第一章 引 论

计算机代数是符号计算的一个主要分支，它所研究的对象是抽象的数学符号与概念，如整数、有理数、多项式、理想等．设计处理这些代数对象的算法，将其在计算机上有效地实施，并用以解决各种代数计算和推理问题是计算机代数学的中心课题．本章简要说明计算机代数的起源、发展和现状．

1.1 数学与计算

人类的原始活动如狩猎以及猎物的分配就已离不开数的概念，而人类赖以生存的三维世界又离不开各种形体．引进数 (自然数继而整数和有理数) 以及它们之间的四则运算，研究形 (点、线、圆、三角形等) 以及它们之间的相互关系导致了原始的数学．数学的诞生是人类生存和认识自然界的必然产物．这门处理和研究数和形的学科伴随着人类的进化，见诸于各种古代文明．它的发展满足了各种社会活动如观天测地和商品与货币流通的需要．毋容置疑，数学不能没有数和计算．

人类的文明不是停留在对生存的基本物质需求和实用上．人们认识的不断提高和对物质世界的探索导致了天文学、物理学和化学等．这些学科的建立和发展都离不开数学．同时，人们还需要物质之外的意识和精神生活，从而导致了古典哲学和美学．这些自然科学和社会科学的发展和进步给数学提出了更高的要求，使其不能停滞在简单的计算上．随之而来，数系统不断地得到扩充，从而有了实数和复数．数学家对这些复杂的数系统进行了深入的研究，探讨它们的结构、性质和其中的运算机制，并给出有效的计算方法．数学家不满足于数的基本运算，他们进而追求数学的优美和完善，探讨数系统的相容性和完备性，建立几何学、公理系统和逻辑推理机制，引进有理函数、三角函数和其他超越函数，创建和研究各式各样复杂的数学系统．如此一来便形成了众多数学分支和各种数学理论、体系与流派．但无论如何，计算是贯穿数学发展的一根主线．

1.1.1 存在性数学与构造性数学

存在性数学在于研究数学对象的存在性以及它们的性质和之间的相互关系，它是近代数学发展的主流．我们用下面的定理来说明它与构造性数学思想的差异．

定理 1.1.1 (代数学基本定理) 复数域上每个正次数的一元多项式总有一个复根.

读者不难从代数学教科书中找到该定理的存在性证明. 我们用 $\mathbf{Q}$ 表示 *有理数域*, $\mathbf{C}$ 表示 *复数域*, 而 $\mathbf{Q}[x]$ 和 $\mathbf{C}[x]$ 分别表示以 x 为变元、系数在 $\mathbf{Q}$ 和 $\mathbf{C}$ 中的一元多项式的全体. 上述定理表明, 对任意多项式 $F(x) \in \mathbf{C}[x]$, 其关于 x 的次数为 $n \geqslant 1$, 都存在一个复数 $\bar{x} \in \mathbf{C}$ 使得 $F(\bar{x}) = 0$, 即 $\bar{x}$ 是 F 的根. 存在性数学只关心复根 $\bar{x}$ 的存在性证明, 而不管如何求得 $\bar{x}$. 用这种存在性的证明, 我们还可以得出 "$F(x)$ 恰有 n 个复根" 的结论. 这一结果给出了复数域上一元多项式的基本性质, 非常优美, 且有诸多应用, 因而被称为 *代数学基本定理*. 至于如何具体求出 F 的复根, 那已不是存在性数学所关心的问题.

与之不同, 构造性数学所关心的则是如何具体求出 F 的根. 显然, 根求出之后, 它们的存在性便不证自明. 从构造性的角度, 我们可以提出一系列的问题. 譬如, 如何判断任给多项式 $F(x) \in \mathbf{Q}[x]$ 是否有有理根和实根? 如何求出 F 的所有有理根? 如何隔离 F 的所有实根? 如何计算 F 的所有复根? 依据 Galois 理论, 高次代数方程的解不一定能用根式表示. 但我们可以问, 如何判断任一给定多项式 $F(x) \in \mathbf{Q}[x]$ 的根能否用根式表示? 如果能的话, 又如何具体求出那些根式表示? 尽管这些问题大多有经典的解法, 但就计算效率而言它们仍是当今正在研究的课题.

数学作为一门基础科学, 服务于人类社会, 自然不应背离社会发展的实际需要. 构造性为数学的实用提供了直接保证. 我们并不低估存在性数学的重要价值和意义, 但过分追求数学的理论发展与优美而不顾其实用性似乎有所偏颇. 实际上, 数学的存在性与构造性之间的争论早已针锋相对、由来已久. 我国古代数学的发展向来以构造性为特色, 重方法和实用. 这种中国特色的构造性数学正在当代数学家吴文俊 [44] 的倡导下走向复兴. 构造性数学与存在性数学本应并驾齐驱、相得益彰. 但无论如何, 存在性数学在 20 世纪的数学发展中占据了主导地位.

构造性数学退居其次是有其特定根源的. 即使是很基本的数学对象 (如四次代数方程的根式解), 它们的构造往往都是非常复杂和困难的. 传统的纸笔演算需要消耗数学工作者的大量时间和精力, 因而使其无法从事更具创造性的工作. 随着数学的纵深发展, 与之相应的推理和计算也愈来愈复杂. 有些大型计算, 一个数学家即使花费毕生精力也许都难以完成, 而且其中的大量计算又机械乏味. 在这种情况下, 数学家就不得不有所取舍, 放弃对数学对象的具体构造而去研究它们的存在性.

先进工具的发现总会导致人类文明的重大变革. 现代电子计算机的出现与飞速发展正在改变着我们的工作和生活方式. 而对于与计算机科学和技术紧密相关的数学, 它的发展和进步自然面临着前所未有的机遇和挑战. 如何充分地使用计算机来进行数学的演算、推理、研究、教学和应用? 又如何让数学为计算机科学和技术的发展最大限度地发挥作用?

1.1.2 符号计算与数值计算

顾名思义，计算机是用于计算的机器．有了这样的机器，数学的计算问题便迎刃而解了．当然，事情并没有那么简单，有许多理论问题和实际问题需要我们去考虑和解决．我们将数学计算分为两种：符号计算与数值计算．数值计算是指浮点数如 0.618 和 3.1415926 之间的运算．在现行的计算机编程语言如 C 和 Java 中很容易实现确定精度的浮点运算．比如，使用双精度浮点 (double float) 运算，有

$$3.1415926^5 = 306.0196586846171.$$

显而易见，这样的计算是带有误差的．通常一个数值计算问题的解决需要通过很多步浮点运算来完成，因而有累计误差．在使用数值计算时还需要考虑算法的稳定性，即输入数据的微小挠动是否会引起输出的大幅波动．在使用数值逼近时，又需要考虑逼近是否收敛以及收敛的速度．这些都是数值分析研究的问题，不属本书讨论的范围．

与数值计算不同，符号计算所处理的对象是具有含义的抽象符号，主要研究如何进行这些符号之间的精确运算，因而没有误差．这些符号可以是整数、数学常数、有理函数、多项式理想，也可以是几何图形、逻辑公式和计算程序．本书仅限于介绍代数对象之间的运算．让我们观察下面的计算：

$$\frac{3\,\pi^2 - 27}{\pi + 3} = 3\,\pi - 9, \tag{1.1.1}$$

$$\frac{3 \times 3.1415926^2 - 27}{3.1415926 + 3} \simeq 0.4247777995. \tag{1.1.2}$$

(1.1.1) 式中的计算是精确的符号计算，没有误差，而 (1.1.2) 式中的计算是数值计算，带有误差．前者是通常的数学推导，将分式简化．它的数学意义颇为明显．将 $\pi = 3.1415926$ 代入 (1.1.1) 式的右边，我们有

$$3 \times 3.1415926 - 9 = 0.4247778.$$

由此可见，使用简化后的表达，不仅计算简单而且结果也更加精确．

符号之间的运算可以非常复杂．如何在电子计算机上表示和处理符号对象，设计和实施用于符号计算的有效算法和软件系统便是符号计算的主题．它们无疑会为科学与工程中的各种数学计算问题提供强有力的工具．

读者也许会问，那如何进行实数和复数之间的精确计算呢？数学上，无理数是无法用数字来精确表示的．比如，π 是一个无理数，但 π 本身只是一个符号．这个符号被赋予了特定的含义，并满足若干性质如 $\sin\pi = 0$．数值 3.14 是 π 的一个近似值，而 3.14159265358979 也只是 π 的一个近似值，它们都不等于 π．因而谈论无理数之间的精确计算是没有意义的．但是，我们可以进行实数和复数表示之间的精确计算．(1.1.1) 式就是这种计算的一个实例．

我们不一定能给出一个实数或复数的精确表示，但我们能给出该实数或复数位于的区间．区间可以用有理数表示，因而是精确的，而且区间的长度可以任意小．这种区间表示能使我们有效地处理实数和复数之间的计算 (参阅第八章)．

1.2 计算机代数简介

代数学是数学的一个基本分支，是其他数学分支的基础．它所处理和研究的数学对象是抽象的代数符号与概念，如整数、有理数、多项式、理想等．计算机代数是以计算机为工具、处理研究代数对象的一门新兴学科．它是符号计算的一个主要分支．代数算法的设计、分析、实现及应用构成了计算机代数的主要研究内容．我们将计算机代数所研究的一些基本问题列举如下．

- 大整数、有理数、任意精度的浮点数的表示和基本运算 (加、减、乘、除、幂等)
- 大整数的因子分解和最大公因子计算
- 多项式的表示和基本运算 (加、减、乘、除、伪除等)
- 多项式的因子分解和最大公因子计算
- 有理函数、根式和三角函数的基本运算与化简
- 线性代数计算 (向量和矩阵运算、行列式计算、线性方程组求解等)
- 非线性代数方程和 (半) 代数方程组求解
- 多项式理想的基本运算 (并、交、商、饱和等)
- 实根隔离和实闭域的量词消去
- 代数计算软件和算法的设计与实施

本书以后各章将对其中大多数问题予以讨论．

代数计算冗长繁复，常常让人望而生畏．传统的纸笔演算耗时、费力又易出错，因而不可能用于大规模的计算．现代计算技术为大型符号计算提供了条件．于是如何将基本代数理论算法化、精确化、效率化，如何将有效的算法在计算机上有效地实施，建立完整易用的软件系统，并用来处理形形色色的代数计算都是需要研究的问题．对这些问题的研究便形成了计算机代数这门学科．

计算机代数的发展始于 20 世纪 60 年代初期．其标志是美国 J. Slagle 在 1961 年用表处理语言 Lisp 所写的第一个自动符号积分程序 SAINT. 随后, 几个

基于 Fortran 和 Lisp 的符号计算系统, 如 FORMAC, ALPAK, PM, MATHLAB 等, 相继出现. 这些早期的系统主要是在美国的麻省理工学院、贝尔实验室和 IBM 公司研制开发的. 不难想象, 计算机代数软件系统的开发始终刺激和左右着代数算法的研究. 我们将在 1.4 节中简单介绍一些主要的计算机代数系统, 并在 1.6 节中以 Maple 为例演示部分典型的代数计算.

随着计算机代数系统的流行和广泛使用, 计算机代数的研究也越来越活跃. 国际计算机协会 (Association for Computing Machinery) 支持的学术研讨会 International Symposium on Symbolic and Algebraic Computaion (ISSAC) 每年夏天在欧美和其他国家或地区举行. 在亚洲各国举办的 Asian Symposium on Computer Mathematics (ASCM) 也在逐渐成为国内外计算机代数工作者学术交流和报告研究成果的一个园地. Journal of Symbolic Computation 是发表计算机代数方面研究成果的主要国际期刊. 读者从最近出版的《计算机代数手册》[16] 中可以查到各种有关资料.

1.3 理论、算法与实施

计算机代数的基础是 *构造性* 代数, 辅之以逻辑与算法理论. 这里重点是代数对象的构造而非存在性证明, 并且数学的理论和方法需要更加严密, 而逻辑学与算法理论则有助于将这些数学理论和方法形式化和算法化.

计算机代数中最基本的问题是 *算法设计*, 即依据已有的或者发展新的数学理论, 提出有效的方法, 将这些方法描述为适合实施的 *算法*, 并证明所设计算法的 *正确性* 和 *终止性*. 将这些算法优化并研究其效率 (所需要的计算时间和计算机存储) 便是算法分析的内容. 算法分析的主要方式包括算法的复杂性分析和实验测试.

算法实施 是指将具体的算法翻译为某种特定计算机语言中的程序, 以便这些算法能在计算机上运行. 这里的翻译需要正确、有效.

例 1.3.1 计算非负整数阶乘的算法可以描述如下.

算法 Factorial: $m := \text{Factorial}(n)$. 任给整数 n, 本算法计算 n 的阶乘 m (即 $m = n!$).

F1. 若 $n < 0$, 则指出 “非法输入” 且程序终止.

F2. 若 $n = 0$, 则命 $m := 1$; 否则, 计算 $m := n \cdot \text{Factorial}(n-1)$.

这一算法可以在不同的程序语言中实施. 例如用 Maple 语言, 我们可将其翻译为下列程序.

```
Factorial := proc(n)
     if n < 0 then return(‘Invalid input‘)
```

```
        elif n = 0 then 1
        else n*Factorial(n-1)
        fi
    end:
```

在以后的章节中，我们将会看到各种代数计算算法．它们大多需要调用其他用于基本运算的子算法．我们可以用低级编程语言如 C 或 C++ 来实现这些算法及其子算法，也可以利用计算机代数系统中的高级编程语言和已有的函数来实现它们．按前者实施的算法通常效率高，但编程工作量很大；按后者实施的算法效率较低，但程序容易编写．编程语言的选取、程序员的编程技巧和对算法的理解都对所实施算法的运行效率有很大影响．

1.4 计算机代数系统

计算机代数系统是用于代数计算的计算机软件．设计和实施一个理想的计算机代数系统是长期、复杂的高技术项目，牵涉到计算机软件工程．设计者需要了解计算机硬件和软件的最新发展，预测其未来走向，因而选取适当的基础编程语言，确定软件的特征和开发步骤，需要平衡软件的效率、界面、实用性、易用性、兼容性、可扩展性、开发时间和人力资源等诸多因素．实施者需要有一定的计算机代数知识和编程技巧，顾及程序的结构、可读性和易改性，提供程序的说明文本，并能与其他实施者协调合作．

早期 (1980 年之前) 出现的计算机代数系统基本上都是基于 Lisp 和 Fortran 两种程序语言．这些系统的主要功能是处理多项式和有理函数，用于有关符号和物理计算，其中大多数系统都是在美国研制开发的．我们不再介绍那些已经过时了的软件系统，而只将部分延续至今的计算机代数系统图示如下：

PM $\longrightarrow$ SAC-1 $\longrightarrow$ SAC-2 $\longrightarrow$ SAC/ALDES $\longrightarrow$ SACLIB

Reduce $\longrightarrow$ Reduce 2 $\longrightarrow$ Reduce 3

Scratchpad $\longrightarrow$ Scratchpad II $\longrightarrow$ Axiom

muMATH $\longrightarrow$ Derive

除 Axiom 和 Derive 外，现行的计算机代数系统大多基于 C 语言．这些系统的功能都极为丰富，其中有些是为特殊目的开发的，而那些一般性系统的功能已远远超出了计算机代数的范围．它们可以从事各种符号、数值和图形计算，并提供简单易用的高级编程语言．我们列出部分流行的计算机代数系统及其网址如下．有兴趣的读者可据此获取更多的资讯．

- Aldor (http://www.aldor.org/)
- Axiom (http://arch.axiom-developer.org/)

- CoCoA (http://cocoa.dima.unige.it/)
- Derive (http://www.derive.com/)
- Macaulay 2 (http://www.math.uiuc.edu/Macaulay2/)
- Macsyma (http://www.scientek.com/macsyma/mxmain.htm http://www.gosw.com/gosw/MacsymaInc/Macsyma422UxLx.html)
- Magma (http://magma.maths.usyd.edu.au/magma/)
- Maple (http://www.maplesoft.com/)
- Mathematica (http://www.wolfram.com/products/mathematica/)
- Maxima (http://maxima.sourceforge.net/)
- MuPAD (http://www.mupad.de/)
- Reduce (http://www.uni-koeln.de/REDUCE/)
- Risa/Asir (http://www.asir.org/)
- Singular (http://www.singular.uni-kl.de/)

在本书的附录 A 中，我们将再次提到这些广为流传的计算机代数系统，并介绍它们的若干基本功能.

1.5 问题及应用举例

在这一节里，我们用几个简单的例子来说明计算机代数中的一些基本问题. 也许这能让读者对计算机代数到底能做什么有个粗略的印象.

1. 大整数运算. 例如

$$31!/2^{32} = 1225298442569065513867968 75/64.$$

2. 整数因子分解. 如何有效地将任给正整数分解为素数的乘积. 例如

$$20031020 = 2^2 \times 5 \times 1001551,$$

其中 1001551 为素数.

3. 多项式的最大公因式. 如何有效地计算两个整系数多项式的最大公因式. 例如,

$$\begin{aligned}&\gcd(80\,x^2y^2 + 72\,x^3y^3 + 80\,xy^3 + 72\,x^2y^4 - 160\,zxy^2 - 144\,zx^2y^3,\\ &\qquad -99\,xz^3 - 85\,x^6 - 99\,yz^3 - 85\,yx^5 + 198\,z^4 + 170\,zx^5)\\ &\quad = x + y - 2\,z.\end{aligned}$$

4. *多项式因子分解*. 如何有效地将任给多项式在某一给定的数域上分解为不可约因子的乘积. 例如, 在有理数域上有

$$x^8-y^8=(x-y)\,(x+y)\,(x^2+y^2)\,(x^4+y^4),$$

而在有限域 $\mathbf{Z}_3$ 上有

$$x^8-y^8=(2\,y+x)\,(2\,y^2+2\,yx+x^2)\,(2\,y^2+yx+x^2)\,(x+y)\,(x^2+y^2).$$

5. *多项式理想的准素分解*. 如何有效地将任意给定的一组多项式生成的理想分解为准素理想的交. 例如

$$\langle x^2+y,x^4y-x^3\rangle=\langle x^2-x+1,y+x-1\rangle\cap\langle x+1,y+1\rangle\cap\langle x^3,x^2+y\rangle,$$

这里 $\langle P_1,\cdots,P_s\rangle$ 表示多项式 $P_1,\cdots,P_s$ 生成的理想.

6. *多项式的正定性*. 判定任给多项式是否 (半) 正定. 例如

$$(\forall x,y)\,[x^6-x^4y^2-x^2y^4+y^6-x^4+3\,x^2y^2-y^4-x^2-y^2+1\geqslant 0]$$

成立.

7. *解代数方程组*. 这是计算机代数, 也是数学中的基本问题.

例 1.5.1　求代数曲线

$$F=xy\,(x+y-3)^2=0$$

的临界点. 为此, 计算

$$F_1=\frac{\partial F}{\partial x}=y\,(x+y-3)^2+2\,xy\,(x+y-3),$$
$$F_2=\frac{\partial F}{\partial y}=x\,(x+y-3)^2+2\,xy\,(x+y-3).$$

用计算机代数系统解多项式方程组 $F_1=0$, $F_2=0$ 可得三组解:

$$\{x=0,\,y=0\},\quad \{x=3/4,\,y=3/4\},\quad \{x=3-y,\,y=y\}.$$

计算机代数方法和系统还可以用于几何定理的*机器证明*, 参数曲线和曲面的*隐式化*, 微分方程求解等诸多几何和分析中的计算问题.

1.6 代数计算演示

最后, 我们将在 Maple 系统中进行的一些代数计算复制下来供读者参考. 这些演示性的计算用来说明计算机代数系统的部分功能、用户界面和易用性. 我们不再一一解释进行的计算以及所用的指令和函数 (见图 1.6.1, 图 1.6.2 和图 1.6.3). 读者应该不难琢磨出它们的含义.

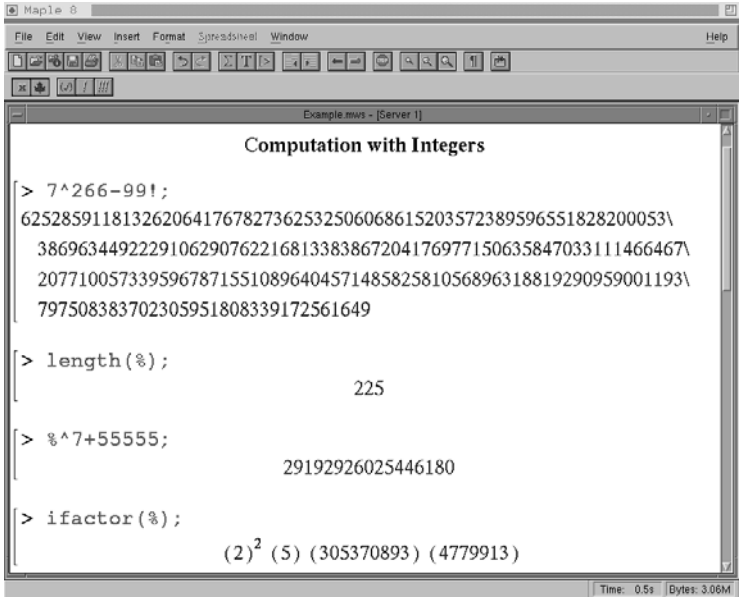

图 1.6.1 Maple 中的整数运算

```
> (b*x+a*b-2)^3*(b*x^2+a*x-2*b);
```

$$(b\,x+a\,b-2)^3\,(b\,x^2+a\,x-2\,b)$$

```
> expand(%);
```

$$\begin{aligned}&3\,b^4\,x^3\,a^2+3\,b^3\,x^3\,a^2-12\,b^3\,x^3\,a-6\,a^2\,b^3\,x^2-12\,b^2\,x^2\,a^2+16\,b\\&+a^3\,b^4\,x^2+12\,b\,x^2\,a+a^4\,b^3\,x-6\,b^4\,x\,a^2-6\,b^2\,x^3\,a-6\,b^4\,x^2\,a\\&+3\,b^4\,x^4\,a-6\,a^3\,b^2\,x+12\,a^2\,b\,x+3\,b^3\,x^2\,a^3+b^3\,x^4\,a+12\,a\,b^2\,x^2\\&+24\,b^3\,x\,a-8\,b\,x^2-8\,a\,x+b^4\,x^5-2\,b^4\,x^3-6\,b^3\,x^4+12\,b^3\,x^2\\&+12\,b^2\,x^3-24\,b^2\,x-2\,a^3\,b^4+12\,a^2\,b^3-24\,a\,b^2\end{aligned}$$

```
> factor(%);
```

$$(b\,x+a\,b-2)^3\,(b\,x^2+a\,x-2\,b)$$

图 1.6.2 多项式运算

```
> solve(%,x);
```

$$\frac{-a+\sqrt{a^2+8\,b^2}}{2\,b},\frac{-a-\sqrt{a^2+8\,b^2}}{2\,b},-\frac{a\,b-2}{b},-\frac{a\,b-2}{b},-\frac{a\,b-2}{b}$$

```
> %[2];
```

$$\frac{-a-\sqrt{a^2+8\,b^2}}{2\,b}$$

```
> subs(b=a,%);
```

$$\frac{-a-\sqrt{9}\,\sqrt{a^2}}{2\,a}$$

```
> simplify(%);
```

$$-\frac{1}{2}-\frac{3}{2}\,\mathrm{csgn}(a)$$

图 1.6.3 方程求解与代数化简

Maple 还有诸多其他功能，如矩阵运算、极限计算、符号微分与积分、求和与求积、微分方程求解、图形视化、文本处理等. 对此我们不一一介绍. 读者可参阅本书的附录 A 和有关 Maple 的图书资料.

本书包含了讲义 [40] 中第一章的全部内容和第三章的部分内容. 它的编写参照了著作 [15, 42, 39, 41] 中的有关章节. 读者在阅读本书时还可以参考文献 [11, 12, 30, 38] 和多部与之有关的中文著作 [17, 25, 43, 45, 48, 50, 51, 52]. 经典名著 [21, 35, 36, 37] 是书中代数基础知识和算法代数的主要参考文献.

习 题

1. 给出一个将任意正整数分解为素数之积的算法. 用该算法将 12, 123, 1234, 12345, $\cdots$ 分解为素数的乘积.

2. 判定 4294967311 是否为素数.

3. 求多项式

$$x^4-5\,x^2+12\,x-2\,x^3-4 \quad \text{和} \quad 3\,x^3-13\,x^2+25\,x-22$$

的最大公因子.