$\left. \frac{dy}{dx} \right|_{x=x_i} = f(x, y) \Big|_{x=x_i}$$

向前差商
公式

由数值微分公式，我们有

$$\frac{y_{i+1} - y_i}{h} \approx f(x_i, y_i)$$

$$y_{i+1} = y_i + h f(x_i, y_i)$$

可以看到，给出初值，就可以用上式求出所有的 $\{y_i\}$

这种方法，称为数值离散方法。求未知函数 y 在这些点（一系列离散点列）上的值的近似。基本步骤如下：

① 对区间作分割： $\Delta_I : a = x_0 < x_1 < \cdots < x_n = b$
求 $y(x)$ 在 x_i 上的近似值 y_i 。 $\{y_i\}$ 称为分割 Δ_I 上的格点函数，我们的目的，就是求这个格点函数。

② 由微分方程出发，建立求格点函数的差分方程。
这个方程应该满足：**A、解存在唯一；B、稳定，收敛；C、相容**

③ 解差分方程，求出格点函数

数值方法，主要研究步骤 ②，即如何建立差分方程，并研究差分方程的性质。

为了考察数值方法提供的数值解，是否有实用价值，
需要知道如下几个结论：

① 步长充分小时，所得到的数值解能否逼近问题的真解；即收敛性问题

② 误差估计

③ 产生的舍入误差，在以后的各步计算中，是否会无限制地扩大；稳定性问题

6.1 Euler公式

做等距分割，利用数值微分代替导数项，建立差分方程。

$$\Delta_I : x_i = \frac{b-a}{m} i$$

称为**局部截断误差**。显然，这个误差在逐步计算过程中会传播，积累。因此还要估计这种积累

1、向前差商公式

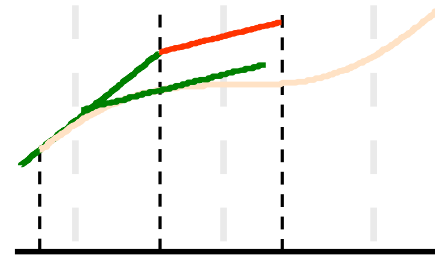
$$\frac{y(x_{n+1}) - y(x_n)}{h} = y'(x_n) + \frac{h}{2} y''(\xi_n)$$

$$\frac{y(x_{n+1}) - y(x_n)}{h} = f(x_n, y(x_n)) + \frac{h}{2} y''(\xi_n)$$

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + \frac{h^2}{2} y''(\xi_n)$$

所以，可以构造**差分方程**

$$y_{n+1} = y_n + hf(x_n, y_n), \text{ 该公式是显式的}$$



定义 在假设 $y_i = y(x_i)$ ，即第 i 步计算是精确的前提下，考虑的截断误差 $R_i = y(x_{i+1}) - y_{i+1}$ 称为局部截断误差 (local truncation error)。


定义 若某算法的局部截断误差为 $O(h^{p+1})$ ，则称该算法具有 p 阶精度或是 p 阶的。

定义 由前面 k 个计算值 y_1, y_2, \dots, y_k 得到的 y_{k+1} 与微分方程的精确解之差 $R_i = y(x_{i+1}) - y_{i+1}$ 称为整体截断误差。




2、收敛性

考察局部误差的传播和积累


$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + \frac{h^2}{2} y''(\xi_n)$$





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

增量
函数



截断误差: $O(h^2)$



欧拉法具有 1 阶精度



3、**稳定性**—误差在以后各步的计算中不会无限制扩大。是格式对舍入误差的抑制作用。

定义 若一种数值方法在节点值 y_n 上存在大小为 δ 的扰动，对于以后各节点值 $y_m (m > n)$ 上产生的偏差均不超过 δ ，则该方法是**稳定的**，否则该初值问题是**不稳定的**。特别地，当 $t \rightarrow \infty$ 时， $y(t)$ 的偏差收敛到零，则称该初值问题是**渐进稳定的**。

注：

- 1) 若一个问题是渐进稳定的，其必然是稳定的；
- 2) 对于不稳定的常微分方程初值问题，初始数据的扰动将使当 $t \rightarrow \infty$ 时的结果**误差无穷大**。因此，为了保证数值求解的有效性，“常微分方程初值问题具有稳定性”非常重要。

■ 考虑下面模型问题的稳定性:

$$\begin{cases} y' = \lambda y, t \geq t_0 \\ y(t_0) = y_0 \end{cases}$$

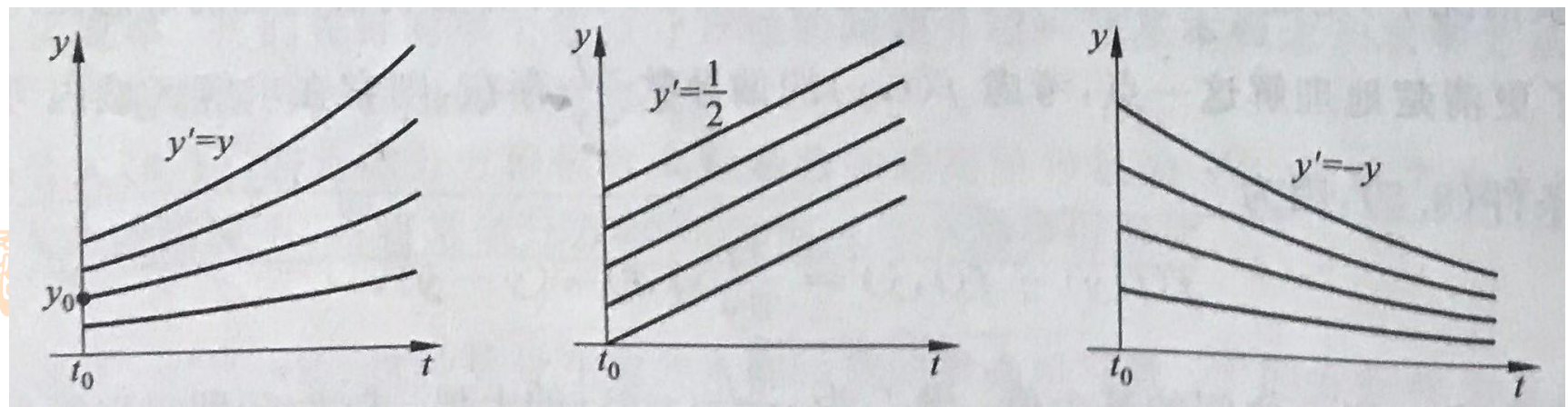
其准确解为 $y(t) = y_0 e^{\lambda(t-t_0)}$.

假设初值经过扰动后变为 $y_0 + \Delta y_0$,

对应的解为 $\hat{y}(t) = (y_0 + \Delta y_0) e^{\lambda(t-t_0)}$,

扰动带来的误差为 $\Delta y(t) = \Delta y_0 e^{\lambda(t-t_0)}$,

当 $t \rightarrow \infty$ 时 $\Delta y(t)$ 的估取法于 λ



(a) $\lambda > 0$ 对应的不稳定问题 (b) $\lambda = 0$ 对应的稳定问题 (c) $\lambda < 0$ 对应的渐进稳定问题
图 初值扰动对上页初值问题的解的影响

若某算法在计算过程中任一步产生的误差在以后的计算中都逐步衰减，则称该算法是**绝对稳定的**（**absolutely stable**）。

一般分析时为简单起见，只考虑如下方程：

$$y' = \lambda y \quad (*) \quad \text{注：}\lambda \text{ 为常数，可以是复数。}$$

当步长取为 h 时，将某算法应用于上式，并假设只在初值产生误差 $\varepsilon_0 = y_0 - \bar{y}_0$ ，则若此误差以后逐步衰减，就称该算法相对于 $\bar{h} = \lambda h$ 绝对稳定， \bar{h} 的全体构成**绝对稳定区域**。

我们称算法 **A** 比算法 **B** 稳定，就是指 **A** 的绝对稳定区域比 **B** 的大。

以欧拉法为例，设 $\text{Re}(\lambda) \leq 0$ ，有：

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

由 y_n 上的扰动 δ_n 引起 y_{n+1} 的误差为：

$$\delta_{n+1} = (1 + h\lambda)\delta_n,$$

要满足 $\delta_{n+1} \leq \delta_n$ ，则要求

$$|1 + h\lambda| \leq 1 \quad (**)$$

该不等式(**)是欧拉法求解模型问题 (*) 保证稳定性的充要条件。

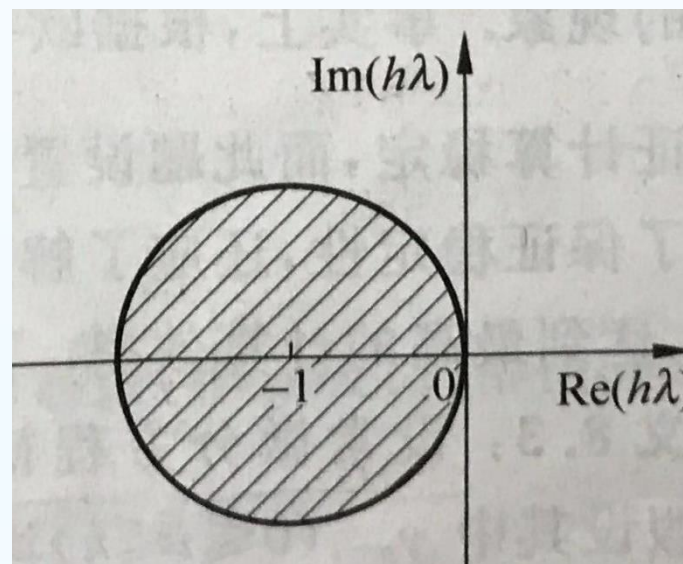


图 欧拉法解上述问题的稳定区域



■考虑稳定性的问题，且 λ 为非零实数的情况，由公式(**)可推导出：

$$h \leq \frac{-2}{\lambda}$$

使用欧拉法计算时，为了保证稳定性，步长 h 不能取得太大。



4、向后差商公式

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

$$\frac{y(x_{n+1}) - y(x_n)}{h} = f(x_{n+1}, y(x_{n+1})) + \frac{h}{2} y''(\xi_n)$$

$$y(x_{n+1}) = y(x_n) + hf(x_{n+1}, y(x_{n+1})) + \frac{h^2}{2} y''(\xi_n)$$

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

是隐格式，要迭代求解

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n) \\ y_{n+1}^{(k+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(k)}), k = 0, 1, \dots \end{cases}$$

隐式欧拉公式具有 1 阶精度。



5、中心差商公式

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

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$$

是多步，2阶格式，该格式不稳定





6、梯形法（改进的欧拉法） —基于数值积分的公式

对微分方程

$$y' = \frac{dy}{dx} = f(x, y) \quad , \quad x \in [a, b]$$

做积分，得：

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



$$y(x_{n+1}) = y(x_n) + \frac{h}{2}[f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))] + \left(-\frac{h^3}{12} f''(\xi)\right)$$

梯形公式:

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

局部截断
误差

截断误差: $O(h^3)$

梯形公式具有2阶精度,
是个隐式的方法, 要用迭代法求解

缺点: 实际计算时要进行多次迭代, 计算量较大。

7、预估-校正法

为了控制计算量，通常只迭代一两次就转入下一步的计算。

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

也常写成下面形式：

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2 \\ k_1 = hf(x_n, y_n), n = 0, 1, 2, \dots \\ k_2 = hf(x_n + h, y_n + k_1) \end{cases}$$

截断误差： $O(h^3)$

预估-校正法是2阶方法，
是个隐式的方法，要用迭代法求解



6.2 Runge—Kutta法（龙格-库塔法）

龙格-库塔法是一类应用较广的**高精度单步法**（计算 y_{n+1} 时，只用到前一步值），简称**R—K法**。





由**Taylor**展开

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

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

$$y''(x) = f_x(x, y) + f_y(x, y) \cdot y'$$

$$y'''(x) = \cdots$$

记为 hT_{n+1}

所以，可以构造格式

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2!} (f_x(x_n, y_n) + f_y(x_n, y_n) \cdot f(x_n, y_n)) + \cdots$$

这种格式使用到了**各阶偏导数**，使用不便。



从另一个角度看，

$$y(x_{n+1}) = y(x_n) + hF(h, x_n, y(x_n), f) + hT_{n+1}$$

取 (x, y) 及其附近的点做线性组合，表示 F ，问题就好办了。当然，要求此时的展开精度相同。这种方法称为 *Runge-Kutta* 法



以2阶为例， 设

$$F(h, x, y, f) = c_1 f(x, y) + c_2 f(x + a_2 h, y + b_{21} h f(x, y))$$

在 (x, y) 处展开，

$$\begin{aligned} F(h, x, y, f) &= c_1 f(x, y) + c_2 \left[f(x, y) + a_2 h f_x(x, y) + b_{21} h f(x, y) f_y(x, y) + O(h^2) \right] \end{aligned}$$

比较

$$y(x_{n+1}) = y(x_n)$$

$$+ h \left[f(x_n, y_n) + \frac{h}{2!} \left(f_x(x_n, y_n) + f_y(x_n, y_n) \cdot f(x_n, y_n) \right) \right] + h T_{n+1}$$

有：

$$\begin{cases} c_1 + c_2 = 1 \\ c_2 a_2 = 1/2 \\ c_2 b_{21} = 1/2 \end{cases}$$

$$\begin{cases} c_1 = c_2 = 1/2 \\ a_2 = 1 \\ b_{21} = 1 \end{cases} \Rightarrow \begin{cases} y_{n+1} = y_n + h/2[K_1 + K_2] \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1) \end{cases}$$

称为**Heun方法**（改进的**Euler公式**），是最简单的**Runge-Kutta方法**。

Runge-Kutta法是具有**共同结构及特点**的一大类，实际应用中**经常被使用**。



一般的Runge—Kutta法构造

$$\left\{ \begin{array}{l} F(h, x, y, f) = c_1 K_1 + c_2 K_2 + \cdots + c_m K_m \\ K_1 = f(x, y) \\ K_2 = f(x + a_2 h, y + b_{21} h K_1) \\ \vdots \\ K_m = f(x + a_m h, y + h \sum_{i=1}^{m-1} b_{mi} K_i) \end{array} \right.$$

常见的为3阶，4阶公式



吉祥如意

3阶龙格-库塔公式:

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





最常用的4阶龙格-库塔公式（标准的4阶龙格-库塔公式）：

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

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

$$k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$



6.3 线性多步法

用若干节点处的 y 及 y' 值的线性组合来近似 $y(x_{n+1})$ 。

其通式可写为：

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \dots + \alpha_k y_{n-k} + h(\beta_{-1} f_{n+1} + \beta_0 f_n + \beta_1 f_{n-1} + \dots + \beta_k f_{n-k})$$

当 $\beta_{-1} \neq 0$ 时，为隐式公式；
 $\beta_{-1} = 0$ 则为显式公式。

$$f_n = f(x_n, y_n)$$



➤ 基于数值积分的构造法

将 $y' = f(x, y)$ 在 $[x_{n-p}, x_{n+1}]$ 上积分, 得到

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

只要近似地算出右边的积分 $I_k \approx \int_{x_{n-p}}^{x_{n+1}} f(x, y(x)) dx$, 则可通过 $y_{n+1} = y_{n-p} + I_k$ 近似 $y(x_{n+1})$ 。而选用不同近似式 I_k , 可得到不同的计算公式。



若积分 $\int_{x_{n-p}}^{x_{n+1}} y'(x) dx$ 用节点 $x_n, x_{n-1}, \dots, x_{n-q}$ 作为积分点，则有

$$\int_{x_{n-p}}^{x_{n+1}} y'(x) dx = h[a_0 y'(x_n) + a_1 y'(x_{n-1}) + \dots + a_q y'(x_{n-q})] + hT_{n+1}$$

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

局部截断误差

$$y(x_{n+1}) = y(x_{n-p}) + h \sum_{j=0}^q a_j f(x_{n-j}, y(x_{n-j})) + hT_{n+1}$$

积分系数 $ha_j = \int_{x_{n-p}}^{x_{n+1}} l_j(x) dx$

Lagrange插值基函数

$$\int_{x_{n-p}}^{x_{n+1}} \frac{y^{(q+2)}(\xi)}{(q+1)!} \omega_q(x) dx$$

这是显格式， $q+1$ 阶 $r+1$ 步格式。 $r = \max\{p, q\}$

同样，若以 $x_{n+1}, x_n, \dots, x_{n-q+1}$ 为积分节点，可以构造 $r+1$ 步 $q+1$ 阶隐格式。



例：建立 $p=1, q=2$ 的显格式

$$p=1, \quad \text{积分区间为 } \int_{x_{n-1}}^{x_{n+1}} y'(x) dx$$

$$q=2, \quad \text{显格式, 积分节点为 } x_n, x_{n-1}, x_{n-2}$$

所以

$$ha_0 = \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_{n-1})(x - x_{n-2})}{(x_n - x_{n-1})(x_n - x_{n-2})} dx = \frac{7}{3} h$$

$$ha_1 = \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_n)(x - x_{n-2})}{(x_{n-1} - x_n)(x_{n-1} - x_{n-2})} dx = -\frac{2}{3} h$$

$$ha_2 = \int_{x_{n-1}}^{x_{n+1}} \frac{(x - x_n)(x - x_{n-1})}{(x_{n-2} - x_n)(x_{n-2} - x_{n-1})} dx = \frac{1}{3} h$$



三步三阶显式格式:

$$y_{n+1} = y_{n-1} + \frac{h}{3} [7f(x_n, y_n) - 2f(x_{n-1}, y_{n-1}) + f(x_{n-2}, y_{n-2})]$$

例: 建立 $p=2, q=2$ 的隐格式

$p=2$, 积分区间为 $\int_{x_{n-2}}^{x_{n+1}} y'(x) dx$

$q=2$, 隐格式, 积分节点为 x_{n+1}, x_n, x_{n-1}

所以 $ha_0 = \int_{x_{n-2}}^{x_{n+1}} \frac{(x - x_n)(x - x_{n-1})}{(x_{n+1} - x_n)(x_{n+1} - x_{n-1})} dx = \frac{3}{4}h$

$$ha_1 = \int_{x_{n-2}}^{x_{n+1}} \frac{(x - x_{n+1})(x - x_{n-1})}{(x_n - x_{n+1})(x_n - x_{n-1})} dx = 0$$

$$ha_2 = \int_{x_{n-2}}^{x_{n+1}} \frac{(x - x_{n+1})(x - x_n)}{(x_{n-1} - x_{n+1})(x_{n-1} - x_n)} dx = \frac{9}{4}h$$

三步三阶隐式格式：

$$y_{n+1} = y_{n-2} + \frac{h}{4} [3f(x_{n+1}, y_{n+1}) + 9f(x_{n-1}, y_{n-1})]$$

若 $p=0$ ，上述两种方法对应的格式被称为**Adams公式**。

例如， $q=0$ 时，对应的**Adams**格式即为**Euler**格式；

$q=1$ 时，对应的**两步Adams显式格式**为

$$y_{n+1} = y_n + \frac{h}{2} [3f(x_n, y_n) - f(x_{n-1}, y_{n-1})]$$

对应的**两步Adams隐式格式**为

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

也称为**梯形公式**。



四步Adams显式公式

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

四步Adams隐式公式



$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2})$$



吉祥如意

■ 习题课（2020-06-11）





§ 6.4 方程组和高阶方程的数值解法（选）

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, \dots, y_m) \\ \vdots \\ \frac{dy_m}{dx} = f_m(x, y_1, \dots, y_m) \\ y_1(a) = \eta_1 \\ \vdots \\ y_m(a) = \eta_m \end{cases}, \quad a \leq x \leq b$$

写成向量的形式：

$$\begin{cases} \frac{dY}{dx} = F(x, Y) \\ Y(a) = \eta \end{cases}$$

各种方法都可以直接运用过来。以两个方程的方程组为例

$$\begin{cases} \frac{dy}{dx} = f(x, y, z) \\ \frac{dz}{dx} = g(x, y, z) \\ y(a) = y_0 \\ z(a) = z_0 \end{cases}, \quad a \leq x \leq b$$

Euler公式

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n, z_n) \\ z_{n+1} = z_n + hg(x_n, y_n, z_n) \\ y_0 = y(x_0), z_0 = z(x_0), n = 0, 1, 2, \dots \end{cases}$$

改进的欧拉法:

$$\left\{ \begin{array}{l} y_{n+1}^{(0)} = y_n + hf(x_n, y_n, z_n) \\ z_{n+1}^{(0)} = z_n + hg(x_n, y_n, z_n) \\ y_{n+1}^{(k+1)} = y_n + \frac{h}{2}[f(x_n, y_n, z_n) + f(x_{n+1}, y_{n+1}^{(k)}, z_{n+1}^{(k)})] \\ z_{n+1}^{(k+1)} = z_n + \frac{h}{2}[f(x_n, y_n, z_n) + f(x_{n+1}, y_{n+1}^{(k)}, z_{n+1}^{(k)})] \\ k = 0, 1, 2, \dots \end{array} \right.$$



Runge-Kutta公式

$$\begin{pmatrix} y_{n+1} \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} y_n \\ z_n \end{pmatrix} + \frac{h}{6} (K_1 + 2K_2 + 2K_3 + K_4)$$

$$K_1 = \begin{pmatrix} f(x_n, y_n, z_n) \\ g(x_n, y_n, z_n) \end{pmatrix}$$

$$K_2 = \begin{pmatrix} f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_1^{(1)}, z_n + \frac{h}{2} K_1^{(2)}\right) \\ g\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_1^{(1)}, z_n + \frac{h}{2} K_1^{(2)}\right) \end{pmatrix}$$



吉祥如意

$$K_3 = \begin{pmatrix} f(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_2^{(1)}, z_n + \frac{h}{2} K_2^{(1)}) \\ g(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_2^{(1)}, z_n + \frac{h}{2} K_2^{(1)}) \end{pmatrix}$$

$$K_4 = \begin{pmatrix} f(x_n + h, y_n + h K_3^{(1)}, z_n + h K_3^{(1)}) \\ g(x_n + h, y_n + h K_3^{(1)}, z_n + h K_3^{(1)}) \end{pmatrix}$$





四阶亚当斯显式公式:

$$\left\{ \begin{array}{l} y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \\ z_{n+1} = z_n + \frac{h}{24}(55g_n - 59g_{n-1} + 37g_{n-2} - 9g_{n-3}) \end{array} \right. \\ n = 4, 5, 6, \dots$$





$$\begin{cases} \frac{du}{dt} = 0.05u(1 - \frac{u}{20}) - 0.002uv \\ \frac{dv}{dt} = 0.09v(1 - \frac{v}{15}) - 0.15uv \\ u(0) = 0.193 \\ v(0) = 0.083 \end{cases}$$

1、
$$\vec{F} = \begin{pmatrix} f(u, v, t) \\ g(u, v, t) \end{pmatrix} = \begin{pmatrix} 0.05u(1 - u/20) - 0.002uv \\ 0.09v(1 - v/15) - 0.15uv \end{pmatrix}$$

2、确定方法，然后求解

4阶 **Runge-Kutta** 法, $h=1$

(0.20276	0.0881157)
(0.213007	0.0934037)
(0.223763	0.0988499)
(0.235052	0.104437)
(0.246902	0.110146)





高阶方程:

$$\left\{ \begin{array}{l} \frac{d^m y}{dx^m} = f_1(x, y, y', \dots, y^{(m-1)}) \\ y(a) = \eta_1 \\ y'(a) = \eta_2 \\ \vdots \\ y^{(m-1)}(a) = \eta_m \end{array} \right. , \quad a \leq x \leq b$$



吉祥如意

令

$$\begin{cases} y = y_1 \\ \frac{dy_1}{dx} = y_2 \\ \vdots \\ \frac{dy_{m-1}}{dx} = y_m \end{cases}$$

则有:

$$\begin{cases} \frac{dy_1}{dx} = y_2 \\ \vdots \\ \frac{dy_m}{dx} = f(x, y_1, \cdots, y_m) \\ y_1(a) = \eta_1 \\ \vdots \\ y_m(a) = \eta_m \end{cases}, \quad a \leq x \leq b$$

6.5 边值问题的数值解法（选）

二阶常微分方程的边值问题（**Boundary-value problem**）一般可归纳为三类：

微分方程 $\frac{d^2 y}{dx^2} = f(x, y, \frac{dy}{dx}), x \in [x_0, x_n]$

边界条件

第一边值问题 $y(x_0) = y_0, y(x_n) = y_n$

第二边值问题 $\frac{dy}{dx} \Big|_{x_0} = y'_0, \frac{dy}{dx} \Big|_{x_n} = y'_n$

第三边值问题 $\frac{dy}{dx} \Big|_{x_0} - \alpha_0 y(x_0) = \alpha_1, \frac{dy}{dx} \Big|_{x_n} + \beta_0 y(x_n) = \beta_1$

其中 $\alpha_0 \geq 0, \beta_0 \geq 0, \alpha_0 + \beta_0 > 0$



6.5.1 有限差分法(矩阵法) (选)

二阶线性方程的一般形式:

$$y'' + p(x)y' + q(x)y = f(x), a \leq x \leq b$$



将区间 $[a, b]$ 进行等距划分, 取节点 $x_i = a + ih$ ($i = 0, \dots, N$)。



在每一个节点处，将 y' 和 y'' 用差商近似表示，要求相同的截断误差。

$$y''(x) = \frac{\frac{y(x+h) - y(x)}{h} - \frac{y(x) - y(x-h)}{h}}{h} - \frac{h^2}{12} y^{(4)}(\xi)$$

$$= \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + O(h^2)$$

$$y'(x) = \frac{y(x+h) - y(x-h)}{2h} + O(h^2)$$



假设 $y_i = y(x_i)$,

于是有

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i,$$

$$i = 0, 1, \dots, n-1$$

整理后得如下形式:

$$a_i y_{i-1} + b_i y_i + c_i y_{i+1} = d_i, i = 1, 2, \dots, n-1$$



对于第一边值条件，有

$$\begin{cases} y_0 = \alpha \\ y_n = \beta \end{cases}$$

差分方程组：

$$\begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-2} & b_{n-2} & c_{n-2} \\ & & & a_{n-1} & b_{n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-2} \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 - a_1 \alpha \\ d_2 \\ \vdots \\ d_{n-2} \\ d_{n-1} - c_{n-1} \beta \end{bmatrix}$$

用追赶法求解。



6.5.2 打靶法（试射法）（选）

二阶微分方程第一边值问题

$$\begin{cases} y'' = f(x, y, y') \\ y(a) = \alpha, y(b) = \beta \end{cases}$$

基本思想：把边值问题化为初值问题来解。

具体作法：反复调整初始时刻的斜率 $y'(a)$ 的值 m ，

使得初值问题
$$\begin{cases} y'' = f(x, y, y') \\ y(a) = \alpha, y'(a) = m \end{cases}$$

的解满足另一个边值条件 $y(b) = \beta$ ，也就是从初值问题的经过点 (a, α) ，而且有不同斜率的积分曲线中，寻找一条通过 (b, β) 的曲线。

Thanks!

感谢中科大张瑞老师!

