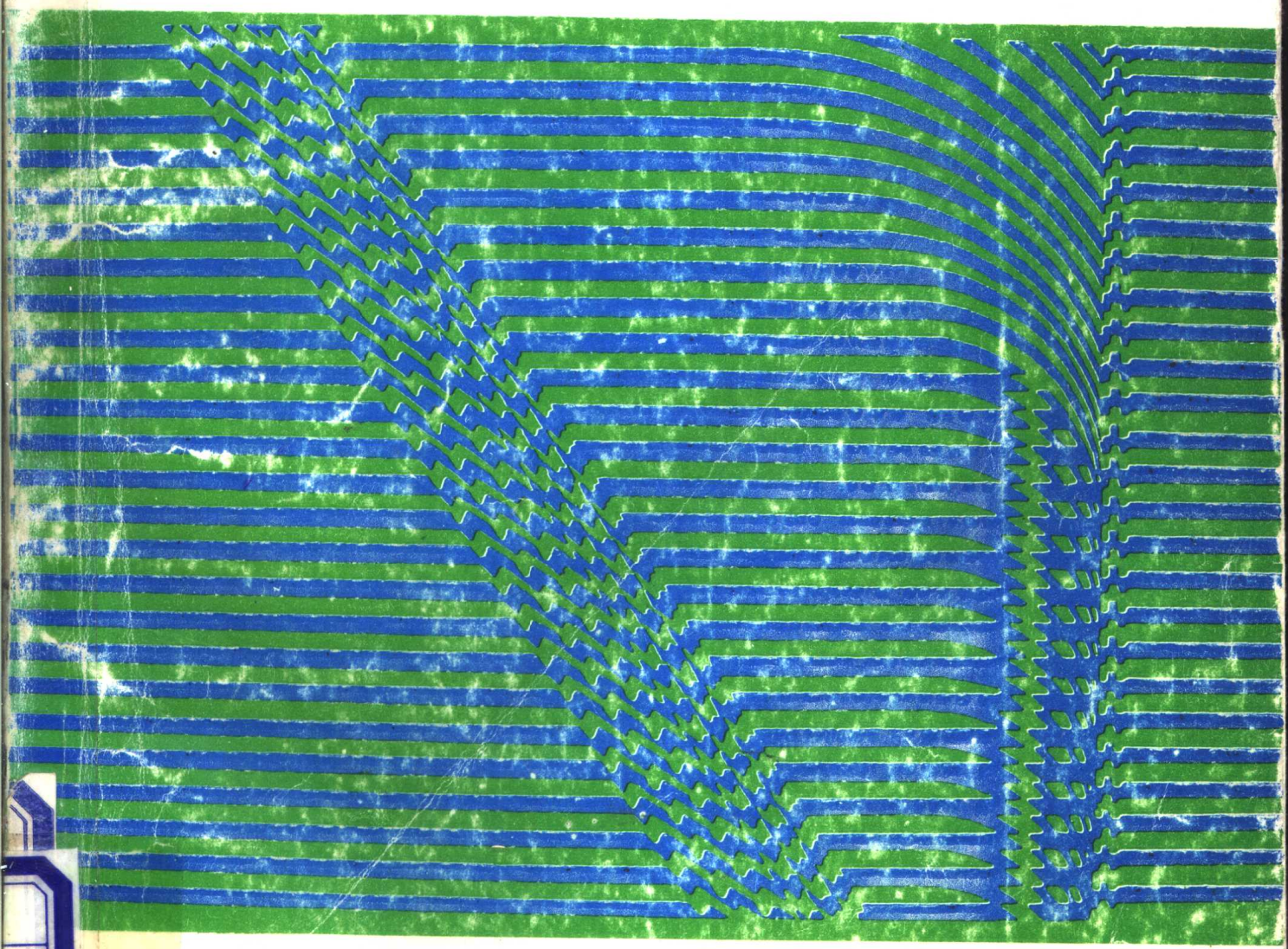
# Z80汇编语言
# 《实用子程序集》

于长丰　马炳魁　编著



陕　西　电　子　编　辑　部

# Z80 汇编语言
# 《实用子程序集》

于长丰　　马炳魁　编著

陕西电子编辑部

# 前　言

近年来，随着计算机应用的不断深入和普及，Z80类微型计算机已被广泛地应用于教学、工业控制、智能仪器仪表、一般事务管理等领域中。在我国，拥有Z80微型计算机的用户很多，如何更好地使用这些微型计算机，使它们发挥更大的效用是一件具有现实意义的任务。出于这个目的，我们编写了这本书。

本书第一章，介绍了常用数学函数的几种快速算法。重点讨论在给定精度下，如何利用这些算法来提高函数的运算速度和减小数表存贮空间等问题。本章给出的某些快速算法是首次提出，比常用算法有显著的优越性，特别适用于实时系统中的数据处理。

本书的第二章到第九章，给出了一般常用子程序，包括有代码转换、算术运算、基本函数、检索排序、数据采集等。对于Z80汇编语言初学者，学习和掌握这部分程序的设计方法及编程技巧尤为重要。

本书第十、十一、十二、十五章，给出了工程上常用的几个算法子程序，包括有解线性方程组、数据拟合、快速傅氏变换等。这些程序在一般科学计算、数据处理和工程设计中是经常用到的。在汇编语言级编制这些程序不是一件容易的事，这可使用户节省不少时间和精力。

本书第十四章为DDC系统应用子程序，包括算术运算、数字滤波、PID控制算法等。这些程序对于解决过程控制中的某些实际问题，有较强的实用价值。

在某些中大规模科学计算和数据处理任务中，经常用到双精度基本函数的运算，对用户来说，编制这些程序不但费时而且困难较大，因此，本书第十六章给出了十一个双精度基本函数子程序，以供用户参考使用。

本书的第三、五、十四、十六章是独立的程序模块，用户可直接移植使用。其它章节，如第二、十、十一、十二、十五章只要把所调用的子程序的地址重新安排，也可成为独立的程序模块。

本书对于从事计算机应用、软件开发的科技人员，特别是对使用Z80微型计算机的科技人员有一定的实用与参考价值，也可作为中高等院校有关专业的教材参考书。

书中全部程序都在DJS—O 4IC单板计算机上调试通过，内容正确可靠。

本书的出版工作得到了高级工程师张忠智同志，航空工业部远东机械制造公司孙洁等同志的帮助和支持，在此谨致衷心感谢。

由于作者水平有限，书中定有不少缺点和错误，敬请读者批评指正。

作者

1988　7

# 目　　录

# 第一章　函数及快速算法

## §1.1　引　言

基本函数运算在过程控制、数据处理等方面往往是不可缺少的。本章将着重介绍一些常用算术函数的算法及程序设计方法。

编制基本函数程序常用的方法是幂级数展开法。这是一种递推性算法，由于需多次进行乘加运算，从而影响了程序的执行速度，这在某些时间性要求很强的任务中是不够理想的。针对这一问题，本章给出了几种新的算法（下称快速算法），使完成一次函数操作仅需1至3次乘除运算，其计算量比幂级数展开法约缩小10倍。

快速算法的基本思想是，采用计算与查表相结合的技术来提高运算速度。一般说来，精度要求越高，则数表占内存就越大，但如果算法选择的适当，这个问题是可以得到较好解决的。就基本函数程序而言，主要有三方面指标需要考虑，那就是程序的运算精度、计算量（用乘加次数来衡量）及数表存贮空间。要兼顾这三方面指标，就得研究一套有效的算法。例如，求指数函数$2^x$（$0\leqslant x<1$）的值，要使精度达到$10^{-7}$，采用线性插值法编程，则计算量为1次乘法，数表（一维）内占存1539字节；若采用HPLI法编程，则计算量为2次乘法，数表（二维）占内存192字节；而若采用多重插值法，如4重插值，则计算量为5次乘法，数表（5维）占内存60字节。从本例可以看出，若提高某一指标的质量，那么相应的另一指标的质量就会下降；要使各指标的质量得到兼顾，则HPLI法较为理想。当然，根据实际需要可以强调某一方面的指标，比如，特别要求运算速度时，可采用线性插值法编程。

## §1.2　对数、指数、幂函数快速算法分析

### 一、算法概要

对于任意实数x，若$x\neq 0$，则可表示成$\pm 2^q\cdot 2^\alpha$的形式，其中q为零或整数，$\alpha$为一非负小数或零，正负号取决于$x$的符号。不失一般性，设$x>0$，则有$x=2^q\cdot 2^\alpha$，于是得到对数、指数、幂函数及平方根函数的计算公式：

$$\ln x=\ln 2^q\cdot 2^\alpha=(q+\alpha)\ln 2 \qquad (\alpha=\log_2\frac{x}{2^q}) \qquad (1-1)$$

$$e^X = 2^{x/\ln 2} = 2^{[x/\ln 2]} \cdot 2^{<x/\ln 2>} \qquad (1-2)$$

$$x^T = \left(2^q \cdot 2^\alpha\right)^T = 2^{[(q+\alpha)T]} \cdot 2^{<(q+\alpha)T>} \qquad (1-3)$$

$$\sqrt{x} = \sqrt[r]{2^{q+\alpha}} = 2^{[(q+\alpha)/2]} \cdot 2^{<(q+\alpha)/2>} \qquad (1-4)$$

其中〔 〕表示整数部分，〈 〉表示一个非负小数。式中幂指数是〔 〕的部分是不需要计算的，这样从上面计算公式中可得到两个有代表性的函数

$$y = 2^x \qquad x \in [0, 1) \qquad (1-5)$$

$$y = \log_2 x \qquad x \in [1, 2) \qquad (1-6)$$

这是一对反函数，我们只要编制出求解$2^x$和$\log_2 x$的程序来，那么对数、指数、幂函数及平方根函数通过简单的移位和加减运算便可求出结果。

## 二、线性插值法

在快速算法中，线性插值法是一种通用性较强，而且最简单和容易掌握的算法，其特点是计算量小，程序设计简单。

设函数f（x）是区间〔a，b〕上的单调连续函数，且a，b均为整数（这样假设是为了编程的方便）。把区间〔a，b〕n等份，并使$n=(b-a)2^m$，m为零或整数，则节点$x_i = a + \Delta x \cdot i$，$i = 0, 1, \cdots, n$。对于每个小区间$[x_i, x_{i+1}]$函数f（x）用直线来近似代替，就是

$$f(x) \approx \varphi(x) = \frac{f_{i+1} - f_i}{\Delta x}(x - x_i) + f_i$$

$$= 2^m (f_{i+1} - f_i)(x - x_i) + f_i \qquad (2-1)$$

其中$f_i$为节点函数值，$(f_0, f_1, \cdots f_n)$已作为常数存入单元。这样x值给定后，可用相对应的直线方程来求函数f（x）的近似值。设FDRS为数表首址，B为一个数据所占内存的字节数，则$f_i$的存放地址为

$$FDR_i = FDRS + B \cdot i \qquad (2-2)$$

其中$i = (x_i - a)2^m$。一般说来，B只取几个字节，如B＝4，所以通过移位或相加的办法可求出B·i值，不必采用乘运算。用（2－1）式求f（x）的近似值，其计算量为一次乘法。下面我们来讨论直线方程与被插值函数 f（x）的误差。

设函数f（x）在区间〔a，b〕上各阶导数存在，不失一般性，我们可讨论第i段直线方函与f（x）的误差。对于任意的$x = x_i + h$，$h \in [0, \Delta x)$，令

$$R_i(x) = f(x) - \varphi(x) \qquad (2-3)$$

式中$\varphi(x) = \frac{h}{\Delta x}(f_{i+1} - f_i) + f_i$，将f（x），$f_{i+1}$在$x_i$处展成2阶泰勒级数

$$f(x) = f(x_i) + f'(x_i)\, h + \frac{f''(\xi)h^2}{2!} \qquad \xi \in (x_i, x)$$

$$f_{i+1} = f(x_i) + f'(x_i)\Delta x + \frac{f''(\eta)(\Delta x)^2}{2!} \qquad \eta \in (x_i, x_{i+1})$$

代入（2—3）整理得

$$R_1(x)=\frac{h}{2!}(hf''(\xi)-\Delta xf''(\eta)) \qquad (2-4)$$

令$\eta=\xi+\delta$，$\delta\in(0,\eta-\xi)$ 将$f''(\eta)$在$\xi$点处展成泰勒数级代入（2—4）整理得

$$R_1(x)=\frac{h}{2!}\left(hf''(\xi)-\Delta xf''(\xi)-\Delta x\sum_{k=3}^{\infty}\frac{f^{(k)}(\xi)\delta^{k-2}}{(k-2)!}\right) \qquad (2-5)$$

由于$\Delta xh\delta$相对于$\Delta xh$为高阶无穷小，故

$$R_1(x)\approx\frac{h}{2!}f''(\xi)(h-\Delta x) \qquad (2-6)$$

由上式看出当$h=0$，$\Delta x$，即$x=x_1, x_{1+1}$时，$R(x)=0$。可进一步证明

$$|R_1(x)|\leqslant\frac{(\Delta x)^2}{8}max|f''(x)| \quad x\in[x_1, x_{1+1}) \qquad (2-7)$$

现结合指数函数$y=2^x$，$x\in[0,1)$，讨论一下n值与计算精度的关系。在区间[0，1）上我们用最大平均相对误差来估计精度，即

$$\varepsilon_{max}=\frac{1}{n}\sum_{i=0}^{n-1}\varepsilon_{imax}=\frac{1}{n}\sum_{i=0}^{n-1}\left|\frac{(\Delta x)^2 maxf''(x)}{8f(x_1)}\right| \quad x\in[x_1, x_{1+1}) \qquad (2-8)$$

式中$maxf''(x)|x=x_{1+1}=2^{x_{i+1}}(\ln 2)^2$，$f(x_1)=2^{x_1}$。将其代入（2—8）式整理得：

$$\varepsilon_{max}=\frac{(\Delta x)^2(ln2)^2\cdot 2^{\Delta x}}{8} \qquad (2-9)$$

设V为数表所占空间，则$V=(n+1)B$，下面给出$V-\varepsilon_{max}$的计算表

表1.1

| B | 2 | | | 3 | | |
|---|---|---|---|---|---|---|
| n | $2^4$ | $2^5$ | $2^6$ | $2^7$ | $2^8$ | $2^9$ |
| $\varepsilon_{amx}$ | $2.45\times10^{-4}$ | $5.99\times10^{-5}$ | $1.48\times12^{-5}$ | $3.69\times10^{-6}$ | $9.18\times10^{-7}$ | $2.29\times10^{-7}$ |
| V | 34 | 66 | 130 | 387 | 771 | 1539 |

## 三、高精度线性插值法（HPLI）

由表1.1可看出，线性插值法虽然计算量小，但精度要求较高时，数表的存贮空间是很可观的，为解决这一问题，下面给出高精度线性插值法，以后简称HPLI（High Precision Linear Interpolation）法。

我们仍以函数$2^x$，$x\in[0,1)$为例来分析。把区间[0，1）n等份，并使$n=2^m$

（m为正整数），则有

$$2^x = 2^{x_i} \cdot 2^h = (1 + E_i)\, 2^h \qquad (3-1)$$

式中$x = x_i + h$，$h \in [0, \Delta x)$，$E_i = 2^{x_i} - 1$，$i = 0, 1 \cdots, n-1$对于$2^h$按线性插值法来处理，把区间$[0, \Delta x)$ $l$等份，并使$l = 2^k$（k为正整数），则有

$$2^h = \varphi(h) = \frac{2^{h_{j+1}} - 2^{h_j}}{\Delta h}(h - h_j) + 2^{h_j} \qquad (3-2)$$

把（3—2）式代入（3—1）式得

$$2^x = (1 + E_i)\left[\left(\frac{2^{h_{j+1}} - 2^{h_j}}{\Delta h}\right)(h - h_j) + 2^{h_j}\right] \qquad (3-3)$$

式中$\Delta h = \frac{\Delta x}{l} = \frac{1}{2^{k+m}}$，$h_j = \Delta h \cdot j$，$j = 0, 1, \cdots, l-1$，(3—3)式即计算$2^x$的算式。

对于$E_i$，$2^{h_j}$组成数表按

$(E_0, E_1 \cdots, E_{n-1})$

$(2^{h_0}, 2^{h_1}, \cdots, 2^{h_{(l-1)}})$

的形式顺序存入单元，共有$(2^m + 2^k)$个数据。

现在估计一下计算精度，根据（2—3），（2—6）式得绝对误差

$$R_{ij}(x) = 2^{x_i}\frac{\beta}{2l}\, 2^{\xi}(\ln 2)^2 (\beta - \Delta h) \qquad (3-4)$$

其中$\xi \in (h_j, h)$，$\beta \in [0, \Delta h)$，根据（2—7）式又可知

$$|R_{ij}(x)| \leqslant 2^{x_i}\frac{(\Delta h)^2}{8}\max 2^{\eta}(\ln 2)^2 \quad \eta \in [h_j, h_{j+1})$$

$$= 2^{x_i} \cdot 2^{h_{j+1}}\frac{(\Delta h)^2}{8}(\ln 2)^2 \qquad (3-5)$$

在区间$[0, 1)$上函数$2^x$的最大平均相对误差为

$$\bar{\varepsilon}_{max} = \frac{1}{n}\sum_{i=0}^{n-1}\bar{\varepsilon}_{imax}$$

$$= \frac{1}{n}\sum_{i=0}^{n-1}\left(\frac{1}{l}\sum_{j=0}^{l-1}\varepsilon_{ijmax}\right)$$

$$= \frac{1}{n}\sum_{i=0}^{n-1}\left(\frac{1}{l}\sum_{j=0}^{l-1}\frac{|R_{ijmax}|}{2^{x_i} \cdot 2^{h_j}}\right)$$

$$= \frac{(\Delta h)^2}{8}(\ln 2)^2 2^{\Delta h} \qquad (3-6)$$

HPLI算法的数表所占存储空间

$$V=(2^m+2^k)\ B \qquad (3-7)$$

由$\Delta h=\frac{1}{2^{m+k}}$得$2^k=\frac{1}{\Delta h 2^m}$代入（3—7）式得

$$V=(2^m+\frac{i}{\Delta h 2^m})\cdot B \qquad (3-8)$$

当精度给定后，$\Delta h$为一常数，由（3—8）式容易证明当$m=K$时V取得极小值，故我们取

$$V_{min}=2^{m+1}\cdot B=2^{k+1}\cdot B \qquad (3-9)$$

下面给出$v_{min}-\varepsilon_{max}$表

表1.2

| B | 2 | | 3 | | | 4 | | |
|---|---|---|---|---|---|---|---|---|
| m（或K） | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $\bar{\varepsilon}_{max}$ | $2.45\times10^{-4}$ | $1.48\times10^{-5}$ | $9.18\times10^{-7}$ | $5.73\times10^{-8}$ | $3.58\times10^{-9}$ | $2.24\times10^{-10}$ | $1.40\times10^{-11}$ | $8.74\times10^{-12}$ |
| $V_{min}$ | 16 | 32 | 96 | 192 | 384 | 1024 | 2048 | 4096 |

比较表1.1和1.2可看出，在精度要求相近的情况下，数表空间相差很可观。另外，用（3—3）式计算$2^x$其计算量为2次乘法运算，所以HPLI法是一种较理想的算法。

**四、有限增量微分法**

前面介绍了指数$2^x$，$x\in[0,1)$的计算方法，下面我们讨论对数函数$\log_2 x$，$x\in[1,2)$的算法。

函数$f(x)=\log_2 x$采用有限增量微法进行计算，即

$$f(x)=f(x_1)+f'(x_1+\theta h)h \qquad (4-1)$$

其中$x=x_1+h$，$h\in[0,\Delta x)$，$\theta\in(0,1)$。上式是该函数$f(x)$的精确表达式，而由于$\theta$不能确定，所以不可能直接利用该式进行计算。但我们知道$f(x_1)$为$f(x)$的线性主部，$f'(x_1+\theta h)h$只是函数值的增量，又因为在实际计算中h很小，所以可以取$h\to0$时的$\theta$值，即$\theta_0=\lim\limits_{h\to0}\theta(x_1,h)$。为了讨论$\theta_0$的值，首先给出下面结论：

设$f(x)\in C^n_{[a,b]}$（$n=0,1,2\cdots$），且$f''(x_1)\neq0$，则$\theta_0=\lim\limits_{h\to0}\theta(x_1,h)=\frac{1}{2}$

证明：将$f(x)$在$x_1$处展成泰勒级数，代入（4—1）式整理得

$$f'(x_1+\theta h)=\sum_{n=1}^{\infty}\frac{f^{(n)}(x_1)}{n!}h^{n-1}$$

上式两边对h积分得

$$\frac{1}{\theta}(f(x_1+\theta h)-f(x_1))=\sum_{n=1}^{\infty}\frac{f^{(n)}(x_1)}{n!\ n}h^n$$

再把$f(x_1+\theta h)$展成泰勒级数代入上式整理得

$$\sum_{n=2}^{\infty}\frac{f^{(n)}(x_1)h^n\theta^{n-1}}{n!}=\sum_{n=2}^{\infty}\frac{f^{(n)}(x_1)h^n}{n!\ n}$$

上式又可写成

$$\frac{f''(x_1)h^2\theta}{2!}+\sum_{n=3}^{\infty}\frac{f^{(n)}(x_1)h^n}{n!}\theta^{n-1}=\frac{f''(x_1)h^2}{2!\ 2}+\sum_{n=3}^{\infty}\frac{f^{(n)}(x_1)}{n!\ n}h^n \quad (4-2)$$

由于$f(x)\in C^n_{[a,b]}$，所以存在一个足够大的常数M，使得$|f^{(n)}(x_1)|\leqslant M$。容易证明（4—2）式中，当$h\to 0$时等式两边的无穷级数趋近于0，这样，从（4—2）式中可知，当$f''(x)\neq 0$时，$\theta_0=\lim_{h\to 0}\theta(x_1,h)=\frac{1}{2}$。证毕，于是函数可用下式进行计算

$$f(x)\approx\varphi(x)=f(x_1)+f'(x_1+h/2)\,h \quad (4-3)$$

下面直接给出上式绝对误差和精度的表达式

$$R_1(x)=\frac{h^3}{2}\left(\frac{f^{(3)}(\xi)}{3}-\frac{f^{(3)}(\xi)}{4}-\frac{f^{(4)}(\tau)\delta}{4}\right)$$

$$\approx\frac{h^3}{24}f^{(3)}(\xi) \quad (4-4)$$

$$|R_{1max}|=\frac{(\Delta x)^3}{24}\max|f^{(3)}(x)| \qquad x\in[x_1,x_{1+1}] \quad (4-5)$$

$$\varepsilon_{max}=\frac{1}{n}\sum_{i=0}^{n-1}\varepsilon_{imax}$$

$$=\frac{1}{n}\sum_{i=0}^{n-1}\frac{|R_{imax}|}{|f(x)|}$$

$$=\frac{(\Delta x)^3}{24n}\sum_{i=0}^{n-1}\cdot\frac{|f^{(3)}{}_{max}(x)|}{|f(x)|} \quad (4-6)$$

其中$\xi\in(x_i, x)$，$\eta\in(\xi, \xi+\delta)$（不妨设$\delta>0$），$x\in[x_i, x_{i+1}]$，$f(x)$为$[x_i, x_{i+1}]$上的平均变化率。

以下讨论$\log_2 x$ $x\in[1, 2)$的具体算法。

把区间$[1, 2)$n等份，并使$n=2^m$（m为正整数）。

令$x=x_j+p=\Delta xj+p$ $p\in[0, \Delta x)$，则有

$$\log_2 x=\log_2 x_j+\log_2\left(1+\frac{P}{x_i}\right)$$

$$=A_j+\log_2(1+B_jP) \qquad (4-7)$$

其中$A_j=\log_2 x_j$，$B_j=\frac{1}{x_j}$ $(j=0, 1, 2, \cdots, n-1)$它们组成数表

$$(A_0, A_1, \cdots, A_{n-1})$$

$$(B_0, B_1, \cdots, B_{n-1})$$

顺序存入单元中，共有$2^{m+1}$个数据。令$t=1+B_jP$代入（4—7）式得

$$\log_2 x=A_j+\log_2 t \qquad (4-8)$$

其中$t\in[1, 1+\Delta X)$，把区间$[1, 1+\Delta X)$$l$等份，并使$l=2^k$（K为正整数），则

$\Delta t=\frac{\Delta x}{l}=\frac{1}{2^{m+k}}$，$t=t_i+h=1+\Delta ti+h$，$h\in[0, \Delta t)$ $(i=0, 1, 2, \cdots, l-1)$

因此（4—8）式又可写成

$$\log_2 x=A_j+\log t_i+\log_2\left(1+\frac{h}{t_i}\right)=A_j+F_i+\log_2\left(1+\frac{h}{t_i}\right) \qquad (4-9)$$

式中$F_i$为小区间$[1, 1+\Delta X)$的节点函数值，共$2^k$个数据，它们组成数表

$$(F_0, F_1, \cdots, F_{l-1})$$

顺序存入单元中。在（4—9）式中$\log_2\left(1+\frac{h}{t_i}\right)$用有限增量微分法计算，并取$\theta_0=\frac{1}{2}$，则

$$\log_2 x\approx A_j+F_i+\log_2 1+\frac{h/t_i}{(1+h/2t_i)\ln 2}$$

$$=A_j+F_j+\frac{h}{(t_i+h/2)\ln 2} \qquad (4-10)$$

上式即函数$\log_2 x$的计算公式，其运算量为两次乘法（包括由$1+BjP$求t的一次乘法）一次除法。根据（4—5）式可以求出最大绝对误差为

$$|R_m(x)|=\frac{(\Delta t)^3}{12\ln 2} \quad x\in[1, 2) \qquad (4-11)$$

精度用下式进行估计

$$\varepsilon_{max}=\frac{|R_m(x)|}{\int_1^2|f(x)|dx}$$

$$= \frac{(\Delta t)^3}{12\ln 2}\left(2-\frac{1}{\ln 2}\right)$$

$$\approx 0.067(\Delta t)^3 \qquad (4-12)$$

式中 $\Delta t = \frac{1}{2^{m+k}}$

数表占内存空间为

$$V = (2^{m+1}+2^k)B \qquad (4-13)$$

当精度给定后 $\Delta t$ 为一常数，把 $2^k = \frac{1}{\Delta t 2^m}$ 代入（4—13），然后V对 m 求导数并令其等于零，解得当 $m=k-1$ 时V取得极小值，故有

$$V_{min} = 2^{m+2}B = 2^{k+1}B \qquad (4-14)$$

$V_{min} - \varepsilon_{max}$ 表如下：

表1.3

| B | 2 | | 3 | | 4 | | 5 | |
|---|---|---|---|---|---|---|---|---|
| m(=k-1) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $\varepsilon_{max}$ | $2.04\times10^{-6}$ | $3.19\times10^{-8}$ | $4.99\times10^{-10}$ | $7.80\times10^{-12}$ | $1.22\times10^{-13}$ | $1.90\times10^{-15}$ | $2.98\times10^{-17}$ | $4.65\times10^{-19}$ |
| V | 32 | 64 | 192 | 384 | 1024 | 2048 | 5120 | 10240 |

## 五、牛顿迭代法

牛顿迭代法的基本思想是将非线性方程 $f(x)=0$ 转化为线性方程的迭代计算来求其近似解。

将方程 $f(x)=0$ 用一阶泰勒级数展开：

$$f(x) \approx f(x_0) + f'(x_0)(x-x_0) = 0$$

设 $f'(x_0) \neq 0$，则由上式解得

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

我们取x作为原来方程的新的近似根 $x_1$，依次类推可得牛顿法的迭代公式

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \qquad (5-1)$$

现在以开平方计算为例来说明，设 $x=\sqrt{a}$，则有 $x^2-a=0$，利用（5—1）式得

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right) \qquad (5-2)$$

下面考察相对误差，$\varepsilon_k$ 的性态

$$\varepsilon_k = \frac{|x_k - \sqrt{a}|}{|x_k|} = \left|1 - \frac{\sqrt{a}}{x_k}\right|$$

$$\varepsilon_{k+1}=\frac{|x_{k+1}-\sqrt{a}|}{|x_{k+1}|}=\left|1-\frac{2\sqrt{a}/x_k}{1+(\frac{\sqrt{a}}{x_k})^2}\right|$$

上面两式联立求解得

$$\varepsilon_{k+1}=\frac{\varepsilon_k^2}{(1-\varepsilon_k)^2+1} \qquad (5-3)$$

当$\varepsilon_k$很小时$\varepsilon_{k+1}\approx\frac{\varepsilon_k^2}{2}$，由此可见，牛顿法的误差$\varepsilon_{k+1}$与$\varepsilon_k$的平方成正比。所以牛顿法的误差具有平方收敛性。

从迭代公式可看出，在一定精度要求下，迭代初值$x_0$的选择直接影响着迭代次数K，$x_0$越接近于$\sqrt{a}$则迭代次数越少，也即收敛速度越快。为了提高算法的效率，下面讨论一下迭代初值的选择问题。

为便于分析，我们以浮点数开平方为例来讨论，J代表阶码表达的指数，那么，操作数可表达为：

$$y=\sqrt{x2^J}=\begin{cases}\sqrt{x}2^{\frac{J}{2}} & (J\text{为偶数})\\ \sqrt{x}2^{\frac{J+1}{2}}\frac{1}{\sqrt{2}} & (J\text{为奇数})\end{cases}$$

除去阶码部分的运算，主要是计算$\sqrt{x}$。对于规格化的浮点数，有$\frac{1}{2}\leqslant x<1$。把区间$[\frac{1}{2},1]$ n等份，并使$n=2^m$（m为正整数），则节点$x_i=0.5+\frac{i}{2n}$。（$i=0,1,\cdots,n$，）

令$z=\sqrt{x}$　$x=x_i+h$，　$h\in[0,\frac{1}{2n})$，取$x_i$作为迭代初值代入迭代公式得

$$z_1=\frac{1}{2}(\sqrt{x_i}+\frac{x}{\sqrt{x_i}}) \qquad (5-4)$$

$$z_2=\frac{1}{2}(z_1+\frac{x}{z_1}) \qquad (5-5)$$

下面分析一下相对误差

$$\varepsilon_i=\frac{|z_1-\sqrt{x}|}{|z_1|}=\left|1-\frac{2\sqrt{xx_i}}{x_i+x}\right|$$

把$x=x_i+h$代入上式整理得

$$\varepsilon_i=\left|1-\frac{2\sqrt{1+h/x_i}}{2+h/x_i}\right|$$

不难证明$\frac{h}{x_i}$增加时，$\varepsilon_i$也增加，固取$h_{max}=\frac{1}{2n}$，　$x_i=x_{i\min}=\frac{1}{2}$，代入上式得

$$\varepsilon_{1max}=\left|1-\frac{2\sqrt{n(n+1)}}{2n+1}\right| \qquad (5-6)$$

把（5—6）代入（5—3）式得

$$\varepsilon_{2max}=\left|1-4\frac{(2n+1)\sqrt{n(n+1)}}{8n^2+8n+1}\right| \qquad (5-7)$$