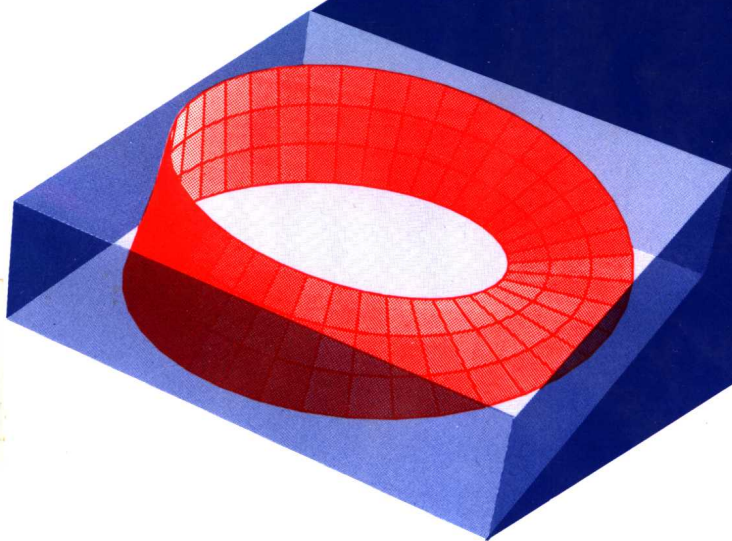


Mathematica

数学软件系统的应用 及其程序设计

裘宗燕 编

北京大学出版社



新登字(京)159号

内 容 提 要

本书介绍当今世界最流行的一个综合性计算机数学软件系统——*Mathematica*。该系统是一个集成化的计算机软件,它的主要功能包括符号表达式计算、数值计算和图形等方面。它的主要使用者包括从事各种理论工作(数学、物理等)的科学工作者,从事实际工作的工程技术人员,高等、中等学校的教师和学生等。本书通过对这个系统各方面的分析和许多实例,帮助读者了解 *Mathematica* 系统和以它为代表的一类新型计算工具的原理和使用。书中各章均附有习题,供读者学习时参考。本书可以作为大专院校理科各系有关课程的参考书,也可以为读者自学使用。

书 名: *Mathematica* 数学软件系统的应用 及其程序设计

著 作 责 任 者: 裘宗燕

责 任 编 辑: 孙 晔

标 准 书 号: ISBN 7-301-02421-5/TP·198

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学内

电 话: 出版部 2752015 发行部 2559712 编辑部 2752032

排 印 者: 北京飞达印刷厂印刷

发 行 者: 北京大学出版社

经 销 者: 新华书店

版 本 记 录: 850×1168 32开本 8.375印张 210千字

1994年7月第一版 1999年4月第三次印刷

定 价: 15.00元

序 言

计算机能够为人类做什么？这是缠绕在无数人头脑中的一个谜。虽然计算机科学的研究早已发现了计算机不能解决的问题，但是由于那些问题通常非常抽象，与实际距离很远，它们除了告诉人们计算机绝不是万能法宝之外，没有为人们提供太多的信息。而且，对于任何客观的问题，如果把它推到广而又广，推广到非常一般性的问题去，就可能使它们变得无法解决（或无法用计算机解决）。反过来讲，许多一般性的问题可能不能有（或者现在还没有找到）一个用计算机解决的方法，但是它们的一些特殊情况，往往是人们非常关心的而且又具有实际价值的子问题，却能够用计算机有效地处理。发现和挖掘出这样的问题，找出（设计和实现）有效的计算机解决方法，开拓计算机应用的新领域，是计算机科学技术研究和开发的主要目的之一。今天，计算机的应用早已远远超出人们通常用“计算”这个词所描述的范围。人们用计算机做的工作早已不仅仅是数值计算和一般的数据处理，计算机在各种非数值计算方面的工作帮助人们解决了各个方面的难题。功能强大的计算机系统不仅是工程设计、经营管理不可缺少的助手，也已经成为科学研究工作者手中的有力武器。

人们都非常了解数学在计算机科学技术的发展中不可替代的重要作用，可以说，没有数学研究的贡献，就没有今天的计算机。但是反过来讲，许多数学工作者原来却看不起计算机。他们认为计算机能够做的就是算算数值，不过是算得快一点，而真正的数学研究要做的是演算、推理和证明，是建立和使用抽象的符号系统。现在，情况已经开始改变了，今天计算机科学技术的发展已经开始给数

学以回报,一批具有演算推理能力的计算机系统开始崭露头角,它们作为更具有数学意义的计算工具的价值已经和正在得到数学界越来越广泛的承认。不了解和认识这种情况,不积极行动起来去学习和掌握新的数学工具,对于从事数学方面工作和需要使用数学做工作的科学技术人员可能是一个失误。将来回过头看,这个问题可能看得更清楚。

实际上,需要做公式演算和推理的不仅仅是数学工作者和理论工作者,许多工作在实际领域的工程技术人员也需要做这样的事。在设计解决实际问题的计算公式、建立数学模型、分析实际问题的过程中,数学演算的作用是非常明显的。过去对于这些问题,许多工程技术人员往往凭实际经验简单地处理,或者套用可能“合用”的公式。对于复杂问题,这些方法是不可靠的,因而可能是很危险的。之所以产生这种情况,除了人的因素以外,主要是由于这些问题的解决一般都比较复杂,处理起来太耗费时间和精力。今天,强有力的计算机数学工具的出现使人们有可能采取更加科学的态度对待这些问题。

新一代计算机数学工具的出现会使人们做数学和使用数学解决问题的方法发生显著的变化。一些原来人们引以自豪的能力,例如很强的符号演算能力、做许多页演算不出笔误的能力等等,可能逐渐变得不再那么有价值了。因为,这其中许多事情由计算机做起来显然更容易。但是也应当看到,任何计算机系统(无论是今天已有的,还是明天将要出现的),其能力都是有限的,它们不可能为我们解决在实际中面临的一切问题。通用的系统总是只能完成一些最常用的基本的工作;专用的系统只能解决一个(或一些)非常专门的问题。而实际问题由于其缤纷繁杂、丰富多采永远不可能为任何“万能”的系统所覆盖。人们对问题的理解和洞察力永远是无比重要的,如何把一个实际面临的问题变成当前可用的计算机系统能够处理的问题,并且控制、操作计算机系统处理并解决之,这就体现出人的能力。使用计算机处理数学问题,人的工作将主要是

分析问题、明确处理方法、规划解决问题的途径、操作指挥计算机的工作,并在工作中不断分析问题和实际面临的情况,调整工作的方向和方法,最终达到问题的解决。在使用计算机的工作中,人可以摆脱许多机械性的工作细节,把自己的注意力集中到更有创造性的问题上。这样就可能用较少的时间做出比较有价值的结果。对于做实际工作的人情况也应当是类似的。

使用计算机解决数学问题(和解决其他问题一样)可能发展出一些新的方法。例如先提出假说,用计算机做先期检验,可能是各种简化情况的检验,以考验假说的可能性。然后再考虑如何从理论上严格处理。计算机的图形能力、数值计算能力和符号计算能力都可能在处理数学问题的过程中发挥作用。问题是在什么时候使用什么功能和怎样利用它们。

今天,人们已经开发出一些能够帮助人处理和解决数学问题的计算机系统。在一些国家,如美国,这样的系统已经得到比较广泛的应用。它们被用在研究和实际工作的许多领域,帮助人们解决了许多问题。越来越多的人认识到这类系统的有用性,在自己的工作中开始把它们作为一种助力。本书要详细介绍的 *Mathematica* 系统是近几年国外开发出来的一个取得了很大成功的系统。它可以被看作是新一代的计算机数学系统的一个代表,已经为相当多的用户接受和使用。对于这个系统的了解不但可以直接掌握一种非常有用的数学工具,对于学习其他类似的计算机系统也会有极大帮助。当然,作为一类新的计算机应用系统,它们的研究开发还处于初步阶段,许多问题还有待于进一步研究。把这一类系统应用于实际的过程中一定会提出许多有价值的新问题,从而推动它们的改进和发展。

Mathematica 系统是美国 Wolfram 研究公司开发的一个功能强大的计算机数学系统。它提供了范围广泛的数学计算功能,支持在各个领域工作的人们做科学研究和工程中的各种计算。它的主要使用者包括从事各种理论工作(数学、物理、...)的科学工作者。

从事实际工作的工程技术人员,高等、中等学校教师和学生等。这个系统可以帮助人们解决各种领域里的涉及比较复杂的符号计算和数值计算的理论和实际问题。从某种意义上讲, *Mathematica* 是一个复杂的、功能强大的解决计算问题的工具。它可以自动完成许多复杂的计算工作,如求一个表达式的(符号)积分、做一个多项式的因式分解等等。人们也可以操作它、指挥它去一步一步地处理研究领域里的或工程中的复杂的实际问题,就像机械工人操作机床加工复杂的工件似的。在过去,人们只能用纸和笔作为工具去处理这样的问题,用自己的头脑去记忆、考察和判断。由 *Mathematica* 这样一类系统的出现带来的思维与解题工具的革新很可能对各种研究和工程领域产生意义深远的影响。

1 *Mathematica* 能够做什么

Mathematica 是一个集成化的计算机软件系统。它的主要功能包括三个方面:符号演算、数值计算和图形。*Mathematica* 可以完成许多符号演算和数值计算的工作。例如,它可以做多项式的各种计算(四则运算、展开、因式分解等);有理式的各种计算。它可以求多项式方程、有理式方程和超越方程的精确解和近似解;做数值的或一般表达式的向量和矩阵的各种计算。*Mathematica* 还可以求一般函数表达式的极限、导函数,求积分,做幂级数展开,求解某些微分方程等等。使用 *Mathematica* 可以做任意位的整数的精确计算、分子分母为任意位整数的有理数的精确计算(四则运算、乘方等);可以做任意位精确度的数值(实数值或复数值)的数值计算。这个系统的所有内部定义的整函数和数值(实数值和复数值)计算函数也都具有这样的性质。使用 *Mathematica* 可以非常方便地作出以各种方式表示的一元和二元函数的图形,可以根据需要自由地选择画图的范围和精确度。通过对这些图形的观察,人们常常可以迅速形象地把握对应函数的某些特征,这些特征仅仅从函数的符号表

达式一般是很难认识的。

Mathematica 系统的能力还不仅仅在于具有上述这些功能,更重要的是在于它把这些功能融合在一个系统里,使它们成为一个有机整体。在使用 *Mathematica* 的过程中,使用者可以根据自己的需要一会儿从做符号演算转去作图形,一会儿又转去做数值计算,等等。这种灵活性为使用者带来很大的方便,经常能使一些看起来非常复杂的问题变得易如反掌,使问题处理起来得心应手。在学习和使用的过程中读者一定会进一步体会到这些。*Mathematica* 还是一个很容易扩充的系统,它的用于描述符号表达式和对它们的计算的一套记法实际上构成了一个功能强大的程序设计语言,用这种语言可以比较方便地定义用户需要的各种函数,如符号计算函数、数值计算函数、作图函数或其他具有复杂功能的函数,完成用户需要的各种工作。系统本身提供了一批用这个语言写出来的完成各种工作的程序包,在需要时可以调入系统使用。用户自己也可以用这个语言写出自己的专门用途的程序或软件包来。

2 *Mathematica* 系统的历史和现状

用计算机做代数计算、符号计算的工作开始于五六十年代。在70年代人们开发了几个比较成功的计算机代数系统。其中最著名的是 REDUCE 系统和 MACSYMA 系统。它们被用在解决许多复杂的计算问题的工作中,取得了很大成功。

Mathematica 系统是美国物理学家 Stephen Wolfram 领导的一个小组开发的,后来他们成立了 Wolfram 研究公司。1987 年推出了系统的 1.0 版,1989 年推出了改进的 1.2 版,并在美国和世界上广为流传,得到好评。1991 年该公司又推出了系统的 2.0 版,对原有的系统做了不小的扩充(扩充了二百多个系统函数和变量),在一些基本问题的处理上也做了一些改动。最近又在某些机器上推出了 2.1 版。Wolfram 公司自己声称现在仅在美国就有十

几万人经常性地使用 *Mathematica* 系统,用它解决研究和工程计算领域中的问题。

Mathematica 的基本系统主要是用 C 语言开发的,因此可以比较容易的移植到各种计算机和运行环境上。在微型机上可以用的有 MS-DOS386 版本(可以在没有 387 数学协处理器的 386 机器上运行)和 MS-DOS386/387 版本(可以在 486 和带 387 数学协处理器的 386 机器上使用)。硬件上一般要求有 4 兆内存和一定的硬盘空间用于放置系统的临时文件,如果机器有更大的内存,系统的工作速度可以进一步提高。另外,还有在 MS-WINDOWS 上运行的版本,这个版本的用户界面和使用方式都利用了 WINDOWS 的能力和方式,使用起来比较方便。但是它的使用需要比较大的内存,速度也比较慢,似乎不能显示出很大的优越性。在苹果公司的 Macintosh 微型机上也有 *Mathematica* 的版本。Wolfram 公司还把这个系统移植到许多主要的工作站和某些大型机上,例如 Sun 工作站、DEC 工作站和 IBM 的 R-6000 和 SGI 的工作站等等。

3 *Mathematica* 是一个什么样的系统

Mathematica 是一个交互式的计算系统。这里说的交互式是指:在使用 *Mathematica* 系统的时候,计算是在使用者(用户)和 *Mathematica* 互相交换、传递信息数据的过程中完成的。用户通过输入设备(一般讲是计算机的键盘)给系统发出计算的指示(命令),*Mathematica* 系统在完成了给定的计算工作后把计算结果告诉用户(一般通过计算机的显示器)。从这个意义上说,*Mathematica* 可以看成是一个非常高级的计算器。它的使用方式也与使用计算器类似,只是它的功能比一般的计算器强大的多,能接受的命令也丰富的多。用这个系统的术语,*Mathematica* 接受的命令都被称作表达式,系统在接受了一个表达式之后就对它进行处理(这个处理

过程叫做对表达式求值),然后把求得值(计算结果)送回来。

与一般的程序设计语言不同,*Mathematica* 的处理对象不限于数(整数和近似数)。它的处理对象是一般的符号表达式,也就是具有一定的结构和意义的复杂符号表示。数是一种最简单的表达式,它们没有内部结构。数学中的代数表达式也是符号表达式的例子,它们可以具有相当复杂的结构。一般地说,一个表达式是由一些更简单的部分构成的。数和代数式都是 *Mathematica* 能够处理的处理对象。

不同计算机上 *Mathematica* 系统的基本部分是一样的,只是它们的系统界面形式、用户与系统交互的方式可能有所不同。*Mathematica* 的界面基本上有两种,一种是行文形式的,一种是图形形式的。使用行文形式界面的系统时,用户一行一行地输入命令,一个命令输入完毕,*Mathematica* 系统就立刻处理这个命令,并且返回计算结果。图形方式界面的系统使用起来更灵活,使用者不但可以用键盘输入,还可以利用鼠标器等输入设备,可以通过选菜单等方式向系统发出命令。在 MS-DOS 上运行的 *Mathematica* 系统的界面是行文方式的,而运行在 MS-WINDOWS 上的系统具有图形用户界面,它可以利用 WINDOWS 系统提供的各种能力。

本书的内容尽量考虑不依赖于某种特定的具体机器,但是又考虑到当前国内的实际情况。目前国内使用最多的是运行在 PC386/486 微型机的 DOS 系统上的 *Mathematica*。因此书后附录专门讨论了与这种系统的使用有关的一些问题。书中各个章节里将不讨论与界面的使用有关的问题,我们将假定用户总是使用键盘,以行文方式输入表达式。从作者自己的使用经验看,对于用 *Mathematica* 系统做科学和工程计算的人来说,使用微型机上的这种系统是最节约资源和有效的。

本书可以作为学习 *Mathematica* 系统的自学读本,也可以作为高等学校理科一门 *Mathematica* 课程或以这个系统为基础的计

计算机符号演算课程的参考书。作者曾用本书的内容为北京大学理科的教师和研究生开办过几次学习班,为北京大学理科本科生和研究生开过几次选修课。作者非常感谢所有参加学习班和选修课的教师和同学,正是由于和他们的充分而深入的讨论,作者才能积累起比较多的素材和经验,编写出这本书。作者感谢林建祥教授,是他一直在国内大力宣传 *Mathematica* 系统,并鼓励作者开始这方面的工作。作者感谢叶其孝教授仔细地审阅了本书的全文,提出了一些重要的意见和建议;感谢北京大学出版社的孙晔和邱淑清老师为本书的出版所做的认真而细致的工作。

作者

1993年9月于北京大学

目 录

第〇章 系统启动和退出	1
第一章 数、函数、变量和表达式	3
1.1 数的表示和计算	3
1.2 计算的精度问题	8
1.3 常用数学函数	10
1.4 变量和算式	12
1.5 画出函数的图形	16
1.6 Mathematica 使用中的几个问题	21
练习题	24
第二章 符号演算和数值计算	26
2.1 符号表达式的一般处理过程	26
2.2 基本的表达式变换	27
2.3 表与表的生成	31
2.4 微积分及其他数学运算	35
2.5 求解方程	42
2.6 函数和函数的定义	46
2.7 数值数学与数据处理	51
2.8 保存和使用已经得到的计算结果	54
2.9 关系判断和逻辑表达式	58
2.10 交互式计算, 程序包的使用	59
练习题	61
第三章 Mathematica 的图形	63
3.1 基本二维函数作图	64
3.2 图形的重新显示、组合、存储和输出	71
3.3 二维参数图形	74
3.4 三维作图	78

3.5 三维参数图形	81
3.6 等值线图、密度图和数据作图	84
3.7 Mathematica 图形表达式的构造	87
3.8 图形的构建	93
练习题	96
第四章 表与表达式的结构	99
4.1 表的生成	99
4.2 表的结构与结构操作	103
4.3 表作为一种表示结构	114
4.4 表达式的结构	117
4.5 表达式的结构操作	123
4.6 与表达式有关的判断	125
4.7 与表达式结构有关的函数	126
4.8 纯函数,一些缩写形式	127
4.9 其他表达式操作函数	130
练习题	132
第五章 变换规则、模式和表达式求值	134
5.1 变换规则与表达式的求值	134
5.2 用规则定义函数	138
5.3 模式与变换规则	141
5.4 规则定义和规则类	145
5.5 规则的使用条件与顺序	148
5.6 参数数目不定的函数	155
5.7 规则与程序对象的关系	156
5.8 函数的属性与属性的定义	158
5.9 表达式的求值过程	161
5.10 可选参数、缺省值问题	164
5.11 模式的完整结构	168
5.12 模式匹配的能力限制	170
5.13 对表达式的操作	171
练习题	171
第六章 程序结构与程序设计	174

6.1 过程与局部变量	174
6.2 执行控制,循环与分支结构	176
6.3 程序流的控制	181
6.4 程序设计中的几个问题	184
6.5 字符与字符串	189
6.6 程序包的结构	191
6.7 名字与上下文	195
6.8 几个例子	200
练习题	205
第七章 错误处理、输入和输出	207
7.1 错误信息的检查与处理	207
7.2 输出形式	211
7.3 输出	215
7.4 输入	216
7.5 几个重要的系统变量	219
练习题	222
第八章 例子和课程作业	224
8.1 利用系统的作图函数	224
8.2 曲线在复映射下的象	228
8.3 几个课程作业	230
附录 A 常用 Mathematica 系统函数	232
附录 B 386/486微机 MS-DOS 上使用	
Mathematica 系统的几个问题	243
参考文献	254

第〇章 系统启动和退出

使用 *Mathematica* 系统做计算首先必须启动它。启动行文形式界面的系统的方式是通过在键盘输入系统的启动命令(在 MS-DOS 上启动系统的方法参见附录 B), 在 WINDOWS 系统下启动 *Mathematica* 则是通过选择系统的图标。输入了启动命令, 几秒钟后计算机的屏幕上就会显示出 *Mathematica* 系统的提示符(图形界面的系统可能不显示系统提示符, 但显示系统界面):

```
In[1] :=
```

这时系统已经进入交互状态, 在等待用户输入的命令。方括号里面的数字是输入表达式(输入命令)的编号, 第一次是数字 1, 以后接着是 2, 3, ...。在 *Mathematica* 的交互状态下, 使用者可以输入任何形式上符合系统要求的表达式, 输入完成后按一下回车键(Enter 键), 系统就会按照表达式的要求执行计算, 然后把计算结果显示在屏幕上(图形界面的系统一般使用另外的键向系统发出“执行”命令, 例如 MS-WINDOWS 上的系统用 Shift+Enter 组合键或 Ins 键要求系统对输入进行计算)。结果显示的形式是:

```
Out[xx] = ...
```

方括号里是输出编号, 与相应的输入一致。等号后面是表示结果的表达式。我们首先应当记住系统的退出命令, 它的形式是 Quit。任何时候在交互状态下, 用户只要键入 Quit 命令, *Mathematica* 系统

就会结束执行,计算机将返回启动 *Mathematica* 之前的状态。

Mathematica 的表达式是由字母、数字和标点符号组成的。它区分字母的大写和小写,这就是说,在需要写大写字母的地方必须用大写字母,对于小写字母也是一样。否则系统将会错误地解释输入的命令,执行错误的动作,得到错误的结果。在每一个表达式输入完毕之后应当按回车键。

本书后面的章节里要举许多实例。这些实例总是放在独立的行里,放在系统的提示符(In[xx]':=)后面,在后面的章节里,系统提示符也不用了。为了节约篇幅,大部分实例的系统输出没有包括在文中。此外,大家知道 *Mathematica* 有很强的作图能力,用它画出丰富多采的图形,可以很方便地画出各种彩色图形。考虑到目前国内印刷出版的实际情况,书中仅给出了少数几个例子的图形。后面大量与图形有关的实例都没有画出对应的图形,而只用文字说明,实际图形留给读者自己从计算机屏幕上去观察了。

Mathematica 是一个计算机软件系统,和其他的计算机系统一样,它对于使用者的输入形式有比较严格的要求。用户必须按照系统的要求书写命令(在 *Mathematica* 里一切完整的输入都是表达式),系统才能正确地对其处理,得到所需要的结果。因此,读者应当注意在学习使用中逐步熟悉 *Mathematica* 系统的表达式描述方式。为了方便使用,*Mathematica* 系统在设计时考虑了使它的输入形式尽量符合一般数学的习惯。但是由于计算机输入的某些限制,例如,用计算机键盘无法方便地输入上下标、分式、根号等,系统在这种地方只能采用变通的方式,读者应当理解这种情况。

第一章 数、函数、变量和表达式

在这一章里将讨论 *Mathematica* 的基本功能和怎样使用它去做计算。读者应当先了解这个系统关于表达式书写的基本规定,包括最基本的各种数的表示,系统的最常用的各种函数,以及如何书写一般的代数表达式问题。

1.1 数的表示和计算

Mathematica 的数分为两大类:一类是直接用数字(和小数点)写出来的数;另一类是系统的内部常数,它们对应着常用的数学常数。

1.1.1 整数和整数计算

直接写出的数(一般计算机语言里称之为直接数)包括精确的整数和有理数。整数由一串连续的数字组成,数字之间不能有空格或其他字符。*Mathematica* 和一般的程序设计语言不同的一点是它可以对任意大的整数进行计算。它将保持输入的和计算出的整数永远是精确的,不把大的整数转换为浮点数。如果在 *Mathematica* 的提示符状态下输入了一个整数,例如:

```
In[1] := 12345678987654321
```

系统将马上回答:

Out[1] = 12345678987654321

这就是 *Mathematica* 对这个整数计算(求值)的结果。*Mathematica* 的算术运算符包括加减乘除和乘方,它们分别用字符 $+$ 、 $-$ 、 $*$ 、 $/$ 和 $^$ 表示。两个整数做加减乘和乘方运算的结果总是整数,例如:

In[2] := 234^{12}

将得到一个非常大的整数。可以很容易地算出 2 的 1000 次方:

In[3] := 2^{1000}

这是一个 300 多位的数。

在这里有几个问题要注意:

1. 表示乘法的“ $*$ ”可以用空格符号代替。在两个整数之间放一个空格就表示求它们的乘积;

2. 算术运算的优先顺序符合一般数学中的习惯,先乘方、再乘除、最后加减。可以用括号改变计算的顺序;

3. 对于一般的算术运算符,连续的几个同级运算从左到右进行,例如 $3 + 4 - 5$ 表示 $(3 + 4) - 5$, $540 / 6 * 2$ 表示 $(540 / 6) * 2$ 而不是 $540 / (6 * 2)$,这一点一定要注意。乘方运算的结合顺序不同,是从右到左进行的。这意味着 $(2^3)^4$ 应当写成 $(2^3)^4$,而 2^3^4 表示的是 2^{3^4} 。

4. 与一般程序设计语言和数学的习惯一样,负号用减号表示,直接写在数的前面。

由于整数是精确数,如果参加运算的都是整数,结果也应当是整数的话,系统将永远给出精确的结果。当然,越大的数计算起来用的时间越长。