

Mathematica

有限元分析与工程应用

鲍四元 编 著

- **工具的先进性**

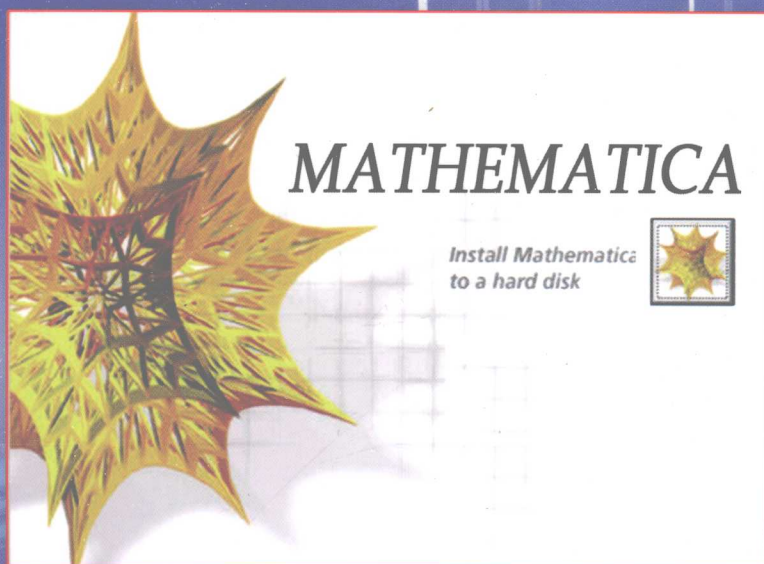
本书采用世界著名的数学软件Mathematica，新颖、简洁、准确，可操作性强。“一个令人诧异的产品：能做计算器，能做编程语言，简直就是天才的产物。Mathematica可以做任何事，只要你有足够的技巧和想象能力……”——Byte.com

- **算例的渐进性**

算例丰富，逐步给出输出结果，读者可对照验证。

- **内容的实用性**

本书根据笔者的实践经验选择内容，包括有限元静力分析与动力分析两部分。注重实用性。



82-39

清华大学出版社

Mathematica 有限元分析与工程应用

鲍四元 编 著

清华大学出版社

北 京

024/1.82-39
B306

内 容 简 介

Mathematica 是世界著名的数学软件。本书不仅介绍有限元分析的基本理论以及实际工程问题的应用,而且强调如何应用 Mathematica 实现算例的求解。对弹簧元、杆元、桁架元、梁元、平面刚架元、三角形元、四边形元和四面体元,循序渐进地介绍了其分析方法。

本书提供了大量的静力问题和动力问题的示例,具有很高的工程应用价值。本书既使用了 Mathematica 的交互应用,也为提高效率而编制了若干模块,而且为了输出更直观,不少问题在后处理上还用到 Mathematica 的图形输出功能。本书的绝大部分程序可从网站资源中下载。

本书理论阐述透彻,语言新颖、简洁、准确,可操作性强,可作为高等院校相关专业的教材或参考书,对于从事实际工作的工程技术人员也可作为重要的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Mathematica 有限元分析与工程应用/鲍四元编著. —北京:清华大学出版社,2010.1

ISBN 978-7-302-21151-8

I. M… II. 鲍… III. 有限元分析—应用软件, Mathematica IV. O241.82

中国版本图书馆 CIP 数据核字(2009)第 177340 号

责任编辑:刘天飞 桑任松

封面设计:杨玉兰

版式设计:北京东方人华科技有限公司

责任校对:周剑云

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:三河市兴旺装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:18.25 字 数:439 千字

版 次:2010 年 1 月第 1 版 印 次:2010 年 1 月第 1 次印刷

印 数:1~4000

定 价:28.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:032421-01

前言

随着计算机技术的发展,有限元技术在各个工程领域正日益显示出强大的生命力。有限元技术是求解各种复杂数学、物理问题的重要方法,利用该方法几乎可以获取任意复杂工程结构的各种信息,还可以直接对工程设计进行各种评判,可以对各种工程事故进行技术分析。有限元技术是一种数值计算技术,它的独特之处在于适用于解析方法不能求解的问题。但是,掌握有限元技术不仅仅是一个纯粹的理论问题,它要求应用人员具有一定的程序设计能力,才能在工程实践领域加以应用,否则只能“望洋兴叹”。本书就是在这样一个背景下完成的,希望使读者能够编制结构清晰、可读性强的有限元程序。

本书的程序语言选择了世界著名的数学软件 **Mathematica**。它是一个令人诧异的产品:能做计算器,能做编程语言,简直就是天才的产物。本书利用了 **Mathematica** 的特色——符号运算功能,也利用了其数值计算功能。

在工科院校的教学,有限元的基础理论是基本内容,在应用上则逐渐分为程序编写与软件应用(如 **ANSYS**、**Marc**、**Algor** 等)两种方式。由于学生数学计算能力的限制,加上有限元自身的复杂性,不少学生在理论上掌握不深刻,程序编写能力不过关。而从本书可知,借助于 **Mathematica** 可以做到:

进行数学推导与验证。**Mathematica** 的符号语言不同于一般程序语言的枯燥难懂,因此可集中精力在数学推导与计算上,而不需要把时间花费在写程序上(主要由于 **Mathematica** 的命令非常高效、简洁,相当于二次开发后的语言)。如果编写出实用的模块,则比 **C** 语言更方便、高效。

不仅学习有限元理论,而且学会写有限元程序。在有限元学习中,由于不善于写程序,导致具体的算例不易得到结果,从而降低了学习理论的兴趣,甚为可惜。数学软件 **Mathematica** 的三五行程序,有时抵得上一般高级语言(如 **FORTRAN**、**C**)程序的上百行,从而提高了效率。“写程序”也变得简单许多。

近六年来,笔者一直利用 **Mathematica** 进行研究与教学工作。越来越多的人认识到,数学软件是学习和应用的“登山缆车”,有助于学生消除畏惧感,提高学习兴趣。另外,各行各业中越来越多的科技人员把 **Mathematica** 作为研究工具。

本书介绍的内容主要包括有限元静力分析与动力分析两部分。前 6 章属于静力部分,包括弹簧元、杆元、桁架元、梁元、平面刚架元、三角形元、四边形元和四面体元等;动力分析包括特征值问题和受迫响应问题两章,其中受迫响应介绍了 3 种积分方法:经典的

纽马克法、威尔逊法及最近提出的精细积分法。“科学计算已经同理论和实验共同构成当代科学研究的三大支柱”，本书既介绍上述基本理论，也包含大量翔实的算例。

本书根据笔者的实践经验选择内容，注重实用性。本书适合于初学者，算例中逐步给出输出结果，读者可对照验证。另外，对相关领域从事有限元程序设计与应用的技术人员也有指导作用。

在本书出版之际，感谢西北工业大学工程力学系邓子辰教授的指导，华中科技大学机械学院黄永安博士后的策划、激励与帮助，也感谢笔者所在单位苏州科技学院土木系提供的优良环境。最后感谢清华大学出版社的合作，还要特别感谢家人的支持与鼓励。

由于笔者的水平与时间有限，书中难免存在错误和不当之处，恳请读者予以指正。

编者

目 录

第 1 章 绪论	1	3.1.3 平面刚架元	44
1.1 有限元方法的步骤	1	3.2 实例	48
1.2 用于有限元分析的 Mathematica 模块	2	3.3 小结	102
1.3 Mathematica 在有限元分析中 使用的命令指南	3	第 4 章 平面问题中的三角形单元	103
1.3.1 Mathematica 软件简介	3	4.1 双线性三角形单元	103
1.3.2 Mathematica 的特点	4	4.1.1 双线性三角形单元的 局部坐标	103
1.3.3 本书常用的 Mathematica 命令	4	4.1.2 采用 Mathematica 确定形 函数	104
第 2 章 弹簧元与杆元	12	4.1.3 程序编制的若干要点	106
2.1 平面弹簧元、杆元的简介与 比较	12	4.1.4 实例	112
2.1.1 平面弹簧元、杆元的简介	12	4.2 二次三角形单元	121
2.1.2 杆元的形函数	12	4.2.1 采用 Mathematica 确定形函数	121
2.1.3 单元刚度矩阵与基本方程	14	4.2.2 程序编制时相关的模块	123
2.1.4 模块分析	16	4.2.3 实例	125
2.2 实例	18	4.3 小结	128
2.3 小结	35	第 5 章 平面问题中的四边形单元	129
第 3 章 平面桁架元、梁元与 平面刚架元	36	5.1 双线性四边形单元	129
3.1 平面桁架元、梁元与平面 刚架元的简介	36	5.1.1 双线性四边形单元的 局部坐标	129
3.1.1 平面桁架元	36	5.1.2 采用 Mathematica 确定形 函数	129
3.1.2 梁元	40	5.1.3 程序编制的若干要点	131
		5.1.4 实例	137
		5.2 二次四边形单元	152

5.2.1 采用 Mathematica 确定形函数	153	第 8 章 特征值问题	217
5.2.2 程序编制时相关的模块	154	8.1 特征值问题的基本理论	217
5.2.3 实例	156	8.1.1 动力学系统的方程	217
5.3 单元刚度矩阵的解析形式	165	8.1.2 特征值问题	217
5.3.1 直接积分方法	165	8.2 梁单元和矩形实体单元的特征值	218
5.3.2 高斯数值积分方法	167	8.2.1 集中质量梁单元的特征值和特征矢量	218
5.4 小结	169	8.2.2 四结点四边形单元求解平面实体的特征值问题	224
第 6 章 三维线性四面体单元	170	8.2.3 八结点四边形单元求解平面实体的特征值问题	228
6.1 三维线性四面体单元的简介	170	8.3 小结	232
6.1.1 四面体单元的局部坐标	170	第 9 章 受迫振动的动力响应	233
6.1.2 采用 Mathematica 确定形函数	172	9.1 动力响应的一般解法	233
6.1.3 程序编制的若干要点	174	9.1.1 纽马克算法	234
6.2 实例	179	9.1.2 威尔逊算法	234
6.3 小结	191	9.1.3 精细积分法	235
第 7 章 矩形薄板弯曲单元	192	9.2 受迫振动的一些实例	237
7.1 薄板的基本概念和基本方程	192	9.2.1 分布式质量梁的受迫振动——纽马克算法和精细积分法	237
7.1.1 薄板的基本概念和基本假定	192	9.2.2 矩形实体的受迫振动分析——纽马克算法	247
7.1.2 弹性薄板弯曲的小挠度理论	193	9.2.3 矩形实体的受迫振动分析——威尔逊算法	251
7.2 矩形薄板单元 ACM	198	9.2.4 矩形实体的受迫振动分析——精细积分法	253
7.2.1 位移模式	198	9.3 小结	258
7.2.2 单元应变、应力和内力	204	附录 A 等效节点荷载	259
7.2.3 单元刚度矩阵	207	附录 B 一个含有前、后处理的完整四边形单元应用程序	261
7.2.4 节点荷载列阵和位移边界条件	208	附录 C Mathematica 的常用函数	278
7.3 小结	216	参考文献	283

第 1 章 绪 论

有限元法是一种求解偏微分方程的方法。这一方法首先需要对方程所在空间进行离散化。所谓离散化,是将一个大的区域划分成一些结构简单、形状任意的局部小区域(即有限单元)。在离散化的同时,偏微分方程也要作相应的变化,即每个子区域上的偏微分方程被转化为矩阵形式的方程(往往是线性的)。这些矩阵方程把单元中某些特定点(即节点)上的已知量(称为输入)与同一点上的未知量(称为输出)联系起来。

然后,需要对这些矩阵方程求解。方法有两种,一种方法是将子区域上的矩阵方程按节点叠加起来,组装得到一个总体矩阵方程而求解,这种求解对应于整体刚度矩阵和整体质量矩阵;另一种方法是采用“逐个单元”法求解,以避免产生整体矩阵或其元素数目巨大带来的问题。本书主要使用第一种方法,即组装成整体矩阵的方法。

本书介绍杆、梁、桁架、平面刚架、平面实体等结构,涉及弹簧单元(具体的“单元”可简称为“元”,此处的弹簧单元可简称为“弹簧元”,其他类似)、杆单元、梁单元、刚架单元、三角形单元、四边形单元、四面体单元等,这些单元可按函数插值的次数分为两类,如线性的和二次的。在各种单元的介绍中都进行了一定的阐述和推导,在实现这些推导时,数学软件 **Mathematica** 起到了重要的作用。不仅如此,在应用有限元方法求解各章所附的例题中,数学软件实现了具体的运算及数值、图形输出。**Mathematica** 的符号运算功能大大地加快了解决问题的速度。

此外,本书讨论的结构都是线弹性的。

通过本章的基本介绍和实例演示,可使读者对 **Mathematica** 的使用和功能有一个初步的了解,有助于读者消除对 **Mathematica** 的陌生感,加快 **Mathematica** 的上手速度。

本书所有章节的例子都是基于 **Microsoft Windows XP** 系统和 **Mathematica 4.0**。

1.1 有限元方法的步骤

有限元方法主要用于解决工程问题,它是一种数值近似方法。本书主要解决结构工程问题,但有限元方法也适用于工程中的其他领域。对于线弹性结构的静力问题,使用有限元方法的解题步骤往往包括以下 6 步。

(1) 离散区域——这个步骤需要把结构分解为若干单元和节点。对于离散系统,如桁

架和刚架，由于结构已经离散化，故只有一种离散结果。而对于连续系统，如实体结构和板壳，则需主观上决定离散到什么程度。而且离散得越细，结果往往会越精确。本书为避免计算规模过大，手动完成这一步。

(2) 建立单元刚度矩阵——建立域内的单元刚度矩阵。各单元的刚度矩阵通过调用 Mathematica 模块建立。

(3) 集成整体刚度矩阵——把各单元的刚度矩阵集成后得到整体刚度矩阵。这个过程按照“对号入座”的方法，并编制相应的 Mathematica 模块。

(4) 引入边界条件——设置待求解问题中的已知条件，如支座位移、外加荷载、给定位移等。这些条件主要通过手工添加。

(5) 解方程——解有限元方法对应的基本方程。这一步骤主要通过 Mathematica 中的命令 Solve 得到线性方程组的解。在常规有限元解法中，需要按照边界条件分解整体刚度矩阵，然后用高斯消去法求解修正的方程组。这里由于 Mathematica 的功能，可以免去这些步骤，从而得到大大简化。

(6) 后处理——得到额外的信息。如支反力、单元节点力、单元应力和内力图。该步骤主要由 Mathematica 实现。

从上述步骤可知，解题中使用了 Mathematica 和手动操作相结合的方法。使用手动操作的步骤是离散化(第 1 步)和设置边界条件(第 4 步)，其他步骤中繁琐的计算都由 Mathematica 实现。

1.2 用于有限元分析的 Mathematica 模块

本书涉及的资源文件包含了若干 Mathematica 模块。主要的模块名称如下(具体的功能及详细使用方法参考各章的内容)：

- GenerateSpringKm, AssembleSpringKm, SpringElementForce;
- GenerateLinearRodKm, GenerateLinearRodKm, RodElemForce;
- GenerateQuadraticRodKm, AssembleQuadraticRod;
- TrussElement, TrussAssemble, TrussElemStress;
- BeamElement, BeamAssemble;
- PlaneElemKm, PlaneAssemble;
- GenerateLinearTriangKm, AssembleLinearTriang, GenerateQuadTriangKm, AssembleQuadTriang;
- GenerateLinearQuadKm, GenerateQuadraticQuadKm, LinearQuadEleStress, Elem-

PrincipleStress, AssembleLinearQuadKm, AssembleQuadraticQuad;

- GenerateTetrahedronKm, AssembleTetrahedron;
- GenerateQuadraticQuadKm。

1.3 Mathematica 在有限元分析中使用的命令指南

Mathematica 已经广泛应用到科研的各个方面,下面分别介绍 Mathematica 软件系统和 Mathematica 的特点,然后介绍本书所用到的常用 Mathematica 命令。

1.3.1 Mathematica 软件简介

在符号计算软件出现以前,数值计算方法总是以数值计算为基础,每步运算的输入是数值,输出也是数值,因此许多问题的计算方法也都是基于数值计算。这种方法可以反映具体计算目标的某一状态或不同状态的具体数值,但很难反映计算目标的全部性质。随着 Mathematica、Maple 等能进行符号计算的软件的出现,给符号计算带来了极大的方便,以前靠手工难以完成的符号计算,现在可以准确、快速地完成。许多学者在各自的科学研究中已成功地运用相关符号计算软件解决了许多问题,可以通过对表达式的分析,来研究计算目标的各类性质。

Mathematica 是一款功能强大的计算机应用软件,由美国 Wolfram Research 公司开发,自 1998 年推出 Mathematica 1.0 后,在计算技术领域引起了很大震动。Mathematica 是一个完全集成环境下的符号运算系统,具有强大的数值运算功能、符号计算功能和绘图功能,如今 Mathematica 已广泛应用于数学、物理学、力学、化学以及工程领域。

Mathematica 处理符号的能力与处理数字的能力一样强,允许用户做多种数学运算,可以对多项式进行展开、因式分解,可以对其中的元素进行提取,也可以进行求积、求和、求导、积分与微分等运算,因此我们可以运用 Mathematica 进行符号运算。

Mathematica 可以做许多符号演算工作:各种有理式的计算;求多项式、有理式方程和超越方程的精确解和近似解;一般代数式的向量、矩阵的各种计算;求极限、导数、积分,幂级数展开;求解某些微分方程等。Mathematica 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,做具有任意位精度的数值(实、复数值)的计算。所有 Mathematica 系统内部定义的整函数、实(复)函数也具有这样的性质。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。通过这些图形,我们可以形象地把握住函数的某些特性,而这些特性一般很难从函数的符号表达式中看出来。

1.3.2 Mathematica 的特点

Mathematica 是世界上通用计算系统中最强大的系统。自从 1988 年发布以来,它已经对如何在科技和其他领域运用计算机产生了深刻的影响。Mathematica 数学软件是一款集成化的计算机软件,它的主要功能有三个方面:符号运算、数值计算和图形绘制。

在符号运算方面,Mathematica 能够进行多项式的各种运算(四则运算、合并、展开、因式分解等);能够做有理表达式的许多计算;能够解多项式方程和一些有理方程、超越方程的解析解和数值解;能够做微积分中的许多运算(极限、导数、不定积分和定积分、幂级数展开和幂级数的运算、求解微分方程等);能够做向量和矩阵的一系列运算。

在数值计算方面,Mathematica 能够做精确的数值计算,也能够解任意精确度的近似解;能够做实数计算,也能在复数范围中进行复数运算。

在图形绘制方面,Mathematica 能够绘制平面图形(直角坐标作图和极坐标作图,还有参数方程作图),也能够绘制三维的曲线和曲面图形。

Mathematica 软件的特点如下:

(1) 内容丰富,功能齐全。Mathematica 能够进行初等数学、高等数学及工程数学的各种数值计算和符号运算。特别是它的符号运算功能,给公式的推导带来了极大的方便。它有很强的绘图能力,能方便地画出各种美观的曲线和曲面,甚至可以进行动画设计。

(2) 语法简练,编程效率高。Mathematica 的语法规则简单,语句简练。和其他高级语言(如 C、FORTRAN 语言)相比,其语法规则和表示方式更接近数学运算的思维和表达方式。用 Mathematica 编程,用较少的语句,就可以完成复杂的运算和公式推导等任务。

(3) 操作简单,使用方便。Mathematica 命令易学易记,运行也非常方便。用户可以和 Mathematica 进行交互式的“对话”,逐个进行命令的执行;也可以进行“批量”处理。

(4) 和其他语言交互。Mathematica 和其他高级语言,如 C、FORTRAN 语言等能进行简单的交互。可以调用 C、FORTRAN 语言等的输出并转化为 Mathematica 的表示形式,也可以将 Mathematica 的输出转化为 C、FORTRAN 语言的表示形式。

Mathematica 具有能够完成符号运算、数学图形绘制,甚至动画制作等多种操作的强大功能,并且具有所见即所得的数学运算按钮,所以 Mathematica 软件应用非常广泛。

1.3.3 本书常用的 Mathematica 命令

下文给出 Mathematica 的一个简短的使用指南,此处并没有全面介绍 Mathematica,而是描述了在本书中使用到的一些基本命令。

假定读者已经成功启动了 Mathematica 开发环境。在 Mathematica 中，输入表达式后，按下 Shift+Enter 键，这时系统开始计算，然后输出计算结果，并且系统自动对第 n 次输入和输出附上次序标识 In[n]和 Out[n]，注意输入标识 “In[n]:=” 是计算后才出现的，不需要用户输入。

1. Mathematica 进行算术运算与函数运算

如下所示，输入标量和简单的运算是很简单的。

```
In[1]:= 3*4+5
Out[1]= 17

In[2]:= Cos[30*Pi/180]
Out[2]=  $\frac{\sqrt{3}}{2}$ 

In[3]:= x=4
Out[3]= 4

In[4]:= 2/Sqrt[3+x]
Out[4]=  $\frac{2}{\sqrt{7}}$ 

In[5]:=N[ $\pi$ ,50]
Out[5]= 3.1415926535897932384626433832795028841971693993751
```

 **注意：** Mathematica 的基本运算加、减、乘、除和乘方的符号分别为+、-、*、/、^。乘法既可以用 “*” 表示，又可以用空格表示，如 $2\,3=2*3=6$ 。

利用 Mathematica 函数时，以下几点一定要牢牢记住：

(1) Mathematica 中大小写是有区别的，如 Name、name、NAME 等是不同的变量或函数名。

(2) 系统所提供的大部分功能以系统函数的形式给出，内部函数一般写全称，而且一定是以大写英文字母开头，如 Sin[x]、Sqrt[x]。

(3) N[]是 Mathematica 近似运算，N[]语法如下：

N[表达式]	(*可求 5 位有效数字的近似值*)
N[表达式,n]	(*可求 n 位有效数字的近似值*)

(4) 如果不让 Mathematica 输出运算数据，可以在命令行的结尾输入分号，如下面的示例。如果命令行结尾没有分号，Mathematica 运算将显示输出内容。

```
In[6]:= y = 32;
In[7]:= z = 5;
In[8]:= x = 2*y - z;
In[9]:= w = 3*y + 4*z
Out[9]= 116
```

2. Mathematica 中用户的自定义函数

在 Mathematica 中, 用户的自定义函数可分为两种。

1) 不带附加条件的自定义函数

在 Mathematica 中自定义函数, 语法如下:

$f[x_]:=$ 关于 x 的表达式

“ $:=$ ”为定义符号, 定义符号左边是函数名和方括号内说明的自变量, 右边是函数的表达式, 定义式左边方括号内自变量后紧跟的下划线符号“ $_$ ”是不能省略的。

例 1.1 求函数 $y = \frac{-1+6x}{4+3x}$ 的函数值 $y(3)$ 、 $y(x+4)$ 。

解 Mathematica 程序如下:

```
In[1]:= y[x_] := (-1 + 6*x)/(4 + 3*x)
In[2]:= y[3]
Out[2]= 17/13
In[3]:= y[x + 4]
Out[3]= (-1+6(x+4))/(4+3(x+4))
```

2) 带附加条件的自定义函数

带附加条件的自定义函数, 语法如下:

$f[x_]:=$ 关于 x 的表达式/;条件

或者

$f[x_]:=$ If[条件, 满足条件时的表达式, 不满足条件时的表达式]

例 1.2 定义函数 $f(x) = \begin{cases} 0 & \text{当 } -1 < x \leq 0 \text{ 时} \\ x^2 & \text{当 } 0 < x \leq 1 \text{ 时} \\ 3-x & \text{当 } 1 < x \leq 2 \text{ 时} \end{cases}$, 并计算 $f(-0.5)$ 、 $f(0.5)$ 、 $f(1.5)$ 。

解 Mathematica 程序如下:

```
In[1]:= f[x_] := 0 /; -1 < x <= 0 (* 定义 -1 < x ≤ 0 时的分段函数 *)
In[2]:= f[x_] := x^2 /; 0 < x <= 1 (* 定义 0 < x ≤ 1 时的分段函数 *)
```

```
In[3]:= f[x_] := 3 - x /; 1 < x <= 2 (* 定义 $1 < x \leq 2$ 时的分段函数 *)
In[4]:= f[-0.5] (* 调用 f, 得到自变量为-0.5时的函数值 *)
In[5]:= f[0.5] (* 调用 f, 得到自变量为 0.5 时的函数值 *)
In[6]:= f[1.5] (* 调用 f, 得到自变量为 1.5 时的函数值 *)

Out[4]= 0
Out[5]= 0.25
Out[6]= 1.5
```

结果输出 $f[-0.5]$ 、 $f[0.5]$ 和 $f[1.5]$ 对应的值 0、0.25、1.5。

 **注意：** 上述命令行中出现了在 Mathematica 中添加的注释，即用(*.....*)的形式，其中省略号代表注释的内容。

3. Mathematica 的 Help 命令

使用 Mathematica 中的 Help 命令可以获得 Mathematica 命令的详细使用方法。下面的示例给出了用 Help 命令获取 Table 命令的使用方法。

选择 Help 菜单下的 Help Browser 命令，出现 Help Browser 对话框，在 Go To 文本框中输入 Table，然后在右侧单击 Table 项，便可以获得如下帮助：

```
Table[expr, {imax}] generates a list of imax copies of expr.
Table[expr, {i, imax}] generates a list of the values of expr when i runs
from 1 to imax.
Table[expr, {i, imin, imax}] starts with i = imin.
Table[expr, {i, imin, imax, di}] uses steps di.
Table[expr, {i, imin, imax}, {j, jmin, jmax}, ...] gives a nested list.
The list associated with i is outermost.
```

再单击 Further Examples，可以获得 Table 的使用实例，如下所示：

```
Here are the squares of the first 10 integers (* 给出前 10 个自然数的平方 *).
In[1]:= Table[i^2, {i, 10}]

Out[1]= {1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

This gives the cubes of the integers from -3 to 3 (* 给出-3 至 3 之间的整数的立方 *).
In[2]:= Table[j^3, {j, -3, 3}]

Out[2]= {-27, -8, -1, 0, 1, 8, 27}
```

4. 矩阵运算

下面的示例展示了如何输入矩阵并实现简单的矩阵运算。

```
In[1]:= M = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
In[2]:= MatrixForm[M]

Out[2]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$


In[3]:= K = {{2}, {0}, {-3}};
In[4]:= w = M.K(* 计算两矩阵 M 和 K 的乘积 *);
In[5]:= Print["w=", w]
In[6]:= Print["w=", MatrixForm[w]]

Out[5]=w={{-7}, {-10}, {-13}}
Out[6]=w=

$$\begin{pmatrix} -7 \\ -10 \\ -13 \end{pmatrix}$$


In[6]:= L = Inverse[{{6, -2}, {-2, 4}}](* 求矩阵的逆 *);
In[7]:= MatrixForm[L]

Out[7]//MatrixForm=

$$\begin{pmatrix} \frac{1}{5} & \frac{1}{10} \\ \frac{1}{10} & \frac{3}{10} \end{pmatrix}$$


In[8]:= a = L[[1, 1]];
In[9]:= b = L[[1]];
In[10]:= Print["a=", a]
In[11]:= Print["b=", b]

Out[10]=a= $\frac{1}{5}$ 
Out[11]=b={ $\frac{1}{5}$   $\frac{1}{10}$ }

In[12]:= Transpose[w] (* 求矩阵的转置 *)

Out[12]= {{-7, -10, -13}}
```



- 注意:**
- ① 矩阵与矩阵的乘法用符号“.”来表示, 而不能用“*”。
 - ② `Inverse[ ]` 可用来求矩阵的逆矩阵。
 - ③ `Transpose[ ]` 可用来求矩阵的转置矩阵。
 - ④ `MatrixForm[ ]` 可使矩阵以标准形式输出。
 - ⑤ 对于一个矩阵 P 而言, $P[[i, j]]$ 表示矩阵 P 的第 i 行、第 j 列的元素, $P[[i]]$ 表示矩阵的第 i 行。
 - ⑥ 和其他计算机语言一样, 在 Mathematica 中, `Print[ ]` 表示结果的输出格式。

5. Mathematica 的作图命令

Mathematica 软件不但可以进行符号运算, 还可以进行画图。

1) 作一元函数图形

在直角坐标系中, 利用 Mathematica 作一元函数图形用下列基本命令:

`Plot [函数表达式, {自变量, 自变量上限, 自变量下限}]`

例 1.3 绘制函数 $y = 3^x, x \in [-2, 2]$ 的图形。

解 Mathematica 程序及运行结果如下:

```
In[1]:= Plot[3 ^ x, { x, -2, 2 }]
```

8

6

4

2

-2

-1

1

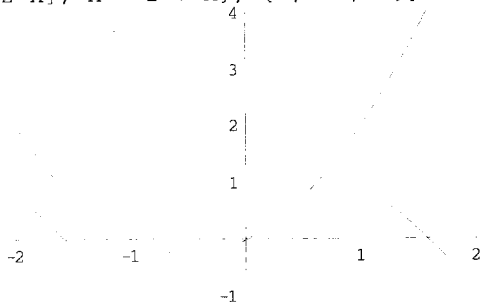
2

```
Out[1]= -Graphics -
```

例 1.4 将函数 $y = \sin 2x$ 和 $y = x^2 + x, x \in [-2, 2]$ 的图形绘制在同一直角坐标系内。

解 Mathematica 程序及运行结果如下:


```
In[2]:= Plot [{Sin[2 x], x ^ 2 + x}, {x, -2, 2}]
```



```
Out[2]= Graphics
```

2) 作数据集的图形

Mathematica 用于绘制数据集的图形命令，与前面介绍的绘制函数图形的命令是相似的。

(1) 绘出在 x 的值为 $1, 2, \dots$ 时, $y_1, y_2, \dots$ 的图形, Mathematica 程序如下:

```
ListPlot[{y1, y2, ...}]
```

(2) 绘制离散点 (x_i, y_i) 的图形, Mathematica 程序如下:

```
ListPlot[{x1, y1}, {x2, y2}, ...]
```

(3) 把离散点连成曲线, Mathematica 程序如下:

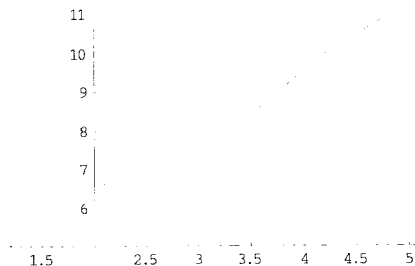
```
ListPlot[{y1, y2, ...}, PlotJoined -> True]
```

```
ListPlot[{x1, y1}, {x2, y2}, ..., PlotJoined -> True]
```

例 1.5 已知数组 (1.2, 5)、(2, 6.7)、(3.4, 8.2)、(4.2, 10)、(5, 11.3), 作出离散点图, 并把这些点用线连接起来。

解 Mathematica 程序及运行结果如下:

```
In[1]:= ListPlot[{1.2, 5}, {2, 6.7}, {3.4, 8.2}, {4.2, 10}, {5, 11.3}],  
PlotJoined -> True]
```



```
Out[1]= Graphics
```