

Maple

计算机代数系统 应用及程序设计

李世奇 杜慧琴 编著



华航Z0195092



重庆大学出版社

Maple 计算机代数系统应用及程序设计

李世奇 杜慧琴 编著

重庆大学出版社

内 容 提 要

本书介绍当今国际上广泛使用的计算机代数系统:Maple。该数学软件系统的主要功能包括符号处理、数学图形描绘和数值计算等方面,处理的数学问题包括初等数学、高等数学、线性代数以及运算微积、微分方程、图论、数论、群论、组合数学、概率统计等方面;凡是公式能表达和解决的问题,Maple 都能进行符号运算和数值计算,它是数学工作者、科技工作者、大中学校师生的强大数学辅助工具。本书通过对 Maple 的讲解和大量实例,帮助读者了解 Maple 的概念、使用方法和程序设计。本书可作为各级各类学校的学习 Maple 以及数学有关课程的教科书和参考书,又可作为科研、工程技术人员进行数学计算和公式推导的参考书,也可为读者自学之用。

图书在版编目(CIP)数据

Maple 计算机代数系统应用及程序设计/李世奇,杜慧琴编著. - 重庆:重庆大学出版社,1999.4

ISBN 7-5624-1923-X

I . M… II . ①李… ②杜… III . 数学 - 应用软件 - 程序设计 IV . 01

中国版本图书馆 CIP 数据核字(1999)第 06712 号

Maple 计算机代数系统应用及程序设计

李世奇 杜慧琴 编著

责任编辑 肖顺杰

*

重庆大学出版社出版发行

新华书店经销

四川外语学院印刷厂印刷

*

开本:787×1092 1/16 印张:20.5 字数:512 千

1999 年 5 月第 1 版 1999 年 5 月第 1 次印刷

印数:1—4 000

ISBN 7-5624-1923-X/TP·207 定价:25.00 元

序

随着计算机软件技术的发展,各种数学软件相继出现并不断完善,这一过程已经持续 30 多年了。早在 60 年代就出现了通用的数学软件包 MACSYMA,它是一个巨型系统,有大量的辅助软件包,功能很强,因对数学计算起过重要作用而名噪一时。之后又出现过通用型数学软件包 REDUCE,它是基于高能物理的软件包扩充而成的,也在数学软件包世界中充当过重要角色。在这种趋势推动下,一些针对不同任务的专用数学软件包也不断涌现,如 60 年代发展起来的符号积分系统,用于高能物理计算的 Schoonship 系统,用于天体力学的 CAMAL,用于代数几何的 MACAULAY,用于交换代数的 COCOA,用于群论的 CAYLEY,用于李群李代数的 EDLiA,用于数论的 PARI,用于数理统计的 SAS,用于优化计算的 LinDo 等系统。当前,在世界范围内比较流行的数学通用软件包当首推: Mathematica, Maple, Matlab 等等,它们功能更强,能适用于各种不同的机型,对数学计算,计算机辅助教学(CAI),计算机辅助设计(CAD),乃至数学推理的机器证明等都起到了很大的推动作用。

数学软件包的主要作用自然是数学计算。但是数学计算一词在许多场合都被狭隘地理解为数值计算,而忽略数学计算的另一部分即“符号计算”。这是应当予以纠正的。符号计算又称为符号代数计算或符号代数,正式名称是“计算机代数”。实现计算机代数的软件系统称为“计算机代数系统”。

Maple 是当今流行的计算机代数系统之一,它是由加拿大滑铁卢大学于 1980 年研制成功的,并以加拿大国树——枫树(maple)命名。目前,国内的 Maple 用户很多,不但涉及数值计算,计算机辅助教学,并引入到定理机器证明中,影响很大,但尚无一本正式介绍 Maple 的专著。作者正是适应这种需要而编写此书的。作者集多年从事计算机程序设计、计算数学方面的教学经验,对 Maple 的最新版本的 Maple V R5 作了全面系统的介绍:着重对 Maple 的数值计算,符号演算,二维、三维图形描绘等三大功能作了阐述,介绍了 Maple 在其他数学领域的功能;对 Maple 程序设计语言的特色及应用,以及如何扩展 Maple 的功能作了比较系统的说明。该书可作为 Maple 的专用教材,也可用于数学实验指导(如数学建模活动中的计算机软件的参考书),还可为从事工程技术的人员参考。

当前国家教育部已正式开展面向 21 世纪的教学改革的一系列研究,其中“数学实验”将作为迎接 21 世纪高等学校数学教学的一门新课。我相信,本书的出版将有助于数学实验的开设,有助于我国数学教学改革事业的发展与提高。同时,Maple 作为一种数学软件平台也将对我国高科技发展作出贡献。希望本书早日出版面世,以飨读者。

杨万年

1999 年元月 10 日于重庆大学

目 录

第一章 概述	(1)
1.1 计算机代数与计算机代数系统	(1)
1.2 Maple 计算机代数系统	(2)
1.3 Maple 工作页	(21)
第二章 Maple 基础应用	(25)
2.1 算术计算	(25)
2.2 符号计算	(29)
2.3 方程求解和多项式	(36)
2.4 微积分	(42)
2.5 线性代数	(47)
2.6 Maple 作图	(52)
2.7 Maple 系统结构和软件包	(72)
第三章 初等数学	(74)
3.1 数的计算	(74)
3.2 初等代数	(88)
第四章 函数与方程	(117)
4.1 函数	(117)
4.2 方程	(128)
第五章 微积分	(140)
5.1 极限	(140)
5.2 级数	(143)
5.3 微分	(150)
5.4 积分	(158)
5.5 微分方程	(170)
第六章 线性代数	(180)
6.1 线性代数软件包 linalg	(180)

6.2 向量与矩阵定义	(183)
6.3 向量计算	(193)
6.4 矩阵计算	(197)
第七章 Maple 作图	(228)
7.1 常用作图函数	(228)
7.2 作图软件包 plots	(237)
7.3 软件包 plottools	(258)
第八章 Maple 程序设计	(262)
8.1 Maple 语言特点	(262)
8.2 Maple 类型	(263)
8.3 Maple 程序设计	(265)
8.4 程序调试	(285)
8.5 输入与输出	(293)
8.6 扩展 Maple 功能	(295)
8.7 Maple 程序实例	(299)
附录	(304)
附录一 Maple 常用数学函数表	(304)
附录二 Maple 软件包	(306)
附录三 Maple 标准库函数	(307)
参考文献	(321)

第一章 概 述

计算机进行数学符号演算,意味着用键盘和显示器代替传统的纸和笔进行数学运算,这就是所谓的“计算机代数”;进行数值、表达式、方程、公式等数学符号运算的计算机程序系统,称之为“计算机代数系统”。对于从事现代科技、知识经济的科技工作者,对于培养具有现代知识结构人才的教育工作者,以及对于正在学习和使用数学的大、中学生乃至小学生,计算机代数系统对他们来说,都是不可缺少的重要的辅助手段;总之,对于每一个进行数学运算的人,计算机代数系统无疑是一个极其强有力的工具。

1.1 计算机代数与计算机代数系统

1.1.1 计算机代数

“计算”一词在过去仅仅是数值计算的意思,现在已经远远超出其含义。人们现在都知道:计算机所进行的“计算”,不仅仅是数值计算和数据处理,而且还可以是非数值计算。计算机的诞生缘于数值计算,而且计算机在相当一段时间里的应用也主要是进行数值计算,但是人们更希望计算机能进行数学符号运算。数学意义上的计算,有数值计算和符号计算两大类。对于数值计算,指的是计算机处理的数据和处理后的结果是数值,这种方式就是数值计算方法;而对于符号计算,计算机处理的数据和处理后的结果是符号,这种方式就属于计算机代数的范畴。所谓符号,可以是字母、公式,也可以是数字。因此,在符号计算中仍然可以进行数值计算,数值计算可以看成是符号计算的特例。符号计算实际上就是数学演算、数学推理和数学证明,通过计算机来完成这些演算、推理和证明,就是计算机代数。

计算机代数也能进行数值计算,但是与数值计算方法还是有相当大的差异:数值计算方法是近似计算,而计算机代数对于数值计算是精确计算,二者相互不能替代,两种方法都有存在的必要。

计算机代数作为一门新兴的学科和数学工具,使人们认识数学和解决数学问题的方法发生了深刻的变化。解决数学问题的过程中,建立数学模型、选择计算公式等工作离不开符号演算和描画函数图形;这许多繁杂的符号演算由计算机来完成显得更简单明了,许多复杂的图形由计算机来描绘显得更准确生动;总之,许多数学问题由计算机来完成显然效率倍增,而且可以避免人为的错误。使用计算机代数来处理数学问题,人们的注意力着重在分析问题、解决问题的本质方面,利用工具摆脱了具体的机械的工作细节,辅助人们做更加富有创造性的智能化的工作。早在 1671 年,著名的数学家莱布尼兹(Leibniz)说过:“让一些杰出的人才像奴隶般地把时间浪费在计算上是不值得的。”他渴望有朝一日能有计算机把科学家从这种奴隶般的计算中解放出来,这个愿望可以说由于计算机代数的诞生和应用才真正实现了。现在,人们已经利用计算机代数发现、验证、证明和解决了许多许多的数学问题。计算机代数这门学科还在发展

中,虽然它还不可能解决目前的所有数学问题,但是它的符号计算方法和能力已经显现出了巨大的威力。

1.1.2 计算机代数系统

计算机代数系统是计算机软件系统之一,是根据计算机代数原理编写的程序系统。今天,计算机代数系统已经得到广泛应用。目前,世界上著名的计算机代数系统有:Maple、Mathematica 和 MACSYMA 等。

计算机代数开始于五六十年代。70 年代人们开发了几个较为成功的通用的计算机代数系统,如 REDUCE 系统和 MACSYMA 系统。尽管它们能解决一些数学计算问题,但是早期的计算机代数系统大部分是用 LISP 语言开发的,运行速度较慢,用户再开发也不方便,而且几乎都在大型计算机上使用,将其使用在微计算机上体现不出计算机代数系统的优越性。现在使用的上述三种计算机代数系统的基本系统都是用 C 语言开发的,因此可以较容易的移植到各种计算机和运行环境,目前广泛使用的微计算机(486/586/Pentium)都可以运行计算机代数系统,程序运行速度也大为提高。一般,计算机代数系统都有 DOS 和 Windows 两种版本。Windows 上运行的版本,用户界面和使用方式更为方便。

1.2 Maple 计算机代数系统

1.2.1 关于 Maple

1.2.1.1 Maple 功能

国际流行的计算机代数系统 Maple,是一个优秀的符号运算软件系统,它提供了范围广泛的数学计算功能,主要包括三个方面:符号演算、二维和三维图形描绘以及数值计算。Maple 能解决如下问题:

- 初等数学中的有理数运算(精确计算和大有理数计算),代数式运算,初等函数变换,映射与集合运算,方程求根等
- 复数运算,向量代数,平面和空间解析几何,线性代数中矩阵的各种运算和变换,线性方程组求解
- 微积分中的函数的极限,导数,积分,泰勒级数、傅立叶级数展开,以及数列与级数求和,求积等
- 微分方程的解析解,数值解以及通过方向场图形得出近似解
- 二维和三维作图

此外,Maple 还提供了解决诸如图论、数论、群论、组合数学、概率统计、特殊函数(运算微积)和其他应用数学(如金融数学)方面的问题的软件包。总之,凡是能用公式解决的问题,Maple 都能解决;它能有效地辅助人们在数学计算方面的繁琐事务,进而去从事更加智能化和深层次的问题的研究。

1.2.1.2 Maple 用户环境

Maple 是加拿大 Waterloo 大学符号计算小组于 1980 年开始研制,经过多年的应用和开发,

至今发展到 Maple V Release 5。本书介绍的就是 Maple 最新版本 Maple V R5。据称,现在 Maple 用户在全世界近千万。Maple 是一个开放的适用于各种不同的计算机环境的计算机代数系统,它含有三部分:用户界面 Iris,基本部分(内核)Kernel 和外部库 Library:

- Iris — 语法分析(对用户的输入进行分析),主要功能:
 - 表达式显示(按照书写形式),图形处理,特殊用户界面
- Kernel — 翻译器(将用户输入翻译成机器代码),主要功能:
 - 内存管理
 - 整数域 Z , 有理数域 Q , 实数域 R , 扩展有理数域 $Q[x]$ 的计算和处理
- Library — 外部库函数,包含应用软件包和在线帮助

用户界面 Iris 和基本部分 Kernel 是 Maple 的主体,由 C 语言编写而成,用户启用 Maple 时,就将它们调入计算机,直接可以执行。外部库 Library 的库函数由 Maple 语言编写而成,在用户需要时才调入计算机。

Maple 是一个交互式的系统,系统界面十分友好。熟悉 Windows 操作系统的用户使用 Maple 没有任何困难。DOS 操作系统下的 Maple 属于较低的版本。

Maple 的所有操作通过用户输入 Maple 命令来实现。每一条命令实际上是 Maple 的一个函数,因此 Maple 本质上是一个函数系统。Maple 的函数现在已经超过 3000 个,随着版本的不断升级,提供的函数还会更多,功能还会更强。表达式是 Maple 函数的操作对象。正确构造 Maple 表达式,熟悉和理解 Maple 函数的功能,懂得 Maple 处理表达式的基本算法,是掌握和运用 Maple 的关键。

Maple 提供具有良好结构的程序设计高级语言,有可供选择的数据结构:函数、序列、集合、数组、表、记录表等等,有用于符号处理、类型测试的函数,用户可以根据自己的用途和需要,方便地定义函数和设计自己的软件包,扩展 Maple 的功能。

使用 Maple 的过程中如有问题,可以通过“在线帮助”获得问题的解答。任何时候都可以通过 Maple 系统的 Help,找到解决问题的具体函数的用法。对于初学者,Maple 系统的 Help 中,提供“新用户巡礼”(New User's Tour),使之尽快了解和掌握 Maple 的使用。

1.2.1.3 Maple 资源

Maple V R5 软件有两张 CD-ROM 盘(或若干软磁盘)。

Maple V R5 要求的软硬件环境:

- Intel486DX, Pentium(奔腾)或其它合适的处理器;486SX 需要协处理器
- 32MB 的硬盘自由空间
- 8MB 内存(RAM)
- Microsoft Windows 3.1x, Windows 95 或 Windows NT4.0

Maple V R5 安装成功后,其资源(以图标形式出现)有:

- Maple V Release 5 — wmaple.exe 是启动运行 maple 文件(鼠标双点此图标)
- Maple V Release 5 Parallel Server(除 Windows3.1x 外)
 - wmaple.exe 是运行多用户模式文件
- Command Line Maple — mapledos.exe 是执行 Maple V R5 命令行文件
- Mint — mint.exe 是执行 Maple 语法检测文件

- March — march.exe 是执行管理 Maple 库档案文件
- Update Source — updtsrc.exe 是更新 Maple V R4 的 TEXT 文件与 Maple V R5 一致的工具文件
- ReadMe — readme.txt 介绍 Maple V R5 设备安装概要和最近变化
- Command Line ReadMe — cmdline.txt 提供 Command Line 的形式和选项
- Mint ReadMe — mint.txt 提供 Mint 的信息(含 Maple 代码的语法检测)
- March ReadMe — march.txt 提供 March 信息
- Update Source ReadMe — updtsrc.txt 提供 Update Source 信息
- Introduction — 在 Maple 的工作页提供最新的 Maple 修改情况

安装的文件有:

- afm — 子目录,所含文件为字体
- bin.wnt — 上述执行文件和文本文件
- etc — 子目录,含两个文件 Maple2e.sty 用于处理 Maple 工作页面以及随同 LaTeX 文件的输出,latex.txt 提供关于 LaTeX 文件的格式和打印的信息
- lib — 子目录,所含文件均为 Maple 库文件:
 - maple.hdb — Maple 的 help 数据库
 - maple.ind — Maple 库索引文件
 - maple.lib — Maple 库函数和软件包
- license — 子目录,含有认可信息
- update — 子目录,用于存储 Maple 库的更新(一般为空目录)

对于 Windows 3.1x,还有用于升级的文件:

- updt3to4.exe — Maple V R3 文本文件更新为 R4 文本文件的工具,同样也可用于更新为 R5 的文本文件

1.2.2 Maple 语言

启动 Maple 的方法与启动计算机的其它应用程序一样,在 Windows 上启动 Maple 是双点 Maple 图标,在 DOS 下则是在子目录 bin 下运行 Maple.exe 文件。Maple 同其他计算机代数系统一样,本质上是计算机语言。所以,使用 Maple 与使用通常的计算机语言无异。直接输入 Maple 命令行,Maple 就会马上给以相应的结果;如果将一些命令编成程序,Maple 则会逐条执行命令,给出程序相应的结果。

1.2.2.1 基本符号

Maple 语言的基本符号有字母、数字、记号和分隔符:

- 字母 — a, b, ..., z 和 A, B, ..., Z; 注意字母大小写有相当大的区别,希腊字母用 alpha, beta, Pi 等代替(屏幕显示是 α, β, π)
- 数字 — 0, 1, 2, ..., 9

- 记号 — Maple 记号有关键字(保留字),操作符,变量名,字符串,自然数和标点符号
- 关键字 — Maple 的关键字(见表 1.1)在 Maple 里有特殊的意义,不能作为变量名或改作其他用途

表 1.1 Maple 关键字表

关 键 字	用 途
if, then, elif, else, fi	— 条件语句
for, from, in, by, to, while, do, od	— 循环语句
while, do, od	— 当语句
proc, local, global, end, option, options, description	— 程序设计
read, save	— 读写语句
quit, done, stop	— 结束 Maple
union, minus, intersect	— 集合运算符
and, or, not	— 布尔运算符
mod	— 模数运算

- 保留字 — Maple 的保留字是指预先有特定意义的记号,如数学函数名 sin, abs 等;如函数名 diff, factor 等;如类型名 integer, list 等;不要改作他用
- 操作符 — Maple 的操作符按照结合方式,分为二元操作符(见表 1.2)和一元操作符(见表 1.3),空元操作符仅一种 %, %% , % % %

表 1.2 Maple 二元操作符表

操 作 符	用 途	操 作 符	用 途
+	— 加	<	— 小于
-	— 减	< =	— 小于等于
*	— 乘	>	— 大于
/	— 除	> =	— 大于等于
* *	— 乘方	=	— 等于
^	— 乘方	< >	— 不等
\$	— 序列	- >	— 箭头操作符
@	— 复合	union	— 集合并
@ @	— 再复合	minus	— 集合差
,	— 逗号,表达式分隔符	intersect	— 集合交
'	— 字符串定界符	and	— 逻辑与
.	— 小数点,点连接符	or	— 逻辑或
..	— 省略号,范围符	& +	— 非交换的加
mod	— 模数运算	: =	— 指定运算
&string	— 中性操作(string 是另一操作符,如 & *)	::	— 类型声明或模式限定


```

> L := [seq(i, i=1..10)];
                      L := [1,2,3,4,5,6,7,8,9,10]
> L[3]; # 表中第三个元素
                      3
> # 集合,注意:集合是用{}框起的,用符号$产生序列
> S := {x$x= 1..10};
                      S := {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
> S[9]; # 集合中第九个元素,注意:不能用S{9}
                      9

```

1.2.2.2 Maple 数

数是 Maple 处理数学问题的对象,是 Maple 语言的重要数据类型 `numeric`。

Maple 处理的数有整数,分数,浮点数(小数),有理数,无理数和复数等,还有一些表示常数的符号。Maple 将根式形式的无理数称为代数数。

Maple 处理的数:

- 整数 — 类型为 `integer`,有如下子类型:
 - `negint` — 负整数
 - `posint` — 正整数
 - `nonnegint` — 非负整数
 - `even` — 偶数
 - `odd` — 奇数
 - `prime` — 质数
- 分数 — 类型为 `fraction`,也是 `rational`,形式为两个整数的商,如 $2/3$
- 浮点数 — 类型为 `float`,浮点数一定是小数形式或科学记数法形式,如 $.5$ 和 $1e-8$
- 有理数 — 类型为 `rational`,注意:1 和 $1/2$ 都是有理数,而 0.5 不是有理数
- 代数数 — 类型为 `algunum`,注意:根式形式的数和分数为代数数,如 $\sqrt{2}$, $2/3$ 等,特别是 `RootOf` 定义的根形式的数如 $5/\text{RootOf}(z^2-2, z)$
- 复数 — 类型为 `complex`,复数的虚部单位用 `I` 表示(也可以自定义),如 $3+4*I$
- 常数集 — Maple 用符号定义的常数(见表 1.4)。常数集的名称为 `constants`,类型为 `symbol`;用户可以添加符号常数

表 1.4 Maple 常数集表

数学常数	Maple 名	近似值
π , 圆周率	Pi	3.141592654
C, Catalan 数	Catalan	0.915965594
γ , Euler-Mascheroni 常数	gamma	0.577215664
逻辑值真,假	true, false	
无穷大 ∞	infinity	

其中:

$$C = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}$$

$$\gamma = \lim_{n \rightarrow \infty} \sum_{k=1}^n \left(\frac{1}{k} - \ln(n) \right)$$

例 1.2 Maple 数示例

> # 一个大整数与其后的质数

> 12345678909876543210987654321; nextprime(%);

12345678909876543210987654321

12345678909876543210987654347

> 1/2, -3/56; # 分数

$$\frac{1}{2}, -\frac{3}{56}$$

> evalf(Pi, 100); # 计算浮点数, 求出 π 的 100 位

3.1415926535897932384626433832795028841971693993751058209749445923

07816406286208998628034825342117068

> 1.4e-8;

.14 10⁻⁷

> sqrt(2); # 代数数, 就是无理数

$$\sqrt{2}$$

> 5/Rootof(x^2-3);

$$5 \frac{1}{\text{RootOf}(_Z^2 - 3)}$$

> allvalues(%); # 求出上式表示的所有值

$$\frac{5}{3}\sqrt{3}, -\frac{5}{3}\sqrt{3}$$

> # Maple 常数集 (Maple V R3 还有自然对数的底 E)

> constants;

false, γ , ∞ , ture, Catalan, Fall, π

> sqrt(-1); # 复数单位 I 是 -1 的平方根

I

> (3+4*I)*(3-4*I);

25

> 1/(a-I);

$$\frac{1}{a-I}$$

> evalc(%); # 在复数域上的赋值运算

$$\frac{a}{a^2+1} + \frac{I}{a^2+1}$$

```

> # 测试数据类型
> 1234567890987654321: nextprime(%): # 结果见前面
> type(%, prime); # 1234567890987654321 不是质数
                                false
> type(%, prime); # 1234567890987654321 后的质数当然是质数
                                true
> type(1998, odd); # 1998 不是奇数
                                false
> type(0.5, rational); # 0.5 不是有理数
                                false
> type(0.5, float); # 0.5 是浮点数
                                true
> type(-3, complex); # -3 是复数
                                true

```

在这里,我们一再强调数的类型,是因为 Maple 对数的处理与数的类型密切相关;不同的类型采用不同的函数进行处理。数的类型与我们对数域的认识基本一致,类型 `numeric` 含有整数 `integer`,分数 `fraction`,浮点数 `float`,有理数 `rational` 等类型;这些又都是复数 `complex` 类型。不仅是 Maple 的数,对 Maple 所有的操作对象都存在类型问题。读者应注意各种类型在 Maple 里与我们平常理解上的细微差异,不要用我们惯有的概念来理解和使用 Maple,否则会出现一些小小的问题。比如,在 Maple 里数和式是截然不同的概念,处理方式就完全不同:

```

> type(100, square); # 测试 100 是否完全平方
                                true
> type(x^2+2*x+1, square); # 测试  $x^2+2x+1$  是否完全平方
                                false

```

关于类型测试见第八章。

1.2.2.3 Maple 变量

- 变量 — 类型为 `name` 和 `symbol`(或其它具体类型),一般命名原则是:字母开头,随后跟字母,数字,或下划线,或是其它符号,注意:
 - Maple 的保留字和函数名等一般不能作为变量名
 - 用反单引号框起来的任何字符串可作为变量名
 - 与其它高级语言不同,在 Maple 里变量名的定义限制较少:变量名的第一个字符可以为其它字符,如 `~` 或 `_`
 - 变量名的字符数没有限制
 - 变量名可以由连接函数 `cat` 或点连接符产生
 - 变量可以被赋予任何表达式,未赋值时变量的值就是本身
 - 变量未经指定,一律为全局变量 `global`
 - 变量 `name`、符号 `symbol` 和字符串 `string` 可以互相转换

例 1.3 Maple 变量示例

```

> '1' := 2; # 将 2 赋给变量名 '1'
1 := 2

> '1'; # 显示变量 '1' 的值
2

> _ := x -> x^2; # 定义一个函数名 _
_ := x -> x^2

> _(15); # 求函数_(15)的值
225

> cat('m', 'a', 'ple'); # 用函数 cat 产生变量名
maple

> type(maple, name);
true

```

Maple 变量也有下标变量:

例 1.4 下标变量示例

```

> x[i];
xi

> y[i,j];
yi,j

> a[1][2,3];
a12,3

> type(%, indexed); # 类型是下标变量
true

```

1.2.2.4 Maple 表达式

表达式是 Maple 语言的基本实体和操作对象,包括常量,变量,函数,序列,数组,表,集合和其他数据类型;用各种运算符(算术、逻辑、关系、范围、点连接符等等)和括号与种种数据类型构成的有意义的式子都是 Maple 表达式。由此可见,表达式的类型是十分丰富和复杂的,不是唯一的一种类型。

1. Maple 表达式的内部表现形式

对于一个表达式,Maple 建立一个表达式树。例如,表达式 $5\sqrt{a^3b} + \sin(x)^2 - 7e^{(2t)}\cos(\alpha) - 3$ 的表达式树见图 1.1:

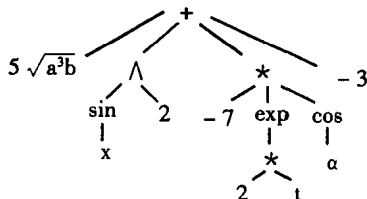


图 1.1 多项式 $5\sqrt{a^3b} + \sin(x)^2 - 7e^{(2t)}\cos(\alpha) - 3$ 的表达式树

在表达式树的第一个节点(根节点)是加号+,表示这个表达式的类型是'+'.对于+号,有四个分支,表示对+号有四个项;再分别为节点,找出各自的分支,得到各自相应的项;如此向下,直到没有了节点,剩下的就是表达式树的叶子。这里的叶子是不可再分的整数,变量等。图 1.1 中的单项式 $5 * \text{sqrt}(a^3 * b)$ 的表达式树请读者分析。

可以看出,表达式树的节点为操作符,叶子为不可再分的数,变量等操作数。对于表达式树的操作函数,称为原始函数,见表 1.5。

表 1.5 Maple 原始函数表

函 数 名	作 用
<code>type(f, t)</code>	对表达式 f 按类型 t 测试
<code>nops(f)</code>	返回 f 中的操作数的数目
<code>op(i, f)</code>	选出 f 中的第 i 个操作数
<code>subsop(i = g, f)</code>	将 f 中的第 i 个操作数用 g 替换

例 1.5 原始函数示例

> `f := 5 * sqrt(a^3 * b);` # 定义表达式

$$f := 5\sqrt{a^3 b}$$

> `type(f, '*');` # 测试类型(操作符)

true

> `nops(f);` # 计算 f 关于 * 号的操作数数目

2

> `op(f);` # 选出 f 关于 * 号的操作数

$$5, \sqrt{a^3 b}$$

> `op(0, op(2, f));` # 选出 f 的第二个操作数的上方节点类型(操作符)

> `op(op(2, f));` # 选出 f 的第二个操作数的操作数

$$a^3 b, \frac{1}{2}$$

> # 选出上式中第一个操作数 $a^3 * b$ 的上方节点类型(操作符)

> `op(0, op(1, op(2, f)));`

*

> `op(op(1, op(2, f)));` # 选出操作数 $a^3 * b$ 的操作数

$$a^3, b$$

> `subsop(1 = y, 2 = z, f);`

yz

从此例可以看出,表达式的类型是由表达式树的节点符号(操作符)所决定的,这个符号可由函数 `op(0, f)` 选出。

2. 表达式中的其它数据类型

表达式中还有如下数据类型: