

数值分析
及其应用

国际科学技术出版社

部分符号说明

| | |
|-------------------------------------|--|
| $C[a, b]$ | $[a, b]$ 区间上定义的连续函数集 |
| $C^*[a, b]$ | $[a, b]$ 区间上定义的, n 阶导数连续可微的函数集 |
| $C^\infty[a, b]$ | $[a, b]$ 区间上定义的光滑函数 (即, 任意阶导数都连续可微) 集 |
| R^n | n 维实空间 |
| C^n | n 维复空间 |
| $fl(y)$ | 实数 y 的浮点形式 |
| $O(\cdot)$ | 收敛次数或数量次数 |
| δ_{ij} | Kroneker 符号, $\delta_{ij} = \begin{cases} 1 & \text{当 } i=j \\ 0 & \text{当 } i \neq j \end{cases}$ |
| $x_n \leftrightarrow X_n$ | 付立叶变换对 |
| Π_n | 次数不超过 n 的多项式的集合 |
| H_n | 次数不超过 n , 首项系数是 1 的多项式的集合 |
| $\{x_k\}_0^n, \{x_k\}_{k=0}^n$ | 表示一组互异节点 x_0, x_1, \dots, x_n . |
| $\{x_k, f_k\}_0^n$ | 表示 $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$ 。其中 x_0, x_1, \dots, x_n 互异。 |
| (a_{ij}) | 表示矩阵 A , 它的 i 行 j 列元素是 a_{ij} |
| $[A, b]$ | 线性代数方程组系数矩阵 A 和自由项 b 组成的增广矩阵 |
| $O_{n \times n}, O$ | 零矩阵 |
| $I_{n \times n}, I$ | 单位矩阵 |
| $\ x\ $ | 向量 x 的模 |
| $\ x\ _p$ | 向量 x 的 p 模 $p=1, 2, \infty$ |
| $\ A\ $ | 矩阵 A 的模 |
| $\ A\ _p$ | 矩阵 A 的 p 模 $p=1, 2, \infty$ |
| $\rho(A)$ | 矩阵 A 的谱半径 |
| $K(A)$ | 矩阵 A 的条件数 |
| (x, y) | 向量 x 和 y 的内积 |
| \in | 属于 |
| $\mathcal{X}(x_0, x_1, \dots, x_n)$ | 表示区间 $[a, b]$ 。其中 $a = \min\{x_0, x_1, \dots, x_n\}$ $b = \max\{x_0, x_1, \dots, x_n\}$ |

目 录

部分符号说明

第一章 基础知识

| | |
|-----------------------|----|
| 1.1 误差 | 1 |
| 1.1.1 误差源 | 1 |
| 1.1.2 误差的初等分析 | 2 |
| 1.2 计算机的算术运算 | 3 |
| 1.2.1 定点数及其运算 | 3 |
| 1.2.2 浮点数及其运算 | 5 |
| 1.3 误差传播 | 7 |
| 1.3.1 算术运算的误差传播 | 7 |
| 1.3.2 函数计算的误差传播 | 9 |
| 1.4 函数的模 | 10 |
| 1.5 切彼晓夫正交多项式 | 10 |
| 1.6 正交多项式的一般性质 | 14 |
| 1.6.1 一般正交多项式的构成和递推关系 | 14 |
| 1.6.2 正交多项式的零点 | 16 |
| 1.6.3 函数的正交多项式展开 | 16 |
| 1.6.4 离散情况下的正交多项式 | 17 |
| 1.7 几种常用的正交多项式 | 18 |
| 1.7.1 第二类切彼晓夫多项式 | 18 |
| 1.7.2 勒让德多项式 | 19 |
| 1.7.3 拉盖尔多项式 | 20 |
| 1.7.4 埃尔米特多项式 | 20 |
| 小 结 | 21 |
| 附录一 部分切彼晓夫多项式 | 21 |
| 习题一 | 22 |

第二章 插值

| | |
|-------------------|----|
| 2.1 引言 | 24 |
| 2.2 拉格朗日插值多项式 | 24 |
| 2.3 差分、差商和牛顿插值多项式 | 28 |
| 2.4 埃尔米特插值多项式 | 33 |
| 2.5 样条插值 | 36 |
| 小 结 | 41 |

| | |
|-----|----|
| 习题二 | 41 |
|-----|----|

第三章 数值微分与数值积分

| | |
|--------------------------|----|
| 3.1 数值微分 | 44 |
| 3.2 牛顿-柯特斯求积公式 | 49 |
| 3.2.1 闭型牛顿-柯特斯求积公式 | 49 |
| 3.2.2 开型牛顿-柯特斯求积公式 | 53 |
| 3.2.3 牛顿-柯特斯求积公式的舍入误差 | 54 |
| 3.2.4 牛顿-柯特斯求积公式的收敛性 | 55 |
| 3.3 组合型求积公式和自适应积分 | 56 |
| 3.3.1 组合型求积公式 | 58 |
| 3.3.2 自适应积分方法 | 58 |
| 3.3.3 龙贝格积分 | 60 |
| 3.4 高斯型积分 | 64 |
| 3.4.1 引言 | 64 |
| 3.4.2 一般的高斯型求积公式 | 64 |
| 3.4.3 用正交多项式表示高斯型求积公式的系数 | 67 |
| 3.4.4 几种常用的高斯型求积公式 | 69 |
| 3.5* 数值积分的进一步讨论 | 75 |
| 3.5.1 奇异积分 | 75 |
| 3.5.2 样条积分 | 75 |
| 3.5.3 重积分的计算 | 76 |
| 小 结 | 79 |
| 习题三 | 79 |

第四章 常微分方程初值问题

| | |
|-------------------------------|----|
| 4.1 引言 | 82 |
| 4.2 尤拉方法 | 84 |
| 4.3 龙格-库塔方法 | 89 |
| 4.4 单步法的使用 | 95 |
| 4.4.1 李查逊加速和误差估计 | 95 |
| 4.4.2* 误差控制和龙格-库塔-Fehlberg 方法 | 95 |
| 4.5 阿达姆斯方法 | 95 |
| 4.5.1 阿达姆斯方法 | 95 |
| 4.5.2 | 95 |

4.5.3 阿达姆斯公式的使用和加速 103

4.5.4 预报-校正格式的收敛性 104

4.5.5* 变步长阿达姆斯预报-校正算法 ... 105

4.6* 米尔尼方法和哈明方法 107

4.6.1 米尔尼方法 107

4.6.2 哈明方法 109

4.7* 埃尔米特方法 110

4.8* 例 112

4.9 常微分方程组的数值解 115

4.10* 稳定性 118

4.10.1 齐次常系数线性差分方程 118

4.10.2 差分格式的稳定性 119

小结 123

习题四 123

第五章 函数逼近与计算

5.1 引言 127

5.2 函数的一致逼近 127

5.2.1 一致逼近的基本定理 127

5.2.2* 里米兹算法 129

5.3 多项式展开的函数逼近方法 130

5.3.1 台劳多项式逼近的局限性 130

5.3.2 函数的切彼晓夫多项式逼近 131

5.3.3 多项式的计算 134

5.3.4 幂级数精简 134

5.4 最小二乘曲线拟合 136

5.4.1 最小二乘的一般原理 136

5.4.2 多项式的最小二乘逼近 139

5.4.3 正交多项式的最小二乘逼近 141

5.4.4 Gram 多项式的最小二乘逼近 143

5.4.5 例 143

5.5* 函数的有理分式逼近 145

5.5.1 Pade逼近 145

5.5.2 Pade逼近的几个例题 146

5.5.3 有理分式计算 148

5.5.4 用切彼晓夫级数构造有理分式逼近 149

5.6* 周期函数的最小二乘法和离散付立叶变换 152

5.6.1 最小二乘法 152

5.6.2 离散付立叶变换 153

5.6.3 离散付立叶变换与周期函数的最小二乘法 156

5.7* 快速付立叶变换(FFT) 157

5.7.1 FFT的直观推导 157

5.7.2 $N = 2^r$ 的库利-图基算法推导 161

5.8* 实数据的FFT算法 165

5.8.1 同时计算两个实函数的FFT算法 165

5.8.2 用 N 点FFT计算 $2N$ 个实数的付立叶变换 167

小结 168

附录二 一些常用的初等函数逼近公式 169

习题五 172

第六章 非线性方程求根

6.1 二分法 175

6.2 函数迭代 176

6.2.1 一般单点迭代 176

6.2.2 多点迭代 177

6.3 弦截法和抛物线法 178

6.3.1 试位法 179

6.3.2 弦截法 181

6.3.3 抛物线法 184

6.4 牛顿迭代法 185

6.4.1 单根情况下的牛顿迭代公式 186

6.4.2 重根情况下的牛顿迭代公式 189

6.5 实多项式的求根方法 191

6.5.1 求多项式全部实根或复根的途径 191

6.5.2 用牛顿迭代法求多项式的根 193

6.5.3* 劈因子迭代法 194

6.5.4 实多项式系数误差对根的影响 198

6.6* 实多项式的实根分布 199

6.7* 非线性方程组 202

6.7.1 解非线性方程组的牛顿迭代法 203

6.7.2 最速下降法 205

小结 207

习题六 207

第七章 解线性代数方程组的直接法

7.1 引言 211

7.2 线性代数的基本理论 212

7.2.1 解的存在性和唯一性 212

7.2.2 内积和向量空间 212

| | | | |
|-------------------------------|-----|---------------------------------------|-----|
| 7.2.3 矩阵的特征值和特征向量 | 213 | 8.3 雅可比迭代 | 265 |
| 7.2.4 标准型 | 216 | 8.3.1 雅可比迭代 | 265 |
| 7.3 向量、矩阵的模 | 217 | 8.3.2 雅可比迭代的收敛性 | 267 |
| 7.3.1 向量模 | 217 | 8.4 赛德尔迭代 | 271 |
| 7.3.2 矩阵模 | 218 | 8.4.1 赛德尔迭代 | 271 |
| 7.4 高斯消去法 | 221 | 8.4.2 赛德尔迭代的收敛性 | 272 |
| 7.4.1 高斯消去法 | 222 | 8.5* 松弛法 | 275 |
| 7.4.2 高斯消去法与LU分解 | 225 | 8.5.1 松弛迭代的一般形式 | 275 |
| 7.5 选主元和加比例因子的 高斯消去法 | 229 | 8.5.2 坐标松弛法 | 277 |
| 7.5.1 高斯主元消去法 | 229 | 8.5.3 最优斜量法 | 279 |
| 7.5.2 加比例因子的高斯消去法 | 231 | 小 结 | 280 |
| 7.6 高斯消去法的变形 | 233 | 习题八 | 280 |
| 7.6.1 杜利特尔方法 | 233 | 第九章 矩阵特征值和特征向量的解法 | |
| 7.6.2 对称正定矩阵的平方根法 | 235 | 9.1 引言 | 283 |
| 7.6.3 三对角线矩阵的追赶法 | 237 | 9.2 幂法求按模最大特征值 | 283 |
| 7.7* 行列式和逆矩阵 | 241 | 9.3 幂法的变形与加速 | 290 |
| 7.7.1 行列式的计算 | 241 | 9.3.1 简单移位法 | 290 |
| 7.7.2 求逆矩阵 | 241 | 9.3.2 幂法的加速 | 290 |
| 7.7.3 分块法求逆矩阵 | 244 | 9.3.3 反幂法 | 292 |
| 7.8 误差分析 | 246 | 9.3.4 内积法 | 292 |
| 7.8.1 自由项扰动对解的影响 | 248 | 9.4 雅可比方法 | 294 |
| 7.8.2 矩阵的条件数 | 248 | 9.5* Givens方法和Householder 方法 | 298 |
| 7.8.3 $\ A^{-1}\ $ 的估计 | 250 | 9.5.1 Givens方法 | 298 |
| 7.8.4 系数矩阵扰动对解的影响 | 251 | 9.5.2 Householder方法 | 299 |
| 7.8.5* 舍入误差的影响 | 252 | 9.5.3 对称三对角阵的特征值 | 303 |
| 7.8.6* 剩余、误差及迭代校正法 | 253 | 9.5.4 求特征向量 | 305 |
| 小 结 | 255 | 9.6* LR和QR算法 | 306 |
| 习题七 | 255 | 9.6.1 一般原理 | 306 |
| 第八章 解线性代数方程组的迭代法 | | 9.6.2 收敛性 | 307 |
| 8.1 向量和矩阵的极限 | 259 | 9.6.3 计算技术 | 309 |
| 8.2 迭代法 | 262 | 小 结 | 318 |
| 8.2.1 迭代法的一般形式 | 262 | 习题九 | 318 |
| 8.2.2 迭代法的收敛性 | 263 | 参考书目 | 321 |
| 8.2.3 迭代法的收敛速度 | 264 | | |

第一章 基础知识

1.1 误差

1.1.1 误差源

除了极个别的情况外，数值计算总是近似计算，实际计算结果与理论结果之间存在着误差。数值分析的任务之一是，将误差控制在一定的容许范围内，或者至少对误差有所估计。为了研究在求解实际问题时产生的误差，我们回顾一下实际问题求解的过程。

(1) 将实际问题抽象为一个数学问题，或者说用一组数学公式描述一个实际问题。如，沿直线运动的物体在 t 时刻的位置坐标可用下式表示：

$$x_t = x_0 + x_0' t + \frac{1}{2} x_0'' t^2 + \frac{1}{3!} x_0''' t^3 + \dots$$

式中 x_0, x_t 分别表示运动物体在初始时刻 $t_0=0$ 和 t 时刻的位置。 $x_0', x_0'', x_0''' \dots$ 分别表示初始时刻物体的速度，加速度，加速度的变化率等等。

(2) 用有限过程取代数学公式中的极限（无限）过程。如假设物体在某一段时间内作匀加速运动，于是 x_t 可简化为

$$x_t = x_0 + x_0' t + \frac{1}{2} x_0'' t^2$$

(3) 为数学模型提供全部参数，如 x_0, x_0', x_0'', t 。

(4) 根据数学模型和给出的参数值写出程序并在机器上运行，求出计算结果。

分析上述过程，可以发现计算结果含有下列几种误差：

- (1) 数学模型与客观事物之间的描述误差。
- (2) 用有限过程代替数学模型无限过程时产生的截断误差。
- (3) 计算或测量模型参数的误差。
- (4) 在计算机有限字长环境中运算产生的舍入误差。
- (5) 上述过程可能出现的过失误差。

这五种误差中，过失误差是应该避免的。建立数学模型的工作不是数值分析研究的内容。参数选择往往涉及仪表的精度和测量技术。数学模型的离散化、有限化，截断误差和舍入误差的分析，误差的传播规律是数值分析研究的主要内容。值得指出的是，一个题目由不同的程序员，用不同的程序语言描述，计算结果的误差可能相差十分悬殊。因为不同的计算次序，不同的软件系统和字长对计算结果的影响是明显的，有时得从上面的分类来区分。特别是截断误差和舍入误差互为因果关系，相互影响，因而在理论上还是实践上误差控制和估计都有一定的困难。

1.1.2 误差的初等分析

现在介绍误差、相对误差、有效数字和它们之间的关系。

定义 1.1 设 x_T 为真值 (精确值), x_A 为近似值

$$\text{ERR}(x_A) = x_T - x_A \quad (1.1)$$

称为近似值 x_A 的误差。

一般说来, 真值 x_T 很难给出, 因此求 $\text{ERR}(x_A)$ 是困难的。实践中用 $|\text{ERR}(x_A)|$ 尽可能小的上界估计 x_A 的误差, 称为误差限 E_e , 显然 $x_A \in [x_T - E_e, x_T + E_e]$ 。

定义 1.2 x_A 的误差与真值 x_T 的比称为 x_A 的相对误差, 记为 $\text{REL}(x_A)$

$$\text{REL}(x_A) = (x_T - x_A) / x_T \quad (1.2)$$

由于 x_T 是未知量, 因此 $\text{REL}(x_A)$ 无法计算, 和计算 $\text{ERR}(x_A)$ 类似, 人们用 $|\text{REL}(x_A)|$ 尽可能小的上界估计 x_A 的相对误差, 称为相对误差限 E_r 。实践中, 常用 $\text{REL}^*(x_A) = (x_T - x_A) / x_A$ 代替 (1.2) 式定义的相对误差。

例 1.1

$$x_T = e = 2.7182818 \dots$$

$$x_A = 19/7 = 2.7142857 \dots$$

$$\text{ERR}(x_A) = 0.003996 \dots$$

$$\text{REL}(x_A) = 0.00147 \dots \quad \square$$

定义 1.3 将近似值 x_A 表示为

$$x_A = \pm \beta^e (a_1 \beta^{-1} + a_2 \beta^{-2} + \dots + a_n \beta^{-n}) \quad (1.3)$$

式中 e 为指数, a_1, a_2, \dots, a_n 为尾数且 $a_1 \neq 0$, β 是正整数称为基底。若近似值 x_A 满足

$$|x_T - x_A| \leq \frac{1}{2} \beta^{e-n} \quad (1.4)$$

则称 x_A 有 n 位有效数字 a_1, a_2, \dots, a_n 。

(1.3) 中, $\beta = 10$ 时表示十进制数, $\beta = 2$ 时表示计算机的二进制规格化浮点数。

例 1.2

$$(1) \quad x_T = 1/3 \quad x_A = 0.333$$

$$|x_T - x_A| \approx 0.00033 < 0.5 \times 10^{-3}$$

由于 x_A 的第一位非零数字至 10^{-3} 位有三位数字, 故 x_A 有 3 位有效数字。

$$(2) \quad x_T = 23.496 \quad x_A = 23.494$$

$$|x_T - x_A| = 0.002 < 0.5 \times 10^{-2}$$

x_A 的第一位非零数字至 10^{-2} 位有四位数字, 故 x_A 有 4 位有效数字。当 x_A 的最后一位有舍入, 取值 23.49 时, 引进了新的舍入误差, 这时 x_A 就不再有四位有效数字。

$$(3) \quad x_T = 0.02138 \quad x_A = 0.02144$$

$$|x_T - x_A| = 0.00006 < 0.5 \times 10^{-3}$$

在 10^{-3} 位之前, x_A 仅有一位非零数字, 故 x_A 有两位有效数字。 \square

(1.4) 式说明, x_A 的有效数字位数越多, x_A 的误差限就越小。下面讨论有效数字和 $\text{REL}^*(x_A)$ 的关系。

1.1. (1.3) 式表示的近似数 x_A 若具有 n 位有效数字, 则相对误差 $\text{REL}^*(x_A)$

满足

$$|\text{REL}^*(x_A)| \leq \frac{1}{2a_1} \beta^{-(n-1)} \quad (1.5)$$

式中 $a_1 \neq 0$, 是 x_A 的第一位有效数字。

证明 由(1.3)知, $|x_A| \geq a_1 \beta^{e-1}$, 故

$$|\text{REL}^*(x_A)| = |x_T - x_A| / |x_A| \leq \left(\frac{1}{2} \beta^{e-n} \right) / (a_1 \beta^{e-1}) = \frac{1}{2a_1} \beta^{-(n-1)} \quad \square$$

定理 1.2 (1.3)式表示的近似数 x_A 若满足

$$|\text{REL}^*(x_A)| \leq \frac{1}{2(a_1 + 1)} \beta^{-(n-1)} \quad (1.6)$$

则 x_A 至少有 n 位有效数字。

证明 因为

$$|x_T - x_A| = |x_A| |\text{REL}^*(x_A)|, \quad |x_A| < (a_1 + 1) \beta^{e-1}$$

故

$$|x_T - x_A| < (a_1 + 1) \beta^{e-1} \frac{1}{2(a_1 + 1)} \beta^{-(n-1)} = \frac{1}{2} \beta^{e-n}.$$

即 x_A 至少有 n 位有效数字 □

这两个定理说明, 近似数 x_A 的有效数字越多, 它的相对误差越小。反之, x_A 的相对误差越小, 它的有效数字越多。

1.2 计算机的算术运算

数值计算的现代工具是数字计算机。数字机的字长和运算方式对数值计算的结果有直接的影响。对给定的数值方法, 一个注意到计算机有限字长和运算方式的程序员, 可以写出具有较高计算精度的程序。反之, 也会得到十分粗糙甚至完全失真的计算结果。因此, 了解计算机数的表示和运算方式, 对使用计算机十分必要。

1.2.1 定点数及其运算

定点数是多数计算机都具备的数据类型, 定点数值计算是这些计算机最基本的运算方式。定点数的表示通常采用二进制原码或补码形式。它的运算过程与人们用铅笔和纸进行计算类似。一个明显的差别是, 有的系统把小数点定在数的最左边(图1-1), 此时数的绝对值不大于1。而另一些系统把小数点定在数的最右边(图1-2), 此时数的最右一位表示整数1。计算机中数的符号多用左边第一位表示。零表示正数, 1表示负数, 基底为2。

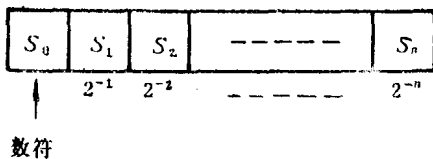


图 1-1 定点数的表示之一

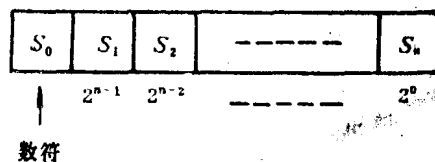


图 1-2 定点数的表示之二

图1-2表示的定点数相当于FORTRAN中的整型数，用于程序的逻辑控制和数组的下标等等。

例 1.3 在八位二进制原码系统中，将小数点定在最右边，第一位表示数符。

$$2 = (0000\ 0010)_2^{\ominus} \quad -10 = (1000\ 1010)_2 \quad \square$$

在补码系统中，正数表示与原码相同，负数的补码等于原码变反，再将尾数加1而符号位不变。如，-4的二进制补码是1111 1100。于是，两个数相减可以简化为减数变补与被减数相加。如

$$\begin{aligned} -4 - 2 &= (1111\ 1100)_2^{\text{补码}} - (0000\ 0010)_2 \\ &= (1111\ 1100)_2^{\text{补码}} + (1111\ 1110)_2^{\text{补码}} \\ &= (1111\ 1010)_2^{\text{补码}} \end{aligned}$$

数值计算的多数数据，不仅有整数部分，而且还有小数部份。在用定点系统进行科学计算时，人们常把小数点定在最左边，为此采用加比例因子的办法，把原始数据映照到 $[-1, 1)$ 区间。如，用 $1/2, 1/4, 1/8$ 做参数，引用函数 $\sin 2\pi x$ 就可以计算 $\sin \pi, \sin(\pi/2), \sin(\pi/4)$ 的值。

例 1.4 在8位二进制系统中，小数点定在最左边， $3/4 = (01100000)_2, -1/8 = (1001\ 0000)_2^{\text{原码}} = (1111\ 0000)_2^{\text{补码}}$ 。算术运算过程是

$$\begin{aligned} \frac{3}{4} - \frac{1}{4} &= (0110\ 0000)_2 - (0010\ 0000)_2 \\ &= (0110\ 0000)_2 + (1110\ 0000)_2^{\text{补码}} \\ &= (0100\ 0000)_2 \\ \frac{1}{4} \times \left(-\frac{1}{8}\right) &= (0010\ 0000)_2 \times (11110000)_2^{\text{补码}} \\ &= (1111\ 1100)_2^{\text{补码}} \\ \left(-\frac{1}{8}\right) / \left(\frac{1}{4}\right) &= (1111\ 0000)_2^{\text{补码}} / (0010\ 0000)_2 \\ &= (1100\ 0000)_2^{\text{补码}} \quad \square \end{aligned}$$

值得指出的是，在左边定小数点的定点运算系统中，不仅要求运算操作数的绝对值小于1，而且还要求算术运算结果的绝对值小于1。如， $x=0.5, y=0.6, z=0.5+0.6=1.1$ 超过了定点数的上界，产生上溢出。这是因为 x, y 比例因子选择不当引起的。又如，在16位二进制系统的计算机上计算 $w=2^{-7} \times 2^{-9} / 2^{-10}$ 。用下面两种算法，得到完全不同的结果。

算法1 $w = (2^{-7} \times 2^{-9}) / 2^{-10} = 0 / 2^{-10} = 0$

算法2 $w = (2^{-7} / 2^{-5}) \times (2^{-9} / 2^{-5}) = 2^{-2} \times 2^{-4} = 2^{-6}$

由于计算机字长的限制，算法1中分子的运算产生下溢出。这些事实说明，在编制定点运算程序时，要尽量避免运算结果的上溢出或下溢出，为此，比例因子和计算次序的选择要十分慎重。近年来，由于浮点功能部件成本的下降，用定点系统进行科学计算已越

⊖ (0000 0010)₂表示二进制数

来越少。在用计算机进行实时控制的领域里，采用定点系统进行计算和控制仍十分普遍。这是因为，定点运算字长短、逻辑简单，因此有十分高的运算速度，加上多数控制系统的输入输出数据精度低，数据的定点形式易于数字量和模拟量的相互转换等等，因而大大简化了系统的复杂性、降低了系统成本。但是，用定点系统描述一个算法，要想获得比较高的运算精度和比较快的运算速度，对程序员的要求是很高的。以前这样的程序多用汇编语言编写，由于83年公布的标准Ada语言已有定点运算系统，今后，人们可以用Ada语言写定点运算的程序了。

1.2.2 浮点数及其运算

用于数值计算的计算机多采用浮点系统。因为用浮点方式表示的数有比较大的取值范围，浮点算术运算有较高的计算精度，从而为编制程序提供了方便。目前多数计算机都有浮点运算的能力。某些微小型计算机为了降低成本甚至用软件实现浮点运算。当然，这样运算速度就很低了。

在字长为 l 的二进制计算机系统中，浮点数由一位符号位， p 位指数（含指数的符号位）和 q 位尾数组成（图1-3）。

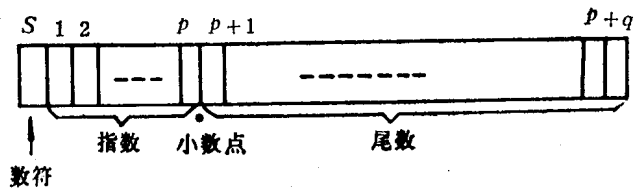


图 1-3 浮点数的表示

不同的计算机系统，字长 l 和 p ， q 的选择是不同的，指数和尾数的表示方式也不同。多数机器的指数用补码表示，尾数用原码。如，PDP-11计算机单精度浮点数 $l=32$ ，占两个机器字， $p=8$ ， $q=23$ 。表示数的范围是 $(-0.17 \times 10^{39}, 0.17 \times 10^{39})$ 。CRAY-1巨型计算机 $l=64$ ，占一个机器字， $p=16$ ， $q=47$ 。表示数的范围是 $(-10^{2466}, 10^{2466})$ 。双精度浮点数字长是单精度的两倍，阶码与单精度相同。因此数的表示范围并没扩大，但尾数字长增加一倍，导致有效数字增加。

为减少有效数字丢失，提高运算精度，计算机的浮点数通常用规格化的形式表示。一个非零的规格化浮点数，它的小数点后一位是1。尾数 m 满足 $0.5 \leq |m| < 1$ 。

例 1.5 $(1101.101)_2$ 规格化浮点数是 $2^4 \times 0.1101101$ ， $(0.00101)_2$ 规格化浮点数是 $2^{-3} \times 0.101$ 。 □

两个形如图 1-3 所示的规格化浮点数 x ， y 经 ω^* 操作后，得到一个 l 位结果，其中 ω^* 表示计算机进行的加减乘除操作。设所有运算结果都是先规格化然后再舍入，这一过程用 $f_l(x\omega^*y)$ 表示，当运算结果的尾数 m 超过 q 位时，舍入就会发生。为了方便，假设 ω^* 在 $2l$ 长的寄存器中执行， q 位二进制尾数舍入值 m_r 的定义是

$$m_r = \begin{cases} 2^{-q} \lfloor 2^q m + \frac{1}{2} \rfloor & \text{当 } m > 0 \\ 2^{-q} \lceil 2^q m - \frac{1}{2} \rceil & \text{当 } m < 0 \end{cases} \quad (1.7)$$

函数 $\lfloor x \rfloor$ 表示小于、等于 x 的最大整数。 $\lceil x \rceil$ 表示大于、等于 x 的最小整数。公式 (1.7) 表示人们常用的四舍五入规则。根据这个规则，舍入误差的绝对值为 $|e| \leq 2^{-q-1}$ ，而最大相对误差是 $2^{-q-1}/(1/2) = 2^{-q}$ 。

下面讨论两个规格化浮点数 $x = 2^{f_x} m_x$, $y = 2^{f_y} m_y$ 进行加、减、乘、除运算的过程，先从最简单的乘法开始。

(1) 乘法

$fl(x \times y)$ 的运算过程是：(1) 指数相加，尾数相乘。(2) 将运算结果规格化。(3) 尾数按 (1.7) 式舍入。这一过程的计算公式是

$$fl(x \times y) = 2^{f_x + f_y} m_x m_y \quad (1.8)$$

因为 x, y 都是规格化的数，所以 $1/4 \leq |m_x m_y| < 1$ ，当 $1/4 \leq |m_x m_y| < 1/2$ 时，将结果左移一位，同时阶码（指数）减 1，实现结果的规格化。最后进行舍入

$$fl(x \times y) = \begin{cases} 2^{f_x + f_y} (m_x m_y + e) & \text{当 } \frac{1}{2} \leq |m_x m_y| < 1 \\ 2^{f_x + f_y - 1} (2 m_x m_y + e) & \text{当 } \frac{1}{4} \leq |m_x m_y| < \frac{1}{2} \end{cases} \\ = x \times y (1 + e_m) \quad (1.9)$$

其中 $|e_m| \leq 2|e| \leq 2^{-q}$

由此可见，浮点乘积值等于精确乘积值再乘以 $(1 + e_m)$ ，相对误差界与舍入一个数的相对误差界相同。

(2) 除法

$fl(x/y)$ 的运算过程是：分子和分母的指数相减，尾数相除（要避免商的值大于 1），再对结果进行规格化和舍入。

设 $y \neq 0$ ，用公式描述这一算法是

$$fl(x/y) = 2^{f_x - f_y + 1} (2^{-1} m_x / m_y) \quad (1.10)$$

其中 $1/4 \leq |2^{-1} m_x / m_y| < 1$

使用和乘法类似的误差分析方法

$$fl(x/y) = x/y (1 + e_d) \quad (1.11)$$

式中 $|e_d| \leq 2^{-q}$ 。除法的相对误差界与舍入一个数的相对误差界相同。

(3) 加法和减法

$fl(x \pm y)$ 的运算过程是：(1) 绝对值较小的操作数（设 $|y| < |x|$ ）尾数右移 $f_x - f_y$ 位。

(2) 尾数 m_x 与移位后的尾数 m_y 相加（减）并进行规格化，当尾数溢出时右移一位指数加 1。(3) 结果指数加 f_x 并对尾数进行舍入。于是

$$fl(x \pm y) = (m_x \pm m_y \times 2^{f_y - f_x}) \times 2^{f_x} \quad (1.12)$$

当和差的真值非零时

$$2^{-q} \leq |m_x \pm m_y \times 2^{fv-fz}| < 2 \quad (1.13)$$

为了得到规格化的运算结果, 必须将(1.13)式的结果移 t 位, 其中 $-1 \leq t \leq q-1$, 当 $t = -1$ 时表示右移一位, 反之左移 t 位。一般说来 t 满足

$$2^{-t-1} \leq |m_x \pm m_y \times 2^{fv-fz}| < 2^{-t} \quad (1.14)$$

考虑到计算的舍入误差

$$\begin{aligned} fl(x \pm y) &= [(m_x \pm m_y \times 2^{fv-fz})2^t + e] \times 2^{fz-t} \\ &= (x \pm y) [1 + 2^{-t}e / (m_x \pm m_y \times 2^{fv-fz})] \end{aligned}$$

令 $e_a = 2^{-t}e / (m_x \pm m_y \times 2^{fv-fz})$ 则

$$fl(x \pm y) = (x \pm y)(1 + e_a) \quad (1.15)$$

由(1.14)可得 $|e_a| \leq 2|e| \leq 2^{-q}$

由此可见, 所有浮点运算的相对误差都不超过尾数的最后一位。在上面的讨论中, 假设运算是在字长为 $2l$ 的双寄存器上进行的。如果使用的寄存器只有 l 位, 计算结果的舍入误差会有很大变化, 如在 l 位寄存器上进行加减运算, (1.15)中的 $|e_a| \leq 3/2 \times 2^{-q}$ 。

尽管浮点数的范围比较大, 但仍会出现上溢出和下溢出的问题。若计算机所能表示的最大浮点数是 $M \times 2^F$, 其中 F 是最大正指数, $M = 1 - 2^{-q}$ 。当 $p = 8$ 时, $F = 128 - 1 = 127$ 。若 $(1/2 \times 2^{70}) \times (1/2 \times 2^{80})$, 则 $1/2 \times 2^{140} > 1 \times 2^{127}$ 发生上溢出。其次, $1/2 \times 2^{-F}$ 是系统所能表示的最小规格化浮点数。当 $p = 8$ 时, $1/2 \times 2^{-80} / (1/2 \times 2^{50})$ 得 $1/2 \times 2^{-129} < 1/2 \times 2^{-128}$, 于是产生下溢出。和定点系统一样, 上溢出的计算结果是错误的, 下溢出的计算结果为零。

1.3 误差传播

这一节讨论加、减、乘、除运算的误差传播问题。令 ω 表示数学定义的算术运算 +、-、 \times 、 $/$ 。 ω^* 表示计算机系统的相应操作。设

$$x_T = x_A + e \quad y_T = y_A + \eta$$

$x_T \omega y_T$ 的数学运算与近似值在计算机上的运算 $x_A \omega^* y_A$ 差是

$$x_T \omega y_T - x_A \omega^* y_A = (x_T \omega y_T - x_A \omega y_A) + (x_A \omega y_A - x_A \omega^* y_A) \quad (1.16)$$

上式第一项表示真值与近似值做 ω 运算产生的误差, 第二项是同一对操作数按 ω 运算和 ω^* 运算得到的误差。前者称为传播误差, 后者正是第二节讨论的计算舍入误差。

1.3.1 算术运算的误差传播

(1) 加减法的误差传播

$$(x_T \pm y_T) - (x_A \pm y_A) = (x_T - x_A) \pm (y_T - y_A) = e \pm \eta \quad (1.17)$$

(1.17)式说明两个数和或差的误差等于两个数误差的和或差。即

$$ERR(x_A \pm y_A) = ERR(x_A) \pm ERR(y_A) \quad (1.18)$$

对于相对误差

$$\begin{aligned} REL(x_A \pm y_A) &= [ERR(x_A) \pm ERR(y_A)] / (x_T \pm y_T) \\ &= REL(x_A) x_T / (x_T \pm y_T) \pm REL(y_A) y_T / (x_T \pm y_T) \quad (1.19) \end{aligned}$$

当 x_T, y_T 同号时, $x_A + y_A$ 的相对误差变化不大, 但 x_T, y_T 异号时, $x_A + y_A$ 的相对误差可能被放大很多倍。反之, 当 x_T, y_T 异号时, $x_A - y_A$ 相对误差变化不大, 但 x_T 和 y_T 同号时, $x_A - y_A$ 的相对误差可能被放大很多倍。因此, 实践中应尽量避免两个近似的同号数相减或异号数相加, 这样才能减少有效数字的丢失。

例 1.6 解二次方程 $x^2 - 26x + 1 = 0$ 。使用二次方程的求根公式得

$$x_T^{(1)} = 13 + \sqrt{168} \quad x_T^{(2)} = 13 - \sqrt{168}$$

查五位平方根表 $\sqrt{168} \approx 12.961$ 。利用有效数字定义, $|\sqrt{168} - 12.961| \leq 0.0005$ 。于是

$$x_A^{(1)} = 13 + 12.961 = 25.961$$

$$x_A^{(2)} = 13 - 12.961 = 0.039$$

$$|\text{ERR}(x_A^{(1)})| = |\text{ERR}(x_A^{(2)})| \leq 0.0005$$

相对误差

$$|\text{REL}^*(x_A^{(1)})| \leq \frac{0.0005}{25.9605} \approx 1.9 \times 10^{-5}$$

$$|\text{REL}^*(x_A^{(2)})| \leq \frac{0.0005}{0.0385} \approx 1.3 \times 10^{-2}$$

尽管 $x_A^{(2)}$ 的绝对误差比较小但相对误差却很大。原因是计算 $x_A^{(2)}$ 时有效数字的丢失比较多。如果把计算 $x_A^{(2)}$ 的公式改为

$$x_T^{(2)} = 13 - \sqrt{168} = \frac{1}{13 + \sqrt{168}} \quad \text{则}$$

$$x_A^{(2)} = \frac{1}{25.961} \approx 0.03851932$$

这是 $x_T^{(2)}$ 比较好的近似, 因为

$$\begin{aligned} |x_T^{(2)} - x_A^{(2)}| &\leq \left| x_T^{(2)} - \frac{1}{25.961} \right| + \left| \frac{1}{25.961} - 0.03851932 \right| \\ &\leq |\text{ERR}(1/25.961)| + 5 \times 10^{-9} \\ |\text{REL}(x_A^{(2)})| &\leq |\text{REL}(1/25.961)| + 5 \times 10^{-9} / 0.038519 \dots \\ &\leq 1.9 \times 10^{-5} + 1.3 \times 10^{-7} \leq 1.91 \times 10^{-5} \end{aligned}$$

$x_A^{(2)}$ 的相对误差与 $x_A^{(1)}$ 的相对误差基本一样。 □

(2) 乘法的误差传播

$$\text{ERR}(x_A y_A) = x_T y_T - x_A y_A \approx y_T \text{ERR}(x_A) + x_T \text{ERR}(y_A) \quad (1.20)$$

相对误差的传播规律是

$$\text{REL}(x_A y_A) = (x_T y_T - x_A y_A) / (x_T y_T) \approx \text{REL}(x_A) + \text{REL}(y_A) \quad (1.21)$$

(3) 除法的误差传播

$$\begin{aligned} \text{ERR}(x_A / y_A) &= (x_T / y_A) (y_A - y_T) / y_T + (x_T / y_A) (x_T - x_A) / x_T \\ &\approx (x_T / y_T) (\text{REL}(x_A) - \text{REL}(y_A)) \end{aligned} \quad (1.22)$$

$$\text{REL}(x_A / y_A) = (\text{REL}(x_A) - \text{REL}(y_A)) / (1 - \text{REL}(y_A)) \quad (1.23)$$

当 $|\text{REL}(y_A)| \ll 1$ 时

$$\text{REL}(x_A/y_A) \approx \text{REL}(x_A) - \text{REL}(y_A) \quad (1.24)$$

(1.21)和(1.24)式表明,乘法和除法的相对误差传播不快。

1.3.2 函数计算的误差传播

设 $f(x)$ 在 $[a, b]$ 上连续可微,我们讨论用 x_A 代替 x_T 计算 $f(x)$ 的误差 $f(x_T) - f(x_A)$ 。

当 x_A 接近于 x_T 时,由台劳展式得

$$f(x_T) - f(x_A) \approx f'(x_T)(x_T - x_A)$$

于是

$$\text{ERR}(f(x_A)) \approx f'(x_T)\text{ERR}(x_A) \quad (1.25)$$

即,用 x_A 计算 $f(x)$ 产生的误差与 x_A 的误差成正比,比例因子是 $f'(x_T)$ 。

$$\text{REL}(f(x_A)) \approx \text{ERR}(x_A)(f'(x_T)/f(x_T)) \quad (1.26)$$

用 x_A 计算 $f(x)$ 时产生的相对误差,与 x_A 的误差和 $f'(x_T)/f(x_T)$ 成正比。

例 1.7 $\sin(\pi/5) - \sin(0.628) \approx \cos(\pi/5)[\pi/5 - 0.628] \approx 0.00026$

$$[\sin(\pi/5) - \sin(0.628)]/\sin(\pi/5) \approx [\cos(\pi/5)/\sin(\pi/5)]$$

$$\cdot (\pi/5 - 0.628) \approx 0.00041. \quad \square$$

对区域 D 上连续可微的二元函数 $f(x, y)$, 用 (x_A, y_A) 代替 (x_T, y_T) 计算函数 $f(x, y)$ 的误差 $f(x_T, y_T) - f(x_A, y_A)$ 可用二元函数的台劳公式获得, 当 (x_A, y_A) 接近 (x_T, y_T) 时

$$\begin{aligned} \text{ERR}(f(x_A, y_A)) &= f(x_T, y_T) - f(x_A, y_A) \\ &\approx f_x(x_T, y_T)(x_T - x_A) + f_y(x_T, y_T)(y_T - y_A) \end{aligned}$$

于是

$$\text{ERR}(f(x_A, y_A)) \approx f_x(x_T, y_T)\text{ERR}(x_A) + f_y(x_T, y_T)\text{ERR}(y_A) \quad (1.27)$$

上式中的 $f_x = \partial f / \partial x$, $f_y = \partial f / \partial y$ 。

$$\begin{aligned} \text{REL}(f(x_A, y_A)) &\approx (f_x(x_T, y_T)/f(x_T, y_T))\text{ERR}(x_A) \\ &\quad + (f_y(x_T, y_T)/f(x_T, y_T))\text{ERR}(y_A) \end{aligned} \quad (1.28)$$

(1.27)和(1.28)给出了 $f(x_A, y_A)$ 的误差和相对误差估计。

例 1.8 求 $f(x, y) = x^y$ 的误差估计式。

$$f_x = yx^{y-1}, \quad f_y = x^y \ln x$$

$$\text{ERR}(x_A^y) \approx y_T x_T^{y_T-1} \text{ERR}(x_A) + x_T^{y_T} \ln x_T \text{ERR}(y_A) \quad (1.29)$$

$$\text{REL}(x_A^y) \approx y_T (\text{REL}(x_A) + \text{REL}(y_A) \ln x_T) \quad (1.30)$$

x_A^y 的相对误差与 y_T 和 $\ln x_T$ 的值密切相关, 当 $y_T = y_A = 500$, $x_T = 1.2$, $x_A = 1.2001$ 时 $x_T^{y_T} = 3.89604 \times 10^{99}$, $x_A^{y_A} = 4.06179 \times 10^{99}$, $\text{REL}(x_A^y) = -0.0425$, 它与 $\text{REL}(x_A) = 8.3 \times 10^{-5}$ 比较就相当大了。 \square

与二元函数一样,还可以写出多元函数 $f(x_1, x_2, \dots, x_n)$ 的误差公式

$$\text{ERR}(f(x_{1A}, x_{2A}, \dots, x_{nA})) \approx \sum_{i=1}^n f_{x_i}(x_{1T}, x_{2T}, \dots, x_{nT}) \text{ERR}(x_{iA}) \quad (1.31)$$

$$\begin{aligned} \text{REL}(f(x_{1A}, x_{2A}, \dots, x_{nA})) &\approx \left[\sum_{i=1}^n f_{x_i}(x_{1T}, x_{2T}, \dots, x_{nT}) \text{ERR}(x_{iA}) \right] \\ &\quad / f(x_{1T}, x_{2T}, \dots, x_{nT}) \end{aligned} \quad (1.32)$$

1.4 函数的模

前几节讨论了近似数的误差, 运算的舍入误差, 以及误差的传播。误差的“大小”都可以用数字度量。但是, 区间 $[a, b]$ 或一组离散点 $\{x_i\}_n$ 上给出的函数 $f(x)$ 与 $g(x)$ 的偏离程度又如何度量呢? 为此, 必须引入函数模 (也称范数) 的概念。

如果函数 $f(x) \in C[a, b]$, $f(x)$ 在区间 $[a, b]$ 的切彼晓夫模定义为

$$\|f(x)\|_\infty = \max_{a \leq x \leq b} |f(x)| \quad (1.33)$$

欧几里德模定义为

$$\|f(x)\|_2 = \sqrt{\int_a^b f^2(x) dx} \quad (1.34)$$

$\|f(x)\|_\infty$ 和 $\|f(x)\|_2$ 都满足

- (1) $\|f(x)\| \geq 0$ 当且仅当 $f(x) \equiv 0$ 时, $\|f(x)\| = 0$
- (2) $\|\alpha f(x)\| = |\alpha| \|f(x)\|$ 其中 α 为常数
- (3) 区间 $[a, b]$ 的任意连续函数 $f(x), g(x)$ 满足

$$\|f(x) + g(x)\| \leq \|f(x)\| + \|g(x)\| \quad (1.35)$$

由(1.35)可直接推出

$$\begin{aligned} \|f(x) - g(x)\| &\leq \|f(x) - h(x)\| + \|h(x) - g(x)\| \\ |\|f(x)\| - \|g(x)\|| &\leq \|f(x) - g(x)\| \end{aligned}$$

在离散场合, 给出函数 $f(x)$ 在一组离散点 $\{x_i\}_n$ 上的函数值 $\{f(x_i)\}_n$ 后, 函数的切彼晓夫模定义为

$$\|f(x)\|_\infty = \max_{0 \leq i < n} |f(x_i)| \quad (1.36)$$

函数的欧几里德模定义为

$$\|f(x)\|_2 = \sqrt{\sum_{i=0}^n f^2(x_i)} \quad (1.37)$$

函数的模是十分重要的概念。第五章讨论, 用多项式 $P_n(x)$ 逼近函数 $f(x)$ 的问题。函数的一致逼近采用切彼晓夫模 $\|f(x) - P_n(x)\|_\infty$, 度量 $P_n(x)$ 与 $f(x)$ 在区间 $[a, b]$ 或一组离散点 $\{x_i\}_n$ 上的偏离程度, $\|f(x) - P_n(x)\|_\infty$ 称为 $f(x)$ 与 $P_n(x)$ 的偏差。而函数的最小二乘逼近是用欧几里德模 $\|f(x) - P_n(x)\|_2$ 度量的。不同的应用领域可以选用不同的模。

1.5 切彼晓夫 (Chebyshev) 正交多项式

切彼晓夫引进了一组特殊的多项式, 它们在数值分析的理论 and 实践中起着十分重要的作用。这一节介绍切彼晓夫正交多项式的定义和性质。

定义 1.4 在区间 $[-1, 1]$ 定义

$$T_n(x) = \cos(n \arccos x) \quad (1.38)$$

为 n 次第一类切彼晓夫多项式, 简称切彼晓夫多项式。

表面看来, $T_n(x)$ 是一个三角函数。令

$$\begin{aligned} \theta &= \arccos x & x &\in [-1, 1] \quad \text{则} \\ x &= \cos \theta & \theta &\in [0, \pi] \end{aligned}$$

利用三角恒等式

$$\cos(n+1)\theta = 2\cos\theta\cos n\theta - \cos(n-1)\theta \quad (1.39)$$

可以导出递推关系

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (1.40)$$

由 $T_n(x)$ 的定义(1.38)知, $T_0(x) = 1, T_1(x) = x$ 代入(1.40)

$$\begin{aligned} T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \\ T_5(x) &= 16x^5 - 20x^3 + 5x \\ &\dots\dots\dots \end{aligned} \quad (1.41)$$

反之, x^n 也可以表示成 $T_0(x), T_1(x), \dots, T_n(x)$ 的线性组合

$$\begin{aligned} 1 &= T_0(x) \\ x &= T_1(x) \\ x^2 &= \frac{1}{2}(T_0(x) + T_2(x)) \\ x^3 &= \frac{1}{4}(3T_1(x) + T_3(x)) \\ x^4 &= \frac{1}{8}(3T_0(x) + 4T_2(x) + T_4(x)) \\ x^5 &= \frac{1}{16}(10T_1(x) + 5T_3(x) + T_5(x)) \\ &\dots\dots\dots \end{aligned} \quad (1.42)$$

部分切彼晓夫多项式与幂函数之间的关系列于表1-1和表1-2。 $T_n(x)$ 的三角表达式(1.38)只在区间 $|x| \leq 1$ 时成立, 若把 $[-1, 1]$ 上的切彼晓夫多项式用于区间 $[a, b]$ 可做变换

$$\xi = \frac{1}{2}[(b-a)x + (a+b)] \quad x \in [-1, 1]$$

于是多项式

$$T_n\left(\frac{2\xi - a - b}{b - a}\right) = \cos\left(n \arccos \frac{2\xi - a - b}{b - a}\right) \quad (1.43)$$

为区间 $[a, b]$ 上的 n 次切彼晓夫多项式。

根据切彼晓夫多项式的定义和递推关系可以得出一系列虽然简单却十分重要的性质。

$$(1) \text{ 奇偶性 } T_n(x) = (-1)^n T_n(-x) \quad (1.44)$$

将 $T_0(x), T_1(x)$ 代入 (1.40) 可立即推出 (1.44).

(2) 因为余弦函数的绝对值不大于 1, 因此对任意 $x \in [-1, 1]$ 有 $|T_n(x)| \leq 1$.

(3) 由三项递推公式 (1.40) 可见, $T_n(x)$ 的首项系数是 2^{n-1} , 因此 $\tilde{T}_n(x) = 2^{1-n} T_n(x)$ 是首项系数为 1 的第一类切彼晓夫多项式.

(4) $T_n(x)$ 的 n 个根 (零点) 都是单重实根并全部落在区间 $(-1, 1)$, 它们是

$$x_m^{(n)} = \cos \frac{2m-1}{2n} \pi \quad m=1, 2, \dots, n \quad (1.45)$$

不难看出, $x_m^{(n)}$ 不是均匀分布的而是凝聚在区间 $(-1, 1)$ 的两端.

(5) $T_n(x)$ 的全部极值点和极值. 因为 $T_n(x)$ 的极值点是 $T_n'(x) = 0$ 的根

$$T_n'(x) = n \sin(n \arccos x) / \sqrt{1-x^2} \quad (1.46)$$

故极值点是

$$x_m^{(n)} = \cos \frac{m}{n} \pi \quad m=1, 2, \dots, n-1 \quad (1.47)$$

它们都在区间 $(-1, 1)$ 内. 极值是

$$T_n\left(\cos \frac{m}{n} \pi\right) = \cos m \pi = (-1)^m \quad m=1, 2, \dots, n-1 \quad (1.48)$$

上式表明, $T_n(x)$ 的极大值为 1, 极小值为 -1. 在区间的端点

$$T_n(1) = 1 \quad T_n(-1) = (-1)^n \quad (1.49)$$

因此, 当 $x \in [-1, 1]$ 时, $T_n(x)$ 的值在 $[-1, 1]$ 振荡.

(6) 正交性. 切彼晓夫多项式族 $\{T_n(x)\}$ 在区间 $[-1, 1]$ 关于权函数 $\ominus / \sqrt{1-x^2}$ 正交, 即

$$\int_{-1}^1 T_n(x) T_m(x) / \sqrt{1-x^2} dx = \begin{cases} 0 & \text{当 } m \neq n \\ \pi & \text{当 } m = n = 0 \\ \frac{\pi}{2} & \text{当 } m = n > 0 \end{cases} \quad (1.50)$$

实际上, 用变换 $x = \cos \theta$, 上式化为

$$\int_0^\pi \cos n \theta \cos m \theta d\theta = \frac{1}{2} \int_0^\pi [\cos(m+n)\theta + \cos(m-n)\theta] d\theta = \begin{cases} 0 & \text{当 } m \neq n \\ \pi & \text{当 } m = n = 0 \\ \frac{\pi}{2} & \text{当 } m = n \neq 0 \end{cases}$$

(7) 任何一个 m 次多项式 $P(x)$ 都可以唯一的表示成 $T_0(x), T_1(x), \dots, T_m(x)$ 的常系数组合

\ominus 关于权函数的定义请见定义 1.5.