

第一章:  $f(x) \leftarrow p_n(x)$

第二章:  $\int_a^b f(x)dx \approx \sum_{k=1}^n A_k f(x_k)$

第三章:  $\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}$

# 第三章 常微分方程的差分方法

## § 1 欧拉方法

## § 2 改进的欧拉方法

## § 3 龙格-库塔方法



在工程和科学技术的实际问题中，常需要解常微分方程。但常微分方程组中往往只有少数较简单和典型的常微分方程（例如线性常系数常微分方程等）可求出其解析解。对于变系数常微分方程的解析求解就比较困难，而一般的非线性常微分方程就更不用说了。在大多数情况下，常微分方程只能用近似法求解。这种**近似解法**可分为两大类：一类是**近似解析法**，如级数解法、逐次逼近法等；另一类则是**数值解法**，它给出方程在一些离散点上的近似解。

在具体求解微分方程时，需要具备某种**定解条件**，微分方程和定解条件合在一起组成**定解问题**。定解条



件有两种：一种是给出积分曲线在初始点的状态，称为**初始条件**，相应的定解问题称为**初值问题**；另一种是给出积分曲线首尾两端的状态，称为**边界条件**，相应的定解问题则称为**边值问题**。

例如，弹簧-质量系统的振动问题（[图3-1](#)），作一定的简化后，可用一个二阶常微分方程

$$\frac{d^2 x}{dt^2} + \frac{c}{m} x = 0$$

来描述。式中， $x$ 是质量 $m$ 离平衡位置（0点）的距离； $t$ 是时间； $c$ 是弹簧常数。

当弹簧在振动开始时刻 $t = t_0$ 时的初始位置 $x(t_0) = x_0$ 和初速度

$$\left. \frac{dx}{dt} \right|_{t=t_0} = x'(t_0) = x'_0$$

确定时，弹簧的振动规律 $x(t)$ 也就唯一确定。这就是一个常微分方程的初值问题，可写成：

$$\begin{cases} \frac{d^2 x}{dt^2} + \frac{c}{m} x = 0 \\ x(t_0) = x_0 \\ x'(t_0) = x'_0 \end{cases} \quad (t > t_0)$$

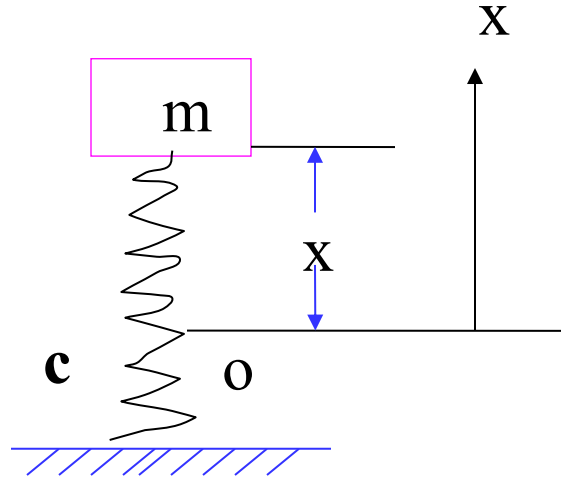


图3-1

本章先从一阶常微分方程的初值问题：

$$\begin{cases} \frac{dy}{dx} = f(x, y), x \in [a, b] \\ y(x_0) = y_0 \end{cases}$$

(1.1)

出发进行讨论。

由常微分方程的理论知，只要上式中的函数 $f(x,y)$ 在区域 $G = \{a \leq x \leq b, -\infty < y < \infty\}$ 内连续，且关于 $y$ 满足李普希兹(Lipschitz)条件，即存在常数 $L$ (它与 $x, y$ 无关)使

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

对  $G = \{a \leq x \leq b, -\infty < y < \infty\}$  内任意两个 $y_1$ 和 $y_2$ 都成立，则方程的解必定存在且唯一。下面的分析均假定满足上述条件。

初值问题 (1.1) 的数值解法，常采用差分方法，即把一个连续的初值问题离散化为一个差分方程来求解。即将 (1.1) 离散化后，求找其解 $y = y(x)$ 在一系列离散节点

$$a = x_0 < x_1 < \dots < x_i < \dots < x_n = b$$

上的近似值  $y_0, y_1, \dots, y_n$ 。

两相邻节点间的距离

$$h_i = x_{i+1} - x_i \quad (i=0,1,2,\dots,n-1)$$

称为步长。当  $h_i = h$ （常值）时称为等步长，有

$$x_i = x_0 + ih, (i = 1, 2, \dots, n)$$

或

$$x_{i+1} = x_i + h, (i = 0, 1, 2, \dots, n-1)$$



因为初值问题中的初始条件  $y(a) = y_0 = \alpha$  为已知，即可利用已知的  $y_0$  来求出下一节点处  $y(x_1)$  的近似值  $y_1$  再从  $y_1$  来求出  $y_2, \dots$ ，如此继续，直到求出  $y_n$  为止。这种用按节点的排列顺序一步一步地向前推进的方式求解的差分算法称为“**步进式**”或“**递进式**”算法。它是初值问题数值解法的各种差分格式的共同特点。因此，只要能写出由前几步已知信息

$$y_0, y_1, \dots, y_i$$

来计算的递推公式（即**差分格式**），即可完全表达该种算法。

## § 1 欧拉方法

### 1.1 欧拉格式

对于初值问题

$$\begin{cases} \frac{dy}{dx} = f(x, y), x \in [a, b] \\ y(x_0) = y_0 \end{cases} \quad (1.1)$$

先将其离散化，即把 $[a, b]$ 作 $n$ 等分，得各离散节点

$$x_i = a + ih \quad (i=0, 1, 2, \dots, n-1)$$

式中  $h=(b-a)/n$

设 $y = y(x)$ 为方程(1.1)的解。

如果用向前差商近似代替 (1.1) 式中的导数项  $y'(x_i)$ , 即

$$y'(x_i) \approx \frac{y(x_{i+1}) - y(x_i)}{h}, (i = 0, 1, 2, \dots, n-1)$$

代入 [\(1.1\)](#), 可得:

$$\begin{aligned} y(x_{i+1}) &\approx y(x_i) + hf(x_i, y(x_i)) \\ (i &= 0, 1, 2, \dots, n-1) \end{aligned}$$


若将 $y(x_{i+1})$ 和 $y(x_i)$ 的近似值分别记为 $y_{i+1}$ 和 $y_i$ ，  
则得

$$y_{i+1} = y_i + hf(x_i, y_i) (i = 0, 1, 2, \dots, n-1) \quad (2)$$


这就是**欧拉（Euler）公式**，又称**欧拉格式**。利用它可由已知的初值 $y_0$ 出发，逐步算出 $y_1, y_2, \dots, y_n$ 。这类形式的方程也称为**差分方程**。

当假定 $y_i$ 为准确，即在 $y_i = y(x_i)$ 的前提下来估计误差 $y(x_{i+1}) - y_{i+1}$ ，这种截断误差称为**局部截断误差**。由(1)和(2)可知，欧拉格式在节点 $x_{i+1}$ 处的局部截断误差显然为：

$y = y(x_{i+1})$  在  $(x_i, y_i)$  点处的泰勒展开式为:

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2} y''(\xi_i)$$


1式

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


2式

如果局部截断误差为  $O(h^{p+1})$ ，则称这种数值算法的精度为P阶。故欧拉格式的精度为一阶。从几何意义上来讲欧拉格式，可如[图3-2](#)中所示。由方程[\(1.1\)](#)知，其积分曲线  $y = y(x)$  上任意一点  $(x, y)$  的切线斜率  $dy/dx$  都等于函数  $f(x, y)$  的值。从初值点  $P_0$ （即点  $(x_0, y_0)$ ）出发，作积分曲线  $y = y(x)$  在  $P_0$  点上的切线  $\overline{P_0 P_1}$ （其斜率为  $f(x_0, y_0)$ ），与直线  $x = x_1$  相交于点  $P_1$ （即点  $(x_1, y_1)$ ），得到  $y_1$  作为  $y(x_1)$  的近似值，则有

$$y_1 = y_0 + (x_1 - x_0) \left. \frac{dy}{dx} \right|_{x=x_0} = y_0 + hf(x_0, y_0)$$

$$y(x_1) = y(x_0) + \int_{x_0}^{x_1} f(x, y(x)) dx$$

相比较可知，这时是用切线段  $\overline{P_0 P_1}$  近似代替了曲线段  $P_0 P_1'$ ； $P_1$  点近似代替了  $P_1'$  点； $y_1$  近似代替了  $y(x_1)$ ； $hf(x_0, y_0)$  近似代替了

$$\int_{x_0}^{x_1} f(x, y(x)) dx$$

递推继续从  $P_1$  点出发，作一斜率为  $f(x_1, y_1)$  的直线

$\overline{P_1 P_2}$ ，与直线  $x = x_2$  相交于  $P_2$  点（即点  $(x_2, y_2)$ ）

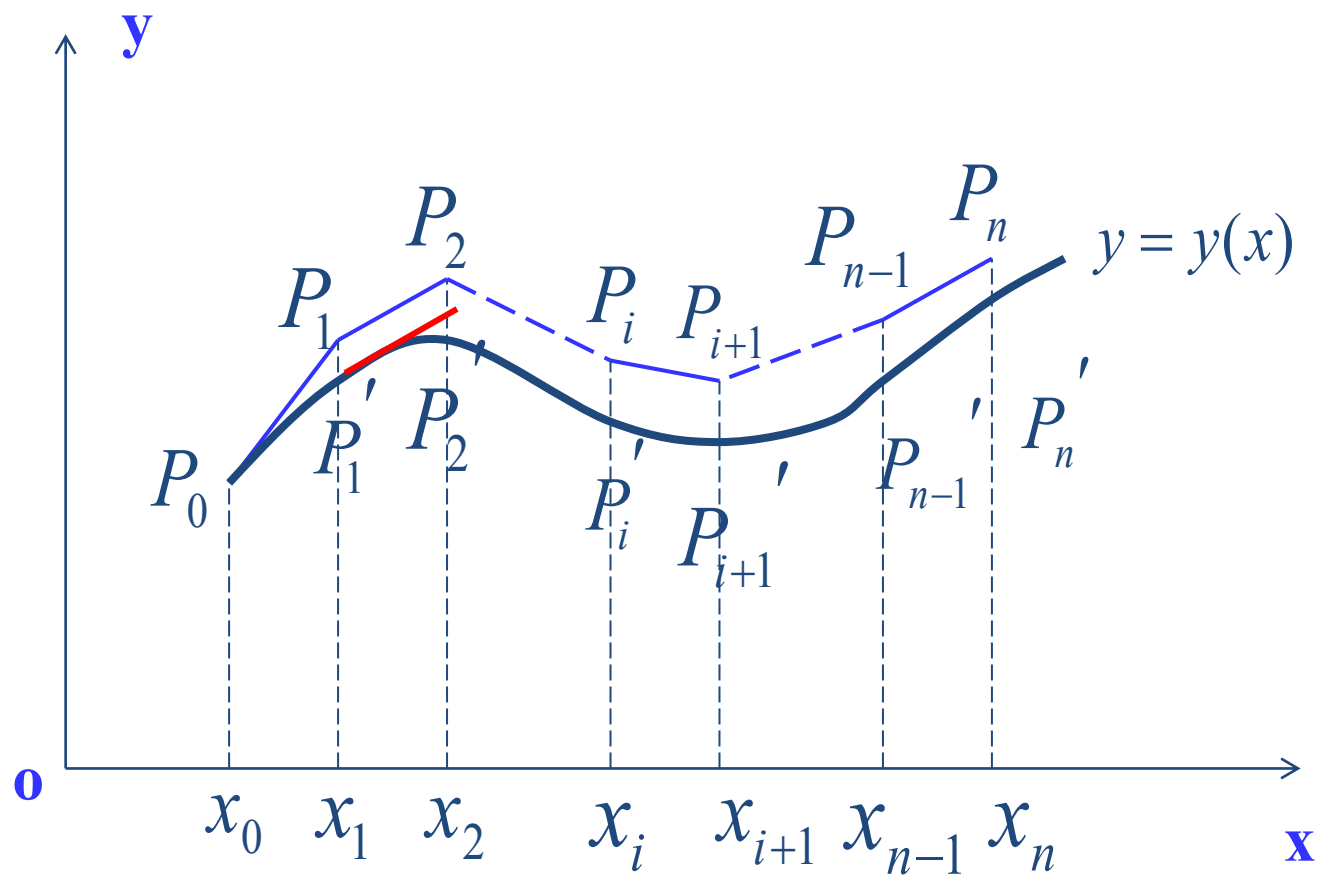


图3-2

与



得到  $y_2$  作为  $y(x_2)$  的近值; .....如此继续, 直到  $P_n$  点。这样, 得出一条折线  $P_0 P_1 P_2 \dots P_i \dots P_n$  近似代替积分曲线  $P'_0 P'_1 P'_2 \dots P'_i \dots P'_n$ 。当步数越多时, 由于误差的积累, 折线  $P_0 P_1 P_2 \dots P_i \dots P_n$  可能会越来越偏离真解  $P'_0 P'_1 P'_2 \dots P'_i \dots P'_n$  曲线。

显然，欧拉格式具有递推性，在计算 $y_{i+1}$ 时只要用到前一步所得结果 $y_i$ 一个信息就够了，因此是一种**单步格式**或称**一步格式**。也可以用其他数值微分公式导出略异于**欧拉公式**的其他形式算式。例如，用向后差商表示两点数值微分公式

$$y'(x_{i+1}) = \frac{y(x_{i+1}) - y(x_i)}{h}$$
$$(i = 0, 1, \dots, n-1)$$

代入

$$y'(x_{i+1}) = f(x_{i+1}, y(x_{i+1})), (i = 0, 1, 2, \dots, n-1)$$

可得

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + hf(x_{i+1}, y(x_{i+1})) \\ &\approx y(x_i) + hf(x_{i+1}, y(x_i)) \\ (i &= 0, 1, 2, \dots, n-1) \end{aligned}$$

$y(x_{i+1})$ 和 $y(x_i)$ 用其近似值 $y_{i+1}$ 和 $y_i$ 代入, 则得

$$\begin{aligned} y_{i+1} &= y_i + hf(x_{i+1}, y_{i+1}) \\ (i &= 0, 1, 2, \dots, n-1) \end{aligned} \quad (6)$$

其局部截断误差为：

$$\begin{aligned} R_i &= -\frac{h^2}{2} y''(\xi_i) \quad (x_{i+1} < \xi_i < x_i) \\ &= O(h^2) \quad (i = 0, 1, 2, \dots, n-1) \end{aligned} \quad (7)$$

(6)称为**向后欧拉格式**，又称为**隐式欧拉格式**。因为在此式的右边也包含未知的 $y_{i+1}$ ，所以是 $y_{i+1}$ 的一个函数方程，故称它为**隐式格式**。(2)的右边则没有未知的 $y_{i+1}$ ，因此是一种**显式格式**。隐式格式的计算当然比显式格式要困难得多，一般情况下，只能用迭代法求解，计算量比较大。

再如，用中心差商表示的三点数值微分公式

$$y'(x_i) = \frac{1}{2h} [y(x_{i+1}) - y(x_{i-1})] \\ (i = 0, 1, 2, \dots, n-1)$$

代入

$$y'(x_i) = f(x_i, y(x_i)) \quad (i = 0, 1, 2, \dots, n)$$

可得

$$y(x_{i+1}) = y(x_{i-1}) + 2hf(x_i, y(x_i)) \\ \approx y(x_{i-1}) + 2hf(x_i, y(x_i)) \\ (i = 0, 1, 2, \dots, n-1)$$

$y(x_{i+1})$ ,  $y(x_i)$ 和 $y(x_{i-1})$ 分别用其近似值代入, 则得

$$\begin{aligned} y_{i+1} &= y_{i-1} + 2hf(x_i, y_i) \\ (i &= 0, 1, 2, \dots, n-1) \end{aligned} \quad (8)$$

显然, 其局部截断误差为

$$\begin{aligned} R_i &= \frac{h^3}{3} y^{(3)}(\xi_i) \quad (i = 0, 1, 2, \dots, n-1) \\ &= O(h^3) \end{aligned} \quad (9)$$

我们也可以用泰勒展开来获得两步欧拉公式的局部截断误差。

1.首先写出  $y = y(x_{n+1})$  在  $(x_n, y_n)$  点处的泰勒展开式为:

$$y(x_{n+1}) = y(x_n) + 2hy'(x_n) + \frac{(2h)^2}{2!} y''(x_n) + \frac{(2h)^3}{3!} y'''(\xi_1) \\ (x_i < \xi_1 < x_{i+1})(i = 0, 1, 2, \dots, n-1)$$

2.写出  $y' = y'(x_{n+1})$  在  $(x_n, y_n)$  点处的泰勒展开式为:

$$y'(x_{n+1}) = y'(x_n) + hy''(x_n) + \frac{h^2}{2!} y'''(\xi_2) \\ (x_i < \xi_2 < x_{i+1})(i = 0, 1, 2, \dots, n-1)$$

3.写出  $y'' = y''(x_{n+1})$  在  $(x_n, y_n)$  点处的泰勒展开式为:

$$y''(x_{n+1}) = y''(x_n) + hy'''(\xi_3) \\ (x_i < \xi_3 < x_{i+1})(i = 0, 1, 2, \dots, n-1)$$

将3式和2式, 代入1式, 整理可得书上99页倒数第二行的结果

不论是显式欧拉格式 (2)，还是隐式欧拉格式 (6)，它们都是**单步格式**或称为**一步格式**。因为它们在计算 $y_{i+1}$ 时只用到前一步所得结果 $y_i$ 一个信息；而格式 (8) 则除了 $y_i$ 外，还需用到更前一步所得信息 $y_{i-1}$ ，即需调用前两步的信息，因此 (8) 称为**两步欧拉格式**，或称为**中点欧拉格式**。

比较 (3)，(7) 和 (9) 可知，两步欧拉格式比显式或隐式欧拉格式具有更高的精度，因为它的局部截断误差是 $O(h^3)$ 。

由 (3) 和 (7) 可见，显式欧拉格式与隐式欧拉格式的局部截断误差的符号正好相反，因此可以设想取 (2) 和 (6) 的平均，即两式相加除以2，得



$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \quad (10)$$

$$(i = 0, 1, 2, \dots, n-1)$$

使误差相互抵消，提高误差阶数，从而提高算法的精度。事实上，格式(10)的局部截断误差为 $O(h^3)$ ，即其精度为二阶。(10)称为**梯形格式**，是一种隐式格式。从几何角度上来看，梯形格式是取 $[x_i, x_{i+1}]$ 的两端点的平均斜率。从图3-3可见，用梯形格式得到的 $P$ 点比用显式欧拉格式得到的 $A$ 点和用隐式欧拉格式得到的 $B$ 点都要合理。

欧拉格式(2)的程序图见图3-4

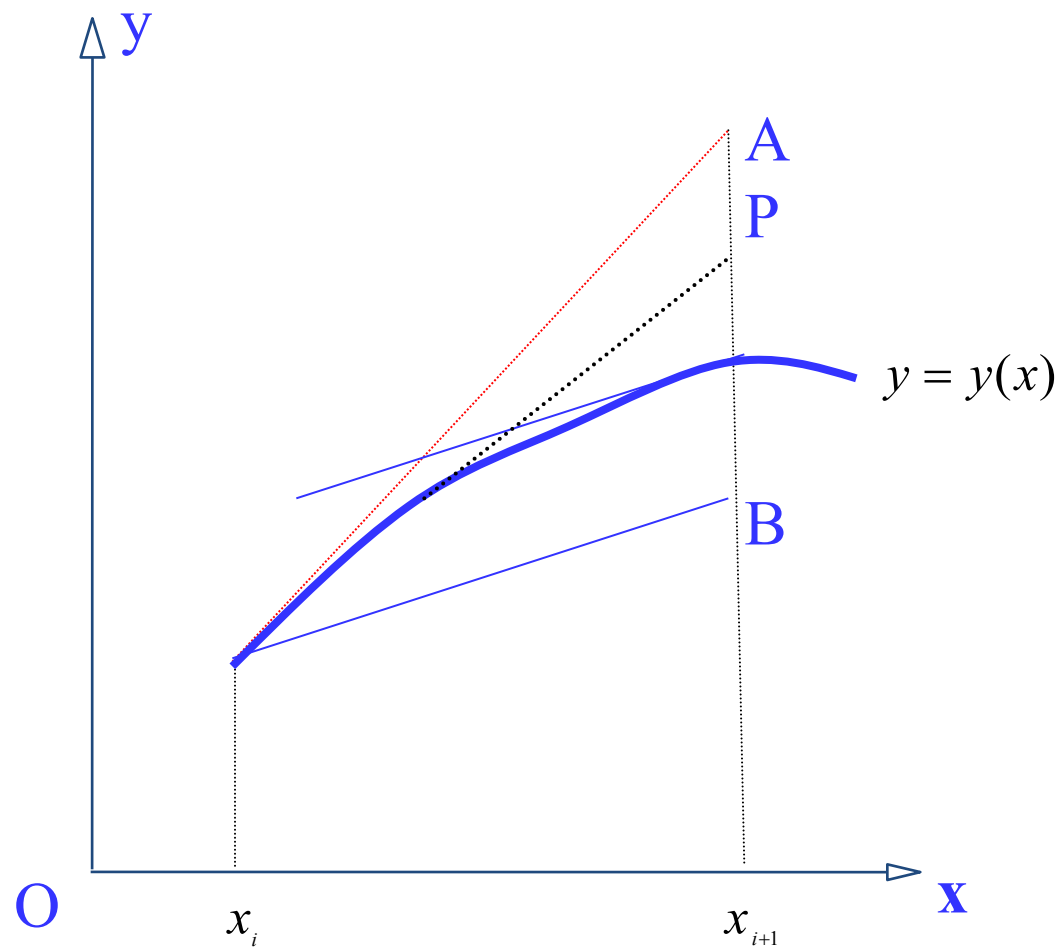


图3-3

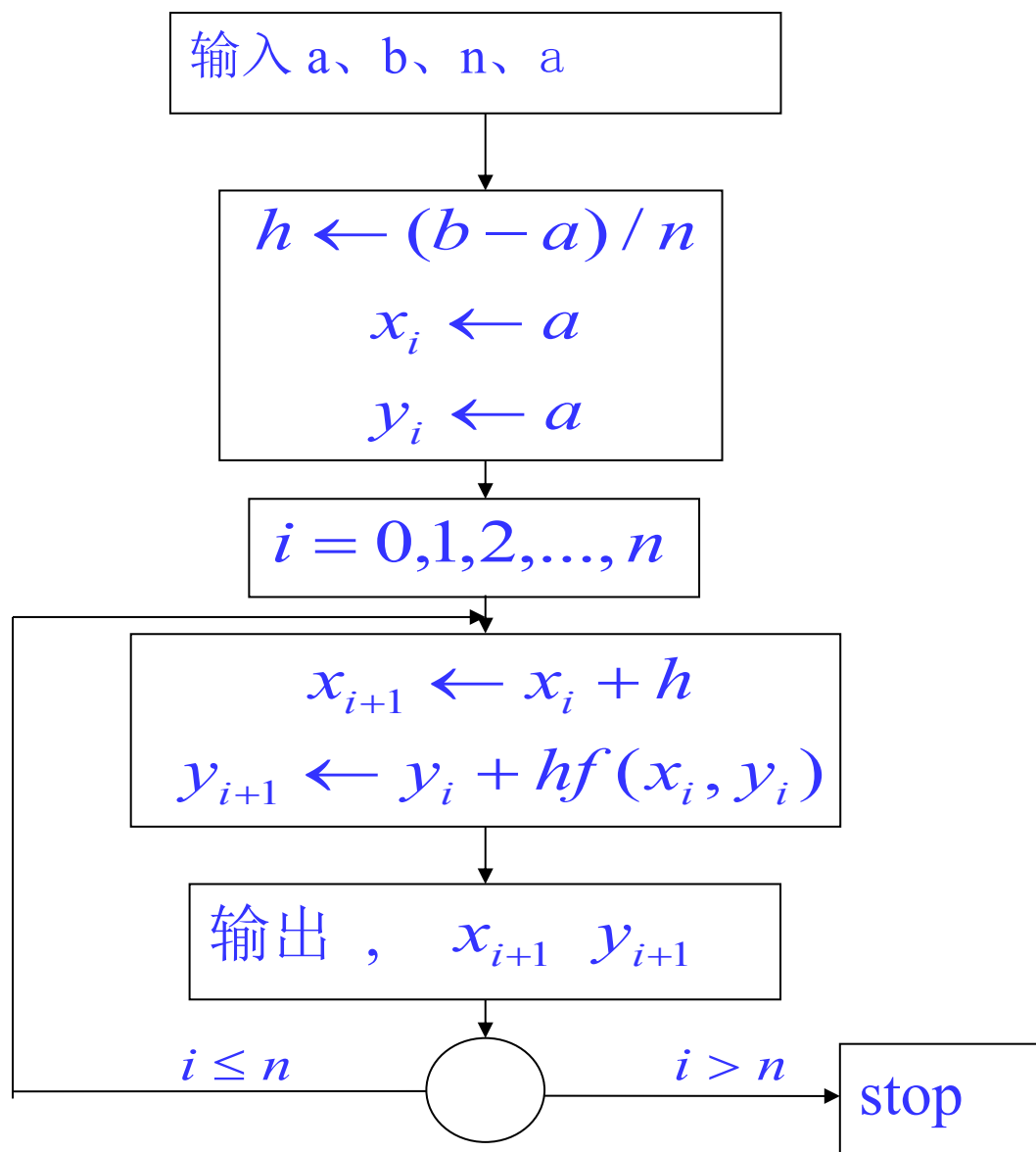


图3-4

## § 2 改进的欧拉方法

显式欧拉格式计算工作量比较小，但精度低。梯形格式虽提高了精度，但为隐式格式，需用迭代法求解，计算工作量大。综合这两种格式可得到改进的欧拉格式。

先用欧拉格式 [\(2\)](#) 求出一个初步的近似值  $\overline{y_{i+1}}$ ，称为预测值，它的精度不高，接着用梯形格式 [\(10\)](#) 对它校正一次，即迭代一次，求得  $y_{i+1}$ ，称为校正值。这种预测-校正方法称为改进的欧拉格式：

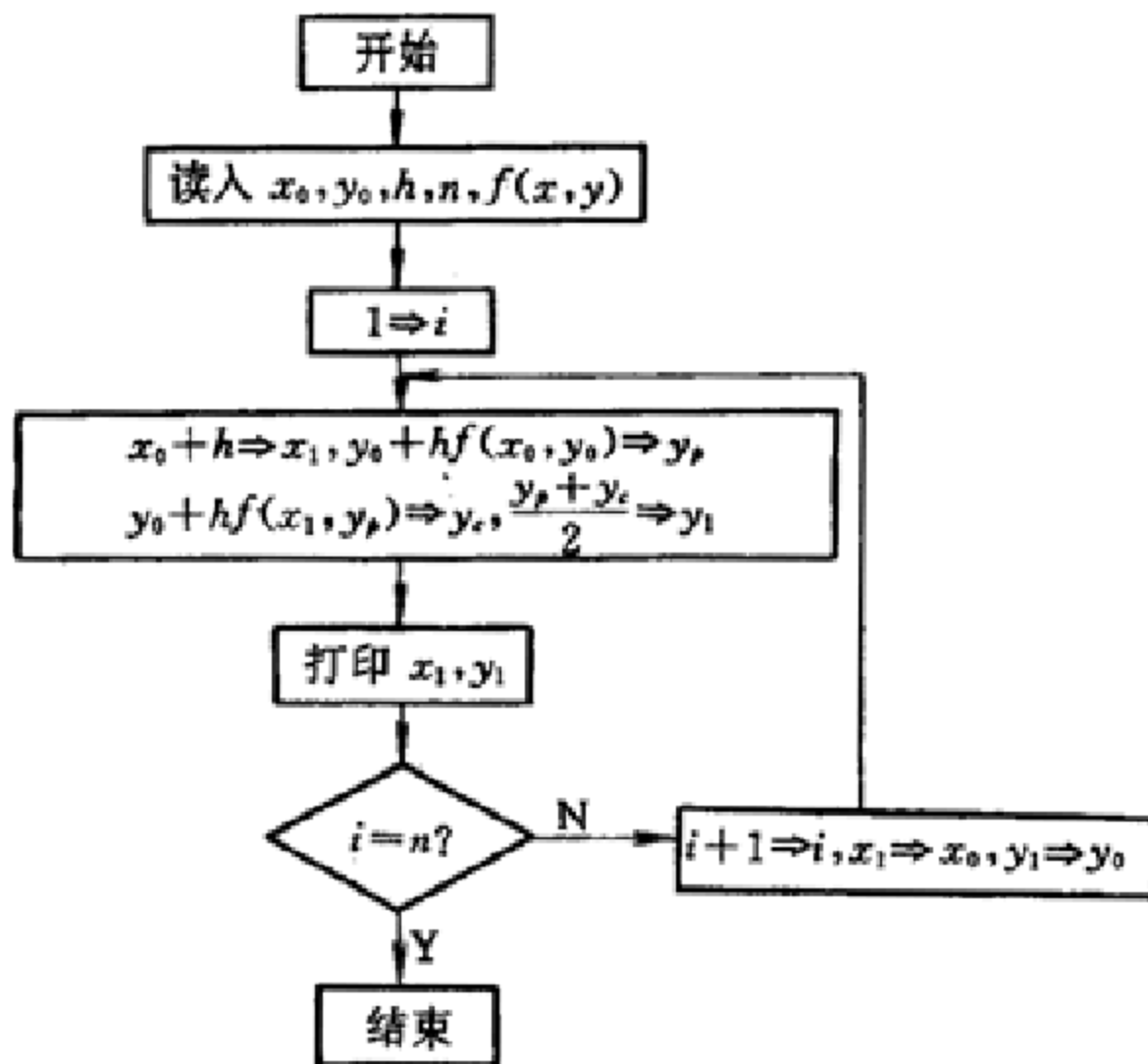
$$\begin{array}{l} \text{预测} \longrightarrow \\ \text{校正} \longrightarrow \end{array} \left\{ \begin{array}{l} \overline{y_{i+1}} = y_i + hf(x_i, y_i) \\ y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, \overline{y_{i+1}})] \\ (i = 0, 1, 2, \dots, n-1) \end{array} \right. \quad (11)$$

可以证明，(11) 格式的精度为二阶。它是一种一步显式格式。(11) 也可写成

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))] \\ (i = 0, 1, 2, \dots, n-1) \quad (12)$$

或将它写成下列平均化的形式：

$$\begin{cases} y_p = y_i + hf(x_i, y_i) \\ y_c = y_i + hf(x_{i+1}, y_p) \\ y_{i+1} = \frac{1}{2}(y_p + y_c) \\ (i = 0, 1, 2, \dots, n-1) \end{cases} \quad (13)$$



改进的欧拉格式 (13) 的程序框见上图。

**例 1** 试分别用欧拉格式和改进的欧拉格式求解下列初值问题,并比较两法所得结果的精度:

$$\begin{cases} \frac{dy}{dx} = -y, x \in [0,1] \\ y(0) = 1 \end{cases}$$

**解** 取步长 $h=0.1$ . 这样, 欧拉格式和改进的欧拉格式的具体算式分别为

$$y_{i+1} = y_i + 0.1(-y_i) = 0.9y_i$$

及



$$\begin{cases} y_p = y_i + 0.1(-y_i) = 0.9y_i \\ y_c = y_i + 0.1(-y_p) = 0.91y_i \\ y_{i+1} = \frac{1}{2}(y_p + y_c) = 0.905y_i \end{cases}$$

两式格式的计算结果分别列于[表3-1](#)中。由此表可见，与精确解  $y = e^{-x}$  相比，改进的欧拉格式的精度较欧拉格式有明显的提高。



表3-1

$x_i$	欧拉格式		改进的欧拉格式		精确解
	$y_i$	$ y(x_i) - y_i $	$y_i$	$ y(x_i) - y_i $	$y(x_i)$
0	1		1		1
0.1	0.9000000	$4.8374 \times 10^{-3}$	0.9050000	$1.626 \times 10^{-4}$	0.9048374
0.2	0.8100000	$8.7308 \times 10^{-3}$	0.8190250	$2.942 \times 10^{-4}$	0.8187308
0.3	0.7290000	$1.18182 \times 10^{-2}$	0.7412176	$3.994 \times 10^{-4}$	0.7408182
0.4	0.6561000	$1.4220 \times 10^{-2}$	0.6708020	$4.820 \times 10^{-4}$	0.6703200
0.5	0.5904900	$1.60407 \times 10^{-2}$	0.6070758	$5.451 \times 10^{-4}$	0.6065307
0.6	0.5314410	$1.73706 \times 10^{-2}$	0.5494036	$5.920 \times 10^{-4}$	0.5488116
0.7	0.4782969	$1.82884 \times 10^{-2}$	0.4972102	$6.249 \times 10^{-4}$	0.4965853
0.8	0.4304672	$1.88618 \times 10^{-2}$	0.4499753	$6.463 \times 10^{-4}$	0.4493290
0.9	0.3874205	$1.91492 \times 10^{-2}$	0.4072276	$6.579 \times 10^{-4}$	0.4065697
1	0.3486784	$1.92010 \times 10^{-2}$	0.3685410	$6.616 \times 10^{-4}$	0.3678794

