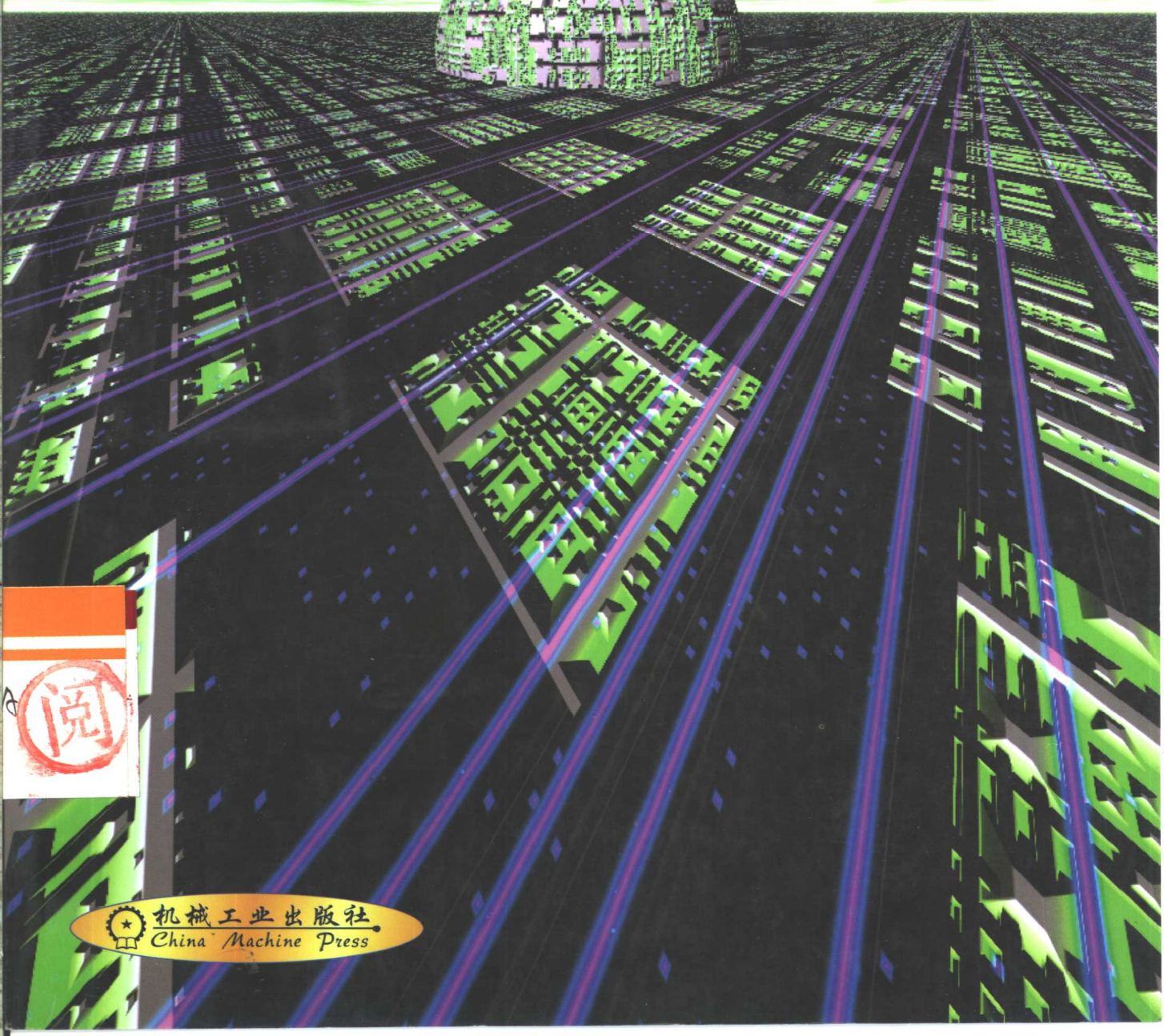


掌握和精通

Maple



马春庭 等编著



机械工业出版社
China Machine Press

掌握和精通 Maple

马春庭、高萍、吴开腾、徐立新等编著



机械工业出版社

Maple 语言是一种优秀的数学计算工具软件,1999 年 12 月 6 日,Waterloo Maple Inc.发布世界上第一种综合分析计算系统(World's First Fully Integrated Analytical Computation System) Maple V Release 6。Maple 语言以强大的符号运算功能为其重要特征,并且集符号运算、数值计算、可视化和程序设计于一体。本书是第一本较详细介绍 Maple 语言使用方法的中文图书。内容包括:系统的安装和基本操作、Maple 语言基本组成、符号运算、二维和三维图形设计、动画设计、数值计算功能以及与 MATLAB 相互调用技术等等。

本书不但详细介绍了 Maple 语言的句法、语法以及程序设计技术,而且提供了大量的运用实例。

本书适合于大中学生、研究生、教师或工程技术人员阅读。

图书在版编目(CIP)数据

掌握和精通 Maple/马春庭等编著. —北京:机械工业出版社,2000.9

ISBN 7-111-02045-6

I. 掌… II. 马… III. 数学-应用软件, Maple-程序设计

IV. 01

中国版本图书馆 CIP 数据核字(2000)第 68361 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:曲彩云

封面设计:姚毅

责任印制:路琳

高等教育出版社印刷厂印刷·新华书店北京发行所发行

2000 年 9 月第 1 版第 1 次印刷

787mm×1092mm1/16·19.5 印张·482 千字

0001—5000 册

定价:32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

前 言

计算机的高速、智能化推动了信息革命的发展,数学的应用正在向一切领域渗透。Maple 语言作为通用型的数学符号工具,正推动着工程技术、人工智能和数学的发展。

Maple 语言是由加拿大 Waterloo Maple Inc.1984 年开始推出的工程计算软件。到本书完稿时,最新版本是被称为世界上第一种综合分析计算系统(**World's First Fully Integrated Analytical Computation System**)的 Maple V Release 6。

Maple 语言的应用已经扩展到数学、计算机学、物理、经济和工程技术等领域。由于 Maple 语言强大的符号计算功能、灵活多变的二维、三维以及动画制作技术,使得它在国内外成为最普及的数学软件之一。在国内越来越多的高校学生、教师和工程技术人员正在学习这个工程数学工具软件。国内近年来最为普及的数学软件 MATLAB 语言中的符号运算功能也是借用于 Maple 语言,其核心部分由 Waterloo Maple Inc.提供。

我们相信在本书成书的几年内,Maple 语言也会成为第二种被中国人广泛接受的数学工具软件,如果在此期间本书能为 Maple 软件引进和普及起到微薄之力,那将使我们感到莫大的欣慰。

本书包括八章的内容:第 1 章介绍符号运算系统与 Maple 语言的概况;第 2 章介绍 Maple 系统的安装、用户界面、常用工具条的使用;第 3 章是 Maple 语言的组成和基本结构;第 4 章介绍 Maple 语言的过程设计知识;第 5 章为如何进行过程的调试;Maple 语言不但在符号运算方面独具特色,而且在数值计算方面也颇具独到之处,从本书的第 6 章可见一斑;第 7 章介绍 Maple 语言的二维、三维以及动画实现功能;在本书的最后一章介绍 Maple 语言在数值计算方面如何发挥 MATLAB 语言的特长和如何将 Maple 程序的源代码转换为通用的 C 语言源代码。

参加本书编写的有马春庭、高萍、吴开腾、徐立新等同志,马春庭负责全书的组织编写工作。其中高萍硕士编写了第 3 章内容,徐立新博士编写了第 5 章,其余章节主要由马春庭编写,吴开腾副教授负责本书的审核任务。参加本书编写的还有王平、张进忠、谭业双、房立清、王希武、高秀峰、王蕾、吴海平、李劲松、张伟等同志,对此表示诚挚的谢意。

由于 Maple 语言是加拿大 Waterloo Maple Inc 设计开发的,在国内到我们脱稿时仍没有一本介绍如何使用的中文图书或较详细的中文资料供参考,许多 Maple 语言中的英文术语是第一次译为中文,对某些单词汉译的表达我们也深感不是十分满意,加上水平有限,对书中的疏漏和错误之处,恳请读者朋友批评指正。

作者的电子信箱: mctbit@263.net

Waterloo Maple Inc.公司联系地址:

Waterloo Maple Inc.

57 Erb Street West

Waterloo, Ontario

N2L 6C2

目 录

前言

第1章 符号运算与 Maple 语言	1
1.1 计算机符号运算系统	1
1.2 符号运算的特征	2
1.3 Maple 语言	6
1.3.1 Maple 语言发展简历	7
1.3.2 Maple 系统的组成	8
第2章 认识系统	10
2.1 系统安装	10
2.1.1 系统要求	10
2.1.2 系统安装	11
2.1.3 系统运行	15
2.2 认识环境	17
2.2.1 文档操作 (File)	17
2.2.2 编辑操作 (Edit)	19
2.2.3 视图操作 (View)	24
2.2.4 插入菜单 (Insert)	29
2.2.5 格式菜单	30
2.2.6 电子表格 (Spreadsheet)	33
2.2.7 选项 (Options)	34
2.2.8 窗口 (Windows)	36
2.2.9 帮助 (Help)	37
2.3 图形设置	41
2.3.1 平面图形设置	42
2.3.2 三维图形设置	43
2.3.3 动画设置	47
第3章 Maple 语言	49
3.1 句法	49
3.1.1 Maple 语言的句法	49
3.1.2 Maple 语言的语义	50
3.2 Maple 的语言组成单位	50
3.2.1 语言组成单元	50
3.2.2 语法规则	50
3.2.3 特殊字符	62
3.3 语句	65
3.3.1 赋值语句	66

3.3.2	选择语句·····	70
3.3.3	循环语句·····	72
3.3.4	文件操作语句·····	76
3.3.5	空操作与退出语句·····	77
3.4	表达式·····	78
3.4.1	表达式基本运算·····	78
3.4.2	表达式的非数值计算的操作符·····	83
第4章	过程·····	96
4.1	过程的定义·····	96
4.1.1	典型过程的定义·····	96
4.1.2	映射符·····	97
4.1.3	无标识符过程及其组合·····	97
4.1.4	过程简化·····	98
4.2	参数传递·····	99
4.2.1	参数申明·····	99
4.2.2	参数序列·····	100
4.2.3	局部变量与全局变量·····	101
4.3	过程选项和描述部分·····	105
4.3.1	选项部分·····	105
4.3.2	描述部分·····	107
4.4	过程返回值·····	108
4.4.1	参数赋值式返回·····	108
4.4.2	显式返回·····	110
4.4.3	出错返回·····	111
4.4.4	未换算数值返回·····	113
4.5	典型过程举例·····	114
4.5.1	二叉树搜索·····	114
4.5.2	快速排序算法·····	116
4.5.3	契比雪夫多项式·····	118
4.6	自定义函数包·····	120
第5章	程序调试·····	124
5.1	调试过程举例分析·····	124
5.2	调用调试器·····	132
5.2.1	显示过程的语句·····	132
5.2.2	设置断点·····	132
5.2.3	显式断点设置·····	133
5.2.4	观察点设置·····	135
5.2.5	错误信息观察点·····	136
5.3.	系统状态检查与改变·····	138

5.4. 复合语句运行控制·····	144
第6章数值计算·····	148
6.1 初等代数·····	148
6.1.1 整数、有理数和小数·····	148
6.1.2 数值求和和求积·····	149
6.1.3 复数和特殊函数·····	150
6.1.4 表达式的展开、因式分解和化简·····	151
6.1.5 等式与不等式求解·····	155
6.2 高等数学·····	157
6.2.1 微分与积分·····	157
6.2.2 定积分·····	158
6.2.3 极限·····	160
6.2.4 分段函数·····	161
6.2.5 级数展开式·····	162
6.2.6 微分方程式·····	164
6.3 矩阵运算·····	170
6.3.1 矩阵的基本运算·····	170
6.3.2 特殊矩阵·····	172
6.4 代码转换·····	173
6.5 数理统计·····	175
第7章图形编程·····	178
7.1 基本作图函数·····	178
7.2 调用作图库函数编程·····	181
7.2.1 直线作图·····	181
7.2.2 闭合曲线作图·····	185
7.2.3 彩带作图程序·····	187
7.3 Maple 语言的图形数据结构·····	190
7.3.1 PLOT 图形数据结构·····	193
7.3.2 求和作图·····	195
7.3.3 PLOT3D (三维作图) 数据结构·····	198
7.4 使用图形数据结构编程·····	201
7.4.1 编写图形基本单元·····	201
7.4.2 作齿轮图·····	202
7.4.3 多边形的网格(MESH)化·····	205
7.5 使用图形工具包编程·····	206
7.5.1 比萨饼作图·····	207
7.5.2 图形投影·····	208
7.5.3 瓦面 (Tiling) 作图·····	210
7.5.4 史密斯圆图·····	211

7.5.5	多面体网格作图	212
7.6	动画	216
7.7	颜色编程	224
7.7.1	创建颜色表	226
7.7.2	添加图形的颜色信息	228
7.7.3	创建棋盘图	231
第 8 章	Maple 语言与其它语言的接口及优化	233
8.1	MATLAB 与 Maple 语言	233
8.1.1	Maple 语言与 MATLAB 语言比较	233
8.1.2	Maple 语言的矩阵功能	236
8.1.3	MATLAB 语言的符号功能	245
8.2	Maple 语言调用 MATLAB 函数	247
8.2.1	调用设置	247
8.2.2	实现 Maple 对 MATLAB 的连接	248
8.3	Maple 程序转换为 C 语言代码	256
8.3.1	预转换过程	256
8.3.2	将表达式、表达式矩阵、等式列转换为 C 代码	261
8.3.3	将 Maple 过程转换为 C 程序源码	265
8.4	Maple 语言过程优化与合并	273
8.4.1	Maple 的优化	273
8.4.2	Maple 多个过程的合并	275
附录 1	Maple 语言设定常用类型	279
附录 2	Maple 提供的主要函数名	281
附录 3	Maple 常用函数一览表	282
附录 4	线性代数工具包的函数	296
附录 5	本书使用的 Maple 常用术语对照表	301
参考文献		302

第1章 符号运算与 Maple 语言

在通常条件下，我们大家所说的“计算机”指用作数值计算的计算机。数值计算不仅包括像数值加减乘除的基本代数运算，也包括像数学函数求导、多项式求值、求矩阵的特征值和特征向量等类似的计算。显然这种通过算符运算来计算数值的功能是非常重要的。由于数值计算使用浮点来运算，所以结果不可避免地带有近似性。近几十年来，计算机技术为了将人们从复杂繁琐的数值计算中解放出来而得到了迅猛的发展。但是数学运算还包括另一重要的方面，那就是所谓的符号和代数计算。简而言之，就是计算比具体数值更具有代表性的计算对象——符号。这些符号不但可以代表整数、小数、实数或复数，而且可代表像多项式、有理函数、方程式等代数结构，甚至可以表示如群、环、域、体等更抽象的代数结构。进一步讲，运算对象的符号结果形式更强调数学问题的最终结果——表示为一个完整的公式或寻找一个符号型近似结果。但是这里所说的“符号”指根据代数规律计算得到的准确结果而不是使用浮点数来近似表示计算的结果。多项式的因式分解、函数的判别式、函数的级数展开、不等式组的解的分析、方程式的精确解和数学表达式的化简都是符号代数计算的例子。

在近 30 年来，随着符号代数理论不断发展，计算机数学计算取得了突破性成就。符号代数又称符号计算、符号处理、公式处理和计算机代数等。计算机上使用相应的符号计算的工具（软件）也称为符号计算程序、符号数字计算器、符号管理程序和计算机代数系统。同时“符号计算”的名称被用在不同的领域上，例如逻辑推导编程和人工智能领域，实际上这些领域与数学运算关系不太密切。

1.1 计算机符号运算系统

这里我们将简单回顾计算机符号运算系统的发展。计算机符号运算系统主要分为两类：专用系统和通用系统。

专用符号运算系统为解决在物理或数学某一方面的问题而设计的。例如，在高能物理方面的 SCHOOSCHIP、天体力学方面的 CAMAL、在数学特定领域的如群论理论方面的 Cayley 和 GAP、数论方面的 PARI、关系代数方面的 CoCoA、解析几何方面的 Macaulay、分析微分方程方面的 DELiA 等等。本书的内容将围绕通用系统的 Maple 语言来展开，但是不应忽略专用系统在许多科学领域所发挥的巨大作用。而且专用系统由于它们特殊的记号和数据结构而显示出了独具的特色和高效率。

通用符号运算系统包含大量的数据结构和数学函数，而且尽可能地覆盖不同的应用领域。最早的通用符号运算系统而且今天仍被广泛应用的是 MACSYMA (Macsyma Inc. PC 机最新版本为 2.4) 和 REDUCE (ZIB 公司, 1999 年 4 月 30 日发布的 REDUCE)。这两种符号运算系统产生于 60 年代。MACSYMA 提供大量的辅助软件包，因而功能强大，但

由于它依赖于非标准的 LISP 语言而且需要占用大量的计算机系统资源（硬盘空间和内存空间），所以较早的版本只适用于少数的大型计算机上，近年的 PC 版本正逐渐改变这种状况。REDUCE 最早是由应用于高能物理的专用符号运算系统逐渐发展到现在的通用符号运算系统，与 MACSYMA 相比最显著的特征是一个非常开放的系统（系统提供完整的资源代码），易于扩展和修改。最新的 REDUCE Version 3.7 在功能和兼容性方面有较大发展，而且适用于更广泛的计算机类型。

在 80 年代，MuMATH 和它的后续版本 DERIVE（Soft Warehouse Inc，有 DOS 和 Windows 两种版本，Windows 最新版本为 Version 4）是最早的非编程符号运算计算器。另一个国内用户也经常使用的通用系统为 Mathematica（最新版本 R4）。它是第一个集符号、数值和图形于一个用户界面友好的环境的数学符号运算软件。它的另一个优点是一种结构化的用户级编程语言。Maple 语言在国外通用符号运算系统中，独具特色，是一种最优秀的符号运算系统。

1.2 符号运算的特征

计算机的符号运算系统各不相同，但它们又具有某些共同特征，下面我们以 Maple 语言为例说明。

首先，符号运算系统（Maple 语言）是交互式编程语言。与数值计算系统相比较，符号运算系统使用符号表达式来进行数学运算。简而言之，用户输入一条数学表达式或一条指令，计算机的符号运算系统马上开始运行并给出结果。用户可以接着继续输入指令。下面我们将使用 Maple 语言计算一个实函数

$$f(x) = \arctan\left(\frac{2x^2 - 1}{2x^2 + 1}\right)$$

的最小值。在以后的章节可知，实际上 Maple 语言会自动求得最小值的。这里只是应用来显示 Maple 语言的交互符号运算过程。下面显示的过程使用的是 Maple R5.1 的 Windows 版本。

```
> f:=arctan((2*x^2-1)/(2*x^2+1));
```

$$f := \arctan\left(\frac{2x^2 - 1}{2x^2 + 1}\right)$$

```
> derivative:=diff(f,x);
```

$$\text{derivative} := \frac{4 \frac{x}{2x^2 + 1} - 4 \frac{(2x^2 - 1)x}{(2x^2 + 1)^2}}{1 + \frac{(2x^2 - 1)^2}{(2x^2 + 1)^2}}$$

```

[> normal(derivative);
      4  $\frac{x}{x^4 + 1}$ 
[> minimum_value:=subs(x=0,f);
      minimum_value := arctan(-1)
[> minimum_value;
       $-\frac{1}{4}\pi$ 
[> approximation:=evalf(minimum_value);
      approximation := -.7853981635
[> |

```

当在 Windows 系统运行 Maple 后, 显示带有 “>” 提示符的工作页界面。在 Maple 提示符后输入数学函数后加 “;” (分号) 结尾并输入回车键。最后输入的分号和回车键是告诉 Maple 一条指令输入完毕, 让其开始 “工作”。这时在所输入的数学函数下面显示文本效果数学符号化公式。Maple 每运行一条指令就会产生一个水平行, 用来分隔输入/输出行。

在上例中, 完成了以下内容:

- (1) 首先定义了一个函数 f ;
- (2) 然后求得这个以 x 为自变量的函数的导数并赋予变量名 $derivative$, Maple 得到的结果是一个相当复杂的表达式;
- (3) 通过 Maple 语言的函数 `normal()` 将 $derivative$ 化简;
- (4) 根据化简的结果可知, 当 $x=0$ 时, 导数的值为零, 即原函数此时具有最小值。将 $x=0$ 通过 Maple 内部函数 `subs()` 代入原函数 f 中得到最小值, 并将最小值赋予变量 $minimum_value$ 。但此时的结果是一个表达式;
- (5) 直接调用变量 $minimum_value$ 就得到符号结果, 其中含有内部参数 π ;
- (6) 调用 Maple 内部函数 `evalf()` 即可得到本函数 f 的最小值并赋值与变量 $approximation$ 。

可以发现编写程序的基本思路与人工求解数学问题的过程基本相似。读者在第二步中, 也许会问为什么 Maple 系统不自动化简多项式表达式呢? 这是因为 Maple 系统不能确定何时需要化简和什么形式是比较简单的。在多数情况下, 一个多项式可能存在几种不同的多项式形式, 只能根据具体的环境来确定哪个是比较简化的。例如有理式

$$\frac{(x^2 - 1)(x^2 - x + 1)(x^2 + x + 1)}{(x^2 - 1)^6}$$

可以简化为

$$\frac{x^6 - 1}{(x - 1)^6}$$

也可转化为适于积分运算的形式

$$1 + \frac{6}{(x-1)^5} + \frac{15}{(x-1)^4} + \frac{20}{(x-1)^3} + \frac{15}{(x-1)^2} + \frac{6}{(x-1)}$$

自动化简的另一个问题是在计算过程中，Maple 系统无法预测结果的长短和形式，所以，在需要中间结果时要随时进行人工“干预”。也就是说，在一个运算过程中可能是优秀的结果可能在另一运算过程为错误的选择。例如，有时我们认为因式分解得到的形式是比较简单的表达式

$$x^8 + 8x^7 + 28x^6 + 56x^5 + 70x^4 + 56x^3 + 28x^2 + 8x + 1$$

分解为

$$(x+1)^8$$

但是分解下式将会使您感到惊讶！

$$x^{26} + x^{13} + 1$$

分解为

$$(x^{24} - x^{23} + x^{21} - x^{20} + x^{18} - x^{17} + x^{15} - x^{14} + x^{12} - x^{10} + x^9 - x^7 + x^6 - x^4 + x^3 - x + 1)(x^2 + x + 1)$$

所以，Maple 语言只自动地化简一些简单形式的多项式，例如 $x+0$ 简化为 x 、 $x+x+x$ 简化为 $3x$ 。其它需要判断的化简运算只能通过用户来控制，Maple 语言提供了完成这些过程的工具。

符号运算系统（Maple 语言）的第二个特征是提供精确的结果。从第一个例子可以看出，Maple 语言提供的精确解为 $-\frac{1}{4}\pi$ 而不是 0.785398。不过不用担心，Maple 语言已经

为用户准备了求解达到用户要求精度的函数。例如 Maple 语言计算可以为 $\tan^2(\frac{\pi}{12})$ 提供一个精确的数值。下面的例子得到了小数点后达到 25 位数值的精度。

```
[> real_num:=tan(Pi/12)^2;
      real_num:=(2-sqrt(3))^2
> real_num:=expand(real_num);
      real_num:=7-4*sqrt(3)]
```

```
[> approx:=evalf(real_num,30);
      approx := .07179676972449082589021463396
[> |
```

符号运算系统(Maple 语言)的第三个特征,也是其最重要的特征是强大的符号运算功能。

通常,使用计算机一般高级语言编程求解方程时,主要的思路是告诉计算机如何一步一步做下去。如果想观察某个方程的曲线只能先求得一组数值解,将求得的数值解通过图形坐标描绘为图形。这样很难得到比较复杂的符号表达式的真实图形。

下面这个四阶偏微分方程式

$$\left(\frac{\partial^2}{\partial x^2}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) + n^2\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\right)f = 0$$

的解为

$$\frac{\sin\left(\frac{nz\sqrt{x^2+y^2+z^2}}{\sqrt{y^2+z^2}}\right)}{\sqrt{x^2+y^2+z^2}}$$

这个结果是否正确呢?

一般我们通过计算机只能将其中的参数有限次赋值来进行验证。下面通过 Maple 语言来验证所求得解的正确性。

```
[> settime:=time();
[> f:=sin(n*z*sqrt(x^2+y^2+z^2)/sqrt(y^2+
      z^2))/sqrt(x^2+y^2+z^2);
      sin\left(\frac{nz\sqrt{x^2+y^2+z^2}}{\sqrt{y^2+z^2}}\right)
      f:=\frac{\sqrt{x^2+y^2+z^2}}{\sqrt{x^2+y^2+z^2}}

[> simplify(diff(diff(f,x$2)+diff(f,y$2)+
      diff(f,z$2),x$2)+n^2*(diff(f,x$2)+diff
      (f,y$2)));
      0

[> cup_time:=(time()-settime)*seconds;
      cup_time := 4.443 seconds
[>
```

由程序运行结果可知,在计算机系统为 PIII450,内存 64MB,运行的时间为 4.443s。

在求解一般工程问题时,微分与积分运算是会常常遇到的。但是如果通过手工解需要大量的时间和精力。而且有时需要查找专门的数学手册。使用 Maple 语言会事半功倍的。例如积分式

$$\int \frac{1}{x\sqrt{bx+cx^2}} dx = -2 \frac{\sqrt{bx+cx^2}}{bx}$$

下面使用 Maple 来积分:

```
[> 1/(x*sqrt((b*x+c*x^2)));
      1
      ───
      x√bx+cx2
> normal(integrate(1/(x*sqrt((b*x+c*x^2
    ))) , x));
      √bx+cx2
      ───
      -2  bx
> normal(diff(%,x));
      1
      ───
      x√bx+cx2
> ]
```

符号运算的其它特殊功能包括多项式积分、线性或非线性方程的求解、冗余方程和差分方程求解、数值和符号系数矩阵的运算等等。

在国外,因为符号运算系统为人们提供了比笔和纸更精确和高效地解决数学问题的工具而称为“数学专家系统”。

1.3 Maple 语言

“Maple”汉语的意思是“枫叶”, Maple 语言是加拿大(Canadian) Keith Geddes 和 Gaston Gonnot 教授在于 1980 年 11 月在 Waterloo 大学(the University of Waterloo, 位于安大略湖[Ontario,北美洲中东部]美国和加拿大共有的湖)开始设计开发的用于科研和教育的数学软件。最早的版本只用了大约 3 周的时间就完成了。在赞助者的推动下,大约在 1982、1983 年左右, Maple 语言被推广到美国和欧洲的一些大学。应用的领域扩展到数学、计算机学、物理、经济和工程等。

1.3.1 Maple 语言发展简历

开始 Maple 由 B 语言编写成, 但很快转用 C 语言 (C 语言由 B 语言发展而来)。

在 1984 年 Watcom Inc. 开始代理销售 Maple (当时的版本为 V3.3), 到 1987 年末, Maple 4.0、4.1 和 4.2 发布。

1988 年 Waterloo 开始直接销售并继续完善 Maple。一年后发布的 4.3 版本是开拓性的, 它可以在 20 种不同的操作系统平台运行 (包括当时作为主流机型的 PC 机)。这时的开发小组包括本公司的雇员和 Waterloo 大学的专家。

最早的 Maple V 发布于 1990 年 (这就是今天看到的 Maple V5.1、Maple V 5.2、Maple V6.0 的原因)。Maple V 包括新的图形用户界面 (GUI)、3D 图形和支持 UNIX 的 XWindow。

Waterloo Maple Software 在 90 年代的早期联合了不同的公司来共同提高系统的功能。其中包括 Mathsoft Inc. (MathStation 和 Mathcad 就是它开发的) 和 Visual Numerics Inc.。Waterloo Maple Software 购买了加利福尼亚州的一家软件公司以获取在符号运算、图形和动画方面的先进技术。

Maple V R3 发布于 1994 年 3 月。它的特征是显著提高了符号和数值运算的算法、用户界面更简单、工作页可输出为 LaTeX 格式、新的联机帮助系统和一种扩展的可编程的语言。当时一家联邦德国的计算机杂志这样评论: “图形用户界面的 Maple 是一种对初学者非常容易并可很快掌握的而且达到了最新技术发展水平的软件, 特别是 MS Windows 版本的推出 (Maple is an [GUI] application that is state-of-the-art that offers enough comfort to beginners and performance to advanced users. Especially the version for MS Windows turned out very well)”

1995 年针对 Microsoft Word 6.0 的 MathOffice 开始发布。它为将 Maple V Release 3 的计算结果和图形输入到 Word 文件提供了界面。

很快, 1996 年 1 月 Release 4 发布, 它增强了编辑功能并提供了崭新的图形用户界面 (GUI), 并且重新设计和扩充了 (包括内部和外部) 函数。

1997 年 12 月, Waterloo Maple Inc. 开始销售 Maple V Release 5 for Windows。GUI 进一步完善, 并出现了弹出式菜单、调色板、实时 3D 图形, 增加了微分的软件包。Release 5 另一个特征是提供了与 Matlab5 信息交流的界面。1998 年 2 月 23 日发布了 UNIX 和 Macintosh 的 R5。6 月发布日文的版本。

1998 年 4 月 20 日, Waterloo Maple Inc. 隆重庆祝公司成立 10 周年。公司的员工由开始的 5 人增加到 75 人。1998 年夏天公司搬到 Waterloo (沃特卢, 加拿大东南部城市) 的 Seagram 商业区。

1998 年 8 月 12 日, Waterloo Maple Inc. 与英国剑桥的 the Numerical Algorithms Group (NAG) 结成联盟开发下一代解决数学问题的软件。NAG 是著名的数值和统计算法开发商。它的产品是计算机代数系统 Axiom (在国外与 Maple V, Mathematica, and MuPAD 一样著名)。当时, Waterloo Maple Inc. (公司简称 WMI) 计划在 1999 年完成开发下一代 Maple V 系统此系统将包括 NAG 的数值函数库。

1998 年 10 月 27 日, Waterloo Maple Inc. 发布 Maple V Release 5.1。

1998 年 11 月, Waterloo Maple Inc. 发布 Maple V Release 5.2。

1999 年 12 月 6 日, Waterloo Maple Inc. 发布世界上第一种综合分析计算系统 (World's First Fully Integrated Analytical Computation System) Maple V Release 6。

2000 年 2 月 23 日, Waterloo 开始向用户提供 Windows、Macintosh、UNIX 和 Linux 系统环境的 Maple V Release 6。

2000 年 4 月 6 日, Waterloo 开始提供适合学校使用的学生版本。

1.3.2 Maple 系统的组成

Maple 系统包括三部分: 称为 Iris (英语意思彩虹之女神 [希腊神]) 的用户界面、基本代数运算器——kernel 和外部的函数库——library。

表 1-1 Maple 系统各部分功能

部 分	功 能
Iris	判断输出是否正确 显示表达式 图形 特殊用户界面: 工作页
Kernel	控制中断 内存管理 执行基本的代数运算
Library	函数库 应用软件包 联机帮助

由 Iris 和 kernel 组成了 Maple 系统最小的运行系统, 这两部分是由 C 语言编写。在 Maple 语言启动时就会调入内存中。

Iris 主要处理数学表达式的输入、表达式的显示、函数的描绘和支持用户其它与系统的联系。在 MS-Windows 等操作系统中, 提供称为工作页 (worksheets) 的界面。

Maple 系统的 kernel 处理用户的输入和基本的代数运算如小数计算和基本多项式计算。为了提高运算速度, 它包括一些用低级语言编写的经常使用的代数过程, 如多项式运算的 Degree、Coeff 和 Expand 函数。存储管理也属于 kernel 的职责范围。Maple 系统计算的最重要的特征是: 在全部的段落中, Maple 系统只保留每个表达式或子表达式的一份拷贝。这样不需要重新运算所需的标识符, 以减少运行操作。

绝大多数的 Maple 系统的函数通过 Maple 语言来编写并且保存在外部库内。Maple 系统会很聪明地自动调用常用的外部函数, 很少使用的外部函数必须通过用户来调用。具体的用法在以后的内容中介绍。需要特别指出的是 Maple 系统仅仅在内存中保存重要的东西而不浪费空间给用户不感兴趣的内容。所以 Maple 语言需要很少的内存。

总之, Maple 语言是一种优秀的综合性高级编程语言。它支持多种数据结构, 如: 函数、序列、集合、表和数组等。同时, 也提供了使用方便的对各种数据结构的操作, 如选择、组合等。Maple 语言提供的外部函数库是开放型的, 用户不但可以打开函数进行学习

其编程技巧，而且可以修改其中的语句以满足用户的不同使用。用户也可将自己编写的程序（函数）扩展到外部函数库中。Maple 语言也提供了可以为自动或用户设置的对程序运行工程的记录，这样方便了用户对程序的调试。