第六章 常微分方程数值解法

哈尔滨工业大学数学学院
http://homepage.hit.edu.cn/LiuWenjie

§6.1 引言

微分方程 (包括常微分方程和偏微分方程) 主要用来描述宏观和微观世界的基本运动规律, 在物理、化学、生物、经济等多个领域发挥着广泛的作用.

常微分方程初值问题

$$\begin{cases} y'(x) = f(x,y), & a < x \le b \\ y(a) = y_0 \end{cases} \tag{1}$$

定义 如果存在实数 $L > 0$, 使得

$$|f(x,y_1) - f(x,y_2)| \le L\,|y_1 - y_2|, \quad y_1, y_2 \in \mathbb{R}$$

则称 f 关于 y 满足利普希茨 (Lipschitz) 条件, L 称为 f 的利普希茨常数.

定理 当 f 在区域 $D = \{(x,y) | a \le x \le b, y \in \mathbb{R}\}$ 上连续, 且关于 y 满足利普希茨条件时, 对任意 $x_0 \in [a,b], y_0 \in \mathbb{R}$, 常微分方程初值问题(1)式当 $x \in [a,b]$ 时存在唯一的连续可微解 $y(x)$.

§6.1 引言: 数值解法

取一系列点

$$x_0 = a < x_1 < \cdots < x_n < \cdots$$

求相应点的近似值

$$y(x_0) = y_0, y(x_1) \approx y_1, \ldots, y(x_n) \approx y_n, \ldots$$

$y_0, y_1, \ldots, y_n, \ldots$ 称为数值解. $h = x_n - x_{n-1}$ 称为步长. 用离散化方法建立求 $y(x_n)$ 的近似值 y_n 的递推格式, 由此求得解 $y(x)$ 在各节点上的近似值

这种数值解法分为两大类：

(i) 单步法：若求 y_{n+1}，只需利用它前一步的信息 y_n，则称这种方法为单步法。它由 y_0 出发，可求得 $y_1, y_2, y_3 \cdots$

(ii) 多步法：若求 y_{n+1}，需利用它前面至少两个点的信息，$y_n, y_{n-1}, \ldots$，则称这种方法为多步法.

§6.1 引言: 数值解法研究的主要问题

(i) 方法的推导：采用的离散化手段，精度准则.

(ii) 收敛性：差分方程的解是否充分逼近初值问题的解.

(iii) 稳定性：初始数据、计算过程中每步产生的误差对以后各步解的影响，这种误差传播是否可控制、甚至是衰减的.

常用离散化方法:

(i) 数值微分

(ii) 数值积分

(iii) Taylor 展开

§6.2.1 Euler 法

把区间 $[a,b]$ 做 N 等分, 称均匀网格，网格点

$$x_i = a + ih, \quad i = 0,1,2,\ldots,N$$

其中 $h = (b-a)/N = x_{i+1} - x_i$ 为步长。

Taylor 展开法

将 $y\left(x_{n+1}\right)$ 在 x_n 处 Taylor 展开

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

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

得差分方程

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

称此方法为 Euler 法, 显格式.

§6.2.1 Euler 法

数值微分 利用化导数为差商的方法

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

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

得差分方程

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

§6.2.1 Euler 法

利用数值积分的方法

在 $[x_n, x_{n+1}]$ 上对 $y'(x) = f(x, y(x))$ 积分得

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

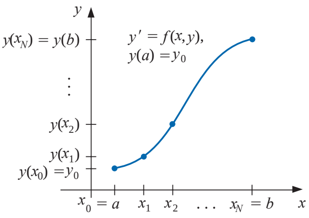
用左矩形求积公式计算定积分有

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

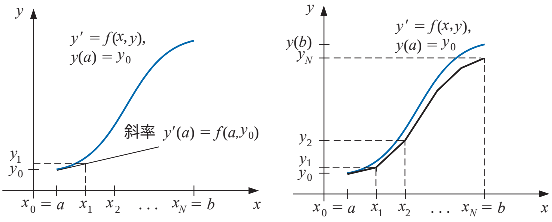
以此得差分方程

$$y_{n+1} = y_n + h f\left(x_n, y_n\right) \quad (n = 0, 1, 2, \ldots)$$

§6.2.1 Euler 法



(a) 方程



(b) Euler 方法

§6.2.2 后退 Euler 法

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

称此方法为后退 Euler 法, 隐式格式.
运用它常采用下面的迭代格式:$n=0,1,2,\ldots$

$$\begin{cases} y_{n+1}^{(0)}=y_n+hf\left(x_n,y_n\right) \\ y_{n+1}^{(k+1)}=y_n+hf\left(x_{n+1},y_{n+1}^{(k)}\right), \quad k=0,1,2,\ldots \end{cases}$$

或者使用 Newton 迭代法求解关于 y_{n+1} 非线性方程: $n=0,1,2,\ldots$

$$\begin{cases} g(y_{n+1})=y_{n+1}-y_n-hf\left(x_{n+1},y_{n+1}\right) \\ y_{n+1}^{(0)}=y_n+hf\left(x_n,y_n\right) \\ y_{n+1}^{(k+1)}=y_{n+1}^{(k)}-\dfrac{g(y_{n+1}^{(k)})}{g'(y_{n+1}^{(k)})}, \quad k=0,1,2,\ldots \end{cases}$$

§6.2.3 梯形方法

用不同的近似公式计算定积分的值, 就得到解初值问题的不同数值解法.
用梯形求积公式计算积分得

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

这个方法称为梯形法. 它是隐格式.
运用它常采用下面的迭代格式:$n=0,1,2,\ldots$

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

显格式与隐格式

如果每个迭代步不需要解方程, 则称为 **显格式**, 反之, 如果每个迭代步需要解方程, 则称为 **隐格式**. Euler 法是显式的, 而梯形法是隐式的.

§6.2.4 改进的 Euler 方法

若梯形法只迭代一次, 便得改进的 Euler 方法

$$\begin{cases} \bar{y}_{n+1} = y_n + hf\left(x_n, y_n\right) \\ y_{n+1} = y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(x_{n+1}, \bar{y}_{n+1}\right)\right] \end{cases} \qquad (n = 0, 1, 2, \ldots)$$

预估与校正

改进的 Euler 法中的第一步称为**预估 (Predictor)**, 第二步称为**校正 (corrector)**. **预估校正**是常微分方程数值求解的重要手段之一.

改进的 Euler 方法也可写成形式

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2}hK_1 + \frac{1}{2}hK_2 \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + h, y_n + hK_1\right) \end{cases}$$

§6.2.4 单步法的局部截断误差与阶

单步法一般形式表示为

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

其中多元函数 φ 与 $f(x,y)$ 有关，当 φ 含有 y_{n+1} 时, 方法是隐式的，若 φ 中不含 y_{n+1} 则为显式方法，所以显式单步法可表示为

$$y_{n+1} = y_n + h\varphi\left(x_n, y_n, h\right) \tag{2}$$

$\varphi\left(x, y, h\right)$ 称为增量函数

定义 设 $y(x)$ 是初值问题(1)式的准确解，称

$$T_{n+1} = y\left(x_{n+1}\right) - y\left(x_n\right) - h\varphi\left(x_n, y\left(x_n\right), h\right)$$

为显式单步法(2)式的局部截断误差.

T_{n+1} 之所以称为局部的，是假设在 x_n 前各步没有误差. 当时 $y_n = y(x_n)$，计算一步，则有

$$\begin{aligned} y\left(x_{n+1}\right) - y_{n+1} &= y\left(x_{n+1}\right) - \left[y_n + h\varphi\left(x_n, y_n, h\right)\right] \\ &= y\left(x_{n+1}\right) - y\left(x_n\right) - h\varphi\left(x_n, y\left(x_n\right), h\right) = T_{n+1} \end{aligned}$$

§9.2.4 单步法的局部截断误差与阶

定义 设 $y(x)$ 是初值问题(1)式的准确解，若存在最大整数 p 使显式单步法(2)式的局部截断误差满足

$$T_{n+1} = y(x_n + h) - y(x_n) - h\varphi(x_n, y(x_n), h) = \mathcal{O}\left(h^{p+1}\right)$$

则称方法(2)具有 p 阶精度.

由 Taylor 展开式知

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

对于 Euler 法

$$\begin{aligned} T_{n+1} &= y\left(x_{n+1}\right) - y\left(x_n\right) - hf\left(x_n, y\left(x_n\right)\right) \\ &= y\left(x_{n+1}\right) - y\left(x_n\right) - hy'\left(x_n\right) \\ &= \frac{h^2}{2!}y''\left(\xi_n\right) \end{aligned}$$

Euler 法是 1 阶方法

§6.2.4 单步法的局部截断误差与阶

$$
\begin{cases}
y_{n+1}=y_n+\frac{1}{2}hK_1+\frac{1}{2}hK_2\\
K_1=f\left(x_n,y_n\right)\\
K_2=f\left(x_n+h,y_n+hK_1\right)
\end{cases}
$$

对于改进 Euler 法，设 $y_n=y\left(x_n\right)$，利用 Taylor 展开式

$$
\begin{aligned}
K_1&=f\left(x_n,y\left(x_n\right)\right)=y'\left(x_n\right)\\
K_2&=f\left(x_n+h,y\left(x_n\right)+hK_1\right)\\
&=f\left(x_n,y\left(x_n\right)\right)+h\frac{\partial}{\partial x}f\left(x_n,y\left(x_n\right)\right)+hK_1\frac{\partial}{\partial y}f\left(x_n,y\left(x_n\right)\right)+\cdots\\
&=f\left(x_n,y\left(x_n\right)\right)+h\left[\frac{\partial}{\partial x}f\left(x_n,y\left(x_n\right)\right)+y'\left(x_n\right)\frac{\partial}{\partial y}f\left(x_n,y\left(x_n\right)\right)\right]\\
&\quad+\cdots\\
&=y'\left(x_n\right)+hy''\left(x_n\right)+\cdots
\end{aligned}
$$

§9.2.4 单步法的局部截断误差与阶

将其代入中 T_{n+1} 有

$$T_{n+1} = y\left(x_{n+1}\right) - y\left(x_n\right) - \frac{h}{2}\left(K_1 + K_2\right) = \mathcal{O}(h^3)$$

改进 Euler 法是 2 阶方法.

§9.2.4 单步法的局部截断误差与阶

Taylor 展开式

$$
\begin{aligned}
y\left(x_{n+1}\right) = y\left(x_n + h\right) &= y\left(x_n\right) + h y'\left(x_n\right) + \frac{h^2}{2!} y''\left(x_n\right) + \cdots \\
&= y\left(x_n\right) + h f\left(x_n, y\left(x_n\right)\right) \\
&+ \frac{h^2}{2!}\left[\frac{\partial}{\partial x} f\left(x_n, y\left(x_n\right)\right) + f\left(x_n, y\left(x_n\right)\right)\frac{\partial}{\partial y} f\left(x_n, y\left(x_n\right)\right)\right] + \cdots
\end{aligned}
$$

理论上讲，只要解 $y(x)$ 充分光滑，通过保留 Taylor 展开式的若干项就可得到任意阶的近似公式

§9.2.4 单步法的局部截断误差与阶

Euler 法也可写成形式

$$\begin{cases} y_{n+1} = y_n + hK_1 \\ K_1 = f\left(x_n, y_n\right) \end{cases}$$

其局部截断误差为 $\mathcal{O}\left(h^2\right)$, 是一阶方法.
对于改进 Euler 法也可写成形式

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2}hK_1 + \frac{1}{2}hK_2 \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + h, y_n + hK_1\right) \end{cases}$$

局部截断误差为 $\mathcal{O}\left(h^3\right)$，是二阶方法.

s 级 Runge-Kutta 法的一般形式

$$
\begin{cases}
y_{n+1} = y_n + h \sum\limits_{i=1}^{s} b_i K_i \\
K_i = f\left(x_n + c_i h, y_n + h \sum\limits_{j=1}^{s} a_{ij} K_j\right), \quad i = 1, 2, \ldots, s
\end{cases}
$$

其中 $h = x_{n+1} - x_n$,a_{ij}, b_i, c_i 都是常数.
这些常数可以用下面 Butcher 矩阵表示

$$
\begin{array}{c|cccc}
c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\
c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\
\vdots & \vdots & \vdots & & \vdots \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\
\hline
 & b_1 & b_2 & \ldots & b_s
\end{array}
$$

§6.3 显式 Runge-Kutta 方法

s 级显式 Runge-Kutta 法的一般形式

$$
\begin{cases}
y_{n+1} = y_n + h\sum\limits_{i=1}^{s} b_i K_i \\
K_1 = f(x_n, y_n) \\
K_i = f\left(x_n + c_i h, y_n + h\sum\limits_{j=1}^{i-1} a_{ij} K_j\right), \quad i = 2, 3, \ldots, s
\end{cases}
$$

其中 $h = x_{n+1} - x_n$,a_{ij}, b_i, c_i 都是常数.$c_1 = 0, a_{ij} = 0 \ j \geq i$. Butcher 矩阵表示

$$
\begin{array}{c|ccccc}
0 & 0 & 0 & \cdots & 0 & 0 \\
c_2 & a_{2,1} & 0 & \cdots & 0 & 0 \\
c_3 & a_{3,1} & a_{3,2} & \ddots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & 0 & 0 \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

§6.3 显式 Runge-Kutta 方法: 一级方法

$s=1$ 计算公式为

$$y_{n+1}=y_n+hb_1K_1,\quad K_1=f(x_n,y_n)$$

这个公式和 $y(x_{n+1})$ 比较:

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

由于 $y'=f(x,y)$ 得到

$$y''(x)=f_x+y'(x)f_y=f_x+ff_y$$

所以

$$y(x+h)=y(x)+hf+\frac{1}{2!}h^2\left(f_x+ff_y\right)+\mathcal{O}\left(h^3\right)$$

假设 $x=x_n$ 时 $y_n=y(x_n), f(x_n,y_n)=f(x_n,y(x_n))$

$$\begin{aligned}T_{n+1}&=y\left(x_{n+1}\right)-y_{n+1}\\&=h\left(1-b_1\right)f\left(x_n,y_n\right)+\frac{1}{2!}h^2\left(f_x+ff_y\right)\Big|_{x=x_n}+\mathcal{O}\left(h^3\right)\end{aligned}$$

选取 $b_1=1$，精度为 1 阶, 此时 1 级显示 Runge-Kutta 方法为 Euler 法

二级 Runge-Kutta 公式一般形式

$$\left\{\begin{array}{l} y_{n+1}=y_n+b_1hK_1+b_2hK_2 \\ K_1=f\left(x_n,y_n\right) \\ K_2=f\left(x_n+c_2h,y_n+a_{21}hK_1\right) \end{array}\right.$$

其中 a_{21},b_1,b_2,c_2 为待定常数.
局部截断误差为: 此时假设 $y_n=y(x_n)$

$$\begin{aligned} T_{n+1} &= y\left(x_{n+1}\right)-y_{n+1}=y\left(x_{n+1}\right)-y\left(x_n\right)-b_1hK_1-b_2hK_2 \\ &= y\left(x_{n+1}\right)-y\left(x_n\right)-h\left[b_1f_n+b_2f\left(x_n+c_2h,y_n+a_{21}hf_n\right)\right] \end{aligned}$$

其中 $y_n=y\left(x_n\right),f_n=f\left(x_n,y_n\right)$.

§6.3 显式 Runge-Kutta 方法: 二级方法

把 K_2 中 f 在 $(x_n, y(x_n))$ 处泰勒展开

$$\begin{aligned}
K_2 =& f\left(x_n + c_2 h, y(x_n) + a_{21} h f_n\right) \\
=& f_n + c_2 h \frac{\partial}{\partial x} f\left(x_n, y(x_n)\right) + a_{21} h f_n \frac{\partial}{\partial y} f\left(x_n, y(x_n)\right) \\
&+ c_2^2 h^2 \frac{\partial^2}{\partial x^2} f\left(x_n, y(x_n)\right) + a_{21}^2 h^2 f_n^2 \frac{\partial^2}{\partial y^2} f\left(x_n, y(x_n)\right) \\
&+ 2 c_2 a_{21} h^2 f_n \frac{\partial^2}{\partial x \partial y} f\left(x_n, y_n\right) + \mathcal{O}\left(h^3\right)
\end{aligned}$$

§6.3 显式 Runge-Kutta 方法: 二级方法

$$
\begin{aligned}
y(x_{n+1}) &= y(x_n + h) \\
&= y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{2!}y'''(x_n) + \mathcal{O}(h^4)
\end{aligned}
$$

$$
\begin{aligned}
y_n' =& f\left(x_n, y_n\right) = f_n \\
y_n'' =& \frac{d}{dx} f\left(x_n, y\left(x_n\right)\right) = f_x'\left(x_n, y_n\right) + f_y'\left(x_n, y_n\right) f_n \\
y_n''' =& f_{xx}''\left(x_n, y_n\right) + 2 f_n f_{xy}''\left(x_n, y_n\right) + f_n^2 f_{yy}''\left(x_n, y_n\right) \\
& + f_y'\left(x_n, y_n\right)\left[f_x'\left(x_n, y_n\right) + f_n f_y'\left(x_n, y_n\right)\right]
\end{aligned}
$$

§6.3 显式 Runge-Kutta 方法: 二级方法

$$
\begin{aligned}
T_{n+1} =& y(x_{n+1}) - y(x_n) - h[b_1 f_n + b_2 f(x_n + c_2 h, y_n + a_{21} h f_n)] \\
=& h f_n + \frac{h^2}{2}[f'_x(x_n, y_n) + f'_y(x_n, y_n) f_n] \\
& - h\big[b_1 f_n + b_2(f_n + c_2 f'_x(x_n, y_n) h + a_{21} f'_y(x_n, y_n) f_n h)\big] \\
& + g\, h^3 + \mathcal{O}(h^4) \\
=& (1 - b_1 - b_2) f_n h + (\frac{1}{2} - b_2 c_2) f'_x(x_n, y_n) h^2 \\
& + (\frac{1}{2} - b_2 a_{21}) f'_y(x_n, y_n) f_n h^2 + g\, h^3 + \mathcal{O}(h^4)
\end{aligned}
$$

其中

$$
\begin{aligned}
g =& f''_{xx}(x_n, y_n) + 2f_n f''_{xy}(x_n, y_n) + f_n^2 f''_{yy}(x_n, y_n) \\
&+ f'_y(x_n, y_n)\left[f'_x(x_n, y_n) + f_n f'_y(x_n, y_n)\right] \\
&- c_2^2 b_2 f''_{xx}(x_n, y_n) - a_{21}^2 b_2 f_n^2 f''_{yy}(x_n, y_n) \\
&- 2c_2 a_{21} b_2 f_n f''_{xy}(x_n, y_n) \\
=& (1 - c_2^2 b_2) f''_{xx}(x_n, y_n) + 2(1 - c_2 a_{21} b_2) f_n f''_{xy}(x_n, y_n) \\
&+ (1 - a_{21}^2 b_2) f_n^2 f''_{yy}(x_n, y_n) \\
&+ f'_y(x_n, y_n)\left[f'_x(x_n, y_n) + f_n f'_y(x_n, y_n)\right].
\end{aligned}
$$

§6.3 显式 Runge-Kutta 方法: 二级方法

一阶条件

$$b_1 + b_2 = 1$$

二阶条件

$$\frac{1}{2} - b_2 c_2 = 0, \quad \frac{1}{2} - b_2 a_{21} = 0$$

等价为

$$b_2 = 1 - b_1, \quad c_2 = a_{21} = \frac{1}{2b_2}$$

若取 $b_1 = 1/2$ 则 $b_2 = 1/2, c_2 = a_{21} = 1$，得计算公式

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2} h K_1 + \frac{1}{2} h K_2 \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + h, y_n + h K_1\right) \end{cases},$$

为改进 Euler 法

§6.3 显式 Runge-Kutta 方法: 二级方法

若取 $b_1 = 0$ 则 $b_2 = 1, c_2 = a_{21} = 1/2$, 得计算公式

$$\begin{cases} y_{n+1} = y_n + hK_2 \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1\right) \end{cases}$$

称其为中点公式
故 $s = 2$ 的显式 R-K 方法的阶只能是 $p = 2$, 而不能得到 3 阶方法。

§6.3 显式 Runge-Kutta 方法: 三级方法

一般形式

$$\begin{cases} y_{n+1} = y_n + b_1 h K_1 + b_2 h K_2 + b_3 h K_3 \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + c_2 h, y_n + a_{21} h K_1\right) \\ K_3 = f\left(x_n + c_3 h, y_n + a_{31} h K1 + a_{32} h K_2\right) \end{cases}$$

常见得三级显式 Runge-Kutta 方法

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}\left(K_1 + 4K_2 + K_3\right) \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_1\right) \\ K_3 = f\left(x_n + h, y_n - hK_1 + 2hK_2\right) \end{cases}$$

它是 3 阶精度方法

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{4}{3} & \frac{1}{6} \end{array}$$

§6.3 显式 Runge-Kutta 方法: 四级方法

常用的四级显式 Runge-Kutta 方法

$$\begin{cases} y_{n+1} = y_n + \frac{1}{6}h\left(K_1 + 2K_2 + 2K_3 + K_4\right) \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hK_1\right) \\ K_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hK_2\right) \\ K_4 = f\left(x_n + h, y_n + hK_3\right) \end{cases}$$

局部截断误差为 $\mathcal{O}\left(h^5\right)$，是四阶方法.

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

s 级显式 Runge-Kutta 法的精度不超过 s

§6.4 单步法的收敛性与稳定性: 收敛性与相容性

显式单步法的一般形式为

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h) \tag{3}$$

其中多元函数 φ 依赖于 f.
$e_n = y(x_n) - y_n$ 称为整体截断误差.
收敛性就是讨论当 $x = x_n$ 固定且 $h = (x_n - a)/n \to 0$ 时 $e_n \to 0$ 的问题.
定义 若一种数值方法 (如单步法(3)) 对于固定的 $x = x_0 + nh$，当 $h \to 0$ 时有 $y_n \to y(x)$，其中 $y(x)$ 是初值问题(1)的准确解，则称该方法是收敛的.

§6.4 单步法的收敛性与稳定性: 收敛性与相容性

定理 假设单步法(3)具有 p 阶精度，且增量函数 $\varphi(x,y,h)$ 关于 y 满足利普希茨条件

$$|\varphi(x,y_1,h)-\varphi(x,y_2,h)|\le L_\varphi|y_1-y_2|$$

又设初值 y_0 是准确的，即 $y_0=y(x_0)$，则其整体截断误差

$$y(x_n)-y_n=\mathcal{O}(h^p)$$

定义 若单步法(3)的增量函数 $\varphi(x,y,h)$ 满足

$$\varphi(x,y,0)=f(x,y)$$

则称单步法(3)与(1) 相容.
定理 p 阶方法(3)与初值问题(1)相容的充分必要条件是 $p\ge 1$

当步长取定后，计算中的误差随着步数的增加会不会积累到超出我们许可的范围，这就是稳定性问题. 单步法(3)应用于模型方程

$$\begin{cases} y'(x) = \lambda y \quad (\Re\lambda < 0) \\ y(a) = y_0 \end{cases}$$

设得到的解为 $y_{n+1} = R(\lambda h) y_n$
当 $\Re\lambda < 0$ 时, 实验方程的精确解 $y(x) = y_0 \mathrm{e}^{\lambda(x-a)}$ 按模递减的, 这要求 (3)的解 y_n 是按模递减的，误差也是递减的，即要求满足 $|R(\lambda h)| < 1$.

§6.4 单步法的收敛性与稳定性: 稳定性

$$\lambda = \Re\lambda + \mathrm{i}\Im\lambda.$$

$$y(x) = y_0 \mathrm{e}^{\lambda(x-a)} = y_0 \mathrm{e}^{\Re\lambda(x-a)} \mathrm{e}^{\mathrm{i}\Im\lambda(x-a)}$$

$$|y(x)| = |y_0 \mathrm{e}^{\Re\lambda(x-a)} \mathrm{e}^{\mathrm{i}\Im\lambda(x-a)}| = |y_0| \mathrm{e}^{\Re\lambda(x-a)}.$$

由于 $\Re\lambda < 0$ 精确解 $y(x) = y_0\mathrm{e}^{\lambda(x-a)}$ 按模递减。让 y_n 也按模递减需要

$$\frac{|y_{n+1}|}{|y_n|} = |R(\lambda h)| < 1$$

其它解释

$$y_n = R^n(\lambda h) y_0$$

y_n 有界需要 $|R(\lambda h)| < 1$

§6.4 单步法的收敛性与稳定性: 稳定性

定义 单步法(3)用于解模型方程 $y' = \lambda y$，若得到的解 $y_{n+1} = R(\lambda h) y_n$ 满足 $|R(\lambda h)| < 1$, 则称单步法是绝对稳定的. 在 $z = h\lambda$ 的平面上，使 $|R(\lambda h)| < 1$ 的变量围成的区域，称为绝对稳定域，它与实轴的交称为绝对稳定区间.

定义 如果数值方法的绝对稳定域包含了 $\{h\lambda | \Re(h\lambda) < 0\}$, 那么称此方法是 A-稳定的.

由定义知 A-稳定方法对步长没有限制.

例 Euler 方法

$$R(h\lambda) = 1 + h\lambda$$

由 $|R(\lambda h)| < 1$ 得绝对稳定区间 $\lambda h \in (-2, 0)$

例 二阶 Runge-Kutta 公式

$$R(h\lambda) = 1 + h\lambda + \frac{(h\lambda)^2}{2!}$$

由 $|R(\lambda h)| < 1$ 得绝对稳定区间 $\lambda h \in (-2, 0)$.

例 三阶 Runge-Kutta 公式

$$R(h\lambda) = 1 + h\lambda + \frac{(h\lambda)^2}{2!} + \frac{(h\lambda)^3}{3!}$$

由 $|R(\lambda h)| < 1$ 得绝对稳定区间 $\lambda h \in (-2.51, 0)$.

例 四阶 Runge-Kutta 公式

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

由 $|R(\lambda h)| < 1$ 得绝对稳定区间 $\lambda h \in (-2.78, 0)$

例 利用 4 阶经典 Runge-Kutta 方法在步长 $h=0.1$ 与 $h=0.2$ 中选取适当的步长求解初值问题

$$\begin{cases} \frac{dy}{dx} = -20y \quad 0 \le x \le 1 \\ y(0) = 1 \end{cases}$$

(要求计算 2 步，即 y_2)

解 标准 4 阶经典 Runge-Kutta 方法为

$$\begin{cases} y_{n+1} = y_n + \frac{1}{6}h\left(K_1 + 2K_2 + 2K_3 + K_4\right) \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hK_1\right) \\ K_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hK_2\right) \\ K_4 = f\left(x_n + h, y_n + hK_3\right) \end{cases}$$

可得

$$y_{n+1} = \left[1 - 20h + \frac{(-20h)^2}{2!} + \frac{(-20h)^3}{3!} + \frac{(-20h)^4}{4!}\right] y_n$$

由 $|R(\lambda h)| < 1$ 得绝对稳定区间 $\lambda h \in (-2.78, 0)$, 即

$$-20h \in (-2.78, 0)$$

所以取步长 $h = 0.1$
计算结果： $y_1 = 1/3, \quad y_2 = 1/9$

例

$$\begin{cases} y' + 10y + 2 = 0 \\ y(0) = 1 \end{cases}$$

应用改进 Euler 方法求解初值问题问步长如何选取方能保证绝对稳定?
并在 $h = 0.1$ 与 $h = 0.2$ 中选取数值稳定的步长计算 $y(0.2)$ 的近似值
解 令 $z = y + \frac{1}{5}$ 原初值问题等价于

$$\begin{cases} z' = -10z \\ z(0) = 1.2 \end{cases}$$

将改进 Euler 方法应用到 $z' = -10z$, 有

$$z_{n+1} = \left[1 - 10h + \frac{(-10h)^2}{2!}\right] z_n$$

由 $|R(-10h)| < 1$ 得绝对稳定区间 $-10h \in (-2, 0)$.

所以取步长 $h = 0.1$. $\quad z_{n+1} = 0.5 z_n$
计算结果: $z_1 = 0.6, \quad z_2 = 0.3$

$$y(0.2) = z(0.2) - 0.2 \approx z_2 - 0.2 = 0.1$$

§6.5 线性多步法

计算 y_{n+1} 时，除用 y_n 的值，还用到 $y_{n-i}(i=1,\ldots,p)$ 的值, 则称此方法为线性多步法. 一般的线性多步法公式可表示为

$$y_{n+1}=\sum_{i=0}^{p}a_iy_{n-i}+h\sum_{i=-1}^{p}b_iy'_{n-i},\quad n=p,p+1,\ldots \tag{4}$$

其中 y_{n-i} 为 $y(x_{n-i})$ 的近似,$y'_{n-i}=f\left(x_{n-i},y_{n-i}\right)$, $h=x_{i+1}-x_i$, $x_{n-i}=x_n-ih$, a_i,b_i 为常数,a_p 及 b_p 不全为零，则称(4)式为线性 $p+1$ 步法.

注

1. 计算时需先给出前面 $p+1$ 个近似值 $y_0,y_1,\cdots,y_p$, 再由(4)式逐次求出 $y_{p+1},y_{p+2},\cdots$.
2. 如果 $b_{-1}=0$, 则称 (4)式为显式 $p+1$ 步法
3. 如果 $b_{-1}\neq 0$, 则称 (4)式为隐式 $p+1$ 步法

§6.5 线性多步法

定义 设 $y(x)$ 是初值问题 (1)的准确解，线性多步法(4)在 x_{n+1} $(n=p,p+1,\ldots)$ 上的局部截断误差为

$$\begin{aligned} T_{n+1} &= y\left(x_{n+1}\right)-y_{n+1} \\ &= y\left(x_{n+1}\right)-\sum_{i=0}^{p} a_i y(x_{n-i})-h\sum_{i=-1}^{p} b_i y'(x_{n-i}), \end{aligned} \tag{5}$$

若对 $y\in\mathbb{P}_r$,$T_{n+1}=0$, 对 $y=x^{r+1}$,$T_{n+1}\neq 0$, 则称方法(4)是 r 阶的。

§6.5 线性多步法

对 T_{n+1} 在 x_n 处做泰勒展开. 对于 $i=-1,0,\ldots,p$

$$\begin{aligned} y\left(x_n-ih\right)= & y\left(x_n\right)-ihy'\left(x_n\right)+\frac{(ih)^2}{2!}y''\left(x_n\right)-\cdots \\ = & \sum_{j=0}^{q}\frac{(-i)^j}{j!}h^j y^{(j)}\left(x_n\right)+\cdots+ \end{aligned} \tag{6}$$

$$\begin{aligned} y'\left(x_n-ih\right)= & y'\left(x_n\right)-ihy''\left(x_n\right)+\frac{(ih)^2}{2!}y'''\left(x_n\right)-\cdots \\ = & \sum_{j=1}^{q}\frac{(-i)^{j-1}}{(j-1)!}h^{j-1}y^{(j)}\left(x_n\right)+\cdots+ \end{aligned} \tag{7}$$

代入(5)式得

$$T_{n+1}=c_0 y\left(x_n\right)+c_1 hy'\left(x_n\right)+\cdots+c_q h^q y^{(q)}\left(x_n\right)+\cdots$$

其中

$$
\begin{aligned}
c_0 &= 1 - \sum_{i=0}^{p} a_i \\
c_1 &= 1 - \left[\sum_{i=0}^{p} (-i) a_i + \sum_{i=-1}^{p} b_i \right] \\
&\vdots \\
c_q &= \frac{1}{q!} \left\{ 1 - \left[\sum_{i=0}^{p} (-i)^q a_i + q \sum_{i=-1}^{p} (-i)^{q-1} b_i \right] \right\} \\
& \qquad q = 2, 3, \ldots
\end{aligned}
\tag{8}
$$

若在公式(4)中选择系数 a_i 及 b_i，使它满足

$$c_0 = c_1 = \cdots = c_r = 0, \quad c_{r+1} \neq 0$$

由定义可知此时所构造的多步法是 r 阶的，且

$$T_{n+1} = c_{r+1} h^{r+1} y^{(r+1)}(x_n) + \mathcal{O}\left(h^{r+2}\right)$$

称右端第一项为**局部截断误差主项**, c_{r+1} 称为误差常数.

定理 线性多步法(4)式是 r 阶的充分必要条件是由(8)式定义的 $c_i(i=0,1,2,\cdots)$ 满足关系式 $c_0=c_1=\cdots=c_r=0, c_{r+1}\neq 0$

定义：相容性 如果 $c_0=c_1=0$, 即

$$\begin{cases} \sum\limits_{i=0}^{p} a_i = 1 \\ \sum\limits_{i=0}^{p}(-i)a_i + \sum\limits_{i=-1}^{p} b_i = 1 \end{cases} \tag{9}$$

称方法(4)式与微分方程(1)相容。

满足相容性，方法至少是 1 阶方法

§6.5 线性多步法: 构造方法

基于 Taylor 展开的构造方法令 $c_0 = c_1 = \cdots = c_r = 0$, 由(8)式得到

$$\begin{cases} \sum\limits_{i=0}^{p} a_i = 1, \\ \sum\limits_{i=0}^{p} (-i) a_i + \sum\limits_{i=-1}^{p} b_i = 1 \\ \vdots \\ \sum\limits_{i=0}^{p} (-i)^q a_i + q \sum\limits_{i=-1}^{p} (-i)^{q-1} b_i, q = 2, 3, \cdots, r \end{cases} \tag{10}$$

式(10)是关于 $2p+3$ 个未知数 $a_i(i = 0, 1, \cdots, p), b_i(i = -1, 0, 1, \cdots, p)$ 的 $r+1$ 个方程的线性方程组. 可以证明：当$r = 2p + 2$时，(10)解存在唯一.

§6.5 线性多步法

4 阶 Simpson 方法

$$y_{n+1}=y_{n-1}+\frac{h}{3}\left[y_{n+1}^{\prime}+4y_{n}^{\prime}+y_{n-1}^{\prime}\right]$$

$$T_{n+1}=-\frac{1}{90}h^{5}y^{5}\left(x_{n}\right)+\cdots, \quad C_{5}=-\frac{1}{90}$$

4 阶 Milne 方法

$$y_{n+1}=y_{n-3}+\frac{4}{3}h\left(2f_{n}-f_{n-1}+2f_{n-2}\right)$$

$$T_{n+1}=\frac{14}{45}h^{5}y^{5}\left(x_{n}\right)+\cdots, \quad C_{5}=\frac{14}{45}$$

4 阶 Hamming 方法

$$y_{n+1}=\frac{1}{8}\left(9y_{n}-y_{n-2}\right)+\frac{3}{8}h\left(f_{n+1}+2f_{n}-f_{n-1}\right)$$

$$T_{n+1}=-\frac{1}{40}h^{5}y^{5}\left(x_{n}\right)+\cdots, \quad C_{5}=-\frac{1}{80}$$

§6.5 线性多步法

基于数值积分的构造方法在 $[x_n, x_{n+1}]$ 上对 $y'(x) = f(x, y(x))$ 积分得

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

用插值多项式逼近 $f(x, y(x))$ 进行数值积分可建立一类多步法-Adams 方法.

如显式 4 步 4 阶 Adams 方法

$$y_{n+1} = y_n + \frac{h}{24}\left[55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}\right]$$

局部截断误差为

$$T_{n+1} = \frac{251}{720} h^5 y^{(5)}\left(x_n\right) + \cdots$$

梯形法

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

隐式 3 步 4 阶 Adams 方法

$$y_{n+1} = y_n + \frac{h}{24}\left[9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}\right]$$

局部截断误差为

$$T_{n+1} = -\frac{19}{720}h^5 y^{(5)}\left(x_n\right) + \cdots$$

§6.6 线性多步法的收敛性与稳定性

定义 设方程(4)的初始条件 $y_k = y_k(h), k = 0, 1, 2, \cdots, p,$ 满足

$$\lim_{h\to 0} y_k(h) = y(x_0), \quad k = 0, 1, 2, \cdots, p.$$

若 $f(x, y)$ 满足连续性，关于第二个变量满足 Lipschitz 条件。对任意固定的 $x \in [a, b]$ 满足

$$\lim_{\substack{h\to 0\\ nh=x-a}} y_n = y(x).$$

则称线性多步法(4)是收敛的.

§6.6 线性多步法的收敛性与稳定性

考察方程

$$\begin{cases} y' = \lambda y, \\ y(a) = \eta. \end{cases} \tag{11}$$

称此方程为试验方程. 它的解为

$$y(x) = \eta e^{\lambda(x-a)}.$$

使用线性多步法(4)求解(11)时，有

$$y_{n+1} = \sum_{i=0}^{p} a_i y_{n-i} + h\lambda \sum_{i=-1}^{p} b_i y_{n-i} \tag{12}$$

等价于

$$(1 - h\lambda b_{-1}) y_{n+1} = \sum_{i=0}^{p} (a_i + h\lambda b_i) y_{n-i} \tag{13}$$

记

$$\boldsymbol{Y}^{(n+1,p)} = (y_{n+1}, y_n, \ldots, y_{n-p+1})^T$$

有

$$\boldsymbol{Y}^{(n+1,p)} = \boldsymbol{A}\boldsymbol{Y}^{(n,p)} = \boldsymbol{A}^2\boldsymbol{Y}^{(n-1,p)} = \cdots = (\boldsymbol{A})^{n+1-p}\boldsymbol{Y}^{(p,p)},$$

其中

$$\boldsymbol{Y}^{(n,p)} = (y_n, y_{n-1}, \ldots, y_{n-p})^T$$

$$\boldsymbol{A} = \begin{pmatrix} d_0 & d_1 & \cdots & d_{p-1} & d_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

$d_i = (a_i + h\lambda b_i)/(1 - h\lambda b_{-1})$.

$$
\begin{aligned}
|r\boldsymbol{I}-\boldsymbol{A}| &= r^{p+1}-\sum_{i=0}^{p} d_i r^{p-i} \\
&= (1-h\lambda b_{-1})^{-1}\left((1-h\lambda b_{-1})r^{p+1}-\sum_{i=0}^{p}(a_i+h\lambda b_i)r^{p-i}\right) \\
&= (1-h\lambda b_{-1})^{-1}(\rho(r)-h\lambda\sigma(r)),
\end{aligned} \tag{14}
$$

其中

$$\rho(r) = r^{p+1} - \sum_{i=0}^{p} a_i r^{p-i}, \quad \sigma(r) = \sum_{i=-1}^{p} b_i r^{p-i}.$$

分别称其为线性多步法(4)的第一和第二特征多项式. 记

$$\pi(r;\lambda h) = \rho(r) - h\lambda\sigma(r), \tag{15}$$

称其为特征多项式. 记 $\pi(r;\lambda h) = 0$ 的根为

$$r_0(h\lambda), r_1(h\lambda), \cdots, r_p(h\lambda).$$

如果根都不相同，方程(12)的通解为

$$y_n = \sum_{i=0}^{p} g_i [r_i(h\lambda)]^n, \tag{16}$$

其中，g_i 是任意常数.

§6.6 线性多步法的收敛性与稳定性

定义 若 $\rho(r)$ 的所有根的模均不大于 1，且模为 1 的根是单根，则称 $\rho(r)$ 以及相应的线性多步法(4)满足根条件.

定理 若线性多步法(4)收敛，则其满足根条件.

定理 线性多步法(4)相容的充分必要条件是

$$\rho(1)=0, \quad \rho'(1)=\sigma(1)$$

定理 线性多步法(4)是相容的，则线性多步法(4)收敛的充分必要条件是线性多步法(4)满足根条件.

例 显式 4 阶 Adams 方法,

$$y_{n+1} = y_n + \frac{h}{24}\left[55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}\right]$$

$$\rho(r) = r^4 - r^3, \sigma(r) = \left(55r^3 - 59r^2 + 37r - 9\right)/24$$

$$\rho(1) = 0, \quad \rho'(1) = \sigma(1) \quad (C_0 = 0, \quad C_1 = 0)$$

从而方法是相容的.
$\rho(r) = 0$ 的 4 个根 0,0,0,1，方法满足根条件。
故显式 4 阶 Adams 方法是收敛的.

例 4 阶 Hamming 方法

$$y_{n+1}=\frac{1}{8}\left(9y_n-y_{n-2}\right)+\frac{3}{8}h\left(y'_{n+1}+2y'_n-y'_{n-1}\right)$$

$$\rho(r)=r^3-\frac{9}{8}r^2+\frac{1}{8},\sigma(r)=\frac{3}{8}r^3+\frac{3}{4}r^2-\frac{3}{8}r$$

$$\rho(1)=0,\quad \rho'(1)=\sigma(1)\quad (C_0=0,\quad C_1=0)$$

从而方法是相容的. $\rho(r)=0$ 的 3 个根 $r_0=1,r_{2,3}=\frac{1\pm\sqrt{33}}{16}$ 方法满足根条件. 故显式 4 阶 Hamming 方法是收敛的.

§6.6.2 线性多步法的稳定性

定义 差分方程 (12)得到解有界，称线性多步法(4)是稳定的

定义 差分方程 (12)得到解趋于 0($|y_n| \to 0, n \to \infty$)，称线性多步法(4)是绝对稳定的

定理 线性多步法(4)是稳定的充分必要条件是它满足根条件.

定理 线性多步法(4)是绝对稳定的充分必要条件是(15)的根 $r_i(h\lambda), i = 0, \ldots, p$ 的模都小于 1 （$|r_i(h\lambda)| < 1, i = 0, \ldots, p$） .

定义 实轴上所有使方法是相对或绝对稳定的 $\bar{h} = h\lambda$ 的集合，称为方法的相对或绝对稳定集.

定义 绝对稳定区间为稳定域（包含复数）与实轴的交集。

定义 若一个方法的绝对稳定区间是 $(-\infty, 0)$，则称此方法是 A–稳定的.

定理 二次多项式 $q(r) = r^2 + ar + b$, 其中 a 和 b 都是实数, r_1 和 r_2 为二次多项式的两个根（可能是共轭复根）, $|r_1| < 1$ 和 $|r_2| < 1$ 充分必要条件,

$$\text{(i) } b < 1, \quad \text{(ii) } 1 + a + b > 0, \quad \text{(iii) } 1 - a + b > 0$$

证明 二次多项式 $q(r) = r^2 + ar + b$ 根为

$$r_1, r_2 = \frac{1}{2}\left(-a \pm \sqrt{a^2 - 4b}\right)$$

当 $a^2 < 4b$ 根为共轭复根, 有 $b = r_1 r_2 = |r_1|^2 = |r_2|^2$. 条件 $|r_1| < 1$,$|r_2| < 1$ 和 $b < 1$ 等价。
由于 $a^2 < 4b < (1+b)^2$ 蕴含 (ii) 和 (iii) 是满足的.

§6.6.2 线性多步法的稳定性

考虑实根情况 $(a^2 \geq 4b)$, 记

$$R = \max\{|r_1|, |r_2|\} = \frac{1}{2}\left(|a| + \sqrt{a^2 - 4b}\right)$$

R 关于 $|a|$ 是一个增函数，当 $|a| = 1 + b$ 时 $R = 1$。因此 $0 \leq R < 1$ 和 $0 \leq |a| < 1 + b$ 等价. $|r_1 r_2| < 1$, 蕴含 $|b| < 1$，条件 (i) 一定成立.
组合实根情况和复根情况, 得到 $|r_1| < 1$ 和 $|r_2| < 1$ 与 $|a| - 1 < b < 1$ 等价。

§6.7 预测–校正方法

当 $b_{-1}\neq 0$ 时，线性多步法 (6.2.1) 式是隐式的，可写成

$$y_{n+1}=b_{-1}hf(x_{n+1},y_{n+1})+\sum_{i=0}^{p}(a_iy_{n-i}+b_ihy'_{n-i}). \tag{17}$$

关于 y_{n+1} 的非线性方程. 给出 y_{n+1}^{0}，使用迭代格式

$$y_{n+1}^{(j+1)}=b_{-1}hf(x_{n+1},y_{n+1}^{(j)})+\sum_{i=0}^{p}(a_iy_{n-i}+b_ihy'_{n-i}), j=0,1,2,\cdots, \tag{18}$$

其中 ${y_{n+1}}^{(j+1)}$ 为 y_{n+1} 的 $j+1$ 次近似值.
当 $h<1/(L|b_{-1}|)$ 时，迭代公式(18)是收敛的，其中 L 为 f 关于 y 的 Lipschitz 条件。
常用显示多步法给出 y_{n+1}^{0}，称为预估步，使用迭代过程(18), 称校正步。

§6.7 预测–校正方法

$y_{n+1}^{(0)}$ 使用显示 $\tilde{p}+1$ 步法计算 $y_{n+1}^{(0)}$，称为预估步

$$[P] \quad y_{n+1}^{(0)} = \sum_{j=0}^{\tilde{p}} \tilde{a}_j y_{n-j}^{(1)} + h \sum_{j=0}^{\tilde{p}} \tilde{b}_j f_{n-j}^{(0)}$$

其中 $f_k^{(0)} = f(x_k, y_k^{(0)})$; $y_k^{(0)}, k = n - \tilde{p}, \ldots, n$ 是预测–校正方法得到解。
计算 f 在点 $(x_{n+1}, y_{n+1}^{(0)})$ 的值（评估步）

$$[E] \quad f_{n+1}^{(0)} = f\left(x_{n+1}, y_{n+1}^{(0)}\right)$$

带入隐式线性多步法（校正步）

$$[C] \quad y_{n+1}^{(1)} = \sum_{j=0}^{p} a_j y_{n-j}^{(1)} + h \sum_{j=0}^{p} b_j f_{n-j}^{(0)} + h b_{-1} f_{n+1}^{(0)}$$

记为 PEC 或 $P(EC)^1$ 方法

§6.7 预测–校正方法

一般的 $P(EC)^{\mu}E^{1-t}$ 方法 (μ 为正整数，$t=0$ 或 1)

使用显示 $\tilde{p}+1$ 步法计算 $y_{n+1}^{(0)}$(预估步)

$$[P] \quad y_{n+1}^{(0)} = \sum_{j=0}^{\tilde{p}} \tilde{a}_j y_{n-j}^{(1)} + h \sum_{j=0}^{\tilde{p}} \tilde{b}_j f_{n-j}^{(0)}$$

$(EC)^{\mu}: k = 0, 1, \ldots, \mu - 1$

$$[E] \quad f_{n+1}^{(k)} = f\left(x_{n+1}, y_{n+1}^{(k)}\right)$$

$$[C] \quad y_{n+1}^{(k+1)} = \sum_{j=0}^{p} a_j y_{n-j}^{(k)} + h \sum_{j=0}^{p} b_j f_{n-j}^{(\mu-t)} + h b_{-1} f_{n+1}^{(k)}$$

E^{1-t} :

$$f_{n+1}^{(\mu-t)} = f\left(x_{n+1}, y_{n+1}^{(\mu-t)}\right)$$

§6.8 一阶方程组与高阶方程

考察一阶方程组

$$y_i' = f_i\left(x, y_1, y_2, \ldots, y_N\right), \quad i = 1, 2, \ldots, N$$

的初值问题，初始条件为

$$y_i\left(x_0\right) = y_i^0, \quad i = 1, 2, \ldots, N$$

若采用向量的记号，

$$\boldsymbol{y} = \left(y_1, y_2, \ldots, y_N\right)^T, \quad \boldsymbol{y}_0 = \left(y_1^0, y_2^0, \ldots, y_N^0\right)^T,$$
$$\boldsymbol{f} = \left(f_1, f_2, \ldots, f_N\right)^T$$

则上述方程组的初值问题可表示为

$$\begin{cases} \boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}) \\ \boldsymbol{y}\left(x_0\right) = \boldsymbol{y}_0 \end{cases}$$

求解这一初值问题的四阶龙格-库塔公式为

$$\boldsymbol{y}_{n+1}=\boldsymbol{y}_n+\frac{h}{6}\left(\boldsymbol{K}_1+2\boldsymbol{K}_2+2\boldsymbol{K}_3+\boldsymbol{K}_4\right)$$

$$\begin{aligned}
\boldsymbol{K}_1&=\boldsymbol{f}\left(x_n,\boldsymbol{y}_n\right)\\
\boldsymbol{K}_2&=\boldsymbol{f}\left(x_n+\frac{h}{2},\boldsymbol{y}_n+\frac{h}{2}\boldsymbol{K}_1\right)\\
\boldsymbol{K}_3&=\boldsymbol{f}\left(x_n+\frac{h}{2},\boldsymbol{y}_n+\frac{h}{2}\boldsymbol{K}_2\right)\\
\boldsymbol{K}_4&=\boldsymbol{f}\left(x_n+h,\boldsymbol{y}_n+h\boldsymbol{K}_3\right)
\end{aligned}$$

§6.8 一阶方程组与高阶方程

化高阶方程为一阶方程组

$$y^{(m)} = f\left(x, y, y', \cdots, y^{(m-1)}\right)$$

初始条件为

$$y(x_0) = y_0, y'(x_0) = y_0', \cdots, y^{(m-1)}(x_0) = y_0^{(m-1)}$$

只要引进新的变量

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

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \qquad \vdots \\ y_{m-1}' = y_m \\ y_m' = f(x, y_1, y_2, \cdots, y_m) \end{cases}$$

初始条件则相应地化为

$$y_1(x_0) = y_0, y_2(x_0) = y_0', \cdots, y_m(x_0) = y_0^{(m-1)}$$