第六章 常微分方程数值解法

6.1 数值方法与基本概念

在科学技术领域中有许多问题可以归结为一类常微分方程的定解问题．然而，在生产实际和科学研究中所遇到的微分方程往往很复杂，至今有许多类型的微分方程尚不能给出解的解析表达式．有时候对有些微分方程即使能求出解析解，也往往因计算量大而不实用．因此，需要研究求解微分方程的数值方法．

本章主要讨论常微分方程初值问题的数值方法，并构造高精度的单步法及多步法．

我们将着重考察如下形式的一阶常微分方程

研究对象

$$\begin{cases} y' = f(x,y), & x \in [a,b], |y| < \infty, \\ y(x_0) = y_0, \end{cases} \tag{1.1}$$

如果存在可微函数 $y(x)$ 满足（1.1）式，则称 $y(x)$ 为问题（1.1）的解．确定函数 $y(x)$ 使它满足(1.1)式的问题称为初值问题．满足问题(1.1)式的一阶可微函数 $y(x)$ 的存在性和唯一性问题自然是我们关心的，对于这个问题，有下面的定理．

定理 1 如果函数 $f(x,y)$ 在区域 $G: a \le x \le b,\ |y| < \infty$ 上连续，且关于 y 满足 Lipschitz（利普希茨）条件，即存在常数 L（称为 Lipschitz 常数）使得对所有 $x \in [a,b]$ 及任意的 y_1, y_2 不等式

$$|f(x,y_1) - f(x,y_2)| \le L|y_1 - y_2| \tag{1.2}$$

均成立，则初值问题(1.1)在 $[a,b]$ 上有唯一解 $y(x)$．

求解常微分方程虽然有各种各样的解析方法，但解析方法只能用来求解一些特殊类型的方程，求解实际问题中归结出来的微分方程主要靠数值方法．

数值方法的实质是微分方程的解 $y(x)$ 被它在一系列离散节点

$$x_0 < x_1 < \cdots < x_n < x_{n+1} < \cdots$$

上的值 $y_0, y_1, \cdots y_n, y_{n+1}, \cdots$ 近似代替．相邻两个节点的间距 $h_n = x_{n+1} - x_n$ 称为步长．今后如不特别说明，只是假定 $h_i = h(i = 1,2,\cdots)$ 为定数，这时节点为 $x_n = x_0 + nh$，$n = 0,1,2\cdots$．

建立一个数值方法时，需要把连续问题（1.1）通过一定的方法建立一个相应的离散化问题，使得在 $N+1$ 个给定点 x_n $(n = 0,1,2,\cdots,N)$ 上，这个离散问题的解 $y_n (n = 0,1,2,\cdots N)$

能够近似连续性问题的解 $y(x_n)$．我们称这个过程为“离散化”.

初值问题（1.1）的数值解法有个基本特点，它们都采取“步进式”，即求解过程顺着节点排列的次序一步一步地向前推进，由给出已知的信息 $y_n, y_{n-1}, y_{n-2}, \cdots$，计算 y_{n+1} 的递推公式.

根据求数值解的递推公式通常将数值方法分为两类：一类是计算 y_{n+1} 时只用到前一点的值 y_n，称为**单步法**；另一类是用到 y_{n+1} 前面 k 点值 $y_n, y_{n-1}, \cdots y_{n-k+1}$，称为 **k 步法**.

数值方法主要研究的内容有，首先是对方程（1.1）离散化，建立求数值解的递推公式；其次，要研究公式的局部截断误差和误差阶，数值解 y_n 与精确解 $y(x_n)$ 的误差估计及收敛性，还有递推公式的计算稳定性等问题.

6.1.1 欧拉法与后退欧拉法

在点 x_n 处的导数 $y'(x_n)$ 可以近似地表示成差商

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

即

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

这就是著名的欧拉公式

$$y_{n+1} = y_n + hf(x_n, y_n). \tag{1.3}$$

若初值 y_0 已知，则依公式（1.3）可逐步算出

$$y_1 = y_0 + hf(x_0, y_0),$$

$$y_2 = y_1 + hf(x_1, y_1),$$

$$\cdots$$

例 1　求解初值问题 $$\begin{cases} y' = y - \dfrac{2x}{y}, & (0 < x < 1) \\ y(0) = 1, \end{cases} \tag{1.4}$$

解　为便于进行比较，本章用多种数值方法求解上述初值问题. 这里先用欧拉方法，欧拉公式的具体形式为

$$y_{n+1} = y_n + h\left(y_n - \frac{2x_n}{y_n}\right).$$

取步长 $h=0.1$， 计算结果见表 7-1.

初值问题（1.4）有解 $y=\sqrt{1+2x}$， 按这个解析式子算出的准确值 $y(x_n)$ 同近似值 y_n 一起列在表 6-1 中， 两者比较可以看出欧拉方法的精度很差.

欧拉法具有一般数值方法的特征，对欧拉法的收敛性和误差分析简单易行，因此可通过对欧拉法的收敛性和误差分析的讨论，掌握对一般数值方法的收敛性、误差分析问题的研究方法.

(近似 精确

表 6-1 计算结果对比

x_n	y_n	$y(x_n)$	x_n	y_n	$y(x_n)$
0.1	1.1000	1.0954	0.6	1.5090	1.4832
0.2	1.1918	1.1832	0.7	1.5803	1.5492
0.3	1.2774	1.2649	0.8	1.6498	1.6125
0.4	1.3582	1.3416	0.9	1.7178	1.6733
0.5	1.4351	1.4142	1.0	1.7848	1.7321

由方程（1.1）有

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

当 h 充分小时有

$$y(x_{n+1})=y(x_n)+hf(x_n,y(x_n))+R_n. \qquad (1.5)$$

为了简化分析我们假设 $y(x_n)=y_n$，并将（1.5）式减（1.3）式得

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

称 $R_n=y(x_{n+1})-y_{n+1}$ 为欧拉法的**局部截断误差**.

如果对方程(1.1)从 x_n 到 x_{n+1} 积分， 得

$$y(x_{n+1})=y(x_n)+\int_{x_n}^{x_{n+1}}f(t,y(t))dt. \qquad (1.6)$$

用左矩形公式对右端积分进行数值计算， 再以 y_n 代替 $y(x_n)$， y_{n+1} 代替 $y(x_{n+1})$ 就得到公式（1.3）

如果用右矩形公式近似右端积分，则得另一个公式

$$y_{n+1}=y_n+hf(x_{n+1},y_{n+1}), \qquad (1.7)$$

称为后退的欧拉法.

后退的欧拉公式与欧拉公式有着本质的区别. 欧拉方法是显式方法，即 y_{n+1} 由 x_n,y_n,x_{n+1},h 明显地表示出来的，这类公式称作是显式的. 然而公式(1.7)的右端含有未知

的 y_{n+1}，它实际上是关于 y_{n+1} 的一个函数方程，这类公式称作是隐式的．隐式方程(1.7)通常用迭代法求解，而迭代过程的实质是逐步显式化．

设欧拉公式

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

给出迭代初值 $y_{n+1}^{(0)}$，用它代入(1.3)式的右端，使之转化为显式，直接计算得

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

然后再用 $y_{n+1}^{(1)}$ 代入(1.7)式，又有

$$y_{n+1}^{(2)}=y_n+hf\left(x_{n+1},y_{n+1}^{(1)}\right)\ .$$

如此反复进行，得

$$y_{n+1}^{(k+1)}=y_n+hf\left(x_{n+1},y_{n+1}^{(k)}\right),\qquad k=0,1,\cdots.\tag{1.8}$$

用迭代法解当然要考虑迭代过程收敛的条件，由于 $f(x,y)$ 对 y 满足利普希茨条件(1.2)． 由（1.8）减 （1.7）得

$$\left|y_{n+1}^{(k+1)}-y_{n+1}\right|=h\left|f\left(x_{n+1},y_{n+1}^{(k)}\right)-f\left(x_{n+1},y_{n+1}\right)\right|\le hL\left|y_{n+1}^{(k)}-y_{n+1}\right|.$$

由此可知，只要 $hL<1$ 迭代法(1.8)就收敛到解 y_{n+1}． 关于后退欧拉方法的公式误差，从积分公式看到它与欧拉法是相似的．

6.1.2 梯形方法

为得到比欧拉法精度高的计算公式，在等式（1.1）中若用梯形求积公式近似右端积分，并用 y_n 代替 $y(x_n)$， y_{n+1} 代替 $y(x_{n+1})$，则得

$$y_{n+1}=y_n+\frac{h}{2}\left[f(x_n,y_n)+f(x_{n+1},y_{n+1})\right],\tag{1.9}$$

称此方法为梯形方法．

梯形公式（1.9）的右端含有未知的 y_{n+1}，因此梯形方法是隐式单步法，可用迭代法求解，仍用欧拉方法提供迭代初值，则梯形法的迭代公式为

$$\begin{cases}y_{n+1}^{(0)}=y_n+hf(x_n,y_n),\\ y_{n+1}^{(k+1)}=y_n+\dfrac{h}{2}\left[f(x_n,y_n)+f\left(x_{n+1},y_{n+1}^{(k)}\right)\right],\\ (k=0,1,2,\cdots).\end{cases}\tag{1.10}$$

为了分析迭代过程的收敛性，将（1.10）式与（1.9）式相减，得

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

于是有

$$\left|y_{n+1}-y_{n+1}^{(k+1)}\right| \leqslant \frac{hL}{2}\left|y_{n+1}-y_{n+1}^{(k)}\right|,$$

式中 L 为 $f(x,y)$ 关于 y 的利普希茨常数．当选取 h 充分小，使得

$$\frac{hL}{2}<1,$$

则当 $k\to\infty$ 时有 $y_{n+1}^{(k)}\to y_{n+1}$，即迭代过程(1.9)是收敛的．

6.1.3 单步法的局部截断误差与阶

初值问题(1.1)的单步法可用一般形式表示为

$$y_{n+1}=y_n+h\varphi(x_n,y_n,y_{n+1},h), \tag{1.11}$$

这里函数 $\varphi(x,y,h)$ 称为增量函数，它可以是 y_n 的非线性函数．$\varphi(x,y,h)$ 依赖于所给定的微分方程（1.1）．例如欧拉法中，$\varphi(x,y,h)=f(x,y)$，与步长 h 无关．它的局部截断误差已给出，那么对一般显式单步法则可如下定义．

定义 1 设 $y(x)$ 是初值问题(1.1)的准确解，称

$$T_{n+1}=y(x_{n+1})-y(x_n)-h\varphi(x_n,y(x_n),h) \tag{1.12}$$

为显式单步法（1.11）的局部截断误差．

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

$$y(x_{n+1})-y_{n+1}=y(x_{n+1})-[y_n+h\varphi(x_n,y_n,h)]$$

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

所以， 局部截断误差可理解为用方法(1.11)计算一步的误差， 也即公式(1.11)中用准确解 $y(x)$ 代替数值解产生的公式误差．根据定义，显然欧拉法的局部截断误差

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

$$=y(x_n+h)-y(x_n)-hy'(x_n)$$

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

这里 $\frac{h^2}{2}y''(x_n)$ 称为局部截断误差主项．显然，$T_{n+1}=O(h^2)$．一般情形的定义如下：

定义 2　显式单步法称为是 p 阶的，如果对于初值问题（1.1）的精确解 $y(x)$，p 是使

$$T_{n+1}=y(x+h)-y(x)-h\varphi(x,y,h)=O\left(h^{p+1}\right), \tag{1.13}$$

成立的最大整数．若将(1.13)展开式写成

$$T_{n+1}=\psi\left(x_n,y(x_n)\right)h^{p+1}+O\left(h^{p+2}\right),$$

则 $\psi\left(x_n,y(x_n)\right)h^{p+1}$ 称为**局部截断误差主项**.

以上定义对隐式单步法也是适用的．例如，对后退欧拉法(1.7)其局部截断误差为

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

$$=hy'(x_n)+\frac{h^2}{2}y''(x_n)+O\left(h^3\right)-h\left[y'(x_n)+hy''(x_n)+O\left(h^2\right)\right]$$

$$=-\frac{h^2}{2}y''(x_n)+O\left(h^3\right).$$

这里 $p=1$，是 1 阶方法，局部截断误差主项为 $-\dfrac{h^2}{2}y''(x_n)$.

同样对梯形法(1.9)有

$$T_{n+1}=y(x_{n+1})-y(x_n)-\frac{h}{2}\left[y'(x_n)+y'(x_{n+1})\right]$$

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

$$-\frac{h}{2}\left[y'(x_n)+y'(x_n)+hy''(x_n)+\frac{h^2}{2}y'''(x_n)\right]+O(h^4)$$

$$=-\frac{h^3}{12}y'''(x_n)+O\left(h^4\right).$$

所以梯形方法(1.9)是二阶的，其局部误差主项为 $-\dfrac{h^3}{12}y'''(x_n)$.

6.1.4 改进的欧拉公式

我们看到，梯形方法虽然提高了精度，但其算法复杂．在应用迭代公式（1.9）进行实际计算时，每迭代一次，都要重新计算函数 $f(x,y)$ 的值，而迭代又要反复进行若干次，计算量很大，而且往往难以预测．为了克服这一缺点，我们先用欧拉公式求得一个初步的近似值 $\bar{y}_{n+1}$，称之为预测值．预测值 $\bar{y}_{n+1}$ 的精度可能很差，再用梯形公式(1.9)式校正一次得 y_{n+1}，而这样建立的预测-校正系统通常称为**改进的欧拉公式**.

预测　$\bar{y}_{n+1}=y_n+hf(x_n,y_n)$；

校正　$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},\ y_n+hf(x_n,y_n))]$；　（1.14）

即使用下面公式

$$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},\bar{y}_{n+1})],\tag{1.15}$$

或表为下列平均简化式

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

例 2　取步长 $h=0.1$，分别用欧拉方法及改进的欧拉方法求解初值问题

$$\begin{cases} \frac{\mathrm{d}y}{\mathrm{d}x}=-y(1+xy), \quad 0\le x\le 1, \\ y(0)=1. \end{cases}$$

解　这个初值问题的准确解为 $y(x)=1/(2e^x-x-1)$．根据题设知

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

(1) 欧拉方法的计算式为

$$y_{n+1}=y_n-0.1\times[y_n(1+x_ny_n)],$$

由 $y_0=y(0)=1$，得

$$y_1=1-0.1\times[1\times(1+0\times1)]=0.9,$$
$$y_2=0.9-0.1\times[0.9\times(1+0.1\times0.9)]=0.8019,$$

这样继续计算下去，其结果列于表 6—2.

(2) 改进的 Euler 方法的计算式为

$$\begin{cases} y_p=y_n-0.1\times[y_n(1+x_ny_n)], \\ y_c=y_n-0.1\times[y_p(1+x_{n+1}y_p)], \\ y_{n+1}=\frac{1}{2}(y_p+y_c), \end{cases}$$

由 $y_0=y(0)=1$，得

$$\begin{cases} y_p=1-0.1\times[1\times(1+0\times1)]=0.9, \\ y_c=1-0.1\times[0.9\times(1+0.1\times0.9)]=0.9019, \\ y_1=\frac{1}{2}(0.9+0.9019)=0.90095 \end{cases}$$

$$\begin{cases} y_p = 0.90095 - 0.1\times[0.90095\times(1+0.1\times 0.90095)] = 0.80274, \\ y_c = 0.90095 - 0.1\times[0.80274\times(1+0.2\times 0.80274)] = 0.80779, \\ y_2 = \dfrac{1}{2}(0.80274+0.80779) = 0.80526 \end{cases}$$

这样继续计算下去，其结果列于表 6—2.

表 6—2 计算结果

	欧拉方法	改进的欧拉方法	准确值
x_n	y_n	y_n	$y(x_n)$
0.1	0.9000000	0.9009500	0.9006235
0.2	0.8019000	0.8052632	0.8046311
0.3	0.7088491	0.7153279	0.7144298
0.4	0.6228902	0.6325651	0.6314529
0.5	0.5450815	0.5576153	0.5563460
0.6	0.4757177	0.4905510	0.4891800
0.7	0.4145675	0.4310681	0.4296445
0.8	0.3610801	0.3786397	0.3772045
0.9	0.3145418	0.3326278	0.3312129
1.0	0.2741833	0.2923593	0.2909884

从表 6-2 可以看出欧拉方法的计算结果只有 2 位有效数字，而改进的欧拉方法却有 3 位有效数字，这表明改进的欧拉方法的精度比欧拉方法高.

6.2 Runge—Kutta 方法

6.2.1 Runge—Kutta 方法的思想

根据微分中值定理，存在 $0<\theta<1$，使得

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

于是，利用方程 $y'(x)=f(x,y)$，得到

$$y(x_{n+1}) = y(x_n) + hf(x_n+\theta h, y(x_n+\theta h)). \qquad (2.1)$$

若记

$$K^* = f(x_n+\theta h, y(x_n+\theta h)),$$

则可将 K^* 视作函数 $y(x)$ 在区间 $[x_n, x_{n+1}]$ 上的平均斜率．由此可见，只要对平均斜率 K^* 作出一种算法，那么由（2.1）式便可相应地导出一种数值方法．例如，当取点 x_n 处的斜率

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

作为平均斜率 K^* 时就得到欧拉法(1.3)，这样做精度当然比较低．而对改进的欧拉法(1.15)，它可以改写成为下面的形式

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(K_1 + K_2), \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_{n+1}, y_n + hK_1). \end{cases} \tag{2.2}$$

可见改进的欧拉法实际上是取 x_n 和 x_{n+1} 两个点上的斜率值 K_1 和 K_2 的算术平均值作为平均斜率 K^* 而构造出来的．这里的 K_2 是通过已知信息 y_n 来预测的．

这个处理过程启发我们，为了构造出较高精度的数值计算公式，可以设法在 $[x_n, x_{n+1}]$ 内多预测几个点的斜率值 $K_1, K_2, \cdots, K_n$，然后将它们加权平均作为 $[x_n, x_{n+1}]$ 上的平均斜率．这一节要推导的高精度单步法 Runge—Kutta（龙格—库塔）法就是根据这一思想来构造的．

r 级 Runge—Kutta 方法（简称 R—K 方法）是这样定义的：

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h), \tag{2.3}$$

其中

$$\varphi(x_n, y_n, h) = \sum_{i=1}^{r} c_i K_i, \tag{2.4}$$

$$K_1 = f(x_n, y_n),$$

$$K_i = f(x_n + \lambda_i h, y_n + h\sum_{j=1}^{i-1} \mu_{ij} K_j), \quad i = 2,3,\cdots,r \tag{2.5}$$

这里 c_i，λ_i，μ_{ij} 均为待定常数，称为 Runge—Kutta 方法的系数．

Runge—Kutta 方法中的系数是这样确定的：设 $y(x)$ 满足微分方程（1.1），将（2.5）式在 x_n 点展开成关于 h 的 Taylor 级数，然后代入到（2.4）式．同时将 $y(x_n + h)$ 在点 x_n 展开成 Taylor 级数，使左右两端 h 的方次不超过 p 的项的系数相等，就得到确定系数 c_i，λ_i，μ_{ij}

的方程组，求该方程组的解也就得到 r 级 p 阶的 Runge—Kutta 方法的算式.

6.2.2 二阶显式 Runge—Kutta 方法

对 $r=2$ 的 R-K 方法，由（2.3），（2.4），（2.5）可得到如下的计算公式

$$\begin{cases} y_{n+1}=y_n+h(c_1K_1+c_2K_2), \\ K_1=f(x_n,y_n), \\ K_2=f(x_n+\lambda_2 h, y_n+\mu_{21}hK_1), \end{cases} \tag{2.6}$$

这里 c_1，c_2，λ_2，μ_{21} 均为待定常数，我们希望适当选取这些系数，使公式阶数 p 尽量高. 根据局部截断误差定义，（2.6）的局部截断误差为

$$T_{n+1}=y(x_{n+1})-y(x_n)-h[c_1f(x_n,y_n)+c_2f(x_n+\lambda_2h,y_n+\mu_{21}hf_n)], \tag{2.7}$$

这里 $y_n=y(x_n)$， $f_n=f(x_n,y_n)$. 为得到 T_{n+1} 的阶 p，要将上式各项在 (x_n,y_n) 处泰勒展开，由于 $f(x,y)$ 是二元函数，故要用到二元泰勒展开，各项展开式为

$$y(x_{n+1})=y_n+hy_n'+\frac{h^2}{2}y_n''+\frac{h^3}{3!}y_n'''+O(h^4),$$

其中

$$\begin{cases} y_n'=f(x\ ,y_n)=f_n, \\ y_n''=\dfrac{d}{dx}f(x_n,y(x_n))=f_x'(x_n,y_n)+f_y'(x_n,y_n)\cdot f_n, \\ y_n'''=f_{xx}''(x_n,y_n)+2f_nf_{xy}'(x_n,y_n)+f_n^2f_{yy}''(x_n,y_n) \\ \qquad\qquad +f_y'(x_n,y_n)[f_x'(x_n,y_n)+f_nf_y'(x_n,y_n)], \end{cases} \tag{2.8}$$

$$f(x_n+\lambda_2h,y_n+\mu_{21}hf_n)=f_n+f_x'(x_n,y_n)\lambda_2h+f_y'(x_n,y_n)\mu_{21}hf_n+O(h^2).$$

将以上结果代入(2.7) 则有

$$\begin{aligned} T_{n+1}=&hf_n+\frac{h^2}{2}\left[f_x'(x_n,y_n)+f_y'(x_n,y_n)f_n\right] \\ &-h\left[c_1f_n+c_2\left(f_n+\lambda_2f_x'(x_n,y_n)h\right)+\mu_{21}f_y'(x_n,y_n)f_nh\right]+O\left(h^3\right) \\ =&(1-c_1-c_2)f_nh+\left(\frac{1}{2}-c_2\lambda_2\right)f_x'(x_n,y_n)h^2+\left(\frac{1}{2}-c_2\mu_{21}\right)f_y'(x_n,y_n)f_nh^2+O\left(h^3\right). \end{aligned}$$

要使公式（2.5）具有 $p=2$ 阶，必须使

$$1-c_1-c_2=0, \frac{1}{2}-c_2\lambda_2=0, \frac{1}{2}-c_2\mu_{21}=0,$$

即
$$c_2\lambda_2=\frac{1}{2},\quad c_2\mu_{21}=\frac{1}{2},\quad c_1+c_2=1, \tag{2.9}$$

（2.9）的解是不唯一的．可令 $c_2=a\neq 0$，则得 $c_1=1-a$，$\lambda_2=\mu_{21}=\dfrac{1}{2a}$.
这样得到的公式称为二阶 Runge—Kutta 方法，如取 $a=1$，则

$$c_1=c_2=1/2，\ \lambda_2=\mu_{21}=1.$$

这就是改进欧拉法（2.2）.

若取 $a=1$，则 $c_2=1$，$c_1=0$，$\lambda_2=\mu_{21}=1/2$，得计算公式

$$\begin{cases} y_{n+1}=y_n+hk_2, \\ K_1=f(x_n,y_n), \\ K_2=f\left(x_n+\dfrac{h}{2},y_n+\dfrac{h}{2}K_1\right). \end{cases} \tag{2.10}$$

称为**中点公式**，相当于数值积分的中矩形公式．（2.10）也可表示为

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

对 $r=2$ 的 R-K 公式（2.6）能否使局部误差提高到 $O(h^4)$？为此需把 K_2 多展开一项，从（2.8）的 y_n'' 看到，展开式中 $f_y'f_x'+f_y'f$ 的项是不能通过选择参数消掉的，实际上要使 h^3 的项为零，需增加 3 个方程，要确定 4 个参数 c_1,c_2,λ_2 及 μ_{21}，这是不可能的． 故 $r=2$ 的显式 R-K 方法的阶只能是 $p=2$，而不能得到三阶公式.

类似地，可建立三阶 Runge—Kutta 方法.

6. 2. 3 三阶与四阶显式 Runge—Kutta 方法

要得到三阶显式 Runge—Kutta 方法，必须 $r=3$．此时(2. 3)，(2. 4)，(2.5）的公式表示为

$$\begin{cases} y_{n+1}=y_n+h(c_1K_1+c_2K_2+c_3K_3), \\ K_1=f(x_n,y_n), \\ K_2=f(x_n+\lambda_2h\ ,y_n+\mu_{21}hK_1), \\ K_3=f(x_n+\lambda_3h\ ,y_n+\mu_{31}hK_1+\mu_{32}hK_2). \end{cases} \tag{2.11}$$

其中 c_1,c_2,c_3 及 $\lambda_2,\mu_{21},\lambda_3,\mu_{31},\mu_{32}$ 均为待定参数，公式（2. 11）的局部截断误差为

$$T_{n+1}=y(x_{n+1})-y(x_n)-h[c_1K_1+c_2K_2+c_3K_3].$$

只要将 K_2,K_3 按二元函数泰勒展开，使 $T_{n+1}=o(h^4)$, 可得待定参数满足方程

$$
\begin{cases}
\lambda_2 = \mu_{21}, \\
\lambda_3 = \mu_{31} + \mu_{32}, \\
c_1 + c_2 + c_3 = 1, \\
c_2\lambda_2 + c_3\lambda_3 = \dfrac{1}{2}, \\
c_2\lambda_2^2 + c_3\lambda_3^2 = \dfrac{1}{3}, \\
c_3\lambda_2\mu_{32} = \dfrac{1}{6}.
\end{cases} \tag{2.12}
$$

这是8个未知数6个方程的方程组,解也不是唯一的.可以得到很多公式.满足条件(2.12)的公式(2.11)统称为三阶龙格-库塔公式.

特别地,当取$c_1 = \frac{1}{4}, c_2 = 0, c_3 = \frac{3}{4}, \lambda_2 = \frac{1}{3}, \lambda_3 = \frac{2}{3}, \mu_{21} = \frac{1}{3}, \mu_{31} = 0, \mu_{32} = \frac{2}{3}$时,得到算式

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{4}(K_1 + 3K_3), \\
K_1 = f(x_n, y_n), \\
K_2 = f(x_n + \dfrac{h}{3}, y_n + \dfrac{h}{3}K_1), \\
K_3 = f(x_n + \dfrac{2}{3}h, y_n + \dfrac{2}{3}hK_2).
\end{cases} \tag{2.13}
$$

此公式称为**三阶** Heun 公式.

当取$c_1 = \frac{1}{6}, c_2 = \frac{2}{3}, c_3 = \frac{1}{6}, \lambda_2 = \frac{1}{2}, \lambda_3 = 1, \mu_{32} = 2$时,代入(2.11)得到算式

$$
\begin{cases}
y_{n+1} = y_n + \dfrac{h}{6}(K_1 + 4K_1 + K_3), \\
K_1 = f(x_n, y_n), \\
K_2 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1), \\
K_3 = f(x_n + h, y_n - hK_1 + 2hK_2).
\end{cases} \tag{2.14}
$$

算式(2.14)称为**三阶 Kutta** 方法,它曾是一种较为广泛使用的三阶方法,比三阶 Heun 方法(2.13)更便于计算.我们还可以推导出更高级高阶的龙格-库塔方法.我们介绍两个常用的四阶的 Runge—Kutta 方法.它们分别由下式给出

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1), \\ K_3 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_2), \\ K_4 = f(x_n + h, y_n + hK_3). \end{cases} \tag{2.15}$$

（2.15）式是最广泛应用的四阶方法．另一个算式是

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{8}(K_1 + 3K_2 + 3K_3 + K_4), \\ K_1 = f(x_n, y_n), \\ K_2 = f(x_n + \dfrac{h}{3}, y_n + \dfrac{h}{3}K_1), \\ K_3 = f(x_n + \dfrac{2}{3}h, y_n - \dfrac{1}{3}hK_1 + K_2), \\ K_4 = f(x_n + h, y_n + hK_1 - hK_2 + hK_3). \end{cases} \tag{2.16}$$

算式（2.16），也称为经典的 Runge—Kutta 方法．特别，当函数 $f(x,y)$ 与 y 无关时，（2.16）式被称为 Simpson 公式．四阶 Runge—Kutta 方法的每一步需要计算四次函数值 f ，可以证明其截断误差为 $O(h^5)$. 不过证明极其繁琐，这里从略．

例 3 设步长 $h = 0.2$，从 $x = 0$ 直到 $x = 1$ 用四阶 Runge—Kutta 方法求解初值问题(1.4)．

解 这里经典的四阶 Runge—Kutta 公式(2.15)具有形式

表 6-3 计算结果

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ K_1 = y_n - \dfrac{2x_n}{y_n}, \\ K_2 = y_n + \dfrac{h}{2}K_1 - \dfrac{2x_n + h}{y_n + \dfrac{h}{2}K_1}, \\ K_3 = y_n + \dfrac{h}{2}K_2 - \dfrac{2x_n + h}{y_n + \dfrac{h}{2}K_2}, \\ K_4 = y_n + hK_3 - \dfrac{2(x_n + h)}{y_n + hK_3}. \end{cases}$$

x_n	y_n	$y(x_n)$
0.2	1.1832	1.1832
0.4	1.3417	1.3416
0.6	1.4833	1.4832
0.8	1.6125	1.6125
1.0	1.7312	1.7321

表 6—3 列出计算结果 y_n，表中 $y(x_n)$ 仍表示准确解.

比较例 3 和例 2 的计算结果，显然 Runge—Kutta 方法的精度更高．要注意，虽然四阶 Runge—Kutta 方法的计算量（每一步要 4 次计算函数 f）比改进的方法（它是一种二阶 Runge—Kutta 方法，每一步只要 2 次计算函数 f）大一倍，但由于这里放大了步长($h = 0.2$)，故表 6-3 和表 6-2 所耗费的计算量几乎相同．这个例子又一次显示了选择算法的重要意义.

然而值得指出的是，Runge—Kutta 方法的推导基于泰勒展开方法，因而它要求所求的解具有较好的光滑性质．反之，如果解的光滑性差，那么，使用四阶 Runge—Kutta 方法求得的数值解，其精度可能反而不如改进的欧拉方法．实际计算时，我们应针对问题的具体特点选择合适的算法.

6.3 单步法的收敛性与稳定性

6.3.1 收敛性

数值解法的基本思想是，通过某种离散化手段将微分方程(1.1)转化为差分方程，如单步法(1.11)，即

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h). \tag{3.1}$$

它在 x_n 处的解为 y_n，而初值问题(1.1)在 x_n 处的精确解为 $y(x_n)$，记 $e_n = y(x_n) - y_n$ 称为整体截断误差．收敛性就是讨论当 $x = x_n$ 固定且 $h = \dfrac{x_n - x_0}{n} \to 0$ 时 $e_n \to 0$ 的问题.

定义 3 若一种数值方法（如单步法（3.1））对于固定的 $x_n = x_0 + nh$，当 $h \to 0$ 时有

$y_n \to y(x_n)$, 其中 $y(x)$ 是(1.1)的准确解，则称该方法是**收敛**的.

显然数值方法收敛是指 $e_n = y(x_n) - y_n \to 0$，对单步法(3.1)有下述收敛性定理：

定理 3 假设单步法(3.1)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

$$\left|\varphi(x, y, h) - \varphi(x, \bar{y}, h)\right| \le L_\varphi \left|y - \bar{y}\right|, \tag{3.2}$$

又设初值 y_0 是准确的，即 $y_0 = y(x_0)$，则其**整体截断误差**有以下估计

$$\left|e_n\right| \le \left|e_0\right| e^{TL_\varphi} + \frac{Ch^p}{L_\varphi}\left(e^{TL_\varphi} - 1\right). \tag{3.3}$$

证明 设以 $\bar{y}_{n+1}$ 表示取 $y_n = y(x_n)$ 用公式(3.1)求得的结果，即

$$\bar{y}_{n+1} = y(x_n) + h\varphi(x_n, y(x_n), h), \tag{3.4}$$

则 $y(x_{n+1}) - \bar{y}_{n+1}$ 为局部误差，由于所给方法具有 p 阶精度，按定义 2，存在定数 C，得

$$\left|\bar{y}_{n+1} - y_{n+1}\right| \le \left|y(x_n) - y_n\right| + h\left|\varphi(x_n, y(x_n), h) - \varphi(x_n, y_n, h)\right|.$$

利用假设条件（3.2），有

$$\left|\bar{y}_{n+1} - y_{n+1}\right| \le \left(1 + hL_\varphi\right)\left|y(x_n) - y_n\right|,$$

从而有

$$\begin{aligned}\left|y(x_{n+1}) - y_{n+1}\right| &\le \left|\bar{y}_{n+1} - y_{n+1}\right| + \left|y(x_{n+1}) - \bar{y}_{n+1}\right| \\ &\le \left(1 + hL_\varphi\right)\left|y(x_n) - y_n\right| + Ch^{p+1}.\end{aligned}$$

即对整体截断误差 $e_n = y(x_n) - y_n$ 成立下列递推关系式

$$\left|e_{n+1}\right| \le \left(1 + hL_\varphi\right)\left|e_n\right| + Ch^{p+1},$$

据此不等式反复递推，可得

$$\left|e_n\right| \le \left(1 + hL_\varphi\right)^n \left|e_0\right| + \frac{Ch^p}{L_\varphi}\left[\left(1 + hL_\varphi\right)^n - 1\right].$$

再注意到当 $x_n - x_0 = nh \le T$ 时

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

最终得下列估计式

$$|e_n| \le |e_0| e^{TL_\varphi} + \frac{Ch^p}{L_\varphi}\left(e^{TL_\varphi} - 1\right). \tag{3.5}$$

由此可以断定，如果初值是准确的，即 $e_0 = 0$, 则(3.3)式成立，证毕.

依据这一定理，判断单步法（3.1）的收敛性，归结为验证增量函数 φ 能否满足利普希茨条件(3.2)．对于欧拉方法，由于其增量函数 φ 就是 $f(x,y)$，故当 $f(x,y)$ 关于 y 满足利普希茨条件时它是收敛的.

再考察改进的欧拉方法（1.15），它可表为

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

此时增量函数

$$\varphi(x_n, y_n, h) = \frac{1}{2}\left[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))\right].$$

这时有

$$\begin{aligned} |\varphi(x,y,h) - \varphi(x,\bar{y},h)| \le \frac{1}{2}[\, &|f(x,y) - f(x,\bar{y})| \\ &+ |f(x+h, y+hf(x,y)) - f(x+h, \bar{y}+hf(x,\bar{y}))|\,]. \end{aligned}$$

假设 $f(x,y)$ 关于 y 满足利普希茨条件，记利普希茨常数为 L，则由上式推得

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

设限定 $h \le h_0$（h_0 为定数），上式表明 φ 关于 y 的利普希茨常数

$$L_\varphi = L\left(1 + \frac{h_0}{2}L\right),$$

因此改进的欧拉方法也是收敛的.

6.3.2 绝对稳定性与绝对稳定域

在实际进行计算时，一方面初始值不一定是完全精确的，带有一定的误差；另一方面，由于计算机的字长有限，在运算中一般总会产生舍入误差，在逐次计算下去的时候，初始数据的误差（或称之为摄动）以及在计算过程中产生的舍入误差，都会传播下去，对以后的计算结果产生影响．所谓的稳定性问题，就是指误差的积累是受到控制的问题．粗略地说，如果计算结果对初始数据的误差以及计算过程中的舍入误差不敏感，就是说明相应的计算方法是稳定的，否则就称为不稳定的.

定义 4 若一种数值方法在节点值 y_n 上大小为 δ 的扰动，与以后各节点 $y_m\ (m>n)$ 上产生的偏差均不超过 δ，则称该方法是**稳定**的.

这里说的稳定性是描述当 $h \to 0$ 时误差对其计算的影响．然而在实际计算中，步长是固定的，并非充分小．为了刻画这种情况下的误差传播和积累情况，引入绝对稳定性概念，

由于绝对稳定性的复杂性，在考虑方法的绝对稳定性时，一般只限于典型方程

$$y' = \lambda y . \qquad (3.6)$$

为了保证方法的绝对稳定，步长 h 和 λ 值都要受到一定的限制．它们的允许范围，就称为相应方法的绝对稳定域．

定义 5 单步法（3.1）用于解模型方程（3.6），若得到的解 $y_{n+1} = E(h\lambda)y_n$，满足 $|E(h\lambda)| < 1$，则称方法（3.1）是**绝对稳定**的．在 $\mu = h\lambda$ 的平面上，使 $|E(h\lambda)| < 1$ 的变量围成的区域，称为**绝对稳定域**，它与实轴的交集称为**绝对稳定区间**．

下面先研究欧拉方法的稳定性．模型方程 $y' = \lambda y$ 的欧拉公式为

$$y_{n+1} = (1 + h\lambda)y_n . \qquad (3.7)$$

设在节点值 y_n 上有一扰动值 ε_n，它的传播使节点值 y_{n+1} 产生大小为的 ε_{n+1} 扰动值，假设用 $y_n^* = y_n + \varepsilon_n$ 按欧拉公式得出 $y_{n+1}^* = y_{n+1} + \varepsilon_{n+1}$ 的计算不再有新的误差，则扰动值满足

$$\varepsilon_{n+1} = (1 + h\lambda)\varepsilon_n \quad .$$

可见扰动值满足原来的差分方程．这样，如果差分方程的解是不增长的，即有

$$|y_{n+1}| \le |y_n| ,$$

则它就是稳定的．这一讨论对于下面研究的其他方法同样适用．

显然，为要保证差分方程（3.7）的解是不增长的，只要选取充分小 h，使

$$|1 + h\lambda| \le 1 . \qquad (3.8)$$

在 $\mu = h\lambda$ 的复平面上，这是以为 $(-1,0)$ 为圆心，1 为半径的单位圆域，称为欧拉法的绝对稳定域．

对隐式单步法，可以同样讨论方法的绝对稳定性．例如对后退欧拉法，用它解模型方程（3.6）可得

$$y_{n+1} = \frac{1}{1 - h\lambda} y_n ,$$

故

$$E(h\lambda) = \frac{1}{1 - h\lambda} ,$$

由 $|E(h\lambda)| = \left|\dfrac{1}{1 - h\lambda}\right| < 1$ 可得绝对稳定域为 $|1 - h\lambda| > 1$，它是以 $(1,0)$ 为圆心，1 为半径的单位圆外部，故绝对稳定区间为 $-\infty < h\lambda < 0$．当 $\lambda < 0$ 时，则 $0 < h < \infty$，即对任何步长均为稳定的．

对隐式梯形法，解模型方程（3.6）得

$$y_{n+1}=\frac{1+\dfrac{h\lambda}{2}}{1-\dfrac{h\lambda}{2}}y_n,$$

故
$$E(h\lambda)=\frac{1+h\lambda/2}{1-h\lambda/2},$$

对 $\mathrm{Re}(\lambda)<0$，有 $|E(h\lambda)|=\left|\dfrac{1+\dfrac{h\lambda}{2}}{1-\dfrac{h\lambda}{2}}\right|<1$，故绝对稳定域为 $\mu=h\lambda$ 的左半平面，绝对稳定区间为 $-\infty<h\lambda<0$，即当 $0<h<+\infty$，梯形法均是稳定的.

6.4 线性多步法

计算常微分方程初值问题（1.1）的数值解的单步方法是利用点 x_n 上的数值解 y_{n+1} 的值．这样的方法其优点是概念上清楚，编制计算程序方便，其缺点是由于它未能充分利用已经提供的信息，所以效率低．如果 y_{n+1} 不仅仅依赖于 y_n，还依赖于 $y_{n-1},y_{n-2},\cdots,y_{n-k}$，那么在计算时利用 $y_n,y_{n-1},y_{n-2},\cdots,y_{n-k}$ 这 $k+1$ 个值其结果会更令人满意．鉴于这个思想，构造出一些“多步方法”.

6.4.1 线性多步法的一般公式

如果计算 y_{n+k} 时，除用 y_{n+k-1} 的值，还用到 $y_{n+i}(i=0,1,\cdots,k-2)$ 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

$$y_{n+k}=\sum_{i=0}^{k-1}\alpha_i y_{n+i}+h\sum_{i=0}^{k}\beta_i f_{n+i},\tag{4.1}$$

其中 y_{n+k} 为 $y(x_{n+k})$ 的近似值，α_i,β_i 为常数，α_0 及 β_0 不全为零，则称（4. 1）为线性 k 步法．计算时需先给出前面 k 个近似值 $y_0,y_1,\cdots,y_{k-1}$,，再由（4.1）逐次求出 $y_k,y_{k+1},\cdots$．如果 $\beta_k=0$ 称（4.1）为显式 k 步法，这时 y_{n+k} 可直接由（4.1）算出；如果 $\beta_k\neq0$，则（4.1）称为隐式 k 步法，求解时与梯形法（1.9）相同，要用迭代法方可算出 y_{n+k}．（4.1）中系数 α_i 及 β_i 可根据方法的局部截断误差及阶确定．下面将加以简单介绍.

设 $y(x)$ 是式（1.1）的解，由于式（4.1）对一般的 $y(x)$ 不能准确成立，所以当把 $y(x)$

代入式（4.1）两端时，一般并不相等．记两端的差为T_{n+k}，称T_{n+k}为线性多步法（4.1）从x_{n+k-1}到x_{n+k}这一步的局部截断误差．即

$$T_{n+k}=L[y(x_n),h]=y(x_{n+k})-\sum_{i=0}^{k-1}\alpha_i y(x_{n+i})-h\sum_{i=0}^{k}\beta_i y'(x_{n+i}).\tag{4.2}$$

若假设$y(x)$充分连续可微，就可以将T_{n+k}在x_n处做泰勒展开，由于

$$y(x_n+ih)=y(x_n)+ihy'(x_n)+\frac{(ih)^2}{2!}y''(x_n)+\frac{(ih)^3}{3!}y'''(x_n)+\cdots,$$

$$y'(x_n+ih)=y'(x_n)+ihy''(x_n)+\frac{(ih)^2}{2!}y'''(x_n)+\cdots,$$

代入（4.2）得

$$T_{n+k}=c_0y(x_n)+c_1hy'(x_n)+c_2h^2y''(x_n)+\cdots+c_ph^{(p)}(x_n)+\cdots,\tag{4.3}$$

其中

$$c_0=1-(a_0+\cdots+a_{k-1}),$$

$$c_1=k-[a_1+2a_2+\cdots+(k-1)a_{k-1}]-(\beta_0+\beta_1+\cdots+\beta_k),$$

$$c_q=\frac{1}{q!}\left[k^q-(a_1+2^qa_2)+\cdots+(k-1)^qa_{k-1}\right]-\frac{1}{(q-1)!}\left[\beta_1+2^{q-1}\beta_2+\cdots+k^{q-1}\beta^k\right],$$

$$q=2,3,\cdots,\tag{4.4}$$

若在公式（4.1）中选择系数α_i及β_i，使它满足

$$c_0=c_1=\cdots=c_p=0,\quad c_{p+1}\neq 0,$$

由定义可知此时所构造的多步法是p阶的，且

$$T_{n+k}=c_{p+1}h^{p+1}y^{(p+1)}(x_n)+O(h^{p+2}),\tag{4.5}$$

称右端第一项为**局部截断误差主项**，c_{p+1}称为**误差常数**．

定义 6 称满足条件$c=c_1=0$，即

$$\begin{cases}\alpha_0+\alpha_1+\cdots+\alpha_{k-1}=1\\ \sum_{i=1}^{k-1}i\alpha_i+\sum_{i=0}^{k}\beta_i=k\end{cases}\tag{4.6}$$

的线性多步法（4.1）是相容的.

显然，当$k=1$时，若$\beta_1=0$，则由（4.6）可求得

$$\alpha_0=1 \ ,\ \beta_0=1$$

此时公式（4.1）为

$$y_{n+1}=y_n+hf_n ,$$

即为欧拉法．从（4.4）可求得$c_2=1/2\neq 0$，故方法为 1 阶精度，且局部截断误差为

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

这和第 2 节给出的定义及结果是一致的.

对 $k=1$, 若 $\beta_1\neq 0$ ，此时方法为隐式公式，为了确定系数 α_0，β_0,β_1，可由 $c_0=c_1=c_2=0$ 解得 $\alpha_0=1,\beta_0=\beta_1=\dfrac{1}{2}$．于是得到公式

$$y_{n+1}=y_n+\frac{h}{2}(f_n+f_{n+1}),$$

即为梯形法．由（4.4）可得$c_3=-\dfrac{1}{12}$，故$p=2$，所以梯形法是二阶方法，其局部截断误差主项是$-\dfrac{h^3y'''(x_n)}{12}$，这与第 2 节中的讨论也是一致的．另外，构造多步法的公式有多种途径，这里着重介绍基于数值积分的方法.

我们知道常微分初值问题(1.1)与积分

$$y(x_{n+1})=y(x_n)+\int_{x_n}^{x_{n+1}}f(t,y(t))\mathrm{d}t \tag{4.7}$$

是等价的．对$\dfrac{\mathrm{d}y}{\mathrm{d}x}=f(x,y(x))$作多项式插值，利用插值型求积公式，便可得到相应的线性多步法公式．下面我们讨论最简单的多步法——阿当姆斯方法

6.4.2 阿当姆斯显式

为了导出求微分方程(1.1)的数值解法，我们将(4.7)右边积分的被积函数用插值多项式来近似替代．从插值角度来看，插值多项式次数越高越精确，但不能过高（因为高次插值会出现龙格现象）．这里，我们取三次插值多项．插值节点除x_n，x_{n+1}外，通常还要在$[x_n,x_{n+1}]$内再取两点作为插值节点，但取区间$[x_n,x_{n+1}]$内的点时，函数值又不知道，由于已知$y_{n-1},y_{n-2},y_{n-3},\cdots$，所以我们取插值节点$x_{n-2},x_{n-1},x_n,x_{n+1}$，作三次插值多项式

$$
\begin{aligned}
p_3(t)=&\frac{(t-x_n)(t-x_{n-1})(t-x_{n-2})}{(x_{n+1}-x_n)(x_{n+1}-x_{n-1})(x_{n+1}-x_{n-2})}f(x_{n+1},y(x_{n+1}))\\
&+\frac{(t-x_{n+1})(t-x_{n-1})(t-x_{n-2})}{(x_n-x_{n+1})(x_n-x_{n-1})(x_n-x_{n-2})}f(x_n,y(x_n))\\
&+\frac{(t-x_{n+1})(t-x_n)(t-x_{n-2})}{(x_{n-1}-x_{n+1})(x_{n-1}-x_n)(x_{n-1}-x_{n-2})}f(x_{n-1},y(x_{n-1}))\\
&+\frac{(t-x_{n+1})(t-x_n)(t-x_{n-1})}{(x_{n-2}-x_{n+1})(x_{n-2}-x_n)(x_{n-2}-x_{n-1})}f(x_{n-2},y(x_{n-2}))\\
=&\frac{1}{6h^3}(t-x_n)(t-x_{n-1})(t-x_{n-2})f(x_{n+1},y(x_{n+1}))\\
&-\frac{1}{2h^3}(t-x_{n+1})(t-x_{n-1})(t-x_{n-2})f(x_n,y(x_n))\\
&+\frac{1}{2h^3}(t-x_{n+1})(t-x_n)(t-x_{n-2})f(x_{n-1},y(x_{n-1}))\\
&-\frac{1}{6h^3}(t-x_{n+1})(t-x_n)(t-x_{n-1})f(x_{n-2},y(x_{n-2}))
\end{aligned}
$$

其中 $h=x_i-x_{i-1}\ (i=n+1,n,n-1)$.

插值余项为

$$
r_3(t)=\frac{F^{(4)}(\xi)}{4!}(t-x_{n+1})(t-x_n)(t-x_{n-1})(t-x_{n-2})\,, \tag{4.8}
$$

其中 $F(x)=f(x,y(x))$， $x_{n-2}<\xi<x_{n+1}$，故有

$$
f(t,y(t))=p_3(t)+r_3(t)\,. \tag{4.9}
$$

将(4.9)代入(4.7)右端的积分，并略去 $r_3(t)$，令 $t=x_n+uh$，再将 $y(x_{n-2})$、$y(x_{n-1})$、$y(x_n)$、$y(x_{n+1})$ 分别用近似值 y_{n-2}、y_{n-1}、y_n、y_{n+1} 表示，得

$$
\begin{aligned}
&y_{n+1}=y_n+\frac{1}{6}hf(x_{n+1},y_{n+1})\int_0^1 u(u+1)(u+2)\mathrm{d}u\\
&-\frac{1}{2}hf(x_n,y_n)\int_0^1(u-1)(u+1)(u+2)\mathrm{d}u\\
&-\frac{1}{2}hf(x_{n-1},y_{n-1})\int_0^1(u-1)u(u+2)\mathrm{d}u\\
&-\frac{1}{6}hf(x_{n-2},y_{n-2})\int_0^1(u-1)u(u+1)\mathrm{d}u.\\
&=y_n+\frac{1}{24}h[9f(x_{n+1},y_{n+1})+19f(x_n,y_n)-5f(x_{n-1},y_{n-1})+f(x_{n-2},y_{n-2})]
\end{aligned} \tag{4.10}
$$

称为**四阶隐式**阿当姆斯（**外推**）**公式**.

若插值节点取为 x_n、x_{n-1}、x_{n-2}、x_{n-3}，则三次插值多项式为

$$
\begin{aligned}
\overline{p}_3(t)=&\frac{1}{6h^3}(t-x_{n-1})(t-x_{n-2})(t-x_{n-3})f(x_n,y(x_n))\\
&-\frac{1}{2h^3}(t-x_n)(t-x_{n-2})(t-x_{n-3})f(x_{n-1},y(x_{n-1}))\\
&+\frac{1}{2h^3}(t-x_n)(t-x_{n-1})(t-x_{n-3})f(x_{n-2},y(x_{n-2}))\\
&-\frac{1}{6h^3}(t-x_n)(t-x_{n-1})(t-x_{n-2})f(x_{n-3},y(x_{n-3})),
\end{aligned}
$$

插值余项为

$$
\bar{r}_3(t)=\frac{1}{4!}F^{(4)}(\xi)(t-x_n)(t-x_{n+1})(t-x_{n-2})(t-x_{n-3}),\qquad x_{n-3}<\xi<x_n,
$$

于是有

$$
f(t,y(t))=\overline{p}_3(t)+\bar{r}_3(t). \tag{4.11}
$$

将(4.11)代入(4.7)右端，略去余项，积分，并用 y_i 近似替代

$$
y(x_i)\ (i=n-3,\ n-2,\ n-1,\ n),
$$

便得

$$
y_{n+1}=y_n+\frac{1}{24}h[55f(x_n,y_n)-59f(x_{n-1},y_{n-1})+37f(x_{n-2},y_{n-2})-9f(x_{n-3},y_{n-3})]. \tag{4.12}
$$

这就是**四阶显式**阿当姆斯（**内插**）**公式**.

由余项表示式(4.8)得，四阶隐式阿当姆斯公式的截断误差为

$$
T_{n+1}=\int_{x_n}^{x_{n+1}}\frac{1}{24}F^{(4)}(\xi)(t-x_{n+1})(t-x_n)(t-x_{n-1})(t-x_{n-2})dt.
$$

令 $t=x_n+uh$，由第二积分中值定理得

$$
\begin{aligned}
T_{n+1}&=\frac{1}{24}F^{(4)}(\xi)(t-x_{n+1})(t-x_n)(t-x_{n-1})(t-x_{n-2})\mathrm{d}u\\
&=-\frac{19}{720}h^5y^{(5)}(\eta),x_{n-2}<\eta<x_{n+1}.
\end{aligned} \tag{4.13}
$$

这表明四阶隐式阿当姆斯方法的局部截断误差为 $O(h^5)$.

同理可得四阶显式阿当姆斯方法的误差为

$$
T_{n+1}=\frac{251}{720}h^5y^{(5)}(\eta),x_{n-3}<\eta<x_n. \tag{4.14}
$$

即四阶显式阿当姆斯方法的局部误差也为 $O(h^5)$.

6.5 数值实验及程序

6.5.1 欧拉方法数值实验与程序

解一阶方程$\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$的初值问题的数值解法，就是寻求$y(x)$在一系列离散节点$x_1 < x_2 < \cdots < x_n < x_{n+1} < \cdots$上的近似值$y_1, y_2, \cdots y_n, y_{n+1}, \cdots$．相邻两个节点的间距$h_n = x_{n+1} - x_n$称为步长．若$h_n = h$为常数，这时节点为$x_n = x_0 + nh$，$n = 0,1,2\cdots$．

数值解法一般都采取“步进式”，即求解过程顺着节点排列的次序一步一步地向前推进，描述这类算法，只要给出已知信息$y_n, y_{n-1}, y_{n-2}, \cdots$计算$y_{n+1}$的递推公式.

欧拉公式： $y_{n+1} = y_n + hf(x_n, y_n)$.

若初值y_0已知，则依上式可逐步算出

$y_1 = y_0 + hf(x_0, y_0)$,

$y_2 = y_1 + hf(x_1, y_1)$,

…………

```
matlab 程序如下：(Euler.m)
% 欧拉法求解微分方程 y'(x)=f_name(x，y)，y(x0)=y0
% 在 n 个离散点 x=[x1，x2，……，xn]'的近似值 y=[y1，y2，……，yn]'.
% 其中 xi=x0+i*h.
% 返回值 x 记录横坐标，y 记录对应列坐标，x、y 均为列向量
function [x，y]= Euler(f_name，x0，y0，n，h)
if nargin<5    h=0.01;    end  % 默认步长 h=0.01
if nargin<4    n=10;      end
x=[x0+h:h:x0+n*h]';
y=zeros(n，1);
y(1)=y0+h*feval(f_name，x0，y0);
for i=1:n-1
   y(i+1)=y(i)+h*feval(f_name，x(i)，y(i));
end
fprintf('\n\n 计算结果\n\n') % 打印结果.
fprintf('       xi        yi\n')
for i=1:n
   fprintf('%12f %12f\n'，x(i)，y(i));
end
matlab 程序如下：(f8_.m)
```

```
function z=f8(x，y)
z=y-2*x/y;
```

例 4　求解初值问题

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

解　计算过程如下：

输入：

```
[x，y]= Euler('f8'，0，1，10，0.1);
```

计算结果

```
        xi          yi
    0.100000     1.100000
    0.200000     1.191818
    0.300000     1.277438
    0.400000     1.358213
    0.500000     1.435133
    0.600000     1.508966
    0.700000     1.580338
    0.800000     1.649783
    0.900000     1.717779
    1.000000     1.784771
```

例 5 考察例 4 中欧拉方法的精度.

解　计算过程如下

matlab 程序：(Euler_Compare.m)

```
% 对于微分方程 y'=y-2x/y，y(0)=1，利用欧拉法计算的结果和真实结果比较
% 解析解: y_=(1+2x)^(1/2)
[x，y]= Euler('f8'，0，1，10，0.1);
n=length(x);y_=(1+2*x).^(1/2);
fprintf('\n\n计算结果比较\n\n')
fprintf('      xi         yi          y(xi)\n')
for i=1:n
    fprintf('%12f %12f %12f\n'，x(i)，y(i)，y_(i));
end
fprintf('\n\n绘图分析\n\n');
 plot(x，y，'d-'，x，y_); legend('欧拉法折线图'，'真实函数图像');
```

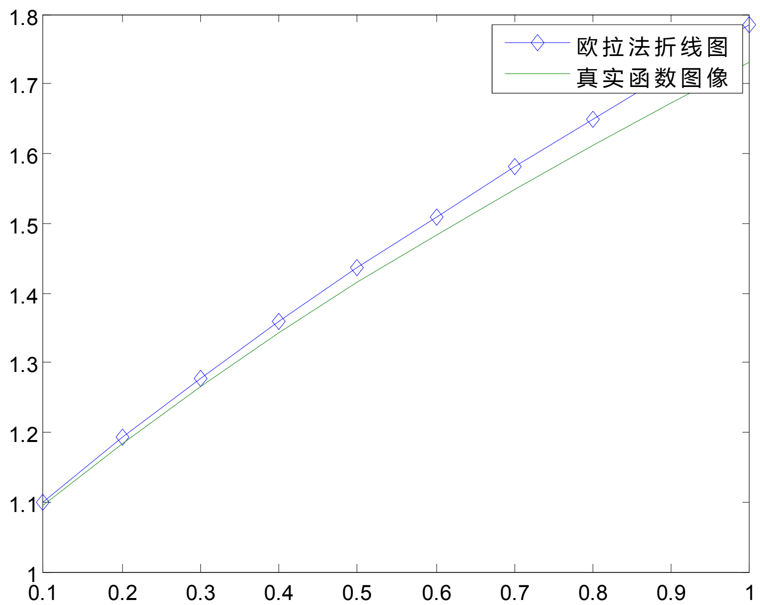
输入：Euler_Compare

输出：计算结果比较

```
        xi          yi           y(xi)
    0.100000     1.100000     1.095445
    0.200000     1.191818     1.183216
    0.300000     1.277438     1.264911
    0.400000     1.358213     1.341641
```

```
0.500000    1.435133    1.414214
0.600000    1.508966    1.483240
0.700000    1.580338    1.549193
0.800000    1.649783    1.612452
0.900000    1.717779    1.673320
1.000000    1.784771    1.732051
```

绘图分析



6.5.2 四阶显式 R-K 方法数值实验与程序

实际上从方程 $y'=f(x,y)$ 等价的积分形式 $y(x_{n+1})-y(x_n)=\int_{x_n}^{x_{n+1}}f(x,y(x))dx$ 可以看成出，若要使公式阶数提高，就必须使右端积分的数值求积公式精度提高，它必然要增加求积节点，为此可将右端用求积公式表示为

$$\int_{x_n}^{x_{n+1}}f(x,y(x))dx\approx h\sum_{i=1}^{r}c_i f(x_n+\lambda_i h,y(x_n+\lambda_i h)).$$

将公式表示为 $y_{n+1}=y_n+h\varphi(x_n,y_n,h)$.

其中：
$$\varphi(x_n,y_n,h)=\sum_{i=1}^{r}c_iK_i,\quad K_1=f(x_n,y_n),$$

$$K_i=f(x_n+\lambda_i h,\ y_n+h\sum_{j=1}^{i-1}\mu_{ij}K_j)\quad i=2,\cdots r$$

这里 c_i，λ_i，μ_{ij} 均为常数．上式方法即为 r 级显式龙格-库塔(Runge-Kutta)法，简称 R-K 方法．当 $r=4$，可得经典的四阶龙格-库塔方法，其计算公式如下：

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

matlab程序如下：（R_K.m）

```
% 四阶显式 R-K 方法求解微分方程 y'(x)=f_name(x，y)，y(x0)=y0
% 在 n 个离散点 x=[x1，x2，……，xn]'的近似值 y=[y1，y2，……，yn]'.
% 其中 xi=x0+i*h.
% 返回值 x 记录横坐标，y 记录对应列坐标，x、y 均为列向量

function [x，y]= R_K(f_name，x0，y0，n，h)
if nargin<5
    h=0.01;  % 默认步长 h=0.01
end
if nargin<4
    n=10;
end

x=[x0+h:h:x0+n*h]';
y=zeros(n，1);

K1=feval(f_name，x0，y0);
K2=feval(f_name，x0+h/2，y0+h/2*K1);
K3=feval(f_name，x0+h/2，y0+h/2*K2);
K4=feval(f_name，x0+h，y0+h*K3);
y(1)=y0+h/6*(K1+2*K2+2*K3+K4);

for i=1:n-1
    K1=feval(f_name，x(i)，y(i));
    K2=feval(f_name，x(i)+h/2，y(i)+h/2*K1);
    K3=feval(f_name，x(i)+h/2，y(i)+h/2*K2);
    K4=feval(f_name，x(i)+h，y(i)+h*K3);
    y(i+1)=y(i)+h/6*(K1+2*K2+2*K3+K4);
end

% 打印结果
fprintf('\n\n 计算结果\n\n')
fprintf('        xi          yi\n')
```

```
for i=1:n
    fprintf('%12f %12f\n', x(i), y(i));
end
```

例 6 求解初值问题

$$\begin{cases} y' = y - \dfrac{2x}{y} & (0 < x < 1) \\ y(0) = 1 \end{cases} .$$

解 计算过程如下：

输入：

```
[x, y]= R_K('f8', 0, 1, 10, 0.1);
```

计算结果

xi	yi
0.100000	1.095446
0.200000	1.183217
0.300000	1.264912
0.400000	1.341642
0.500000	1.414216
0.600000	1.483242
0.700000	1.549196
0.800000	1.612455
0.900000	1.673325
1.000000	1.732056

习 题 六

1. 用梯形法求解初值问题

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

证明其数值解为

$$y_n = \left(\frac{2-h}{2+h}\right)^n,$$

固定 x，取 $h = \dfrac{x}{n}$，证明：$h \to 0$ 时，y_n 收敛于原初值问题的精确解.

2. 导出用 Euler 法求解

$$\begin{cases} y' = \lambda y \\ y(0) = 1 \end{cases},$$

的公式，并证明它收敛于问题的精确解.

3. 用 Euler 公式和改进的 Euler 公式分别求下列初值问题的数值解(取步长 $h = 0.1$ 计算到

y_3)：

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = -2xy \\ y(0) = 1. \end{cases}.$$

4. 取步长 $h = 0.1$，分别用 Euler 方法及改进的 Euler 方法求解初值问题

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = -y(1+xy), \quad 0 \le x \le 1, \\ y(0) = 1. \end{cases}$$

5. 证明: 改进的 Euler 法能精确地解初值问题

$$\begin{cases} y' = ax + b, \\ y(0) = 0. \end{cases}.$$

6. 用梯形方法解常微分方程初值问题

$$\begin{cases} y' = x^2 + 100y^2, \\ y(0) = 0. \end{cases}$$

证明近似解为 $y_n = (\dfrac{2-h}{2+h})^n$.

7. 证明中点公式

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

是二阶的，并求其局部截断误差主项.

8. 证明对任意参数 t，下列 Runge—Kutta 方法是二阶的.

$$\begin{cases} y_{n+1} = y_n + \dfrac{1}{2}h(K_2 + K_3) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + th, y_n + thK_1) \\ K_3 = f(x_n + (1-t)h, y_n + (1-t)hK_1) \end{cases}$$

9. 解初值问题 $y' = 10y - \dfrac{x}{y}, 1 \le x \le 2$，若用梯形公式求解，要使迭代公式

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n) \\ y_{n+1}^{(k+1)} = y_n + \dfrac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})] \end{cases}$$ 收敛，求步长 h 的取值范围.