四 常微分方程数值解法

主要内容

- 引言
- 欧拉方法
  - （显式）欧拉法、隐式欧拉法、二步欧拉法、梯形公式欧拉法
  - 局部截断误差与精度
- 改进的欧拉方法
- 龙格-库塔方法
- 收敛性与稳定性
- 一阶常微分方程组与高阶常微分方程

引言

- 在科学研究与工程技术中的许多问题，都可通过微分方程来建立数学模型进行分析。
- 本章主要以一阶常微分方程为主，介绍常微分方程初值问题的差分方法和相关理论。
- 一阶常微分方程初值问题：

$$\begin{cases} y' = f(x, y) & \longleftarrow \text{微分方程} \\ y(x_0) = y_0 & \longleftarrow \text{初始条件} \end{cases}$$

其中 $f(x, y)$ 是已知函数，y_0 为给定的值。

何时存在唯一解？如何计算 $y(x)$？

引言（续）

（一阶常微分方程）初值问题 $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$

定理： 若 $f(x,y)$ 在某闭区域 R：

$$|x - x_0| \le a, \quad |y - y_0| \le b \qquad (a > 0, b > 0)$$

上连续，且在 R 域内满足李普希兹 (Lipschitz) 条件，即存在正数 L，使得对于 R 域内的任意两值 y_1, y_2，下列不等式成立：

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

则上述初值问题的连续可微的解 $y(x)$ 存在并且唯一。

引言（续）

- 实际生产与科研中，除少数简单情况能获得初值问题的初等解或解析解（用初等函数表示的解）外，绝大多数情况下是求不出初等解的。
- 有些初值问题即便有初等解，也往往由于形式过于复杂而不便处理，而且实际问题往往只要求在某一时刻解的函数值。
- 实用的方法是在计算机上进行数值求解：即不直接求 $y(x)$ 的显式解，而是在解所存在的区间上，求得一系列点 x_n（$n = 0, 1, 2, \ldots$）上解的近似值。

引言（续）

- 一元 Taylor 公式

$$y(x_n+h)=y(x_n)+y'(x_n)h+\frac{y''(x_n)}{2!}h^2+\frac{y'''(x_n)}{3!}h^3+O(h^4)$$

- 二元 Taylor 公式

$$f(x_n+h,y_n+k)$$

$$=f(x_n,y_n)+\frac{\partial f(x_n,y_n)}{\partial x}h+\frac{\partial f(x_n,y_n)}{\partial y}k$$

$$+\frac{1}{2!}\left[\frac{\partial^2 f(x_n,y_n)}{\partial x^2}h^2+2\frac{\partial^2 f(x_n,y_n)}{\partial x\partial y}hk+\frac{\partial^2 f(x_n,y_n)}{\partial y^2}k^2\right]+\cdots$$

$$+\frac{1}{k!}\left[h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right]^k f(x_n,y_n)+\cdots$$

引言（续）

- 为避免混淆，做如下约定：
  - $y(x_n)$：待求函数 $y(x)$ 在 x_n 处的**精确**函数值
  - y_n：待求函数 $y(x)$ 在 x_n 处的**近似**函数值
- 求数值解的主要问题：
  - 如何将 $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$ 离散化，并建立求其数值解的**递推公式**
  - 如何求递推公式的**局部截断误差**，即数值解 y_n 与精确解 $y(x_n)$ 的误差估计
  - 研究递推公式的**稳定性与收敛性**

欧拉（Euler）方法

- **构造数值解法的基本思想**

假设初值问题 $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$ 的解 $y = y(x)$ 唯一存在且足够光滑。

将求解区间 $[a, b]$ 做等分

$$a = x_0 < x_1 < x_2 < \ldots < x_n < \ldots < x_N = b$$

$h = (b - a)/N$ 称为步长；等距节点为

$$x_n = a + nh,\quad n = 0, 1, \ldots, N$$

数值解法就是求精确解 $y(x)$ 在等距节点 x_n 上的值 $y(x_n)$ 的近似值 y_n，$n = 1, 2, \ldots, N$。

欧拉方法（续）

构造数值解法的基本思想是：通过某种离散化方法将

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

在等距节点 $\{x_n\}$ 上离散化，建立关于节点近似值 $\{y_n\}$ 的差分方程，然后结合定解条件由差分方程求出近似值 y_n，$n = 1, 2, \ldots$。

- **三种基本离散方法**
  - 差商
  - 数值积分
  - Taylor展开

四种欧拉方法

局部截断误差

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

方法一　化导数为差商的方法

- 向前差商数值微分公式

$$y'(x_n) = \lim_{h\to 0}\frac{y(x_n+h)-y(x_n)}{h} \approx \frac{y(x_{n+1})-y(x_n)}{h}$$

- 向后差商数值微分公式

$$y'(x_n) = \lim_{h\to 0}\frac{y(x_n)-y(x_n-h)}{h} \approx \frac{y(x_n)-y(x_{n-1})}{h}$$

- 中心差商数值微分公式

$$y'(x_n) = \lim_{h\to 0}\frac{y(x_n+h)-y(x_n-h)}{2h} \approx \frac{y(x_{n+1})-y(x_{n-1})}{2h}$$

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

向前差商

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

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

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

由于在逐步求解的过程中，$y(x_n)$ 的准确值无法求解出来，因此用其近似值 y_n 代替。

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

——（显式）欧拉法，单步

欧拉方法（续）

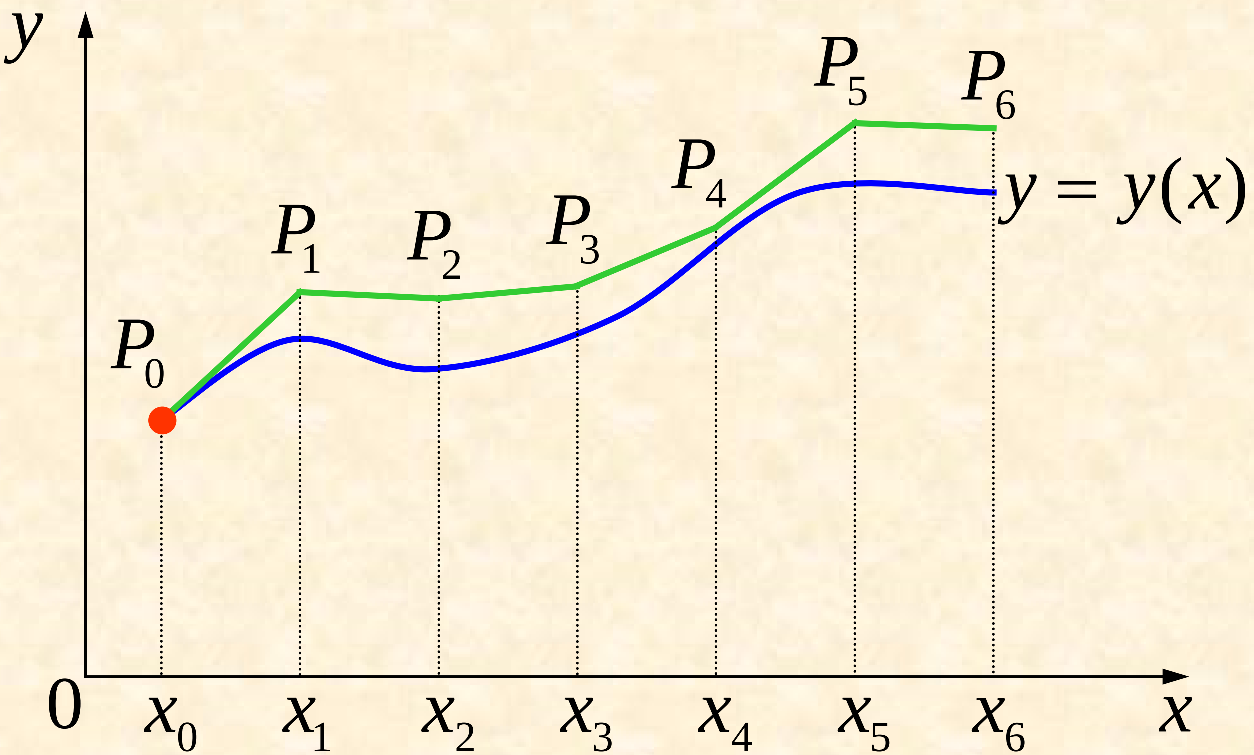
$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n) & n = 0,1,2,\cdots \\ y_0 = y(x_0) \end{cases}$$

- 根据 y_0 可以一步步计算出函数 $y=y(x)$ 在 x_1, x_2, x_3 $x_4, \ldots$ 上的近似值 $y_1, y_2, y_3, y_4, \ldots$。
- 常微分方程数值解是一组离散的函数值数据，它的精确表达式很难求解得到，但可以进行插值计算后用插值函数逼近 $y(x)$。

欧拉方法（续）

欧拉方法的几何意义

$$\boxed{\begin{aligned} y_{n+1} &= y_n + hf(x_n, y_n) \Leftrightarrow \\ y_{n+1} - y_n &= f(x_n, y_n)(x_{n+1} - x_n) \end{aligned}}$$



在P_0点以$f(x_0, y_0)$为斜率作直线交$x=x_1$于P_1点。

在P_1点以$f(x_1, y_1)$为斜率作直线交$x=x_2$于P_2点。

依此类推。

将点$P_0, P_1, \ldots, P_N$连成折线，在几何上看成解曲线的近似曲线！

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

向后差商

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

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

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

含有未知的函数值

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

—— 隐式欧拉法，单步

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

中心差商

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

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

$$y(x_{n+1}) = y(x_{n-1}) + 2hf(x_n, y(x_n))$$

y_1 由其他方法提供

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

需要前两步的计算结果

—— 二步欧拉法，显式

欧拉方法（续）

方法二　数值积分法

将微分方程 $y' = f(x, y)$ 在区间 $[x_n, x_{n+1}]$ 上积分：

$$\int_{x_n}^{x_{n+1}} y' \mathrm{d}x = y(x_{n+1}) - y(x_n) = \underline{\int_{x_n}^{x_{n+1}} f(x, y(x)) \mathrm{d}x}$$

左矩形公式 $\underline{\approx (x_{n+1} - x_n) f(x_n, y(x_n))}$

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

- 左矩形公式
- 右矩形公式
- 中矩形公式
- 梯形公式

- 以近似值 y_n 代替精确值 $y(x_n)$ 可得

（显式）欧拉法：

$$\begin{cases} y_{n+1} = y_n + h f(x_n, y_n) \\ y_0 = y(x_0) \end{cases}$$

欧拉方法（续）

- 在数值积分法推导中，积分的近似值取为积分区间长度与右端点处的函数值乘积，即：

$$\int_{x_n}^{x_{n+1}} y'\mathrm{d}x = y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f\big(x, y(x)\big)\mathrm{d}x$$

右矩形公式 $\approx (x_{n+1} - x_n)f\big(x_{n+1}, y(x_{n+1})\big) = hf\big(x_{n+1}, y(x_{n+1})\big)$

- 这样便得到了隐式欧拉法：

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

- 隐式欧拉法没有显式欧拉法方便

欧拉方法（续）

- 在数值积分法推导中，积分区间长度选为两步步长，即积分区间为：$[x_{n-1}, x_{n+1}]$，则：

$$\int_{x_{n-1}}^{x_{n+1}} y'\mathrm{d}x = y(x_{n+1}) - y(x_{n-1}) = \underline{\int_{x_{n-1}}^{x_{n+1}} f\big(x, y(x)\big)\mathrm{d}x}$$

中矩形公式

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

$$= 2hf\big(x_n, y(x_n)\big)$$

- 以 $y(x)$ 在 x_{n-1}, x_n 上的近似值代替精确值可得二步欧拉法：

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

欧拉方法（续）

- 在数值积分法中，如果用梯形公式近似计算 $f(x, y)$ 在区间 $[x_n, x_{n+1}]$ 上的积分，即：

$$\int_{x_n}^{x_{n+1}} f\big(x, y(x)\big)\mathrm{d}x \approx (x_{n+1} - x_n)\frac{f\big(x_n, y(x_n)\big) + f\big(x_{n+1}, y(x_{n+1})\big)}{2}$$

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

- 用近似值代替精确值可得梯形公式欧拉法：

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

隐式

单步

- 上式右端出现了未知项，可见梯形法是隐式欧拉法的一种；实际上，梯形公式欧拉法是显式欧拉法与隐式欧拉法的**算术平均**。

欧拉方法（续）

方法三 泰勒展开法

- 如果初值问题中的 $f(x,y)$ 充分可微，则可将 $y(x_{n+1})$ 在点 x_n 处展开：

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

- 如果只保留线性项，忽略 h^2 及以上各项，则：

$$y(x_{n+1}) \approx y(x_n) + hy'(x_n) = y_{n+1}$$

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

$$y_n$$

显式欧拉公式 $y_{n+1} = y_n + hf(x_n, y_n)$

例题

- 用显式欧拉法、隐式欧拉法、梯形法求解初值问题：

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

取 $h = 0.1$，计算到 $x = 0.5$，并与精确解进行比较。

解：由已知条件可得：$h = 0.1$，$x_0 = 0$，$y_0 = 1$，

$$f(x, y) = -y + x + 1$$

- 显式欧拉法：

$$\begin{aligned} y_{n+1} &= y_n + h(-y_n + x_n + 1) \\ &= (1 - h)y_n + hx_n + h \\ &= 0.9y_n + 0.1x_n + 0.1 \end{aligned}$$

例题（续）

- 隐式欧拉法： $y_{n+1} = y_n + h(-y_{n+1} + x_{n+1} + 1)$

  化简得：

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

  $$y_{n+1} = \frac{1}{(1+h)}[y_n + h(x_n + h + 1)] = (y_n + 0.1x_n + 0.11)/1.1$$

- 梯形公式欧拉法：

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

  $$[1+(h/2)]y_{n+1} = [1-(h/2)]y_n + (h/2)(x_n + x_{n+1} + 2)$$

  $$y_{n+1} = [(2-h)y_n + h(x_n + x_{n+1} + 2)]/(2+h)$$

  $$= (1.9y_n + 0.2x_n + 0.21)/2.1$$

显式法 $y_{n+1} = 0.9y_n + 0.1x_n + 0.1$

隐式法 $y_{n+1} = (y_n + 0.1x_n + 0.11)/1.1$

梯形法 $y_{n+1} = (1.9y_n + 0.2x_n + 0.21)/2.1$

◆ 计算结果:

x_n	显式法 y_n	隐式法 y_n	梯形法 y_n	精确解 $y(x_n)$
0.0	1	1	1	1
0.1	1.000000	1.009091	1.004762	1.004837
0.2	1.010000	1.026446	1.018594	1.019731
0.3	1.029000	1.051315	1.040633	1.040818
0.4	1.056100	1.083014	1.070097	1.070320
0.5	1.090490	1.120922	1.106278	1.106531

本题的精确解为: $y(x) = x + e^{-x}$

局部截断误差

- 为了简化分析某常微分方程数值算法的误差，现假设 $y_n = y(x_n)$，即在前一步 y_n 准确的前提下，估计：

$$T_{n+1} = y(x_{n+1}) - y_{n+1}$$

- 称上述误差 T_{n+1} 为该常微分方程数值算法的局部截断误差。
- 如果某个常微分方程数值算法的局部截断误差可表示为 $O(h^{p+1})$，则称该数值算法的精度是 p 阶。
- 精度阶数越高，方法的精度越好。
- 欧拉法的精度为一阶；二步欧拉法的精度为二阶；梯形法的精度为二阶。

局部截断误差的分析

- 利用泰勒公式展开，比较各算法与展开式的前几项
  - 显式欧拉法的局部截断误差：

欧拉法 $y_{n+1} = y_n + hf(x_n, y_n)$

令：$y_n = y(x_n)$

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

将 $y(x_{n+1})$ 在 x_n 点处用泰勒公式展开：

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

$$\xi \in (x_n, x_{n+1})$$

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

1 阶精度

$y(x_n)$

隐式欧拉法：$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$

$y(x_{n+1})$

令 $y_n = y(x_n)$

这是基础！

令隐式欧拉法右边的

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

公式右边都是精确的！

$$
\begin{aligned}
y_{n+1} &= y(x_n) + hf(x_{n+1}, y(x_{n+1})) \\
&= y(x_n) + hy'(x_{n+1}) \\
&= y(x_n) + h(y'(x_n) + hy''(x_n) + O(h^2)) \\
&= y(x_n) + hy'(x_n) + h^2 y''(x_n) + O(h^3)
\end{aligned}
$$

将 $y'(x_{n+1})$ 在 x_n 点处Taylor展开

将 $y(x_{n+1})$ 在 x_n 点处Taylor展开：

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

$T_{n+1} = y(x_{n+1}) - y_{n+1} = O(h^2)$

1 阶精度

局部截断误差的分析（续）

补充：二元函数微分中值定理

函数 $f(x, y)$ 在点 (x_0, y_0) 处

$$f(x_0+h, y_0+k)$$

$$=f(x_0, y_0)+hf_x(x_0+\theta h, y_0+\theta k)$$

$$+kf_y(x_0+\theta h, y_0+\theta k)$$

$$=f(x_0, y_0)$$

$$+\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)f(x_0+\theta h, y_0+\theta k)(0<\theta<1)$$

$y(x_n)$

隐式欧拉法：$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$

$$f(x_{n+1}, y_{n+1}) = f\big(x_{n+1}, y(x_{n+1})\big) + f_y(x_{n+1}, \xi)[y_{n+1} - y(x_{n+1})]$$

$$= y'(x_{n+1}) - f_y(x_{n+1}, \xi)T_{n+1}$$

ξ 在 $y(x_{n+1})$ 与 y_{n+1} 之间

$$= y'(x_n) + O(h) - f_y(x_{n+1}, \xi)T_{n+1}$$

$$y_{n+1} = y(x_n) + h[y'(x_n) + O(h) - f_y(x_{n+1}, \xi)T_{n+1}]$$

$$= y(x_n) + hy'(x_n) - hf_y(x_{n+1}, \xi)T_{n+1} + O(h^2)$$

$y(x_{n+1})$ 在 x_n 点处展开： $y(x_{n+1}) = y(x_n) + hy'(x_n) + O(h^2)$

$$T_{n+1} = y(x_{n+1}) - y_{n+1} = hf_y(x_{n+1}, \xi)T_{n+1} + O(h^2)$$

$$T_{n+1} = \frac{1}{1 - hf_y(x_{n+1}, \xi)} O(h^2)$$

1 阶精度

二步欧拉法： $y_{n+1} = \underset{\uparrow\ y(x_{n-1})}{y_{n-1}} + 2hf(x_n, \underset{\uparrow\ y(x_n)}{y_n})$

- 二步欧拉法的局部截断误差：

$$y_{n+1} = y(x_{n-1}) + 2hf\big(x_n, y(x_n)\big) = \boxed{y(x_{n-1}) + 2hy'(x_n)}$$

分别将 $y(x_{n+1}), y(x_{n-1})$ 在 x_n 点处用泰勒公式展开：

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

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

$$T_{n+1} = y(x_{n+1}) - y_{n+1}$$

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

2 阶精度

梯形公式欧拉法：

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

$y(x_n)$

$y(x_{n+1})$

证明梯形公式欧拉法的精度为 2 阶。通用方法！

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

（2）令 $y_n=y(x_n)$，及隐式欧拉法右边的 $y_{n+1}=y(x_{n+1})$

（3）$y_{n+1}=y(x_n)+\frac{h}{2}[y'(x_n)+y'(x_{n+1})]$

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

（4）由泰勒公式： 没有办法的时候考虑泰勒公式！

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

（5）$T_{n+1}=y(x_{n+1})-y_{n+1}=O(h^3)$

2 阶精度

梯形公式欧拉法:

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

$y(x_n)$

$$f(x_{n+1}, y_{n+1}) = f\big(x_{n+1}, y(x_{n+1})\big) + f_y(x_{n+1}, \xi)[y_{n+1} - y(x_{n+1})]$$

$$= y'(x_{n+1}) - f_y(x_{n+1}, \xi)T_{n+1}$$

ξ 在 $y(x_{n+1})$ 与 y_{n+1} 之间

$$= y'(x_n) + hy''(x_n) + O(h^2) - f_y(x_{n+1}, \xi)T_{n+1}$$

$$y_{n+1} = y(x_n) + \frac{h}{2}[2y'(x_n) + hy''(x_n) + O(h^2) - f_y(x_{n+1}, \xi)T_{n+1}]$$

$$= y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) - \frac{h}{2}f_y(x_{n+1}, \xi)T_{n+1} + O(h^3)$$

$y(x_{n+1})$ 在 x_n 点处展开:

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

2 阶精度

$$T_{n+1} = \frac{h}{2}f_y(x_{n+1}, \xi)T_{n+1} + O(h^3) \qquad T_{n+1} = \frac{1}{1 - \frac{h}{2}f_y(x_{n+1}, \xi)}O(h^3)$$

各种欧拉法的比较

方法	精度	评述
显式欧拉法	1	最简单，精度低
隐式欧拉法	1	不便计算，稳定性好
二步欧拉法	2	需要两步初值，且第 2 个初值只能由其它方法给出，可能对后面的递推精度有影响
梯形公式欧 拉 法	2	精度有所提高，但为隐式，需要迭代求解，计算量大

改进的欧拉法

- 从上述例子可以看到，梯形法由于具有二阶精度，其局部截断误差比显式欧拉法和隐式欧拉法小，但梯形法实质上是一种隐式算法。
- 显式欧拉法是一个显式算法，虽然计算量较小，但是精度不高。
- 综合两种方法的长处，可以先用显式欧拉法求出 $y(x_{n+1})$ 的一个粗略近似值，然后用它代入梯形法公式的右端，用梯形法计算 $y(x_{n+1})$ 的较为精确的近似值。

预报值 $\bar{y}_{n+1}$

校正值 y_{n+1}

改进的欧拉法（续）

- 按照上述思想，可以建立如下**预报-校正**系统：

预报：$\bar{y}_{n+1} = y_n + h f(x_n, y_n)$

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

- 按以上两式求解常微分方程的算法称为**改进的欧拉法**，它还可以表示为：

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

平均化形式 $\begin{cases} y_p = y_n + h f(x_n, y_n) & \text{看作欧拉法} \\ y_c = y_n + h f(x_{n+1}, y_p) & \text{看作近似隐式欧拉法} \\ y_{n+1} = (y_p + y_c)/2 & \text{二者算术平均} \end{cases}$ **2 阶精度**

$$\begin{cases} y' = -y + x + 1 \\ y(0) = 1 \end{cases} \qquad h = 0.1, \quad 计算到\ x = 0.5$$

- 用改进欧拉法求上例所述的初值问题并与欧拉法和梯形法比较误差的大小。

解： 采用改进欧拉法的嵌套形式：

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

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

$$= y_n + \frac{h}{2}\{(-y_n + x_n + 1) - [y_n + h(-y_n + x_n + 1)] + x_n + h + 1\}$$

$$= \left[1 + \frac{h(-2+h)}{2}\right] y_n + \frac{h(2-h)}{2} x_n + h$$

$$= 0.905 y_n + 0.095 x_n + 0.1$$

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

$h = 0.1$， 计算到 $x = 0.5$

- 计算结果

x_n	改进欧拉法 y_n	精确解 $y(x_n)$	误差		
			改进欧拉法	欧拉法	梯形法
0.1	1.005000	1.004837	1.6×10^{-4}	4.8×10^{-3}	7.5×10^{-5}
0.2	1.019205	1.019731	2.9×10^{-4}	8.7×10^{-3}	1.4×10^{-4}
0.3	1.041218	1.040818	4.0×10^{-4}	1.2×10^{-2}	1.9×10^{-4}
0.4	1.070802	1.070320	4.8×10^{-4}	1.4×10^{-2}	2.2×10^{-4}
0.5	1.107076	1.106531	5.5×10^{-4}	1.6×10^{-2}	2.5×10^{-4}

- 可见，改进欧拉法的误差数量级与梯形法大致相同，而比欧拉法小得多。

改进的欧拉法的意义

改进的欧拉法的平均化形式
$$\begin{cases} y_p = y_n + h\boxed{f(x_n, y_n)} \\ y_c = y_n + h\boxed{f(x_{n+1}, y_p)} \\ y_{n+1} = (y_p + y_c)/2 \end{cases}$$

k_1

$y_n + hk_1$

k_2

改进的欧拉法是函数 $y(x)$ 在 x_n、x_{n+1} 两个点的导数的近似值的算术平均值代替在点 ξ 上的导数值

$$y_{n+1} = y_n + \frac{h}{2}k_1 + \frac{h}{2}k_2 = y_n + h\boxed{\frac{k_1 + k_2}{2}}$$

- $y(x_{n+1})$ 在点 x_n 处的一阶展开式为：

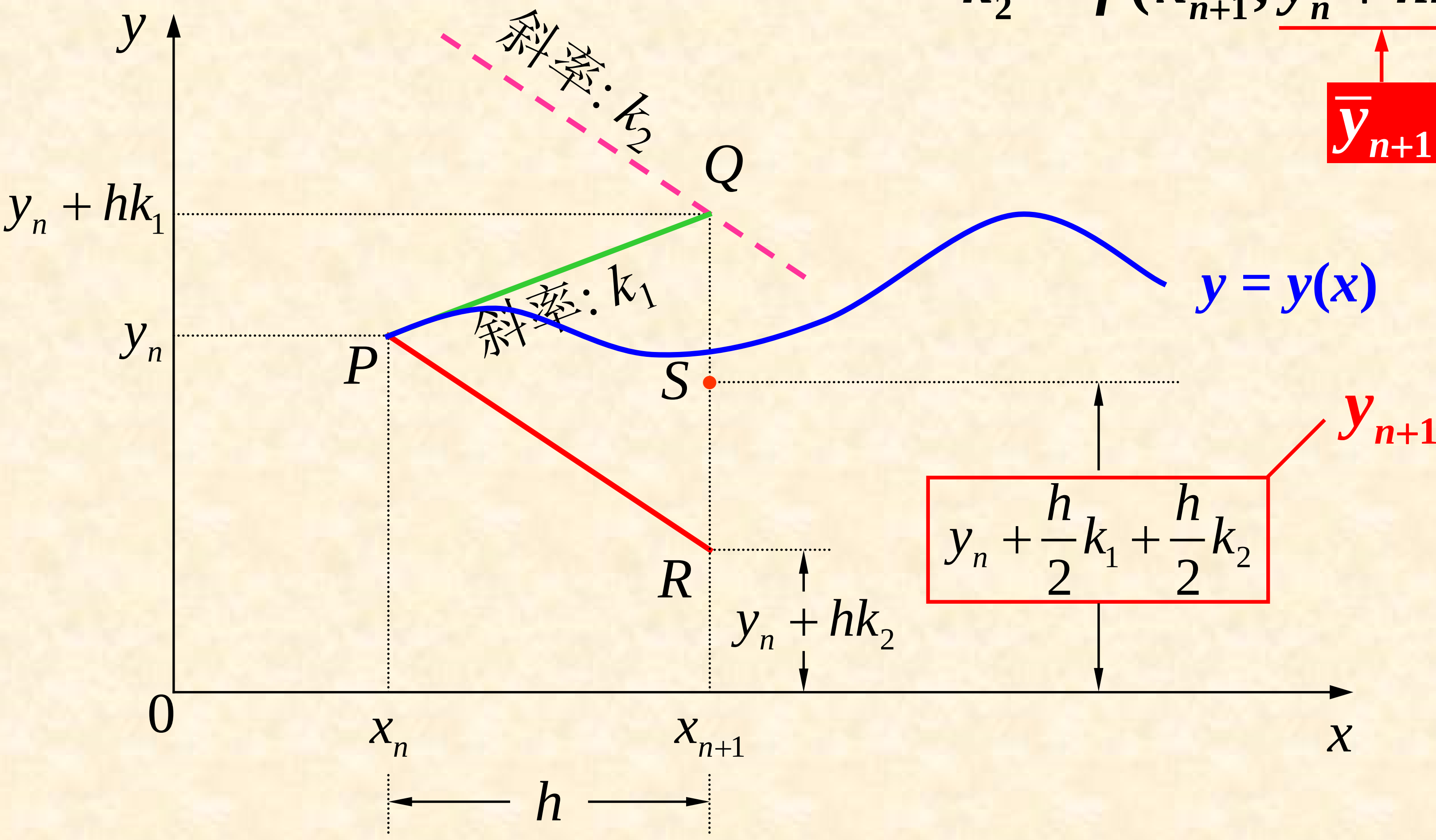
$$y(x_{n+1}) = y(x_n + h) = y(x_n) + h\underline{y'(\xi)} \qquad \xi \in (x_n, x_{n+1})$$

改进的欧拉法的几何意义

$$y_{n+1} = (y_n + hk_1)/2 + (y_n + hk_2)/2$$

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

$$k_2 = f(x_{n+1}, \underbrace{y_n + hk_1}_{\bar{y}_{n+1}})$$



龙格-库塔(Runge-Kutta)方法

- 显式欧拉法（1 阶精度）

$$y_{n+1} = y_n + hf(x_n, y_n) \Rightarrow \begin{cases} y_{n+1} = y_n + hk_1 \\ k_1 = f(x_n, y_n) \end{cases}$$

- 改进的欧拉法（2 阶精度）

$$\begin{cases} y_p = y_n + hf(x_n, y_n) \\ y_c = y_n + hf(x_{n+1}, y_p) \\ y_{n+1} = (y_p + y_c)/2 \end{cases} \Rightarrow \begin{cases} y_{n+1} = y_n + h \cdot \dfrac{k_1 + k_2}{2} \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_{n+1}, y_n + hk_1) \end{cases}$$

k^*

- $y(x_{n+1})$ 在点 x_n 处的一阶泰勒展开式为:

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(\xi)$$

$$= y(x_n) + h\boxed{f(\xi, y(\xi))}$$

$$\xi \in (x_n, x_{n+1})$$

龙格-库塔方法（续）

- 显式欧拉法用一个点的斜率值 k_1 作为平均斜率 k^* 的近似值
- 改进的欧拉公式用二个点的斜率值 k_1 和 k_2 的平均值作为平均斜率 k^* 近似值；
- 改进的欧拉法比显式欧拉法精度高；
- 在 $[x_n, x_{n+1}]$ 这一步内多预报几个点的斜率值，并用其加权平均值作为平均斜率 k^* 的近似值，从而构造出具有更高精度的计算公式，这就是**龙格–库塔方法的基本思想**。

二阶龙格-库塔方法

- 以 k_1 和 k_2 的加权平均来近似取代 k^*

$$\begin{cases} y_{n+1} = y_n + h(c_1 k_1 + c_2 k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + a_2 h, y_n + b_{21} h k_1) \end{cases}$$

c_1, c_2, a_2, b_{21} 为待定系数

$(0 < a_2 \le 1)$

- 为分析局部截断误差，令 $y_n = y(x_n)$，由泰勒公式得：

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

$$y'(x_n) = f(x_n, y(x_n)) = \boxed{f(x_n, y_n)}$$

$$y''(x_n) = f'(x_n, y_n) = \boxed{f_x(x_n, y_n) + f_y(x_n, y_n) \underbrace{\boxed{f(x_n, y_n)}}_{y'(x_n)}}$$

$$y(x_{n+1}) = y(x_n) + hf(x_n, y_n) + \frac{h^2}{2}[f_x(x_n, y_n) + f_y(x_n, y_n) f(x_n, y_n)] + O(h^3)$$

补充：二元泰勒展开式

$$f(x_0+h,\ y_0+k)=f(x_0,y_0)$$

$$+\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)f(x_0,y_0)$$

$$h f_x(x_0,y_0)+k f_y(x_0,y_0)$$

$$+\frac{1}{2!}\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)^2 f(x_0,y_0)$$

$$h^2 f_{xx}(x_0,y_0)+2hk f_{xy}(x_0,y_0)$$

$$+k^2 f_{yy}(x_0,y_0)$$

$$\vdots$$

$$+\frac{1}{n!}\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)^n f(x_0,y_0)$$

$$\sum_{i=0}^{n}C_n^i h^i k^{n-i}\frac{\partial^n f}{\partial x^i\partial y^{n-i}}\Bigg|_{\substack{x=x_0\\ y=y_0}}$$

$$+\cdots$$

$$y(x_{n+1}) = y(x_n) + hf(x_n, y_n) + \frac{h^2}{2}[f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n)] + O(h^3)$$

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

- 用二元泰勒公式展开 $k_2 = f(x_n + a_2 h, y_n + b_{21} h k_1)$

$$k_2 = f(x_n, y_n) + a_2 h f_x(x_n, y_n) + b_{21} h k_1 f_y(x_n, y_n) + O(h^2)$$

- 将 k_1, k_2 代入 $y_{n+1} = y_n + h(c_1 k_1 + c_2 k_2)$ 中可得:

$$y_{n+1} = y_n + c_1 h f(x_n, y_n) + c_2 h[f(x_n, y_n) + a_2 h f_x(x_n, y_n) + b_{21} h k_1 f_y(x_n, y_n)] + O(h^3)$$

$$= y_n + (c_1 + c_2)h f(x_n, y_n) + c_2 a_2 h^2 f_x(x_n, y_n) + c_2 b_{21} h^2 f_y(x_n, y_n) f(x_n, y_n) + O(h^3)$$

$$= y_n + (c_1 + c_2)h f(x_n, y_n) + \frac{h^2}{2}[2c_2 a_2 f_x(x_n, y_n) + 2c_2 b_{21} f_y(x_n, y_n) f(x_n, y_n)] + O(h^3)$$

二阶龙格-库塔方法（续）

$$y_{n+1} = y_n + \boxed{(c_1 + c_2)} h f(x_n, y_n)$$

$$+\frac{h^2}{2}[\boxed{2c_2a_2} f_x(x_n, y_n) + \boxed{2c_2b_{21}} f_y(x_n, y_n) f(x_n, y_n)] + O(h^3)$$

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

$$+\frac{h^2}{2}[f_x(x_n, y_n) + f_y(x_n, y_n) f(x_n, y_n)] + O(h^3)$$

$$\begin{cases} c_1 + c_2 = 1 \\ 2c_2a_2 = 1 \\ 2c_2b_{21} = 1 \end{cases} \Rightarrow \quad y(x_{n+1}) - y_{n+1} = O(h^3)$$

2 阶精度

$$c_1 + c_2 = 1 \qquad 2c_2 a_2 = 1 \qquad 2c_2 b_{21} = 1$$

$$\begin{cases} y_{n+1} = y_n + h(c_1 k_1 + c_2 k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + a_2 h, y_n + b_{21} h k_1) \end{cases}$$

- 四个未知变量，只有三个方程，有无穷多组解
- 每组解构成的龙格-库塔方法均为二阶

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

二阶龙格-库塔方法即改进的欧拉方法

取 $\begin{cases} c_1 = 0 \\ c_2 = 1 \\ a_2 = b_{21} = 1/2 \end{cases}$

变形的欧拉法中点方法

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

三阶龙格-库塔方法

- 三阶龙格–库塔方法是用三个值 k_1, k_2, k_3 的加权平均来近似取代 k^*

$$\begin{cases} y_{n+1} = y_n + h(c_1k_1 + c_2k_2 + c_3k_3) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + a_2h, y_n + b_{21}hk_1) \\ k_3 = f(x_n + a_3h, y_n + b_{31}hk_1 + b_{32}hk_2) \end{cases}$$

$c_1, c_2, c_3, a_2, a_3,$
b_{21}, b_{31}, b_{32}
为常数
$(0 < a_2 < a_3 \le 1)$

- 要使三阶龙格–库塔方法具有三阶精度，必须使其局部截断误差为 $O(h^4)$
- 将 k_1, k_2, k_3 代入 y_{n+1} 的表达式中，在 (x_n, y_n) 处用二元泰勒公式展开，与 $y(x_{n+1})$ 在 x_n 处的泰勒展开式比较

三阶龙格-库塔方法（续）

- 类似二阶龙格–库塔方法的推导过程，8 个待定系数 $c_1, c_2, c_3, a_2, a_3, b_{21}, b_{31}, b_{32}$ 应满足：

$$
\begin{cases}
c_1 + c_2 + c_3 = 1 \\
a_2 = b_{21} \\
a_3 = b_{31} + b_{32} \\
c_2 a_2 + c_3 a_3 = 1/2 \\
c_2 a_2^2 + c_3 a_3^2 = 1/3 \\
c_2 b_{32} a_2 = 1/6
\end{cases}
$$

8 个未知参数，6 个方程，有无穷多组解

库塔公式

$$
\begin{cases}
y_{n+1} = y_n + h(\frac{1}{6}k_1 + \frac{4}{6}k_2 + \frac{1}{6}k_3) \\
k_1 = f(x_n, y_n) \\
k_2 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1) \\
k_3 = f(x_n + h, y_n - hk_1 + 2hk_2)
\end{cases}
$$

- 三阶龙格–库塔方法具有三阶精度

四阶龙格-库塔方法

- 类似可以推出四阶龙格-库塔公式，常用的有：

标准四阶龙格-库塔公式

$$
\begin{cases}
y_{n+1} = y_n + h(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4) \\
k_1 = f(x_n, y_n) \\
k_2 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1) \\
k_3 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2) \\
k_4 = f(x_n + h, y_n + hk_2)
\end{cases}
$$

四阶龙格-库塔方法（续）

吉尔（Gill）公式

$$\begin{cases} y_{n+1} = y_n + h\left(\frac{1}{6}k_1 + \frac{2-\sqrt{2}}{6}k_2 + \frac{2+\sqrt{2}}{6}k_3 + \frac{1}{6}k_4\right) \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\ k_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{\sqrt{2}-1}{2}hk_1 + \frac{2-\sqrt{2}}{2}hk_2\right) \\ k_4 = f\left(x_n + h, y_n - \frac{\sqrt{2}}{2}hk_2 + \frac{2+\sqrt{2}}{6}hk_3\right) \end{cases}$$

四阶龙格-库塔方法（续）

- 四阶龙格–库塔方法具有四阶精度。
- p 阶（$p\geq 2$）龙格-库塔公式的一般形式：

$$\begin{cases} y_{n+1}=y_n+h(c_1k_1+c_2k_2+\cdots+c_pk_p) \\ k_1=f(x_n,y_n) \\ k_2=f(x_n+a_2h,y_n+b_{21}hk_1) \\ \cdots \\ k_p=f(x_n+a_ph,y_n+b_{p1}hk_1+\cdots+b_{p,p-1}hk_{p-1}) \end{cases}$$

其中 a_i, b_{ij}, c_i 为待定系数。

- 分别求出上式 y_{n+1} 及 $y(x_{n+1})$ 在点 (x_n, y_n) 处的泰勒展开，通过比较 h 的相同次幂项的系数确定参数。

四阶龙格-库塔方法（续）

- 因为龙格-库塔方法的导出基于泰勒展开，因而精度主要受解函数的光滑性影响。如果解的光滑性不太好，使用 4 阶龙格-库塔方法求得的数值解，其精度可能反而不如改进的欧拉方法。
- 4 阶以上龙格-库塔方法的计算量太大，并且精度不一定提高，有时反而会降低，因此实际应用中一般选用 4 阶龙格-库塔已足可满足精度要求。

$$\begin{cases} y' = -y + x + 1 \\ y(0) = 1 \end{cases} \quad h = 0.1, \quad \text{计算到 } x = 0.5$$

- 用经典四阶龙格–库塔方法求解前例的初值问题，并与改进欧拉法、梯形法在 $x_5 = 0.5$ 处比较其误差大小

解：采用经典四阶龙格–库塔公式：

$$k_1 = f(x_0, y_0) = -y_0 + x_0 + 1 = 0$$

$$k_2 = f(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_1)$$
$$= -(y_0 + \frac{1}{2}hk_1) + (x_0 + \frac{1}{2}h) + 1 = 0.05$$

$$k_3 = f(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}hk_2)$$
$$= -(y_0 + \frac{1}{2}hk_2) + (x_0 + \frac{1}{2}h) + 1 = 0.0475$$

$$k_4 = f(x_0 + h, y_0 + hk_3)$$
$$= -(y_0 + hk_3) + (x_0 + h) + 1 = 0.09525$$

$$\begin{cases} y' = -y + x + 1 \\ y(0) = 1 \end{cases} \qquad h = 0.1,\quad 计算到\ x = 0.5$$

于是：$y_1 = y_0 + h\left(\frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4\right)$

$= 1 + 0.1 \times \left(\frac{1}{6} \times 0 + \frac{2}{6} \times 0.05 + \frac{2}{6} \times 0.0475 + \frac{1}{6} \times 0.09525\right)$

$= 1.0048375$

同理 $y_2 = 1.0187309$

可计 $y_3 = 1.04081842$

算： $y_4 = 1.07032029$

$y_5 = 1.1065309$

精确解为：

$y(x) = x + e^{-x}$

$y(x_5) = 0.5 + e^{-0.5} \approx 1.106530660$

四阶R-K方法的精度比二阶方法高得多

R-K方法的误差： $|y_5 - y(x_5)| = 2.4 \times 10^{-7}$

改进欧拉法的误差： 5.5×10^{-4}

梯形法的误差： 2.5×10^{-4}

变步长的龙格-库塔方法

- 用四阶R-K方法求解初值问题精度较高，但要从理论上给出误差 $|y(x_n)-y_n|$ 的估计式则比较困难；那么应如何判断计算结果的精度以及如何选择合适的步长 h？
- 通常是通过不同步长在计算机上的计算结果进行近似估计。
- 设 $y(x_n)$ 在 x_n 处的值 $y_n=y(x_n)$，当 $x_{n+1}=x_n+h$ 时 $y(x_{n+1})$ 的近似值记为 $y_{n+1}^{(h)}$，由于四阶 R-K 方法的精度为 4 阶，故局部截断误差为：

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

变步长的龙格-库塔方法（续）

- 若以 $h/2$ 为步长，从 x_n 出发，经过两步计算，得到 $y(x_{n+1})$ 的近似值 $y_{n+1}^{(h/2)}$
- 以上每步的截断误差约为 $c_n(h/2)^5$，于是两步的局部截断误差为：

$$T_{n+1} = y(x_{n+1}) - y_{n+1}^{(h/2)} \approx 2c_n(h/2)^5$$

于是：
$$\frac{y(x_{n+1}) - y_{n+1}^{(h/2)}}{y(x_{n+1}) - y_{n+1}^{(h)}} \approx \frac{2c_n(h/2)^5}{c_n h^5} = \frac{1}{16}$$

整理得：
$$y(x_{n+1}) - y_{n+1}^{(h/2)} \approx \frac{1}{15}\left[y_{n+1}^{(h/2)} - y_{n+1}^{(h)}\right]$$

变步长的龙格-库塔方法（续）

记：$\Delta=\left|\frac{1}{15}\left[y_{n+1}^{(h/2)}-y_{n+1}^{(h)}\right]\right|$，给定的精度要求为 ε

- $\Delta>\varepsilon$，反复将步长折半计算，直至 $\Delta<\varepsilon$，取最终得到的 $y_{n+1}^{(h/2)}$ 作为 $y(x_{n+1})$ 的近似值。
- $\Delta<\varepsilon$，反复将步长加倍计算，直至 $\Delta>\varepsilon$，再将步长折半一次计算，最终得到符合精度要求的 $y(x_{n+1})$ 的近似值。

这种通过加倍或折半处理步长的方法称为变步长的龙格-库塔方法。

单步法的收敛性

- 显式单步法可统一写成：

$\varphi(x, y, h)$

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

增量函数，仅依赖于函数 f，且仅仅是 x_n, y_n, h 的函数

数值解法的基本思路

求 $y = y(x)$

离散化

是否合理？

当 $h \to 0$ 时，近似解是否收敛到精确解？

求 $y(x_n)$，$x_n = x_0 + nh$

某种数值方法

求 $y_n \approx y(x_n)$

$x_n = x_0 + nh$，它应当是一个固定节点，因此 $h \to 0$ 时应同时附带 $n \to \infty$

单步法的收敛性（续）

- 对于 p 阶的常微分方程数值算法，当 $h \to 0$， $n \to \infty$ 时，是否 $y_{n+1} \to y(x_{n+1})$？
- p 阶算法的局部截断误差为：

$$y(x_{n+1}) - y_{n+1} = O(h^{p+1})$$

显然：

$$\lim_{h\to 0, n\to\infty} \left[y(x_{n+1}) - y_{n+1} \right] = 0$$

- 局部截断误差的前提假设是：$y_n = y(x_n)$
- 局部截断误差 $\to 0$ 并不能保证算法收敛

单步法的收敛性（续）

定义：若求解某初值问题的单步数值法，对于固定的 $x_n = x_0 + nh$ 当 $h \to 0$ 且 $n \to \infty$ 时，它的近似解趋向于精确解 $y(x_n)$，即：

$$\lim_{h\to 0, n\to\infty} y_n = y(x_n)$$

则称该**单步法是收敛的**。

定义：称 $\varepsilon_n = y(x_n) - y_n$ 为单步法的近似解 y_n 的**整体截断误差**。

单步法收敛 ⟺ $h \to 0$ 且 $n \to \infty$ 时，y_n 的整体截断误差 $\varepsilon_n \to 0$

单步法的收敛性（续）

收敛性定理

若单步法 $y_{n+1}=y_n+h\varphi(x_n,y_n,h)$ 具有 p 阶精度，且增量函数 $\varphi(x,y,h)$ 关于 y 满足：

- Lipschitz 条件：$\left|\varphi(x,y,h)-\varphi(x,\bar{y},h)\right|\le L_\varphi\left|y-\bar{y}\right|$
- 初值 y_0 是准确的

则该单步法的整体截断误差为：$y(x_n)-y_n=O(h^p)$

—— 单步法的局部截断误差与整体截断误差的关系。

若某单步法满足以上条件，则该方法是收敛的。

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h)$$
$$|\varphi(x,y,h) - \varphi(x,\bar{y},h)| \le L_\varphi |y - \bar{y}|$$

- 假设在前一步 y_n 准确的前提下求得的近似值为：

ε_{n+1}

$$\bar{y}_{n+1} = y(x_n) + h\varphi[x_n, y(x_n), h]$$

算法精度为 p 阶，局部截断误差：$\left|\bar{y}_{n+1} - y(x_{n+1})\right| \le ch^{p+1}$

$$\left|\boxed{y(x_{n+1}) - y_{n+1}}\right| = \left|[y(x_{n+1}) - \bar{y}_{n+1}] + [\bar{y}_{n+1} - y_{n+1}]\right|$$

$$\le \left|y(x_{n+1}) - \bar{y}_{n+1}\right| + \left|\bar{y}_{n+1} - y_{n+1}\right|$$

$$\le ch^{p+1} + \left|\{y(x_n) - y_n\} + \{h\varphi[x_n, y(x_n), h)] - h\varphi(x_n, y_n, h)\}\right|$$

$$\le ch^{p+1} + \left|y(x_n) - y_n\right| + h\left|\varphi[x_n, y(x_n), h)] - \varphi(x_n, y_n, h)\right|$$

$$\le ch^{p+1} + \left|y(x_n) - y_n\right| + hL_\varphi \left|y(x_n) - y_n\right|$$

ε_n

$$= (1 + hL_\varphi)\left|\boxed{y(x_n) - y_n}\right| + ch^{p+1}$$

$$|\varepsilon_{n+1}| \le (1+hL_\varphi)|\varepsilon_n| + ch^{p+1}$$

$$
\begin{aligned}
\left|\varepsilon_n\right| &\le (1+hL_\varphi)\left|\varepsilon_{n-1}\right| + ch^{p+1} \\
&\le (1+hL_\varphi)\left[(1+hL_\varphi)\left|\varepsilon_{n-2}\right| + ch^{p+1}\right] + ch^{p+1} \\
&= (1+hL_\varphi)^2\left|\varepsilon_{n-2}\right| + [(1+hL_\varphi)+1]\cdot ch^{p+1} \\
&\le (1+hL_\varphi)^2\left[(1+hL_\varphi)|\varepsilon_{n-3}| + ch^{p+1}\right] + \left[(1+hL_\varphi)+1\right]ch^{p+1} \\
&= (1+hL_\varphi)^3\left|\varepsilon_{n-3}\right| + [(1+hL_\varphi)^2 + (1+hL_\varphi)+1]\cdot ch^{p+1} \\
&\vdots \\
&= (1+hL_\varphi)^n\left|\varepsilon_0\right| + [(1+hL_\varphi)^{n-1} + \cdots + (1+hL_\varphi)+1]\cdot ch^{p+1} \\
&= (1+hL_\varphi)^n\left|\varepsilon_0\right| + \frac{1\cdot\left[(1+hL_\varphi)^n - 1\right]}{(1+hL_\varphi)-1}ch^{p+1} \\
&= (1+hL_\varphi)^n\left|\varepsilon_0\right| + \left[(1+hL_\varphi)^n - 1\right]\frac{ch^p}{L_\varphi}
\end{aligned}
$$

$$|\varepsilon_n| \le (1+hL_\varphi)^n |\varepsilon_0| + \left[(1+hL_\varphi)^n - 1\right]\frac{ch^p}{L_\varphi}$$

$$\mathrm{e}^x = 1 + x + \frac{\mathrm{e}^\xi}{2}x^2 \ge 1 + x$$ 即： $$\boxed{(1+x)^n \le \mathrm{e}^{nx}}$$

- $y = \mathrm{e}^x$ 为单调增函数，当 $x_n - x_0 = nh \le T$ 时

$$(1+hL_\varphi)^n \le \left(\mathrm{e}^{hL_\varphi}\right)^n = \mathrm{e}^{nhL_\varphi} \le \mathrm{e}^{TL_\varphi}$$

$$|\varepsilon_n| \le (1+hL_\varphi)^n |\varepsilon_0| + \frac{ch^p}{L_\varphi}\left(e^{TL_\varphi} - 1\right)$$

- 若初值是准确的，则 $\varepsilon_0 = 0$，从而整体截断误差为：

$$|\varepsilon_n| \le \frac{ch^p}{L_\varphi}\left(e^{TL_\varphi} - 1\right)$$ $$\boxed{O(h^p)}$$

- 当 $h \to 0$ 且 $n \to \infty$ 时，则 $|\varepsilon_n| \to 0$

单步法的收敛性（续）

- 对于欧拉法，由于其增量函数 φ 就是 f，故当 f 满足 Lipschitz 条件时，它是收敛的。
- 对于改进的欧拉法 $\varphi(x,y,h)=\frac{1}{2}\left[f(x,y)+f\big(x+h,y+hf(x,y)\big)\right]$

$$|\varphi(x,y,h)-\varphi(x,\bar{y},h)|\le\frac{1}{2}[|f(x,y)-f(x,\bar{y})|$$
$$+|f\big(x+h,y+hf(x,y)\big)-f(x+h,\bar{y}+hf(x,\bar{y}))|]$$

假设 f 关于 y 满足 Lipschitz 条件，记 Lipschitz 常数为 L，则由上式推得

$$|\varphi(x,y,h)-\varphi(x,\bar{y},h)|\le L(1+\frac{h}{2}L)|y-\bar{y}|$$

设限定 $h\le h_0$（h_0 为定数），上式表明 φ 关于 y 的 Lipschitz 常数 $L_\varphi=L(1+\frac{h_0}{2}L)$。　　收敛

单步法的稳定性

- 在讨论单步法收敛性时一般认为数值方法本身的计算过程是准确的，实际上并非如此：
  - 初始值 y_0 有误差 $\delta = y_0 - y(x_0)$
  - 后续的每一步计算均有舍入误差
- 这些初始和舍入误差在计算过程的传播中是逐步衰减的还是恶性增长就是数值方法的稳定性问题。

单步法的稳定性（续）

定义：设在节点 x_n 处用数值算法得到的理想数值解为 y_n，而实际计算得到的近似解为 $\tilde{y}_n$，称差值：

$$\delta_n = \tilde{y}_n - y_n$$

为第 n 步的数值解的扰动。

定义：若一种数值方法在节点 x_n 处的数值解 y_n 的扰动 $\delta_n \neq 0$，而在以后各节点 y_m（$m > n$）上产生的扰动为 δ_m，如果：

$$|\delta_m| \le |\delta_n| \qquad (m = n+1, n+2, \cdots)$$

则称该数值方法是稳定的。

单步法的稳定性（续）

- 由于函数 $f(x, y)$ 的多样性，数值稳定性的分析相当复杂，通常只研究模型方程 $y' = \lambda y$（$\lambda < 0$）。
  - 欧拉法： $y_{n+1} = y_n + hf(x_n, y_n)$

    考察模型方程：

    $$y' = \lambda y \quad (\lambda < 0)$$

    即：

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

    假设在节点值 y_n 上有扰动 δ_n，在节点值 y_{n+1} 上有扰动 δ_{n+1}，且 δ_{n+1} 仅由 δ_n 引起（即：计算过程中不再引起新的误差）

针对模型方程： $y' = \lambda y$（$\lambda < 0$）

的显式欧拉法：$y_{n+1} = (1 + h\lambda)y_n$

$$\begin{aligned}\delta_{n+1} &= \tilde{y}_{n+1} - y_{n+1} \\ &= (1+h\lambda)\tilde{y}_n - (1+h\lambda)y_n \\ &= (1+h\lambda)(\tilde{y}_n - y_n) \\ &= (1+h\lambda)\delta_n\end{aligned}$$

欧拉法稳定 $\longleftrightarrow$ $|\delta_{n+1}| \le |\delta_n|$ $\longleftrightarrow$ $|1+h\lambda| \le 1$

即： $-1 \le 1 + h\lambda \le 1$

化简得： $-2 \le h\lambda \le 0$ $\quad \lambda < 0$

欧拉法稳定的条件：

$$\boxed{0 < h \le -\frac{2}{\lambda}}$$

欧拉法条件稳定

隐式欧拉法：$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$

考察模型方程： $y' = \lambda y \quad (\lambda < 0)$

即： $y_{n+1} = y_n + h\lambda y_{n+1}$

化简为： $y_{n+1} = \dfrac{y_n}{1 - h\lambda}$

假设 y_n 上有扰动 $\delta_n = \tilde{y}_n - y_n$，则 y_{n+1} 的扰动为：

$$\delta_{n+1} = \tilde{y}_{n+1} - y_{n+1} = \frac{\tilde{y}_n}{1-h\lambda} - \frac{y_n}{1-h\lambda} = \frac{\delta_n}{1-h\lambda}$$

隐式欧拉法稳定 $\longleftrightarrow$ $|\delta_{n+1}| \le |\delta_n|$ $\longleftrightarrow$ $\left|\dfrac{1}{1-h\lambda}\right| \le 1$

$\lambda < 0$

$\forall h > 0$，上式均成立，所以：

隐式欧拉法无条件稳定

例题

考察初值问题

$$\begin{cases} y' = -100y \\ y(0) = 1 \end{cases}$$

其准确解 $y = \mathrm{e}^{-100x}$ 按指数曲线衰减很快。
为保证欧拉法稳定，步长 h 应不超过 0.02。

$$0 < h \leq -\frac{2}{\lambda}$$

取 $h = 0.025$，用欧拉法和隐式欧拉法计算如下

节点	欧拉法	隐式欧拉法	精确值
0.025	–1.5	0.2857	0.082
0.050	2.25	0.0816	0.0067
0.075	–3.375	0.0233	
0.100	5.0625	0.0067	

一阶常微分方程组

$$\begin{cases} y_1' = f_1(x, y_1, y_2, \cdots, y_m) \\ y_2' = f_2(x, y_1, y_2, \cdots, y_m) \\ \quad \cdots\cdots \\ y_m' = f_m(x, y_1, y_2, \cdots, y_m) \\ y_1(x_0) = s_1, y_2(x_0) = s_2, \cdots, y_m(x_0) = s_m \end{cases}$$

$$\begin{cases} Y'(x) = F(x, Y^T) \\ Y(x_0) = S \end{cases}$$

对单个方程 $y' = f$ 的数值解法，将 y 和 f 理解为向量，可推广到一阶常微分方程组的情形。

$$Y(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_m(x) \end{pmatrix} \quad F(x, Y^T) = \begin{pmatrix} f_1(x, Y^T) \\ f_2(x, Y^T) \\ \vdots \\ f_m(x, Y^T) \end{pmatrix} \quad S = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix}$$

$$Y(x)=\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} \quad F(x,Y^T)=\begin{pmatrix} f_1(x,Y^T) \\ f_2(x,Y^T) \end{pmatrix} \quad S=\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

$$\begin{cases} y_1' = f_1(x,y_1,y_2) & y_1(x_0)=s_1 \\ y_2' = f_2(x,y_1,y_2) & y_2(x_0)=s_2 \end{cases}$$

- 显式欧拉法

$$Y_{n+1}=Y_n+hF(x_n,Y_n^T) \quad \begin{pmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{pmatrix}=\begin{pmatrix} y_{1_n} \\ y_{2_n} \end{pmatrix}+h\begin{pmatrix} f_1(x_n,y_{1_n},y_{2_n}) \\ f_2(x_n,y_{1_n},y_{2_n}) \end{pmatrix}$$

- 隐式欧拉法

$$Y_{n+1}=Y_n+hF(x_{n+1}, Y^T{}_{n+1}) \quad \begin{pmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{pmatrix}=\begin{pmatrix} y_{1_n} \\ y_{2_n} \end{pmatrix}+h\begin{pmatrix} f_1(x_{n+1},y_{1_{n+1}},y_{2_{n+1}}) \\ f_2(x_{n+1},y_{1_{n+1}},y_{2_{n+1}}) \end{pmatrix}$$

- 梯形公式欧拉法

$$\begin{pmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{pmatrix}=\begin{pmatrix} y_{1_n} \\ y_{2_n} \end{pmatrix}+\frac{h}{2}\left[\begin{pmatrix} f_1(x_n,y_{1_n},y_{2_n}) \\ f_2(x_n,y_{1_n},y_{2_n}) \end{pmatrix}+\begin{pmatrix} f_1(x_{n+1},y_{1_{n+1}},y_{2_{n+1}}) \\ f_2(x_{n+1},y_{1_{n+1}},y_{2_{n+1}}) \end{pmatrix}\right]$$

$$Y(x)=\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} \quad F(x,Y^T)=\begin{pmatrix} f_1(x,Y^T) \\ f_2(x,Y^T) \end{pmatrix} \quad S=\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

- 改进的欧拉公式

$$\begin{cases} Y_p = Y_n + hF(x_n, Y_n^T) \\ Y_c = Y_n + hF(x_{n+1}, Y_p^T) \\ Y_{n+1} = (Y_p + Y_c)/2 \end{cases}$$

$$\begin{pmatrix} y_{p1_{n+1}} \\ y_{p2_{n+1}} \end{pmatrix} = \begin{pmatrix} y_{1_n} \\ y_{2_n} \end{pmatrix} + h\begin{pmatrix} f_1(x_n, y_{1_n}, y_{2_n}) \\ f_2(x_n, y_{1_n}, y_{2_n}) \end{pmatrix}$$

$$\begin{pmatrix} y_{c1_{n+1}} \\ y_{c2_{n+1}} \end{pmatrix} = \begin{pmatrix} y_{1_n} \\ y_{2_n} \end{pmatrix} + h\begin{pmatrix} f_1(x_{n+1}, y_{p1_{n+1}}, y_{p2_{n+1}}) \\ f_2(x_{n+1}, y_{p1_{n+1}}, y_{p2_{n+1}}) \end{pmatrix}$$

$$\begin{pmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{pmatrix} = \frac{1}{2}\left[\begin{pmatrix} y_{p1_{n+1}} \\ y_{p2_{n+1}} \end{pmatrix} + \begin{pmatrix} y_{c1_{n+1}} \\ y_{c2_{n+1}} \end{pmatrix}\right]$$

$$Y(x)=\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} \quad F(x,Y^T)=\begin{pmatrix} f_1(x,Y^T) \\ f_2(x,Y^T) \end{pmatrix} \quad S=\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

- 四阶龙格-库塔公式

$$\begin{cases} Y_{n+1}=Y_n+\frac{1}{6}(K_1+2K_2+2K_3+K_4) \\ K_1=hF(x_n,Y_n^T) \\ K_2=hF(x_{n+\frac{1}{2}},\frac{1}{2}K_1^T) \\ K_3=hF(x_{n+\frac{1}{2}},\frac{1}{2}K_2^T) \\ K_4=hF(x_{n+1},K_3^T) \end{cases}$$

$$\begin{pmatrix} k_{1_1} \\ k_{2_1} \end{pmatrix}=h\begin{pmatrix} f_1(x_n,y_{1_n},y_{2_n}) \\ f_2(x_n,y_{1_n},y_{2_n}) \end{pmatrix}$$

$$\begin{pmatrix} k_{1_2} \\ k_{2_2} \end{pmatrix}=h\begin{pmatrix} f_1(x_{n+\frac{1}{2}},\frac{1}{2}k_{1_1},\frac{1}{2}k_{2_1}) \\ f_2(x_{n+\frac{1}{2}},\frac{1}{2}k_{1_1},\frac{1}{2}k_{2_1}) \end{pmatrix}$$

$$\begin{pmatrix} k_{1_3} \\ k_{2_3} \end{pmatrix}=h\begin{pmatrix} f_1(x_{n+\frac{1}{2}},\frac{1}{2}k_{1_2},\frac{1}{2}k_{2_2}) \\ f_2(x_{n+\frac{1}{2}},\frac{1}{2}k_{1_2},\frac{1}{2}k_{2_2}) \end{pmatrix}$$

$$\begin{pmatrix} k_{1_4} \\ k_{2_4} \end{pmatrix}=h\begin{pmatrix} f_1(x_{n+1},k_{1_3},k_{2_3}) \\ f_2(x_{n+1},k_{1_3},k_{2_3}) \end{pmatrix}$$

$$\begin{pmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{pmatrix}=\begin{pmatrix} y_{1_n} \\ y_{2_n} \end{pmatrix}+\frac{1}{6}\left[\begin{pmatrix} k_{1_1} \\ k_{2_1} \end{pmatrix}+2\begin{pmatrix} k_{1_2} \\ k_{2_2} \end{pmatrix}+2\begin{pmatrix} k_{1_3} \\ k_{2_3} \end{pmatrix}+\begin{pmatrix} k_{1_4} \\ k_{2_4} \end{pmatrix}\right]$$

高阶微分方程的初值问题

- 一般通过引入新的变量，将高阶微分方程化为一阶微分方程组的方法进行求解
- m 阶常微分方程：

$$\begin{cases} y^{(m)} = f\left(x, y, y', y'', y''', \cdots, y^{(m-1)}\right) \\ y(x_0) = c_0, y'(x_0) = c_1, \cdots, y^{(m-1)}(x_0) = c_{m-1} \end{cases}$$

- 用变量替换可以将上述的高阶微分方程转化为一阶微分方程组

设： $y_0 = y, y_1 = y', y_2 = y'', \cdots, y_{m-1} = y^{(m-1)}$

$$y_0 = y, y_1 = y', y_2 = y'', \cdots, y_{m-1} = y^{(m-1)}$$

则 m 阶微分方程

$$\begin{cases} y^{(m)} = f\left(x, y, y', y'', y''', \cdots, y^{(m-1)}\right) \\ y(x_0) = c_0, y'(x_0) = c_1, \cdots, y^{(m-1)}(x_0) = c_{m-1} \end{cases}$$

转化为如下的一阶微分方程组：

$$\begin{cases} y_0' = y_1 \\ y_1' = y_2 \\ \quad \vdots \\ y_{m-2}' = y_{m-1} \\ y_{m-1}' = f(x, y_0, y_1, \cdots, y_{m-1}) \end{cases}$$

初始条件为：

$$\begin{cases} y_0(x_0) = c_0 \\ y_1(x_0) = c_1 \\ \quad \vdots \\ y_{m-1}(x_0) = c_{m-1} \end{cases}$$

转化例子

将 $\begin{cases} y'' + \sin y = 0 \quad (0 \le x \le 2) \\ y(0) = 1, \quad y'(0) = 0 \end{cases}$ 转化为微分方程组。

解： 令 $y_0 = y, \quad y_1 = y'$，有：

$$\begin{cases} y_0' = \boxed{y_1} \\ y_1' = \boxed{-\sin y_0} \end{cases}$$

$f_0(x, y_0, y_1)$

$f_1(x, y_0, y_1)$

初始条件：

$$\begin{cases} y_0(0) = 1 \\ y_1(0) = 0 \end{cases}$$

$$Y(x)=\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} \quad F(x,Y^T)=\begin{pmatrix} f_1(x,Y^T) \\ f_2(x,Y^T) \end{pmatrix} \quad \begin{cases} y_0' = \underbrace{y_1}_{f_0(x,y_0,y_1)} \\ y_1' = \underbrace{-\sin y_0}_{f_1(x,y_0,y_1)} \end{cases}$$

$$\begin{pmatrix} k_{0_1} \\ k_{1_1} \end{pmatrix} = h\begin{pmatrix} f_0(x_n, y_{0_n}, y_{1_n}) \\ f_1(x_n, y_{0_n}, y_{1_n}) \end{pmatrix} = h\begin{pmatrix} y_{1_n} \\ -\sin y_{0_n} \end{pmatrix}$$

$$\begin{pmatrix} k_{0_2} \\ k_{1_2} \end{pmatrix} = h\begin{pmatrix} f_0(x_{n+\frac{1}{2}}, \frac{1}{2}k_{0_1}, \frac{1}{2}k_{1_1}) \\ f_1(x_{n+\frac{1}{2}}, \frac{1}{2}k_{0_1}, \frac{1}{2}k_{1_1}) \end{pmatrix} = h\begin{pmatrix} -\frac{1}{2}h\sin y_{0_n} \\ -\sin(\frac{1}{2}hy_{1_n}) \end{pmatrix}$$

$$\begin{pmatrix} k_{0_3} \\ k_{1_3} \end{pmatrix} = h\begin{pmatrix} f_0(x_{n+\frac{1}{2}}, \frac{1}{2}k_{0_2}, \frac{1}{2}k_{1_2}) \\ f_1(x_{n+\frac{1}{2}}, \frac{1}{2}k_{0_2}, \frac{1}{2}k_{1_2}) \end{pmatrix} = h\begin{pmatrix} -\frac{1}{2}h\sin(\frac{1}{2}hy_{1_n}) \\ -\sin(-\frac{1}{4}h^2\sin y_{0_n}) \end{pmatrix}$$

$$\begin{pmatrix} k_{0_4} \\ k_{1_4} \end{pmatrix} = h\begin{pmatrix} f_0(x_{n+1}, k_{0_3}, k_{1_3}) \\ f_1(x_{n+1}, k_{0_3}, k_{1_3}) \end{pmatrix} = h\begin{pmatrix} -h\sin(-\frac{1}{4}h^2\sin y_{0_n}) \\ -\sin(-\frac{1}{2}h^2\sin(\frac{1}{2}hy_{1_n})) \end{pmatrix}$$

$$\begin{pmatrix} y_{0_{n+1}} \\ y_{1_{n+1}} \end{pmatrix} = \begin{pmatrix} y_{0_n} \\ y_{1_n} \end{pmatrix} + \frac{1}{6}\left[\begin{pmatrix} k_{0_1} \\ k_{1_1} \end{pmatrix} + 2\begin{pmatrix} k_{0_2} \\ k_{1_2} \end{pmatrix} + 2\begin{pmatrix} k_{0_3} \\ k_{1_3} \end{pmatrix} + \begin{pmatrix} k_{0_4} \\ k_{1_4} \end{pmatrix}\right]$$

本章小结

- 欧拉方法
- 局部截断误差
- 改进的欧拉方法
- 龙格-库塔方法

  重点二阶，经典四阶
- 收敛性与稳定性

  记住判定定理，会用于判断算法的收敛性和稳定性

- 欧拉方法
  - （显式）欧拉法
  - 隐式欧拉法
  - 二步欧拉法
  - 梯形公式欧拉法

作业

课本 P206：

1，3，4，6，9，12