

应用数学与计算机实训

应用数学与计算 上机实训

王信峰 戈西元 邢春峰 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>



电子工业出版社

应用数学与计算上机实训

王信峰 戈西元 邢春峰 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

本教材是作者在《应用数学与计算实训》的基础上编写的,内容包括:数学实验的软件基础、配合《应用数学与计算(修订版)》编出的12个基础实验、配合《应用数学与计算(修订版)》更深入的7个专题训练、使用《应用数学与计算(修订版)》内容的数学建模综合实习,共四章,另附有4个附录。

本教材是《应用数学与计算(修订版)》的配套教材,它以《应用数学与计算(修订版)》的内容为主线,以功能较强的 Mathematica 4.0 软件为媒介,兼顾 Mathematica 2.2 及更低版本,强调以学生为主体,体现学生自己在练习中学习、体验如何使用数学;强调以学习数学软件为基础,以学习数学的思想、方法为最终目的;强调与理论课教材《应用数学与计算(修订版)》的配套使用。

本教材可作为高等院校高等数学课的教学改革的基础实验教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

应用数学与计算上机实训/王信峰,戈西元,邢春峰编著. —北京:电子工业出版社,2000.9

ISBN 7-5053-5865-0

I. 应... II. ①王...②戈...③邢... III. 计算数学-图形软件 IV. 024

中国版本图书馆 CIP 数据核字(2000)第 40857 号

书 名:应用数学与计算上机实训

编 著 者:王信峰 戈西元 邢春峰

责任编辑:张孟玮

特约编辑:朱强国

排版制作:电子工业出版社计算机排版室监制

印 刷 者:北京兴华印刷厂

装 订 者:三河市双峰装订厂

出版发行:电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:12.25 字数:314 千字

版 次:2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷

书 号: ISBN 7-5053-5865-0
G·525

印 数:5000 册 定价:16.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

前 言

《应用数学与计算实训》出版以来,受到了高等院校广大师生的关心和支持,在此我们首先对他们表示衷心的感谢。

本书将以最新 Mathematica 版本 Mathematica 4.0 为媒介,兼顾 Mathematica 2.2 及更低版本的使用者,以辅助学生学习《应用数学与计算(修订版)》的基本内容为目的,强调与课堂理论课教学相配合;强调以学习数学软件为基础,以学习数学的概念、思想、方法为最终目的。以使学生了解数学在实际问题中的作用,也使学生从一些繁杂的数学计算与推导中解脱出来,从而提高学生对数学的学习兴趣;以学生自己为主导,从而促进学生积极主动地参与数学的教学,以达到教与学的良性循环。

这次编写,删除了《应用数学与计算实训》中的第 1 章内容,增加了一些附录内容,特别是增加了附录 3——Mathematica 命令汇总与附录 2——使用 Mathematica 可能出现的错误,以便于广大读者使用 Mathematica 过程中的查询;将第 2 章 Mathematica 部分菜单简介也加以修订放在附录 1;增加了一些动画程序,并将原有的动画程序汇总在一起放在附录 4,这样使得实验内容的引入更直截了当;在实验内容的改写方面,调整了探索方式(将结果基本放在一段内容最后)的编写作风,将一些结果前放,并将要用的数学理论、方法等放在各实验小段之前,以减小实验难度;同时,根据《应用数学与计算(修订版)》的内容,调整了各实验的内容与次序,以利于更方便地使用本教材。

参加本版编写的人员有:王信峰、戈西元、邢春峰,他们都是多年一直从事高等教育数学教学与研究的一线教师。

本书由北京联合大学基础部朱玉娥副教授主审,编写修订过程中,广大同仁对此书给予了无限关怀,并对教材的编写修订提供了很好的建议,在此,我们表示衷心感谢,也感谢《应用数学与计算实训》的作者代西武、付百文、王朝旺老师的积极合作。同时,我们更感谢电子工业出版社在本书编写与出版过程中的积极支持与帮助。

限于编者的水平,不妥与错误之处在所难免,希望广大读者批评指正。

编 著 者

2000 年 4 月于北京

目 录

第 1 章 数学实验的软件基础	(1)
1.1 Mathematica 系统简介	(1)
1.2 Mathematica 系统使用入门	(1)
1.2.1 系统的算术运算	(1)
1.2.2 代数式与代数运算	(3)
1.2.3 变量与函数	(6)
1.2.4 Mathematica 的绘图初步	(7)
1.2.5 数表及其有关操作	(10)
1.3 Mathematica 流程	(11)
1.3.1 比较运算符与逻辑运算符	(11)
1.3.2 条件语句	(12)
1.3.3 循环语句	(13)
练习题	(14)
第 2 章 基础实验	(15)
实验 2.1 一元函数图形及其变换	(15)
实验 2.2 二元函数作图与立体图形	(24)
实验 2.3 极限及收敛控制条件	(32)
实验 2.4 连续与间断	(40)
实验 2.5 导数与求导有关计算	(49)
实验 2.6 导数应用	(58)
实验 2.7 定积分定义及其计算	(66)
实验 2.8 级数的有关操作	(73)
实验 2.9 微分方程的解析解与近似解	(85)
实验 2.10 矩阵及其应用	(89)
实验 2.11 直方图与分布函数	(100)
实验 2.12 统计分析的有关计算	(105)
问题与思考	(112)
第 3 章 专题训练	(113)
实验 3.1 三维图形及应用	(113)
实验 3.2 二分法与分割枚举	(119)
实验 3.3 黄金分割法	(125)
实验 3.4 迭代法(I)	(129)
实验 3.5 数值积分法	(135)
实验 3.6 迭代法(II)	(139)
实验 3.7 微分方程数值解法与计算机模拟	(143)
第 4 章 综合实习	(149)
实验 4.1 非线性交调的频率设计问题	(149)
实验 4.2 飞行管理问题	(153)

实验 4.3 最优捕鱼策略问题	(157)
实验 4.4 锁具装箱问题	(161)
讨论题	(163)
附录 1 Mathematica 4.0 的几个主要子菜单	(166)
附录 2 使用 Mathematica 可能出现的错误	(170)
附录 3 Mathematica 命令汇总	(173)
附录 4 几个重要概念的动画程序	(183)

第 1 章 数学实验的软件基础

1.1 Mathematica 系统简介

Mathematica 是一个功能强大的计算机系统。它将几何、数值计算与代数有机结合在一起,可用于解决各种领域内涉及的复杂符号计算和数值计算问题,适用于从事实际工作的工程技术人员、学校教师与学生、从事理论研究的数学工作者和其他科学工作者使用。

Mathematica 能进行多项式的计算、因式分解、展开等;进行各种有理式计算;求多项式、有理式方程和超越方程的精确根和近似根;数值的、一般代数式的、向量与矩阵的各种计算;求极限、导数、积分;进行幂级数展开及求解微分方程等。还可以做任意位数的整数或分子分母为任意大整数的有理数的精确计算,进行具有任意位精度的数值(实、复数值)计算。使用 Mathematica 可以很方便地画出用各种方式表示的一元和二元函数的图形。通过这样的图形,我们常可以立即形象地把握住函数的某些特性。

Mathematica 的能力不仅仅在于上面说的这些功能,更重要的在于它把这些功能有机地结合在一个系统里。在使用这个系统时,人们可以根据自己的需要,一会儿从符号演算转去画图形,一会儿又转去做数值计算。这种灵活性能带来极大的方便,常使一些看起来非常复杂的问题变得易如反掌。Mathematica 还是一个很容易扩充和修改的系统,它提供了一套描述方法,相当于一个编程语言,用这个语言可以写程序,解决各种特殊问题。从以下两个例子可以看到这一系统的简单而实用。

(1) 求具有 100 位有效数字的 π 的近似值,输入命令: `N[Pi,100]`, 执行即可得到。

(2) 要画出放牛娃的草帽,输入命令: `Plot3D[Sin[Sqrt[x^2+y^2]],{x,-5,5},{y,-5,5}]`。

Mathematica 有各种不同的版本,常见的有 Mathematica 1.2 for DOS、Mathematica 2.0 for DOS、Mathematica 2.1 for Windows、Mathematica 2.2 for Windows、Mathematica 3.0 for Windows、Mathematica 4.0 for Windows 等。对于 3.0 与 4.0 的版本,其命令的使用有两种不同的方式:面板填写方式与内部命令书写方式。面板填写方式比较简单,它将各种常用命令做成了与普通的数学书写格式一致的面板,使用者只要选用相应面板,并按格式填写执行即可求解,它使初学者学习 Mathematica 更加容易,但要实现更复杂的操作,则需用另外一种方式——命令书写方式。内部命令书写方式适于各种版本的 Mathematica 系统,相对来说更难于记忆,但更广泛适用。以下我们主要以内部命令书写方式介绍这一系统的使用。

1.2 Mathematica 系统使用入门

1.2.1 系统的算术运算

1. Mathematica 的数及其运算

数常以两种形式出现:精确数与浮点数。Mathematica 的数,除几个常用的数学常数外,

与数的通常表示基本相同。常用数学常数的表示：圆周率 π 用 π 或 Pi 表示, E 表示自然对数的底 $e = 2.71828182\cdots$, Degree 表示角度 1 度, I 表示虚数单位 i , Infinity 或 ∞ 表示无穷大 ∞ 。

加、减、乘、除、乘方的算符依次为 $+$ 、 $-$ 、 $*$ 、 $/$ 、 $^$ 。其中乘可以用空格来代替,减号可用来表示一个负数的符号,并直接写在数的前边。

Mathematica 中数的运算规则与数学中数的运算规则相同,其先后次序由低到高依次为:加(减)、乘(除)、乘方,连续几个同级运算(除乘方外)从左到右顺序进行,乘方则从右到左进行。

注意: Mathematica 中仅用小括号 $()$ 来改变运算次序,其他括号有其具体含义,不可随意使用。

2. 数运算的结果

Mathematica 中数运算的结果依不同情况,有以下二种可能方式:

(1) 整数、分数等,总之,不带有小数点的数,它们所组成的算式,将被系统认可为求精确值;例如: $2/3$ 的结果为 $\frac{2}{3}$; $4/10$ 的结果为 $\frac{2}{5}$ 等等。

(2) 式子中若有一个参与运算的数是浮点数(即带有小数点的数)且式子本身的值又是一个确定的数值, Mathematica 系统将求整个式子的近似值,结果以浮点数形式给出。

例 1.1 求式子 $-2^2 \times \left[3 \times \left(\frac{5}{9} \right)^0 \right]^{-1} \times \left[81^{-0.25} + \left(3 + \frac{3}{8} \right)^{-\frac{1}{3}} \right]^{\frac{1}{2}}$ 的值。

解 可以用以下的 Mathematica 系统书写格式:

```
-2^2 * (3 * (5/9)^0)^-1 * (81^-0.25 + (3 + 3/8)^(-1/3))^(1/2)
```

其输出结果为: -1.33333 。

例 1.2 求式子 $100^{0.25} \times \left(\frac{1}{9} \right)^{-\frac{1}{2}} + 8^{-\frac{1}{3}} \times \left(\frac{4}{9} \right)^{\frac{1}{2}} \pi + \left(\frac{8}{9} \right)^0$ 的值。

解 输入 Mathematica 命令

```
100^0.25 * (1/9)^(-1/2) + 8^(-1/3) * (4/9)^(1/2) * Pi + (8/9)^0
```

执行结果为: 11.5340 。

注意: Mathematica 2.2 及以下版本,对于含有数学常数的式子,则分组依上述规则(1)(2)分别进行运算。即对含有浮点数而不含数学常数的部分依上述规则直接进行;对含有数学常数的项除数学常数外依上述规则进行。例 1.2 中的 Mathematica 语句在 2.2 版本中的运行结果为: $10.4868 + \frac{\text{Pi}}{3}$;又若将上式中 $\frac{4}{9}$ 的次幂改成了 0.5,即输入

```
100^0.25 * (1/9)^(-1/2) + 8^(-1/3) * (4/9)^0.5 * Pi + (8/9)^0
```

则输出结果变为: $10.4868 + 0.333333 \text{ Pi}$ 。若表达式中含有其他未确定值的参数,4.0 版本也将按这种方式进行运算。

3. 数的近似值

精确数转换为浮点数有以下方式:

$N[a]$ ——表示求数 a 的近似值,有效位数取 6 位。这一命令也可写为 $a//N$;

$N[a, n]$ ——求 a 的近似值,有效位数由 n 的取值而给定。

其中 a 为数或为有确定数值的表达式。例如对于例 1.2,要想结果为一个浮点数,只需输入:

$100^{0.25} * (1/9)^{(-1/2)} + 8^{(-1/3)} * (4/9)^{0.5} * \text{Pi} + (8/9)^0 / N$

或

$N[100^{0.25} * (1/9)^{(-1/2)} + 8^{(-1/3)} * (4/9)^{0.5} * \text{Pi} + (8/9)^0]$

结果均为:11.5340。要想得到更精确的结果,比如取 20 位有效数字的结果,只要输入

$N[100^{(1/4)} * (1/9)^{(-1/2)} + 8^{(-1/3)} * (4/9)^{(1/2)} * \text{Pi} + (8/9)^0, 20]$

结果为 11.534030531701735742。注意:若表达式中包含有小数,将得不到正确结果。

1.2.2 代数式与代数运算

1. 赋值与代入

$x = a$ 表示把数 a 赋予 x 。执行该语句后 x 即有定义,其代表一个值为 a 的数,只要不再对 x 赋值或清除,也没有退出过系统, x 恒为此值。用这种方法还可以将 x 的值赋给其他的变量,从而此变量也与 x 一样具有了值 a 。例如:先输入表达式: $x^2 + 3x - 10$, 则运行结果为: $x^2 + 3x - 10$ 。输入 $x = 2$, 运行结果为 2, 说明 x 已赋值成 2。此时,如果我们再输入表达式: $x^2 + 3x - 10$, 则输出结果为 0。这说明此时的 x 已经代表数 2 了,而不再是符号 x 。

命令 $x = .$ 或 $\text{Clear}[x]$, 表示取消对 x 的赋值,它们没有输出结果。一般来说,在使用一些变量前,最好先清除一下,这可以避免变量的以前赋值影响以后的计算结果,这种影响有时容易从运行结果发现,但有时也很难发现(而这是最难以忍受的)。因此,应当养成使用变量前先清除变量以前定义的习惯。

一次清除多个变量可用格式: $\text{Clear}[x, y, z, \dots]$ 。

命令 $\text{expr}/.x \rightarrow a$ 表示把表达式 expr 中的 x 全代换成 a 时的结果,其中 $x \rightarrow a$ 叫做代入规则。代入不改变原表达式,只给出表达式将 x 代换成 a 后的结果。例如:仍用上述的式子: $x^2 + 3x - 10$, 输入: $x^2 + 3x - 10 /. x \rightarrow 2$, 则结果为 0。再输入: $x^2 + 3x - 10$, 其结果仍为: $x^2 + 3x - 10$ 。说明表达式没有改变,或者说 x 仍代表字符或变量 x , 没有具体的值。

赋值可以使计算结果的使用更加简单。请分次连续输入以下各语句,并注意输入与输出,理解赋值、代入与清除:(Mathematica 规定用 $(*)$ 括起来的内容,表示注释而不予执行)

```
p = (1 + x + 3y)^4    (* 2.2 版本及以上版本,同时输入多个语句时,各语句不能在同一行中 *)
p                    (* 4.0 版本不能同时输入多行,否则将以各行结果相乘作为最终结果 *)
x = 2
p
p /. y -> 2
x = .                (* 注意只清除了对 x 的赋值,清除语句无输出结果 *)
(1 + x + 3y)^4
```

所得输出有六个,它们依次对应上述六个有输出结果的语句。赋值这种用法在要引用的结果复杂难写时,会更感方便。请注意使用。

2. 代数式的几个操作函数

(1) 多项式的展开与因式分解

除按一般的算术运算计算外,对多项式还有展开与因式分解的操作,命令分别为:Expand[expr]与Factor[expr]。Expand[expr]表示对表达式 *expr* 作展开运算;Factor[expr]表示对表达式 *expr* 进行因式分解。

例 1.3 对 $(1+x+3y)^4$ 先进行展开,然后再进行因式分解。

解 由命令格式,对所给表达式的展开,可输入如下命令

```
p = Expand[(1 + x + 3y)^4]
```

注意这一语句在展开的同时,还对 *p* 进行了赋值,命令的运行结果(输入 *p* 的执行结果)为

$$1 + 4x + 6x^2 + 4x^3 + x^4 + 12y + 36xy + 36x^2y + 12x^3y + 54y^2 + 108xy^2 + 54x^2y^2 + 108y^3 + 108xy^3 + 81y^4$$

(结果较长)对上述结果再进行因式分解,注意到 *p* 已代表上述结果,则命令可写为

```
Factor[p]
```

运行后则返回到结果: $(1+x+3y)^4$ 。由这里应该可以看到运用赋值的简便。

(2) 化简

Simplify[expr] 表示把表达式 *expr* 化简所得结果。例如:化简 Expand[(1+x+3y)^4]的结果,则命令为:

```
Simplify[Expand[(1 + x + 3y)^4]]
```

注意: Simplify[expr]给出 Mathematica 所求得的最短、最简单形式;Factor[expr]则给出表达式 *expr* 因式分解以后的结果。请分次输入以下命令,弄清 Simplify 与 Factor 的区别:

```
Simplify[x^5 - 1]
```

```
Factor[x^5 - 1]
```

3. 关于解方程

Mathematica 有多个命令可以求解一个方程或方程组,如 Solve、Reduce 等。它们均可以用来求方程的所有精确解。Solve 给出的结果形式为代入规则列表;Reduce 则给出方程解的组合条件表示形式。例如:求解方程: $x^4 - 13x^2 + 36 = 0$,用 Solve 有:

```
Solve[x^4 - 13x^2 + 36 == 0, x]
```

执行的结果为 $\{\{x \rightarrow 3\}, \{x \rightarrow -3\}, \{x \rightarrow 2\}, \{x \rightarrow -2\}\}$ 。用命令 Reduce 则有:

```
Reduce[x^4 - 13x^2 + 36 == 0, x]
```

执行的结果为 $x == 3 \vee x == -3 \vee x == 2 \vee x == -2$,可以看出,除了运算结果的输出形式有所不同外,其他没有太大差别。但若求解带有参数的方程,用 Reduce 则可以给出解的所有可能情况,而 Solve 不行。例如求解方程 $ax = b$ 。二者求出的结果则有很大差别。请读者不妨分别用 Solve 与 Reduce 求解这一方程,并比较二者的求解结果。

有些方程,其解无法用精确数来表示,则 Solve 与 Reduce 给不出精确解,但可以求方程解的近似值。例如对于方程 $x^5 - 4x + 2 = 0$,这时可用命令 NSolve 来求方程的近似解,也可用 Solve 与 N 命令结合求方程的近似值。例如(分次输入)

```
Solve[2 - 4x + x^5 == 0, x]//N
NSolve[2 - 4x + x^5 == 0, x]
```

从所得结果可以看出,NSolve 与 Solve 后再 //N 求出了相同结果。

NSolve 与 Solve、Reduce 还可用来求解方程组。例如

```
Solve[{x^2 + y^2 == 2x * y + 4, x + y == 1}, {x, y}]
```

的结果为 $\left\{ \left\{ x - > -\frac{3}{2}, y - > \frac{3}{2} \right\}, \left\{ x - > \frac{3}{2}, y - > -\frac{3}{2} \right\} \right\}$ 。用另外两个语句 NSolve 与 Reduce 可分别得到解的近似值与解的条件格式。

4. 方程消元

给定方程组 $\begin{cases} f(x, y, z) = 0 \\ g(x, y, z) = 0 \end{cases}$, 对 x 消元,命令的格式为

```
Eliminate[{f[x, y, z] == 0, g[x, y, z] == 0}, x]
```

例如: 给定方程 $x^2 + y^2 + 3xy - 4 = 0$ 与 $x - y = 1$, 则对 y 消元时可使用命令

```
Eliminate[{x^2 + y^2 + 3x * y - 4 == 0, x - y == 1}, y]
```

结果表明为 $-5x + 5x^2 = 3$ 。注意: 不能用这一命令求方程或方程组的解。

5. 数列求和

Sum 语句用来求和,NSum 表示求和的近似值,其命令书写格式分别为

```
Sum[an, {n, nmin, nmax, di}]
```

与

```
NSum[an, {n, nmin, nmax, di}]
```

表示求通项 an 中 n 从 $nmin$ 到 $nmax$, 每隔 di 取一个实数时,所得所有项的和(精确值与近似值),当 di 为 1 时,可以将其省略;当 di 与 $nmin$ 均为 1 时,二者均可省略。

例如,分别用 Sum 与 NSum 求和 $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{100}$ 的命令为(注意分句输入)

```
Sum[1/n, {n, 1, 100}] (* 也可以写为: Sum[1/n, {n, 100}] *)
```

```
NSum[1/n, {n, 1, 100}] (* 也可以写为: NSum[1/n, {n, 100}] *)
```

分别得到和的精确值与近似值。

例 1.4 求和 $\frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \cdots + \frac{1}{100}$ 与 $1 + \frac{1}{1.5} + \frac{1}{2} + \frac{1}{2.5} + \frac{1}{3} + \cdots + \frac{1}{100}$ 的近似值。

解 由于第一个和各项可以看成是通项 $a_n = \frac{1}{n}$ 当 n 分别取 2, 4, 6, ... 的值, 相隔为 2; 第二个和的通项可以看成是当 n 分别取 1, 1.5, 2, 2.5, 3, ... 的值, 相隔为 0.5, 因此由 NSum 命令的格式有(分句输入, 分别求值):

```
NSum[1/n, {n, 2, 100, 2}]
NSum[1/n, {n, 1, 100, 0.5}]
```

分句输入并执行后,分别可得和的近似值为: 2.24960 与 9.75606。

1.2.3 变量与函数

1. 变量名

变量用包含任意多的字母数字表示,其中不能带有多余的空格、标点符号、算符等,且数字字符不能放在变量名的最前面。例如 `xx`、`x35`、`xyz` 是变量名, `5x`、`x * y`、`x □ y` (这里我们以符号 `□` 表示空格)不是变量名。以下对空格用法的详细描述将有助于理解。

空格的使用规定是:

- (1) 两个子表达式间的空格(或换行符)总表示它们相乘。例如 `x □ x` 表示 `x` 与 `x` 的乘积。
- (2) 能明确判定是相乘的地方可以省略空格。例如 `5(2 + 3)` 是一个表达式,其值为 30。
- (3) 算术运算符的前面,圆括号、方括号或大括号的前后等地方,有没有空格或有多个空格都不改变表达式的意义。例如 `5 □ (2 + 3)` 和 `5 □ □ (2 + 3)` 均是表达式,其值也为 30。

2. 系统内部常用的数学函数

系统内部常用的数学函数主要有以下 6 类,它们的函数名为:

幂函数 `Sqrt[x]`——求平方根; `Exp[x]`——以 `e` 为底的指数。

对数函数 `Log[a, x]`——以 `a` 为底 `x` 的对数, `Log[x]` 表示以 `e` 为底的对数。

三角函数 `Sin[x]`——正弦; `Cos[x]`——余弦; `Tan[x]`——正切; `Cot[x]`——余切; `Sec[x]`——正割; `Csc[x]`——余割。

反三角函数 `ArcSin[x]`——反正弦; `ArcCos[x]`——反余弦; `ArcTan[x]`——反正切; `ArcCot[x]`——反余切; `ArcSec[x]`——反正割; `ArcCsc[x]`——反余割。

整数的一些运算: `Mod[m, n]`——求 `m` 除 `n` 所得余数(取模运算); `GCD[n1, n2, ...]`——求所有整数 `ni` 的最大公约数; `LCM[n1, n2, ...]`——求所有整数 `ni` 的最小公倍数。

书写系统内部函数名应注意以下几点:

(1) 系统内部函数名都以大写字母开头,后面字母用小写。如 `Sin`、`Cos` 等。假如当函数名可以分成几段时,每段的头一个字母要大写,后面字母用小写。如 `ArcTan`、`ArcSin` 等。

(2) 函数名是一个字符串,其中不能有易引起歧义的字符。例如,将 `ArcSin` 写成 `Arc □ Sin` 是不合法的。

(3) 函数的参数表用方括号括起来,不能用其他括号。`Sin(x + y)` 表示变量 `Sin` 与 `x + y` 的乘积; `Sin[x + y]` 则表示函数 `Sin` 作用到 `x + y` 上的结果。

(4) 有多个参数的函数,参数之间用逗号分隔。例如 `Log[2, 3]` 表示以 2 为底 3 的对数。

函数与数、函数与函数之间的运算方式和数与式、式与式之间的运算相同。例如:函数与函数的复合方式表现为函数名之间的嵌套。例如: `Sin[x] + Cos[x]`、`Sin[x] * Cos[x]` 分别表示函数 `sin x` 与 `cos x` 的和函数与积函数; `Sin[Cos[x]]` 即表示由函数 `sin u` 与 `u = cos x` 复合而成的函数。例如,求 `sin cos 2.1` 的值,可用如下命令

```
Sin[Cos[2.1]]
```

更多的数学函数,详见附录3。

3. 自定义函数

定义一个函数,在 Mathematica 中可以用以下两种方式: $f[x_]:=expr$ 或 $f[x_]=expr$,其中 $expr$ 为函数 $f(x)$ 的表达式。它们用来定义一个自变量为 x 的函数 $f(x)$,两者之间只有微小的区别。只要不退出系统,则函数 $f(x)$ 的定义必然存在,再次定义 $f(x)$,则 $f(x)$ 的定义更换为新的表达式。 $Clear[f]$ 清除 f 的所有定义内容。 $Save[f]$ 可将 f 的定义保存起来,下次仍可使用。

一个自定义函数,可以像 Mathematica 系统内部函数一样使用,除了要按所定义的函数名书写外,其用法与书写规范与内部函数完全一样。例如,首先定义一个函数 $f(x) = x^3$,然后再求这一函数在 $x = 3$ 时的函数值,则可输入如下命令:

```
f[x_]:=x^3; f[3]
```

结果为: 27。

注意: 定义函数的“ $:=$ ”表示延时定义,即在需要函数的当地进行计算;但“ $=$ ”则表示立即定义,即在定义的同时进行了等号右端表达式的计算。

1.2.4 Mathematica 的绘图初步

1. 一元函数的图形

Mathematica 的基本命令形式为:

```
Plot[f[x], {x, xmin, xmax}]。
```

这一命令用来画出函数 $f(x)$ 当自变量 x 在下限 $xmin$ 与上限 $xmax$ 之间变化时的图形。例如,要画出函数 $\sin x$ 在区间 $[-2\pi, 2\pi]$ 的图形,可用以下命令:

```
Plot[Sin[x], {x, -2Pi, 2Pi}]
```

系统运行结果如图 1-1 所示。

画图时,如果希望按照某种意愿给出结果,此时可在画图命令中加入自定的可选项 $option$ 要求,格式为:

```
Plot[f[x], {x, xmin, xmax}, option]。
```

常用的可选项:

AxesLabel 说明要画图的坐标轴标记,缺省时不标记。 $AxesLabel \rightarrow \{time, temp\}$ 表示坐标轴标记横轴为 $time$,纵轴为 $temp$ 。例

```
Plot[Sin[x], {x, -2, 2}, AxesLabel -> {time, temp}]
```

PlotRange 说明要求的画图范围,缺省时为 $Automatic$,即 $PlotRange \rightarrow Automatic$,表示由 Mathematica 系统自动选定,这时,系统按一定的原则确定作图范围,有时可能切掉图形的某些尖峰。当发现系统切掉了重要的尖峰时,可更换选项重画图形。其他可能的值有: All 表示画出函数的全部情况; $\{y_1, y_2\}$ 表示画出纵坐标在区间 $[y_1, y_2]$ 内的图形; $\{\{x_1, x_2\}, \{y_1, y_2\}\}$ 形式给出横坐标在 $[x_1, x_2]$,纵坐标在 $[y_1, y_2]$ 的函数图形。

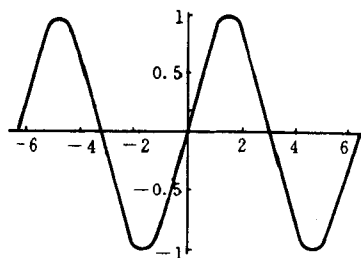


图 1-1

注意:重画的图形仅为没有此选项时图形的局部放大,不改变形状,即使局部有误也如此。

AspectRatio 说明整个图的高宽比,缺省时为 1。可用任何的正实数值以迎合不同的要求。

Axes 指明是否画坐标轴及坐标轴的交点坐标,缺省值为 Automatic。可用 None 说明不画坐标轴,也可用 $\{x, y\}$ 形式的值表示把坐标轴交叉点设在 (x, y) 点的位置。

更多的选项详见附录 3,要查看当前各选项的缺省值,可用命令 `?? Plot`。

一幅图中,可同时画几个函数的图形,格式为: `Plot[{f1[x], f2[x], ...}, {x, xmin, xmax}]`。表示当自变量 x 取值在区间 $[xmin, xmax]$ 内时,在同一坐标系中画出函数 $f_1(x), f_2(x), \dots$ 的图形。例如,将函数 $\sin x, \cos x, \tan x$ 自变量取值在区间 $[-2\pi, 2\pi]$ 内的图形画在同一坐标系中,命令为:

```
Plot[{Sin[x], Cos[x], Tan[x]}, {x, -2Pi, 2Pi}]
```

执行的结果如图 1-2 所示。

2. 平面曲线的参数方程作图

若平面曲线的参数方程为: $\begin{cases} x = x(t) \\ y = y(t) \end{cases} (t_1 \leq t \leq t_2)$ 可用命令: `ParametricPlot[{x[t], y[t]}, {t, t1, t2}]` 作图,绘出图中的横轴为 x 轴,纵轴为 y 轴。例如平面曲线的参数方程为

$\begin{cases} x = \sin^3 t \\ y = \cos^3 t \end{cases} (0 \leq t \leq 2\pi)$, 则 Mathematica 命令为:

```
ParametricPlot[{Sin[t]^3, Cos[t]^3}, {t, 0, 2Pi}]
```

画出的图形如图 1-3 所示(注意 x 为横轴, y 为纵轴)。

ParametricPlot 也有一系列的任选项,常用的与 Plot 命令的任选项相同,有关其选项的缺省值,请用命令 `?? ParametricPlot` 查阅,同时也可以从查阅中了解 ParametricPlot 所有选项的名称写法及其设置值。当然,通过使用,可以了解它们的作用,查字典了解它们的中文意思也可以帮助记忆与理解它们。部分名称及用法详见附录 3。

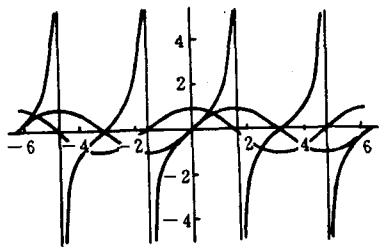


图 1-2

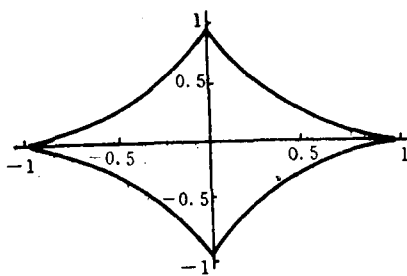


图 1-3

用 ParametricPlot 命令也可以一次将多个参数方程所表示的图形画在同一坐标系中,其命令的格式为: `ParametricPlot[{ {x1[t], y1[t]}, {x2[t], y2[t]}, ..., {t, t1, t2} ]`。

3. 数据点作图

在系统中,数据点用数据表——二、三元数组来表示,因此一组数据用数据点的表表示,如

$\{\{1,3\},\{2,5\},\{5,8\}\}$ 即表示一组数据。在此我们不妨称一组数据为一个数据表。使用命令 ListPlot 可以画出数据表的散点图或由各点顺次连接起来的折线图。格式分别为：

ListPlot[数据表] 或 ListPlot[数据表,PlotJoined->True]

由第二种用法可以看出,ListPlot 也有选项,除上述选项外,Plot 的大多数选项对它都适用。

例 1.5 给定数据表

x	496.8	519.6	623.2	668.1	742.7	854.0	1009.5	1302.4	1754.3	2420.0
y	443.2	453.1	561.4	620.0	682.0	809.6	954.5	1186.0	1582.0	2214.0

试画出这一数据表的散点图与顺次连接各点的折线图。

解 为了后面使用方便,我们将表赋值给 data1,即令 data1 表示这一表,有

```
data1 = {{496.8,443.2},{519.6,453.1},{623.2,561.4},{668.1,620.0},{742.7,682.0},{854.0,
809.6},{1009.5,954.5},{1302.4,1186.0},{1754.3,1582.0},{2420.0,2214.0}}
```

执行后即可画出数据的散点图,命令为

```
ListPlot[data1]
```

命令执行后,可看到坐标系中的散点位置,即数据的散点图。在上述命令中加入一个选项 PlotJoined->True,可以画出将各散点用直线连接起来后的图形

```
ListPlot[data1,PlotJoined->True]
```

关于 ListPlot 的特有选项,详见附录 3,而其选项的当前缺省值,可以用 ?? ListPlot 查阅。

4. 图形重组

为比较所画出的几个图形,常要将这些图形组合显示在同一图中,组合显示的命令为: Show[图形 1,图形 2,...]。它表示将图形 1、图形 2、...中图形显示在一幅图中。注意 Show 所显示的图形应是已经画好的图形。例如:首先用红色画出 $\sin x$ 的图形,并将图形记为 g1,

```
g1 = Plot[Sin[x],{x,-2Pi,2Pi},PlotStyle->{RGBColor[1,0,0]}]
```

(图形略),其次用绿色画出 $\cos x$ 的图形,并将图形记为 g2,

```
g2 = Plot[Cos[x],{x,-2Pi,2Pi},PlotStyle->{RGBColor[0,1,0]}]
```

(图形略),再用蓝色画出 $\tan x$ 的图形,并将图形记为 g3,

```
g3 = Plot[Tan[x],{x,-2Pi,2Pi},PlotStyle->{RGBColor[0,0,1]}]
```

(图形略),最后将它们显示在同一图中有

```
Show[g1,g2,g3]
```

图形如图 1-2 所示。上述过程也可以一次直接用以下命令画出

```
Plot[{Sin[x],Cos[x],Tan[x]},{x,-2Pi,2Pi},
PlotStyle->{RGBColor[1,0,0],RGBColor[0,1,0],RGBColor[0,0,1]}]
```

但有时则需要分别进行。如要将一组数据的散点图与一个函数图画在一起比较,或所画各图形的自变量取值范围不同时,则必须这么做。

1.2.5 数表及其有关操作

1. 表与表的生成

Mathematica 中的表形式上为: $\{a, b, c, \dots\}$, 正如我们已经看到的那样, 它的元素既可以是数, 也可以是表, 甚至可以是其他任何形式的元素。用表可以表示集合, 形式没什么差别。但它到底表示表还是表示集合, 则要根据前后文的用法来判定。

表中元素也可以是表, 如 $\{\{1, 2\}, \{3, 4\}, \{4, 5\}\}, \{\{1, 2, 3\}, \{2, 3, 1\}, \{3, 4, 5\}\}$ 等, 这就是二层表, 二层表中的元素为一表, 常称为子表, 子表中元素还可以是表, 从而有三层表, 四层表等, 二层及二层以上的表称为多层表, 常见的多为单层、二层与三层表。

在 Mathematica 中, 生成一个表可以有多种方法, 以下仅介绍三个有关命令, 其他的参见附录 3。

直接生成: 按顺序写出一个表中的元素, 用逗号隔开并放在一个大括号 $\{\}$ 之中, 即得到一个表。如语句 $tt = \{1, 2, 3, 4\}$ 则表示由 1, 2, 3, 4 按序组成的表, 并同时将此表赋值给 tt 。

通项生成: 其命令格式为: $\text{Table}[\text{expr}, \{n, n1, n2, \text{step}\}]$ 。表示将 n 依次以步长 step 取 $n1$ 到 $n2$ 间的值后, 代入含 n 的表达式 expr 计算所得到的取值表, 例如:

```
Table[1/n^2, {n, 1, 6, 2}]
```

可得到一表, 请注意比较命令与结果, 观察其中 n 的取法。当 $\text{step} = 1$ 时, step 可以省略不写; $\text{step} = 1$ 且 $n1 = 1$ 时, 二者均可省略, 例如(分句输入):

```
Table[1/n^2, {n, 2, 20}]
```

```
Table[1/n^2, {n, 20}]
```

都是合法语句。 step 不等于 1 时, 上述任何一项均不可省略。使用时要注意 $n1 < n2$ 时, step 应取正, $n1 > n2$ 时, step 取负。注意比较以下两个语句的输入与输出(分句输入):

```
Table[1/n^2, {n, 20, 30, 2}]
```

```
Table[1/n^2, {n, 30, 20, -1}]
```

读者也可以违反上述规则, 反其道而行之看一看能得到什么结果。

Table 还可以用来生成多层表, 请用以下命令查看其生成方式。

```
Table[f[i, j], {i, 1, 3}, {j, 4, 6}]
```

迭代生成: 其命令格式为: $\text{NestList}[\text{函数名 } f, \text{初始值 } x, \text{迭代次数 } n]$ 。它的结果为由 $n + 1$ 个元素所组成的表: $\{x, f[x], f[f[x]], \dots, \underbrace{f[f[\dots f[x]\dots]]}_{n \uparrow}\}$ 。输入命令

```
NestList[Sin, 1, 4]
```

看所得结果, 理解 NestList 命令的用法与执行结果。

2. 表的有关操作

元素抽取: $\text{First}[\text{list}]$, $\text{Last}[\text{list}]$, $\text{list}[[n]]$ 分别表示取表 list 的第一个、最后一个、第 n 个

元素;当 list 为多层表时, `list[[n1,n2,...]]` 表示取表 list 的位置为 {n1,n2,...} (即表 list 的第 n1 个子表的 n2 个子表的...) 的元素; `Take[list, 整数 n]` 中 n 可正可负, 正表示取前 n 个, 负表示取后 n 个, `Take[list, {整数 m, 整数 n}]` 取出表 list 的第 m 个到第 n 个元素作成一个新表。

加入元素: `Prepend[list, expr]`, 将表达式 expr 加在原表 list 所有元素的前面; `Append[list, expr]`, 将表达式 expr 加在原表 list 所有元素的后面; `Insert[list, expr, n]`, 将表达式 expr 插在原表 list 的第 n 个位置。

表与表的合并: `Join[list1, list2, ...]`, 表元素间顺序连接合成的表; `Union[list1, list2, ...]` 表表合并, 并删除了重复元素, 按内定顺序排序后的表, 这正是集合的并。

对表的其他常用操作还有许多。如 `Length[list]`, `MemberQ[list, expr]`, `Count[list, expr]`, `FreeQ[list, expr]`, `Position[list, expr]` 等。它们的含义与用法详见附录 3。

注意: (1) 对于表的操作, 一般不改变原表。例如: 置 `tt = {1, 2, 3, 4, 5}`, `Append[tt, 6]` 的执行结果为: {1, 2, 3, 4, 5, 6}, 但此时输入命令 `tt` 查看 `tt` 的情况, 可得执行结果仍为: {1, 2, 3, 4, 5}。另外, 用简短的变量名代替一些较长的式子、表以及后面要介绍的图形等是一个很好的办法, 建议学习使用。

(2) 多层表的操作与单层表相同, 只是所作操作首先是对表的最外层进行的, 如

`{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}[[2]]`

的结果为外层表的前两个元素组成的表 {{1, 2, 3}, {4, 5, 6}}。但要取出子表中元素, 可用以下命令

`{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}[[3, 2]]`

这一命令的结果为 8, 即外层表的第三个元素 {7, 8, 9} 的第二个元素, 或在表中位置为 {3, 2} 的元素。

1.3 Mathematica 流程

1.3.1 比较算符与逻辑运算符

比较算符共有以下 6 个, 它们在 Mathematica 系统中的表示方式及用法如表 1.1 所示:

表 1.1

	等于	大于	小于	小于等于	大于等于	不等于
算符	<code>==</code>	<code>></code>	<code><</code>	<code><=</code>	<code>>=</code>	<code>!=</code>
用法	<code>A == B</code>	<code>A > B</code>	<code>A < B</code>	<code>A <= B</code>	<code>A >= B</code>	<code>A != B</code>

表中的 A、B 可以是数, 也可以是表达式。以上各算式的值均为“真”或“假”, 在 Mathematica 系统中分别用“True”或“False”表示。例如: 输入 `2 > 3`, 执行结果为 False; 输入 `2 < 3`, 则执行结果为 True。

Mathematica 系统可以进行符号运算, 因此应有有关逻辑运算的操作符, 即: 或、与、非操作