序　　言

随着金融市场的开放，新型金融产品层出不穷，国债回购、浮动利率债券、可转换债券等新品种纷纷出现，金融期货、股指期权也是呼之欲出。与发展迅速的金融产品相比，国内对于金融衍生品的研究还很滞后。随着市场化程度的加深，金融机构和投资者持有大量不同类型的金融资产，其中风险不可小视，由于缺乏有效的手段控制风险，金融业在发展中也出现了许多问题。这就要求我们必须冷静下来，认真研究如何管理金融资产，如何测量这些资产风险，提高金融业抗击风险的能力。其中一条有效的路径就是借鉴市场经济发达国家的成功经验，然后和我国的国情相结合，创造出适合我国国情的新型风险管理系统。

本书主要介绍金融从业人员碰到的一些常见问题，覆盖的范围比较广，也有一定的深度。本书的编著主要由青年教师张树德博士完成，他近年来一直从事金融领域的研究工作，特别在用计算机软件解决金融问题方面较有心得。我非常乐意将该书推荐给金融从业人员、研究人员以及金融领域的爱好者，并相信本书对大家了解和掌握现代金融理论与实务，将发挥很好的作用。

陆沪根

前　言

近年来，新型的金融工具不断被开发出来，使用这些工具可以有效地对冲经济风险。经济风险的复杂性要求不断开发更多的金融工具，而随着计算机速度的提高，复杂的金融交易策略和风险管理方法得以实现。

金融学更是一种定量技术，特别是在投资实务领域，对资产组合及金融工具进行定量化分析几乎已经成为金融从业人员必备的基本功。只有通过定量分析才能把风险真正地搞清楚，而且形成一个风险管理框架，所有工作都在框架的指导下展开，最后形成一个非常复杂而庞大的定价与风险管理体系，这个体系由于汇集了众多研究者的智慧而不断完善，真正体现出金融分工的优越性。

金融业开放必然带来新的投资策略与风险管理方法，依靠传统的个人英雄式决策与风险管理模式无法容纳不同背景甚至不同年代投资家的智慧积累，必然被新型交易与风险管理系统所取代，就像 18 世纪机器生产必然淘汰手工作坊式的生产一样。曾几何时，坐庄炒作模式成为股票市场的主流获利方法，市场在淘汰了一批又一批投资者后，市场盈利模式已经悄然发生变化，研究人员正从边缘化走向盈利的核心，拥有顶级的研究人员成为金融机构的宝贵财富。在投资领域分工不断地被细化的情况下，定量化投资方法也衍生出不同的投资策略，从名目繁多的对冲基金品种就可以知道投资策略也是万花筒，只要深入研究，都可以获得不菲的回报。

为了解决现代金融中的计算问题，MathWorks 公司集结了一批优秀的金融研究开发人员，开发了 MATLAB 金融工具箱，该工具箱把金融学中的问题转化为一个个模块，几乎涵盖了所有金融问题。随着版本的更新，其功能也在不断扩大，从基本时间序列格式处理、时间序列统计到 GARCH 模型、资产组合理论运用、衍生产品数值计算、债券类产品收益率计算、利率期限结构、奇异期权等，显示了强大的金融产品定价和风险管理功能。

在欧美 MATLAB 现在已经成为金融工程人员的密切伙伴，世界上超过 2000 多家金融机构运用 MATALB 来管理公司资产，国际货币基金组织、摩根斯坦利等顶级金融机构都是 MathWorks 公司的客户，借助于 MATLAB 强大的运算平台实现和其他软件之间的数据交换，显示出了非常优良的通融性。

很多刚刚接触 MATLAB 的同志对于编写程序存在畏难情绪，实际上 MATLAB 开发者已经考虑到了这一点，MATLAB 的语法非常简单，程序编写非常容易，戏称为草稿纸式的程序语言，没有编程基础的研究者也可以很快地编写出程序。鉴于 MATLAB 的开放性，用

户可以对 MATLAB 中的程序进行二次开发，使之更加符合自己的需要。

本书得到了我的导师俞自由教授的鼓励与支持，同时也得到了股指期货专家刘仲元老师的指导。上海金融学院张树义教授等参加了本书部分章节的编写工作；浦东新区毛力熊副教授毕业于华东师范大学数学系，对金融学研究颇有造诣，参加了本书部分章节的编写；徐全勇博士给出了许多非常好的意见与建议，在此一并表示衷心感谢。

作者相信MATLAB很快会在中国金融界得到广泛运用，本书只是起到抛砖引玉的作用。由于作者水平所限，书中难免存在错误之处，希望读者提出宝贵意见。

张树德

目　　录

第 1 章　MATLAB 运行环境及金融运用

本章介绍 MATLAB 的产生背景，列举了 MATLAB 金融工具箱的基本内容，最后介绍 MATLAB 在金融领域的运用。

1.1　MATLAB 介绍

1.1.1　MATLAB 的产生背景

MATLAB 诞生于 20 世纪 70 年代，是由美国新墨西哥大学计算机系主任 Cleve Moler 博士创造的，当时由于没有合适的软件进行线性代数计算，于是就开发出了 MATLAB 软件。MATLAB 是由 Matrix(矩阵)和 Laboratory(实验室)两个英文单词的前 3 个字母拼写而成。1983 年春天，Cleve Moler 在斯坦福大学访问时，结识了软件工程师 John Little，随后 Cleve Moler、John Little 和 Steve Bangert，用 C 语言共同开发了第二版 MATLAB，使得 MATLAB 不仅具有数值计算功能，而且具有数据可视化功能。

1984 年，Cleve Moler 和 John Little 成立了 MathWorks 公司，推出了商业版 MATLAB。1993 年 MathWorks 公司推出了 MATLAB 4.0 版本，1995 年又推出了 MATLAB 4.2C 版本，1997 年推出了 MATLAB 5.0，2000 年 10 月推出了 MATLAB 6.0，2002 年 8 月 MathWorks 公司向市场推出了 MATLAB 6.5 版本。MATLAB 的每一次版本升级都增加了大量数值计算功能，附带了几十个工具箱，在数值计算、符号计算和图形处理功能上的大大加强，界面越来越友好。2005 年 8 月推出了 MATLAB 7.10 版本。

MATLAB 推出后不久就在科学计算领域站稳了脚跟，20 世纪 90 年代以来，MATLAB 已成为国际公认的优秀计算软件，在大学及业界得到广泛运用。在欧美大学中，应用代数、数理统计、自动控制、电子信号处理、模拟和数字通信、时间序列分析、动态系统仿真等课程的教科书都把 MATLAB 作为配套教材，MATLAB 成为大学生、硕士生和博士生必备的基本工具。

MATLAB 长于数值计算，能处理大量数据，而且效率很高。MathWorks 公司在此基础上开拓了符号计算、文字处理、可视化建模和实时控制能力，增强了 MATLAB 的市场竞争力。MATLAB 作为计算工具和科技资源，可以扩大科学研究范围，提高工程技术效率。

1.1.2 MATLAB 语言的优点

1. 强大的计算功能

MATLAB 4.0 以上各个版本，不仅在数值计算上保持领先，而且开发出了自己的符号运算功能，功能上不逊于 MatCAD、Mathematic 等软件，让研究人员摆脱了烦琐的软件学习过程，可以进行矩阵变换、多项式运算、微积分运算、线性与非线性方程求解、常微分方程求解、插值与数值拟合、统计、时间序列、金融衍生产品定价、资产组合分析、固定收益证券定价和风险管理等。MATLAB 产品分为以下几大类。

- 数据分析；
- 数值和符号计算；
- 工程与科学绘图；
- 控制系统设计；
- 数字图像信号处理；
- 金融计算；
- 建模、仿真、模型开发；
- 应用开发；
- 图形图像界面开发。

2. 简单易学

一般语言编写程序、调试程序需要经过五个步骤：编辑、编译、连接、执行、调试，各个步骤之间是顺序关系，编程过程就是在它们之间进行循环。MATLAB 语言与其他语言相比，较好地解决了上述问题。MATLAB 软件是解释性语言，调试程序的手段非常丰富，速度也快，把编辑、编译、连接和执行融为一体，在同一界面上操作，可以快速查出输入程序中的书写错误、语法错误，提高了用户编写程序的效率。MATLAB 在运行时，可以直接输入命令语句，包括调用 M 文件语句，每输入一条命令，程序就会立即对其进行处理，完成编译、连接、运行全过程。再如，将 MATLAB 源程序编译为 M 文件，编辑后源程序可以立即运行，如果有错，用户界面上也会给出详细的出错信息。MATLAB 软件易懂易学，允许数学形式编写代码，比 Basic、FORTRAN 和 C 语言更加符合书写习惯，它的操作符和功能函数就是数学书上的简单的英文表达式。MATLAB 还拥有强大的帮助系统，可以查询到各种命令的使用说明和详细案例，为便于初学者快速了解 MATLAB 功能，还提供了演示窗口。

3. 高效便捷的矩阵和数组运算

MATLAB 语言和 Basic、FORTRAN 和 C 语言一样都规定了矩阵算术运算、关系运算符、逻辑运算符、条件运算符和赋值运算符，而且这些运算符适用于数组运算。对变量定

义维数，并给出矩阵函数、特殊矩阵专用库函数，使之在处理信号处理、数学建模、系统识别、控制优化等领域问题时变得非常简单，MATLAB 有望成为名副其实的“万能演算稿纸”科学计算语言。

4．可扩充能力强，适于二次开发

MATLAB 的快速发展得益于非常好的营运模式。MATLAB 是一个开放系统，具有非常好的可扩充性和可开发性，用户可以非常方便地看到源程序，可以对源程序进行修改，创建符合自己需求的文件库。对外而言，MATLAB 并不具有排他性，可以和 FORTRAN、C、Visual Basic 通过接口相连接，可以方便地相互调用程序。MATLAB 语言具有丰富的库函数，进行复杂的数学运算时可以直接调用，因此用户可以根据自身需要建立或扩充新库函数。良好的交互性使得程序员可以利用可重复使用程序代码，提高效率。

5．移植性好

MATLAB 是用 C 语言编写的，它继承了 C 语言可移植性好的特点，因此可以移植到 C 语言操作平台上。MATLAB 适用的操作系统有 Windows、UNIX 和 Linux 等，除了内部函数外，MATLAB 的核心文件和工具箱都是公开的，读者可以对源文件进行修改，使其更加符合自己的需要。

6．方便的绘图功能

MATLAB 的绘图功能十分强大，它包含一系列绘图命令，例如线性坐标、对数坐标、半对数坐标、极坐标，只要调用不同的绘图函数即可。图中的标题、坐标轴标注以及网格绘制仅需调用相应的命令，非常简单，并可以绘制不同颜色的点、线、复线、多重线。

当前，MATLAB 软件因为其良好的开放性和稳定性，在工程计算中得到广泛应用，最近 10 年来 MATLAB 已经成为国际公认的标准计算软件，在工程计算领域得到广泛应用，全球超过 500 000 个工程师与科学家使用 MATLAB 软件，2000 多家金融机构使用 MATLAB 软件建立经济、金融模型，帮助他们估计经济中的各种风险，并对风险进行有效管理，提高企业经营效率。

1.1.3 MATLAB 金融工具箱的介绍

MATLAB 是一种交互式计算环境，主要用于数学模型的数值计算。针对不同的用途，MATLAB 自带了大量工具箱，这些工具箱是针对不同程序而设立的，MATLAB 中的金融工具箱就是 MathWorks 公司专门为解决金融问题而开发的程序包。

MATLAB 自带的金融工具箱有以下几种。

1. Financial Toolbox

Financial Toolbox 具有下列功能：①日期数据基本处理；②基于均值一方差分析；③时间序列分析；④固定收益计算；⑤有价证券的收益和价格；⑥统计分析；⑦定价和灵敏度分析；⑧年金和现金流计算；⑨折旧方法分析。

2. Financial Derivatives Toolbox

Financial Derivatives Toolbox 是金融衍生产品工具箱，用于固定收益、金融衍生物以及风险投资评估分析，也可用于各种金融衍生物定价策略以及敏感度分析。

3. Financial Time Series Toolbox

Financial Time Series Toolbox 用于分析金融市场的时间序列数据。金融数据是时间序列数据，例如股票价格或每天的利息波动，可以用该工具箱进行更加直观的数据管理。该工具箱支持下列功能：①提供两种创建金融时间序列的对象(用构造器和转换文本文件)；②可视化金融时间序列对象；③技术分析函数分析投资。

4. Fixed-Income Toolbox

Fixed-Income Toolbox 扩展了 MATLAB 在金融财经方面的应用，可以用固定收益模型进行计算，例如定价、收益和现金流动等有价证券的固定收益计算。支持的固定收益类型包括有价证券抵押回报、社会债券、保证金等。该工具箱还能够处理相应金融衍生物的计算，支持抵押回收有价证券、国债、可转换债券等的计算。

5. GARCH Toolbox

GARCH Toolbox 提供了一个集成计算环境，允许对单变量金融时序数据的易变性进行建模。GARCH Toolbox 使用一个广义 ARMAX/GARCH 复合模型对带有条件异方差的金融时序数据进行仿真、预测和参数识别。GARCH Toolbox 提供了基本工具为单变量广义自回归条件异方差(Generalized Auto Regressive Conditional Heteroskedasticity，GARCH)易变性进行建模。GARCH Toolbox 采用单变量 GARCH 模型对金融市场中的变化性进行分析。

6. 对易变性建模

使用 Gaussian 扰动进行 ARCH/GARCH 最大相似参数的估计。

7. Monte Carlo

使用 Monte Carlo 方法对回报、调整、条件易变性进行模拟。

8. 预测

最小均方差预测条件均值和易变性。

9. 时序建模

确定广义 ARMAX/GARCH 复合时序模型的条件均值和方差。

10. 诊断

前后预测诊断和假设测试，包括 Engle ARCH 测试、Q 测试、相似率 Foxdog 制作测试、AIC/BIC 模型顺序选择准则。

11. 图形

图形分析，包括自相关、互相关、部分自相关。

12. 数据处理

时序数据处理和转换。

上述工具箱基本上囊括了通常的金融计算，适用于金融学术研究，特别适合金融实务工作者进行金融计算。Financial Toolbox 提供了一个基于 MATLAB 的财务分析支撑环境，可以完成许多种财务分析统计任务，从简单计算到全面的分布式应用，财务工具箱都能够用来进行证券定价、资产组合收益分析、偏差分析、优化业务量等工作。

1.2 MATLAB 在金融领域的应用

1.2.1 建模预测新兴市场的金融危机

McNelis 博士执教于 Georgetown 大学，他基于 MATLAB 进行经济分析，是金融界领先的分析技术，在亚洲和南美洲中央银行以及其他金融组织中得到了广泛应用。他同时从如何减轻经济转型中给大众带来的金融困境的角度出发提出了一系列具有建设性的建议，他一直教授学生使用基于 MATLAB 的分析方法。

1997 年，马来西亚、菲律宾、泰国、印尼等国开始的金融危机迅速在世界范围内蔓延，给这些工作造成了巨大损失，经济学家 Paul McNelis 着手研究用现代研究方法和工具预测金融危机，从而减少金融危机带来的损失。Paul McNelis 将他的研究重点放在了印度尼西亚。1997 年秋天，印尼卢布价值急剧下跌，印尼国内对美元的需求到达了一个前所未有的水平，即使后来印尼政府从国际货币基金组织获得了 230 亿美元贷款，局势也没有得到控制。McNelis 在美国国际发展委员会的技术协助下，在印尼银行开始他的研究，在这项雄

心勃勃的计划中，McNelis 始终在利用 MATLAB 这个强大的工具以及其中的 Excel Link(Excel 连接工具箱)、Statistics(统计工具箱)、Optimization(优化工具箱)、Control System(控制系统)、Neural Netowrk(神经网络)和 System Identification(系统识别工具箱)进行研究。

McNelis 着手分析印尼 13 年来每个月对货币的需求量，包含了金融危机这段时间。经济学家通常使用的线性分析方法和误差修正方法并不能适应当时的情况，他需要确定一种非常有效的方法来分析这些经济数据，同时，还要尽力减少数据的巨大波动以降低预测结果产生错误的可能性，例如金融危机期间市场对美元的高需求。鉴于 MATLAB 具有强大的数据处理和数值计算功能，易于使用，并且可以处理超大规模数据集，McNelis 选择了 MATLAB 作为研究工具。他相信通过结合线性模型和神经网络模型可以获得更为准确的结果。McNelis 对神经网络的优点解释道：估算不仅仅是对数据顺序化处理，从输入 X 就可以得到 Y，而是采用并行方式来处理，隐含层面中的多个神经元将对输入的数据同时进行处理。McNelis 分析过程的核心内容是和 Pittsburgh 大学的 John Duffy 教授一起开发的遗传算法。在开发这个算法的过程中，他们使用了 MATLAB 中的 Statistics 工具箱，同时还使用了向量化函数来加快数据处理速度，在搜索方法中使用了 Optimization 工具箱中的非线性最小化函数。

在收集完数据以后，McNelis 开始使用传统的线性模型来寻求可以得到的最好结果，然后从这个模型中得到输入值来构建神经网络。在定义神经网络过程中，他首先从一个简单网络开始，例如，隐含层面中只有 3～4 个神经元，然后使用混合方式来训练这个网络，开始时利用遗传算法为神经网络来寻找一个系数集，随后利用这些系数转向一个非线性梯度递减方法。McNelis 使用了 Neural Network 工具箱中的前馈结构来将输入和输出关联起来，他曾经在不同的金融应用中试验过不同的神经网络结构，但是最好的还是有一个隐含层面的前馈体系结构，因为隐含层中的每一个神经元使用了工具箱中的激活函数，输入被传送到隐含层，通过激活函数进行挤压，最终神经元作为线性组合被传送到输出层，从而增加了神经网络的预测能力。

该模型具有很强的预测能力。McNelis 开发的神经网络模型同传统的线性模型相比，可以获得相当高的精确度，并且 GARCH 的使用使这个模型的预测能力更进一步。印尼银行现在正在使用 McNelis 模型来预报货币需求量以及预测通货膨胀率，增强他们抵御通货膨胀压力的能力。McNelis 认为他的模型同时可以用来监控汇率波动，可作为预测危机的有效预警系统。

1.2.2　建立和验证新的期权定价模型

全球众多的金融研究人员都在试图开发出一种不同于标准 Black-Scholes 期权定价模型的新的有价证券定价模型，其中来自英国曼彻斯特商学院、国际证券、投资和银行专家

Dimitrios Gkamas 使用 MATLAB 中的统计和优化工具箱开发出一个创新系统，不仅可以快速计算金融资产价格和风险利率，而且还可以用丰富的 3D 图像来表现它们。

1998 年，由于东南亚市场的崩溃和国际股票市场的巨大灾难，泰国的资本市场处于极其低迷的状态，为了描述如此规模且无法预测的市场行为，Gkamas 寻求期权定价技术，他计划用 LIFFE(伦敦国际金融期货交易所)的数据来验证新模型的性能。

该项目要求计算软件具有处理大量数据的能力(LIFFE 数据库包括每天收盘价、行权价格、Black-Scholes 期权模型隐含波动率、伦敦金融时报指数 FTSE 的 100 家欧洲买权合同隐含的指标期货价格)。有价证券的价格计算公式包含了复杂的数学理论，所以需要一个容易使用的软件可以让它快速地执行这些复杂的计算。

根据曼彻斯特大学计算中心的建议，Gkamas 选择了 MATLAB，利用这个强大的计算环境，不仅可以开发数学算法和进行快速计算，而且可以用一系列丰富的 3D 图像使它们可视化。MATLAB 是一个非常容易使用的工具，Gkamas 没有经过正规培训就很快地掌握了 MATLAB。

Gkamas 的大部分工作是简化随机波动率公式，并把它们加入到 MATLAB 中。因为伦敦国际金融期货交易所的数据以 Excel 文件格式提供，所以 Gkamas 编写了一个 Excel 宏对数据进行排序和筛选。为了输出数据，还编写了一个宏将数据以文本文件形式输出，在 MATLAB 中编写了一个专门函数用于读取数据和输出结果，这可以使数据转换工作简单和可靠。

Gkamas 说 MATLAB 允许他实现“金融衍生工具定价的定解型态或数值算法。”使用 MATLAB 和它的工具箱，能进行复杂的金融计量经济分析和假设检验，使用优化的工具箱使实际与理论期权定价误差的平方和最小化，校验模型是否违背市场数据，然后估计隐含参数，检查它们的演化过程，使用统计工具箱开展大量数据的常规统计分析。

3 年后 Gkamas 打算用 MATLAB 为现存模型开发图形用户界面，投资银行和其他金融研究机构对 Gkamas 的随机波动率模型产生极大兴趣，Gkamas 逐渐成为金融界著名的顾问。Gkamas 说到：“MATLAB 在金融计算和期权定价研究领域中是一个非常优秀的工具，也是很好的教学工具。曼彻斯特大学计算中心告诉我 MATLAB 在计算能力和易用性方面是最好的程序，我已经用它用了 3 年，完全同意这种评价。”

MATLAB 是强大的工具而且易于快速掌握。Gkamas 当初学习 MATLAB 的目的在于及时完成项目，他指出：“我不是程序员，没有计算机科学学位，但我发现 MATLAB 容易使用，不需要任何正规培训。”

MATLAB 在期权定价和风险敏感度可视化方面是很好的工具。例如 Gkamas 用 3D 图像显示 Black-Scholes 期权模型波动率、货币和时间关系。他说：“将伦敦金融时报指数 FTSE 的 100 个隐含波动面图形化以后，我们马上就可以清楚地看到不同随机波动模型所面临的问题所在。”

Gkamas 计划扩展他的研究领域，其中包括衍生金融工具，这种应用的开发相对比较简单，可以使用 MATLAB 编译器和 MATLAB C/C++数学库，这些工具将自动产生 DLL，节省编程时间，提高计算速度。

1.2.3 MathWorks 公司的金融业主要客户

MathWorks 公司致力于金融领域客户的开发，一些大的金融机构都把 MATLAB 作为重要的开发工具。公司客户除了国际货币基金组织外还有下面一些客户。

Ferderal Reserve Bank	Putnam Investment
Goldman Sachs	Prudential Securities
J.P.Morgan	Bank of America
Morgan Stanley	John Hancock
Soloman Smith Barney	Freddie Mac
Ameica RE	Fannie Mae
Merrill Lynch	Moody's Investors
Ernst&Young	Scudder Investments
Deloitte&Touche	Fidelity Investments
Price Waterhourse Coopers	FleetBoston

金融业使用 MATLAB 主要包括以下几个方面。

- 常微分与偏微分方程；
- 信号过程；
- 曲线拟合；
- 线性代数；
- 矩阵操作；
- 概率分布；
- 描述性统计；
- 非线性回归模型；
- 无约束条件下的非线性最小二乘法；
- 二次规划和线性规划；
- 有约束的线性最小二乘法；
- 金融数据处理与格式转换；
- 货币格式；
- 金融数据的图表技术；

- 现金流的计算与分析；
- 利率期限结构计算；
- 证券类衍生产品定价与分析；
- 资产组合分析；
- 利率模型；
- 利率期限结构敏感性分析；
- 条件均值与方差；
- 对冲分析；
- GARCH 模型。

图 1.1 是 MATLAB 金融工具箱的框架图。

从图 1.1 可以看出，MATLAB 金融工具箱的框架图由 MATLAB 基本程序、数据存取和特定功能金融工具箱几个部分构成。

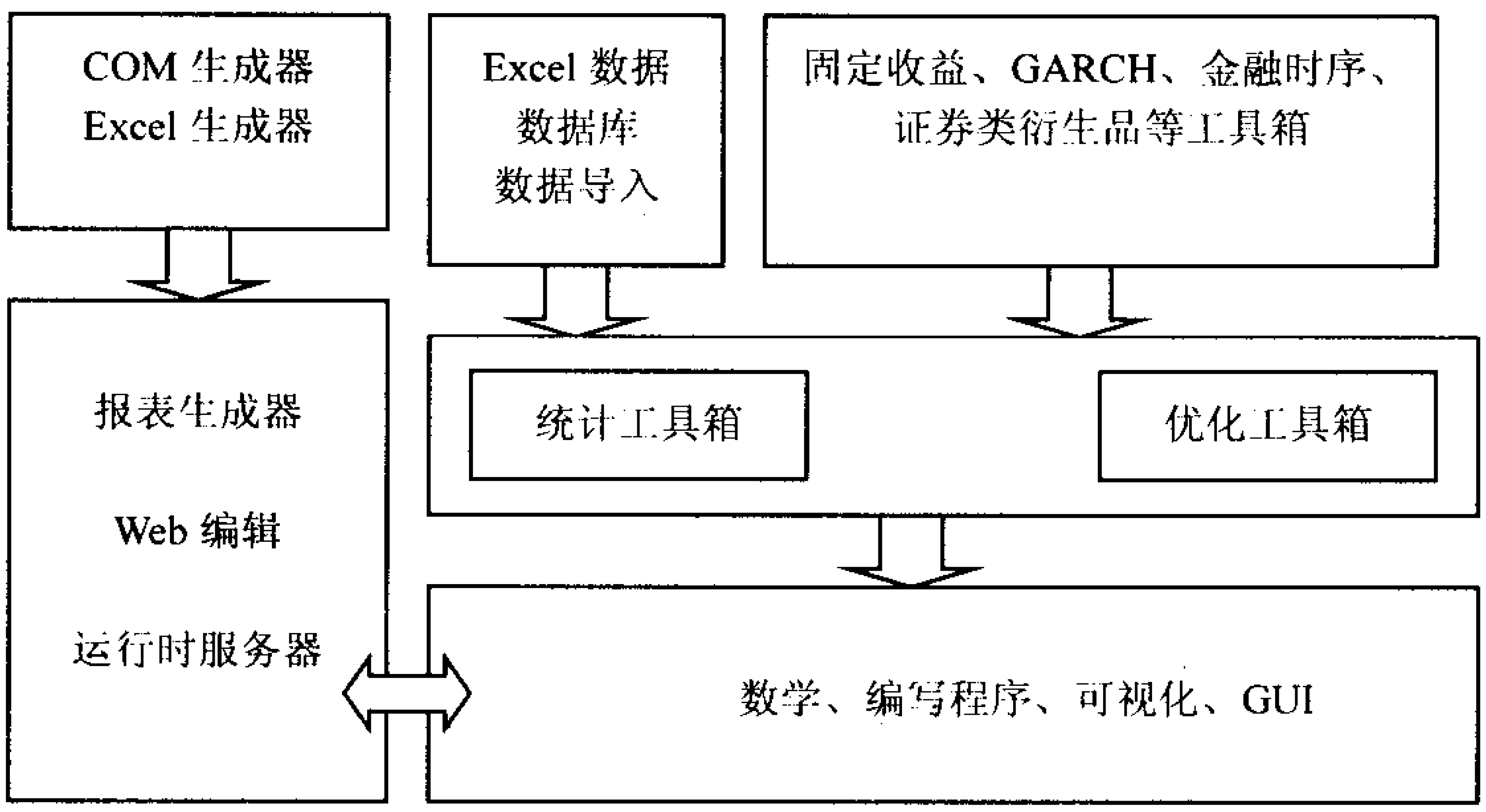


图 1.1　MATLAB 金融工具箱框架图

MATLAB 代码具有开放性，非常适合第三方开发，一些金融机构在 MATLAB 系统上开发了很多具有应用背景的软件。The RisKontrol Group GmbH 公司是一家服务于中央银行、政府监管机构、养老基金、再保险、对冲基金的专业性公司[①]，公司创始人 Jerome L.Kreuser 博士在世界银行工作了 24 年，图 1.2 是 The RisKontrol Group GmbH 公司基于 MATLAB 为特定客户开发的风险管理系统界面。

① 网址： http://www.riskontroller.com/

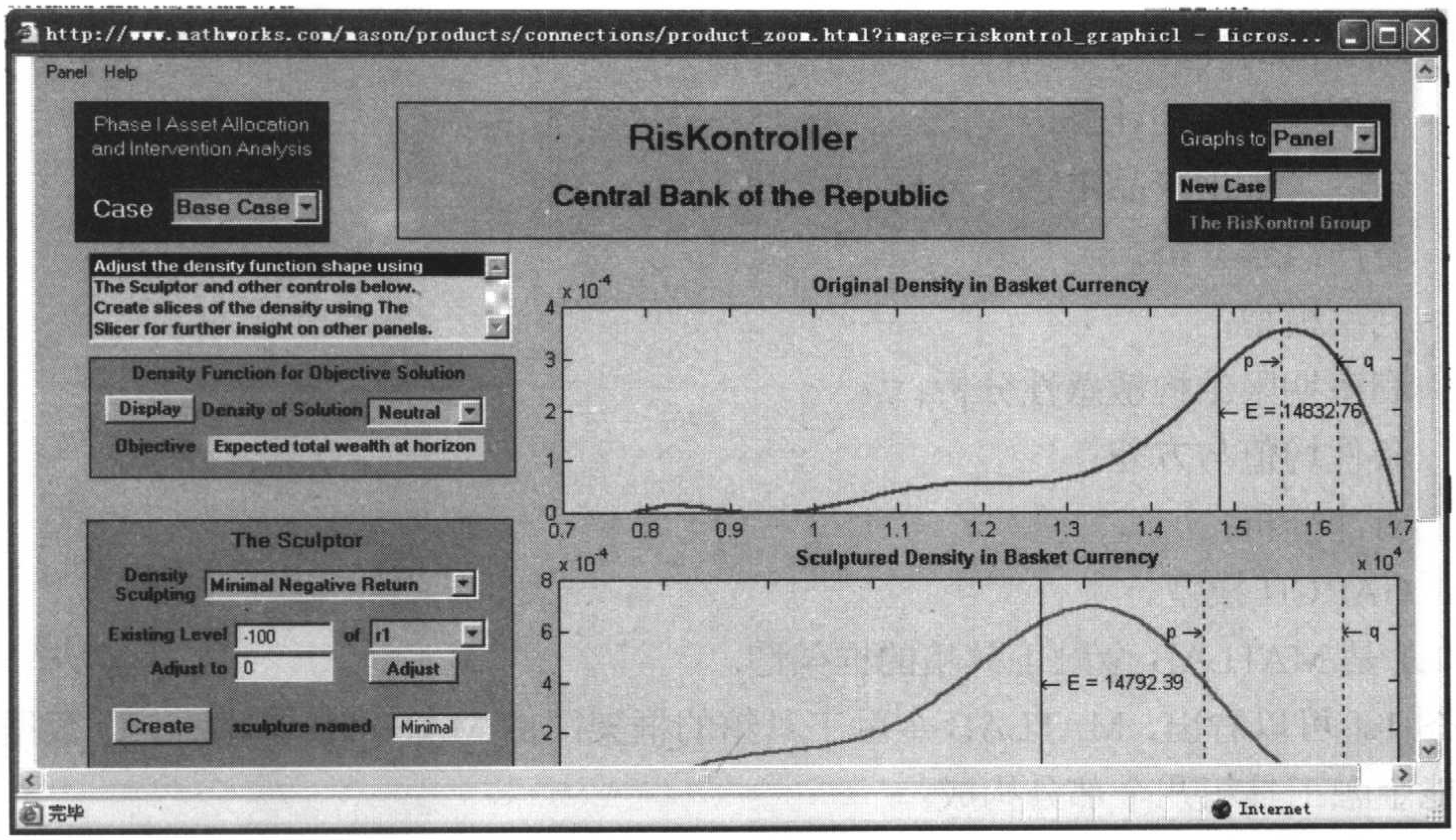


图 1.2　The RisKontrol Group GmbH 开发的风险管理软件界面

思　考　题

1. MATLAB 在数值计算方面具有哪些特点。
2. 从互联网上寻找 MATLAB 在金融理论与实务运用方面的例子。

第 2 章　MATLAB 数值计算初步

本章主要介绍 MATLAB 数值计算基本功能，要求掌握变量与常量的基本用法，熟悉单元数组与结构数组的特点，掌握向量与数组的基本运算，利用索引函数对变量进行操作，掌握矩阵运算，学会对特殊形式的矩阵进行赋值。

2.1　变量与常量

MATLAB 数据类型主要包括数字、字符串、矩阵(数组)、单元数组以及结构数据，下面我们将详细介绍这些数据类型。MATLAB 中变量命名应遵循以下原则。

(1) 变量名大小写代表不同变量。

(2) 变量名长度不超过 31 位，31 位后面的字符被忽略。

(3) 变量名必须以字母开头，变量名中可以有字母、数字和下划线，但不能有标点。

MATLAB 中变量分为局部变量与全局变量，局部变量仅在函数文件中有效，全局变量在主程序与函数文件中有效。如果定义全局变量需要对变量事先进行声明，只需在变量前加上关键字 global[①]。

MATLAB 中还有一些预定义变量，例如 pi 表示圆周率，NaN 表示不确定值，eps 表示浮点运算的相对精度为10^{-52}，Inf 表示无穷大，Realmin 表示最小正浮点数2^{-1022}，Realmax 表示最大正浮点数2^{1023}。

2.1.1　数字变量

MATLAB 中一般代数表达式的输入就如同在草稿纸上演算一样，如四则运算符号可直接用+、-、*、/，非常方便，因此 MATLAB 被称为演算纸式科学计算语言。

对于简单的数字运算，可以直接在窗口中以惯用形式输入，例如计算 23×45，可以在 Command 窗口中直接执行如下命令：

```
>> 23*45
ans =
1035
```

① 全局变量容易引起一些不便，使用时应注意前后统一。

计算结果自动保存在变量 ans 中。

MATLAB 中数值有多种显示格式，在默认情况下，如果数据为整数，则以整型表示；如果为实数，则以保留小数点后面的 4 位浮点型表示。MATLAB 中的所有数据按照 IEEE 浮点标准所规定的常型格式存储，数值有效范围是$10^{-308}\sim10^{308}$。

MATLAB 的输入格式完全继承了 C 语言风格，如正负号、小数点和科学记数法等。

MATLAB 的输出格式可由 format 函数控制，format 函数的主要功能是控制变量的输出形式，其内容如表 2.1 所示。

表 2.1 format 函数内容

类　型	结　果	例　子
+	+，-，空格	+
bank	美元或者美分，小数点后保留 2 位有效数字	3.14
compact	尽量把输出的形式控制在一行	theta = pi/2　theta=1.5708
hex	十六进制	
long	长型输出，双精度 15 位，单精度 8 位	3.14159265358979
long e	长型输出，浮点格式，双精度 15 位，单精度 8 位	3.141592653589793e+00
long g	最合适的长型输出，双精度 15 位，单精度 8 位	3.14159265358979
loose	换行输出，便于阅读	theta = pi/2 theta=1.5708
rat	以最简分数形式输出	3/2
short	短型输出，保留 5 位有效数字	3.1416
short e	短型浮点输出，保留 5 位有效数字	3.1416e+00
short g	短型最合适输出，保留 5 位有效数字	3.1416

下面以 sqrt(3)为例说明 format 的运用，代码如下：

```
>> a=sqrt(3)
a =
   1.7321
>> format short;a
a =
    1.7321
>> format long;a
a =
   1.73205080756888
>> format hex;a
a =
```

```
  3ffbb67ae8584caa
>> format bank;a
a =
  1.73
>> format +;a
a =
+
>> format long e;a
a =
   1.732050807568877e+000
>> format short e;a
a =
 1.7321e+000
>> format short g;a
a =
      1.7321
>> format long g;a
a =
          1.73205080756888
>> format rational;a
a =
 1351/780
```

如果要查询当前输出格式，可以进行如下操作：

```
>> format long
>> get(0,'Format')
ans =
long
```

可以知道当前输出形式为 long 输出格式。

2.1.2 字符串操作

字符串运算是高级语言不可缺少的部分，MATLAB 的字符串运算功能是非常丰富的，特别是 MATLAB 增加了符号运算工具箱(Symbollic Toolbox)之后，字符串函数运算功能大大加强，已经不再是简单的字符串运算，而是成为了 MATLAB 符号运算表达式的基本单元。

1. 字符串赋值

MATLAB 中可以直接对字符串进行赋值，例如：

```
>> a='hello'
a =
hello
```

这样就把字符串“hello”值赋给了变量 a。

上述过程可以通过调用函数 char 来完成，代码如下：

```
>> a=char('h','e','l','l','o');
>> a=a'
a =
hello
```

此外，字符串中每个字符(含空格)都是字符串数组的一个元素，例如：

```
>> s='i use MATLAB'
s =
i use MATLAB
```

size 函数用来查看变量维数，s 是一个 1 行 12 列数组。

```
>> size(s)
ans =
    1    12
```

利用矩阵的方式同样可以对字符串变量进行赋值，例如：

```
>> s=['i use MATLAB']
s =
i use MATLAB
```

2. 字符串和转换函数

将字符串转换为对应 ASCII 码是的 double 函数，代码如下：

```
>> s=['i use MATLAB'];
>> double(s)
ans =
   105    32   117   115   101    32   109    97   116   108    97    98
```

把字符数组转化为字符串的是 cellstr 函数，代码如下。

```
>> s=['i use MATLAB']';
>> cellstr(s)
ans =
    'i'
    ''
    'u'
    's'
    'e'
    ''
    'm'
    'a'
    't'
```

```
'l'
'a'
'b'
```

具体而言，字符串与数组之间的转换函数如表 2.2 所示。

表 2.2　字符串与数组之间的转换函数

函数名	功　能	函数名	功　能
num2str	数字转换为字符串	str2num	字符串转换为数字
int2str	整数转换为字符串	str2int	字符串转换为整数
mat3str	矩阵转换为字符	str2mat	多字符串组成字符矩阵

3. 数值与数字字符串之间相互转换

将数值转换为数字字符串，保留 2 位有效数字，代码如下：

```
>> B=rand(2,4)
B =
    0.9501    0.6068    0.8913    0.4565
    0.2311    0.4860    0.7621    0.0185
>> B_str=num2str(B,2)
B_str =
0.95      0.61      0.89      0.46
0.23      0.49      0.76     0.019
```

将数字字符串转换为数值，代码如下：

```
>> B=str2num(B_str)
B =
    0.9500    0.6100    0.8900    0.4600
    0.2300    0.4900    0.7600    0.0190
```

4. 把字符串转换为双精度数值

```
>> str2double('23+45i')
ans =
  23.0000 +45.0000i
```

5. 字符串操作

MATLAB 对字符串的操作与 C 语言几乎相同，如表 2.3 所示。

表 2.3　字符串操作函数

函数名	功　能	函数名	功　能
strcat	连接串	strrep	串替代

续表

函数名	功能	函数名	功能
strvcat	垂直连接串	strtok	寻找串中的记号
strcmp	串比较	upper	大写串
strncmp	比较串前 n 个字符	lower	小写串
findstr	在其他串中寻找此串	blanks	空格串
strjust	证明字符数组	deblank	删掉串中的空格
strmatch	查找匹配的字符串		

6. 字符串对变量操作

如果已知 a1=1；a2=2；a3=3；a4=4；a5=5，现在需要把这 5 个变量放在一个数组变量 a 中，在 MATLAB 中编写脚本文件 utility.m 如下：

```
name='a';
for i=1:5
name(2)=int2str(i); %把 name 中的第二个字母换成 i
a(i)=eval(name);    %eval 函数将 name 当做运算表达式，并进行运算
end
a
```

运行脚本文件可以看到 a 的内容如下：

```
>>utility
a =
     1     2     3     4     5
```

7. 字符串循环

字符串循环是 MATLAB 所具有特色的功能，是通过对字符串中的字符进行循环实现字符串索引。

下面我们对字符串“utility”进行循环，代码如下：

```
s='utlity';
for i=s
    i
end
```

显示结果如下：

```
i =
u
i =
t
```

```
i =
l
i =
i
i =
t
i =
y
```

2.1.3　单元型变量与结构变量

单元型变量是 MATLAB 中较为特殊的一种数据类型，从本质上讲，单元型变量实际上是一种以任意形式数组为元素的多维数据。

1. 单元型变量

单元矩阵中的不同位置可有不同的数据类型，单元数组中的不同位置也可以存放不同类型的数据，单元数组可以包含几个字符串、几个数字。

单元型变量可以通过两种方式定义，一种是用赋值语句直接定义，另外一种是由 cell 函数预先分配存储空间，然后定义。

用赋值语句直接定义单元型变量的例子如下：

```
>> a=[1 2;3 4];
>> b={1:4,a,'i use MATLAB'}
b =
    [1x4 double]    [2x2 double]    'i use MATLAB'
```

如果需要查看每个单元的内容，可以输入如下代码：

```
>> b{1}
ans =
     1     2     3     4
>> b{2}
ans =
     1     2
     3     4
>> b{3}
ans =
i use MATLAB
```

2. 单元型变量的相关函数

在 MATLAB 语言中，单元型变量的函数如表 2.4 所示。

表 2.4　MATLAB 语言的单元型变量函数

函数名	功能	函数名	功能
cell	生成单元型变量	deal	输入输出控制
cellfun	作用于单元变量的函数	cell2struct	单元型转化结构型
celldisp	显示单元变量内容	struct2cell	结构型转化为单元型
cellplot	图形显示单元变量	iscell	判断是否为单元型
num2cell	数值型转变为单元型	reshape	改变单元数组结构

下面给单元变量 f 进行赋值，代码如下：

```
>> f={'MATLAB',45,78,32}
```

可以用 cellplot 函数显示 f 的内容，代码如下：

```
>> cellplot(f)
```

所显示的内容如图 2.1 所示。

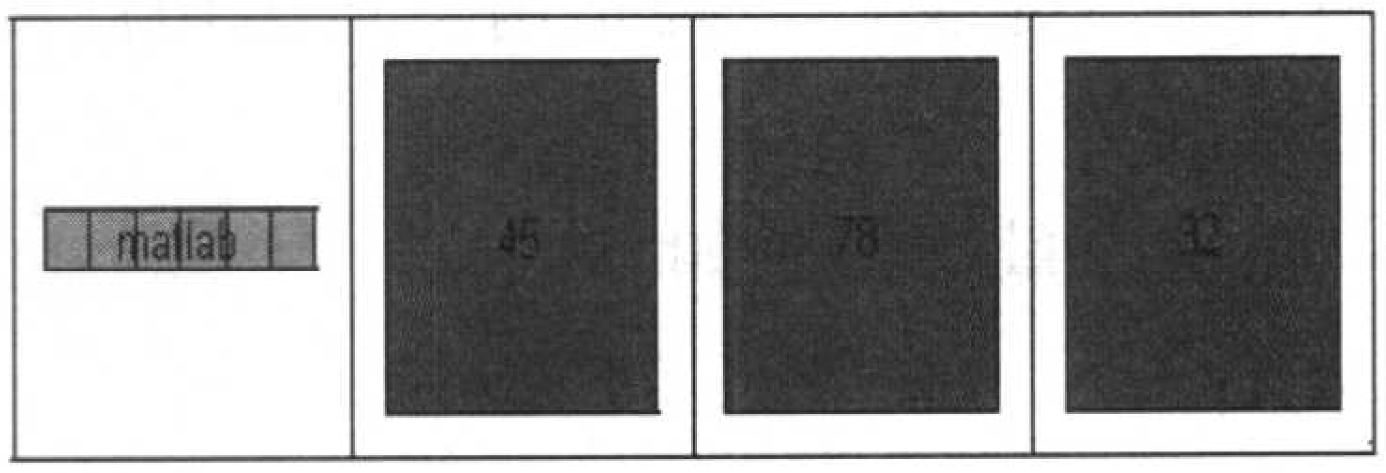


图 2.1　单元型变量内容

3. 结构数组

MATLAB 中的结构数组类似于 C 语言中的“结构体”数据，它包含了许多不同类型的数据项，结构数组中的不同数据项即为一个属性。

MATLAB 语言中提供了一些专门对结构数组进行操作的函数，如表 2.5 所示。

表 2.5　结构数组的操作函数

函数名	功能	函数名	功能
fieldname	查询结构数组的属性名	isfield	返回 1 表示数组，0 不是
getfield	获取结构数组的属性值	rmfield	删除属性
isstruct	返回 1 表示是结构数组，返回 0 表示不是结构数组	setfield	设置属性值

下面我们直接用结构数组描述公司员工的姓名、工龄和工资。

第 1 种方法是直接用 struct 函数创建，代码如下：

```
>> clerk=struct('name','李 明','age','20','wage','2500')
clerk =
    name: '李 明'
     age: '20'
    wage: '2500'
```

这样就创建了名为 clerk 的结构数组。

第 2 种方法是采用赋值语句创建结构数组，代码如下：

```
>> clerk.name='李 明';
>> clerk.age=20;
>> clerk.wage=2500;
>> clerk
clerk =
    name: '李 明'
     age: 20
    wage: 2500
```

下面输入第二位员工的姓名、年龄和工资。

```
>> clerk(2).name='张 军';
>> clerk(2).age=23;
>> clerk(2).wage=3000;
```

下面开始对 clerk 中的元素进行访问，代码如下：

```
>> clerk(1)
ans =
    name: '李 明'
     age: 20
    wage: 2500
>> clerk(2)
ans =
    name: '张 军'
     age: 23
    wage: 3000
```

2.2　矩阵及向量运算

2.2.1　矩阵生成

1. 在 Command 窗口中直接输入矩阵数据

下面在 Command 窗口中输入矩阵中的数据，代码如下：

```
>> a=[1 2 3 ;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
```

2．单位阵

MATLAB 中生成单位阵的函数是 eye 函数，eye(n)表示生成 $n \times n$ 阶单位阵，eye(m,n)表示生成 $m \times n$ 阶单位阵，下面给出几种生成单位阵的方法。

```
>> eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
```

生成的是 3 阶单位阵。

再如：

```
>> eye(3,4)
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
```

生成的是 3 行 4 列的单位阵。

此外 MATLAB 中还可以生成与给定矩阵相同阶数的单位阵。

给定矩阵 ***a***，下面生成与 ***a*** 相同阶数的单位阵，代码如下：

```
>> a=[1,2,3;4,5,6;7,8,9]
a =
     1     2     3
     4     5     6
     7     8     9
>> eye(size(a))
ans =
     1     0     0
     0     1     0
     0     0     1
```

3．零矩阵

零矩阵就是所有元素都为 0 的矩阵，其函数是 zeros，用法同 eye 函数，例如：

```
>> zeros(3,4)
ans =
```

```
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

4. 1 矩阵

1 矩阵就是所有元素都为 1 的矩阵，其函数是 ones，用法同 eye 函数，例如：

```
>> ones(3,4)
ans =
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

5. 利用 diag 函数创建对角矩阵

MATLAB 中用 diag 函数来建立对角矩阵，格式一般为 diag(X,n)，X 表示对角线元素，n 表示对角矩阵元素，n>0 时为上对角矩阵元素，n<0 时表示下对角矩阵元素。

如果需要创建一个对角线元素为 1 到 5 的对角矩阵，可以执行以下命令：

```
>> a=[1,2,3,4,5];
>> diag(a)
ans =
     1     0     0     0     0
     0     2     0     0     0
     0     0     3     0     0
     0     0     0     4     0
     0     0     0     0     5
```

如果需要建立一个次对角线元素为 ***a*** 的对角矩阵，可以执行如下命令：

```
>> diag(a,-1)
ans =
     0     0     0     0     0     0
     1     0     0     0     0     0
     0     2     0     0     0     0
     0     0     3     0     0     0
     0     0     0     4     0     0
     0     0     0     0     5     0
```

6. 矩阵变向

矩阵变向包括矩阵旋转、左右翻转、上下翻转，分别由函数 rot90、fliplr 和 flipud 来实现。函数 flipdim 用来对指定维数进行翻转。各函数的调用格式如下：

```
rot90(A)          %将矩阵 A 逆时针方向旋转 90°
rot90(A,k)        %将矩阵 A 逆时针方向旋转 90×k°
fliplr(A)         %将 A 左右翻转
```

```
flipud(A)          %将矩阵 A 上下翻转
flipdim(A, dim)    %将矩阵 A 第 dim 维翻转，dim=1 表示对行翻转，dim=2 表示对列翻转
>> a=[1 2 3;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> rot90(a)
ans =
     3     6     9
     2     5     8
     1     4     7
>> rot90(a,-1)
ans =
     7     4     1
     8     5     2
     9     6     3
>> fliplr(a)
ans =
     3     2     1
     6     5     4
     9     8     7
>> flipdim(a,1)
ans =
     7     8     9
     4     5     6
     1     2     3
```

7. 矩阵迹

MATLAB 中求矩阵迹的函数是 trace。例如求一个 3 阶魔方矩阵迹，可以执行如下命令：

```
>> a=magic(3)
a =
     8     1     6
     3     5     7
     4     9     2
>> trace(a)
ans =
    15
```

计算出矩阵迹为 15。

8. 均匀分布随机矩阵

MATLAB 中生成均匀分布随机矩阵的函数是 rand，其调用格式如下：

```
Y = rand(n)                %生成 n×n 随机矩阵，其元素在(0,1)内
Y = rand(m,n)              %生成 m×n 随机矩阵
Y = rand([m n])            %生成 m×n 随机矩阵
Y = rand(m,n,p,…)          %生成 m×n×p×…随机矩阵或数组
Y = rand([m n p…])         %生成 m×n×p×…随机矩阵或数组
Y = rand(size(A))          %生成与矩阵 A 相同大小的随机矩阵
rand                       %无变量输入时只产生一个随机数
s = rand('state')          %采用 Marsaglia 减法生成随机数
rand('state', s)           %使状态重置为 s
rand('state', 0)           %重置发生器到初始状态
rand('state', j)           %重置发生器到第 j 个状态
rand('state', sum (100*clock))        %每次重置到不同状态
```

下面我们产生一个 3×4 阶的随机矩阵，代码如下：

```
>> R=rand(3,4)
R =
  0.9501    0.4860    0.4565    0.4447
  0.2311    0.8913    0.0185    0.6154
  0.6068    0.7621    0.8214    0.7919
```

下面产生一个在区间[10, 20]内均匀分布的 4 阶随机矩阵，代码如下：

```
>> a=10;b=20;
>> x=a+(b-a)*rand(4)
x =
   19.2181   19.3547   10.5789   11.3889
   17.3821   19.1690   13.5287   12.0277
   11.7627   14.1027   18.1317   11.9872
   14.0571   18.9365   10.0986   16.0379
```

9. 正态分布随机矩阵

MATLAB 中生成正态分布随机矩阵的函数是 randn。其调用方式如下：

```
Y = randn(n)               %生成 n×n 正态分布随机矩阵
Y = randn(m,n)             %生成 m×n 正态分布随机矩阵
s = randn('state', s)      %重置状态为 s
s = randn('state', 0)      %重置发生器为初始状态
s = randn('state', j)      %重置发生器到第 j 状态
s = randn('state', sum(100*clock))      %每次重置到不同状态
```

10. 矩阵行列式

MATLAB 中计算矩阵行列式的值用函数 det。

例如，求矩阵$\begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{pmatrix}$行列式值的代码如下：

```
>> a=[1 2 3 ;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9
>> det(a)
ans =
     0
```

11．矩阵特征值

MATLAB 中计算矩阵特征值的函数是 eig。

下面求矩阵$\boldsymbol{a}=\begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{pmatrix}$的特征值，代码如下：

```
>> a=[1 2 3 ;4 5 6;7 8 9]
a =
     1     2     3
     4     5     6
     7     8     9

>> eig(a)
ans =
   16.1168
   -1.1168
   -0.0000①
```

矩阵特征值分别为 16.1168、-1.1168 和 0。

2.2.2　向量运算

MATLAB 中矩阵运算的书写习惯和平常的书写习惯很相似，需要注意的是，矩阵运算符前面不用加点号，特别地，点除“./”表示对应元素相除，而向量相除运算其前面需要加点号。

① 非常小的负数。

1. 向量点乘

向量点乘表示对应元素相乘，例如：

```
>> a=[1 2 3];b=[2 3 4];
>> a.*b
ans =
     2     6    12
```

这表示向量对应元素相乘，如果 ***a***、***b*** 都是同阶矩阵，同样表示矩阵中对应元素相乘，例如：

```
>> a=[1,2,3;4 5 6; 7 8 9];b=eye(3);
>> a.*b
ans =
     1     0     0
     0     5     0
     0     0     9
```

2. 向量内积计算

向量内积是指两个向量的对应元素相乘再求和，要求两个向量维数应该相同，例如 $(a_1,a_2,a_3,\cdots,a_n)$ 和 $(b_1,b_2,b_3,\cdots,b_n)$ 内积为 $a_1b_1+a_2b_2+a_3b_3+\cdots+a_nb_n$。MATLAB 中求向量内积的函数为 dot，例如：

```
>> a=[1 2 3];b=[2 3 4];
>> dot(a,b)
ans =
    20
```

3. 数据索引

数据索引 find 函数是 MATLAB 中比较常用的函数，例如 X=[1 0 4 -3 0 0 0 8 6]，如果要使 X 中大于 0 的元素变成 1，可以执行以下函数：

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> y=find(X>0);
>> X(y)=1
X=
     1     0     1    -3     0     0     0     1     1
```

或者执行如下命令：

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> X(find(X>0))=1
X =
     1     0     1    -3     0     0     0     1     1
```

进一步地，如果要使 X 中大于 0 小于 2 的数变为 1，可以执行以下命令：

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> X(find(X>0 & X<1))=1
X =
     1     0     4    -3     0     0     0     8     6
```

如果要求 X 中前两个大于 0 的数变为 1，可以执行以下命令：

```
>> X = [1 0 4 -3 0 0 0 8 6];
>> X(find(X>0 ,2))=1
X =
     1     0     1    -3     0     0     0     8     6
```

对于矩阵而言，情况要变得复杂一点，例如求 3 阶魔方矩阵中大于 3 的元素，魔方矩阵是指矩阵中元素为整数，而且每行、每列、对角线元素之和相等。Find 函数对矩阵中元素进行搜寻时的顺序是从第一列开始从上到下进行计数，然后从第二列开始从上到下进行计数，例如：

```
>> M = magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
>> r=find(M>3)
r =
     1
     3
     5
     6
     7
     8
```

从上述结果可以看出，矩阵 $\boldsymbol{M}$ 从第一行开始计数，从左到右第 1、3、5、6、7、8 个元素大于 3，再如：

```
>> M(find(M>3))=1
M =
     1     1     1
     3     1     1
     1     1     2
```

实际上借助于逻辑函数 abs 可以实现类似功能，我们将魔方矩阵中大于 3 的元素改为 2，代码如下：

```
>> a=magic(3)
```

```
a =
     8     1     6
     3     5     7
     4     9     2
>> L=abs(a>3)
L =
     1     0     1
     0     1     1
     1     1     0
>> a(L)=2
a =
     2     1     2
     3     2     2
     2     2     2
```

4. 查找矩阵中不相同元素的个数

在矩阵计算中，有时需要统计不相同元素的个数，这时可以用 unique 函数来统计，例如：

```
>> a=1:9
a =
     1     2     3     4     5     6     7     8     9
>> unique(a)
ans =
     1     2     3     4     5     6     7     8     9
```

表示不相同元素的个数为 9 个，再如：

```
>> a=[1 2 2 4 4 5]
a =
     1     2     2     4     4     5
>> unique(a)
ans =
     1     2     4     5
```

表示不相同元素的个数为 4 个，分别是 1、2、4 和 5。

此外还可以直接求得矩阵中不相同元素个数而不显示元素，代码如下：

```
>> length(unique(a))
ans =
    9
```

其中 unique(*a*)找出不相同元素的位置，length 计算出 *a* 中元素的个数。

2.2.3 矩阵运算

1. 矩阵累积求和

调用方式

```
a=cumsum(x,n)
```

输入参数

```
x              %矩阵
n              %n=1，表示沿行方向累计求和；n=2，表示沿列方向累积求和
```

例如对矩阵沿列方向累积求和，代码如下：

```
>> a=ones(2,5)
a =
     1     1     1     1     1
     1     1     1     1     1
>> cumsum(a,2)
ans =
     1     2     3     4     5
     1     2     3     4     5
```

对矩阵沿行方向累积求和，代码如下：

```
>> cumsum(a,1)
ans =
     1     1     1
     2     2     2
     3     3     3
```

2. MATLAB 中数据的显示格式

MATLAB 中有两种方式显示数据格式，一是通过 MATLAB 界面上的菜单进行设置；另外也可以在 MATLAB Command 窗口中直接执行相关命令。若用菜单设置 MATLAB 数据显示格式，可选择菜单栏中 File 菜单下的 Preference 命令，将会出现 Data 对话框，这可以借助于滚动条来设置 MATLAB 的数据显示格式，除了可以选择数据显示位数长短外，还可以设置数据显示格式是疏松型(Loose)还是紧凑型(Cpmpact)。

3. 符号求解方程

【例 2-1】求解下列方程组。

```
>> [x,y]=solve('x*y=6,x+y=5','x,y')
x =
```

```
  3
  2
y =
   2
 3
```

由上面的结果可以知道(x,y)解为$\begin{cases}x=3\\y=2\end{cases}$或$\begin{cases}x=2\\y=3\end{cases}$。

4. *LU* 分解

高斯消元法求解线性方程，包括把增广矩阵转换为三角矩阵形式的过程，消去阶段的工作是把矩阵 ***A*** 分解成为下三角矩阵 ***L*** 与上三角矩阵 ***U*** 的乘积，这种计算矩阵 ***L***、***U*** 的过程称为 ***LU*** 分解法。如果矩阵 ***A*** 是正定矩阵，那么进一步地还有更高效的 Cholesky 分解法：***A=C'C***，***C*** 是上三角矩阵。

调用方式

```
[L,U] = lu(X)
Y = lu(X)
[L,U,P] = lu(X)
 [L,U,P,Q] = lu(X)
[L,U,P] = lu(X,thresh)
[L,U,P,Q] = lu(X,thresh)
```

其中，在[***L***,***U***]=lu(***X***)公式中，***L*** 为置换矩阵，***U*** 为下三角矩阵，满足 ***L***×***U***=***X***。

在[***L***,***U***,***P***]=lu(***X***) 公式中，***L*** 为上三角矩阵，***U*** 为下三角矩阵，***P*** 为置换矩阵，满足 ***L***×***U***=***P***×***X***。

下面给出矩阵 ***A*** 的形式，对矩阵 ***A*** 进行 ***LU*** 分解，代码如下：

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> [L,U] = lu(A)
L =
    0.1429      1.0000      0
    0.5714      0.5000      1.0000
    1.0000      0           0
U =
     7.0000     8.0000      0
     0          0.8571      3.0000
     0          0           4.500
```

给出的矩阵 ***L*** 与 ***U*** 形式满足要求。

如果 ***L*** 为下对角矩阵，主对角线上的元素为 1，***U*** 为上对角矩阵，则代码如下：

```
>> [L,U,P] = lu(A)
```

```
L =
    1.0000         0         0
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
U =
    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000
P =
     0     0     1
     1     0     0
     0     1     0
```

下面验证 $\boldsymbol{L}\times\boldsymbol{U}=\boldsymbol{P}\times\boldsymbol{A}$。

```
>> L×U-P×A
ans =
     0     0     0
     0     0     0
     0     0     0
```

如解线性方程组

$$\begin{cases} x_1+0.2x_2+0.5x_3=1 \\ 0.2x_1+0.4x_2+x_3=2 \\ 0.5x_1+0.1x_2+0.6x_3=1.5 \end{cases}$$

记

$$\boldsymbol{A}=\begin{pmatrix} 1 & 0.2 & 0.5 \\ 0.2 & 0.4 & 1 \\ 0.5 & 0.1 & 0.6 \end{pmatrix},\quad \boldsymbol{x}=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix},\quad \boldsymbol{b}=\begin{pmatrix} 1 \\ 2 \\ 1.5 \end{pmatrix}$$

则线性方程组可写为 $\boldsymbol{Ax}=\boldsymbol{b}$。

根据 $\boldsymbol{LU}$ 分解有 $\boldsymbol{LUx}=\boldsymbol{b}$，记 $\boldsymbol{Y}=\boldsymbol{L}^{-1}\boldsymbol{b}$，则 $\boldsymbol{x}=\boldsymbol{U}^{-1}\boldsymbol{Y}$ 为方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的解。

```
>> a=[1 0.2 0.5;0.2 0.4 1;0.5 0.1 0.6];
>> b=[1 2 1.5]';
>> [L,U]=lu(a)
L =
    1.0000        0                 0
    0.2000        1.0000            0
    0.5000        0                 1.0000
U =
    1.0000        0.2000            0.5000
    0             0.3600            0.9000
    0             0                 0.3500
```

```
>> y=L^-1*b;
>> x=U^-1*y;
```

从上面的结果可以看出，***L*** 为下对角矩阵，***U*** 为上对角矩阵，***x*** 是方程的解。

2.3　插值与拟合

MATLAB 中涉及的插值函数主要有一维插值(interp1)、样条插值(spline)和 Hermite 插值。

2.3.1　一维插值

如果给出一组数组(x,y)，需要找出 x,y 之间的函数关系 $f(.)$，例如 $y=f(x)$，然后给定 x_1，求 y_1，使得 $y_1=f(x_1)$，这样的函数关系 f 有很多种，分别叫做不同插值方法，如果 x 是一维变量，就称为一维插值，如果 x 是二维数组就称为二维插值，类似地有三维插值。

函数名称：一维插值

调用方式

```
Y1=interp1(x,y,x1,'method')
```

输入参数

```
x   %自变量
y   %因变量
x1  %给定的自变量值
```

method 插值算法方式，默认值为线性插值，其取值见表 2.6。

输出参数

```
Y1  %与自变量 x1 对应的函数值
```

表 2.6 是各种插值算法特点的比较。

表 2.6　插值算法特点的比较

插值算法	含　义	特　点	用　途
linear	线性插值	较快	最常用
cubic	三次多项式插值	较慢，精度高	作平滑用
spline	三次样条插值	最慢，精度高，平滑	作平滑用
nearst	最邻近插值	最快，精度低。	特大数据处理

【例 2-2】自变量保存在数组 x 中，与 x 对应的值保存在 y 中，求 x_1=0.34 时对应的

y 值。

代码如下：

```
>> x=0:0.1:2;y=2*x;
>> x1=0.34;
>> y1=interp1(x,y,x1,'linear')  %三次线性插值
y1 =
    0.6800
>> y1=interp1(x,y,x1,'spline')  %样条插值
y1 =
    0.6800

> y1=interp1(x,y,x1,'cubic')  %三次多项式插值
y1 =
    0.6800
>> y1=interp1(x,y,x1,'nearest') %最邻近插值
y1 =
    0.6000
```

从上面的计算结果可以看出最邻近插值法的计算效果最差。

2.3.2 样条插值

MATLAB 中处理样条插值的函数是 spline，其调用方式如下：

调用方式

```
pp = spline(x,Y)
yy = spline(x,Y,xx)
```

注意，自变量 x 是单调增加的。

下面是一个样条函数插值的例子：

```
>> x=0:0.1:2;y=2*x;
>> x1=0.34;
>> y1=spline(x,y,x1)
Y1 =
0.6800
```

2.3.3 Hermite 插值

很多实际问题不但要求在节点处相等，而且要求节点处的导数也相等，甚至高阶导数也相等，Hermite 插值可以保持节点处的导数相一致，为简单起见下面讨论三阶 Hermite 插值，相应函数为 pchip，代码如下：

```
>> x = -3:3;
>> y=2*x;
>> t=-1:0.5:1;
>> p = pchip(x,y,t) %x 为自变量，y 为因变量，t 为待插值向量
p =
-2    -1     0     1     2
```

2.4　符号计算

MATLAB 不仅可以进行数值计算，而且可以进行符号计算。

1. 声明符号变量和符号表达式

例如，在提示符下输入 3/5，可以得到：

```
>> 3/5
ans =
    0.6000
```

则 3/5 变成了小数。

而用 sym 函数则得到：

```
>> sym(3)/sym(5)
 ans =
 3/5
```

结果仍然是分数。

比如定义一个符号矩阵

$$A=\begin{pmatrix}1/2 & 2/3\\ 1/4 & 4/7\end{pmatrix}$$

的代码如下：

```
>> A=sym('[1/3,2/3;1/4,4/7]')
 A =
    [ 1/3,  2/3]
    [ 1/4,  4/7]
```

2. 合并同类项

用 collect 函数可以合并同类项，例如合并 $x^2+2xy+y^2$ 同类项，代码如下：

```
>> collect(x^2+2*x*y+x*y^2)
 ans =
 x^2+(2*y+y^2)*x
```

3. 简化符号运算表达式

有时为了计算方便，需要对计算结果给出简单的表达式，下面我们用 simple 函数对公式进行简化，代码如下：

```
>> syms x y
>> y=x^2+2*x*y+y^2
 y =
 x^2+2*x*y+y^2
 >> simple(y)
    simplify:
    x^2+2*x*y+y^2
    radsimp:
 x^2+2*x*y+y^2
 combine(trig):
 x^2+2*x*y+y^2
 factor:
 (x+y)^2
 expand:
 x^2+2*x*y+y^2
 combine:
 x^2+2*x*y+y^2
 convert(exp):
 x^2+2*x*y+y^2
 convert(sincos):
 x^2+2*x*y+y^2
 convert(tan):
 x^2+2*x*y+y^2
 collect(x):
 x^2+2*x*y+y^2
 mwcos2sin:
 x^2+2*x*y+y^2
 ans =
 (x+y)^2
```

公式 $x^2+2xy+y^2$ 的简化结果为 $(x+y)^2$。

4. 符号矩阵运算

MATLAB 可以完成矩阵的四则运算，下面是对符号矩阵进行操作，代码如下：

```
>> syms a b c d x y z
>> A=[a b ;c d]
 A =
      [ a, b]
      [ c, d]
```

```
>> B=sym('[x y ;c d]')
   B =
      [ x, y]
      [ c, d]
```

符号矩阵加法运算操作的代码如下：

```
>> A+B
 ans =
      [ a+x, b+y]
      [ 2*c, 2*d]
```

符号矩阵乘法运算操作的代码如下：

```
>> A*B
ans =
    [ a*x+b*c, a*y+b*d]
    [ c*x+d*c, c*y+d^2]
```

符号矩阵除法的代码如下：

```
>>C1= A/B
  C1=
      [    (-b*c+a*d)/(x*d-c*y), -1/(x*d-c*y)*(a*y-x*b)]
      [                  0,                   1]
```

矩阵除法也可以用乘以逆矩阵表示。

```
>> C2=A*inv(B)
C2 =
[  a*d/(x*d-c*y)-b*c/(x*d-c*y), -a*y/(x*d-c*y)+b*x/(x*d-c*y)]
[                           0, -c*y/(x*d-c*y)+d*x/(x*d-c*y)]
```

现在对 C1 与 C2 进行简化，结果如下：

```
>> a1=simple(C1);
>> a2=simple(C2);
>> a1
 a1 =
      [ (-c*b+a*d)/(x*d-c*y), (-y*a+x*b)/(x*d-c*y)]
      [                  0,                   1]
>> a2
 a2 =
      [ (-c*b+a*d)/(x*d-c*y), (-y*a+x*b)/(x*d-c*y)]
      [                  0,                   1]
```

经过简化处理后二者相等。如果需要清除这些变量，可以使用 clear 变量，代码如下：

```
>>clear a1 a2
```

这样 a1、a2 就从内存中删掉了。如果需要全部清除变量，则代码如下：

```
>>clear
```

5. 求符号矩阵的秩

MATLAB 中可以求符号矩阵的秩，代码如下：

```
>> A=sym('[a b ;c d]')
 A =
 [ a, b]
     [ c, d]
>> rank(A)
 ans =
    2
```

矩阵 $\boldsymbol{A}$ 的秩为 2。

6. 符号微分和差分

例如，对 $y=x^2$ 求导，代码如下：

```
>> y=sym(x^2)
 y =
 x^2
>> diff(y)
   ans =
      2*x
```

例如，$y=x^2\sin t$，求 $\frac{\mathrm{d}y}{\mathrm{d}t}$，代码如下：

```
>> y=sym('x^2*sin(t)')
 y =
x^2*sin(t)
>> diff(y,'t')
   ans =
       x^2*cos(t)
```

7. 符号代数方程求解

求下列线性方程组的解

$$\begin{cases}10x_1-x_2=9\\-x_1+10x_2-2x_3=7\\-2x_2+10x_3=6\end{cases}$$

首先将上面的线性方程组写成矩阵形式

$$\begin{pmatrix} 10 & -1 & 0 \\ -1 & 10 & -2 \\ 0 & -2 & 10 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}=\begin{pmatrix} 9 \\ 7 \\ 6 \end{pmatrix}$$

然后利用 MATLAB 中的矩阵形式求解，代码如下：

```
>> a=[10, -1, 0;-1, 10, -2;0, -2, 10]
a =
    10    -1     0
    -1    10    -2
     0    -2    10
>> b=[9;7;6]
b =
     9
     7
     6
>> linsolve(a,b)
ans =
    0.9958
    0.9579
0.7916
```

上面的解是近似解，下面用符号函数求解线性方程组 $\begin{cases} x+y=1 \\ x-y=0 \end{cases}$，代码如下：

```
>> s=solve('x+y=1','x-y=0','x','y')
s =
    x: [1x1 sym]
    y: [1x1 sym]
```

方程的解被保存在结构数组 s 中，s.x 为 x 的解，s.y 为 y 的解，查看内容如下：

```
>> s.x
 ans =
 1/2
 >> s.y
 ans =
 1/2
```

方程组的解为 $x=1/2$，$y=1/2$。

8. 非线性方程组求解

例如，求解非线性方程组 $\begin{cases} x+e^y=2 \\ y+e^x=2 \end{cases}$，代码如下：

```
>> s=solve('x+exp(y)=2','y+exp(x)=2','x','y')
s =
```

```
    x: [1x1 sym]
    y: [1x1 sym]
>> s.x
ans =
1.9952
>> s.y
ans =
-5.3542
```

方程组的解为 1.9952 和−5.3542。

9. 因式分解

MATLAB 中的因式分解函数是 factor，例如：

```
>> syms x
>> factor(x^3+1)
   ans =
   (x+1)*(x^2-x+1)
```

factor 函数也可以对整数进行素数分解，例如：

```
>> factor(3887)
ans =
13    13    23
```

从上面的结果可以看出，3887 素数分解为 $3887=13\times13\times23$。

2.5 MATLAB 编程基本知识

2.5.1 脚本文件与函数文件

如果完成一件比较复杂的任务，需要输入很多命令，直接从指令窗计算变得非常麻烦，这时只要把这些命令集成到一个文件中，每次只要在 MATLAB 中输入文件名就可以了。这时编写脚本文件最合适，脚本文件运行后，所产生的所有变量都保存在 MATLAB 基本工作区中。

函数文件是特殊的 m 文件，其常见格式如下：

```
function  输出变量列表=函数名（输入变量列表）
```

函数文件相当于对 MATLAB 进行了二次开发，其作用与形式与其他高级语言相同，都是为满足特定目标而编写的子函数。函数文件头标志是 function，脚本文件无此要求，脚本文件执行完后，变量仍然保存在内存中，而函数文件执行完后变量就被清除，除非对其进

行了全局变量申明。

2.5.2　编程注意事项

由于 MATLAB 是解释性语句，运行速度较 VC++、FORTRAN 等语言要慢得多，因此在编写一些大型程序时需要注意程序运行速度，尽量减少循环、判断等语句的使用，尽量用 MATLAB 自带的函数编写程序。为加快 MATLAB 运行速度，编写程序时应该注意以下内容。

运用矩阵运算，避免使用循环语句。因为 MATLAB 语言是一种解释性语言，循环语句运算耗费很多时间，因此计算中应尽可能地少用循环语句，多采用矩阵运算。如果需要进行多重循环，在循环的外环执行循环次数少的，内环执行循环次数多的。

大型矩阵预先定维。对于变量事先赋值，用以确定矩阵维数，例如 $m \times n$ 维矩阵变量 M，可以用 zeros(m,n)对 M 进行赋值。

优先考虑内部函数。矩阵运算时应该尽可能采用内部函数，因为这些函数是由更底层的 C 语言编写的，运行的速度要快于矩阵的循环运算。

优化算法。在近似计算时需要考虑步长、收敛性等问题，采用优化算法可以减少工作量，而且可以提高精度。

不要设计多用途、面面俱到的函数。多功能集于一身的函数，很可能使函数的理解、测试、维护等变得困难。

防止程序中的垃圾代码。序中的垃圾代码不仅占用额外的空间，而且常常影响程序的功能与性能，很可能给程序的测试、维护等造成不必要的麻烦。

防止输入错误。此类错误一般是由于把“<=”误写成“<”或“>=”误写成“>”等造成的，由此引起的后果，很多情况下是很严重的，所以编程时，一定要在这些地方小心，当编完程序后，应对这些操作符进行彻底检查。

2.5.3　程序排版格式

MATLAB 程序的排版格式应注意以下内容。

MATLAB 编程最好采用缩进方法编写。函数或过程的开始、结构的定义及循环、判断等语句中代码都要采用缩进风格，case 情况处理语句也要遵从语句缩进要求。

较长语句(>80 字符)要分成多行书写，长表达式要在低优先级操作符处划分新行，操作符放在新行之首，划分出的新行要进行适当缩进，使排版整齐，语句易读。

说明性部分应进行注释，注释应列出版权说明、版本号、生成日期、作者、内容、功能、与其他文件的关系、修改日志等，头文件注释中还应有函数功能的简要说明。

下面这段头文件头注释比较标准，当然并不局限于此格式。

```
**************************************************
 Copyright (C), 1988-1999, Huawei Tech. Co., Ltd.
 File name:                % 文件名
 Author:      Version: Date:    % 作者、版本及完成日期
 Description:              % 用于详细说明此程序文件完成的主要功能，与其他模块
                           % 或函数的接口，输出值、取值范围、含义及参数间的控
                           % 制、顺序、独立或依赖等关系
 Others:                   % 其他内容说明
 Function List:            % 主要函数列表，每条记录应包括函数名及功能的简要说明
1.
2.
 History:                  % 修改历史记录列表，每条修改记录应包括修改日期、修改
                           % 者及修改内容简述
   1. Date:                % 程序编写日期
   2. Author:              % 程序作者
   3. Modification:        % 程序修改者
```

思 考 题

1. 对结构数组进行赋值，s.x 内容为矩阵$\begin{pmatrix}1 & 4\\ 3 & 5\end{pmatrix}$，s.y 为字符串“MATLAB Financal ToolBox”。

2. 利用 diag 函数构建对角矩阵$\begin{pmatrix}1 & 2 & 3 & 1\\ 0 & 2 & 2 & 3\\ 0 & 0 & 3 & 2\\ 0 & 0 & 0 & 5\end{pmatrix}$。

3. 求解非线性方程组$\begin{cases}x^3+y=31\\ x+y=7\end{cases}$。

4. 求导数$y=\dfrac{e^x+x^{1/2}}{x}$。

5. 求矩阵$\begin{pmatrix}1 & 3.5 & 1.5\\ 0.5 & 3.4 & 6\\ 2.5 & 3 & 2\end{pmatrix}$的迹与秩。

6. 求解线性方程组$\begin{cases}x+y+z=4\\ 2x+y=3\\ y+z=3\end{cases}$。

第 3 章　金融时间序列数据分析

本章应重点掌握金融时间序列数组的各种赋值方法，时间序列变量的运算及索引，掌握从相关股票行情软件中导入数据到 MATLAB 中，学会计算时间序列的自相关系数和偏相关系数。学会调用时间序列的 GUI，掌握处理 MATLAB 缺失数据的基本方法。了解时间序列的 ARX 与 ARMAX 模型的检验、GARCH 模型的检验与模拟。

3.1　MATLAB 中时间序列变量的创立

3.1.1　时间序列数组的创立和数据文件的读取

由于金融数据大部分表现为时间序列，而时间序列数据的运算和绘图与一般的数据存在差别，因此 MATLAB 用时间序列格式保存时间序列数据。时间序列变量的后缀名为.fints，该变量把时间数据保存在第一列，其他列为观察值，时间序列变量运算时变量的内容发生变化，而时间不变。下面介绍 MATLAB 中时间变量的处理函数。

1．利用 fints 函数创立日期型数组

MATLAB 中创立日期型数组的函数是 fints，例如：

```
  >> price = [1:6]'
price =
     1
     2
     3
     4
     5
     6
>> dates = [today:today+5]'
dates =
      732721
      732722
      732723
      732724
      732725
      732726
>> tsobjkt = fints(dates, price)
```

```
tsobjkt =
   desc:  (none)
   freq:  Unknown (0)
   'dates:  (6)'    'series1:  (6)'
   '13-Feb-2006'    [            1]
   '14-Feb-2006'    [            2]
   '15-Feb-2006'    [            3]
   '16-Feb-2006'    [            4]
   '17-Feb-2006'    [            5]
   '18-Feb-2006'    [            6]
```

利用 whos 命令可查看内存中的变量，显示变量的信息如下：

```
>> whos
  Name           Size                 Bytes  Class
  price          6x1                  48  double array
  dates          6x1                  48  double array
  tsobjkt        6x1                  1266  fints object
Grand total is  66 elements using 1362 bytes
```

从输出结果中可以看出变量 tsobjkt 是 fints 型数组。

2．金融时间序列文件的读取

MATLAB 中的 ascii2fts 函数把 ASCII 文件的内容保存为 MATLAB 中的 fints 型时间序列变量。

调用方式

```
tsobjkt=ascii2fts(filename, timedata, descrow, colheadrow, skiprows)
```

输入参数

```
filename        %文件名，用单引号括起来
timedata        %判定是不是“天”数据，若是则输入字符串“t”，若不是则输入“nt”
descrow         %确定 ASCII 文件中文字说明的行数
colheadrow      %说明每列变量名所在的行数
skiprows        %ASCII 文件中不需要输入的列
```

输出参数

```
tsobjkt         %MATLAB 中 fints 型时间序列数据
```

例如，work 目录下有 at.txt 文件，用 DOS 命令下的 type 函数显示文件内容，代码如下：

```
>> !type at.txt
16-Feb-2006    1
17-Feb-2006    2
18-Feb-2006    3
```

```
19-Feb-2006    4
20-Feb-2006    5
21-Feb-2006    6
```

注意，在 Command 窗口下输入"!"后就可以直接执行 DOS 命令，例如用 DOS 中的 rename 命令改变文件的名称，代码如下：

```
>> !rename at.txt at.dat                %利用 DOS 命令将.txt 格式文件命名为.dat 文件
>> tsobjkt=ascii2fts('at.dat')          %读取 at.dat 文件中的数据
```

这样 at.txt 中的数据读入 MATLAB 中，变量名为 tsobjkt，类型为 fints 型数据，打开 tsobjkt 就可以看见具体内容，代码如下：

```
tsobjkt =
     desc:  at.dat
    freq:  Unknown (0)
    'dates:  (6)'    'series1:  (6)'
    '16-Feb-2006'    [             1]
    '17-Feb-2006'    [             2]
    '18-Feb-2006'    [             3]
    '19-Feb-2006'    [             4]
    '20-Feb-2006'    [             5]
    '21-Feb-2006'    [             6]
```

可以观察文件 at.txt 的内容，显示如下：

```
>> !type at.txt
ABC Company Stock
16-Feb-2006    1
17-Feb-2006    2
18-Feb-2006    3
19-Feb-2006    4
20-Feb-2006    5
21-Feb-2006    6
```

由于第一行不是数据而是描述数据的内容，因此直接调用函数 ascii2fts 就无法识别该文件，代码如下：

```
>> tsobjkt=ascii2fts('at.dat')
```

显示错误信息如下：

```
?? Error using ==> ascii2fts
Either the file contents do not correspond to the input
information, or the file contents contain time information
which was not indicated in the function call.
The actual error generated when reading the text file via
TEXTREAD is:
```

```
Trouble reading floating point number from file (row 1, field 2) ==> open
```

因子 at.dat 的第一行不是数字，ascii2fts 函数无法识别第一行内容，这时需要把第一行作为描述内容加以申明，执行如下命令：

```
>> tsobjkt=ascii2fts('at.dat',1)
```

这样 at.dat 的第一行就作为描述性内容保存到 tsobjkt 中，其内容如下：

```
tsobjkt =
     desc:  ABC Company Stock
    freq:  Unknown (0)
    'dates:  (6)'    'series1:  (6)'
    '16-Feb-2006'    [           1]
    '17-Feb-2006'    [           2]
    '18-Feb-2006'    [           3]
    '19-Feb-2006'    [           4]
    '20-Feb-2006'    [           5]
    '21-Feb-2006'    [           6]
```

进一步地，显示 at.dat 的内容如下：

```
>> !type at.dat
ABC Company Stock
DATE         CLOSE
16-Feb-2006    1
17-Feb-2006    2
18-Feb-2006    3
19-Feb-2006    4
20-Feb-2006    5
21-Feb-2006    6
```

at.dat 文件中描述的内容行数有两行，第一行说明是 ABC 公司的股价，第二行说明是日期与收盘价，at.dat 文件描述性内容的行数为 2 行，对数据的说明是第二行，执行下列命令：

```
>> tsobjkt=ascii2fts('at.dat',1,2)
```

这样 at.dat 的第一行就作为描述性内容保存到 tsobjkt 中。

```
tsobjkt =
     desc:  ABC stock prices
     freq:  Unknown  (0)
    'dates:  (6)'    'CLOSE:  (6)'
    '16-Feb-2006'    [          1]
    '17-Feb-2006'    [          2]
    '18-Feb-2006'    [          3]
    '19-Feb-2006'    [          4]
```

```
   '20-Feb-2006'    [          5]
   '21-Feb-2006'    [          6]
```

这样“close”也被读入 tsobjkt 中。

实际上 ascii2fts 也可以直接读入 txt 类型的文件，如可以执行下列命令：

```
>> tsobjkt=ascii2fts('at.txt')
 tsobjkt =
   desc:  at.txt
   freq:  Unknown (0)
   'dates:  (6)'    'series1:  (6)'
   '16-Feb-2006'    [          1]
   '17-Feb-2006'    [          2]
   '18-Feb-2006'    [          3]
   '19-Feb-2006'    [          4]
   '20-Feb-2006'    [          5]
   '21-Feb-2006'    [          6]
```

如果不需要读入第 4 列和第 5 列(注意日期不算)，可以将这两列用中括号括出来[4,5]，例如 ascii2fts(‘at.txt’，［4,5］)，这样 tsobjkt 就不会把 at.tat 的第 4 列和第 5 列读进来，代码如下：

```
>> !type at.dat
Staples, Inc. (SPLS)
Daily prices
DATE     OPEN    HIGH     LOW      CLOSE    VOLUME
Starting date: 04/08/1996
Ending date:   04/07/1999
4/8/96   19.50   19.75    19.25    19.375   548500
4/9/96   19.75   20.125   19.375   20       1135900
>> tsobjkt = ascii2fts('at.dat', 1, 3, [4 5])
tsobjkt =
desc:  Staples, Inc. (SPLS)
freq:  Unknown (0)
'dates: (2)'         'OPEN: (2)'    'HIGH: (2)'   'LOW:  (2)'
'08-Apr-1996'      [19.5000]      [19.7500]    [19.2500]
'09-Apr-1996'      [19.7500]      [20.1250]    [19.3750]
```

如需要读入小时数据，要规定读入文件的格式，代码如下：

```
>> type myfts_file2.txt
My FTS with Time
dates             times    Data1    Data2
01-Jan-2001       11:00    10.000000    5.000000
01-Jan-2001       12:00    2.000000     0.000000
02-Jan-2001       11:00    6.000000     8.000000
```

```
02-Jan-2001     12:00    5.000000     4.000000
03-Jan-2001     11:00    9.000000     6.000000
03-Jan-2001     12:00    8.000000     8.000000
>> MyFts = ascii2fts('myfts_file2.txt','t',1,2,1)
 MyFts =
    desc:  My FTS with Time
    freq:  Unknown (0)
    'dates:  (6)'     'times:  (6)'       'Data1:  (6)'          'Data2:  (6)'
    '01-Jan-2001'     '11:00'              [        10]          [         5]
    '     "     '     '12:00'              [         2]          [         0]
    '02-Jan-2001'     '11:00'              [         6]          [         8]
    '     "     '     '12:00'              [         5]          [         4]
    '03-Jan-2001'     '11:00'              [         9]          [         6]
    '     "     '     '12:00'              [         8]          [         8]
```

3.1.2 时间序列数组运算

1. 日期运算

1) 查找现在时刻

MATLAB 中查找现在时刻的函数是 now，表示从 0000 年 1 月 1 日至现在时刻的时间。

例如，现在时刻是 2007 年 1 月 22 日 17 点 44 分 49 秒，可以用 now 函数查找现在时刻，代码如下：

```
>> now
ans =
733064.7394612615
```

表示 0000 年 1 月 1 日至 2007 年 1 月 22 日 17 时 44 分 49 秒的时间是 733064.7394612615。

2) 查询当天日期

today 函数表示自 0000 年 1 月 1 日至当天的天数，如今天是 2006 年 2 月 6 日，星期一，用如下命令查找：

```
>> today
ans =
      732714
```

表示 0000 年 1 月 1 日至今天有 732714 天。

3) 序数型日期转换为字符串日期

datestr 函数和 today 函数的功能相反，如果知道距离 0000 年 1 月 1 日的天数，用 datestr 函数就可以计算出日期，例如查询第 732714 天的日期可以执行以下命令：

```
>> datestr(732714)
ans =
06-Feb-2006
```

如需要控制输出格式，还需增加更多控制参数，如果输出格式为月一天一年，小时:分钟:秒，每天 2 个 12 小时(mmmm, dd, yyyy HH:MM:SS.FFF AM)，需要调用形式 dt = datestr(now, 'mmmm dd, yyyy HH:MM:SS.FFF AM')，例如确定距离 0000 年 1 月 1 日为 733064.7394612615 的时刻，可以执行如下命令：

```
>> datestr(733064.7394612615,'mmmm dd, yyyy HH:MM:SS.FFF AM')
ans =
January 22, 2007  5:44:49.453 PM
```

对应的时刻为 2007 年 1 月 22 日下午 5 点 44 分 49.453 秒。如果写成符合我国书写习惯的日期型，可以输入下面命令：

```
>> datestr(733064.7394612615,'yyyy,mm, dd, HH:MM:SS.FFF AM')
ans =
2007,01, 22,  5:44:49.453 PM
```

4)　确定每月的第几天

调用方式

```
Day = day(D)
```

输入参数

```
D  %如果是整数表示从 0000 年 1 月 1 日至今的天数，也可以是日期型字符串
```

输出参数

```
Day  月的天数
```

例如我们计算 2007 年 1 月 22 日至今天的天数，代码如下：

```
>> day1=day('22-Jan-2007')
day1 =
    22
```

2007 年 1 月 22 日距离 0000 年 1 月 1 日的天数是 733064 天，同样可以确定天数，代码如下：

```
>> day1=day(733064)
day1 =
    22
```

5) 查询星期

MATLAB 中查询星期的函数是 weekday。

调用方式

```
[N, S] = weekday(D)
```

输入参数

```
D  %如果是整数表示从 0000 年 1 月 1 日至今的天数，也可以是日期型字符串
```

输出参数

```
N  %星期
S  %星期的缩写
```

2007 年 1 月 22 日距离 0000 年 1 月 1 日的天数是 733064 天，可以确定星期数，代码如下：

```
>> [n, s] = weekday(733064)
n =
     2
s =
Mon
```

整数 733064 表示的天数是 2007 年 1 月 22 日，n 表示是星期一，一周的第 2 天[①]，s 是星期一的缩写。实际上如果直接给出 2007 年 1 月 22 日的日期格式也可以得到同样的结果，代码如下：

```
>> [n, s] = weekday('22-Jan-2007')
n =
     2
s =
Mon
```

6) 查询月份

MATLAB 用 month 函数查询月份。

调用方式

```
[MonthNum, MonthString] = month(D)
```

输入参数

同上。

---

① 按照西方的习惯，星期日是一周的第一天。

输出参数

```
MonthNum        %月份
MonthString     %月份的缩写
```

7) 把字符串型日期转换为序数型日期

序数型日期表示距离 0000 年 1 月 1 日的天数，字符型日期转换为序数型日期的函数为 datenum，例如：

```
>> datenum('03-aug-2000')
ans =
     730701
```

上面的结果表示 2000 年 8 月 3 日距离 0000 年 1 月 1 日有 730701 天。

8) 计算日期间隔

计算两个日期之间的间隔天数需要用 daysact 函数，例如计算 2006 年 2 月 27 日至 2006 年 3 月 1 日之间的天数可以执行以下命令：

```
>> daysact('27-feb-2006','01-mar-2006')
ans =
     2
```

9) 特定日期抽取函数

在研究股市是否存在周末效应时就需要从日数据中抽取周、月、季度、半年度的数据。通常的处理方法是很烦琐的，但 MATLAB 金融时间序列工具箱有专门的函数完成此项功能。

(1) todaily 函数：从时间序列中抽取日数据。

(2) toweekly 函数：从时间序列中抽取周末数据。

(3) tomonthly 函数：从时间序列中抽取月末数据。

(4) toquarterly 函数：从时间序列中抽取季度末数据。

(5) tosemi 函数：从时间序列中抽取半年度末数据。

(6) toannual 函数：从时间序列中抽取年度末数据。

【例 3-1】下面以金牛股份(股票代码 000937)为例，说明如何将分析家软件的数据导入 MATLAB 中，然后进行相关操作。分析家软件是一款比较好的股票行情分析软件，可以从分析家网站(http://www.fenxijia.com)免费下载。

步骤如下：

(1) 打开分析家软件，输入股票代码 000937 就出现金牛股份股价图，日 K 线窗口如图 3.1 所示。

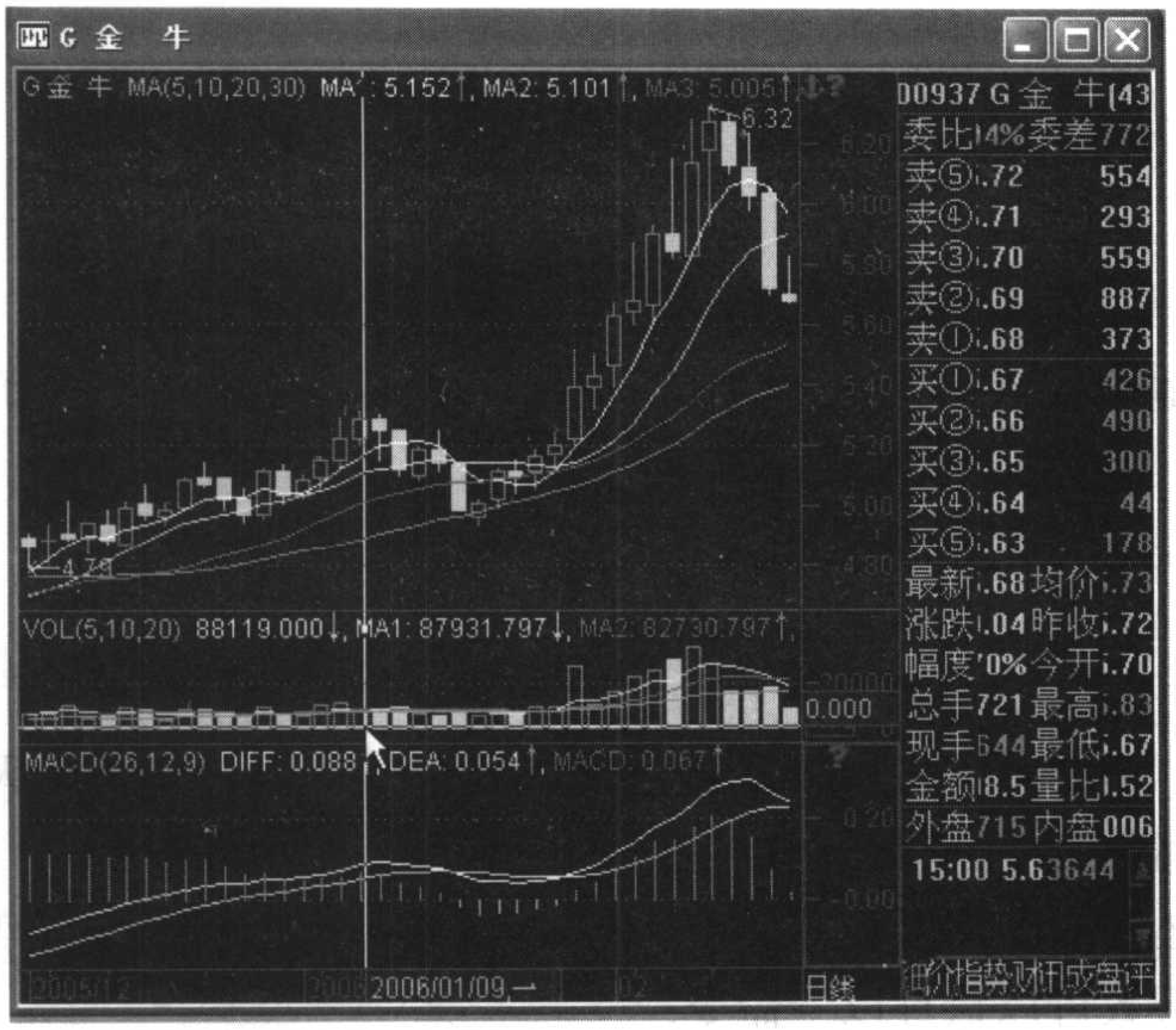


图 3.1 金牛股份的日 K 线

(2) 右击，在弹出的快捷菜单中选择【复制数据】命令，弹出一个提示框，如图 3.2 所示。

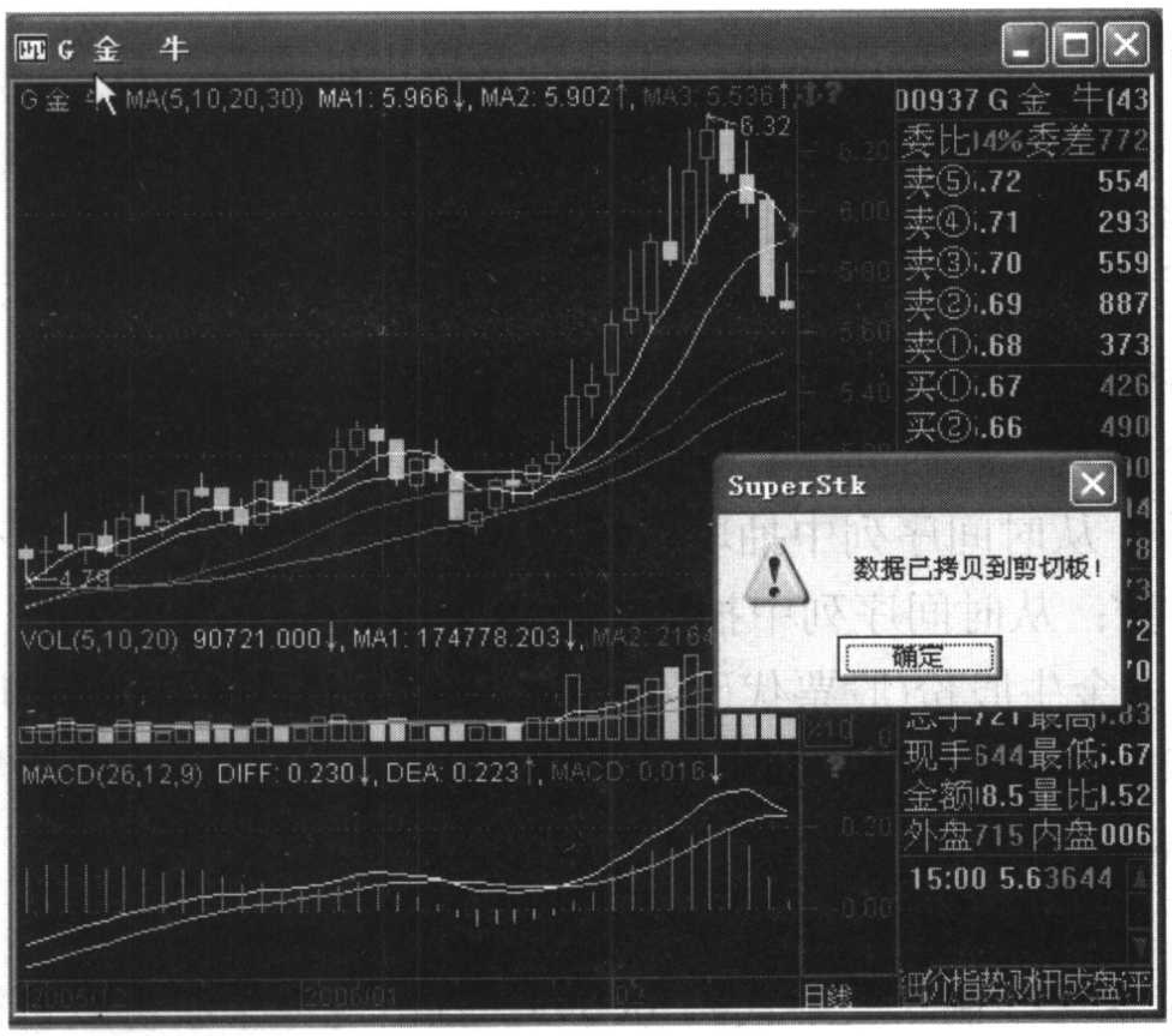


图 3.2 金牛股份日交易数据复制图

(3) 打开 Excel，将数据粘贴到 Excel 文件中，如图 3.3 所示。

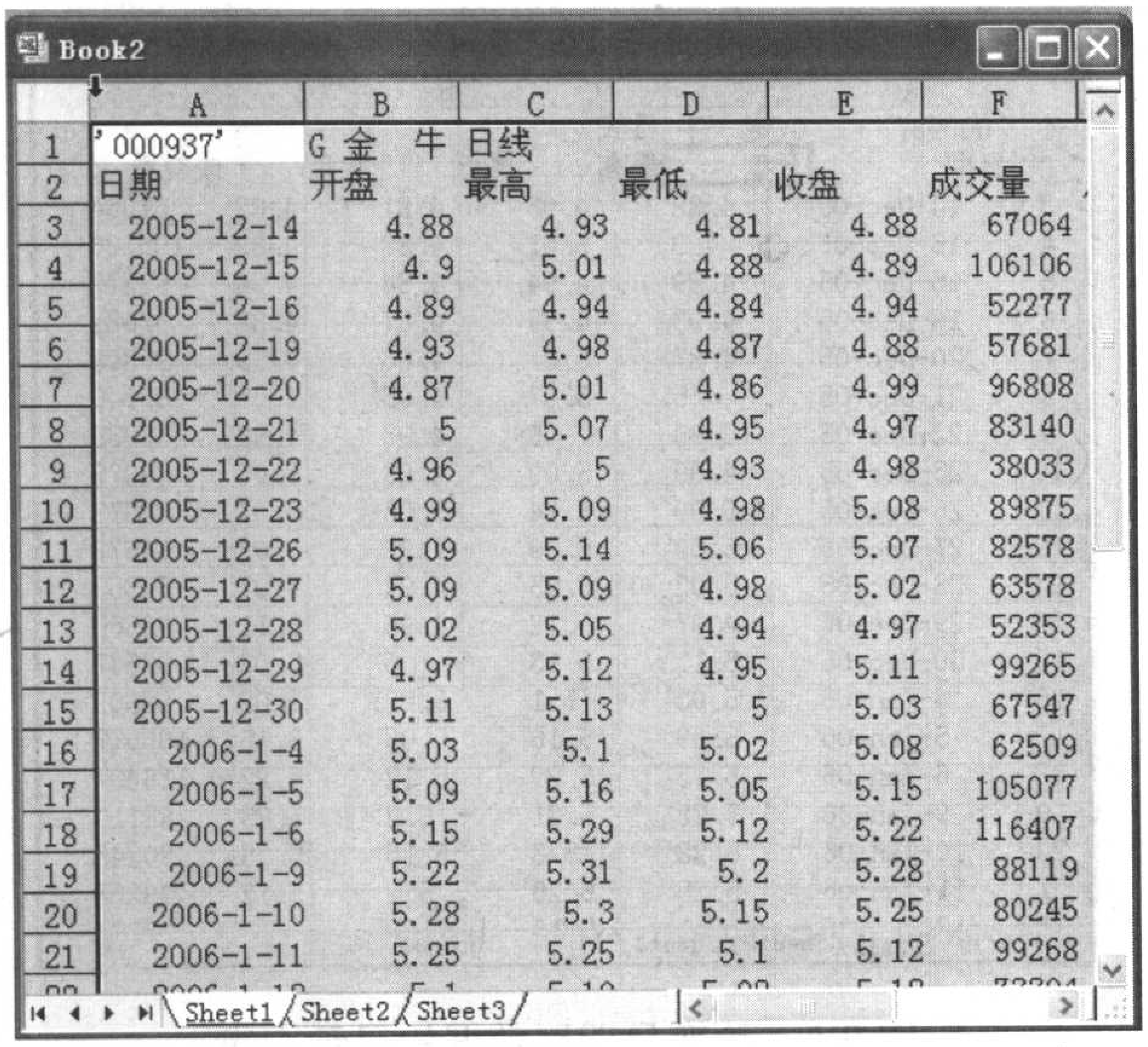


图 3.3 将分析家数据复制到 Excel 中

(4) 由于日期型格式不符合 MATLAB 默认格式，单击 A 列，右击，在弹出的快捷菜单中选择【设置单元格格式】[①]命令，就会出现如图 3.4 所示的对话框。

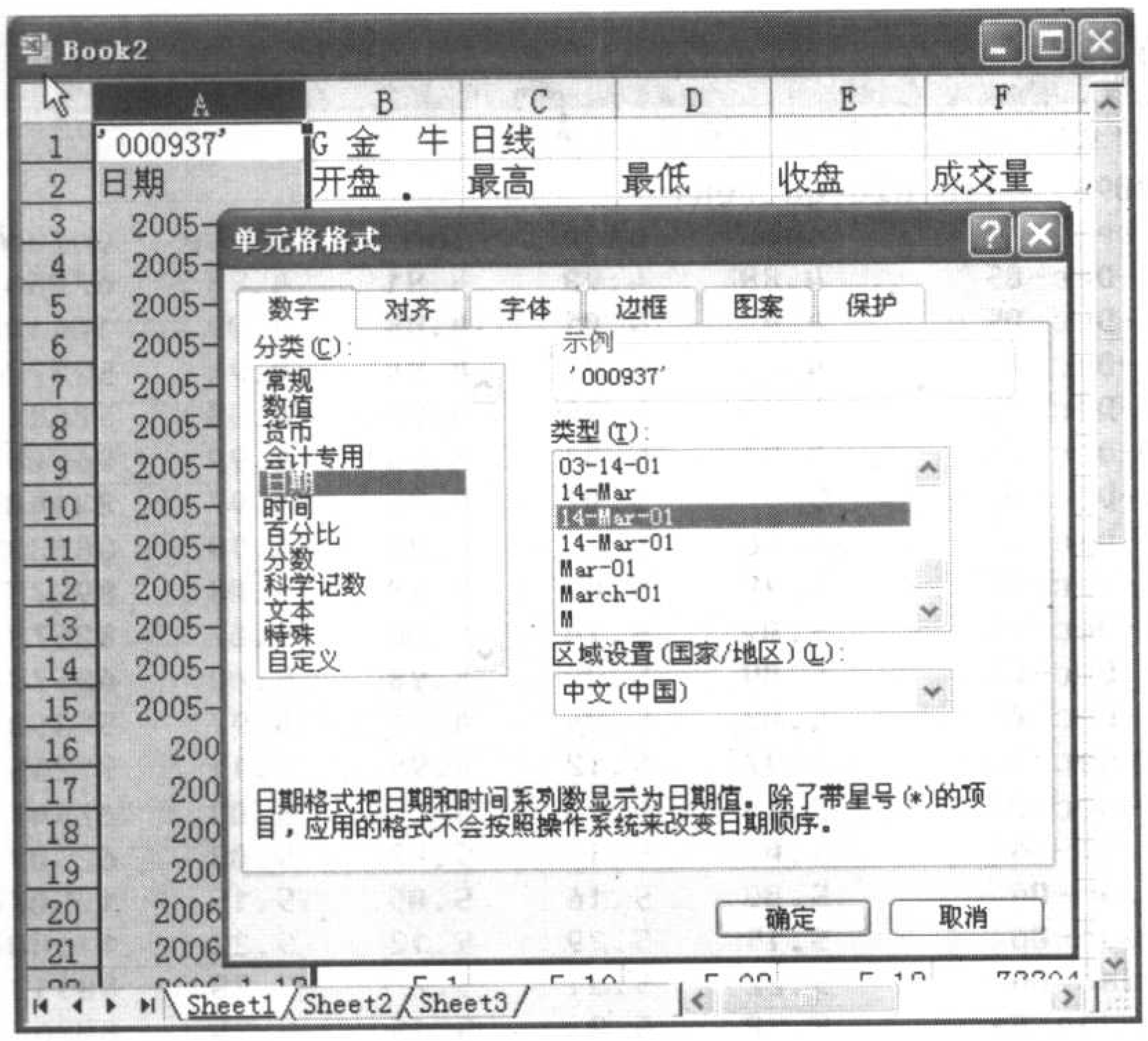


图 3.4 Excel 的日期设置

(5) 对日期型数据进行修改，选择如图 3.4 所示的格式，单击【确定】按钮，日期的显示格式发生了变化，如图 3.5 所示。

---

① MATLAB 中不能识别年－月－日的格式，应将其变成西方的日期格式。

Book2

	A	B	C	D	E	F
1	'000937'	G 金 牛	日线			
2	日期	开盘	最高	最低	收盘	成交量
3	14-Dec-05	4.88	4.93	4.81	4.88	67064
4	15-Dec-05	4.9	5.01	4.88	4.89	106106
5	16-Dec-05	4.89	4.94	4.84	4.94	52277
6	19-Dec-05	4.93	4.98	4.87	4.88	57681
7	20-Dec-05	4.87	5.01	4.86	4.99	96808
8	21-Dec-05	5	5.07	4.95	4.97	83140
9	22-Dec-05	4.96	5	4.93	4.98	38033
10	23-Dec-05	4.99	5.09	4.98	5.08	89875
11	26-Dec-05	5.09	5.14	5.06	5.07	82578
12	27-Dec-05	5.09	5.09	4.98	5.02	63578
13	28-Dec-05	5.02	5.05	4.94	4.97	52353
14	29-Dec-05	4.97	5.12	4.95	5.11	99265
15	30-Dec-05	5.11	5.13	5	5.03	67547
16	4-Jan-06	5.03	5.1	5.02	5.08	62509
17	5-Jan-06	5.09	5.16	5.05	5.15	105077
18	6-Jan-06	5.15	5.29	5.12	5.22	116407
19	9-Jan-06	5.22	5.31	5.2	5.28	88119
20	10-Jan-06	5.28	5.3	5.15	5.25	80245
21	11-Jan-06	5.25	5.25	5.1	5.12	99268

Sheet1 / Sheet2 / Sheet3

图 3.5　改变日期型数据的显示方式

(6) 拖动鼠标把所要复制的内容确定下来，右击，在弹出的快捷菜单中选择【复制】命令，打开 Windows 中的记事本，把复制的内容粘贴到记事本中，注意把汉字翻译成英文，这样便于 MATLAB 识别，如图 3.6 所示。

at - 记事本

文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

```
000937  jin niu gu fen
date            open    high    low     close   volum
14-Dec-05       4.88    4.93    4.81    4.88    67064
15-Dec-05       4.9     5.01    4.88    4.89    106106
16-Dec-05       4.89    4.94    4.84    4.94    52277
19-Dec-05       4.93    4.98    4.87    4.88    57681
20-Dec-05       4.87    5.01    4.86    4.99    96808
21-Dec-05       5       5.07    4.95    4.97    83140
22-Dec-05       4.96    5       4.93    4.98    38033
23-Dec-05       4.99    5.09    4.98    5.08    89875
26-Dec-05       5.09    5.14    5.06    5.07    82578
27-Dec-05       5.09    5.09    4.98    5.02    63578
28-Dec-05       5.02    5.05    4.94    4.97    52353
29-Dec-05       4.97    5.12    4.95    5.11    99265
30-Dec-05       5.11    5.13    5       5.03    67547
4-Jan-06        5.03    5.1     5.02    5.08    62509
5-Jan-06        5.09    5.16    5.05    5.15    105077
6-Jan-06        5.15    5.29    5.12    5.22    116407
9-Jan-06        5.22    5.31    5.2     5.28    88119
10-Jan-06       5.28    5.3     5.15    5.25    80245
```

图 3.6　粘贴数据到记事本

(7) 将数据保存到 MATLAB7/work 目录下的 at.txt 文件，用 type 命令显示内容如下：

```
>> !type at.txt
000937
date        open   high   low    close  volum
14-Dec-05   4.88   4.93   4.81   4.88   67064
15-Dec-05   4.9    5.01   4.88   4.89   106106
16-Dec-05   4.89   4.94   4.84   4.94   52277
19-Dec-05   4.93   4.98   4.87   4.88   57681
20-Dec-05   4.87   5.01   4.86   4.99   96808
21-Dec-05   5      5.07   4.95   4.97   83140
22-Dec-05   4.96   5      4.93   4.98   38033
23-Dec-05   4.99   5.09   4.98   5.08   89875
26-Dec-05   5.09   5.14   5.06   5.07   82578
27-Dec-05   5.09   5.09   4.98   5.02   63578
28-Dec-05   5.02   5.05   4.94   4.97   52353
29-Dec-05   4.97   5.12   4.95   5.11   99265
30-Dec-05   5.11   5.13   5      5.03   67547
4-Jan-06    5.03   5.1    5.02   5.08   62509
5-Jan-06    5.09   5.16   5.05   5.15   105077
6-Jan-06    5.15   5.29   5.12   5.22   116407
9-Jan-06    5.22   5.31   5.2    5.28   88119
```

下面用 ascii2fts 函数将 at.txt 中的数据读入 MATLAB 中，代码如下：

```
>> jinniu=ascii2fts('at.txt',2,2)
jinniu =
desc: date      open      high      low       close     volum
freq: Unknown (0)
'dates: (39)'    'open: (39)'   'high: (39)'   'low: (39)'   'close: (39)'   'volum: (39)'
'14-Dec-2005'    [  4.8800]     [  4.9300]     [  4.8100]    [  4.8800]      [   67064]
'15-Dec-2005'    [  4.9000]     [  5.0100]     [  4.8800]    [  4.8900]      [  106106]
'16-Dec-2005'    [  4.8900]     [  4.9400]     [  4.8400]    [  4.9400]      [   52277]
'19-Dec-2005'    [  4.9300]     [  4.9800]     [  4.8700]    [  4.8800]      [   57681]
'20-Dec-2005'    [  4.8700]     [  5.0100]     [  4.8600]    [  4.9900]      [   96808]
'21-Dec-2005'    [       5]     [  5.0700]     [  4.9500]    [  4.9700]      [   83140]
'22-Dec-2005'    [  4.9600]     [       5]     [  4.9300]    [  4.9800]      [   38033]
'23-Dec-2005'    [  4.9900]     [  5.0900]     [  4.9800]    [  5.0800]      [   89875]
'26-Dec-2005'    [  5.0900]     [  5.1400]     [  5.0600]    [  5.0700]      [   82578]
'27-Dec-2005'    [  5.0900]     [  5.0900]     [  4.9800]    [  5.0200]      [   63578]
```

这样 at.txt 的数据内容都保存在变量 jinniu 中，在 Command 窗口中执行 whos 命令查看 jinniu，代码如下：

```
>> whos
Name          Size                    Bytes  Class
jinniu       39x5                      3380  fints object
Grand total is 329 elements using 3380 bytes
```

变量 jinniu 的格式是 fints 型数据，下面将日期型数据变成周类型，代码如下：

```
>> weekjn=toweekly(jinniu)
```

weekjn 就是每周的时间、开盘、最高、最低、收盘、成交量，代码如下：

```
>> weekjn=toweekly(jinniu)
weekjn =
 desc: TOWEEKLY: date    open     high      low      close     volum
freq: Weekly (2)
'dates: (10)'    'open: (10)'  'high: (10)'  'low: (10)'  'close: (10)'   'volum: (10)'
'16-Dec-2005'   [   2.9340]   [   2.9760]   [  2.9060]   [    2.9420]   [4.5089e+004]
'23-Dec-2005'   [   4.9500]   [   5.0300]   [  4.9180]   [    4.9800]   [7.3107e+004]
'30-Dec-2005'   [   5.0475]   [   5.0975]   [  4.9675]   [    5.0325]   [7.0686e+004]
'06-Jan-2006'   [   5.0950]   [   5.1700]   [  5.0475]   [    5.1200]   [      87885]
'13-Jan-2006'   [   5.2060]   [   5.2600]   [  5.1280]   [    5.1940]   [8.0438e+004]
'20-Jan-2006'   [   5.0500]   [   5.1175]   [  5.0025]   [    5.0925]   [7.6823e+004]
'27-Jan-2006'   [   5.3120]   [   5.4500]   [  5.2520]   [    5.3760]   [1.3607e+005]
'03-Feb-2006'   [   5.3900]   [   5.5000]   [  5.3200]   [    5.4300]   [     104944]
'10-Feb-2006'   [   5.6980]   [   5.9800]   [  5.6480]   [    5.8380]   [     258138]
'17-Feb-2006'   [   6.0620]   [   6.1480]   [  5.8760]   [    5.9660]   [1.7478e+005]
```

将日期型数据变成月类型，只需执行下面的命令：

```
>> monthjn=tomonthly(jinniu)
monthkjn =
desc: TOMONTHLY: date    open     high      low      close     volum
freq: Monthly (3)
'dates: (3)'     'open: (3)'   'high: (3)'   'low: (3)'   'close: (3)'   'volum: (3)'
'30-Dec-2005'   [   4.2579]   [   4.3157]   [  4.2136]   [    4.2671]   [6.2409e+004]
'31-Jan-2006'   [   5.1975]   [   5.2850]   [  5.1370]   [    5.2280]   [9.7563e+004]
'28-Feb-2006'   [   5.7458]   [   5.9011]   [  5.6632]   [    5.7574]   [1.5914e+005]
```

2. 将时间序列数据转化为其他类型数据

1) 将时间序列数据保存为文本文件

调用方式

```
stat = fts2ascii(filename, tsobj, exttext)
```

输入参数

```
tsobj          %需要转化的 fints 型格式数据
filename       %新文件名称
exttext        %不需要的描述项的行数，也可以不输入
```

输出参数

```
stat           %转换成功标志，1 表示转换成功；0 表示转换不成功
```

例如 MATLAB 中的 fints 型变量 s 的内容如下：

```
>> s
s =
    desc:  (none)
```

```
    freq:  Unknown  (0)
    'dates:  (6)'    'series1:  (6)'
    '19-Feb-2006'    [        1.0100]
    '20-Feb-2006'    [        2.0200]
    '21-Feb-2006'    [        3.0300]
    '22-Feb-2006'    [        4.0400]
    '23-Feb-2006'    [        5.0500]
    '24-Feb-2006'    [        6.0600]
```

下面将变量 s 的内容保存到 aa.txt 文件中，代码如下：

```
>> a=fts2ascii('aa.txt',s)
a =
     1
```

a=1 表示转化成功，MATLAB 的 work 目录下生成了 aa.txt 文件，用 type 命令浏览文件中的内容如下：

```
>> type aa.txt
   dates          series1
19-Feb-2006       1.010000
20-Feb-2006       2.020000
21-Feb-2006       3.030000
22-Feb-2006       4.040000
23-Feb-2006       5.050000
24-Feb-2006       6.060000
```

2)　将时间序列数据转换为矩阵数据

将 fints 型数据转化为矩阵数据的函数是 fts2mat。

调用方式

```
tsmat = fts2mat(tsobj, datesflag)
```

输入参数

```
tsobj               %需要转换的原始数据
datesflag           %表示不输出日期到矩阵中 0(MATLAB 默认值)，1 表示输出日期到矩阵中
```

输出参数

```
tsmat  %转换后的矩阵
```

例如 s 是 fints 型数据，现转化为矩阵，如果不输入 datesflag，则日期就不会出现，代码如下：

```
>> fts2mat(s)
ans =
    1.0100
```

```
    2.0200
    3.0300
    4.0400
    5.0500
    6.0600
```

如果将 datesflag 设为 1，则日期也被输出，代码如下：

```
>> fts2mat(s,1)
ans =
1.0e+005 *
    7.3273    0.0000
    7.3273    0.0000
    7.3273    0.0000
    7.3273    0.0000
    7.3273    0.0001
    7.3273    0.0001
```

输出的第一列为序数型日期。

3) fints 型数据求最大值、最小值、均值、标准差和排序

对 fints 型数据求最大值、最小值、均值、标准差和排序的函数分别是 max、min、mean、std 和 sortfts。

4) 实现时间序列的转换

convertto 函数实现时间序列的转换功能，功能同 toweeekly、tomonthly 等函数。

调用方式

```
newfts = convertto(oldfts, newfreq)
```

输入参数

```
oldfts              %需要转换的数据
newfreq             %转换的目标，可以根据要求选择，具体如下：
                     "D"或"d"   为天
                     "W"或"w"   为周
                     "M"或"m"   为月
                     "Q"或"q"   为季度
                     "S"或"s"   为半年
                     "A"或"s"   为年
```

输出参数

```
newfts  %转换后的数据
```

对于前面的方法也可以用 convertto 函数实现，代码如下：

```
>> weekjn=convertto(jinniu,'w')
weekjn =
desc: TOWEEKLY: date     open     high     low      close     volum
freq: Weekly (2)
'dates: (10)'   'open: (10)'  'high: (10)'  'low: (10)'  'close: (10)'  'volum: (10)'
'16-Dec-2005'   [  2.9340]    [  2.9760]    [  2.9060]   [  2.9420]     [4.5089e+004]
'23-Dec-2005'   [  4.9500]    [  5.0300]    [  4.9180]   [  4.9800]     [7.3107e+004]
'30-Dec-2005'   [  5.0475]    [  5.0975]    [  4.9675]   [  5.0325]     [7.0686e+004]
'06-Jan-2006'   [  5.0950]    [  5.1700]    [  5.0475]   [  5.1200]     [      87885]
'13-Jan-2006'   [  5.2060]    [  5.2600]    [  5.1280]   [  5.1940]     [8.0438e+004]
'20-Jan-2006'   [  5.0500]    [  5.1175]    [  5.0025]   [  5.0925]     [7.6823e+004]
'27-Jan-2006'   [  5.3120]    [  5.4500]    [  5.2520]   [  5.3760]     [1.3607e+005]
'03-Feb-2006'   [  5.3900]    [  5.5000]    [  5.3200]   [  5.4300]     [     104944]
'10-Feb-2006'   [  5.6980]    [  5.9800]    [  5.6480]   [  5.8380]     [     258138]
'17-Feb-2006'   [  6.0620]    [  6.1480]    [  5.8760]   [  5.9660]     [1.7478e+005]

>> monthjn=convertto(jinniu,'m')
monthjn =
desc: TOMONTHLY: date     open     high     low      close     volum
freq: Monthly (3)
'dates: (3)'    'open: (3)'   'high: (3)'   'low: (3)'   'close: (3)'   'volum: (3)'
'30-Dec-2005'   [  4.2579]    [  4.3157]    [  4.2136]   [  4.2671]     [6.2409e+004]
'31-Jan-2006'   [  5.1975]    [  5.2850]    [  5.1370]   [  5.2280]     [9.7563e+004]
'28-Feb-2006'   [  5.7458]    [  5.9011]    [  5.6632]   [  5.7574]     [1.5914e+005]
```

5)　时间序列数据的抽取

extfield 函数实现时间序列数据的抽取。

调用方式

```
ftse = extfield(tsobj, fieldnames)
```

输入参数

```
tsobj          %原始数据
fieldnames     %原始数据中的字段名
```

输出参数

```
ftse           %需要的数据
```

如在数据 weekjn 中我们只要收盘价，并且命名为变量 weekjnclose，这时可以执行如下命令：

```
>> weekjnclose=extfield(weekjn,'close')
 weekjnclose =
   desc:  TOWEEKLY: date        open    high     low     close   volum
   freq:  Weekly (2)
```

```
'dates:  (10)'    'close:  (10)'
'16-Dec-2005'     [      2.9420]
'23-Dec-2005'     [      4.9800]
'30-Dec-2005'     [      5.0325]
'06-Jan-2006'     [      5.1200]
'13-Jan-2006'     [      5.1940]
'20-Jan-2006'     [      5.0925]
'27-Jan-2006'     [      5.3760]
'03-Feb-2006'     [      5.4300]
'10-Feb-2006'     [      5.8380]
'17-Feb-2006'     [      5.9660]
```

这样 weekjn 数据中的收盘价就保存到 weekjnclose 中。

实际上在 MATLAB 中，weekjn 是一个结构数据，close 是 weekjn 的一个属性，可以用结构变量的方法查看其内容，代码如下：

```
>> weekjnclose=weekjn.close
 weekjnclose =
   desc:  TOWEEKLY: date        open     high     low     close    volum
   freq:  Weekly (2)
   'dates:  (10)'    'close:  (10)'
   '16-Dec-2005'     [      2.9420]
   '23-Dec-2005'     [      4.9800]
   '30-Dec-2005'     [      5.0325]
   '06-Jan-2006'     [      5.1200]
   '13-Jan-2006'     [      5.1940]
   '20-Jan-2006'     [      5.0925]
   '27-Jan-2006'     [      5.3760]
   '03-Feb-2006'     [      5.4300]
   '10-Feb-2006'     [      5.8380]
   '17-Feb-2006'     [      5.9660]
```

6) 将价格时间序列转化为收益率时间序列

price2ret 函数表示将价格时间序列转换为收益率时间序列，计算方法为 $r_t = \log(P_t / P_{t-1})$，下面是一个例子：

```
>> s=100*exp(0.1*[0:5])
s =100.0000  110.5171  122.1403  134.9859  149.1825  164.8721
>> price2ret(s)
ans =0.1000    0.1000    0.1000    0.1000    0.1000
```

7) 将收益率时间序列转化为价格时间序列

ret2price 函数可以将收益率时间序列转化为价格时间序列，计算方法为 $P_t = P_{t-1} * \exp(r_t)$，下面是一个例子：

```
>> s=[0.1 0.1 0.1 0.1 0.1 0.1]
s =0.1000    0.1000    0.1000    0.1000    0.1000    0.1000
>> p=100*ret2price(s)
p =100.0000  110.5171  122.1403  134.9859  149.1825  164.8721  182.2119
```

3. 处理时间序列中缺失的数据

金融时间序列的数据有时会有一些数据的缺失，需要将缺失的数据补齐后才能分析，fillts 函数可以利用插值的方法把缺失的数据补齐。

调用方式

```
newfts = fillts(oldfts, method)
```

输入参数

```
oldfts  %原始数据
method  %处理缺失值的方法，主要有如下几种：
 'linear' 或 'le'     %线性插值法(MATLAB 默认值)
 'cubic'  或 'c'      %3 次插值
 'spline' 或 's'      %样条法
 'nearest' 或 'n'     %最近法
 'pchip'  或 'p'      %逐段光滑的 3 次 Hemite 多项式法
```

输出参数

```
newfts  %处理后的数据
```

下面是一个处理缺失数据的例子，代码如下：

```
>> randn('seed',0);        % 随机数初值为 0
>> a=randn(6,1);           % 生成 6 行 1 列的随机数向量
>> b=[today:today+5]'      % 从今天到后面 5 天
b =
     732979
     732980
     732981
     732982
     732983
     732984
>> fts=fints(b,a)  %生成 fints 格式数据
 fts =
    desc:  (none)
   freq:  Unknown (0)
   'dates:  (6)'    'series1:  (6)'
   '29-Oct-2006'    [        1.1650]
   '30-Oct-2006'    [        0.6268]
   '31-Oct-2006'    [        0.0751]
   '01-Nov-2006'    [        0.3516]
```

```
   '02-Nov-2006'    [      -0.6965]
   '03-Nov-2006'    [       1.6961]
>> fts(3)=NaN; %将第3个数据变为缺失值NaN
>> fts(3)         %查看改变后的fts变量中第3个数据的值
ans =
    desc:  (none)
   freq:  Unknown (0)
   'dates:  (1)'    'series1:  (1)'
   '31-Oct-2006'    [          NaN]
>> newdata=fillts(fts,'linear') %调用fillts函数中的线性插值处理fts中的缺失值
newdata =
    desc:  Filled
   freq:  Unknown (0)
   'dates:  (6)'    'series1:  (6)'
   '29-Oct-2006'    [       1.1650]
   '30-Oct-2006'    [       0.6268]
   '31-Oct-2006'    [       0.4892]
   '01-Nov-2006'    [       0.3516]
   '02-Nov-2006'    [      -0.6965]
   '03-Nov-2006'    [       1.6961]
```

从上面的结果可以知道，缺失值变为 0.4892。

3.2 金融时间序列的统计特征

MATLAB 自带的金融时间序列数据有 3 个，分别是 DEM2GBP、NASDAQ 和 NYSE。DEM2GBP 为德国马克兑英镑的日时间序列，时间为 1984 年 2 月 2 日到 1991 年 12 月 31 日，共计 1975 个观察样本。NASDAQ 为美国 NASDAQ 证券市场的日收盘指数，时间为 1990 年 1 月 2 日到 2001 年 12 月 31 日，共计 3028 个观察样本，NASDAQ 收盘数据可以从 NASDAQ 网站上得到，网址是 http://www.marketdata.nasdaq.com/mr4b.html。NYSE 为纽约证券交易所的日收盘指数，时间为 1990 年 1 月 2 日到 2001 年 12 月 31 日，共计 3028 个观察样本，NYSE 可以从 NYSE 网站上得到，网址是 Http://www.nyse.com/markinfo/markinfo.html。上述 3 个数据包含在 MAT 型数据文件中，文件名为 garchdata.mat。

3.2.1 相关系数和偏相关系数

相关性用于描述两个或两个以上变量之间的非确定性关系，两个变量之间的相关性叫做简单相关，三个或三个以上变量之间的相关性叫做多重相关。衡量相关性的指标主要有相关系数与偏相关系数。

1. 相关系数

衡量两个变量之间(假设分别为 x_i, y_i)线性关系程度的数量指标称为相关系数。

$$r = \frac{\sum(x_i - \overline{x})\sum(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}\sqrt{\sum(y_i - \overline{y})^2}}$$

其中，x_i, y_i 是观察值，$\overline{x}, \overline{y}$ 是 x_i, y_i 的均值。

MATLAB 中计算相关系数的函数是 corrcoef，其用法如下。

调用方式

```
R = corrcoef(x,y)
```

输入参数

```
x  %观察变量
y  %观察变量
```

输出参数

```
R  %观察变量的相关系数矩阵
```

下面求 MATLAB 自带数据 NYSE 与 NASDAQ 之间的相关系数矩阵，代码如下：

```
>> load garchdata
>> corrcoef(NASDAQ,NYSE)
ans =
    1.0000    0.9052
    0.9052    1.00002
```

2. 偏相关系数

一般地，在多个变量 $y, x_1, x_2, \cdots, x_k$ 之间，如果只考虑 y 与 $x_i(i=1,2,3,\cdots,k)$ 之间的相关性，而消除其他变量的影响，这种相关叫**偏相关**。

调用方式

```
[a,b] = parcorr(Series)
[PartialACF,Lags,Bounds] = parcorr(Series,nLags,R,nSTDs)
```

输入参数

```
series          %时间序列数据
nLags           %偏相关系数的阶数，默认值为 20
R               %AR(R)模型时间序列的阶数 R，要求是非负整数，而且大于 nLags。当 R= []
                 时，系统默认为 0。在计算一般的偏相关系数时，我们假定 nLags 阶数以后
                 不存在相关，但有时这并不符合实际情况。如果时间序列是 AR(10)时计算 2
                 阶偏相关系数，就不能假设大于 2 阶后的相关性为 0，而应该把 AR(10)模型
                 告诉系统
```

```
nSTDs           %偏相关系数的 95%置信水平下的标准差，为σ/√n ， n 为样本数
```

输出参数

```
a               %阶数
b               %与阶数对应的偏相关系数
PartialACF      %偏相关系数
Lags            %滞后阶数
Bounds          %偏相关系数的置信区间
>> load garchdata;                       %读取 MATLAB 自带的数据
>> demo2gbp=price2ret(DEM2GBP);          %将价格转化为收益率序列
>> parcorr(demo2gbp);                    %求收益率序列的偏相关系数
```

偏相关系数结果如图 3.7 所示。

图 3.7 中上下两条横线分别表示偏相关系数的上下界(置信度为 0.05)，超出边界的表示存在相关性，从图中可以看出德国马克兑英镑存在 14 阶的自相关。

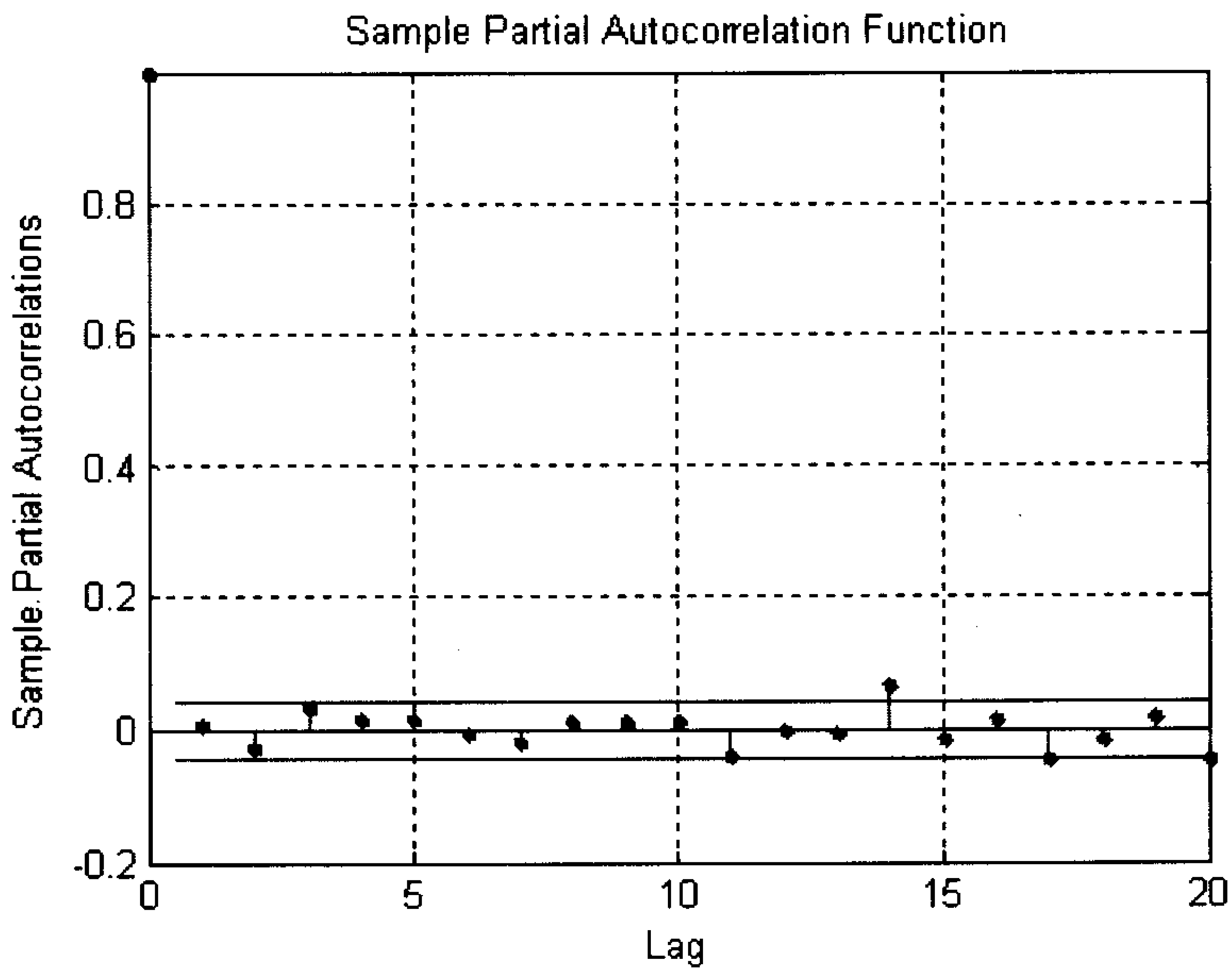


图 3.7　偏相关系数图

如果需要保存偏相关系数，可以执行以下命令：

```
>> [a,b]=parcorr(demo2gbp);
>> [b,a]
ans =
     0    1.0000
```

```
   1.0000    0.0094
   2.0000   -0.0254
   3.0000    0.0347
   4.0000    0.0187
   5.0000    0.0189
   6.0000   -0.0029
   7.0000   -0.0166
   8.0000    0.0149
   9.0000    0.0146
  10.0000    0.0126
  11.0000   -0.0374
  12.0000   -0.0011
  13.0000   -0.0049
  14.0000    0.0693
  15.0000   -0.0120
  16.0000    0.0156
  17.0000   -0.0433
  18.0000   -0.0108
  19.0000    0.0224
  20.0000   -0.0444
```

这样各阶的偏相关系数都保存在数组 a 中，可以看出时间序列 demo2gbp 的一阶偏相关系数为 0.0094，二阶偏相关系数为-0.0254，三阶偏相关系数为 0.0347，其他阶数的偏相关系数以此类推。

3. 自相关系数

下面是计算自相关系数的例子，代码如下：

```
>> load garchdata;                    %读取 MATLAB 自带的数据
>> demo2gbp=price2ret(DEM2GBP);       %将价格序列转化为收益率序列
>> autocorr(demo2gbp);                %求收益率序列的自相关系数
```

自相关系数结果如图 3.8 所示。

图 3.8 中上下两条横线分别表示自相关系数的上下界，超出边界的部分表示存在相关关系。

如果需要知道各阶的相关系数，可以执行以下命令：

```
>> [a,b]=autocorr(demo2gbp);
>> [b,a]
ans =
   0    1.0000
   1.0000    0.0094
   2.0000   -0.0253
   3.0000    0.0342
```

```
 4.0000    0.0200
 5.0000    0.0175
 6.0000   -0.0024
 7.0000   -0.0162
 8.0000    0.0163
 9.0000    0.0162
10.0000    0.0111
11.0000   -0.0374
12.0000   -0.0013
13.0000   -0.0010
14.0000    0.0673
15.0000   -0.0120
16.0000    0.0099
17.0000   -0.0367
18.0000   -0.0089
19.0000    0.0260
20.0000   -0.0470
```

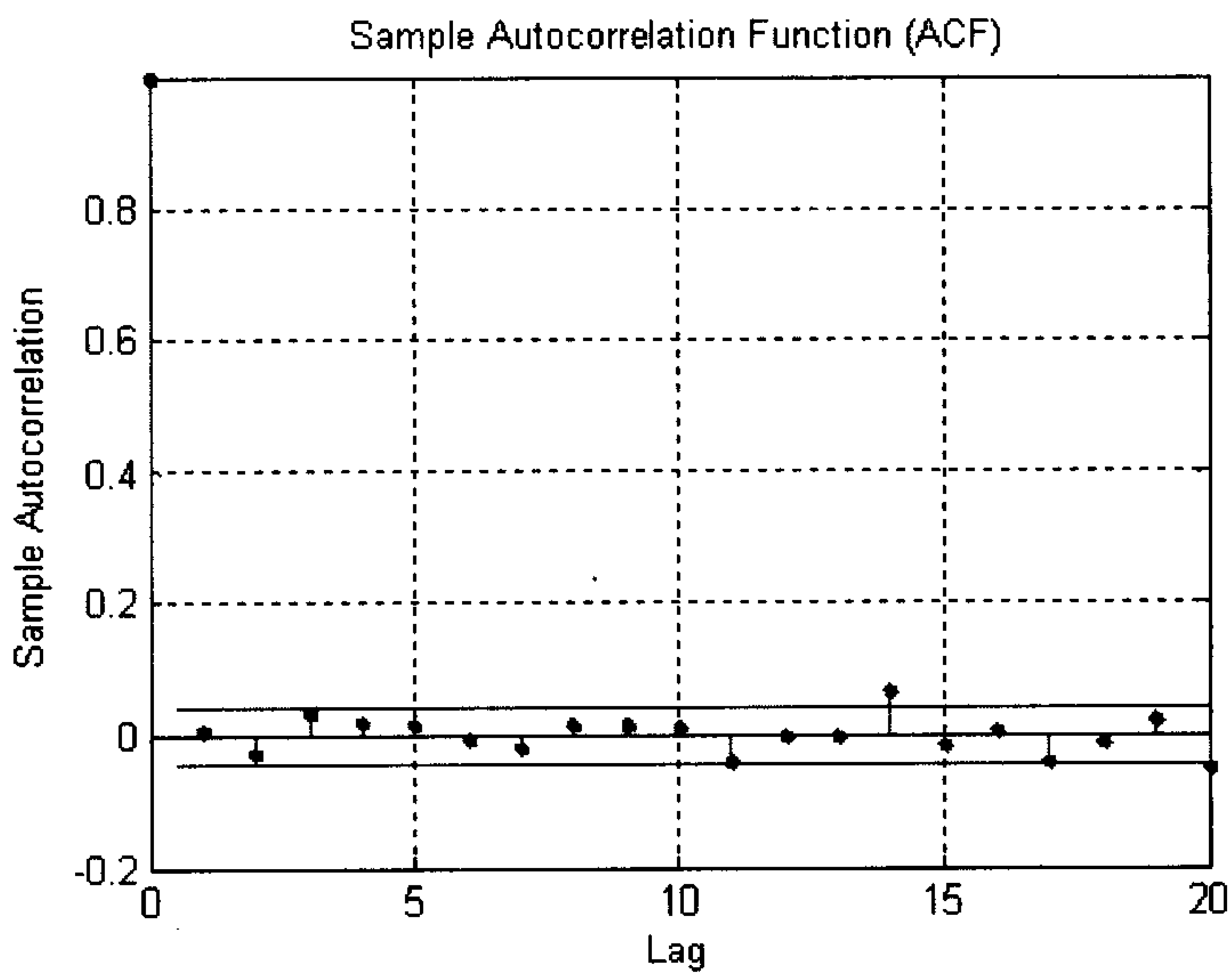


图 3.8　自相关系数图

这样数组 a 就储存了各阶的自相关系数，数组 a 的第一个数为 1.0000，表示 0 阶自相关系数，可以不予以考虑。而是从第二个数开始计算自相关系数，例如第二个数 0.0094 表示一阶自相关系数为 0.0094，第三个数表示二阶自相关系数为-0.0253，其他阶数的自相关系数情况以此类推。

3.2.2　金融时间序列界面功能介绍

MATLAB 中自带了金融时间序列的用户图形界面(GUI)，GUI 不需要在 Command 窗口下进行操作，只要直接对时间序列数据进行处理，就可以把结果用图形表示出来，充分发挥 MATLAB 强大的图形图像功能。下面具体介绍 GUI 菜单内容，输入如下代码：

```
>>ftsgui
```

时间序列主窗口如图 3.9 所示。

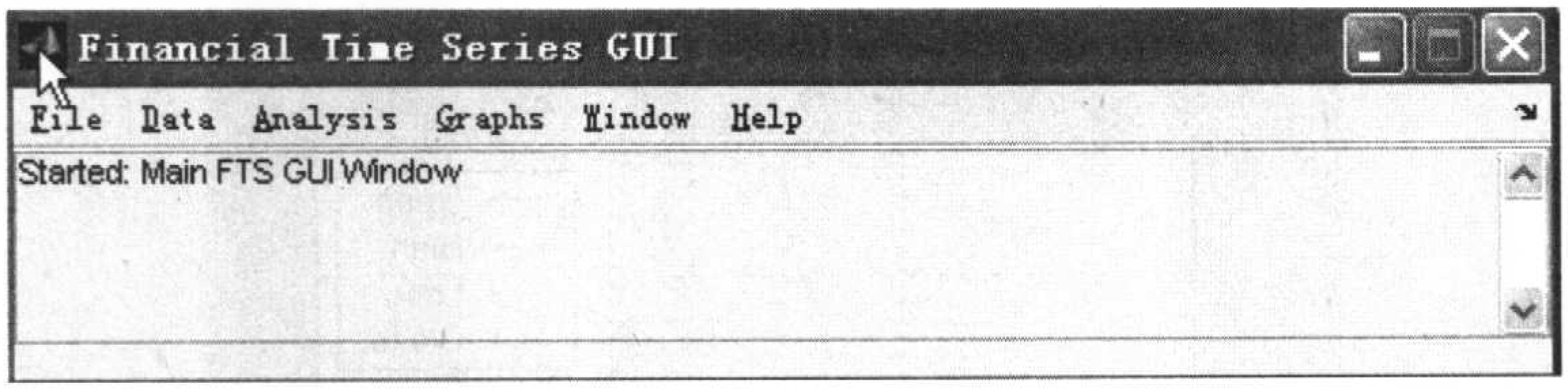


图 3.9　MATLAB 中的时间序列 GUI

主窗口有 6 个菜单选项，分别是 File、Data、Analysis、Graphs、Window 和 Help。举例来说，MATLAB 自带有迪斯尼公司的股价数据文件，数据文件名是 disney.mat，disney.mat 数据中含有迪斯尼股价的时间、开盘价、最高价、最低价、收盘价等内容，而且有缺失的数据，缺失的数据内容为 NaN 符号。在 C:\MATLAB\7\toolbox\ftseries\ftsdata 目录下可以找到该文件。调入 disney.mat 文件后，主界面窗口如图 3.10 所示。

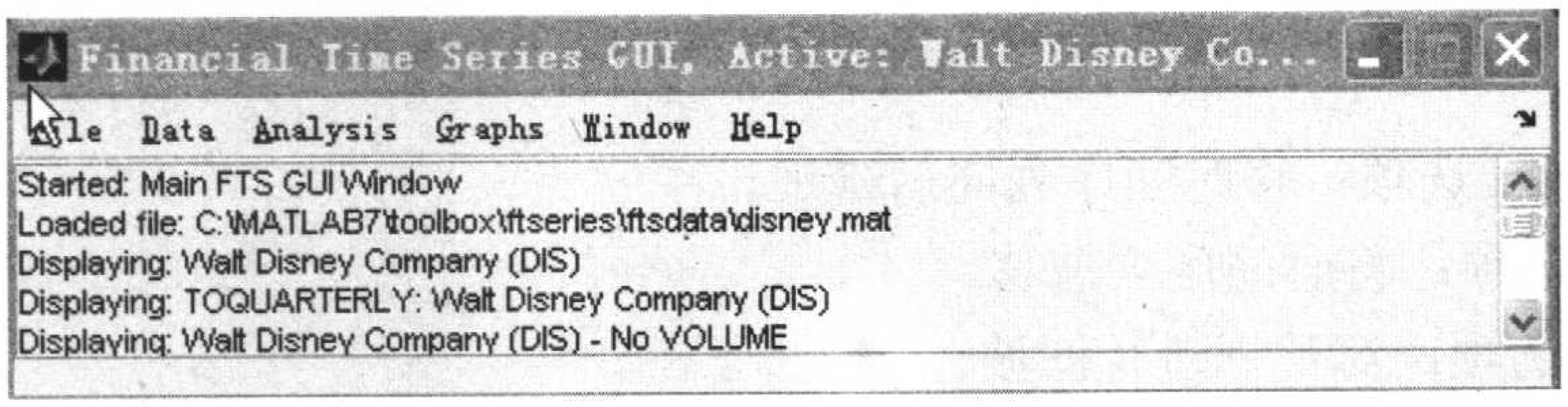


图 3.10　MATLAB 中的时间序列 GUI 数据文件窗口

同时还出现另外 3 个窗口，如图 3.11 所示。这是因为 disney 文件中含有 3 个.fints 型的数组 dis、dis_nv 和 q_dis，3 个窗口分别对应于不同的数组。

下面分别介绍一下各菜单的主要功能。

1)　File 菜单

主要负责文件的输入与输出、关闭 GUI 窗口、打印分析结果以及图像等功能。

2)　Data 菜单

主要是数据的处理功能，其选项的功能如下。

(1)　Fill Missing Data 选项：用插值法处理缺失数据，方法有 Linear(线性插值)、Cubic(3 次插值)、Spline(3 次样条插值)、Nearest(最近点插值)、Pchip(逐段光滑的 3 次 Hemite 插值

多项式法)和 Constant(用户自行输入)。

(2) Smooth Data 选项：平滑数据，方法有 Box method、Gauss method 和 Exponential method 3 种。

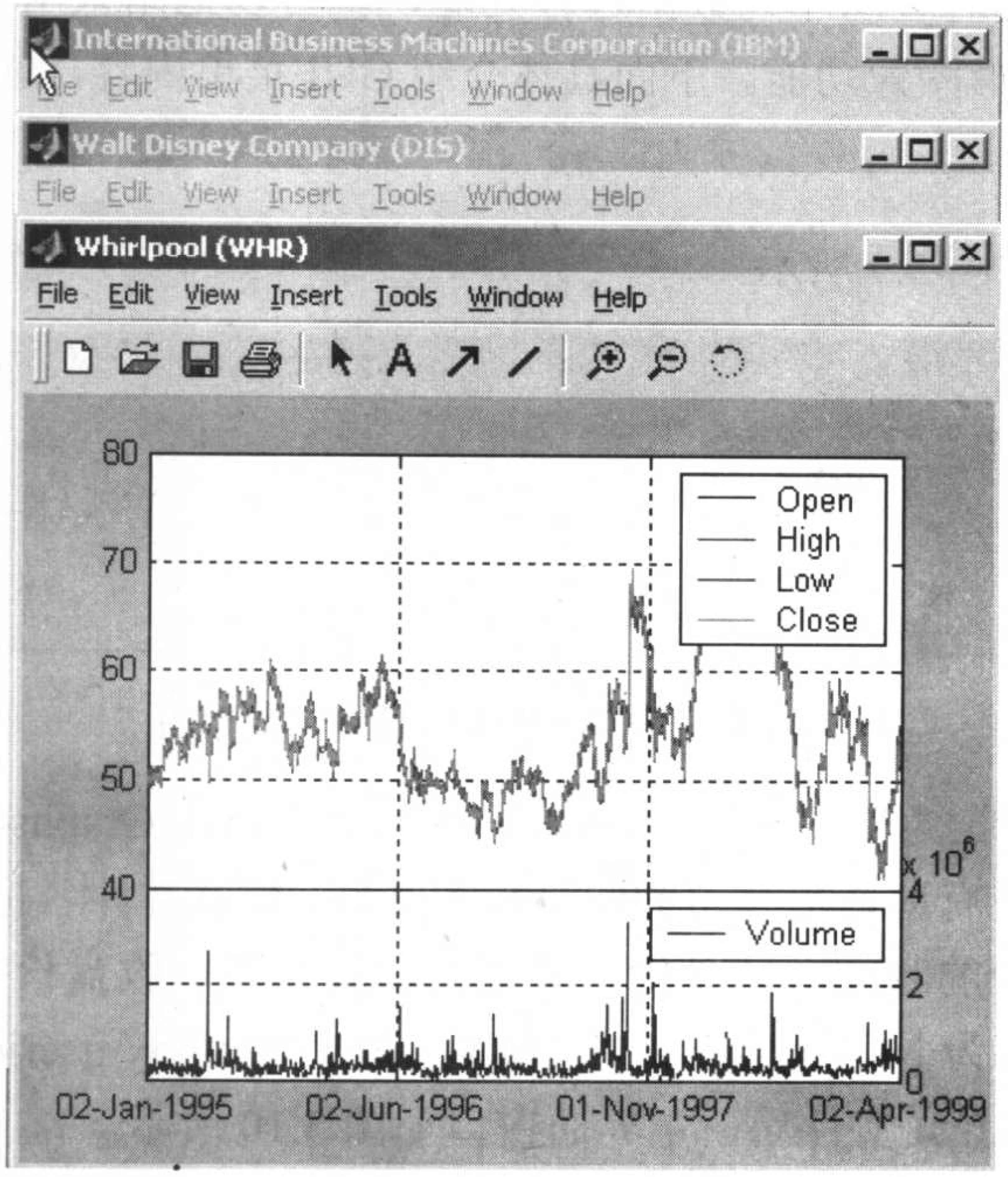


图 3.11 时间序列 GUI 中迪斯尼股价图

(3) Lag Data 选项：将时间序列向后延迟。

(4) Lead 选项：将时间序列前移。

(5) Filter 选项：对数据进行滤波。

(6) Box-Cox 变换：Box-Cox 变换可以将非正态分布变为正态分布，属于指数类型的变换。Box-Cox 变换公式如下：

$$data(\lambda)=\begin{cases}\dfrac{data^{\lambda}-1}{\lambda} & \lambda\neq 0\\ \log(data) & \lambda=1\end{cases}$$

其中 λ 的选择可以使得对数极大似然函数达到最大值。

(7) Convert Data Frequency To 选项：分析周期转换，将时间序列转化为日数据、周数据、月数据、季度数据、半年数据和年数据。

3) Analysis 菜单

主要是数据分析，包括了对股票价格进行技术分析的函数，选项有 Exp(对序列指数化)、Log(以 e 为底对时间序列取对数)、Log10(以 10 为底对时间序列取对数)和 Log2(以 2 为底对

时间序列取对数)，具体选项的功能如下。

(1) Basic Statistics 选项：时间序列的基本统计功能，如均值、方差等统计特征。

(2) Difference 选项：对时间序列进行一阶差分。

(3) Periodic Average 选项：对任意时间段的数据取平均，方法有二，一是直接输入研究的交易时间段的长度；二是输入开始日期与截止日期。

(4) Technical Analysis 选项：对股价进行技术分析，常用的 MACD、RSI、OBV 能量潮指标等都在其中。

4) Graphs 菜单

主要功能是对股票的价格进行作图。

(1) Line Plot 选项：股价的折线图。

(2) Bar Chart 选项：股价的 Bar 图。

(3) Horizontal Bar Chart 选项：股价的水平 Bar 图。

(4) 3D Bar Chart 选项：股价的三维 Bar 图。

(5) Horizontal 3D Bar Chart 选项：股价的水平三维 Bar 图。

(6) Candle Plot 选项：股价的蜡烛图，中国与日本等国喜欢用蜡烛图分析股价。

(7) Hight-Low 选项：股价的高低图，英美投资者喜欢用股价的高低图分析股价。

(8) Interactive Chart 选项：股价的互动图，迪斯尼股价的互动图如图 3.12 所示。

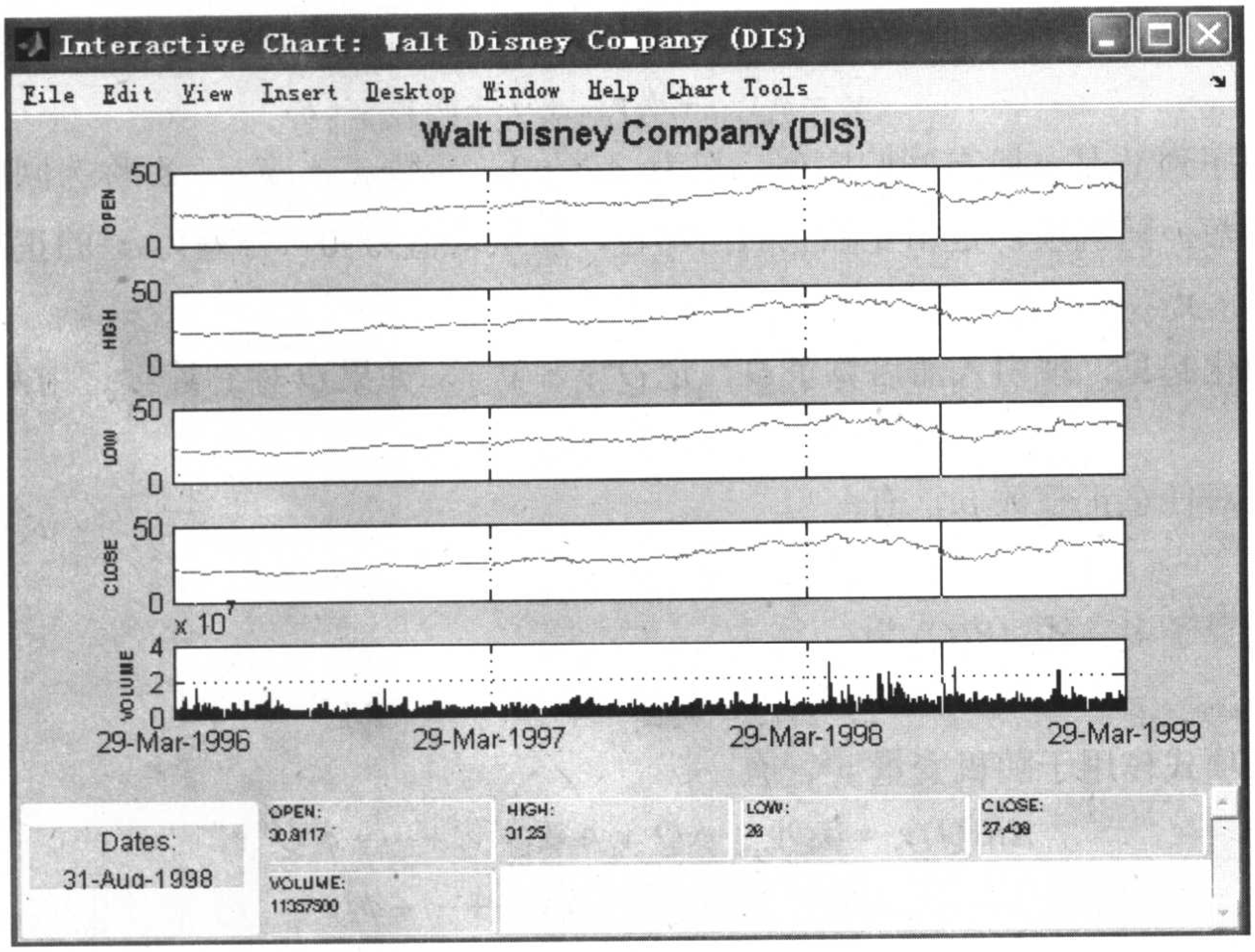


图 3.12　迪斯尼股价互动图

互动图中有一条可以任意移动的竖线，互动图的下方有 6 个格子，分别是 Dates(日期)、OPEN(开盘价)、HIGH(最高价)、LOW(最低价)、CLOSE(收盘价)和 VOLUME(成交量)。当移动竖线时下面的 6 个格子里的数据会跟着变化。图中可以读出竖线停留在 1998 年 8 月 31 日时，当天的开盘价、最高价、最低价、收盘价和成交量分别为 30.8117、31.25、26、27.438 和 11357500。

3.3 时间序列模型

时间序列分析的研究对象是一系列随时间变化而又相互关联的动态数据。博克斯(George Box)和詹金斯(Gwilym Jenkins)对时间序列的研究有独特贡献，1970 年他们合著的《时间序列分析：预测与控制》是这方面的权威著作。

时间序列模型有 3 种基本类型：自回归(AR，Auto-Regressive)模型、移动平均(MA，Moving-Average)模型以及自回归移动平均(ARMA，Auto-Regressive Moving-Average)模型。

3.3.1 时间序列模型介绍

1. 自回归过程 AR 模型

如果时间序列 y_t 是它的前期值和随机项的线性函数，即可以表示为如下形式：

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \tag{3.1}$$

则称该时间序列 y_t 是 p 阶自回归序列，记为 $AR(p)$。参数 $\phi_1, \phi_2, \phi_3, \cdots, \phi_p$ 称为回归系数，是模型待估参数。随机项 ε_t 是相互独立的白噪音，服从均值为 0，方差为 σ_t^2 的正态分布，白噪音 ε_t 与 $y_{t-1}, y_{t-2}, \cdots, y_{t-p}$ 相互独立。

为了简化起见，现引入滞后算子 Q，记 $Q^k y_t = y_{t-k}$，如果 Q 对 y_t 连续运用两次，就有

$$Q^2 y_t = Q(Qy_t) = Q(y_{t-1}) = y_{t-2}$$

类似地对任意正整数 p，有

$$Q^p y_t = y_{t-p}$$

如果记算子多项式 $AR(Q)$ 为

$$AR(Q) = \phi_1 Q + \phi_2 Q^2 + \phi_3 Q^3 + \cdots + \phi_p Q^p$$

算子多项式作用于随机变量 y_t，有

$$\begin{aligned} AR(Q) y_t &= \phi_1 Q y_t + \phi_2 Q^2 y_t + \phi_3 Q^3 y_t + \cdots + \phi_p Q^p y_t \\ &= \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \cdots + \phi_p y_{t-p} \end{aligned}$$

这样式(3.1)可以写为

$$y_t = \phi_1 Q y_t + \phi_2 Q^2 y_t + \phi_3 Q^3 y_t + \cdots + \phi_p Q^p y_t = AR(Q) y_t \tag{3.2}$$

进一步地，模型可以简写为

$$y_t = AR(Q) y_t + \varepsilon_t \tag{3.3}$$

$AR(Q)$ 的形式是多项式，$AR(Q)$ 的过程平稳性条件是多项式 $AR(Q)$ 的根均在单位圆外，即多项式 $AR(Q)=0$ 的根大于 1。

2. 移动平均过程 MA 模型

如果时间序列 y_t 是随机项的线性组合，即可以表示为如下形式：

$$y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{3.4}$$

则称时间序列 y_t 是 q 阶移动平均序列，记为 $MA(Q)$，实参数 $\theta_1,\theta_2,\theta_3,\cdots,\theta_q$ 为移动平均系数，也是模型的待估参数，同样引入滞后算子，并令

$$MA(Q) = 1 - \theta_1 Q - \theta_2 Q^2 - \cdots - \theta_q Q^q$$

则式(3.4)可以写成

$$y_t = MA(Q)\varepsilon_t \tag{3.5}$$

3. 自回归移动平均过程 ARMA 模型

如果时间序列 y_t 是它的随机项和前期的线性函数，可以写成如下形式：

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \cdots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \tag{3.6}$$

则称该时间序列 y_t 是自回归移动平均时间序列，式(3.6)为(p,q)阶的自回归移动平均模型，记为 $ARMA(p,q)$，参数 $\phi_1,\phi_2,\phi_3,\cdots,\phi_p$ 称为回归系数，参数 $\theta_1,\theta_2,\theta_3,\cdots,\theta_q$ 为移动平均系数，显然 $AR(p)$ 和 $MA(q)$ 都是 $ARMA(p,q)$ 的特殊情况。如果 $ARMA(p,q)$ 模型中阶数 $q=0$，则变成 $AR(Q)$，如果 $p=0$ 则变成 $MA(Q)$，引入滞后时间算子，式(3.6)可以写为

$$y_t = AR(Q) y_t + MA(Q)\varepsilon_t$$

4. 多变量的时间序列模型

MATLAB 的时间序列模型非常多，在 ARMA 模型、AR 模型基础之上又扩展了许多新的模型，如多变量的 ARMAX 和 ARX。ARMAX 模型与 AR、MA、ARMA 模型的区别在于引入了自变量，使得可以处理自变量与因变量之间的关系。MATLAB 中的时间序列模型如下：

$$ARX: A(Q)y_t = B(Q)x_{t-nk} + \varepsilon_t$$

$$ARMAX: A(Q)y_t = B(Q)x_{t-nk} + C(Q)\varepsilon_t$$

$$OE: F(Q)y_t = B(Q)x_{t-nk} + F(Q)\varepsilon_t$$

$$BJ: A(Q)y_t = \frac{B(Q)}{F(Q)}x_{t-nk} + \frac{C(Q)}{D(Q)}\varepsilon_t$$

$$PEM: A(Q)y_t = \frac{B_1(Q)}{F_1(Q)}y_{t-nk_1} + \frac{B_2(Q)}{F_2(Q)}y_{t-nk_2} + \ldots + \frac{B_{nu}(Q)}{F_{nu}(Q)}y_{t-nk_{nu}} + \frac{C(Q)}{D(Q)}\varepsilon_t$$

其中，$A(Q)$、$B(Q)$、$C(Q)$、$D(Q)$ 和 $F(Q)$ 都是含有延迟因子的多项式。

从上面可以看出，在 ARMAX 中，$B(Q)=0$ 时，ARMAX 变为 ARMA 模型，即

$$A(Q)y_t = C(Q)\mathrm{e}_t$$

5. 评价时间序列模型的 FPE 准则、AIC 准则、BIC 准则

最终预报误差的定阶准则简称为 FPE 准则(Final Prediction Error)，是 1971 年由 Akaike 提出的，主要用于 AR 模型的定阶。FPE 准则是以 AR 模型的一步误差达到最小的相应的阶作为 AR 模型的阶，用其预报效果的优劣来确定 AR 模型的阶数。

用 AR 模型定阶的步骤如下。

第 1 步：任意选取正整数 $P \in \left[\frac{N}{10}, \frac{N}{5}\right]$，其中 N 是样本的个数，P 作为 AR 模型最大的阶。

第 2 步：依次计算 FPE_h，其中 $h = 1,2,3,\cdots,P$，使得 FPE 最大的正整数 $\hat{P}$ 为 AR 模型的阶。

对于时间序列模型，AIC 与 BIC 也是判别时间序列模型优劣的标准，MATLAB 中 AIC 与 BIC 的计算方法如下。

$$AIC = (-2 \times LLF) + 2 \times NumParams$$

$$BIC = (-2 \times LLF) + NumParams \times \log(NumObs)$$

其中 LLF 为极大似然比，$NumParams$ 为待估参数的个数，$NumObs$ 为样本数。一般而言，AIC 与 BIC 的值越小说明模型越好。

3.3.2 时间序列模型估计

1. AR 模型的调用

AR 模型形式如下：

$$A(Q)y_t = \varepsilon_t$$

其中，$A(Q)$ 为延迟多项式，MATLAB 中对上述模型的估计方法是最小二乘方法。

调用方式

```
m = ar(y,n)
[m ,refl] = ar(Y,N,Approach,Window)
```

输入参数

```
Y              %观察值
N              %AR 模型的阶数
Approach       %标示计算模型参数的方法
```

Approach 的取值如表 3.1 所示。

表 3.1　Approach 的取值内容

方　法	Forward-Backward 方法	最小二乘法	Yule-Walker 方法	Burg's Lattic-Based 方法	Geometric Lattic 方法
Approach 值	fb	ls	yw	burg	gl

```
Window          %处理 Y 中缺失值的方法
Window= "now"   %表示观察值中没有缺失值
Window= "yw"    %表示 Yule-Walker 方法处理缺失值
```

输出参数

```
m              %AR 模型的文字形式
ref            %AR 模型的系数
```

【例 3-2】给出深发展 2005 年 10 月 21 日至 2006 年 9 月 29 日的交易日收盘价收益率，收益率保存在变量 y 中，用 2 阶的 AR 模型进行估计，代码如下：

```
>>n=2;                    %确定 AR 模型的阶数为 2 阶
>> q=ar(y,2);
Discrete-time IDPOLY model: A(q)y(t) = e(t)
A(q) = 1 - 0.05464 q^-1 - 0.01041 q^-2
Estimated using AR ('fb'/'now') from data set y
Loss function 0.000576822 and FPE 0.000587809
Sampling interval: 1
```

从上面的结果可以知道，2 阶的 AR 模型可以写成如下形式：

$$AR(2)y_t = y_t - 0.05464y_{t-1} - 0.01041y_{t-2} = \varepsilon_t$$

也即

$$y_t = 0.05464y_{t-1} + 0.01041y_{t-2} - \varepsilon_t$$

模型中参数的估计采用了默认的“Forward-Backward”方法，上述模型的损失函数为 0.000576822，FPE 准则的值为 0.000587809。下面确定 AR 模型的滞后阶数，我们采用偏相

关系数进行判断，首先我们计算样本的偏相关系数，代码如下：

```
>> parcorr(y)
```

显示的偏相关系数如图 3.13 所示。

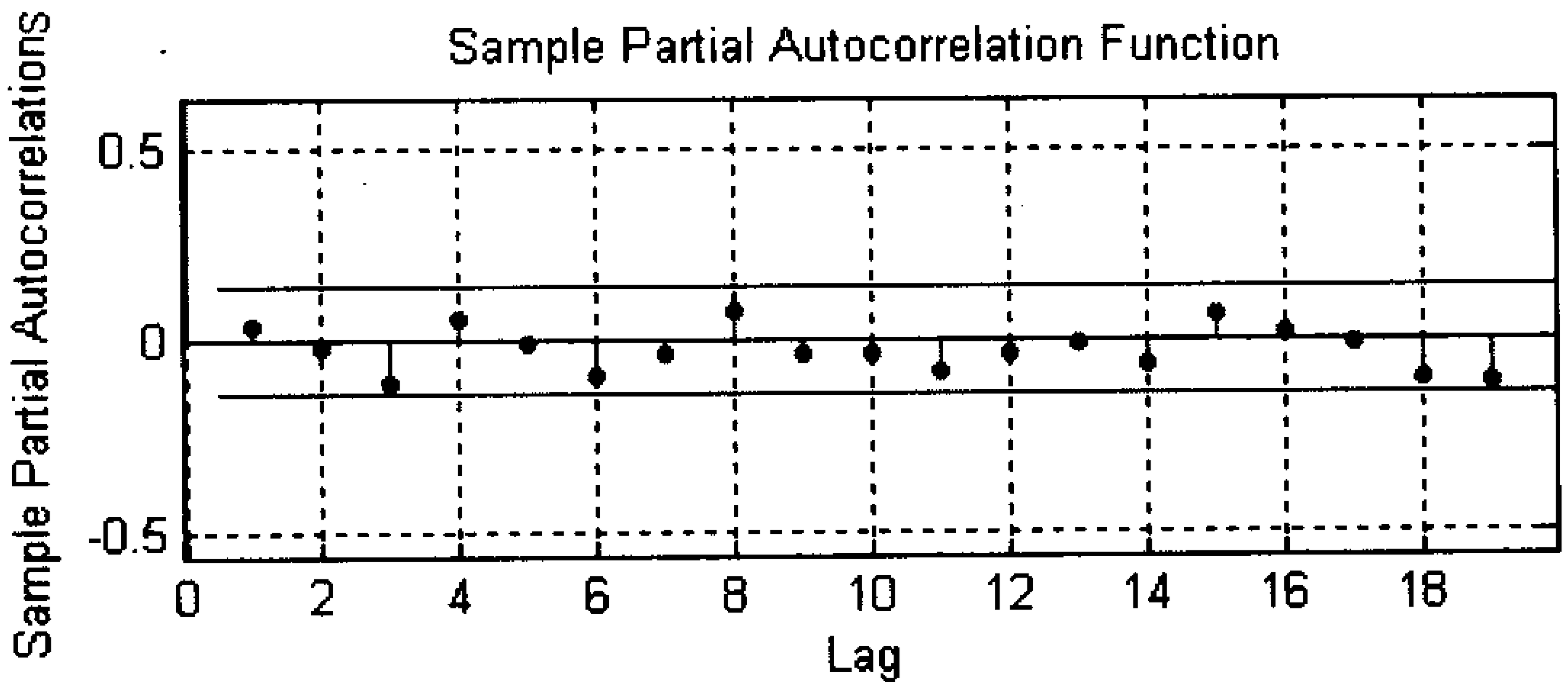


图 3.13　深发展股票收益率的偏相关图

从图 3.13 可以看出，偏相关系数都落在置信区间内，说明 AR 模型可能不适合描述深发展的收益率。

【例 3-2】给出上证指数从 2005 年 10 月 21 日至 2006 年 9 月 29 日的日收盘价的收益率(保存在 MATLAB 中的变量 y 中)，考虑使用 MA 时间序列模型进行拟合。

第 1 步：计算时间序列的自相关系数 ACF，确定 MA 模型的滞后阶数，代码如下：

```
>> autocorr(y)
```

显示的自相关系数如图 3.14 所示。

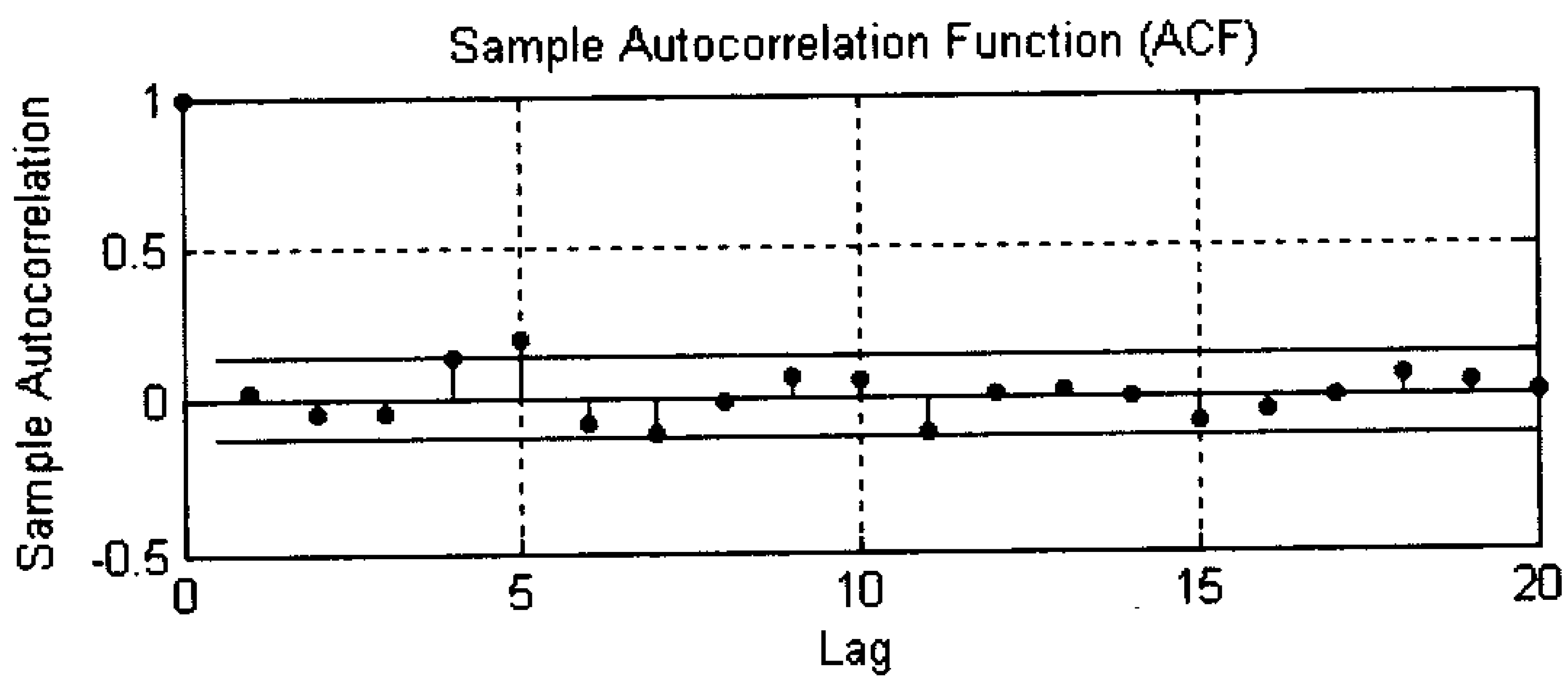


图 3.14　上证指数收益率的自相关图

可以看出 5 阶偏相关系数落在置信区间外，所以考虑用 5 阶的 MA 模型。

第 2 步：给出阶数为 5 的 MA 模型的形式。

注意到 ARMAX 的模型形式如下：

$$A(Q)y_t = B(Q)x_{t-nk} + C(Q)\varepsilon_t$$

令 $A(Q)=1$，$B(Q)=0$，可以得到如下 MA 模型：

$$y_t=\varepsilon_t-\theta_1\varepsilon_{t-1}-\theta_2\varepsilon_{t-2}-\cdots-\theta_q\varepsilon_{t-q}$$

```
>> z=iddata(y)  %将上证指数序列 y 转为 ARMAX 函数可以识别的参数
Time domain data set with 213 samples
Sampling interval: 1
Outputs      Unit (if specified)
   y1
>> armax(z,'nc',5)  %ARMAX 模型中只需输入 C(Q) 的阶数即可
Discrete-time IDPOLY model: y(t) = C(q)e(t)
C(q) = 1 + 0.01409 q^-1 + 0.05519 q^-2 - 0.004185 q^-3 + 0.09333 q^-4
       + 0.1789 q^-5
Estimated using ARMAX from data set z
Loss function 0.000145294 and FPE 0.000152486
Sampling interval: 1
```

得到 MA(5)的形式如下：

$$\begin{aligned}y_t&=C(Q)\varepsilon_t=(1+0.01409Q+0.05519Q^2-0.004185Q^3+0.09333Q^4+0.1789Q^5)\varepsilon_t\\&=\varepsilon_t+0.01409\varepsilon_{t-1}+0.05519\varepsilon_{t-2}-0.004185\varepsilon_{t-3}+0.09333\varepsilon_{t-4}+0.1789\varepsilon_{t-5}\end{aligned}$$

【例 3-3】估计 ARMA 模型，我们仍然用上一个例子中的数据。

ARMAX 模型形式如下：

$$A(Q)y_t=B(Q)x_{t-nk}+C(Q)\varepsilon_t$$

上述模型中，给出 $A(Q)$、$C(Q)$ 的阶数即可变成 ARMA 模型，即

$$A(Q)y_t=C(Q)\varepsilon_t$$

假设 ARMA 模型的阶数为 $p=2$，$q=2$，在 Command 窗口中执行以下命令：

```
>> data=iddata(y)  %将上证指数收益率序列转换为 ARMAX 函数可以识别的数据
Time domain data set with 213 samples
Sampling interval: 1
Outputs      Unit (if specified)
   y1
>> armax(data,'na',2,'nc',2)   %调用 armax 函数估计参数
Discrete-time IDPOLY model: A(q)y(t) = C(q)e(t)
A(q) = 1 + 0.6458 q^-1 + 0.5283 q^-2
C(q) = 1 + 0.69 q^-1 + 0.6164 q^-2
Estimated using ARMAX from data set data
Loss function 0.00015252 and FPE 0.000158501
Sampling interval: 1
```

从上面的结果可以看出，滞后多项式 $A(Q)$、$B(Q)$ 的形式如下：

$$A(Q)y_t=y_t+0.6458y_{t-1}+0.5283y_{t-1}$$

$$B(Q)\varepsilon_t=\varepsilon_t+0.69\varepsilon_{t-1}+0.6164\varepsilon_{t-2}$$

ARMA 的模型如下：

$$y_t = -0.6458y_{t-1} - 0.5283y_{t-1} + \varepsilon_t + 0.69\varepsilon_{t-1} + 0.6164\varepsilon_{t-2}$$

ARMA 模型的损失函数值为 0.00015252，FPE 准则的值为 0.000158501。

下面我们考虑时间序列模型的相互转化。

2. 将有限阶的 ARMA 模型转化为无限阶的自回归 AR 模型

理论上 ARMA 模型可以转化为 AR 模型，ARMA 模型的形式如下：

$$y_t = \sum_{i=1}^{R}\phi_i y_{t-i} + \varepsilon_t + \sum_{j=1}^{M}\theta_j\varepsilon_{t-j} \tag{3.7}$$

实际上 ARMA 模型可以写成如下形式：

$$y_t = \sum_{i=1}^{\infty}\pi_i y_{t-i} + \varepsilon_t \tag{3.8}$$

式(3.8)右边虽然有无穷项，但实际上可以根据需要选取一个上限。

调用方式

```
InfiniteAR = garchar(AR,MA,NumLags)
```

输入参数

```
AR              %AR 部分的阶数
MA              %MA 部分的阶数
NumLags         %截取的阶数
```

输出参数

```
InfiniteAR      %与 ARMA 模型等价的 AR 模型
```

【例 3-4】我们给出模拟的 ARMA 模型如下：

$$y_t = 0.5y_{t-1} - 0.8y_{t-2} + \varepsilon_t - 0.6\varepsilon_{t-1} + 0.08\varepsilon_{t-2} \tag{3.9}$$

要求将上述 ARMA 模型转换为 AR(∞) 模型，要求取到第 20 阶近似。

在 Command 窗口中执行以下命令：

```
>>PI = garchar([0.5 -0.8], [-0.6 0.08], 20);
>> PI'
ans =
   -0.1000
   -0.7800
   -0.4600
   -0.2136
   -0.0914
   -0.0377
   -0.0153
   -0.0062
```

```
-0.0025
-0.0010
-0.0004
-0.0002
-0.0001
-0.0000
-0.0000
-0.0000
-0.0000
-0.0000
-0.0000
-0.0000
```

PI 就是 AR(∞) 模型中前 20 项的系数。

3.3.3 ARX 与 ARMAX 模型的估计

ARX 模型就在 AR 模型基础上加入了外部解释变量 X。

1. ARMAX 模型的估计

估计 ARMAX 模型的代码如下：

```
m = armax(data,orders)
m = armax(data,'na',na,'nb',nb,'nc',nc,'nk',nk)
m = armax(data,orders,'Property1',Value1,…,'PropertyN',ValueN)
```

MATLAB 中 armax 函数的主要格式如下：

$$A(q)y(t)=B(q)u(t-nk)+C(q)e(t)$$

参数 na、nb、nc 的不同之处在于：

$$\begin{array}{ll} na & A(q)=1+a_1q^{-1}+a_2q^{-2}+,\cdots,+a_{na}q^{-na} \\ nb & B(q)=b_1+b_2q^{-1}+b_3q^{-2}+,\cdots,+b_{nb}q^{-nb+1} \\ nc & C(q)=1+c_1q^{-1}+c_2q^{-2}+,\cdots,+c_{nc}q^{-nc} \end{array}$$

其中

$$A(q)y(t)=y(t)+a_1y(t-1)+a_2y(t-2)+,\cdots,+a_{na}y(t-na)$$
$$B(q)u(t-nk)=b_1u(t-nk)+b_2u(t-nk-1)+,\cdots,+b_{nb}u(t-nb-nk+1)$$
$$C(q)\mathrm{e}(t)=\mathrm{e}(t)+c_1\mathrm{e}(t-1)+c_2\mathrm{e}(t-2)+,\cdots,+c_{nc}+e(t-nc)$$

【例 3-5】估计 ARMAX 模型，数据是深发展收益率(000001)与上证指数收益率，我们选用的时间段为 2005 年 10 月 21 日到 2006 年 9 月 29 日，深发展的收益率保存在变量 y 中，上证指数的收益率保存在变量 u 中，收益率为算术收益率$\left(t\text{时刻收益率}=\dfrac{P_t}{P_{t-1}}-1\right)$。我们采用

ARMAX 模型进行估计，在 MATLAB 中执行如下命令：

```
>>  z=iddata(y,u)
Time domain data set with 213 samples.
Sampling interval: 1
Outputs      Unit (if specified)
   y1
Inputs       Unit (if specified)
   u1
>> z=iddata(y,u)
>> m=armax(z,[2,2,2,1])
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + C(q)e(t)
A(q) = 1 - 1.273 q^-1 + 0.9398 q^-2

B(q) = -0.305 q^-1 + 0.2697 q^-2

C(q) = 1 - 1.184 q^-1 + 0.8833 q^-2

Estimated using ARMAX from data set z
Loss function 0.000590723 and FPE 0.000626166
Sampling interval: 1
```

从上面的分析结果可以看出，模型的形式如下：

$$A(q)y(t) = y_t - 1.273y_{t-1} + 0.9398y_{t-2}$$
$$B(q)u(t) = -0.305u_{t-1} + 0.2697u_{t-2}$$
$$C(q)e(t) = e_t - 1.184e_{t-1} + 0.8833e_{t-2}$$

根据 $A(q)y(t) = B(q)u(t) + C(q)e(t)$ 可以得出

$$y_t - 1.273y_{t-1} + 0.9398y_{t-2} = -0.3051u_{t-1} + 0.2697u_{t-2} + e_t - 1.184e_{t-1} + 0.8833e_{t-2}$$

简化得

$$y_t = 1.273y_{t-1} - 0.9398y_{t-2} - 0.305u_{t-1} + 0.2697u_{t-2} + e_t - 1.184e_{t-1} + 0.8833e_{t-2}$$

这样就给出了深发展时间序列与上证指数的关系，损失函数值为 0.000590723，FPE 准则的值为 0.000626166。

接下来我们估计 ARMA 模型，在 MATLAB 中执行如下命令：

```
>> na=2;nc=3;
>> orders = [na nc]
orders =
     2     3
>> z=iddata(y)
>> m = armax(z,orders)
```

```
Discrete-time IDPOLY model: A(q)y(t) = C(q)e(t)
A(q) = 1 + 0.1337 q^-1 + 0.9058 q^-2
C(q) = 1 + 0.1239 q^-1 + 0.9062 q^-2 - 0.06353 q^-3
Estimated using ARMAX from data set z
Loss function 0.000611174 and FPE 0.00064143
Sampling interval: 1
```

上述模型等价于

$$y_t + 0.1337y_{t-1} + 0.9058y_{t-2} = e_t + 0.1239e_{t-1} + 0.9062e_{t-2} - 0.06353e_{t-3}$$

整理得到 ARMA 模型形式如下：

$$y_t = -0.1337y_{t-1} - 0.9058y_{t-2} + e_t + 0.1239e_{t-1} + 0.9062e_{t-2} - 0.06353e_{t-3}$$

损失函数值为 0.000611174，FPE 准则的值为 0.00064143。

2. ARX 模型的估计

ARX 模型具有如下形式：

$$A(Q)y_t = B(Q)u(t-nk) + \varepsilon_t$$

其中，$A(Q)$、$B(Q)$ 都是滞后算子多项式。

MATLAB 中的 arx 函数可以对 ARX 模型进行估计，下面介绍 arx 函数的使用方法。

调用方式

```
m = arx(data,orders)
m = arx(data,'na',na,'nb',nb,'nk',nk)
m= arx(data,orders,'Property1',Value1,…,'PropertyN',ValueN)
```

输入参数

```
data        %观察样本值
orders      %确定 ARX 的滞后多项式的阶数
na          %ARX 模型中滞后多项式 A(Q) 的阶数
nb          %ARX 模型中滞后多项式 B(Q) 的阶数
nk          %ARX 模型中自变量的滞后阶数
```

输出参数

```
m      %ARX 模型的特征参数
```

【例 3-6】研究深发展收益率(000001)与上证指数收益率之间的关系，我们选用的时间段为 2005 年 10 月 21 日到 2006 年 9 月 29 日，深发展的收益率保存在变量 y 中，上证指数的收益率保存在变量 u 中，收益率为算术收益率$\left(t\text{时刻收益率}=\dfrac{P_t}{P_{t-1}}-1\right)$。我们采用 ARX 模型进行估计，在 MATLAB 中执行如下命令：

```
>> z=iddata(y,u)
Time domain data set with 213 samples.
```

```
Sampling interval: 1
Outputs      Unit (if specified)
   y1
Inputs       Unit (if specified)
   u1
>> m=arx(z,'na',2,'nb',3)
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)
A(q) = 1 - 0.1118 q^-1 - 0.09373 q^-2
B(q) = -0.2222 q^-1 - 0.2341 q^-2 - 0.07306 q^-3
Estimated using ARX from data set z
Loss function 0.000557467 and FPE 0.000584268
Sampling interval: 1
```

从上面的结果可以看出，滞后多项式 $A(Q)$、$B(Q)$ 如下：

$$A(Q)y_t = y_t - 0.1118y_{t-1} - 0.09373y_{t-2}$$

$$B(Q)u_t = -0.2222u_{t-1} - 0.2341u_{t-2} - 0.07306u_{t-3}$$

ARX 模型的形式如下：

$$y_t = 0.1118y_{t-1} + 0.09373y_{t-2} - 0.2222u_{t-1} - 0.234u_{t-2} - 0.07306u_{t-3} + \varepsilon_t$$

3. 广义线性模型 PEM

广义线性模型 PEM 的形式如下：

$$A(Q)y_t = \frac{B_1(Q)}{F_1(Q)}y_{t-nk_1} + \frac{B_2(Q)}{F_2(Q)}y_{t-nk_2} + \cdots + \frac{B_{nu}(Q)}{F_{nu}(Q)}y_{t-nk_{nu}} + \frac{C(Q)}{D(Q)}\varepsilon_t$$

其中，$A(Q)$，$B_i(Q)$，$C(Q)$，$D(Q)$ 和 $F_i(Q)$ 为滞后多项式。

MATLAB 中的 pem 函数调用方式如下：

调用方式

```
m = pem(data,'na',na,'nb',nb,'nc',nc,'nd',nd,'nf',nf,'nk',nk)
```

输入参数

```
data    %iddata 型时间序列数据，需要将观察值转换为 iddata 型数据
```

【例 3-7】 深发展、上证指数从 2005 年 10 月 21 日到 2006 年 9 月 29 日的日收益率分别保存在变量 y,$u1$ 中，然后以 $u1$ 为基础生成 $u2$、$u3$ 变量，估计 PEM 模型。

在 MATLAB 中执行如下命令：

```
>> u2=u1+0.001*rand(213,1);u3=u1+0.001*rand(213,1); %以 u1 为基础生成 u2、u3
变量
>> z=iddata(y,[u1,u2,u3]) %将观察值用 iddata 函数转换为 pem 函数可以识别的 iddata
型数据
Time domain data set with 213 samples.
Sampling interval: 1
```

```
Outputs      Unit (if specified)
  y1
Inputs       Unit (if specified)
  u1
  u2
  u3
```

调用 pem 函数估计 PEM 模型的代码如下：

```
>> ma=pem(z,'na',2,'nb',[2 2 1],'nc',2,'nd',2,'nf',[2 2 1])
Discrete-time IDPOLY model: A(q)y(t) = [B(q)/F(q)]u(t) + [C(q)/D(q)]e(t)
A(q) = 1 + 0.3977 q^-1 + 0.9468 q^-2
B1(q) = -4.231 q^-1 - 2.662 q^-2
B2(q) = 0.6485 q^-1 + 4.67 q^-2
B3(q) = 3.281 q^-1
C(q) = 1 + 0.4057 q^-1 + 0.9706 q^-2
D(q) = 1 - 0.08875 q^-1 - 0.142 q^-2
F1(q) = 1 - 0.4959 q^-1 + 0.1912 q^-2
F2(q) = 1 - 0.2297 q^-1 + 0.1249 q^-2
F3(q) = 1 + 0.1269 q^-1
Estimated using PEM from data set z
Loss function 0.00055632 and FPE 0.000656333
Sampling interval: 1  D
```

从上面可以得到 PEM 模型中各滞后多项式具有如下形式：

$$A(Q)=1+0.3977Q+0.9468Q^2$$
$$B_1(Q)=-4.231Q-2.662Q^2$$
$$B_2(Q)=0.6485Q+4.67Q^2$$
$$B_3(Q)=3.281Q$$
$$C(Q)=1+0.4057Q+0.9706Q^2$$
$$D(Q)=1-0.08875Q-0.142Q^2$$
$$F_1(Q)=1-0.4959Q+0.1912Q^2$$
$$F_2(Q)=1-0.2297Q+0.1249Q^2$$
$$F_3(Q)=1+0.1269Q$$

将上面的滞后多项式依次代入 PEM 模型得

$$A(Q)y_t=\frac{B_1(Q)}{F_1(Q)}u_{1,t}+\frac{B_2(Q)}{F_2(Q)}u_{2,t}+\frac{B_3(Q)}{F_3(Q)}u_{3,t}+\frac{C(Q)}{D(Q)}\varepsilon_t$$

4. Box-Jenkins 模型参数估计

Box-Jenkins 模型具有如下形式：

$$y_t = \frac{B(Q)}{F(Q)}u_{t-nk} + \frac{C(Q)}{D(Q)}\varepsilon_t$$

MATLAB 中用 bj 函数估计 Box-Jenkins 模型，其调用方式如下。

调用方式

```
m = bj(data,'nb',nb,'nc',nc,'nd',nd,'nf',nf,'nk',nk)
```

输入参数

```
data  %iddata 型时间序列数据，需要将观察值转换为 iddata 型数据
'nb','nf','nc','nd','nk'    %Box-Jenkins 模型中各滞后多项式的阶数
```

【例 3-8】已知深发展、上证指数从 2005 年 10 月 21 日到 2006 年 9 月 29 日的日收益率、收益率分别保存在变量 y、u 中，用 Box-Jenkins 模型估计深发展与上证指数之间的关系。

```
>> z=iddata(y,u)
Time domain data set with 213 samples.
Sampling interval: 1
Outputs      Unit (if specified)
   y1
Inputs       Unit (if specified)
   u1
>> m=bj(z,'nb',2,'nf',2,'nc',1,'nd',2)
Discrete-time IDPOLY model: y(t) = [B(q)/F(q)]u(t) + [C(q)/D(q)]e(t)
B(q) = -0.175 q^-1 + 0.2317 q^-2
C(q) = 1 - 0.9276 q^-1
D(q) = 1 - 1.013 q^-1 + 0.1258 q^-2
F(q) = 1 - 1.294 q^-1 + 0.9776 q^-2
Estimated using BJ from data set z
Loss function 0.000597513 and FPE 0.000639975
Sampling interval: 1
```

将上述结果代入 Box-Jenkins 模型即可。

3.4 GARCH 模型参数估计

3.4.1 GARCH 模型介绍

GARCH 表示广义自回归条件异方差(Generalized Auto Regressive Conditional Heteroscedasticity)。GARCH 模型分为均值方程与方差方程两部分。

均值方程形式如下：

$$y_t = C + \sum_{i=1}^{R} \phi_i y_{t-i} + \varepsilon_t + \sum_{j=1}^{M} \theta_j \varepsilon_{t-j} + \sum_{k=1}^{N_x} \beta_k X(t,k)$$

其中，ϕ_i 为自回归系数，θ_j 为移动平均系数，ε_t 为新信息，y_t 表示回报，$X(t,k)$ 为 y_t 的解释变量矩阵。

GARCH(P,Q)模型方差方程形式如下：

$$\sigma_t^2 = K + \sum_{i=1}^{P} G_i \sigma_{t-i}^2 + \sum_{j=1}^{Q} A_j \varepsilon_{t-j}^2$$

其中系数满足下列约束条件：

$$\sum_{i=1}^{P} G_i + \sum_{j=1}^{Q} A_j < 1$$

$$k > 0, G_i > 0, A_j > 0，\quad i = 1,2,3,\cdots,P；\quad j = 1,2,3,\cdots,Q$$

GJR(P,Q)型方差方程形式如下：

$$\sigma_t^2 = k + \sum_{i=1}^{P} G_i \sigma_{t-1}^2 + \sum_{j=1}^{Q} A_j \varepsilon_{t-j}^2 + \sum_{j=1}^{Q} L_j S_{t-j}^- \varepsilon_{t-j}^2$$

其中，$S_{t-j}^- = \begin{cases} 1 & \varepsilon_{t-j} < 0 \\ 0 & \varepsilon_{t-j} \geqslant 0 \end{cases}$

以及

$$\sum_{i=1}^{P} G_i + \sum_{j=1}^{Q} A_j + \frac{1}{2} \sum_{j=1}^{Q} L_j < 1$$

$$k > 0, G_i \geqslant 0, A_j \geqslant 0, A_j + L_j \geqslant 0；\quad i = 1,2,3,\cdots,P；\quad j = 1,2,3,\cdots,Q$$

EGARCH 型方差方程形式如下：

$$\log \sigma_t^2 = k + \sum_{i=1}^{P} G_i \log \sigma_{t-i}^2 + \sum_{j=1}^{Q} A_j \left[\frac{|\varepsilon_{t-j}|}{\sigma_{t-j}} - E\left\{ \frac{|\varepsilon_{t-j}|}{\sigma_{t-j}} \right\} + \sum_{j=1}^{Q} L_j \left(\frac{\varepsilon_{t-j}}{\sigma_{t-j}} \right) \right]$$

其中

$$E\left\{ \frac{|\varepsilon_{t-j}|}{\sigma_{t-j}} \right\} = \begin{cases} \sqrt{2/\pi} & \text{Gaussian型} \\ \sqrt{\dfrac{v-2}{\pi}} \dfrac{\Gamma\left(\dfrac{v-1}{2}\right)}{\Gamma\left(\dfrac{v}{2}\right)} & \text{Student' s型} \end{cases}$$

v 为自由度，要求 $v>2$。

3.4.2 GARCH(P,Q)模型参数估计

.1. GARCH 模型相关参数的设定

MATLAB 中设定 GARCH 模型参数的函数是 garchset，它可以把 GARCH 函数需要输入的参数规范化，便于估计。

调用方式

```
Spec = garchset('Parameter1',Value1,'Parameter2',Value2,…)
```

输入参数

```
Parameter1          %GARCH 模型中相关参数的名称，包括各个参数的内容
Value1              %对应参数的值
```

输出参数

```
Spec                %MATLAB 中 garchset 函数可以识别的输入格式
```

如需要建立 GARCH(1,1)模型，需要执行如下命令：

```
>> spec=garchset('p',1,'q',2)
spec =
          Comment: 'Mean: ARMAX(0,0,?); Variance: GARCH(1,2) '
     Distribution: 'Gaussian'
                C: []
    VarianceModel: 'GARCH'
                P:1
                Q:2
                K:[]
            GARCH:[]
             ARCH:[]
```

这样结构变量 spec 中保存了 GARCH 模型的相关信息，spec.GARCH 表示 GARCH 模型中 GARCH 项的系数，spec.ARCH 表示 ARCH 项的系数。

在 MATLAB 中执行如下命令：

```
>> spec.ARCH=[0.03 0.03]
spec =
          Comment: 'Mean: ARMAX(0,0,?); Variance: GARCH(1,2) '
     Distribution: 'Gaussian'
                C: []
    VarianceModel: 'GARCH'
                P: 1
```

```
            Q: 2
            K: []
        GARCH: 0.0200
         ARCH: [0.0300 0.0300]
```

【例 3-9】 已知 GARCH(1,1)的形式如下

$$y_t = \varepsilon_t$$

$$\sigma_t^2 = 0.0001 + 0.9\sigma_{t-1}^2 + 0.05\varepsilon_{t-1}^2$$

要求生成 MARLAB 中 GARCH 可以识别的格式，对比 GARCH(P,Q)型的一般式(3.2)可知，在 spec 函数中 $c = 0$，$k = 0.0001$，$garch = 0.9$，$arch = 0.05$。

在 MATLAB 中执行如下命令：

```
>> spec=garchset('c',0,'k',0.0001,'garch',0.9,'arch',0.05)
spec =
        Comment: 'Mean: ARMAX(0,0,?); Variance: GARCH(1,1) '
   Distribution: 'Gaussian'
              C: 0
  VarianceModel: 'GARCH'
              P: 1
              Q: 1
              K: 1.0000e-004
          GARCH: 0.9000
           ARCH: 0.0
```

从上面内容可以看出，GARCH 模型的结构保存在结构数组 spec 中，如果需要进一步观察 spec 中的内容，可以参照结构数组的显示方法。例如需要观察 ARCH 的参数可以执行如下命令：

```
>> spec.ARCH
ans =
    0.0500
```

MATLAB 中与 GARCH 模型相关的函数有 3 个，分别是 garchfit、garchred 和 garchsim。

2. 模拟生成单变量 GARCH(P,Q)型数据

模拟 GARCH(P,Q)首先需要确定模型的格式，如阶数 P,Q 的值、模拟的次数等，对于例 3-9 中的 GARCH 模型，模拟出 10 个样本值，代码如下：

```
>> spec=garchset('c',0,'k',0.0001,'garch',0.9,'arch',0.05);
>> y=garchsim(spec,10);        %spec 为 GARCH 的形式，1000 是模拟的次数，模拟结果保
                                 存在 y 中
y =
```

```
   0.0180
  -0.0299
   0.0215
  -0.0017
   0.0302
   0.0394
   0.0157
   0.0069
   0.0617
   0.0125
```

3. GARCH 模型的参数估计

MATLAB 中对时间序列用 GARCH 模型估计的函数是 garchfit，其调用方式如下。

调用方式

```
[Coeff,Errors,LLF] = garchfit(spec,Series)
```

输入参数

```
spec                %GARCH 模型的格式
Series              %时间序列观察值
```

输出参数

```
Coeff               %模型的参数信息，为结构数组，garchcount(coeff)可以返回参数的个数
Errors              %估计的误差，为结构数组
LLF                 %模型的极大似然比
```

【例 3-10】 首先调用模拟函数生成 GARCH(1,1)型数据，模型参数见例 3-9，然后进行估计，代码如下：

```
>> spec = garchset('C',0,'K',0.0001,'GARCH',0.9,'ARCH',0.05);
>> y = garchsim(spec,1000);
>> spec=garchset('p',1,'q',1);
>> garchfit(spec,y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   Diagnostic Information
Number of variables: 4
Functions
 Objective:                            garchllfn
 Gradient:                             finite-differencing
 Hessian:                              finite-differencing (or Quasi-Newton)
 Nonlinear constraints:                  armanlc
 Gradient of nonlinear constraints:    finite-differencing
Constraints
```

```
 Number of nonlinear inequality constraints:      0
 Number of nonlinear equality constraints:        0
 Number of linear inequality constraints:         1
 Number of linear equality constraints:           0
 Number of lower bound constraints:               4
 Number of upper bound constraints:               4
Algorithm selected
   medium-scale
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 End diagnostic information

                                max                    Directional   First-order
 Iter F-count          f(x)   constraint    Step-size   derivative
                   optimality Procedure
    0      5     -1627.24   -0.0002369
    1     25     -1627.27   -0.0002369    3.05e-005          -380     1.49e+003
    2     34     -1631.16   -0.0002221       0.0625          23.9     8.12e+004
    3     50     -1631.16   -0.0002232     0.000488          2.45     8.09e+003
    4     57     -1631.24   -0.0001674         0.25          2.31     9.55e+003
    5     70     -1631.24   -0.0001746      0.00391          1.72     1.03e+004
    6     79     -1631.31   -0.0002031       0.0625          1.15      1.7e+004
    7     86     -1631.35   -0.0002004         0.25        0.0747     2.45e+003
    8     91     -1631.36   -0.0001936            1      -0.00151           385
    9     96     -1631.36   -0.0001928            1     4.61e-006          23.9
   10    102     -1631.36   -0.0001928          0.5    -3.77e-007          11.7
Hessian modified
Optimization terminated: magnitude of directional derivative in search
 direction less than 2*options.TolFun and maximum constraint violation
  is less than options.TolCon.
No active inequalities.
  Mean: ARMAX(0,0,0); Variance: GARCH(1,1)
  Conditional Probability Distribution: Gaussian
  Number of Model Parameters Estimated: 4
                                  Standard          T
  Parameter        Value            Error          Statistic
 -----------    -----------    ------------     -----------
           C    -0.001049        0.0013893       -0.7551
           K     0.00019296      7.3878e-005       2.6119
    GARCH(1)      0.82523        0.048653         16.9617
     ARCH(1)      0.093188       0.025125          3.7090
   Log Likelihood Value: 1631.36
```

得到信息、条件标准差和回报如图 3.15 所示。

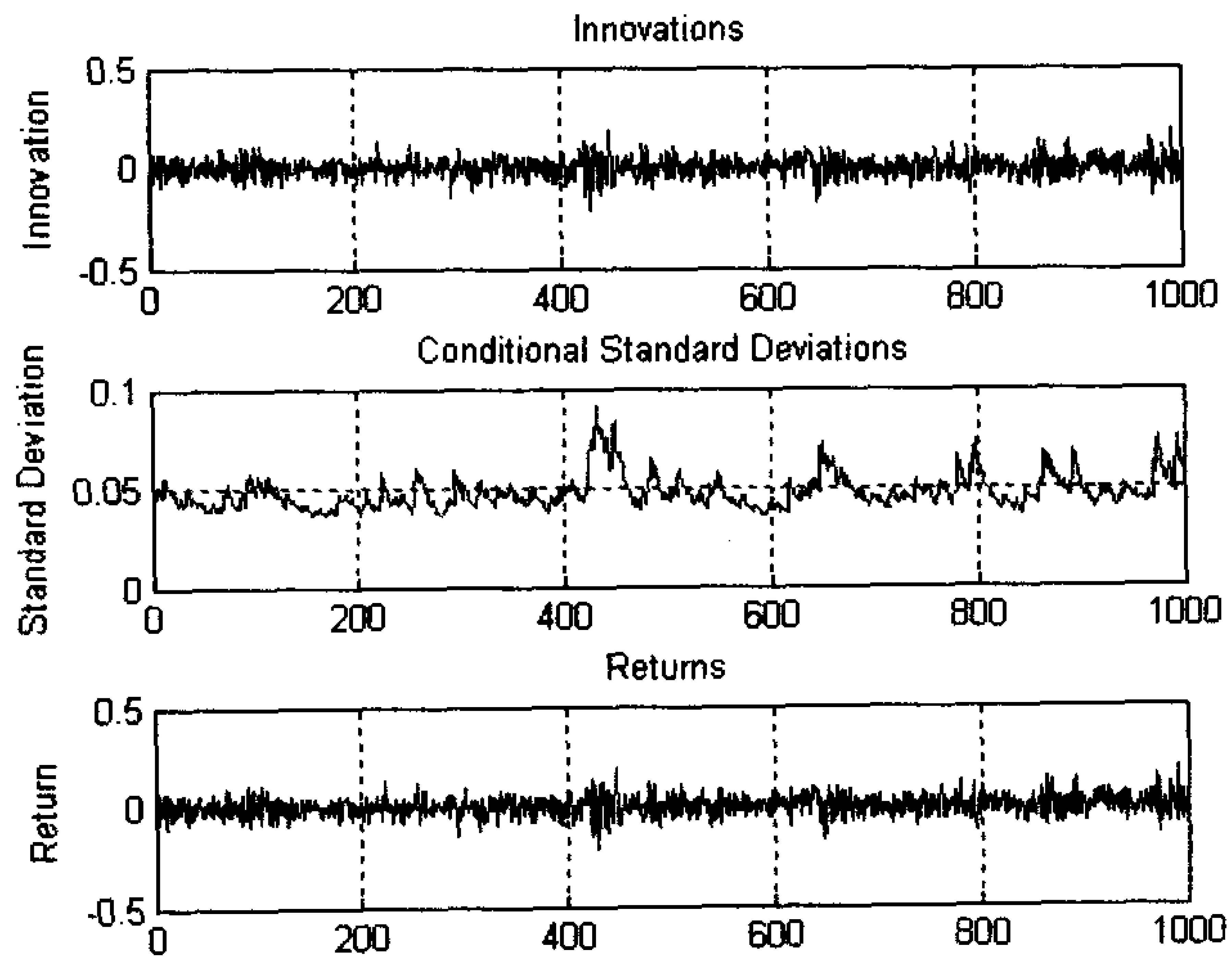


图 3.15　GARCH 模型结果示意图

分析的结果和预期相吻合。

【例 3-11】下面我们调用 MATLAB 自带的数据进行 GARCH 分析，代码如下：

```
>> load garchdata;
>> rate=price2ret(NYSE); %MATLAB 自带的纽约交易所的指数
>> spec=garchset('m',1,'p',1,'q',2,'display','off')
>> spec=garchset(spec,'r',2)
spec =
          Comment: 'Mean: ARMAX(2,1,?); Variance: GARCH(1,2) '
     Distribution: 'Gaussian'
                R: 2
                M: 1
                C: []
               AR: []
               MA: []
    VarianceModel: 'GARCH'
                P: 1
                Q: 2
                K: []
            GARCH: []
             ARCH: []
          Display: 'off'
>> garchfit(spec,rate)
Maximum Function Evaluations or Iterations Reached.
  Mean: ARMAX(2,1,0); Variance: GARCH(1,2)
  Conditional Probability Distribution: Gaussian
  Number of Model Parameters Estimated: 8
```

```
                              Standard            T
  Parameter        Value          Error          Statistic
 -----------    -----------   ------------    -----------
          C    7.9353e-005    5.0325e-005       1.5768
      AR(1)     0.96152       0.10017           9.5993
      AR(2)     -0.11766      0.019752         -5.9570
      MA(1)     -0.85642      0.099186         -8.6346
          K    8.8133e-007    1.5782e-007       5.5843
   GARCH(1)     0.90923        0.0073655      123.4450
    ARCH(1)     0.077516       0.015049         5.1508
    ARCH(2)     0.0042049      0.014952         0.2812

 Log Likelihood Value: 10435.1

>> [aic bic]= aicbic(LLF11,NumParams,155),%本例中样本观察值的个数 155
aic =
 -694.8347
bic =
 -682.6610
```

模型的 AIC 和 BIC 的值分别为-694.8347 和-682.6610。

估计的均值方程为

$$y_t = 7.9353e-005+0.96152y_{t-1} - 0.11766y_{t-2} + \varepsilon_t - 0.85642\varepsilon_{t-1}$$

方差方程为

$$\sigma_t^2 = 8.8133e-007 + 0.90923\sigma_{t-1}^2 + 0.077516\varepsilon_{t-1}^2 + 0.0042049\varepsilon_{t-2}^2$$

思 考 题

1. 有两种股票的收盘价如表 3.2 所示。

表 3.2 两种股票的收盘价

日 期	股 票 A	日 期	股 票 B
2007-1-5	1.98	2007-1-5	3
2007-1-6	2.2	2007-1-7	3.1
2007-1-9	2	2007-1-10	3.2
2007-1-11	2.1	2007-1-12	4

试编写程序将股票 A 与股票 B 的日期进行对齐，使得结果如表 3.3 所示。

表 3.3 对齐后的股价

日 期	股 票 A	股 票 B
2007-1-5	1.98	3
2007-1-6	2.2	**3**
2007-1-7	**2.2**	3.1
2007-1-9	2	3.1
2007-1-10	**2**	3.2
2007-1-11	2.1	**3.2**
2007-1-12	**2.1**	4

注：黑体字是增加的内容。

2. 从分析家中获取浦发银行(600000)2006 年 5 月 1 日至 2006 年 10 月 8 日的日收盘价数据到 MATLAB 中，将收盘价转换为收益率数据，并用 GRACH 进行拟合。

3. 计算浦发银行与上证指数从 2006 年 5 月 1 日至 10 月 8 日股价与收益率的偏相关系数。

4. 将深发展(000001)2006 年的日收盘价数据导入 Excel，然后导入 MATLAB 中，保存形式为 MATLAB 中的金融时间序列格式。

5. 用 GARCH 模型检验深发展(000001)2006 年度的收益率。

6. 股票价格经常受到一些消息的影响，例如福建的股票受到“三通”的影响而出现短期波动，界龙实业容易受到“迪斯尼”影响，B 股市场经常受到“A、B 股合并”传闻的影响，试分析如何持有该类股票，应如何处理，是否应该卖出。还有一些股票受到利空传闻的影响，如诉讼案件，受到监管部门的处罚，虽然这些消息对股票的基本面不会构成实质性利空，但是短期往往会下跌，如果遇到这样的情况是否应该卖出，试用统计结果进行实证，说明原因。

7. 从理论上讲，证券市场收益率主要来源于红利与资本利得，在我国证券市场上市公司连续多年回报股东的并不多，即使是公用事业类上市公司能够持续派发红利的也不多见，市场的投机气氛比较浓厚，价格的波动率相对较大，通常用收益率标准差代替市场波动率，比较我国与标准普尔 500 波动率之间的差距。

第 4 章　固定收益证券计算

固定收益证券是门类最多同时也是非常重要的金融产品。本章重点在于固定收益现金流和利率期限结构的计算。要求读者运用 MATLAB 固定收益工具箱计算现值、将来值、久期与凸度，掌握利用现有的债券品种推出利率期限结构，并且对新国债品种进行定价。

4.1　固定收益证券基本概念

固定收益证券是一组现金流稳定的证券，在证券家族中占有重要地位。固定收益证券有如下几个主要特征：偿还期、面值、票面利率等。债券是以借贷协议形式发行的证券，是固定收益证券的主要形式。息票债券指发行者在有效期内向持有人支付利息(通常每半年一次)的合约，又称为息票支付，之所以称为息票，是因为在将计算机引入金融实务之前，人们通常需要将债券息票剪下邮寄回发行方索取利息(Coupon Rate)。

有许多固定收益证券赋予发行者或投资者某些权力，这些特征看上去很简单，但每个特征都需要很多知识诠释，如果几个特征集中到一起，将使得固定收益证券定价与风险管理变得非常复杂。MATLAB 固定收益工具箱用于计算欧美市场的固定收益证券，我国的债券市场和欧美存在一些差别，读者利用金融工具箱分析固定收益时应注意产品间的区别。

4.1.1　美国的固定收益证券种类

美国的固定收益证券可以分为以下几种。

短期国库券(Treasury bills，又称 T-bills)：美国政府通过竞标发售的一种短期证券，价格相对面值有折扣，所以不会像大部分债券一样支付定额利息，其面值一般为 1～10 万美元，期限为 3 个月、6 个月或 9 个月不等。

政府票据(Treasury note，又称 T-note)：美国政府发行的中期票据，期限为 1～10 年，以息票方式支付利息。

长期国库券(Treasury bonds，又称 T-bonds)：美国政府发行的长期债券，期限为 10 年以上，面值为 1～10 万美元，以息票支付利息，报价以半年期债券为参照。2002 年美国取消了 30 年期国债的发行。

零息票债券(Zero-Coupon Bond)：买卖价格相对面值有重大折让的企业或市政债券，出现大额折让是由于债券并无任何利息，这类债券在发行时加入折扣，或由一家银行除去息

票，然后包装成为零息票债券发行，投资者在债券到期以面值赎回时可实现利润。

长期零息券一般是由附带息票票据创造出来的，购买息票国债的经纪人可以要求财政部停止债券的现金支付，使其成为独立证券序列，这时每一证券都具有获得原始债券收益要求权，例如一张 10 年期债券被“剥离”成 20 份半年期债券，每张都可视为零息票，它们的到期日从 6 个月到 10 年不等，最后本金支付是另一张零息证券，所有支付都单独计算，并配有自己的 CUSIP 号码(统一由美国证券鉴定程序委员会颁布)，具有这种标识符的证券都可以在联邦银行及其分支机构上进行电子交易，财政部仍旧具有支付责任。由于这些债券息票被“剥离”了，因此被称为本息剥离式国债(Separate Trading of Registered Interest and Principal of Securities，STRIPS)。

美国的 CD 存单(Certificate Deposit)是银行等金融机构向存款人发行的证券，可以买卖，属于货币市场工具，偿还期短于 1 年，一般按照“Actual/360”规则计算利息。CD 存单是按照面值发行的，利率是单利计算，利息计算公式为

$$\text{利息}=100\times c\times D/360$$

其中 c 是票面利率，D 为发行日至到期日之间的天数。

4.1.2 固定收益证券相关概念

固定收益证券交易有很多惯例，这些惯例涉及运作和清算诸多专业领域，这部分内容非常丰富，本节中我们来介绍一些基本规则。

交易日(Trade Date)：交易日就是买卖双方达成交易的日期。如果是通过拍卖方式购入证券，交易日就是拍卖结果被确认、购买者被告知他们分摊数量及价格的日期；如果固定收益证券由一承购集团成员所购买，交易日与这一集团牵头者最终将承销证券分配给成员的日期一致。

交割日(Exercise Date，又称结算日)：就是合同到期日，一般指买入方支付价格和卖出方交割证券的日期。美国国库券交割日为交易日之后的第一个营业日(交割术语为 T+1)。交割日也可以由交易者之间商定，如果交割日刚好支付利息，那么债券当天出售者获得当天利息支付，而债券购买者获得其余款项。有时通过 Fed Wine 机构交割证券，交易日即为交割日。

起息期间(或息票期间)：是上一次付息日与下一次付息日之间的天数。

交割日距离到期日的天数(Days from Settlement to Maturity，DSM)：一般规则是 DSM 包括交割日而不包括到期日，这样买方有动力尽早交易，获得当天收益，卖方在交割当天就获得资金使用权，因此到期日不能获得利息。

起息日到交割日天数(Days from Coupon to Settlement，DCS)：从计息日(含)到交割日(不含)之间的天数，付息日作为下一个利息期限的第一天而不计入 DCS。

到期日(Maturity)：是指固定收益证券债务合约终止的日期，到期日发行人应还清所有的本金与利息。很多固定收益证券如定期存款、短期国库券、商业票据、再回购协议、外汇掉期、零息票债券等只有一个到期日，日期计算都以这个到期日为基准，另外一些工具如附息票据、互换等，这些工具的到期日不止一个，对于长期债券来讲，到期日就包含利息支付日和债券赎回的最终到期日。

本金(Principal)和票面利率(Coupon Rate)：本金有时称为面值(Par Value)，是指固定收益票面金额。票面利率是发行人支付给持有人利息，有时也称名义利率(Nominal Rate)，票面利率一般指年利率。

4.1.3　常见应计期间计算方法

实务中应计期间有以下几种计算方法。

(1) Act/Act：分子分母都是按照实际天数计算，大多数年份是 365 天，闰年是 366 天。

(2) Act/360：按实际天数计息，一年按 360 天计算。

(3) Act/365：按实际天数计息，一年按 365 天计算。

(4) Act/365(Japanese)：按实际天数计息，每月 30 日，每年 365 天，不考虑闰年。

(5) Act/365(闰年 366)：按实际天数计算，一年 365 天，闰年 366 天。

(6) 30/360(SIA)：美国证券业协会(Securities Industry Association，SIA)规定，应计天数规则如下：①如果起始日与到期日都是二月最后一天，到期日按 30 号。例如某债券起始日为 2006 年 2 月 28 日，到期日为 2007 年 2 月 28 日，则应按照 2006 年 2 月 28 日至 2007 年 2 月 30 日计算。②如果起始日为某月 31 日，或者为二月最后一天，则改为 30 日；如果到期日也为某月 31 号，则最后一天改为 30 日。例如某债券起始日为 2006 年 1 月 31 日，到期日为 2006 年 3 月 8 日，则应按照 2006 年 1 月 30 日至 2006 年 3 月 8 日计算。如果该债券到期日为 2006 年 3 月 31 日，则应按照 2006 年 1 月 31 日至 2006 年 3 月 30 日计算。

(7) 30/360(PSA)：每月 30 天，每年 360 天，如果到期日为 2 月的最后一天，按照 30 日计算。

(8) 30/360(ISDA)：每月有 30 天，每年按 360 天计算。如果起始日为某月的 31 号，则改为 30 号；如果起始日为某月的 30 号，但到期日为某月的 31 号，则改为 30 号。例如某债券起始日为 1 月 31 日，到期日为 6 月 30 日，应按照 1 月 30 日至 6 月 30 日计算；如果起始日为 1 月 31 日，到期日为 12 月 31 日，应按照 1 月 30 日至 12 月 30 日计算。

(9) 30/360(European)：每月 30 日，每年 360 天，如果遇到起始日或到期日是 31 日，直接按照 30 日计算利息。

(10) 30/360：每月 30 日，每年 360 天，如果起息日是 31 日，则应该转换成 30 日。如果到期日是 31 日且起息日是 30 日或 31 日，则到期日应转换为 30 日。

(11) 30E/360：每月 30 日，每年 360 天，到期日为 31 日时按照 30 天计算。

前面 4 种方法比较易于理解，就是以实际天数计算。但注意两个日期之间的天数只能包括其中之一。例如 11 月 10 日到 11 月 15 日，实际天数是 4 天，11 月 15 日不应包含进去。

现在主要解释 30/360 和 30E/360 方法，我们用下面的标记解释。

D1/M1/Y1 表示第一个日期的日(D1)、月(M1)、年(Y1)；

D2/M2/Y2 表示第二个日期的日(D2)、月(M2)、年(Y2)。

30/360 规则是假设每月只有 30 天，每年 360 天，如果 D1 为 31，将之改为 30；如果 D2 为 31 而且 D1 为 30 或 31，将 D2 改变为 30，否则将之保留为 31，两个日期之间的天数为

$$(Y2-Y1)\times 360+(M2-M1)\times 30+(D2-D1)$$

例如 5 月 1 日至 5 月 30 日只有 29 天，5 月 1 日至 5 月 31 日期间有 30 天。

30E/360 规则是假设每月只有 30 天，每年有 360 天。假如 D1 为 31，将之改为 30；如果 D2 为 31，将之改为 30。接着，两个日期之间的天数为

$$(Y2-Y1)\times 360+(M2-M1)\times 30+(D2-D1)$$

与 30/360 日计数惯例中相同，5 月 1 日至 5 月 30 日之间有 29 天。然而，与先前日计数惯例不同，5 月 1 日至 5 月 31 日期间只有 29 天，而不是 30 天。这是因为，D2 为 31，被改为 30。表 4.1 为全球主要债券市场日计数惯例和息票付款频率。

表 4.1　全球主要债券市场日计数惯例和息票付款频率

市　场	息票付款	日 计 数
美国政府债券	每半年	ACT/ACT
美国公司债券	每半年	30/360
美国政府机构债券	每年	30/360
	每半年	
	每季	
美国市政债券	每半年	30/360
英国政府债券	每半年	ACT/365
澳大利亚政府债券	每半年	ACT/ACT
新西兰政府债券	每半年	ACT/ACT
加拿大政府债券	每半年	ACT/ACT
德国政府债券	每年	30E/360
瑞士政府债券	每年	30E/360
荷兰政府债券	每年	30E/360
欧洲债券	每年	30E/360
意大利政府债券	每年	30E/360
法国债券	每年	ACT/ACT

续表

市　场	息票付款	日 计 数
丹麦政府债券	每年	30E/360
瑞典政府债券	每年	30E/360

4.1.4　美国国债报价方式

美国中长期国债、本金分离登记交易(Strips，简称债券剥离)采用分数报价，分数部分用冒号或者连字符“-”分开，用 1/32 的倍数表示，例如 95-3/4，表示 95.75 美元。表 4.2 说明百分比报价与其对应的实际价格。

表 4.2　国债报价方式实例

百分比报价	面　值	对应小数形式	实际价格
98	1000	0.98	980
88$\frac{1}{2}$	10000	0.885	8850
951$\frac{1}{64}$	100000	0.9517188	95171.88
100	5000	1.00	5000
104$\frac{3}{4}$	10000	1.0475	10475
105$\frac{19}{32}$	1.055937	100000	1035937.50
107$\frac{3}{8}$	1.07375	250000	257768.44
106	1.06	1000	1060.00

全价(Full Price)①包括了应计利息(Accrued interest)，净价(Clean Price)②等于全价减去应计利息，即

$$净价=全价-应计利息$$

需要注意的是，在全价计算中，下一笔息票付款是一个贴现值，但在应计利息中，它却是非贴现值。应计利息计算公式如下

$$AI = c\times\frac{上次付息日到交割日之间的天数}{计息周期天数}$$

其中 AI 为应计利息，c 为每半年支付金额。

【例 4-1】2000 年 7 月 17 日交割的公司债券的息票利率为 10%，面值 100 元，2006 年 3 月到期，交割日和下一个付息日(2000 年 9 月 1 日)之间的天数为 44 天，计息周期天数为

① 又称脏价(Dirty Price)

② 又称平价(Flat Price)

180天，从上一个付息日(2000年3月1日)到交割日之间天数为136(180−44)天(天数计算规则是30/360)。应计利息为

$$AI=5\times\frac{136}{180}=3.7778$$

在美国，债券交易按净价报价，其他市场一般按全价报价。

4.1.5 绝对利差、静态利差(Static Spread)和期权调整后利差(Option Adjusted Spread，OAS)

不同债券的到期收益率之差称为收益率溢价，或者利差(Yield Spread)。绝对利差指的是两个债券品种的到期收益率之差，例如某个债券的到期收益率为5%，同期国债的收益率为4%，那么绝对利差是1%(5%−4%)。

静态利差是指假定投资者持有债券到偿还期，债券所实现的收益会比同期国债高多少。静态利差不是公司债券到期收益率与国债收益率简单相减，而是衡量债券到期收益率曲线超过国债到期收益率曲线的程度。静态利差也被称为Z−利差，Z表示Zero，是指波动率为零时的利差。静态利差的计算公式为

$$P_0=\sum_{t=1}^{N}\frac{C_t}{(1+r_t+r_{ss})^t}+\frac{F}{(1+r_N+r_{ss})^N}$$

其中，r_{ss}为静态利差，r_t是t时刻利率，r_N是N时刻利率。

由于利差没有考虑到某些债券的含权属性，因此该指标存在一定问题。例如一个公司的债券到期收益率为8%，而同期国债收益率为6%，这里存在2个百分点利差，但是如果该公司是含权的(债券上面附加了一些保证条款)，比如可以提前回购，这时利差就属于期权价值，称为期权调整利差(Option Adjusted Spread，OAS)。

4.2 现金流计算函数

4.2.1 固定收益证券基本概念

1. 现值(Present Value)与终值(Future Value)

现值是指人们将来可以收到的1元钱在价值上要低于现在的1元钱。终值是指现在投入1元钱，投资者将来某个时刻收到的钱。

2. 即期利率(Spot Interest Rate)与远期利率(Forward Interest Rate)

所谓即期利率是指某个给定时点上无息债券的到期收益率，即期利率可以看作是一个

即期合约的利率水平，这种合约一旦签订，资金立即从债权人转移到借款人手里，由借款人在将来某个特定时点按照合约中标明的利率水平连本带利全部还清。

如果投资者以价格 P_1 购买期限为 n 年的无息债券，债券到期后从发行人那里获得一次性现金支付为 M_n，那么第 n 年期即期利率 r_n 与 P_1、M_n 之间关系如下：

$$P_1 = M_n/(1+r_n)^n$$

对于期限较长的附息债券，即期利率的确定方式有所不同，例如某投资者以 P_2 价格购买期限为 2 年、面值为 F 的附息债券，每年支付利息为 C，在这种情况下，通常用一年期无息债券来计算一年期即期利率 r_1，那么两年期即期利率 r_2 计算需要求解下面的方程：

$$P_2 = \frac{C}{(1+r_1)^1} + \frac{C+F}{(1+r_2)^2}$$

所谓远期利率是指未来两个时点之间的利率水平，远期利率可以看做是与一个远期合约有关的利率水平，一个远期利率在现在签订的合约中规定，但与未来一段时期有关，这也就是说，远期合约利率的条件现在已经确定，但实际交割将在以后进行。

3. 零息利率(Zero-Rate)

零息利率是零息债券持有到期收益率，如果市场上有各种不同期限的零息债券，很容易衡量零息利率。然而，现实中零息债券非常少，一般通过货币市场和资本市场中的其他工具估算零息利率，例如通过平价收益率曲线建立零息利率曲线，这个过程称为“剥离”方法。

4. 当前收益率(Current Yield)

当前收益率为债券每年支付的利息和价格之比，例如债券的价格为 100 元，每年支付的利息为 5 元，则当前收益率为 5/100=5%。

5. 到期收益率(Yield To Maturity，YTM)

固定收益证券对应于一系列现金流组合，投资者在初始时刻支付一笔现金，将来时刻可以收到利息，如果使用某个收益率对将来现金流进行贴现，而贴现结果正好等于初始时刻支付的现金，这个收益率就是到期收益率。到期收益率隐含两个假设，一是投资者持有到期；二是利息再投资的利率是不变的。到期收益率又称内部收益率。

4.2.2　现金流基本计算

1. 现金流现值

现金流现值的计算公式如下：

$$PV = P_0 + \frac{P_1}{(1+r)} + \frac{P_2}{(1+r)^2} + \frac{P_3}{(1+r)^3} + \cdots + \frac{P_n}{(1+r)^n}$$

其中 P_i 为第 i 期现金流序列，r 为贴现率。现值计算过程也称为折现，因此现值有时称为折现值(discounted value)，利率 r 称为折现率(discount rate)。MATLAB 中计算现金现值的函数为 pvvar。

调用方式

```
PresentVal = pvvar(CashFlow, Rate)
```

输入变量

```
CashFlow  %各期的现金流序列
Rate  %贴现率
```

输出变量

```
PresentVal  %现金流的现值
```

【例 4-2】一项投资各年现金流如表 4.3 所示，贴现率为 0.08，求其现值。

表 4.3　各年现金流

当　前	第 1 年	第 2 年	第 3 年	第 4 年	第 5 年
-10000	2000	1500	3000	3800	5000

记各期现金流为 CashFlow，把现金流 CashFlow 写成向量形式[-10000 2000 1500 3000 3800 5000]，执行下面的命令：

```
>> CashFlow=[-10000 2000 1500 3000 3800 5000];Rate=0.08;
>> pvvar(CashFlow, Rate)
ans =
1. 7154e+003
```

计算现值为 1715.4。

注意，pvvar 函数计算现值时必须输入当前时刻现金投入，一般计算现值时不会考虑当前时刻投入，例如在上例中如果直接计算第 1 年、第 2 年、第 3 年、第 4 年和第 5 年现值，必须将当前时刻现金流设为 0，MATLAB 中输入命令如下：

```
>> CashFlow=[0 2000 1500 3000 3800 5000];
>> pvvar(CashFlow, 0.08)
ans =
1.715e+004
```

【例 4-3】表 4.4 为一项投资现金流及其对应日期，贴现率为 0.09，求其现值。

表 4.4　投资各个日期发生的现金流

现金发放日	12-Jan-1987	14-Feb-1988	3-Mar-1988	14-Jun-1988	1-Dec-1988
金额	-10000	2500	2000	3000	4000

与例 4-1 不同的是，给定了现金流发生日期，把日期存入变量 IrrCFDates，按照 PresentVal = pvvar(CashFlow, Rate, IrrCFDates)格式调用。Command 窗口中执行如下命令：

```
>> CashFlow = [-10000, 2500, 2000, 3000, 4000];
>> IrrCFDates = ['01/12/1987'
                 '02/14/1988'
                 '03/03/1988'
                 '06/14/1988'
                 '12/01/1988'];
>> PresentVal = pvvar(CashFlow, Rate, IrrCFDates)
PresentVal =
  142.1648
```

计算出现值为 142.1648。

2. 计算现金流终值

现金流现值考虑的时点是现在，而终值考虑的时点是债券到期日，MATLAB 中用 fvvar 函数计算终值。

调用方式

```
FutureVal = fvvar(CashFlow, Rate, IrrCFDates)
```

输入参数

同 pvvar 函数。

输出参数

同 pvvar 函数。

【例 4-4】现金流同例 4-1，求该现金流终值。可以在 MATLAB 中执行如下命令：

```
>> CashFlow=[-10000 2000 1500 3000 3800 5000];Rate=0.08;
>> FutureVal = fvvar(CashFlow, Rate)
  FutureVal =
  2.5205e+003
```

如果将 FutureVal 贴现到当前时刻就是现值，代码如下：

```
>> FutureVal/(1+Rate)^5
  ans =
  1.7154e+003
```

贴现结果同例 4-1。

3. 计算年金利率

如果每年的现金流都是一样的，这种现金流称为年金。如果现金支付发生在年初称为及时年金，发生在年末称为普通年金。

调用方式

```
Rate = annurate(NumPeriods, Payment, PresentValue, FutureValue, Due)
```

输入参数

```
NumPeriods       %偿还期间数
Payment          %每次偿还的金额
PresentValue     %年金的现值
FutureValue      %(Optional)年金的终值
Due              %(Optional)普通年金 Due=0(默认值)，在期末支付年金；及时年金 Due=1
                  在期初支付年金
```

输出参数

```
Rate             %年金的收益率。
```

【例 4-5】有一笔贷款金额为 5000 元，4 年还清，每月 130 元，求月利率。4 年合计 4*12=48 个月。

计算月利率，在 MATLAB 中执行以下命令：

```
>> Rate = annurate(4*12, 130, 5000, 0, 0)
Rate =
    0.0094
```

该笔贷款月利率为 0.94%。若计算年贷款利率，在 MATLAB 中执行以下命令：

```
>> 12*Rate
ans =
    0.1132
```

年利率为 11.32%。

4. 计算年金期间数

调用方式

```
NumPeriods = annuterm(Rate, Payment, PresentValue, FutureValue, Due)
```

输入参数

同上。

输出参数

同上。

【**例 4-6**】如果一个储蓄账户中已经有了 1500 美元，计划每月存入 200 美元，利率为 4%，多少月后可以达到 6000 美元？

在 MATLAB 中执行以下命令：

```
>>NumPeriods = annuterm(0.04/12, 200, 1500, 6000, 0)
NumPeriods =
   21.2206
```

大概 21.2206 月后可以到达 6000 美元。

5. 计算内部收益率函数

内部收益率是使投资现金流现值等于投资价格的收益率，内部收益率的计算公式为

$$P=\frac{C_1}{(1+r)}+\frac{C_2}{(1+r)^2}+\frac{C_3}{(1+r)^3}+\cdots+\frac{C_N}{(1+r)^N}$$

其中，C_i 为第 i 年度现金流，P 为债券价格，N 为年数，r 为内部收益率。

MATLAB 中计算内部收益率的函数是 irr。

调用方式

```
Return = irr(CashFlow)
```

输入变量

```
CashFlow          %从 0 期开始的各期现金流
```

输出变量

```
Return            %内部收益率
```

计算现金流的函数有几种，irr 函数就是一种，irr 函数的调用形式为：

```
Return = irr(CashFlow)
```

【**例 4-7**】一项投资各期现金流如表 4.5 所示。

表 4.5　各期投资现金流

第 0 期	第 1 期	第 2 期	第 3 期	第 4 期
-5000	1000	2000	3000	4000

问该项投资的内部收益率是多少？

把各期现金流保存在变量 CashFlow 中，把 CashFlow 写成向量形式[-5000　1000　2000　3000　4000]，Return 为该项目的内部收益率，执行如下命令：

```
>> CashFlow=[-5000 1000 2000 3000 4000]
CashFlow = -5000       1000       2000       3000       4000
>> irr(CashFlow)
ans = 0.2727
```

则该项投资的内部收益率为 27.27%。

【例 4-8】某公司债券票面利率为 8%，1 年支付一次利息，期限为 5 年，价格为 103.7 元，到期收益率为 7.09%，同期的国债收益率期限结构如表 4.6 所示。

表 4.6　国债利率期限结构

年	1	2	3	4	5
到期收益率	4.51	4.68	4.84	4.99	5.14

试计算静态利差与绝对利差。

首先我们根据公司债券计算到期收益率，在 MATLAB 中执行以下命令：

```
>> r1=irr([-103.7 8 8 8 8 108])
r1 =
   0.0710
```

我们得到公司债券到期收益率为 7.1%。下面我们计算 5 年期国债到期收益率，考虑求解面值为 100 元的 5 年期国债票息，设票息为 c，则有下面方程：

$$100=\frac{c}{(1+4.51\%)}+\frac{c}{(1+4.68\%)^2}+\frac{c}{(1+4.84\%)^3}+\frac{c}{(1+4.99\%)^4}+\frac{100+c}{(1+5.14\%)^5}$$

解得 $c=5.1094$。

我们找出对应的到期收益率，在 MATLAB 中执行以下命令：

```
>> r2=irr([-100,c,c,c,c,100+c])
r2 =
   0.0511
```

这样 5 年期国债对应的到期收益率为 5.11%，绝对利差为 1.99%(7.1%−5.11%=1.99%)。

下面我们求解静态利差。静态利差实际上就是求解下面的方程：

$$103.71=\frac{8}{(1+4.51\%+r_{ss})}+\frac{8}{(1+4.68\%+r_{ss})^2}+\frac{8}{(1+4.84\%+r_{ss})^3}+\frac{8}{(1+4.99\%+r_{ss})^4}+\frac{100+8}{(1+5.14\%+r_{ss})^5}$$

下面调用 MATLAB 中的函数进行求解：

```
>>eq=sym('103.7=8/(1+0.0451+x)+8/(1+0.0468+x)^2+8/(1+0.0484+x)^3...
      +8/(1+0.0499+x)^4+(100+8)/(1+0.0514+x)^5');
>> a=solve(eq,'x')
```

我们看到 a 的解有很多，包括复数根与负根，但只有一个(a=0.02)符合要求，所以静态利差为 0.02。

4.2.3　计算复杂形式现金流

1．根据贴现率、债券发行日、到期日计算债券收益率

MATLAB 中根据贴现率、债券发行日、到期日计算债券收益率的函数是 tbilldisc2yield。

调用方式

```
[BEYield MMYield] = tbilldisc2yield(Discount, Settle, Maturity)
```

输入参数

```
Discount        %贴现率
Settle          %结算日
Maturity        %到期日
```

输出参数

```
BEYield         %根据一年 365 天计算的收益率
Maturity        %根据一年 360 天计算的收益率
```

【例 4-9】某债券结算日为 2002 年 10 月 1 日，到期日为 2003 年 3 月 31 日，年贴现率为 0.0497，求债券收益率。

在 MATLAB 中执行以下命令：

```
>> Discount = 0.0497;
>> Settle = '01-Oct-02';
>> Maturity = '31-Mar-03';
>> [BEYield MMYield] = tbilldisc2yield(Discount, Settle, Maturity)
BEYield =
       0.0517
MMYield =
       0.0510
```

2．根据债券收益率计算贴现率

tbillyield2disc 函数是 tbilldisc2yield 函数的逆函数。

【例 4-10】一种债券计息日是 2002 年 10 月 1 日，到期日是 2003 年 3 月 31 日，收益率是 4.97%，求其贴现率。执行命令如下：

```
>> Yield = 0.0497;Settle = '01-Oct-02';Maturity = '31-Mar-03';
>> Discount = tbillyield2disc(Yield, Settle, Maturity)
Discount =
   0.0485
```

3. 计算债券价格

根据债券特征计算出债券价格的函数是 tbillprice。

调用方式

```
Price = tbillprice(Rate, Settle, Maturity, Type)
```

输入参数

```
Rate            %债券的年收益率
Settle          %债券的结算日
Maturity        %债券的到期日
Type            %(Optional)债券的类型，Type=1(默认值)表示货币市场工具，Type=2 表
                  示债券，Type=3 表示贴现率①
```

输出参数

```
Price           %债券的价格
```

【**例4-11**】已知债券结算日是2002年10月1日，到期日是2003年3月1日，债券收益率为4.5%，求该债券价格，这时可以执行如下命令：

```
>> Rate = 0.045;Settle = '01-Oct-02';Maturity = '31-Mar-03';
>> Price = tbillprice(Rate, Settle, Maturity, Type)
Price =
   97.8172
```

4. 将年回报率转化为相应的月回报率

MATLAB 中将年回报率转换为月回报率的函数是 effrr。

调用方式

```
Return = effrr(Rate, NumPeriods)
```

输入参数

```
Rate            %债券的年回报率
NumPeriods      %年支付利率的次数
```

输出参数

```
Return          %转化后的利率
```

【**例 4-12**】一项投资为 9 年，年回报率为 9%，问平均每月投资回报率是多少？首先把 9%写成小数形式，然后调用 effrr 函数，在 MATLAB 中执行如下命令：

```
>> Return = effrr(0.09, 12)
```

① 货币市场按一年 360 天计算，债券按一年 365 天计算，贴现率按一年 360 天计算。

```
Return =
    0.0938
```

这样每月收益率为 9.38%。

5. 债券价格给定的零息券收益率

当债券价格给定时，计算零息券到期收益率的函数是 zeroyield。

调用方式

```
Yield = zeroyield(Price, Settle, Maturity, Period, Basis, EndMonthRule)
```

输入参数

```
Price           %债券价格
Settle          %债券的结算日
Maturity        %债券的到期日
Period          %(Optional) 年发放票息的频率，默认值是 2
Basis           %(Optional)应计天数的计算方式①，内容如下：
                0 = actual/actual (default)
                1 = 30/360
                2 = actual/360
                3 = actual/365
                4 = 30/360 (PSA)
                5 = 30/360 (ISDA)
                6 = 30/360 (European)
                7 = actual/365 (Japanese)
EndMonthRule    %(Optional)月末法则。仅对到期日是 30 日或者小于 30 日有效，0 表示发放
                 票息的日期相同，1(默认值)表示票息在最后一天发放
```

输出参数

```
Yield           %债券到期的收益率
```

MATLAB 中到期收益率有如下两种形式，如果是零息券或者只有一个似息票期(如果债券除了零息外还有其他支付方式)的赎回期，收益率计算公式如下：

$$Yield = \left(\frac{RV - P}{P}\right)\left(\frac{M \times E}{DSR}\right)$$

上式前面一项是投资美元收益，后面一项是年度利率。如果赎回期前有多个支付票息期间，则应采用下列公式计算：

$$Yield = \left[\left(\frac{RV}{P}\right)^{\frac{1}{N_q - 1 + \frac{DSC}{E}}} - 1\right] \times M$$

① 本书中的其他函数如需输入参数 Basis，如无特别声明，其含义均同本例。

其中，*DSC* 表示从结算日到下一个似息票日的天数，*DSR* 表示从结算日到赎回日的天数，*E* 表示似债券的天数，*M* 表示一年中似息票的时间段，N_q 表示在计息日到赎回日之间似息票的时间段，*P* 表示面值为 100 美元的零息票的价格，*RV* 表示赎回的价格，*Yield* 表示持有债券到赎回日收益。

【例 4-13】计算一个短期债券收益率，如果结算日为 1993 年 6 月 24 日，到期日为 1993 年 11 月 1 日，应计期间计算方法为 Actual/Actual，该债券价格为 95 元，求其收益率。

在 MATLAB 中执行以下命令：

```
>> Settle   = '24-Jun-1993';
>> Maturity = '1-Nov-1993';
>> Basis    = 0;
>> Price    = 95;
>> Yield = zeroyield(Price, Settle, Maturity, [], Basis)
Yield =
    0.1490
```

6. 固定收益到期收益率(Yield To Maturity)

调用方式

```
Yield = bndyield(Price, CouponRate, Settle, Maturity, Period, Basis,
                 EndMonthRule, IssueDate, FirstCouponDate, LastCouponDate,
                 StartDate, Face)
```

输入参数

```
Price           %债券的净价(Clean Price)
CouponRate      %票息率
Settle          %结算日
Maturity        %到期日
Period          %(Optional) 每年支付票息的次数，可以为 0, 1, 2, 3, 4, 6, 12, 默认
                  值为 2
Basis           %(Optional) 应计天数法则
EndMonthRule    %(Optional) 月末法则
IssueDate       %(Optional) 发行日
FirstCouponDate %(Optional) 首次付息日
LastCouponDate  %(Optional) 最后一次付息日
StartDate       %(Optional) 现金收到日
Face            %(Optional) 债券的面值，默认值为 100
```

输出参数

```
Yield           %到期收益率
```

【例 4-14】已知债券的票息率为 0.05，结算日为 1997 年 1 月 20 日，到期日为 2002 年

6 月 15 日，每年支付两次票息，天数方法为 actual/actual，求到期收益率。

在 MATLAB 中执行以下命令：

```
>> Price = 95;
>> CouponRate = 0.05;
>> Settle = '20-Jan-1997';
>> Maturity = '15-Jun-2002';
>> Period = 2;
>> Basis = 0;
>> Yield = bndyield(Price, CouponRate, Settle, Maturity, Period, Basis)
Yield =
   0.0610
```

债券到期收益率为 0.0610。

7. 现金流转换为对应的债券

MATLAB 中将现金流转换为对应债券的函数是 cfamounts。

调用方式

```
[CFlowAmounts, CFlowDates, TFactors, CFlowFlags] = cfamounts(CouponRate,
Settle, Maturity, Period, Basis, EndMonthRule, IssueDate, FirstCouponDate,
LastCouponDate, StartDate, Face)
```

输入参数

```
CouponRate       %债券票息
Settle           %结算日
Maturity         %到期日
Period           %息票支付的频率，取值为 0，1，2，3，4，6，12，默认值为 2，表示每半年
                  支付 1 次利息
Basis            %(Optional)应计天数法则
EndMonthRule     %(Optional)月末法则
IssueDate        %(Optional)债券发行日
FirstCouponDate  %(Optional)债券息票首付日
LastCouponDate   %(Optional)最后一次付息日
StartDate        %(Optional)开始日期
Face             %面值
```

输出参数

```
CFlowAmounts     %现金流的数量
CFlowDates       %现金流的日期
TFactors         %时间因子(time factors)。时间因子的计算公式如下
```

$$PV = \frac{CF}{\left(1+\frac{z}{2}\right)^{TF}}$$

其中，*PV* 为现金流现值，*CF* 为现金流，*z* 是经过风险调整的年收益率，*TF* 是时间因子

```
CFlowFlags        % 现金流类型，其内容如下：
                  0 到期日的累计利息
                  1 初次现金流，发放的日期一般小于正常的息票期间(Coupon Period，相邻
                    两次票息间隔的时间)
                  2 初次现金流，发放时间大于正常的票息期间
                  3 到期现金流为名义票息
                  4 到期日现金流为本金加利息
                  5 距离到期日不满一个息票期间，最后一次息票期间短于其他息票期间，到期
                    现金流小于名义现金流
                  6 由于最后一个息票期间长于其他息票期间，导致到期现金流高于正常的到期
                    现金流
                  7 距离到期日不满一个息票期间，现金流为到期现金流
                  8 距离到期日不满一个息票期间，现金流小于到期现金流
                  9 距离到期日不满一个息票期间，现金流大于到期现金流
                  10 零息券的到期现金流
```

【例 4-15】将两个现金流转换为对应的债券，现金流的结算日为 1993 年 11 月 1 日，到期日为 1994 年 12 月 15 日与 1995 年 6 月 15 日，票息分别为 6%与 5%，债券发放票息次数分别为 4 次与 2 次。应计天数计算规则分别是 Actual/Actual 与 30/360(SIA)，试给出对应债券。

在 MATLAB 中执行以下命令：

```
>> Settle = '01-Nov-1993';
>> Maturity = ['15-Dec-1994';'15-Jun-1995'];
>> CouponRate= [0.06; 0.05];
>> Period = [4; 2];
>> Basis = [1; 0];
>> [CFlowAmounts, CFlowDates, TFactors, CFlowFlags] =
cfamounts(CouponRate,Settle, Maturity, Period, Basis)
CFlowAmounts =
   -0.7667    1.5000    1.5000    1.5000    1.5000  101.5000
   -1.8989    2.5000    2.5000    2.5000  102.5000       NaN
CFlowDates =
     728234      728278      728368      728460      728552      728643
     728234      728278      728460      728643      728825         NaN
TFactors =
         0    0.2404    0.7403    1.2404    1.7403    2.2404
         0    0.2404    1.2404    2.2404    3.2404       NaN
CFlowFlags =
```

```
     0     3     3     3     3     4
     0     3     3     3     4   NaN
>> datestr(CFlowDates(1,1:6))
ans =
               01-Nov-1993
               15-Dec-1993
               15-Mar-1994
               15-Jun-1994
               15-Sep-1994
               15-Dec-1994
>> datestr(CFlowDates(2,1:5))
ans =
               01-Nov-1993
               15-Dec-1993
               15-Jun-1994
               15-Dec-1994
               15-Jun-1995
```

表 4.7 分别是两种现金流所对应的债券。

表 4.7 现金流转化为债券格式

日期	93-11-1	93-12-15	94-3-15	94-6-15	94-9-15	94-12-15
债券 1	1.5	1.5	1.5	1.5	1.5	101.5
债券 2	2.5	2.5	2.5	2.5	102.5	NaN

4.2.4 短期债券回购计算

调用方式

```
TBEDiscount = tbillrepo(RepoRate, InitialDiscount, PurchaseDate, SaleDate,
                   Maturity)
```

输入参数

```
RepoRate            %债券的回购率
InitialDiscount     %初始贴现率
PurchaseDate        %购买日期
SaleDate            %卖出日期
Maturity            %到期日
```

输出参数

```
TBEDiscount         %回购盈亏平衡点的贴现率
```

记 T_1 为 2002 年 9 月 26 日到 2002 年 12 月 26 日的天数，T_2 为 2002 年 10 月 26 日到 2002

年 12 月 26 日的天数，T_1、T_2 之间的关系示意图如图 4.1 所示。

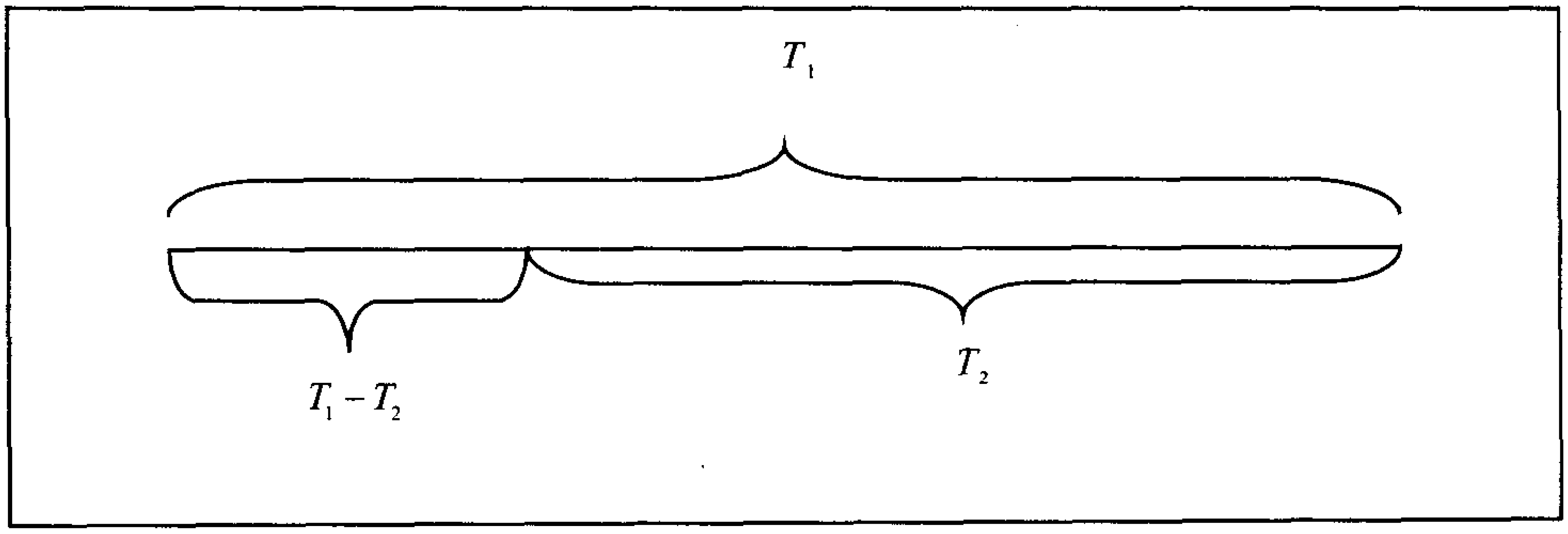


图 4.1 T_1、T_2 之间的关系示意图

从图 4.1 中可以看出，投资者可以有两种选择，一种是从 2002 年 9 月 26 日持有债券一直到期满；另一种是从 2002 年 9 月 26 日到 2002 年 10 月 26 日进行回购，然后从 2002 年 10 月 26 日到 2002 年 12 月 26 日投资其他品种债券，该债券贴现率应使得上述两种选择的最终收益率相等。

【例 4-16】短期债券初始贴现率为 0.045，债券到期日为 2002 年 4 月 3 日，购买债券日期为 2002 年 1 月 3 日，卖出日期为 2002 年 2 月 3 日，回购利率为 0.045，求该项投资盈亏平衡点贴现率。

在 MATLAB 中执行以下命令：

```
>> RepoRate=0.045;
>> InitialDiscount = 0.0475;
>> PurchaseDate = '3-Jan-2002';
>> SaleDate = '3-Feb-2002';
>> Maturity = '3-Apr-2002';
>> TBEDiscount=tbillrepo(RepoRate,InitialDiscount,PurchaseDate,SaleDate,
               Maturity)
   TBEDiscount =
             0.0491
```

【例 4-17】Abel 银行与 Baker 达成一笔 5000 万美元 7 天回购的协议，抵押品是市场价格为 102.00 美元，年利率为 8%的美国政府债券，为达成回购协议，其价格定为 101.00 美元，年利率为 7.5%，求债券回购价格。

分析：Baker 得到 49 504 950 美元面值的债券抵押(50 000 000/101×100)，这样 Abel 获得了 5 000 000 美元借款，在回购期满后，Abel 按照协议利率(7.5%)支付了 72 917 美元利息(50 000 000×7.5%×7/360)，Baker 则归还全部抵押债券。

在 MATLAB 中执行以下命令：

```
>> MarketPrice=102;AgreePrice=101;RepoRate=0.075;ParValue=100;
              Amount=50000000;
>> Bond=Amount/AgreePrice;TotalAmount=Amount+Amount*RepoRate*7/360;
>> RepoPrice=TotalAmount/Bond
   RepoPrice =
          101.1473
```

债券回购价格应该为 101.1473 美元。

4.2.5 对美国短期债券进行定价

调用方式

```
Price =tbillprice(Rate, Settle, Maturity, Type)
```

输入参数

```
Rate              %债券的收益率
Settle            %债券的结算日
Maturity          %债券的到期日
Type              %(Optional)债券的类型，1 表示货币市场(actual/360)，2 表示债券
                   (actual/365)，3 表示贴现率(actual/360)，默认值是 1
```

输出参数

```
Price      %债券的价格
```

【例 4-18】 短期美国国债利率为 4.5%，结算日为 2002 年 10 月 1 日，到期日为 2003 年 3 月 31 日，求该债券价格。

在 MATLAB 中执行以下命令：

```
>> Rate = 0.045;
>> Settle = '01-Oct-02';
>> Maturity = '31-Mar-03';
>> Type = 2; %计息方法采用 actual/365 方法
>> Price = tbillprice(Rate, Settle, Maturity, Type)
    Price =
         97.8172
```

实际上该债券利率采用 actual/365 方法，从 2002 年 10 月 1 日到 2003 年 3 月 31 日，实际天数为 181 天。

```
>> Rate = 0.045;
```

```
>> price=Face/[1+Rate*181/365]
   price =
          97.8172
```

4.2.6 国库券收益

调用方式

```
[MMYield, BEYield, Discount] = tbillyield(Price, Settle, Maturity)
```

输入参数

```
Price              %面值为 100 的国库券的价格
Settle             %结算日
Maturity           %到期日
```

输出参数

```
MMYield            %货币市场的收益①
BEYield            %债券市场的收益
Discount           %债券的贴现率
```

【例 4-19】 已知债券的价格为 98.75，结算日期为 2002 年 10 月 1 日，到期日为 2003 年 3 月 31 日，将其分别折算成货币市场收益率、债券市场收益率和贴现率。

在 MATLAB 中执行以下命令：

```
>> Price = 98.75;
>> Settle = '01-Oct-02';
>> Maturity = '31-Mar-03';
>> [MMYield, BEYield, Discount] = tbillyield(Price, Settle,Maturity)
  MMYield =
           0.0252
  BEYield =
          0.0255
  Discount =
          0.0249
```

计算得到折算成货币市场收益率为 0.0252，债券市场收益率为 0.0255，贴现率为 0.0249。

---

① 货币市场的收益率按照实际天数/360(actual/360)计算，债券市场收益率按照实际天数/365(actual/365)计算。

4.2.7　可转让定期存单(CD)定价

1. 可转让定期存单累计收益

MATLAB 中计算可转让定期存单累计收益率的函数为 cdai，面值默认 100 元，应计天数法则默认 actual/360(SIA)。

调用方式

```
AccrInt = cdai(CouponRate, Settle, Maturity, IssueDate, Basis)
```

输入参数

同前。

输出参数

```
AccrInt           %可转让定期存单的累计收益
```

【例 4-20】已知可转让定期存单票息为 0.05，结算日 2002 年 1 月 2 日，到期日 2002 年 3 月 31 日，发行日为 2001 年 10 月 1 日，求其累计收益。

在 MATLAB 中执行以下命令：

```
>> CouponRate=0.05;
>> Settle='02-Jan-02';
>> Maturity='31-Mar-02';
>> IssueDate='1-Oct-01';
>> AccrInt=cdai(CouponRate, Settle, Maturity, IssueDate)
AccrInt =
    1.2917
```

可转让定期存单(Certificate of Deposit，CD)累计收益为 1.2917。

2. 计算 CD 收益率

MATLAB 中计算可转让定期存单收益率的函数为 cdyield。

调用方式

```
Yield = cdyield(Price, CouponRate, Settle, Maturity, IssueDate, Basis)
```

输入参数

```
Price          %面值是 100 美元的净价(Clean Price)。如果是肮脏价①(Dirty Price)
                可以用函数 cdai 计算实际利息
CouponRate     %每年的收益率
Settle         %结算日
```

① 有时称全价。

```
Maturity        %到期日
IssueDate       %发行日
Basis           %(Option)应计天数法则
```

输出参数

```
Yield           %CD的收益率
```

【例 4-21】 可转让定期存单(CD)的发行日是 2002 年 10 月 1 日，结算日是 2002 年 1 月 2 日，到期日是 2002 年 3 月 31 日，交易价格为 101.125 美元，T、T_1 与 T_2 之间的关系如图 4.2 所示，求该可转让定期存单的收益率。

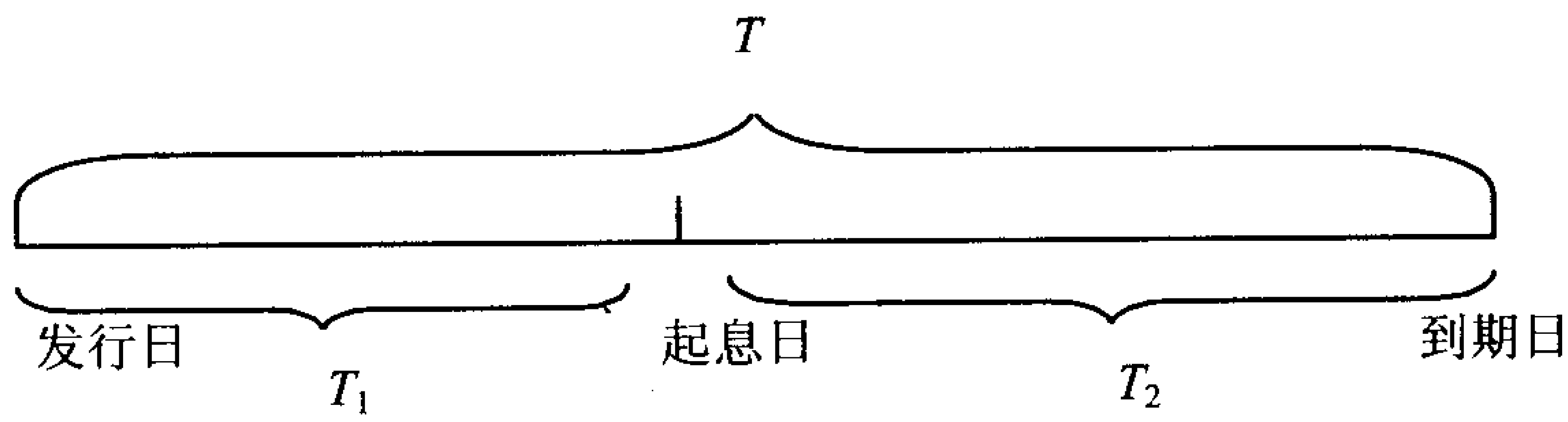


图 4.2　可转让存单时间示意图

起息日的票息=票息×T_1，则

$$\text{起息日收益}=\frac{\text{结算价}}{100}+\text{起息日的票息}$$

则有

$$\text{可转让定期存单的收益}=\left(\frac{1+\text{息票收益率}\times T}{\text{起息日收益}}-1\right)\times T_2$$

实际上根据公式有：

```
>> Price = 101.125;
>> T1=0.2583; %将 T1 折算成年
>> T2=0.2444;
>> T=0.5027;
>> AccrInt=CouponRate* T1;
>> B = Price/100 + AccrInt;
>> Yield = ( (1 + CouponRate* T) / B  - 1 ) * (1/T2)
Yield =
   0.0039
```

下面调用 cdyield 函数可以得到同样结果：

```
>> Price = 101.125;
>> CouponRate = 0.05;
>> Settle='02-Jan-02';
>> Maturity='31-Mar-02';
```

```
>> IssueDate ='1-Oct-01';
>> Yield = cdyield(Price, CouponRate, Settle, Maturity, IssueDate)
   Yield =
       0.0039
```

3. 可转让存款 CD 价格

调用方式

```
[Price, AccrInt] = cdprice(Yield, CouponRate, Settle, Maturity, IssueDate,
Basis)
```

输入参数

同前。

输出参数

```
Price           %CD 价格
AccrInt          %CD 累计收益
```

【例 4-22】 CD 收益率为 0.0525，票息率为 5%，结算日 2002 年 1 月 2 日，到期日 2002 年 3 月 31 日，发行日 2001 年 10 月 1 日，每半年支付一次票息。

在 MATLAB 中执行以下命令：

```
>> Yield          =  0.0525;
>> CouponRate      =  0.05;
>> Settle         =  '02-Jan-02';
>> Maturity        =  '31-Mar-02';
>> IssueDate       =  '1-Oct-01';
>> [Price, AccruedInt] = cdprice(Yield, CouponRate, Settle, Maturity,
                          IssueDate)
Price =
       99.9233
AccruedInt =
       1.2917
```

该 CD 价格为 99.9233，累计收益为 1.2917。

4.2.8　可转换债券定价

可转换债券是可转换为股票的债券，兼有公司债券和股票双重特点，一般情况下事先规定票面利率、转股价格、转股比率和转换期。票面利率就是债券持有到期时的收益率，转股价格是指可转债在转股期间转化为基础股票的每股价格。转换比率是指发行转债的公司向投资者约定比率，按照这一比率，投资者可以将手中的债券转换为相应数量的股票。转换期是指转债转换为股票起始日至结束日的期间，通常有下面 4 种情况：

(1) 发行后某日至到期前某日。

(2) 发行后某日至到期日。

(3) 发行日至到期日前某日。

(4) 发行日至到期日。

在前两种情况下，可转换债券有一段时间的锁定期，这段时间内债券持有人不可以转换为股票。

MATLAB 中采用 Cox-Ross-Rubinstein 二叉树方法计算可转换债券价格。

调用方式

```
[CBMatrix, UndMatrix, DebtMatrix, EqtyMatrix] = cbprice(RiskFreeRate,
StaticSpread, Sigma, Price, ConvRatio, NumSteps, IssueDate, Settle, Maturity,
CouponRate, Period, Basis, EndMonthRule, DividendType, DividendInfo,
CallType, CallInfo, TreeType)
```

输入参数

```
RiskFreeRate    %无风险利率
StaticSpread    %静态价差，为超过无风险利率部分
Sigma           %股票波动的标准差
Price           %标的资产的价格
ConvRatio       %转换比例
NumSteps        %计算的步数
IssueDate       %发行日
Settle          %结算日
Maturity        %到期日
CouponRate      %息票率
Period          %二叉树离散的时间段
Basis           %(Optional)应计天数法则，取值如下：
                 0 = actual/actual (default)
                 1 = 30/360 (SIA)
                 2 = actual/360
                 3 = actual/365, 4 = 30/360 (PSA)
                 5 = 30/360 (ISDA)
                 6 = 30/360 (European)
                 7 = act/365 (Japanese)
EndMonthRule    %月末法则，每月最后一日的处理方式。当月末等于 30 日或者小于 30 日时该
                 法则有效，0 表示忽略该法则，说明利息的支付日小于 30 号，1 表示红利的
                 支付总是在月末，默认值为 1
DividendType    %红利类型，0 表示红利以绝对形式发放，1 表示以收益率形式发放
DividendInfo    %除息信息，缺失值表示没有除息。第一行是除息日期，第二行是除息的数量
CallType        %看涨期权的类型，0(默认值)表示现金价，1 表示净价(Clean Price)，不
                 含应计利息的价格
CallInfo        %第一列为日期，面值为 100 元的债券的回购价格，默认值为无赎回条款
TreeType        %树形图的类型，0 为默认值，表示二叉树，1 表示三叉树
```

输出参数

```
CBMatrix        %可转换债券的可转换价格矩阵，CBMatrix(1, 1)为可转换价格
UndMatrix       %二叉树格式的债券价格
DebtMatrix      %可转换债券价格的债券部分，为一矩阵形式
EqtyMatrix      %可转换债券价格的股票部分，为一矩阵形式
```

【例 4-23】已知无风险利率为 0.05，标的资产波动标准差为 0.3，可转换债券转换比率为 1：1，二叉树时间离散数目为 200 个时间段。债券发行日为 2002 年 1 月 2 日，结算日为 2002 年 1 月 2 日，到期日为 2007 年 1 月 2 日，息票率为 0.04，每年支付 2 次利率，天数采用 1 = 30/360 (SIA)方法。月末法则默认值为 1，红利以绝对数量美元支付(而非比率)，在 2004 年 1 月 2 日以 110 元的价格赎回。

在 MATLAB 中执行以下命令：

```
mSteps     = 200;
IssueDate    = datenum('2-Jan-2002');
Settle       = datenum('2-Jan-2002');
Maturity     = datenum('2-Jan-2007');
CouponRate   = 0.04;
Period       = 2;
Basis        = 1;
EndMonthRule = 1;
DividendType = 0;
DividendInfo = [];
CallInfo     = [datenum('2-Jan-2004'), 110];
CallType     = 1;
TreeType     = 1;
% 下面计算可转换债券的价格
for j = 0:0.005:0.015;
StaticSpread = j;
     for i = 0:10:100
        Price = 40+i;
        [CbMatrix, UndMatrix, DebtMatrix, EqtyMatrix] = ...
         cbprice(RiskFreeRate, StaticSpread, Sigma, Price, ...
         ConvRatio, NumSteps, IssueDate, Settle, ...
         Maturity, CouponRate, Period, Basis, EndMonthRule, ...
         DividendType, DividendInfo, CallType, CallInfo, ...
         TreeType);
         convprice(i/10+1,j*200+1)  =  CbMatrix(1,1);
         stock(i/10+1,j*200+1)      =  Price;
       end
end
   plot(stock, convprice);
   legend({'+0 bp'; '+50 bp'; '+100 bp'; '+150 bp'});
```

```
title ('根据二叉树方法计算的有效价差-200 步')
xlabel('证券价格');
ylabel('可转换债券价格');
text(50, 150, ['票息为 4% 每半年支付一次.', sprintf('\n'), ...
    '2 年后按全价为 110 赎回.' sprintf('\n'), ...
    '5 年后到期.'],'fontweight','Bold')
```

图 4.3 是可转换债券价格与股票价格的关系示意图。

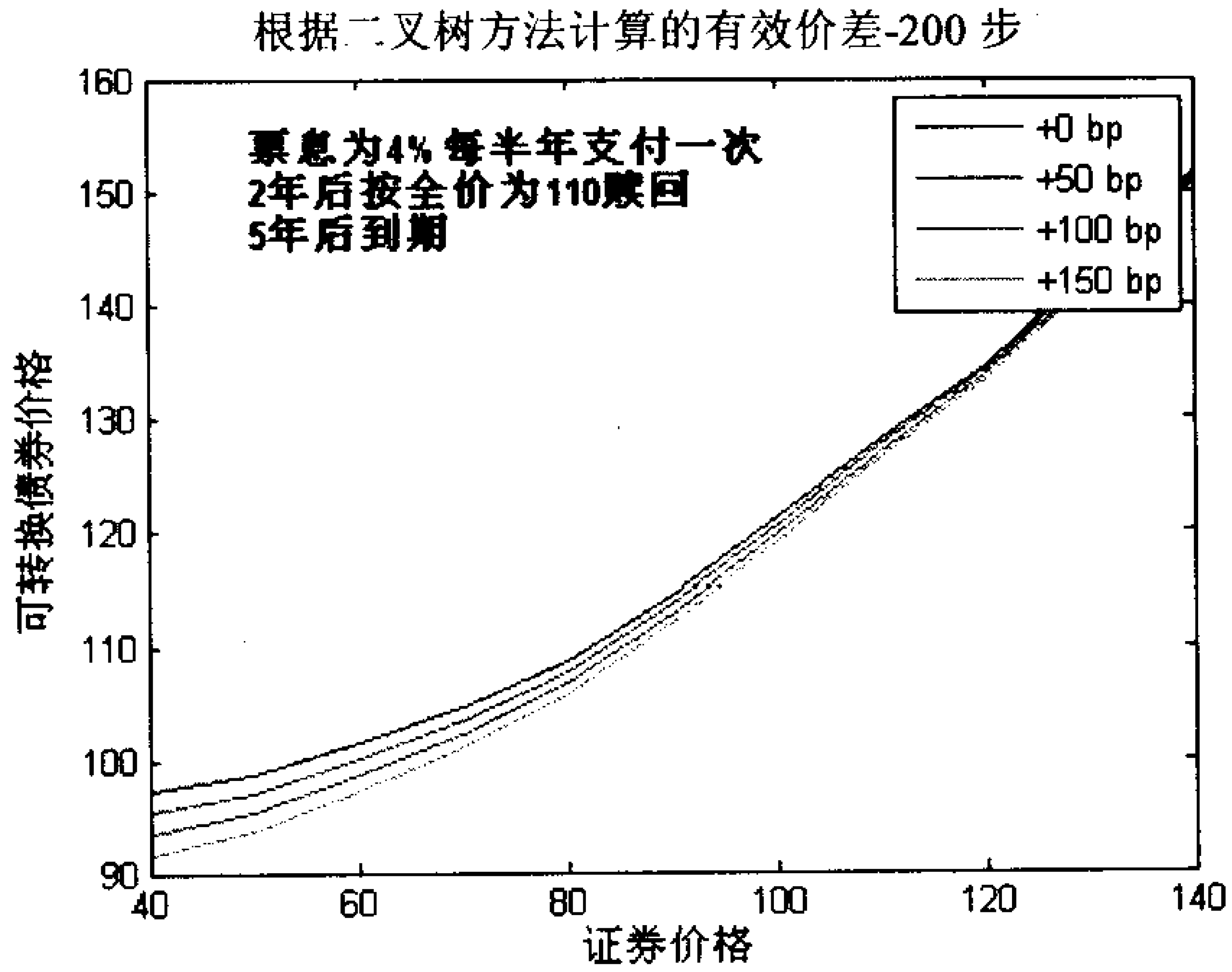


图 4.3　可转换债券价格与股票价格的关系图

1. 创建结构

MATLAB 金融工具箱中的函数输入变量很多，输入很不方便，为更加便于读者使用，这时就需要确定输入变量的格式，instaddfield 函数就是专门完成此项功能，该函数输入变量为“Type”，“FieldName” ，“Data”，其中“Type”定义一个新结构，“FieldName”是这个新结构的名称，“Data”是数据。此外还有一个可选项是“Classlist”。

2. 由利率生成贴现率

利率的结构如表 4.8 所示。

表 4.8　利率结构

开始时间	终止时间	利　率
15 Feb 2000	15 Aug 2000	0.05
15 Feb 2000	15 Feb 2001	0.056
15 Feb 2000	15 Aug 2001	0.06
15 Feb 2000	15 Feb 2002	0.065
15 Feb 2000	15 Aug 2002	0.075

若需要求出贴现率，这时可以利用 rate2disc 函数进行转化。

调用方式 1

```
disc=rate2disc(Compounding, Rates, EndTimes)
```

输入参数

Compounding	%一年中支付利息的次数，可以选 1，2，3，4，6，12。Compounding =2 表示一年支付 2 次利息，也就是半年支付一次利息。3 表示一年支付 3 次利息，平均 4 个月支付一次利息；4 表示每季度支付一次利息；6 表示平均 2 个月支付一次利息；12 表示每月支付一次利息；1 表示一年支付一次利息。365 表示每天支付一次利息
Rates	%年利率，用小数表示，如 5%写成 0.05
EndTimes	%结束时间，注意 EndTimes 的单位与 Compounding 一致。例如同样是 2 年 compounding=2 时，表示半年支付一次利息；2 年等于 4 个半年，EndTimes 就必须取 4compounding=4 时，表示一季度支付一次利息；2 年等于 8 个半年，EndTimes 就必须取 8

输出参数

disc	%贴现率

计算方式如下，F 表示一年中的计息频率，R 为利率，T 表示时间，$disc$ 表示贴现率，则

$$disc = 1\Big/\left(1+\frac{R}{F}\right)^T$$

【例 4-24】 一种债券年利率为 0.05，半年支付一次利息，到期日为 2 年，记其贴现率为 $disc$，则 MATLAB 中执行以下命令：

```
>> R=0.05;F=2;T=4;
>> disc=rate2disc(F, R, 4)
   disc =
          0.9060
```

计算出贴现率为 0.9060。

调用方式 2

```
Disc = rate2disc(Compounding, Rates, EndDates, StartDates,ValuationDate)
```

输入参数

Compounding	%一年中支付利息的频率
Rates	%年利率，以小数表示
EndDates	%债券到期日，输入日期型字符
StartDates	%债券开始计息日
ValuationDate	%评估日，默认值等于 StartDates 值

【例 4-25】如果一种债券，年利率为 5%，发行日为 15-Feb-2002，到期日为 15-Aug-2002，半年支付一次利息，需要计算 15-Feb-2002 到 15-Feb-2002 的贴现率，则其贴现率计算如下：

```
>> Compounding=2;
>> Rates=0.05;
>>EndDates='15-Aug-2002';  StartDates='15-Feb-2002';
ValuationDate='15-Feb-2002';
>> Disc = rate2disc(Compounding, Rates, EndDates, StartDates, ValuationDate)
  Disc =
       0.9756
```

贴现率为 0.9756。

如果需要计算 2002 年 4 月 15 日到 2002 年 8 月 15 日的贴现率，只需改动 ValuationDate 即可。

4.2.9 固定收益久期与凸度

久期是金融学中的一个重要概念，(Duration)是由佛雷德里希 • 麦考利(Federich Macaulay)和约翰 • 西克斯爵士(John Hicks)大致在同一时间发现的，当时麦考利的目标是找出一个可以用来比较期限相同但是支付结构不同的债券的方法，西克斯的目标是衡量债券对于利率风险敞口。

麦考利久期 *MD* 的计算公式如下：

$$MD = -\frac{\mathrm{d}P}{\mathrm{d}r}\frac{1+r}{P}$$

MATLAB 中计算久期的函数是 cfdur，计算公式如下：

$$D = \frac{PV(t_1)t_1 + PV(t_2)t_2 + \cdots + PV(t_n)t_n}{PV}$$

其中，$PV(t_i)$ 表示 t_i 时期现值。

业内人士通常使用的是修正久期(Modified Duration)，修正久期是在收益率改变而债券的预期现金流不变的情况下，收益率变化 1%时债券价格变化的百分比。修正久期的计算公式如下：

$$\text{修正久期} = \frac{\text{麦考利久期}}{1+\text{债券到期收益率}/k}$$

其中，k 表示每年支付利息次数。

调用方式

```
[Duration, ModDuration]=cfdur (CashFlow, Rate)
```

输入参数

```
CashFlow              %各期现金流
Rate                  %贴现率
```

输出参数

```
Duration                %久期
ModDuration             %修正久期
```

【例 4-26】一项投资各期现金流如表 4.9 所示。

表 4.9　各期投资现金流

第 1 期	第 2 期	第 3 期	第 4 期	第 5 期
1000	2000	3000	4000	5000

贴现率为 0.025，问该项投资久期是多少？

首先将各期现金流保存到变量 CashFlow，然后调用 cfdur 函数。

输入命令如下：

```
>>CashFlow=[1000 2000 3000 4000 5000]
CashFlow =
        1000         2000         3000         4000         5000
>> [Duration, ModDuration] = cfdur(CashFlow, 0.025)
Duration =
    3.6279
ModDuration =
    3.5394
```

从上面可以看出，久期是 8.9709，修正久期是 8.7521。

久期本质上是价值曲线在当前利率和债券价格点的斜率，凸性则是斜率的变化量。债券价格 P 随利率 r 的变化而变化，习惯上就可以把债券价格视为利率函数，利用泰勒展开得到

$$\Delta P \approx \frac{\mathrm{d}P}{\mathrm{d}r}\Delta r + \frac{1}{2}\frac{\mathrm{d}^2 P}{\mathrm{d}r^2}(\Delta r)^2 \tag{4.1}$$

式(4.1)两边同时除以价格 P，式(4.1)左边变成债券价格变化率，右边变成

$$\frac{\Delta P}{P} \approx \frac{\frac{\mathrm{d}P}{\mathrm{d}r}}{P}\Delta r + \frac{1}{2}\frac{\frac{\mathrm{d}^2 P}{\mathrm{d}r^2}}{P}(\Delta r)^2 \tag{4.2}$$

这样我们定义凸度 C 如下：

$$C = \frac{\frac{\mathrm{d}^2 P}{\mathrm{d}r^2}}{P} \tag{4.3}$$

则式(4.2)左边为债券价格变化率，右边关于 Δr 的一次项与二次项系数分别是久期与凸度，式(4.2)可以理解如下：

$$\text{债券价格变化}=-\text{久期}\times\text{债券价格}\times\text{债券收益率的变化}+\frac{1}{2}\text{凸度}\times\text{债券价格}\times\text{债券收益率的变化}^2$$

下面根据现金流贴现公式推导出债券凸度。设债券将来各期现金流分别为 $C_1,C_2,\cdots,C_T$，相应时间为 $t_1,t_2,t_3,\cdots,t_T$，债券贴现率为 r，由现值公式得出债券价格表达式为

$$P = \sum_{t=t_1}^{T} \frac{C_t}{(1+r)^t}$$

债券凸度为

$$C = -\frac{1}{P}\frac{d^2 P}{dr^2} = \frac{1}{P(1+r)^2}\left[\sum_{t=t_1}^{T} t(1+t)\frac{C_t}{(1+r)^t}\right]$$

可以证明麦考利久期与凸度之间的关系为

$$C = \frac{1}{(1+r)^2}[S + MD(MD+1)]$$

其中，$S = \sum_{t=t_1}^{T} \frac{c_t(1+r)^{-t}}{P}(t - MD)$，$MD$ 是麦考利久期，S 是衡量现金流的集中程度，当久期给定时，现金流越集中，则债券凸度越大，现金流越分散，则债券凸度越小。

债券资产组合的凸度是每个资产凸度的线性组合。设有 N 种债券，第 i 种债券的投资比重为 x_i，凸度为 C_i，则组合凸度 C_p 为

$$C_p = \sum_{i}^{N} x_i C_i$$

MATLAB 中计算凸度的函数有很多种，其中计算现金流凸度的函数为 cfconv。

调用方式

```
Convexity = cfconv(CashFlow, Yield)
```

输入参数

```
CashFlow        %各期现金流
Yield           %收益率
```

输出参数

```
Convexity       %凸度
```

【例 4-27】某金融产品现金流为连续 9 年支付 2.5，第 10 年还本付息，求该现金流

凸度。

在 MATLAB 中执行以下命令：

```
>> CashFlow = [2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 102.5];
>> Convex = cfconv(CashFlow, 0.025)
   Convex =
     90.4493
```

该现金流凸度为 90.4493。

4.3　利率期限结构

利率期限结构可以为债券定价。利率期限结构确定了各到期日货币的价格，可以为市场发行新金融工具寻找合适的价格。注意，利率期限结构不是债券赎回价格。

利率期限结构是未来利率水平的重要指标，代表着市场未来的利率预期，投资者通过利率期限曲线形状推测未来利率走势。当然解读利率期限结构既是一门科学更是一门艺术，债券投资者、基金经理、企业财务人员都会认真研究利率期限结构，中央银行、财政部也需要解读利率期限结构，结合通货膨胀率等其他信息来制定国家利率政策。

4.3.1　计算利率期限结构

利率期限结构理论主要讨论金融资产到期时收益与到期期限之间的关系。假设某一投资者打算作 N 年长期投资，它可以有两种选择，一种是购买在 N 年底到期的长期债券，并且持有到期满为止，另外一种是先持有 1 年期债券，到期后再购买 1 年期债券，如此不断直到 N 年期满为止。

实际上，任何债券都可以视为零息券的组合，付息债券可以视为一系列到期现金流的组合，债券价格就是这些现金流的现值。要对每一期现金流进行贴现，还需要知道贴现率，一般选择和该现金流日期相同的零息券，这一收益率称为即期利率(Spot Rate)。但是我们知道零息券大都为短期债券，所以我们一般构建国债即期收益率曲线(Spot Rate Curve)的，首先要选择以何种国债收益率曲线为基础，可供选择的国债有新发行国债、部分非新发行国债[①]、所有附息中长期国债和短期国债、零息券等。在计算利率期限结构时首先遇到的问题是美国短期债券与长期债券的标价方式是不同的[②]。

① 新发行的国债(On-the-Run)：在国债市场上拍卖的或者正在流通的国债。

② 美国债券市场上有长期国库券(treasure bonds)与短期国库券(treasure bills，也称 T-bills)，短期国库券的报价以银行折现率为基础。

1．已知债券收益率计算利率期限结构

MATLAB 中采用步步为营法(Boot Straping)计算利率期限结构，如我们计算 3 年期利率期限结构，假设平价收益率如表 4.10 所示。

表 4.10　平价收益率

年	0.5	1.0	1.5	2.0	2.5	3
到期收益率	4.0	4.2	4.6	5.6	5.8	7.0

一年以内债券采用折价销售，一年以上债券每半年支付一次利息，面值为 100 美元，债券票面利率就是到期收益率。

由于一年以内(含一年)的国库券属于零息券，因此年收益率等于即期收益率，下面给出 0.5～1.5 年的国库券现金流。

0.5 年：0.046×100×0.5=2.3

1.0 年：0.046×100×0.5=2.3

1.5 年：0.046×100×0.5+100=102.3

现金流量现值为

$$\frac{2.3}{\left(1+\frac{r_1}{2}\right)}+\frac{2.3}{\left(1+\frac{r_2}{2}\right)^2}+\frac{102.3}{\left(1+\frac{r_3}{2}\right)^3}$$

其中，r_1 为第 0.5 年即期利率，r_2 为第 1.0 年即期利率，r_3 为第 1.5 年即期利率。

由于一年以内的债券为零息券，所以 $r_1=0.04$，$r_2=0.042$，根据现金流现值等于面值得下列方程：

$$100=\frac{2.3}{\left(1+\frac{r_1}{2}\right)}+\frac{2.3}{\left(1+\frac{r_2}{2}\right)^2}+\frac{102.3}{\left(1+\frac{r_3}{2}\right)^3}$$

求解得 $r_3=4.62$。

有了 r_3 就可以求解第 2 年的即期利率。

2 年期国债现金流为

0.5 年：0.056×100×0.5=2.8

1.0 年：0.056×100×0.5=2.8

1.5 年：0.056×100×0.5=2.8

2.0 年：0.056×100×0.5+100=102.8

其现金流现值为

$$\frac{2.8}{\left(1+\frac{r_1}{2}\right)}+\frac{2.8}{\left(1+\frac{r_2}{2}\right)^2}+\frac{2.8}{\left(1+\frac{r_3}{2}\right)^3}+\frac{102.8}{\left(1+\frac{r_4}{2}\right)^4}$$

根据现金流现值等于面值 100，可得出下列方程：

$$\frac{2.8}{\left(1+\frac{r_1}{2}\right)}+\frac{2.8}{\left(1+\frac{r_2}{2}\right)^2}+\frac{2.8}{\left(1+\frac{r_3}{2}\right)^3}+\frac{102.8}{\left(1+\frac{r_4}{2}\right)^4}=100$$

由于 r_1, r_2, r_3 都是已知，根据上面方程求解得 $r_4 = 5.66$。以此类推可以计算出各时点上的即期利率，具体如表 4.11 所示。

表 4.11　各时点上的即期利率

年	0.5	1.0	1.5	2.0	2.5	3
到期收益率	4.0	4.2	4.62	5.66	5.86	7.18

在 MATLAB 中计算利率时间期限结构的函数为 zbtyield。

调用方式

```
[ZeroRates, CurveDates] = zbtyield(Bonds, Yields, Settle, Compounding)
```

输入参数

```
Bonds           %息票的时间、利率、面值
Yields          %债券息票的收益率
Settle          %结算日期
Compounding     %复利的计算方式
```

Compounding 参数具体见表 4.12。

表 4.12　Compounding 参数具体内容①

Compounding	1	2	3	4	6	12	-1
利息支付方式	每年支付一次	每半年支付	每3个月支付	每季度支付	每二月支付	每个月支付	复利

输出参数

```
ZeroRates       %期限结构上日期对应的利率
CurveDates      %期限结构的日期
```

① 本书的 Compounding 参数如无特别说明，其内容同本例。

【例 4-28】已知国债面值是 100 美元，各期收益率如表 4.13 所示。

表 4.13 国债各期收益率

国债品种	票　息	到 期 日	当前收益[①]
3 个月		17-Apr-03	1.15
6 个月		17-Jul-03	1.18
2 年	1.75	31-Dec-04	1.68
5 年	3.00	15-Nov-07	2.97
10 年	4.00	15-Nov-12	4.01
30 年	5.375	15-Feb-31	4.92

下面我们分析由上述品种构成的利率期限结构，代码如下：

```
>>% 首先我们需要输入债券特征
>> Bonds = [datenum('04/17/2003')  0       100;
         datenum('07/17/2003')   0       100;
         datenum('12/31/2004')   0.0175   100;
         datenum('11/15/2007')   0.03     100;
         datenum('11/15/2012')   0.04     100;
         datenum('02/15/2031')   0.05375  100];
>> Yields = [0.0115;
0.0118;
0.0168;
0.0297;
0.0401;
0.0492];
>> Settle = datenum('17-Jan-2003'); % 输入结算日
>> [ZeroRates, CurveDates] = zbtyield(Bonds, Yields, Settle)
ZeroRates =
   0.0115
   0.0118
   0.0168
   0.0302
   0.0418
   0.0550
CurveDates =
     731688
     731779
     732312
     733361
```

① 当前收益(Current Yield)等于票息/市场价格。

```
    735188
    741854
>> datestr(CurveDates) %将天数转换为日期
ans =
17-Apr-2003
17-Jul-2003
31-Dec-2004
15-Nov-2007
15-Nov-2012
15-Feb-2031
```

利率期限结构如表 4.14 所示。

表 4.14　国债利率期限结构

日期	03-4-17	03-7-17	04-12-31	07-11-5	12-11-15	31-2-15
利率	0.0115	0.0118	0.0168	0.0302	0.0418	0.0550

进一步地我们画出利率期限结构示意图，代码如下：

```
>> plot(ZeroRates)
```

得到的利率期限结构示意图如图 4.4 所示。

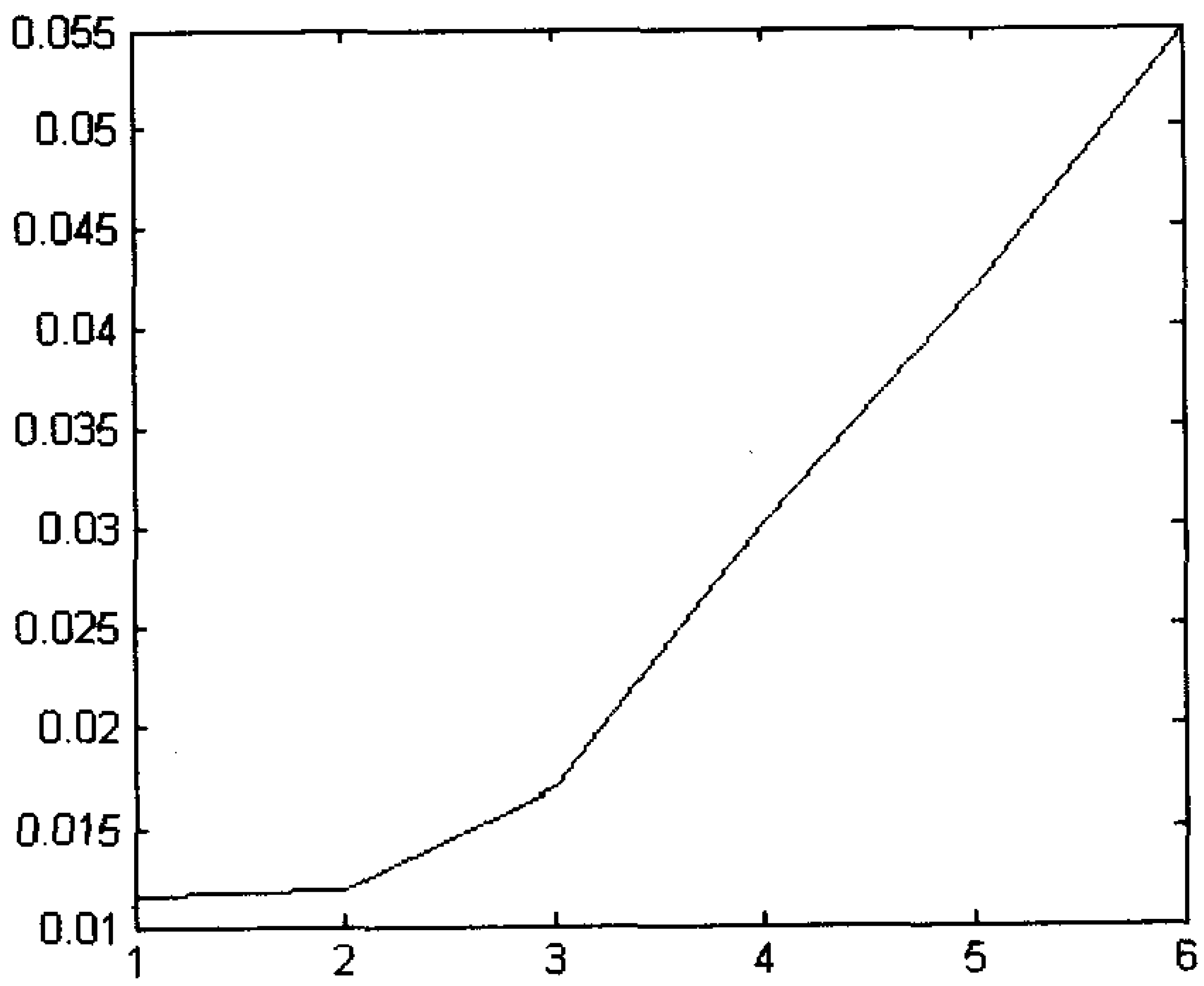


图 4.4　利率期限结构示意图

2. 给定债券定价格求其收益率

在 MATLAB 中给定债券价格，求其收益率函数是 zbtprice。

调用方式

```
[ZeroRates, CurveDates] = zbtprice(Bonds, Prices, Settle, OutputCompounding)
```

输入参数

```
Bonds              %Bonds 为一个向量，分别为债券的日期、利率、票面价值、期间、日期计算方
                    法、月末法则
Prices             %债券的当前价格
Settle             %结算日期
OutputCompounding   %利息的支付方式，可以取下面的值：
                   1    每年支付 1 次票息
                   2   (默认值)每半年支付 1 次票息
                   3    每 4 个月支付 1 次票息
                   4    每季度支付 1 次票息
                   6    每 2 月支付 1 次票息
                   12   每年支付 1 次票息
                   -1   表示连续复利
```

输出参数

```
ZeroRates          %各个时间的利率
CurveDates         %与利率对应的日期
```

【例 4-29】给定债券的起始日、利率、票面价值、价格如表 4.15 所示。

表 4.15 债券的起始日、利率、票面价值、结算日等

起 始 日	利 率	面 值	价 格	付息次数	日期方式	月末法则
6/1/1998	0.0475	100	99.375	2	0	0
7/1/2000	0.06	100	99.875	2	0	0
7/1/2000	0.09375	100	105.75	6	1	0
6/30/2001	0.05125	100	96.875	1	3	1
4/15/2002	0.07125	100	101.125	4	0	0
1/15/2000	0.08	100	103.125	2	3	0
9/1/1999	0.05875	100	99.375	3	0	0
4/30/2001	0.07125	100	101	2	0	0
11/15/1999	0.07	100	101.25	2	3	1
7/1/2001	0.0525	100	96.375	2	3	0
4/30/2002	0.07	100	102.75	2	0	0

债券结算日为 12/18/1997，利率按半年复利计算，求利率期限结构。首先给出各债券特征，代码如下：

```
>> Bonds = [datenum('6/1/1998')   0.0475   100  2  0  0;
            datenum('7/1/2000')   0.06     100  2  0  0;
```

```
          datenum('7/1/2000')    0.09375  100  6  1  0;
          datenum('6/30/2001')   0.05125  100  1  3  1;
          datenum('4/15/2002')   0.07125  100  4  1  0;
          datenum('1/15/2000')   0.065    100  2  0  0;
          datenum('9/1/1999')    0.08     100  3  3  0;
          datenum('4/30/2001')   0.05875  100  2  0  0;
          datenum('11/15/1999')  0.07125  100  2  0  0;
          datenum('6/30/2000')   0.07     100  2  3  1;
          datenum('7/1/2001')    0.0525   100  2  3  0;
          datenum('4/30/2002')   0.07     100  2  0  0];
```

把债券价格保存在 Prices 变量中，代码如下：

```
>> Prices = [99.375;
             99.875;
             105.75 ;
             96.875;
             103.625;
             101.125;
             103.125;
             99.375;
             101.0 ;
             101.25 ;
             96.375;
             102.75 ];
>> Settle = datenum('12/18/1997')
Settle =
     729742
>> OutputCompounding = 2;
>> [ZeroRates, CurveDates] = zbtprice(Bonds, Prices,
                   Settle,OutputCompounding)
ZeroRates =
    0.0616
    0.0609
    0.0658
    0.0590
    0.0648
    0.0655
    0.0606
    0.0601
    0.0642
    0.0621
    0.0627
CurveDates =
     729907
     730364
```

```
730439
730500
730667
730668
730971
731032
731033
731321
731336
```

4.3.2 计算特定时间利率

通常在市场中有很多债券，每种都有发行日与到期日，这时我们需要考虑在此期间中某个时间段的利率，例如 2000 年 2 月 15 日财政部发行分别于 2000 年 8 月 15 日、2001 年 2 月 15 日、2001 年 8 月 15 日、2002 年 2 月 15 日、2002 年 8 月 15 日到期的 5 种国债。如果投资者在 2002 年 2 月到 2002 年 8 月计划对流动资金进行融资，那他就需要关注 2002 年 2 月 15 日到 2002 年 8 月 15 日的利率期限结构，MATLAB 中的 ratetimes 函数可以实现这项功能。

函数 ratetimes 的输入参数分成两个部分，第一部分是说明原始利率期限结构，第二部分是给出新的日期，输出结果是对应于新日期的利率结构。

计算特定时间的利率期限结构，函数的调用方式根据时间格式不同有两种。

调用方式

```
Rates= ratetimes(Compounding, RefRates, RefEndTimes, RefStartTimes,
       EndTimes, StartTimes)
Rates= ratetimes(Compounding, RefRates, RefEndDates, RefStartDates,
       EndDates, StartDates, ValuationDate)
```

输入参数

```
Compounding      %每年支付利息的频率，可以选取 1、2、3、4、6、12、365 等
RefRates         %每个时间段的利率
RefEndTimes      %时间段的结束时刻
RefStartTimes    %时间段的开始时刻
EndTimes         %新时间段的结束时刻
StartTimes       %新时间段的结束时刻
RefEndDates      %时间段的结束日期
RefStartDates    %每个日期段的利率
EndDates         %时间段的结束日期
StartDates       %时间段的开始日期
ValuationDate    %利率的评估日
```

输出参数

```
Rates              %对应于新日期的利率结构
```

【例 4-30】如果已知原始利率期限结构如下：

开始日	结束日	利率
15Feb 2000	15Aug 2000	0.05
15Feb 2000	15 Feb 2001	0.056
15 Feb 2000	15 Aug 2001	0.06
15 Feb 2000	15 Feb 2002	0.065
15 Feb 2000	15 Aug 2002	0.075

现在需要研究对应于下面日期的利率：

开始日	结束日	利率
15 Feb 2000	15 Aug 2000	?
15 Aug 2000	15 Feb 2001	?
15 Feb 2001	15 Aug 2001	?
15 Aug 2001	15 Feb 2002	?
15 Feb 2002	15 Aug 2002	?

我们可以按照如下步骤进行。

首先输入原始利率期限结构，代码如下：

```
>> RefStartDates = ['15-Feb-2000'];
>> RefEndDates= ['15-Aug-2000'; '15-Feb-2001';
                 '15-Aug-2001';'15-Feb-2002'; …
                 '15-Aug-2002'];
>> Compounding = 2;
>> ValuationDate = ['15-Feb-2000'];
>> RefRates = [0.05; 0.056; 0.06; 0.065; 0.075];
```

然后输入新时间段，代码如下：

```
>> StartDates = ['15-Feb-2000'; '15-Aug-2000'; '15-Feb-2001';'15-Aug-2001';
                 '15-Feb-2002'];
>> EndDates= ['15-Aug-2000'; '15-Feb-2001'; '15-Aug-2001';'15-Feb-2002';
                 '15-Aug-2002'];
```

调用 ratetimes 函数计算对应于新时间段的利率，代码如下：

```
>> Rates = ratetimes(Compounding, RefRates, RefEndDates,RefStartDates,
                     EndDates,…StartDates, ValuationDate)
```

结果如下：

```
Rates =
```

```
0.0500
0.0620
0.0680
0.0801
0.1155
```

思 考 题

1. 了解固定收益证券计息天数的计算方法。
2. 编写程序计算固定收益久期与凸度。
3. 计算下列固定收益利率期限结构，面值都为100。

短期国债		到 期 日	当前收益①
3个月		17-Apr-03	1.13
6个月		17-Jul-03	1.17
中长期国债	票 息	到 期 日	当前收益
2年	1.75	31-Dec-04	1.58
5年	3.00	15-Nov-07	2.67
10年	4.00	15-Nov-12	4.01
30年	5.375	15-Feb-31	4.92

用图画出利率期限结构示意图。

4. 本章中静态利差可以用牛顿迭代法进行数值求解，试编写出程序。
5. 试分析国债回购利率与上证指数之间的相关系数。

① 当前收益(current yield)等于票息/市场价格。

第 5 章　资产组合计算

资产组合是实务性比较强的内容，通过本章的学习，要求读者掌握协方差与相关系数之间的相互推导，熟悉资产组合基本理论，学会用 MATLAB 计算投资组合基本参数，如均值与方差、资产组合 Var，重点掌握资产组合有效前沿的计算，能够处理无风险利率以及借贷关系情况下的最优投资组合，会用 MATLAB 规划工具箱求解投资组合最优化问题。

5.1　资产组合基本原理

证券投资组合理论(Portflio Theory)主要研究如何配置各种不同的金融资产，实现资产组合的最佳投资配置。1952 年美国学者马克维茨创立了资产组合理论，该理论在实践中得到广泛运用。

5.1.1　收益率序列与价格序列间的转换

1．将收益率序列转换为价格序列

在处理金融时间序列时，有时需要把收益率序列转换为价格序列。在 MATLAB 中将收益率序列转换为价格序列的函数是 ret2tick。

调用方式

```
[TickSeries, TickTimes] = ret2tick(RetSeries, StartPrice, RetIntervals,
StartTime,Method)
```

输入参数

```
RetSeries        %收益率序列
StartPrice       %(Optional)起始价格，默认值是 1
RetIntervals     %(Optional)收益率序列的时间间隔，默认值是 1
StartTime        %(Optional)价格开始计算的时间，默认值是 0
Method           %(Optional)转换方法。Method='Simple'表示简单方法，
```

$P_{t+1}=P_t(1+r_{t+1})$；Method='Continuous'表示连续法，$P_{t+1}=P_t e^{r_{t+1}}$

输出参数

```
TickSeries       %价格序列
TickTimes        %与价格对应的时间序列
```

【例 5-1】已知资产收益率以及时间间隔如表 5.1 所示。

表 5.1　资产收益率及时间

收益率	0.10	0.05	−0.05
时间间隔(天)	182	91	92

起始价格为 10 元，起始时间为 2000 年 12 月 18 日，试求该资产价格时间序列，收益率采用离散方法。

在 MATLAB 中执行以下命令：

```
>> RetSeries = [0.10  0.05  -0.05]';
>> RetIntervals = [182  91  92]';
>> StartPrice=10;
>> StartTime = datenum('18-Dec-2000');
>> [TickSeries,TickTimes] =
ret2tick(RetSeries,StartPrice,RetIntervals,StartTime)
TickSeries =
   10.0000
   11.0000
   11.5500
   10.9725
TickTimes =
      730838
      731020
      731111
      731203
>> datestr(TickTimes) %将日期数转换为日期
ans =
18-Dec-2000
18-Jun-2001
17-Sep-2001
18-Dec-2001
```

这样就把收益率时间序列转换为价格时间序列，结果如表 5.2 所示。

表 5.2　资产各时间的价格

时间	18-Dec-2000	18-Jun-2001	17-Sep-2001	18-Dec-2001
价格	10.0000	11.0000	11.5500	10.9725
收益率	—	0.10	0.05	−0.05
时间间隔	—	182	91	92

2. 将价格序列转换为收益率序列

MATLAB 中将价格序列转换为收益率序列的函数是 tick2ret。

调用方式

```
[RetSeries, RetIntervals] = tick2ret(TickSeries, TickTimes, Method)
```

输入参数

```
TickSeries       %价格序列
TickTimes        %价格序列对应的时间
Method           %计算收益率的方法，转换方法同上，取值“Simple”时计算方法为算术收益
                  率， 取值 “Continuous” 时为对数计算法
```

输出参数

```
RetSeries        %收益率序列
RetIntervals     %收益率时间间隔
```

【例 5-2】 已知股票的价格时间序列如表 5.3 所示。

表 5.3 股票各时间对应的价格

时间	0	6	9	12
价格	100	110	115	110

求出该股票的收益率时间序列。

在 MATLAB 中执行以下命令：

```
>> TickSeries = [100;110;115;110];
>> TickTimes = [0; 6;  9; 12];
>> [RetSeries, RetIntervals] = tick2ret(TickSeries, TickTimes)
RetSeries =
    0.1000
    0.0455
   -0.0435
RetIntervals =
     6
     3
     3
```

计算时间间隔和收益率的结果如表 5.4 所示。

表 5.4 转换后的收益率

时间间隔	6	3	3
收益率	0.1000	0.0455	-0.0435

5.1.2 协方差矩阵与相关系数矩阵间的转换

MATLAB 中的 corr2cov 函数可以把相关系数矩阵转换为协方差矩阵。

调用方式

```
Covariances = corr2cov(STDs, Correlations)
```

输入参数

```
STDs            %标准差矩阵
Correlations    %相关系数矩阵
```

输出参数

```
Covariances     %协方差矩阵
```

【例 5-3】已知资产组合中有 3 个品种，每个品种的资产收益率、标准差和相关系数如表 5.5 所示。

表 5.5 各资产预期回报、权重和相关系数

		资 产 A	资 产 B	资 产 C
预期回报		0.1	0.15	0.12
相关系数矩阵	资产 A	1	0.8	0.4
	资产 B	0.8	1	0.3
	资产 C	0.4	0.3	1

求该资产的协方差矩阵。

在 MATLAB 中执行以下命令：

```
>> Returns=[0.1 0.15 0.12];
>> STDs= [0.2 0.25 0.18];
>> Correlations=[ 1   0.8  0.4
                  0.8   1   0.3
                  0.4  0.3   1 ];
>> Covariances=corr2cov(STDs, Correlations)
Covariances=
    0.0400    0.0400    0.0144
    0.0400    0.0625    0.0135
    0.0144    0.0135    0.0324
```

5.1.3　资产组合收益率与方差

MATLAB 中计算资产组合回报与方差的函数是 portstats。

调用方式

```
[PortRisk, PortReturn] = portstats(ExpReturn, ExpCovariance, PortWts)
```

输入参数

```
ExpReturn        %期望收益向量
ExpCovariance    %资产的协方差矩阵
PortWts          %资产权重向量
```

输出参数

```
PortRisk         %总资产的标准差
PortReturn       %总资产的收益
```

【例 5-4】某资产组合中有 3 种资产 A、B、C，组合中各资产的预期收益率分别为 0.1、0.2、0.15，权重分别为 0.4、0.2、0.4，具体如表 5.6 所示。

表 5.6　资产组合中各资产的明细表

		资 产 A	资 产 B	资 产 C
协	资产 A	0.0100	−0.0061	0.0042
方	资产 B	−0.0061	0.0400	−0.0252
差	资产 C	0.0042	−0.0252	0.0225
预期回报		0.1	0.2	0.15
资产 1 各资产权重		0.4	0.2	0.4
资产 2 各资产权重		0.2	0.4	0.4

在 MATLAB 中执行如下命令：

```
>> ExpReturn = [0.1 0.2 0.15];
>> ExpCovariance = [0.0100   -0.0061    0.0042
             -0.0061    0.0400   -0.0252
              0.0042   -0.0252    0.0225 ];
>> PortWts=[0.4 0.2 0.4; 0.2 0.4 0.2];
>> [PortRisk, PortReturn] = portstats(ExpReturn, ExpCovariance,PortWts)
PortRisk =
    0.0560
    0.0451
```

```
PortReturn =
    0.1400
    0.1600
```

从上述结果可以看到，这两个资产组合的标准差分别为 0.056、0.0451，资产回报分别为 0.14、0.16。

【例 5-5】假设资产组合中有 5 种资产，收益分别为 0.1、0.12、0.14、0.16、0.2，方差分别为 0.02、0.03、0.01、0.05、0.02，资产收益率各不相关，各资产权重分别为 0.1、0.2、0.3、0.2、 0.2，计算该组合的收益率与方差。

在 MATLAB 中执行以下命令：

```
>> returns=[0.1 0.12 0.14 0.16 0.2]; variances=[0.02 0.03 0.01 0.05 0.02];
>>ws=[0.1 0.2 0.3 0.2 0.2];
>> mean=sum(returns.*ws)
   mean =0.1480
>> variance=sum(variances.*ws.^2)
   variance =
    0.0051
```

该资产组合的收益率为 0.1480，方差为 0.0051。

5.1.4 资产组合 VaR(Value At Risk)

一般被称为“风险价值”或“在险价值”，指在一定的置信水平下，某一金融资产(或证券组合)在未来特定的一段时间内的最大可能损失。假定 JP 摩根公司在 2004 年置信水平为 95%的日 VaR 值为 960 万美元，其含义指该公司可以以 95%的把握保证，2004 年某一特定时点上的金融资产在未来 24 小时内，由于市场价格变动带来的损失不会超过 960 万美元。或者说，只有 5%的可能损失超过 960 万美元。与传统风险度量手段不同，VaR 完全是基于统计分析基础上的风险度量技术，它的产生是 JP 摩根公司用来计算市场风险的产物。

【例 5-6】假设投资者拥有两种资产，资产总价值为 10 000 000 元，资产权重分别为 1/4 与 3/4，这两种资产日波动率的均值分别为 0.003、0.002，标准差分别为 0.02、0.01，这两种资产之间的相关系数为 0.8，时间为 10 天，给定置信度为 99%，求该资产 Var。

首先求总资产方差，公式如下：

$$\bar{\sigma}^2=[w_1\sigma_1 \quad w_2\sigma_2]\begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\begin{bmatrix}w_1\sigma_1\\ w_2\sigma_2\end{bmatrix} \tag{5.1}$$

其中，w_1,w_2 分别为资产组合权重，σ_1,σ_2 为单个资产标准差，ρ 为这两种资产之间的相关系数。

一般地，可将式(5.1)用向量与矩阵形式表示，记 $w=[w_1 \quad w_2 \quad \cdots \quad w_n]$，表示各种资产的权重，$\sigma=[\sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_n]$ 表示各种资产的标准差，资产协方差矩阵记为 cov，则式(5.1)

可以改写为如下形式：

$$\bar{\sigma} = (w \cdot \sigma)\mathrm{cov}(w,\sigma)^{\tau} \tag{5.2}$$

其中“·”为向量点乘符号，表示两个向量对应元素的乘积，MATLAB 中用记号“.*”表示向量点乘。记号“τ”表示向量转置。如果记$\hat{\sigma} = w \cdot \sigma$，则有

$$\bar{\sigma}^2 = \hat{\sigma}\mathrm{cov}(\hat{\sigma})^{\tau}\hat{\sigma}^{\tau}\delta t \tag{5.3}$$

有了资产组合方差，就可以计算出 Var 数值。从正态分布表中可以查到对应于置信度 99%($\alpha = 0.01$)的 $Z_{1-\alpha} = 2.3263$，在各种资产都是服从正态分布的假设下，资产 Var 值为

$$Var = 总资产 \times (\mu \times \delta t - Z_{1-\alpha} \times \sqrt{\delta t} \times \bar{\sigma}) \tag{5.4}$$

具体来讲，计算 Var 的步骤如下。

第 1 步：输入资产权重向量w、各资产的标准差 sigma、资产之间的相关系数 cov，注意协方差矩阵一定是对称矩阵，需要计算时间长度δt。

第 2 步：权重向量点乘标准差向量。

第 3 步：计算资产总的标准差$\bar{\sigma}$。

第 4 步：对于给定置信度α，查正态分布表找到$Z_{1-\alpha}$。

第 5 步：计算 Var，$Var = 总资产 \times (\mu \times \delta t - Z_{1-\alpha} \times \sqrt{\delta t} \times \bar{\sigma})$。

在 Command 窗口中执行如下命令：

```
>> w=[1/4 3/4];ret=[0.003 0.002];sigma=[0.02
0.01];corrcoef=[1,0.8;0.8,1];deltat=10;%w 为权重向量，sigma 为资产方差，deltat
为时间长度，ret 为各资产回报，corrcoef 为相关系数矩阵
>> pret=deltat*dot(w,ret)  % 资产组合回报
>> sig=w.*sigma;
>> tsig=sig*cov*sig'*deltat;
>> var=10^7*(pret-2.3263*sqrt(tsig))   %计算 Var 值
var =
 -6.4930e+005
```

10 天 Var 值为 649 300 元。

实际上 MATLAB 中有专门计算 Var 值的函数，MATLAB 中的 portvrisk 函数可以计算资产组合 Var 值，注意输入总资产期望收益与标准差，而不是组合中各种资产的预期收益率与标准差。

调用方式

```
ValueAtRisk = portvrisk(PortReturn, PortRisk, RiskThreshold, PortValue)
```

输入参数

```
PortReturn         %总资产的回报
```

```
PortRisk          %总资产的标准差
RiskThreshold     %概率阈值，默认值为 0.05
PortValue         %资产总的价值
```

输出参数

```
ValueAtRisk      %概率阈值下的单资产 Var 值
```

【例 5-7】已知资产年回报率为 0.0029，标准差为 0.0308，资产现在价值为 1 亿，求 1% 水平下资产在险价值。

在 MATLAB 中执行以下命令：

```
>> PortReturn = 0.0029;      %资产组合的收益
>>PortRisk = 0.0308;         %资产组合的标准差
>>RiskThreshold = 0.01;      %1%水平下损失的概率
>>PortValue = 1;
>>ValueAtRisk = portvrisk(PortReturn,PortRisk,RiskThreshold,PortValue)
ValueAtRisk =0.0688
```

该资产 Var 等于 0.0688，即该资产损失 0.0688 亿的可能性为 1%。需注意的是金融资产一般并不是正态，而是呈现出肥尾特征，其 Var 较正态分布大。

5.2 资产组合有效前沿

由于证券市场投资存在巨大风险，一般不主张把投资集中在一种产品上。如果一个投资者投资于深证东泰股份(000506)，2001 年 8 月 10 日收盘价为 14.10 元，到了 2006 年 2 月 21 日收盘价为 1.54 元，跌幅高达 89.08%，如果再要回到原来价位需要上涨 9.15 倍，这样的机会是几乎不可能的，如果投资者踩中这样的陷阱恐怕很难再有翻身的机会。

运用组合理论可以有效地降低投资风险，其核心思想是在目标收益率给定的情况下，要求资产组合风险最小。资产组合理论是由马克维茨(H.Markowitz)1952 年提出均值方差理论模型

$$\min:\sigma^2 = x^T V x = \sum_{i,j} x_i x_j \sigma_{i,j}$$

$$\text{s.t}\quad E(r_x) = x^T E(r) = \sum x_i E(r_i);\ \sum x_i = 1$$

其中，V 是协方差矩阵 $V=(\sigma_{i,j})$，r_i 表示第 i 种资产的收益率，x_i 表示第 i 种资产在总资产中所占的份额。

5.2.1　两种风险资产组合收益期望与方差

假设有两种资产 A、B，其收益率分别用 R_1、R_2 表示，协方差分别为 $V=(\sigma_{i,j})$。记资产组合为 P，资产组合收益率、方差分别为 R_P、σ_P^2，x_1、x_2 分别表示投资权重，则有

$$R_P = x_1 R_1 + x_2 R_2$$

该资产组合期望收益率与方差为

$$E(R_P) = x_1 E(R_1) + x_2 E(R_2)$$

$$\sigma_P^2 = (x_1, x_2)\begin{pmatrix} \sigma_1^2 & \sigma_{1,2} \\ \sigma_{1,2} & \sigma_2^2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1^2\sigma_1^2 + 2x_1x_2\sigma_{1,2} + x_2^2\sigma_2^2$$

这样资产组合收益率均值与方差如图 5.1 所示。

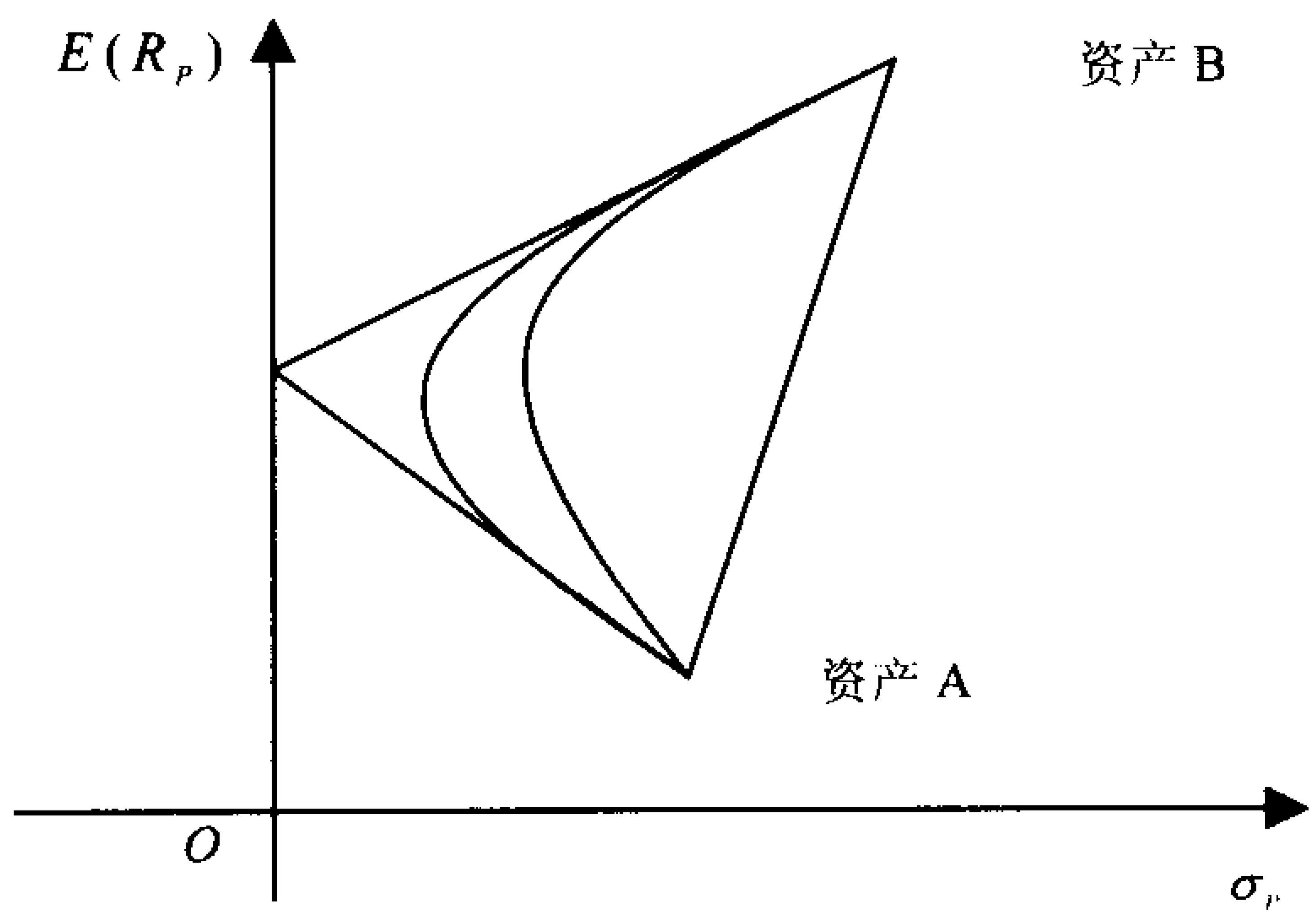


图 5.1　资产组合收益率均值与方差

MATLAB 工具箱中包含了资产均值方差有效前沿函数，这些都是基于 MATLAB 中的最优化理论工具箱。

马克维茨资产组合理论就是寻找一个有效组合，所谓有效组合是指在同样风险水平下具有最高收益，这样不同收益及与最小风险构成有效前沿。

在不允许卖空情况下，求解有效组合目标函数为

$$\min \frac{1}{2} x V x^T$$

$$\text{s.t } x^T \overline{R} = \overline{R}_P \text{；} \quad x_1 + x_2 = 1 \text{，} \quad x_1, x_2 \geqslant 0$$

这是一个约束条件为线性且含有不等式的二次规划方程，给定一个组合收益率就有一个最小方差，组合收益与最小方差构成有效前沿关系，有效前沿如图 5.2 所示。

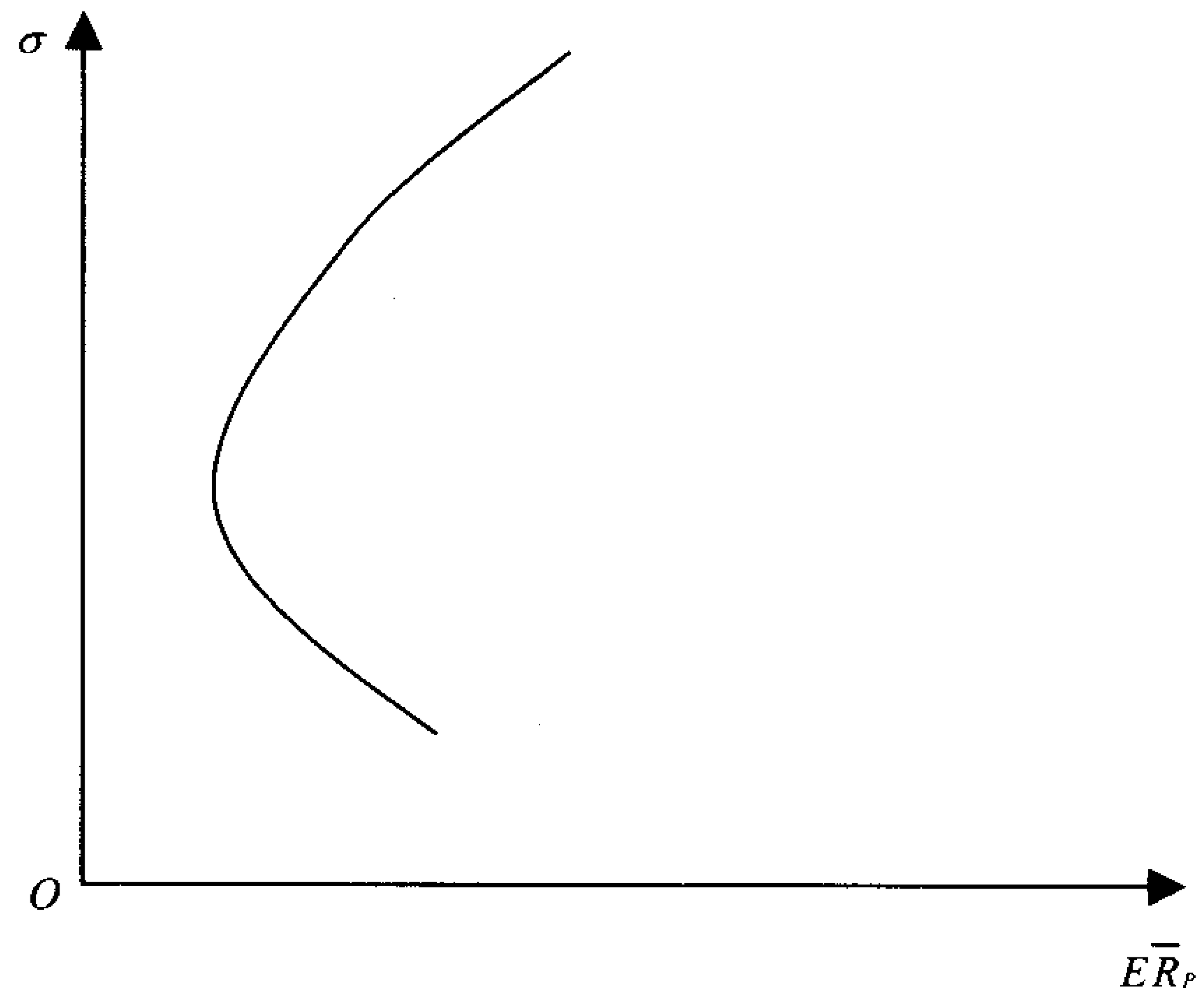


图 5.2 资产组合有效前沿示意图

5.2.2 均值方差有效前沿

MATLAB 中计算均值方差有效前沿的函数为 frontcon。

调用方式

```
[PortRisk, PortReturn, PortWts] =frontcon(ExpReturn, ExpCovariance,
NumPorts, PortReturn, AssetBounds, Groups, GroupBounds)
```

输入参数

```
ExpReturn       %资产组合中每项资产预期回报，为一列行向量
ExpCovariance   %各种资产之间的协方差矩阵，为对称矩阵
NumPorts        %(Optional)在资产组合有效前沿上的点的个数，默认值是 10 个点
PortReturn      %(Optional)有效前沿上每个点的回报
AssetBounds     %(Optional)每种资产权重的上限、下限区间
Groups          %(Optional)如果 G(i,j)=1 表示第 i 个资产属于第 j 个群 G(i,j)=0 表示
                 第 i 个资产不属于第 j 个群
GroupBounds     %(Optional)每种群权重约束区间，默认值规定下限为 0，上限为 1
```

输出参数

```
PortRisk        %组合的标准差
PortReturn      %组合的回报
PortWts         %组合中每个资产的权重
```

【例 5-8】考虑一个三资产组合，分别为资产 1、资产 2 与资产 3，其预期收益率分别为 0.2、0.1、0.15，资产协方差矩阵如表 5.7 所示，求该资产组合有效前沿。

表 5.7　资产协方差阵

	资产 1	资产 2	资产 3
资产 1	0.01	−0.0061	0.0042
资产 2	−0.0061	0.04	−0.0252
资产 3	0.0042	−0.0252	0.0225

在 MATLAB 中执行以下命令：

```
>> ExpReturn = [0.1 0.2 0.15];
>> ExpCovariance = [ 0.0100   -0.0061    0.0042
-0.0061   0.0400   -0.0252
0.0042   -0.0252    0.0225];
>> NumPorts = 4; %资产组合有效前沿上的 4 个点
>> [PortRisk, PortReturn, PortWts] = frontcon(ExpReturn, ExpCovariance,
NumPorts)
PortRisk =
    0.0426
    0.0483
    0.1089
    0.2000
PortReturn =
    0.1569
    0.1713
    0.1856
    0.2000
PortWts =
    0.2134    0.3518    0.4348
    0.0096    0.4352    0.5552
         0    0.7128    0.2872
         0    1.0000         0
```

5.2.3　带约束条件资产组合有效前沿

投资组合中的问题很少有简单的约束，大多数情况下是多种约束，例如监管当局为了控制风险，对资产组合中每种资产的比例加以种种限制，这时就需要考虑多种约束条件下的最优组合问题。

MATLAB 利用均值-方差理论求解资产组合问题，首先是将约束条件写成矩阵形式，例如 $Ax \leqslant b$，或者 $Ax = b$ 形式，下面我们用一个例子说明。

【例 5-9】某资产组合中有 5 种资产构成，第 i 种资产的预期回报率为 $r_i(i = 1,2,3,4,5)$，w_i 为第 i 种资产在总资产中的权重，考虑 w_i 具有如下形式：

$$0 \leqslant w_1 \leqslant 0.35;\quad 0 \leqslant w_2 \leqslant 0.3;\quad 0 \leqslant w_3 \leqslant 0.3$$
$$0 \leqslant w_4 \leqslant 0.4;\quad 0 \leqslant w_5 \leqslant 0.35$$
$$0.2 \leqslant w_1 + w_2 \leqslant 0.6$$
$$0.3 \leqslant w_3 + w_4 + w_5 \leqslant 0.7$$

上述约束条件写成矩阵形式如下：

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0.35 \\ 0 & 1 & 0 & 0 & 0 & 0.30 \\ 0 & 0 & 1 & 0 & 0 & 0.30 \\ 0 & 0 & 0 & 1 & 0 & 0.40 \\ 0 & 0 & 0 & 0 & 1 & 0.5 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & 0 & -0.2 \\ 0 & 0 & -1 & -1 & -1 & -0.3 \\ 1 & 1 & 0 & 0 & 0 & 0.60 \\ 0 & 0 & 1 & 1 & 1 & 0.70 \end{pmatrix}$$

注意约束条件 $0 \leqslant w_1 \leqslant 0.35$ 可以分解成两个约束条件：$0 \leqslant w_1$ (可以写成约束条件 $-w_1 \leqslant 0$)和 $w_1 \leqslant 0.35$，分别对应于矩阵的第 1 行和第 6 行。下面我们计算约束条件下资产组合有效前沿。

调用方式

```
[PortRisk, PortReturn, PortWts] = portopt(ExpReturn, ExpCovariance,
NumPorts, PortReturn, ConSet)
```

输入参数

```
ExpReturn          %资产的期望回报率
ExpCovariance      %资产的协方差
NumPorts           %(Optional) 资产组合中投资品种的个数
PortReturn         %(Optional) 要求组合的回报率
ConSet             %(Optional) 约束条件
```

输出参数

```
PortRisk           %资产组合的风险
```

```
PortReturn          %资产组合的回报
PortWts             %组合中各种资产的权重
```

【例 5-10】设有两种资产，其回报率分别为 0.1、0.3，协方差矩阵为$\begin{pmatrix} 0.02 & 0 \\ 0 & 0.04 \end{pmatrix}$，约束条件如下：

$$x_1 + x_2 = 1$$
$$0 \leqslant x_1 \leqslant 0.2$$
$$0.3 \leqslant x_2$$

求该资产组合有效前沿。

在 MATLAB 中执行如下命令：

```
>>ret=[0.1 0.3];cov=[0.02 0;0 0.04];
>> constr=[1 1 1;1 0 0.2;-1 0 0 ;0 -1 -0.3]; %约束矩阵
>> portopt(ret,cov,[],[],constr)
```

则满足上述条件的资产组合有效前沿如图 5.3 所示。

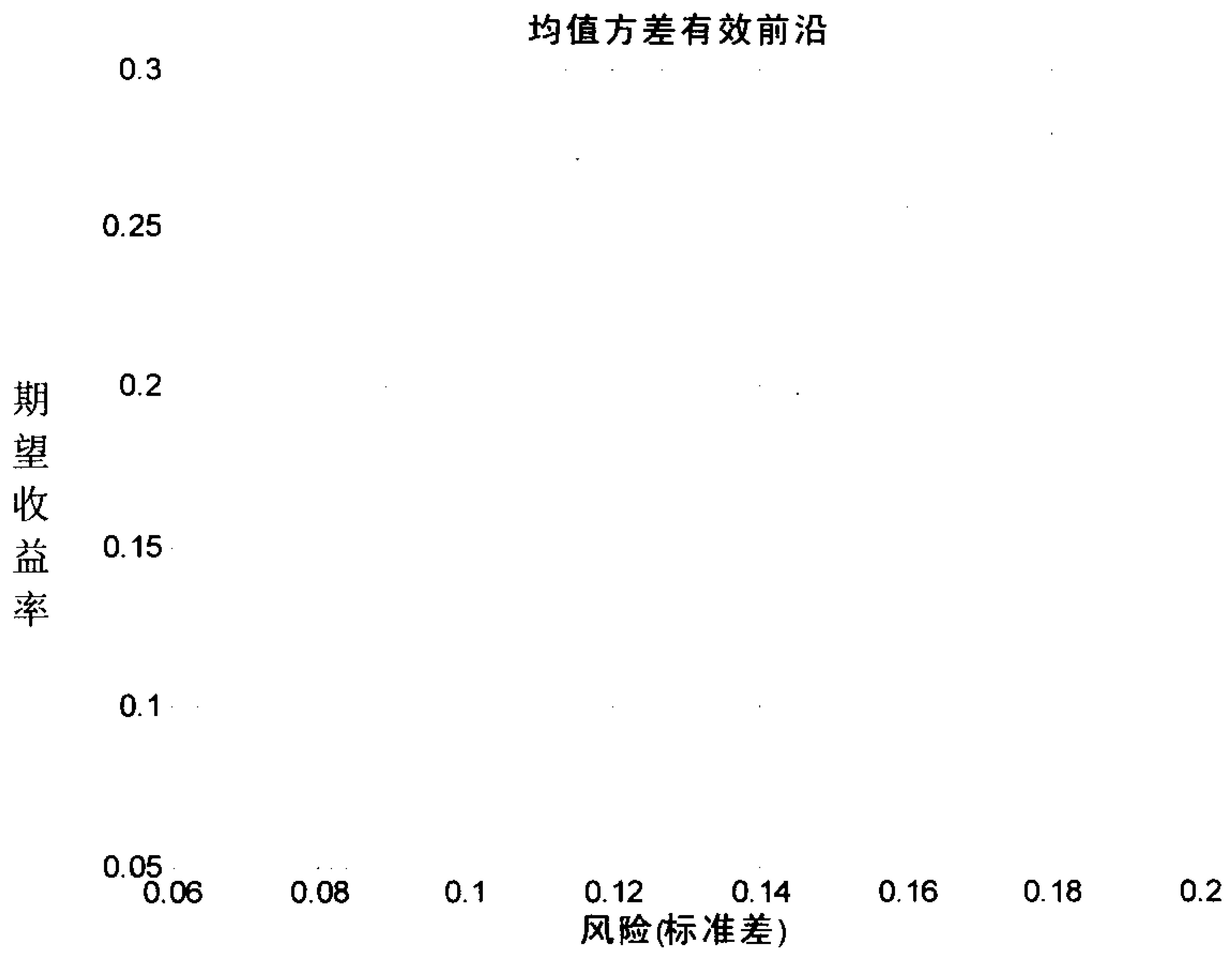


图 5.3　含约束条件均值方差有效前沿示意图

【例 5-11】各资产的相关系数矩阵、预期回报和标准差如表 5.8 所示。

表 5.8 各资产的相关系数矩阵、预期回报和标准差

		资 产 A	资 产 B	资 产 C
相关系数矩阵	资产 A	1	0.8	0.4
	资产 B	0.8	1	0.3
	资产 C	0.4	0.3	1
预期回报		0.1	0.15	0.20
各资产标准差		0.2	0.25	0.18

试给出有效前沿。

在 MATLAB 中执行如下命令：

```
>> Returns      = [0.1 0.15 0.12];
>> STDs        = [0.2 0.25 0.18];
>> Correlations = [ 1   0.8    0.4
                0.8   1   0.3
                0.4  0.3   1 ];
>> Covariances = corr2cov(STDs, Correlations); % 相关系数矩阵转换为协方差矩阵
>>portopt(Returns, Covariances, 20)  % 绘出组合的有效前沿
>>% 然后选择权重
>> rand('state', 0);
>> Weights = rand(1000, 3);       % 产生1000行×3列随机数
>> Total = sum(Weights, 2);       % Weights中的数据沿对列求和①
>> Weights(:,1) = Weights(:,1)./Total;
>> Weights(:,2) = Weights(:,2)./Total;
>> Weights(:,3) = Weights(:,3)./Total; % 这时Weights变成了权重矩阵
```

输入资产组合有效前沿，以及相关资产组合，绘出各个资产组合风险与收益，代码如下：

```
>> [PortRisk, PortReturn] = portstats(Returns, Covariances, Weights);
>>hold on
>>plot (PortRisk, PortReturn, '.r')
>>title('均值一方差有效前沿以及各个资产组合风险与收益。')
>> xlabel('风  险(标准差)')
>> ylabel('期望收益率')
```

---

① 将数据矩阵的每列相加，生成一个新的列向量，即 $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} \end{pmatrix} \Rightarrow \begin{pmatrix} a_{11}+a_{12}+a_{13} \\ a_{21}+a_{22}+a_{23} \\ a_{31}+a_{32}+a_{33} \\ \vdots \\ a_{n1}+a_{n2}+a_{n3} \end{pmatrix}$

```
>>hold off
```

这样资产组合有效前沿和各种资产组合风险与收益点如图 5.4 所示。

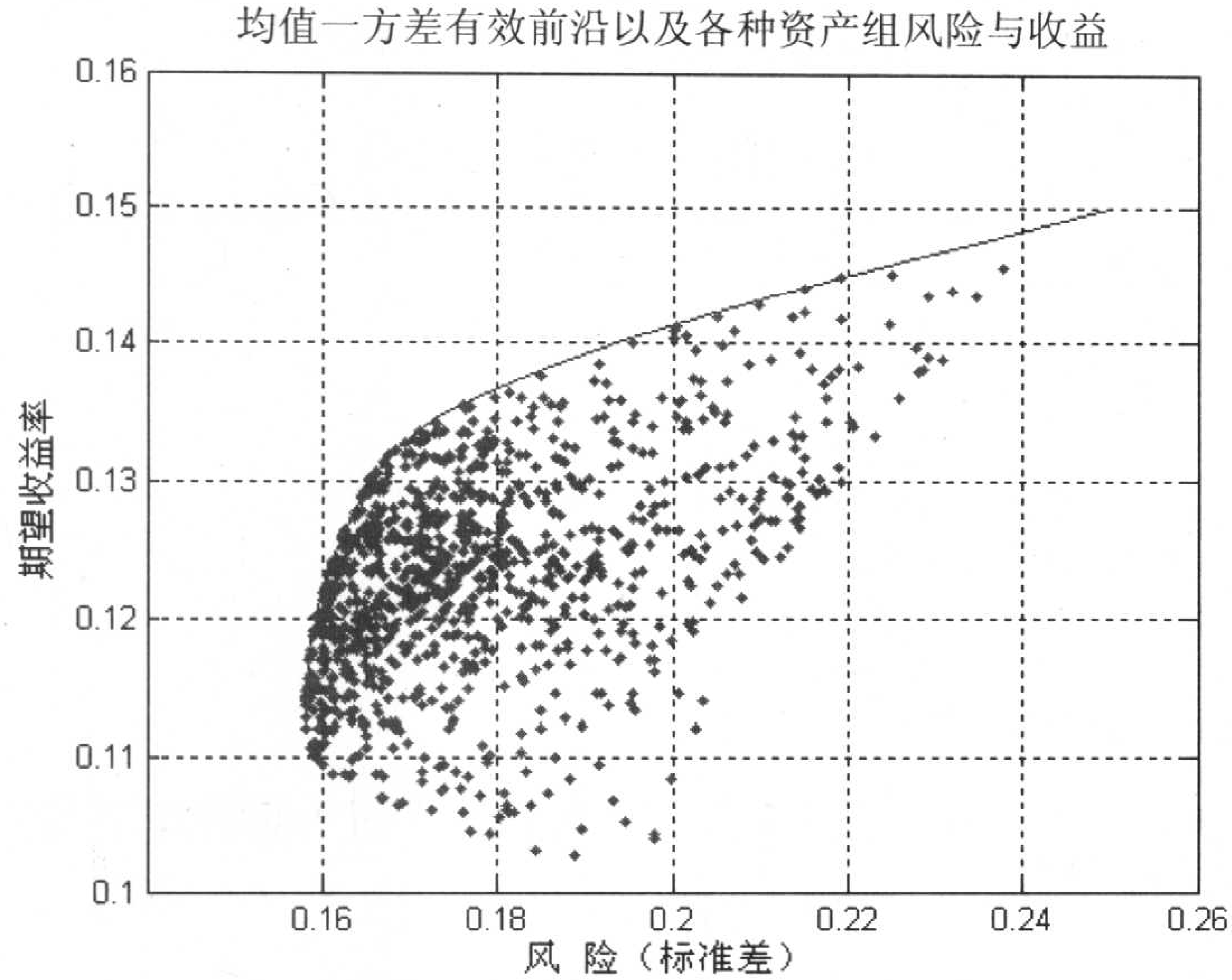


图 5.4　资产组合有效前沿及各个组合的风险与收益

5.2.4　考虑无风险资产及借贷情况下的资产配置

资产组合有效前沿上的点很多，如何选择一个有效点呢？投资者需要根据目标函数权衡风险与回报。MATLAB 中投资者目标函数如下：

$$U = E(r) - 0.5 \times A \times \sigma^2$$

其中，$E(r)$ 表示未来回报，A 表示投资者风险厌恶系数，一般在 2～4 之间，σ 是资产标准差。

投资者决策就是使目标函数最大化，然后对资产进行配置。MATLAB 中考虑无风险资产时的资产配置函数是 portalloc，其功能是根据风险-收益最优原则配置每项资产，其中包括无风险资产。

调用方式

```
[RiskyRisk, RiskyReturn, RiskyWts, RiskyFraction, OverallRisk,
OverallReturn]
    = portalloc(PortRisk, PortReturn, PortWts, RisklessRate, BorrowRate,
RiskAversion)
```

输入参数

```
PortRisk          %有效前沿上每次资产的方差
PortReturn        %有效前沿上每项资产的回报
PortWts           %有效前沿上每项资产的权重
RisklessRate      %无风险利率
BorrowRate        %(Optional)借款利率，默认为没有借贷
RiskAversion      %(Optional)投资者的风险厌恶系数，大多数投资者的风险厌恶系数在2～4
                   之间，通常选择3
```

输出参数

```
RiskyRisk         %风险资产部分的标准差
RiskyReturn       %风险资产部分的回报
RiskyWts          %风险资产的权重
RiskyFraction     %总资产中风险资产的回报
OverallRisk       %总资产的标准差
OverallReturn     %总资产的回报
```

【例 5-12】已知一个组合中含有 3 种资产，每种资产的预期回报与协方差矩阵如表 5.9 所示。

表 5.9 各种资产的预期回报、协方差

		资 产 A	资 产 B	资 产 C
预期回报		0.1	0.2	0.15
协方差	资产 A	0.005	-0.010	0.004
	资产 B	-0.01	0.04	-0.002
	资产 C	0.004	-0.002	0.023

无风险利率为 0.08，借贷利率为 0.12，投资者的风险厌恶系数为 3，要求考虑无风险资产和借贷情况下的最优资产配置。

在 MATLAB 中执行如下命令：

```
>> ExpReturn = [0.1 0.2 0.15];
>> ExpCovariance = [0.005   -0.010    0.004
                   -0.010    0.040   -0.002
                    0.004   -0.002    0.023];
>> [PortRisk, PortReturn, PortWts] = portopt(ExpReturn,ExpCovariance);%由
于没有输入位于有效前沿上的点的数目，MATLAB 默认有效前沿上选取 10 个点，每个点代表一种组
合，每个组合标的准差保存在 PortRisk 中，收益率保存在 PortReturn 中，组合中各资产的权重
保存在 PortWts 中
```

下面调用 portalloc 函数求出考虑无风险资产，以及允许借贷时的资产配置，代码如下：

```
>> RisklessRate  = 0.08;
```

```
>> BorrowRate   = 0.12;
>> RiskAversion  = 3;
>> [RiskyRisk, RiskyReturn, RiskyWts, RiskyFraction, ...
OverallRisk, OverallReturn] = portalloc(PortRisk, PortReturn,...
PortWts, RisklessRate, BorrowRate, RiskAversion)
RiskyRisk =
          0.1283
RiskyReturn =
          0.1788
RiskyWts =
          0.0265    0.6023    0.3712
RiskyFraction =
          1.1898
OverallRisk =
          0.1527
OverallReturn =
          0.1899
```

从结果表明，最优组合的标准差为 0.1283，收益率为 0.1788，每项资产的权重分别为 0.0265、0.6023、0.3712。总资产中风险资产配置的权重为 1.1898，总资产的回报为 0.1899，总资产的标准差为 0.1527。

如果选取有效前沿上的 20 个点，得到结果如下：

```
>> [PortRisk, PortReturn, PortWts] = portopt(ExpReturn,ExpCovariance,20);
>> [RiskyRisk, RiskyReturn, RiskyWts, RiskyFraction,OverallRisk,
OverallReturn]
= portalloc(PortRisk, PortReturn, PortWts, RisklessRate, BorrowRate,
RiskAversion)
RiskyRisk =
   0.1288
RiskyReturn =
   0.1791
RiskyWts =
   0.0057    0.5879    0.4064
RiskyFraction =
   1.1869
OverallRisk =
   0.1529
OverallReturn =
   0.1902
```

从结果中我们可以知道，最优组合的标准差为 0.1283，组合的收益率为 0.1791，每项资产的权重分别为 0.0057、0.5879、0.4064，总资产中风险资产配置的权重为 1.1869，总资产的回报为 0.1902，总资的产标准差为 0.15297，除了资产配置差别比较大，第一项资产配

置进一步减小，其他差别并不大。

5.2.5 线性规划求解资产组合问题

线性规划是研究目标函数和约束条件均为线性的最优化问题，线性规划的标准形式如下：

$$\min_{Ax=b;x\geqslant 0} Cx$$

其中，C 是目标函数矩阵，A 是约束条件矩阵，x 是向量。

标准形式线性问题简称 LP(Linear Programming)问题，MATLAB 中用 lp 函数求解线性规划问题。

MATLAB 中的线性规划形式如下

$$\begin{aligned} \min \quad & fx \\ \text{s}\cdot\text{t} \quad & Ax \leqslant b \\ & Aeq = beq \\ & Lb \leqslant x \leqslant Ub \end{aligned}$$

其中，f,x,b 是向量，A 的形式是矩阵。

调用方式

```
x= linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq,lb,ub)
```

其中，f、A、b 分别为标准线性规划模型中的参数；参数 ub、lb 分别为变量 x 的上界与下界。

【例 5-13】某资产组合中有 3 种资产，各资产的收益率分别为 0.2、0.1、0.15。要求资产 1 与资产 3 的权重小于资产 2 的权重，且没有卖空。求解使得上述收益率最大的投资组合。

首先确定目标函数为：$\max \ 0.2x_1+0.1x_2+0.15x_3$。

资产约束条件可写为

$$x_1+x_3\leqslant x_2,\quad x_1+x_2+x_3=1 \ \text{且} \ 0\leqslant x_1,x_2\leqslant 1,\quad 0.1\leqslant x_3\leqslant 1$$

在 MATLAB 中执行如下命令：

```
>> f=[-0.2 -0.1 -0.15]; %目标函数的向量
>> a=[1 -1 1];
>> b=0;
>> aeq=[1 1 1];
>> beq=1;
>> lb=[0 0 0.1] ;
>> ub=[1 1 1];
>> x=linprog(f,a ,b,aeq,beq,lb,ub)
```

```
Optimization terminated.
x =
    0.4000
    0.5000
    0.1000
```

最后得出资产 1、资产 2、资产 3 的权重分别为 0.4、0.5、0.1。

下面我们考虑二次规划求解资产组合，二次规划问题(Quadratic Programming)的标准形式如下：

$$
\begin{aligned}
&\min \frac{1}{2} x^{\mathrm{T}} H x + q^{\mathrm{T}} x \\
&\mathrm{s \cdot t} \quad Ax \leqslant b \\
&\qquad Aeq * x = beq \\
&\qquad Lb \leqslant x \leqslant Ub
\end{aligned} \tag{5.5}
$$

其中，x 为权重向量，H 为对称矩阵。

约束条件分成两个部分，前一部分约束条件是不等式，后一部分约束条件是等式。在资产组合问题中，H 多为协方差矩阵。

在 MATLAB 中求解此类问题的函数是 quadprog。

调用方式

```
x = quadprog(H,q,A,b)
x = quadprog(H,q,A,b,Aeq,beq)
x = quadprog(H,q,A,b,Aeq,beq,lb,ub)
```

输入参数

H、q、A9 同式(5.1)

【例 5-14】资产组合中有 5 种资产，各资产的收益率和协方差矩阵如表 5.10 所示。

表 5.10　各资产的收益率与协方差矩阵

		资 产 1	资 产 2	资 产 3	资 产 4	资 产 5
协方差矩阵	资产 1	0.2	0.05	-0.01	0.03	0.05
	资产 2	0.05	0.3	0.015	0.01	0.03
	资产 3	-0.01	0.015	0.1	0.02	0.01
	资产 4	0.03	0.01	0.02	0.1	0.015
	资产 5	0.05	0.03	0.01	0.015	0.15
预期收益率		0.2	0.14	0.12	0.05	0.07

要求寻找最优资产组合，使得资产组合收益率为 0.1，且该组合为方差最小的资产组合。

在 MATLAB 中执行如下命令：

```
>> H=[0.2,0.05,-0.01,0.03,0.05;0.05,0.3,0.015,0.01,0.03;
   -0.01,0.015,0.1,0.02,0.01;…
    0.03,0.01,0.02,0.1,0.015;0.05,0.03,0.01,0.015,0.15;]; %各资产协方差矩阵
>> q=[0 0 0 0 0];
>> aeq=[1 1 1 1 1;0.2 0.14 0.12 0.05 0.07]; % 等式约束条件
>> beq=[1;0.1];
>> lb=[0 0 0 0 0];ub=[1 1 1 1 1];          % 权重的上下界
>> quadprog(H,q,[],[],aeq,beq,lb,ub)
  ans =
      0.1067
      0.0743
      0.3379
      0.2984
      0.1826
```

上述结果表明，最优资产配置是资产 1、资产 2、资产 3、资产 4、资产 5 所占比率分别为 10.67%、7.43%、33.79%、29.84%、18.26%。

思 考 题

1．将厦门建发(600153)的价格序列(2006 年 11 月 21 日至 2007 年 1 月 30 日)转换为收益率序列。

2．已知资产日回报率为 0.0025，标准差为 0.0208，资产现在价值为 0.8 亿，求 5%水平下资产的在险价值(Var)。

3．某资产组合中 4 种资产，各资产的收益率与协方差矩阵如表 5.11 所示。

表 5.11 各资产的收益率与协方差矩阵

		资 产 1	资 产 2	资 产 3	资 产 4
协方差矩阵	资产 1	0.2	0.05	-0.01	0.03
	资产 2	0.05	0.3	0.015	0.01
	资产 3	-0.01	0.015	0.1	0.02
	资产 4	0.03	0.01	0.02	0.1
预期收益率		0.2	0.14	0.12	0.05

试计算相关系数，以及资产组合有效前沿。如果规定第 4 种资产的权重小于 20%，其有效前沿如何变化。

4．某基金经理在资产配置时准备投资上证 50ETF 与某只中小盘基金，请参考资产组合

理论提出你的建议，并用程序说明你的建议。

5. 开放式基金在大盘快速下跌时风险很大，试用 Var 给出极端条件下需要的现金以应付不测，并结合实际分析如何对 Var 模型进行改进。

6. 基金经理经常面临短期业绩评价，例如评级机构 3 个月的排名压力，他们往往投资于大盘权重股，造成基金与大盘指数高度相关，试选择一只基金，考虑其与大盘的相关度，如果要求减小投资组合与大盘的相关性，应如何进行资产配置，试提出解决方案，要求编写程序进行说明。

7. 某基金经理准备募集一只对冲基金，策略是选择某只开放式基金与 ST 类股票作对冲，现在考虑需要准备多少金额的现金应付风险，请就该问题给出你的建议，要求既可以保证收益稳定又不让亏损击穿保证金。

8. 某机构于 2005 年 3 月 2 日持有一定数量的邯郸钢铁股份，但是害怕股价下跌会带来风险，影响 2005 年的年报，希望选择另外一只股票进行组合以降低风险，请就上述问题给出你的建议，要求编写程序说明。

第 6 章　金融衍生品计算

金融衍生品定价是金融工程的核心内容，也是金融业中发展最快的领域。本章主要介绍 MATLAB 自带的金融衍生品工具箱，要求掌握工具箱中常见期权的定价方法，利用利率树对利率类衍生产品定价，掌握价格树、利率树构建格式，掌握奇异期权的定价方法。

6.1　金融衍生产品种类

期权(Option)是一种合约，它赋予购买方在规定期限内按事先约定的价格(协议价格[①] (Striking Price)购买或出售一定数量某种标的金融资产(Underlying Assets)的权利。期权购买方为了获得这个权利，需支付给出售方一定金额的费用，称为期权费(Premium)，期权于此后失效的日期叫做到期日(Maturity Data)[②]。

期权分为基本期权和奇异期权两类。

1. 基本期权

基本期权[③](Vanilla Option)是常见的期权，如欧式期权(看涨、看跌期权)、美式期权(看涨、看跌)等。基本期权比较简单，除了行使方式、有效期和执行价，不再包括其他附加内容。基本期权包括下面几种类型。

(1) 欧式期权(European Option)：欧式看涨期权买方(卖方)有权在到期日以事先约定的价格(执行价)买入(或卖出)标的资产，期权买方同时支付期权费来购买这一权利。

(2) 美式期权(American Option)：美式期权和欧式期权内容相同，不同之处在于欧式期权的执行日是在到期日，而美式期权可以在存续期内任意时刻行权，这样就增加了期权使用者的灵活性，因此美式期权价值不会小于欧式期权。

(3) 看涨期权(Call)：该期权购买者可以在未来以商定价格购买标的资产。

(4) 看跌期权(Put)：该期权购买者可以在未来以商定价格卖出标的资产。

下面我们介绍期权内在价值与时间价值。

内在价值(Intrinsic Value)：期权内在价值是指如果立即行权获得的收益，看涨期权内在

---

① 又称执行价格(Exercise Price)。

② 又称满期日(Expiration Data)。

③ 又称普通期权。

价值是标的资产现价和执行价之差。例如某看涨期权执行价为 100 元，股票价格为 107 元，那么内在价值就是 7 元(107 元-100 元)。

时间价值(Time Value)：期权时间价值就是期权价格高于内在价值的部分，期权　时间价值=期权价值-内在价值。就前面的例子而言，如果期权价值为 8 元，那么该期权时间价值为 1 元(8 元-7 元)。

2. 奇异期权(Exotic Option)

奇异期权也叫做“第二代期权”，包括亚式期权、障碍期权、复合期权、回望期权、百慕大期权等。大多数奇异期权是金融机构为满足市场需求而专门设计的，多在场外交易。

(1) 亚式期权：亚式期权是一种路径依赖型期权，由于执行价是平均价格，不容易受到操纵，因而受到投资者青睐。亚式看涨期权到期现金流如下：

$$\max\left(\frac{\sum_{i=1}^{n} S_i}{n} - k\right)$$

其中，$S_i(i=1,2,3,\cdots,n)$，为各个日期标的资产价格，k 为事先约定行权价。

(2) 障碍期权：障碍期权是指期权回报依赖于标的资产价格在一段特定时间内是否达到了某个特定水平，这个特定水平就叫“障碍”水平。障碍期权分为下面 4 种类型。

(3) 上涨入局期权(Up Knock-in)：当标的资产价格超过事先规定的某个特定价格 B，该项期权就会被激活，而且 B 高于合同签订时标的资产的价格。

(4) 上涨出局期权(Up Knock-out)：当标的资产价格超过事先规定的某个特定价格 B，该项期权就会被终止，而且 B 高于合同签订时标的资产的价格。

(5) 下跌入局期权(Down Knock-in)：标的资产价格低于事先约定的水平(称之为障碍价格)时期权被激活。

(6) 下跌出局期权(Down Knock-out)：标的资产价格低于事先约定的水平(称之为障碍价格)时期权失效。

当障碍期权没有被执行时，期权卖方有时需支付给买方一笔费用，这笔费用叫做返还费(Rebates)。

(7) 复合期权：复合期权是以期权为标的的期权，标的可以是欧式期权，也可以是美式期权。复合期权有下列 4 种类型。

① 看涨期权的看涨期权(Call on a call)。

② 看涨期权的看跌期权(Put on a call)。

③ 看跌期权的看涨期权(Call on a put)。

④ 看跌期权的看跌期权(Put on a put)。

(8) 回望期权：回望期权是一种路径依赖型期权，该期权的到期现金流根据标的资产

价格最大值 S_{max} 或者最小值 S_{min} 是否高于或低于执行价 K 来确定。MATLAB 金融工具箱中回望期权包括固定式与浮动式两种，固定式期权执行价在合约签定时已经确定。回望期权根据到期现金流不同分为以下 4 种类型。

① 固定看涨(Fixed Call)：$\mathrm{Max}(0, S_{max} - K)$。

② 固定看跌(Fixed Put)：$\mathrm{Max}(0, K - S_{min})$。

③ 浮动看涨(Floating Call)：$\mathrm{Max}(0, S - S_{min})$。

④ 浮动看跌(Floating Put)：$\mathrm{Max}(0, S_{max} - S)$。

其中，S_{max} 为标的资产从 0 时刻至到期日的最大价格；S_{min} 为标的资产从 0 时刻至到期日的最小价格；K 为期权的执行价；S 为标的资产价格。

(9) 百慕大期权：一般只在固定日期行权，通常为一个月某一天。百慕大期权是美式期权与欧式期权的混合体，与美式期权的区别在于美式期权行权日不固定，而百慕大期权只能在某些固定日期行权。

MATLAB 中衍生产品定价主要通过衍生品工具箱完成，定价函数分为股票类衍生产品与利率类衍生产品两大类。各类金融产品定价方法如表 6.1 和表 6.2 所示。

表 6.1　股票类衍生产品在 MATLAB 中的定价方法①

股票类衍生产品	CRR 型二叉树	EQP 型二叉树
亚式期权(Asian option instrument)	√	√
普通股票期权(Stock option instrument)	√	√
障碍期权(Barrier instrument)	√	√
复合期权(Compound instrument)	√	√
回望期权(Lookback instrument)	√	√

表 6.2　利率类衍生产品在 MATLAB 中的定价方法

利率类衍生产品	HW 模型	BK 模型	BDT 模型	HJM 模型
债券工具(Bond instrument)	√	√	√	√
现金流(Cash Flow)	√	√	√	√
债券期权(Bond option)	√	√	√	√
固定收益票据(Fixed Rate note instrument)	√	√	√	√
浮动收益票据(Floating Rate note instrument)	√	√	√	√

① 表中横栏为定价方法和模型，竖栏表示金融产品种类。

续表

利率类衍生产品	HW 模型	BK 模型	BDT 模型	HJM 模型
利率戴帽期权(Cap option)	√	√	√	√
地板期权(Floor instrument)	√	√	√	√
利率掉期(Swap instrument)	√	√	√	√

6.2 欧式期权计算

欧式期权价格可以通过公式精确求解，下面我们介绍欧式期权定价基本理论。

6.2.1 Black-Scholes 方程

Black-Scholes 方程是金融衍生品最重要的定价公式，下面我们给出 Black-Scholes 方程的推导过程，首先我们介绍 ITO 引理。

ITO 引理 假设标的资产满足如下过程

$$\mathrm{d}x = a(x,t)\mathrm{d}t + b(x,t)\mathrm{d}W_t \tag{6.1}$$

其中 $\mathrm{d}W_t$ 是一个维纳过程，设 $G = G(x,t)$ 是 x 与 t 的函数，函数 G 二次连续可微，则 $G(x,t)$ 遵循如下过程

$$\mathrm{d}G(x,t) = \left(\frac{\partial G}{\partial x}a + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}b^2\right)\mathrm{d}t + \frac{\partial G}{\partial x}b\mathrm{d}W \tag{6.2}$$

证明： 由二元泰勒函数公式

$$\Delta G = \frac{\partial G}{\partial x}\Delta x + \frac{\partial G}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}\Delta x^2 + \frac{\partial^2 G}{\partial x \partial t}\Delta x \Delta t + \frac{1}{2}\frac{\partial^2 G}{\partial t^2}\Delta t^2 + \cdots \tag{6.3}$$

因为

$$\Delta x = a(x,t)\Delta t + b(x,t)\varepsilon\sqrt{\Delta t} \tag{6.4}$$

$$\Delta x^2 = a^2\Delta t^2 + 2ab\Delta t\sqrt{\Delta t} + b^2\varepsilon^2\Delta t \tag{6.5}$$

其中 ε 服从标准正态分布，$E(\varepsilon) = 0$，$E(\varepsilon^2) = 1$，因此 $E(b^2\varepsilon^2\Delta t) = b^2\Delta t$，因为当 $\Delta t \to 0$ 时，$\mathrm{var}(\Delta x^2) \to 0$，式(6.5)有

$$\Delta x^2 = b^2\Delta t + o(\Delta t) \tag{6.6}$$

由式(6.4)得

$$\Delta x \Delta t = a(x,t)\Delta t^2 + b(x,t)\varepsilon(\Delta t)^{3/2} = o(t) \tag{6.7}$$

将式(6.4)、式(6.6)带入式(6.3)得

$$\Delta G=\frac{\partial G}{\partial x}\Delta x+\frac{\partial G}{\partial t}\Delta t+\frac{1}{2}\frac{\partial^2 G}{\partial x^2}b^2\Delta t+o(\Delta t) \tag{6.8}$$

当 $\Delta t\to 0$，得

$$\mathrm{d}G=\frac{\partial G}{\partial x}\mathrm{d}x+\frac{\partial G}{\partial t}\mathrm{d}t+\frac{1}{2}\frac{\partial^2 G}{\partial x^2}b^2\mathrm{d}t \tag{6.9}$$

将 $\mathrm{d}x=a(x,t)\mathrm{d}t+b(x,t)\mathrm{d}W$ 代入式(6.9)得

$$\mathrm{d}G=\left(\frac{\partial G}{\partial x}a+\frac{\partial G}{\partial t}+\frac{1}{2}\frac{\partial^2 G}{\partial x^2}b^2\right)\mathrm{d}t+\frac{\partial G}{\partial x}b\mathrm{d}W$$

命题得证。

下面我们推导出 Black-Scholes 方程。

假设标的资产价格服从几何布朗运动，即

$$\mathrm{d}S=\mu S\mathrm{d}t+\sigma S\mathrm{d}W$$

期权价格为 $f(S,t)$，由 ITO 定理可得

$$\begin{aligned}\mathrm{d}f&=\left(\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 V}{\partial S^2}\right)\mathrm{d}t+\frac{\partial V}{\partial S}\mathrm{d}S\\&=\left(\frac{\partial V}{\partial t}+\mu S\frac{\partial V}{\partial S}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 V}{\partial S^2}\right)\mathrm{d}t+\sigma S\frac{\partial V}{\partial S}\mathrm{d}W\end{aligned}$$

下面我们考虑一个组合 Π：卖出一个看跌期权，同时买入 Δ 数量股票，则

$$\Pi=\Delta S-f$$

$$\begin{aligned}\mathrm{d}\Pi&=\Delta\mathrm{d}S-\mathrm{d}f\\&=\Delta(\mu S\mathrm{d}t+\sigma S\mathrm{d}W)-\left[\left(\frac{\partial V}{\partial t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 V}{\partial S^2}\right)\mathrm{d}t+\frac{\partial V}{\partial S}\mathrm{d}S\right]\end{aligned}$$

$$\mathrm{d}\Pi=\sigma S\left(\Delta-\frac{\mathrm{d}f}{\mathrm{d}S}\right)\mathrm{d}W+\left[\mu S\left(\Delta-\frac{\mathrm{d}f}{\mathrm{d}S}\right)-\frac{\mathrm{d}f}{\mathrm{d}t}-\frac{1}{2}\sigma^2S^2\frac{\mathrm{d}^2 f}{\mathrm{d}S^2}\right]\mathrm{d}t \tag{6.10}$$

如果我们选择 $\Delta=\dfrac{\mathrm{d}f}{\mathrm{d}S}$，$\mathrm{d}\Pi$ 中没有了随机项 $\mathrm{d}W$，如果我们能够随时间变化及时调整 Δ 就可以在整个时间段内将资产变成无风险资产，如果资产组合变成了无风险资产，那么其收益率和无风险资产收益率应该相等。即

$$\mathrm{d}\Pi=r\Pi\mathrm{d}t$$

有

$$\frac{\mathrm{d}f}{\mathrm{d}t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 f}{\partial S^2}+r\Pi=0$$

式中，r 为无风险利率。

整理得到

$$\frac{\partial f}{\mathrm{d}t}+rS\frac{\partial f}{\mathrm{d}S}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 f}{\mathrm{d}S^2}-rf=0$$

上式就是 Black-Scholes 方程，表明金融衍生产品定价可以用偏微分方程表示，这样各种不同衍生证券对应于到期现金流。

欧式看涨期权价格是

$$C=SN\left[\frac{\ln\left(\frac{S}{K}\right)+\left(r+\frac{\sigma^2}{2}T\right)}{\sigma\sqrt{T}}\right]-\mathrm{e}^{-rT}N\left[\frac{\ln\left(\frac{S}{K}\right)-\left(r+\frac{\sigma^2}{2}T\right)}{\sigma\sqrt{T}}\right]$$

其中，S 为股票价格，K 是执行价，$N(\cdot)$ 是正态分布函数，r 是无风险利率，T 是期权存续期，σ 是标准差。

1976 年 Black 研究出期货期权定价模型，该模型假设了期货价格 F 遵循如下几何布朗运动：

$$\mathrm{d}F=\mu F\mathrm{d}t+\sigma\mathrm{d}W_t$$

这里 μ 是期货价格预期增长率，σ 是波动率，$\mathrm{d}W_t$ 是维纳过程。

设欧式期货看涨期权价格为 c，看跌期权价格为 p，则有

$$c=\mathrm{e}^{-rT}[FN(d_1)-KN(d_2)]$$

$$p=\mathrm{e}^{-rT}[XN(-d_2)-FN(-d_1)]$$

其中

$$d_1=\frac{\ln(F/X)+\frac{\sigma^2}{2}T}{\sigma\sqrt{T}}$$

$$d_2=\frac{\ln(F/X)-\frac{\sigma^2}{2}T}{\sigma\sqrt{T}}=d_1-\sigma\sqrt{T}$$

其中，F 为期货价格，K 是执行价，$N(\cdot)$ 是正态分布函数，r 是无风险利率，T 是存续期，σ 是标准差。

6.2.2　欧式期权价格函数

MATLAB 中计算欧式期权价格的函数是 blsprice。

调用方式

```
[Call, Put] = blsprice(Price, Strike, Rate, Time, Volatility, Yield)
```

输入参数

```
Price           %标的资产价格
Strike          %执行价
Rate            %无风险利率
Time            %距离到期日的时间，即期权的存续期
Volatility      %标的资产的标准差
Yield           %标的资产的红利率
```

输出参数

```
Call            %欧式看涨期权价格
Put             %欧式看跌期权价格
```

【例 6-1】股票价格为 100，股票波动率标准差为 0.5，无风险利率为 10%，期权执行价为 95，存续期为 0.25 年，试计算该股票欧式期权价格。

在 MATLAB 中执行如下命令：

```
>> [Call, Put] = blsprice(100, 95, 0.1, 0.25, 0.5)
Call =
   13.6953
Put =
    6.3497
```

从计算结果看，该股票欧式看涨期权价格为 13.6953，欧式看跌期权价格为 6.3497。

6.2.3 欧式期权希腊字母

欧式期权希腊字母也称为期权避险参数，主要是衡量影响期权价格的因素，包括下面几种类型。

(1) 德尔塔值(Delta)：期权德尔塔是考察期权价格随标的资产价格变化的关系，从数学角度看，Delta 是期权价格相对于标的资产价格的偏导数

$$\text{Delta} = \frac{\partial c}{\partial p}$$

其中，c 是期权价格，p 是标的资产价格。

例如某个看涨期权 Delta 值为 0.5，表示当股价变化 ΔS 时，期权价格变化为 $0.5\Delta S$。例如期权价格为 10，股票价格为 100，某个投资者购买了 1 份(100 股股票期权)该股票看涨期权，投资者可以购买 0.5×100=50 股股票来对冲风险，这样的投资组合为 Delta 中性策略，假如股票价格下跌 1 元，投资于股票损失为 50 元，而期权收益为 0.5×100=50 元，无论股票价格如何变化，资产组合 Delta 为 0，这种投资策略又称为 Delta 中性投资策略。

(2) 伽玛(Gamma)：衡量德尔塔与标的资产价格变动的关系，从数学角度看相当于期权价格对于标的资产的二阶偏导数

$$\text{Gamma} = \frac{\partial^2 c}{\partial^2 p}$$

(3) 维伽(Vega)：衡量期权价格与标的资产波动率之间的关系，从数学角度看相当于期权价格对于波动率的偏导数

$$\text{Vega} = \frac{\partial c}{\partial \sigma}$$

其中，σ 为标的资产标准差。

(4) 西塔(Theta)：衡量期权价格与时间变化之间的关系，从数学角度看相当于期权价格对于时间的偏导数

$$\text{Theta} = \frac{\partial c}{\partial t}$$

其中，t 为期权的存续期。

单个期权的 Theta 几乎总为负值，因为越临近到期日，期权不确定性越低，期权越不值钱。

(5) 珞(Rho)：衡量期权价格与无风险利率之间的关系，从数学角度看相当于期权价格对于无风险利率的偏导数

$$\text{Rho} = \frac{\partial c}{\partial r}$$

其中，r 为无风险收益率。

1. 欧式期权 Delta 值

调用方式

```
[CallDelta, PutDelta] = blsdelta(Price, Strike, Rate, Time, Volatility,
Yield)
```

输入参数

同 blsprice 函数。

输出参数

```
CallDelta       %欧式看涨期权 Delta
PutDelta        %欧式看跌期权 Delta
```

【例 6-2】股票价格为 50，执行价为 50，无风险利率为 10%，期权存续期为 0.25，波动率的标准差为 0.3，存续期内股票无红利，计算该期权 Delta 值。

在 MATLAB 中执行如下命令：

```
>> [CallDelta, PutDelta] = blsdelta(50, 50, 0.1, 0.25, 0.3, 0)
CallDelta =
0.5955
PutDelta =
   -0.4045
```

看涨期权 Delta 值为 0.5955，看跌期权 Delta 值为-0.4045。

2. 欧式期权 Gamma 值

调用方式

```
Gamma = blsgamma(Price, Strike, Rate, Time, Volatility, Yield)
```

输入参数

同 blsprice 函数。

输出参数

```
Gamma              %欧式期权 Gamma 值
```

【例 6-3】 股票价格为 50，执行价为 50，无风险收益率为 12%，存续期为 0.25，波动率的标准差为 0.3，存续期内股票无红利，计算该期权 Gamma 值。

在 MATLAB 中执行如下命令：

```
>> Gamma = blsgamma(50, 50, 0.12, 0.25, 0.3, 0)
Gamma =
    0.0512
```

该期权 Gamma 值为 0.0512。

3. 欧式看涨期权 Theta 值

调用方式

```
[CallTheta, PutTheta] = blstheta(Price, Strike, Rate, Time, Volatility,
Yield)
```

输入参数

同 blsprice 函数。

输出参数

```
CallTheta          %欧式看涨期权 Theta 值
PutTheta           %欧式看跌期权 Theta 值
```

4. 欧式期权 Rho 值

调用方式

```
[CallRho, PutRho]= blsrho(Price, Strike, Rate, Time, Volatility, Yield)
```

输入参数

同 blsprice 函数。

输出参数

```
CallRho          %欧式看涨期权 Rho 值
PutRho           %欧式看跌期权 Rho 值
```

5. 欧式期权 Vega

调用方式

```
Vega = blsvega(Price, Strike, Rate, Time, Volatility, Yield)
```

输入参数

同 blsprice 函数。

输出参数

```
Vega               %欧式期权 Vega
```

【例 6-4】股票价格为 50，执行价为 50，无风险利率为 12%，存续期为 0.25，波动率的标准差为 0.3，存续期内无红利，计算该期权 Vega 值。

```
>> Vega = blsvega(50, 50, 0.12, 0.25, 0.3, 0)
Vega =
    9.6035
```

该期权 Vega 值为 9.6035。

6. 欧式期权隐含波动率

已知欧式期权价格，也可以推导出隐含波动率的标准差，然后用隐含波动率与实际波动率相比较，并作为投资决策参考。

调用方式

```
Volatility = blsimpv(Price, Strike, Rate, Time, Value, Limit, Yield,
                Tolerance, Type)
```

输入参数

```
Price            %标的资产当前价格
Strike           %期权执行价
Rate             %无风险利率
```

```
Time            %存续期
Value           %欧式期权价格
Limit           %(Optional)欧式期权波动率上限，默认值是10
Yield           %(Optional)标的资产的分红，折合成年收益率
Tolerance       %(Optional)可以忍受的隐含波动率，默认值为1.0×10^6
Type            %(Optional)欧式期权种类，如果是欧式看涨期权则输入Type = {'call'},
                 如果是欧式看跌期权，则输入Type = {'put'}，默认值为欧式看涨期权
```

输出参数

```
Volatility     %欧式期权隐含波动率，期权类别由Type确定
```

6.2.4 期货期权定价函数

MATLAB 中求解期货期权价格的函数是 blkprice。

调用方式

```
[Call, Put] = blkprice(Price, Strike, Rate, Time, Volatility)
```

输入参数

```
Price           %期货价格
Strike          %期货期权执行价
Rate            %无风险利率
Time            %期权存续期
Volatility      %期货变化标准差
```

输出参数

```
Call             %欧式看涨期权价格
Put              %欧式看跌期权价格
```

6.3 衍生产品定价数值解

6.3.1 CRR 二叉树模型

CRR 二叉树模型(Cox-Ross-Rubinstein 模型)，简称 CRR 模型。

对于一些期权，无法像欧式期权一样有解析解，因此就需要用数值解进行近似计算，二叉树方法就是其中一种，该方法由 J.Cox、S.Ross 和 M.Rubinstein 于 1979 年给出。

二叉树模型首先把时间分成许多小的时间段，记为 Δt，并假设期权价格仅存在上升与

下降两种可能性[①]，上升与下降的比率分别为 u 和 d，对应概率分别为 p 和 $(1-p)$。下面给出建立二叉树的步骤。

【例 6-5】 股票价格为 52，无风险收益率为 10%，期权距离到期日为 5 个月，股票波动率的标准差为 0.4，欧式看跌期权执行价为 52，假设将时间离散为 5 个时间段，利用二叉树模型估计看跌期权价格。

第 1 步：确定 p,u,d 参数。

假设股价初期价格为 S，如果投资无风险资产，经过 Δt 后价值为 $Se^{r\Delta t}$，股票收益期望应为

$$Se^{r\Delta t} = pSu + (1-p)Sd$$

整理得

$$e^{r\Delta t} = pu + (1-p)d \tag{6.11}$$

由于标的资产服从几何布朗运动，经过 Δt 时间段，其方差为 $S^2\sigma^2\Delta t$，必须和离散模型中的资产方差相等，离散资产方差根据公式 $D(X) = EX^2 - (EX)^2$，这样有

$$S^2\sigma^2\Delta t = pS^2u^2 + (1-p)S^2d^2 - S^2[pu + (1-p)d]^2$$

整理得

$$\sigma^2\Delta t = pu^2 + (1-p)d^2 - [pu + (1-p)d]^2 \tag{6.12}$$

选择 u,d 满足下面关系

$$u = 1/d \tag{6.13}$$

从式(6.11)、式(6.12)、式(6.13)可以解出

$$p = \frac{e^{r\Delta t} - d}{u - d}$$

$$u = e^{\sigma\sqrt{\Delta t}}$$

$$d = e^{-\sigma\sqrt{\Delta t}}$$

对于例 6-5 我们计算参数为

$$u = e^{0.4\sqrt{1/12}} = 1.1224$$

$$d = e^{-\sigma\sqrt{\Delta t}} = 0.8909$$

$$p = \frac{e^{-0.1\sqrt{5/12}} - d}{u - d} = 0.5073$$

第 2 步：二叉树结构。

当时间为 0 时，证券价格为 S，时间为 Δt 时，证券价格要么上涨到 Su，要么下跌到 Sd；时间为 $2\Delta t$ 时，证券价格就有 3 种可能，分别为 Su^2，Sud，Sd^2，以此类推，在时间 $i\Delta t$，

① 这样的假设只是为计算方便，不是必要的。

证券价格有 $i+1$ 种可能，用公式表示为

$$Su^j d^{i-j}$$

其中，$j=0,1,2,3,\cdots$，$i=1,2,3,\cdots$。

对于例 6.5 中的期权，其二叉树图结构如图 6.1 所示。

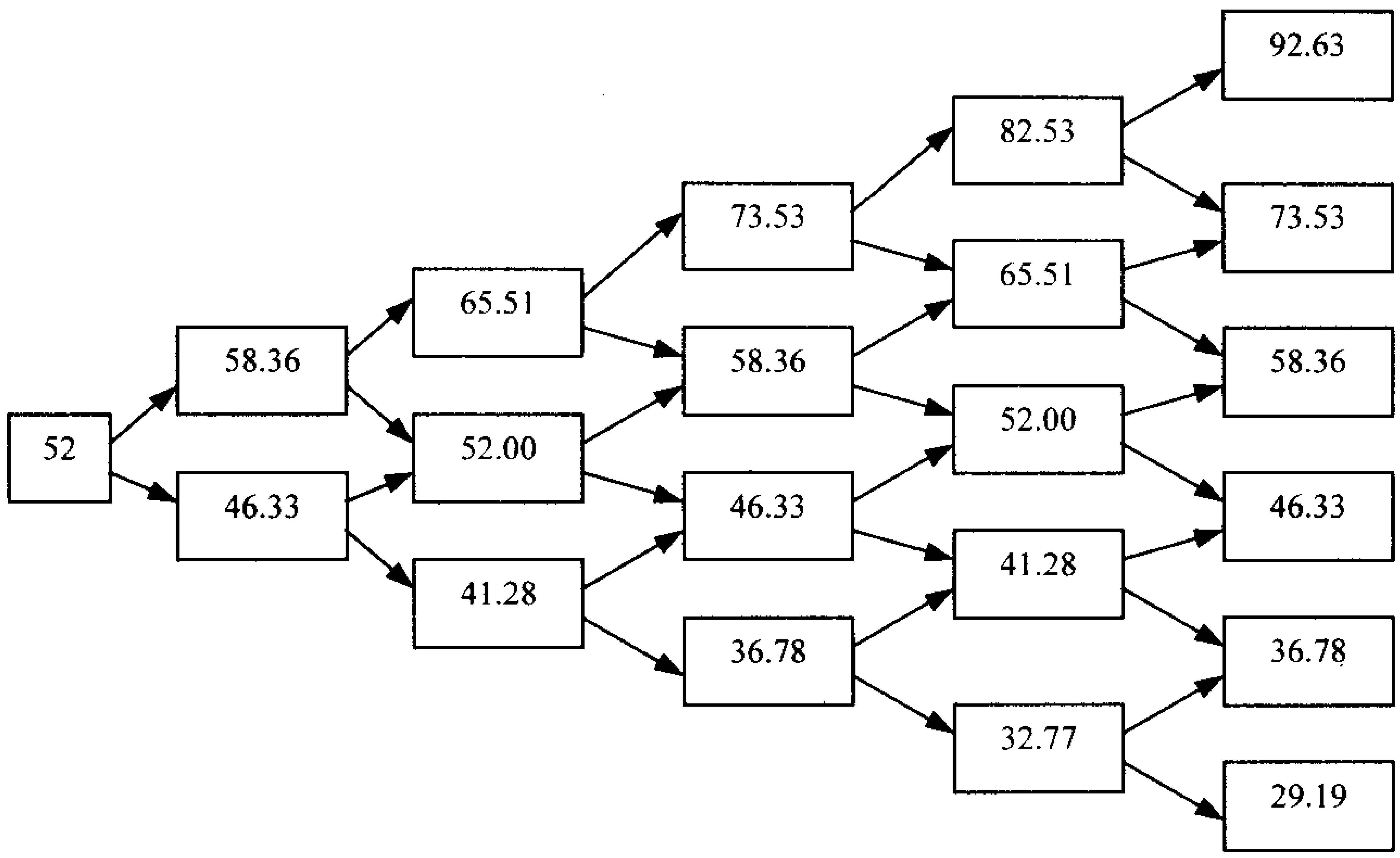


图 6.1　CRR 型二叉树示意图

第 3 步：根据二叉树进行倒推定价。

在二叉树模型中，期权定价从树形图末端开始，采用倒推定价法进行。由于在 T 时刻欧式看跌期权现金流为 $\max(K-S_T,0)$，求解 $T-\Delta t$ 时刻每一节点上的期权价格时都可以通过将 T 时刻期权现金流预期值以无风险收益率进行贴现求出。以此类推，如果是美式期权，就要看树形图每一节点上，提前执行是否比将期权持有到下一期更有利，采用这种方法最终可以求出 0 时刻的期权价值。

假设将欧式看跌期权的存续期分成 N 个长度为 Δt 的小区间，设 $f_{i,j}(0\leqslant i\leqslant N$，$0\leqslant j\leqslant i)$ 表示在时刻 $i\Delta t$ 第 j 个节点处的欧式看跌期权价格，也称 $f_{i,j}$ 为节点 (i,j) 的期权价值，同时 $Su^j d^{i-j}$ 表示节点 (i,j) 处的标的价格，欧式看跌期权到期价值是 $\max(K-S_T,0)$，所以有

$$f_{N,j}=\max(K-Su^j d^{N-j},0)$$

其中，$j=0,1,2,3,\cdots,N$。

当时间从 $i\Delta t$ 变到 $(i+1)\Delta t$ 时，从节点 (i,j) 移动到 $(i+1,j+1)$ 的概率为 p，移动到 $(i+1,j)$ 的概率为 $(1-p)$，则在风险中性情况下

$$f_{i,j}=\mathrm{e}^{-r\Delta t}[pf_{i+1,j+1}+(1-p)f_{i+1,j}]，\quad 0\leqslant i\leqslant N-1，\quad 0\leqslant j\leqslant i$$

当我们选择的时间间隔足够小时，就可以求出欧式看跌期权的精确值。图 6.2 是倒推法过程。

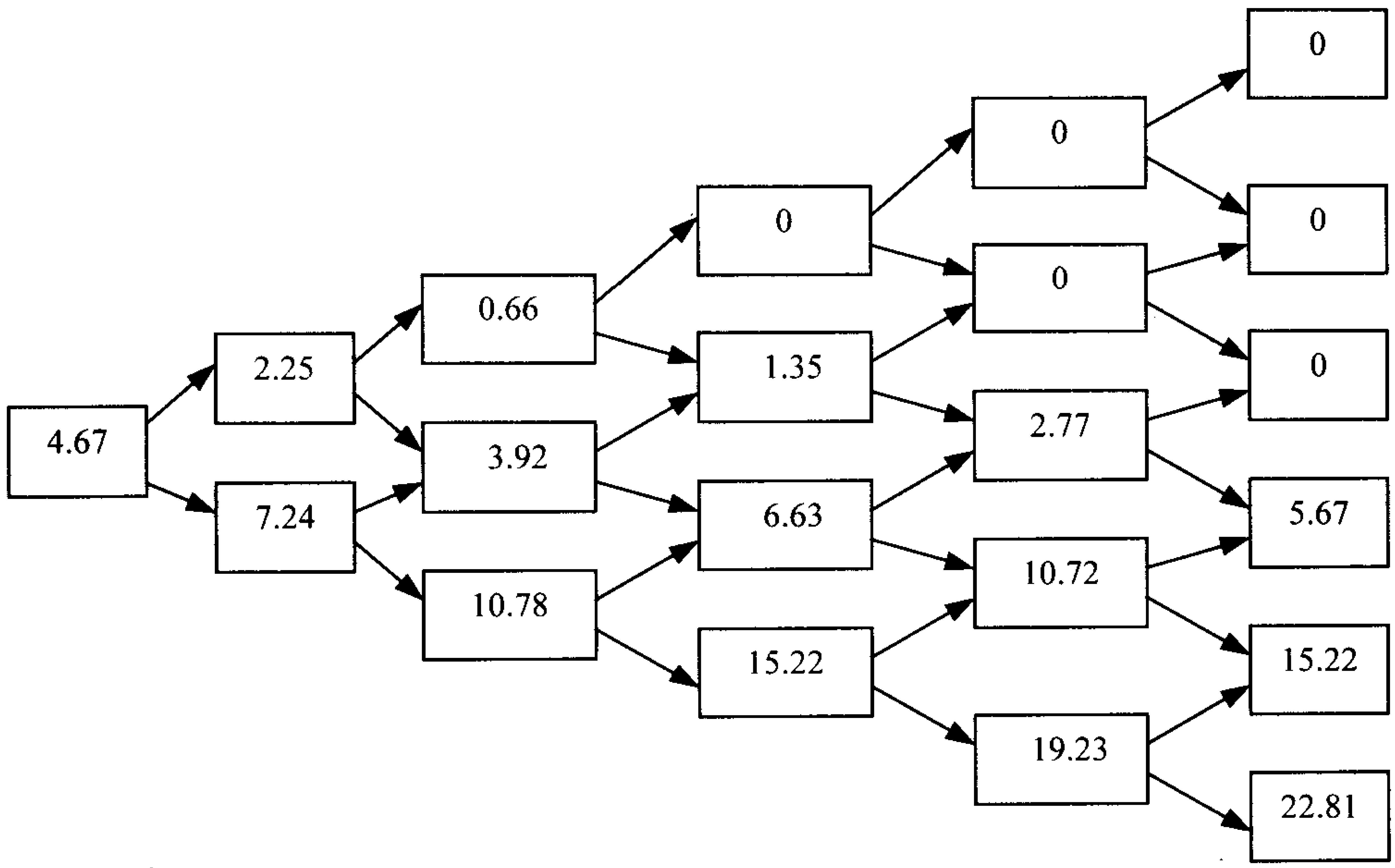


图 6.2　二叉树现金流贴现示意图

从图 6.2 可以看出欧式看跌期权价值是 4.67。

6.3.2　EQP 型二叉树模型

在 CRR 模型中，我们首先建立式(6.11)、式(6.12)，为了减少节点数目，方便计算，假设 $ud=1$，这样式(6.11)、式(6.12)分别变为

$$u+d=2\mathrm{e}^{r\Delta t} \tag{6.14}$$

$$u^2+d^2=2\mathrm{e}^{(2r+\sigma^2)\Delta t} \tag{6.15}$$

这样可以解式(6.14)、式(6.15)得

$$u=\mathrm{e}^{r\Delta t}(1+\sqrt{\mathrm{e}^{\sigma^2\Delta t}-1}) \tag{6.16}$$

$$d=\mathrm{e}^{r\Delta t}(1-\sqrt{\mathrm{e}^{\sigma^2\Delta t}-1}) \tag{6.17}$$

当 $\Delta t\to 0$，式(6.16)、式(6.17)一阶近似变为：

$$u=1+r\Delta t+\sigma\sqrt{\Delta t} \tag{6.18}$$

$$d=1+r\Delta t-\sigma\sqrt{\Delta t} \tag{6.19}$$

实际上，式(6.16)、式(6.17)还可以有更高阶近似

$$u = e^{a\Delta t + \sigma\sqrt{\Delta t}} \tag{6.20}$$

$$d = e^{a\Delta t - \sigma\sqrt{\Delta t}} \tag{6.21}$$

这里 $a = r - \frac{1}{2}\sigma^2$。

6.3.3 二叉树定价函数

MATLAB 中给期权定价采用的方法是 Cox-Ross-Rubinstein(CRR) 二叉树模型，函数名称为 binprice。

调用方式

```
[AssetPrice, OptionValue] = binprice(Price, Strike, Rate, Time, Increment,
Volatility,Flag,DividendRate,Dividend, ExDiv)
```

输入参数

```
Price           %股票价格
Strike          %期权的执行价
Rate            %无风险利率
Time            %期权存续期
Increment       %时间的增量
Volatility      %波动率的标准差
Flag            %确定期权种类，看涨期权(Flag=1)，看跌期权(Flag=0)
DividendRate    %(Optional) 红利发放率。默认值为 0，表示没有红利，如果给出了红利率，
                 Dividend 与 ExDiv 值为 0
Dividend        %(Optional) 标的资产价外的红利金额，除了固定红利之外的红利
ExDiv           %(Optional) 标的资产的除息日期
```

输出参数

```
Price           %二叉树每个节点的价格
Option          %期权在每个节点的现金流
```

【例 6-6】股票价格为 52，无风险收益率为 10%，期权存续期为 5 个月，波动率的标准差为 0.4，在 3 个半月(折合时间为 3.5)发放红利 2.06 元，看跌期权执行价为 50，利用二叉树模型估计看跌期权价格。

在 MATLAB 中执行如下命令：

```
>> [Price,Option]=binprice(52,50,0.1,5/12,1/12,0.4,0,0,2.06,3.5)
Price =
   52.0000    58.1367    65.0226    72.7494    79.3515    89.0642
         0    46.5642    52.0336    58.1706    62.9882    70.6980
         0          0    41.7231    46.5981    49.9992    56.1192
         0          0          0    37.4120    39.6887    44.5467
```

```
     0         0         0         0   31.5044   35.3606
     0         0         0         0         0   28.0688
Option =
  4.4404    2.1627    0.6361         0         0         0
       0    6.8611    3.7715    1.3018         0         0
       0         0   10.1591    6.3785    2.6645         0
       0         0         0   14.2245   10.3113    5.4533
       0         0         0         0   18.4956   14.6394
       0         0         0         0         0   21.9312
```

从计算结果看，Option 第一行第一列就是看跌期权价格，该期权价格为 4.4404 元。

6.4 证券类衍生产品定价函数

6.4.1 标的资产输入格式

MATLAB 对衍生产品定价是通过价格树来完成的，价格树由 3 个部分构成，分别是标的资产特征、无风险利率特征与时间的离散方法；用公式表示为：价格树=证券特征+无风险收益率特征+时间的离散方法。定义标的资产特征、无风险收益率特征的函数比较简单，分别是 stockspec 与 intenvset，定义时间的离散方法有很多种，不同模型定义的方法也不一样。

1. 证券特征定义

调用方式

```
StockSpec=stockspec(Sigma, AssetPrice, DividendType, DividendAmounts,
ExDividendDates)
```

输入参数

```
Sigma              %标的资产波动率
AssetPrice         %标的资产的价格
DividendType       %(Optional)红利发放方式，注意红利发放方式一定是以现金形式，
                     "cash"现金红利绝对额，"constant"常数红利，"continuous"
                     连续形式红利
DividendAmounts    %(Optional)发放红利数量，可以为向量形式，或者用标量表示的每年以
                     固定数量发放的红利
ExDividendDates    %(Optional)除息日，如果红利是连续型的，则不需要该参数
```

【例 6-7】已知标的资产的标准差σ为0.27，当前价格为 50，标的资产红利发放格式如表 6.3 所示。

表 6.3　标的资产红利发放格式

时间	2003 年 1 月 3 日	2003 年 4 月 1 日	2003 年 7 月 5 日	2003 年 10 月 1 日
现金	0.5	0.5	0.5	0.5

在 MATLAB 中保存标的资产格式如下：

```
>> Sigma = 0.27;
>> AssetPrice = 50;
>> DividendType = 'cash';
>> DividendAmounts = [0.50; 0.50; 0.50; 0.50];
>> ExDividendDates = {'03-Jan-2003'; '01-Apr-2003'; '05-July-2003';
                      '01-Oct-2003'}
ExDividendDates =
    '03-Jan-2003'
    '01-Apr-2003'
    '05-July-2003'
    '01-Oct-2003'
>> StockSpec = stockspec(Sigma, AssetPrice, DividendType, …
DividendAmounts, ExDividendDates)
StockSpec =
             FinObj: 'StockSpec'
              Sigma: 0.2700
         AssetPrice: 50
       DividendType: 'cash'
    DividendAmounts: [4x1 double]
    ExDividendDates: [4x1 double]
```

2. 无风险收益率格式

无风险收益率是衍生产品定价模型中的重要参数，实际上任何金融产品都具有风险，绝对无风险收益率是找不到的，一般将违约率低的固定收益率作为无风险收益率，国债违约率最低，一般把国债收益率视为无风险收益率，无风险收益率是国债收益率期限结构。MATLAB 中建立无风险收益率格式的函数如下。

调用方式

```
[RateSpec, RateSpecOld] = intenvset(RateSpec, 'Parameter1', Value1,
'Parameter2', Value2)
```

输入参数

```
RateSpec          %旧的无风险收益率格式
'Parameter1'      %参数 1 的名称
Value1            %参数 1 的值
'Parameter2'      %参数 2 的名称
Value2            %参数 2 的值
```

各个参数内容如下：

```
Disc            %贴现率
Rates           %国债票息
StartDates      %开始日
EndDates        %结束日
ValuationDate   %评估日，即价格树起始时间
Basis           %应计天数计算方式
EndMonthRule    %月末法则
Compounding     %(Optional)票息转换为贴现率方式，默认值为 2，Compounding 可以取 1，
                 2， 3， 4， 6， 12
```

贴现率计算方式如下：

$$Disc = \left(1+\frac{Z}{F}\right)^{-T}$$

其中，F 为计息频率，Z 为票息率，T 为时间长度，$Disc$ 为贴现率。

如果 Compounding = 365，则贴现率计算方式如下：

$$Disc = \left(1+\frac{Z}{F}\right)^{-T}$$

如果 Compounding=-1，表示按复利贴现，则贴现率计算方式如下

$$Disc = \exp(-T \cdot Z)$$

输出参数

```
RateSpec       %无风险利率新格式
RateSpecOld    %无风险利率旧格式
```

【例 6-8】国债利率为复合利率，票息及其支付日如表 6.4 所示。

表 6.4 国债票息支付方式

起息日	2000-1-1	—	—	—
到期日	2001-1-1	2002-1-1	2003-1-1	2004-1-1
利息	0.02	0.03	0.03	0.05

我们建立无风险利率格式如下：

```
>> Compounding = 1;
>> Rates = [0.02; 0.03; 0.04; 0.05];
>> StartDates = ['01-Jan-2000'];
>> EndDates = ['01-Jan-2001';
              '01-Jan-2002';
              '01-Jan-2003';
              '01-Jan-2004'];
```

```
>> ValuationDate = '01-Jan-2000';
>>RateSpec = intenvset('Compounding',1,'Rates', Rates, 'StartDates',
            StartDates, … 'EndDates', EndDates,'ValuationDate',
            ValuationDate)
```

结果显示如下：

```
RateSpec =
          FinObj: 'RateSpec'
     Compounding: 1
            Disc: [4x1 double]
           Rates: [4x1 double]
        EndTimes: [4x1 double]
      StartTimes: [4x1 double]
        EndDates: [4x1 double]
      StartDates: 730486
   ValuationDate: 730486
           Basis: 0
    EndMonthRule: 1
```

如果要显示 RateSpecs 中 Rates 的内容可以用 intenvget 函数，代码如下：

```
>> intenvget(RateSpec,'Rates')
ans =
    0.0200
    0.0300
    0.0400
    0.0500
```

或者直接用结构变量方式打开，代码如下：

```
>> RateSpec.Rates
ans =
    0.0200
    0.0300
    0.0400
    0.0500
```

3. 树图时间离散格式

时间离散格式比较多，不同模型有不同的时间格式。

1) CRR 模型的时间离散格式

调用方式

```
TimeSpec = crrtimespec(ValuationDate, Maturity, NumPeriods)
```

输入参数

```
ValuationData   %评估日，CRR 型树起始日期
Maturity        %到期日
NumPeriods      %离散时间段
```

【例 6-9】期权生效日为 2002 年 7 月 1 日，到期日为 2006 年 7 月 1 日，分 4 段进行离散。

在 MATLAB 中执行如下命令：

```
>> ValuationDate = '1-July-2002';
>> Maturity = '1-July-2006';
>> NumPeriods=4
>> TimeSpec = crrtimespec(ValuationDate, Maturity, NumPeriods)
TimeSpec =
          FinObj: 'BinTimeSpec'
     ValuationDate: 731398
         Maturity: 732859
      NumPeriods: 4
           Basis: 0
    EndMonthRule: 1
            tObs: [0 1 2 3 4]
            dObs: [731398 731763 732129 732494 732859]
```

2)　EQP 模型的时间离散格式

调用方式

```
TimeSpec = eqptimespec(ValuationDate, Maturity, NumPeriods)
```

输入参数

同 crrtimespec 函数。

对于上面例子我们建立 EQP 型的时间离散格式如下：

```
>>  ValuationDate = '1-July-2002';
>> Maturity = '1-July-2006';
>>  TimeSpec = eqptimespec(ValuationDate, Maturity, 4); %时间离散为 4 段
TimeSpec =
          FinObj: 'BinTimeSpec'
    ValuationDate: 731398
         Maturity: 732859
       NumPeriods: 4
           Basis: 0
     EndMonthRule: 1
            tObs: [0 1 2 3 4]
            dObs: [731398 731763 732129 732494 732859]
```

6.4.2 证券类衍生产品二叉树建立

1. CRR 型二叉树函数的调用

调用方式

```
CRRTree=crrtree(StockSpec,RateSpec,TimeSpec)
```

输入参数

```
StockSpec    %股票的格式
RateSpec     %利率的格式
TimeSpec     %时间的离散化方法
```

输出参数

```
CRRTree    %价格树
```

【例 6-10】股票波动的标准差为 0.2，标的资产价格为 50，红利类型为现金红利，除息日期如表 6.5 所示。

表 6.5 股票红利发放方式

日期	2003-1-3	2003-4-1	2003-7-1	2003-10-1
红利	0.5	0.5	0.5	0.5

下面我们给出该证券价格在 MATLAB 中的格式，代码如下：

```
>> Sigma = 0.20;
>> AssetPrice = 50;
>> DividendType = 'cash';
>> DividendAmounts = [0.50; 0.50; 0.50; 0.50];
>> ExDividendDates = {'03-Jan-2003'; '01-Apr-2003'; '05-July-2003';
    '01-Oct-2003'}
   ExDividendDates =
    '03-Jan-2003'
    '01-Apr-2003'
    '05-July-2003'
    '01-Oct-2003'
>> StockSpec = stockspec(Sigma, AssetPrice, DividendType,DividendAmounts,…
ExDividendDates)
   StockSpec =
            FinObj: 'StockSpec'
             Sigma: 0.2000
        AssetPrice: 50
      DividendType: 'cash'
```

```
   DividendAmounts: [4x1 double]
   ExDividendDates: [4x1 double]
```

下面给出无风险利率格式，代码如下：

```
>> ateSpec = intenvset('Rates', 0.05, 'StartDates','01-Jan-2003', 'EndDates',
                    '31-Dec-2003')
ateSpec =
           FinObj: 'RateSpec'
      Compounding: 2
             Disc: 0.9519
            Rates: 0.0500
         EndTimes: 1.9945
       StartTimes: 0
         EndDates: 731946
       StartDates: 731582
    ValuationDate: 731582
            Basis: 0
     EndMonthRule: 1
```

下面我们给出时间输入格式，代码如下：

```
>> ValuationDate = '1-Jan-2003';
>>Maturity = '31-Dec-2003';
>>TimeSpec = crrtimespec(ValuationDate, Maturity, 4)  %离散的时间数目是 4 个
TimeSpec =
           FinObj: 'BinTimeSpec'
    ValuationDate: 731582
         Maturity: 731946
       NumPeriods: 4
            Basis: 0
     EndMonthRule: 1
             tObs: [0 0.2493 0.4986 0.7479 0.9972]
             dObs: [731582 731672 731763 731855 731946]
```

有了股票格式、利率格式、时间格式，下面我们建立 CRR 型树的格式，代码如下：

```
>> CRRTree = crrtree(StockSpec, RateSpec, TimeSpec)
Warning:
RateSpec was not created with continuous compounding. Compounding will be
set to continuous while leaving discount factors unaltered. This will result
in the recalculation of the interest rates.
CRRTree =
       FinObj: 'BinStockTree'
       Method: 'CRR'
    StockSpec: [1x1 struct]
     TimeSpec: [1x1 struct]
```

```
     RateSpec: [1x1 struct]
         tObs: [0 0.2493 0.4986 0.7479 0.9972]
         dObs: [731582 731672 731763 731855 731946]
        STree: {1x5 cell}
      UpProbs: [0.5370 0.5370 0.5370 0.5370]
```

2. 建立 EQP 型二叉树函数

调用方式

```
EQPTree=eqptree(StockSpec,RateSpec,TimeSpec)
```

输入参数

同 crrtree 函数。

输出参数

```
EQPTree       %EQP 型二叉树
```

下面给出建立利率树形图方法。

第 1 步：设定标的资产格式。

在 MATLAB 中执行如下命令：

```
>> Sigma=0.20;
>> AssetPrice = 50;
>> DividendType = 'cash';   %红利以现金方式发放
>> DividendAmounts = [0.50; 0.50; 0.50; 0.50]; %每期发放红利金额
>> ExDividendDates = {'03-Jan-2003'; '01-Apr-2003'; '05-July-2003';
'01-Oct-2003'};%除息日
ExDividendDates =
    '03-Jan-2003'
    '01-Apr-2003'
    '05-July-2003'
    '01-Oct-2003'
>>StockSpec=stockspec(Sigma,AssetPrice,DividendType,DividendAmounts,…
ExDividendDates)
StockSpec =
             FinObj: 'StockSpec'
              Sigma: 0.2000
         AssetPrice: 50
       DividendType: 'cash'
    DividendAmounts: [4x1 double]
    ExDividendDates: [4x1 double]
```

第 2 步：确定无风险利率格式。

在 MATLAB 中执行如下命令：

```
>> RateSpec = intenvset('Rates', 0.05, 'StartDates','01-Jan-2003',
'EndDates', '31-Dec-2003')
RateSpec =
          FinObj: 'RateSpec'
     Compounding: 2
            Disc: 0.9519
           Rates: 0.0500
        EndTimes: 1.9945
      StartTimes: 0
        EndDates: 731946
      StartDates: 731582
   ValuationDate: 731582
           Basis: 0
    EndMonthRule: 1
```

第 3 步：确定树图时间展开格式。

在 MATLAB 中执行如下命令：

```
>> ValuationDate = '1-Jan-2003';
>> Maturity = '31-Dec-2003';
>> TimeSpec = eqptimespec(ValuationDate, Maturity, 4), %EQP 分为 4 个时间段。
TimeSpec =
          FinObj: 'BinTimeSpec'
   ValuationDate: 731582
        Maturity: 731946
      NumPeriods: 4
           Basis: 0
    EndMonthRule: 1
```

第 4 步：建立 EQP 树。

在 MATLAB 中执行如下命令：

```
>> EQPTree = eqptree(StockSpec, RateSpec, TimeSpec)
EQPTree =
       FinObj: 'BinStockTree'
       Method: 'EQP'
    StockSpec: [1x1 struct]
     TimeSpec: [1x1 struct]
     RateSpec: [1x1 struct]
         tObs: [0 0.2493 0.4986 0.7479 0.9972]
         dObs: [731582 731672 731763 731856 731946]
        STree: {1x5 cell}
      UpProbs: [0.5000 0.5000 0.5000 0.5000]
```

第 5 步：观察 EQP 树形图结构。

利用 treeviewer 函数观察 EQP 型二叉树结构，代码如下：

```
>> treeviewer(EQPTree)
```

可以看到树形图如图 6.3 所示，EQPTree 分成 4 个离散的时间段，单击树形图的每个节点，在右边的窗口就会显示节点内容。

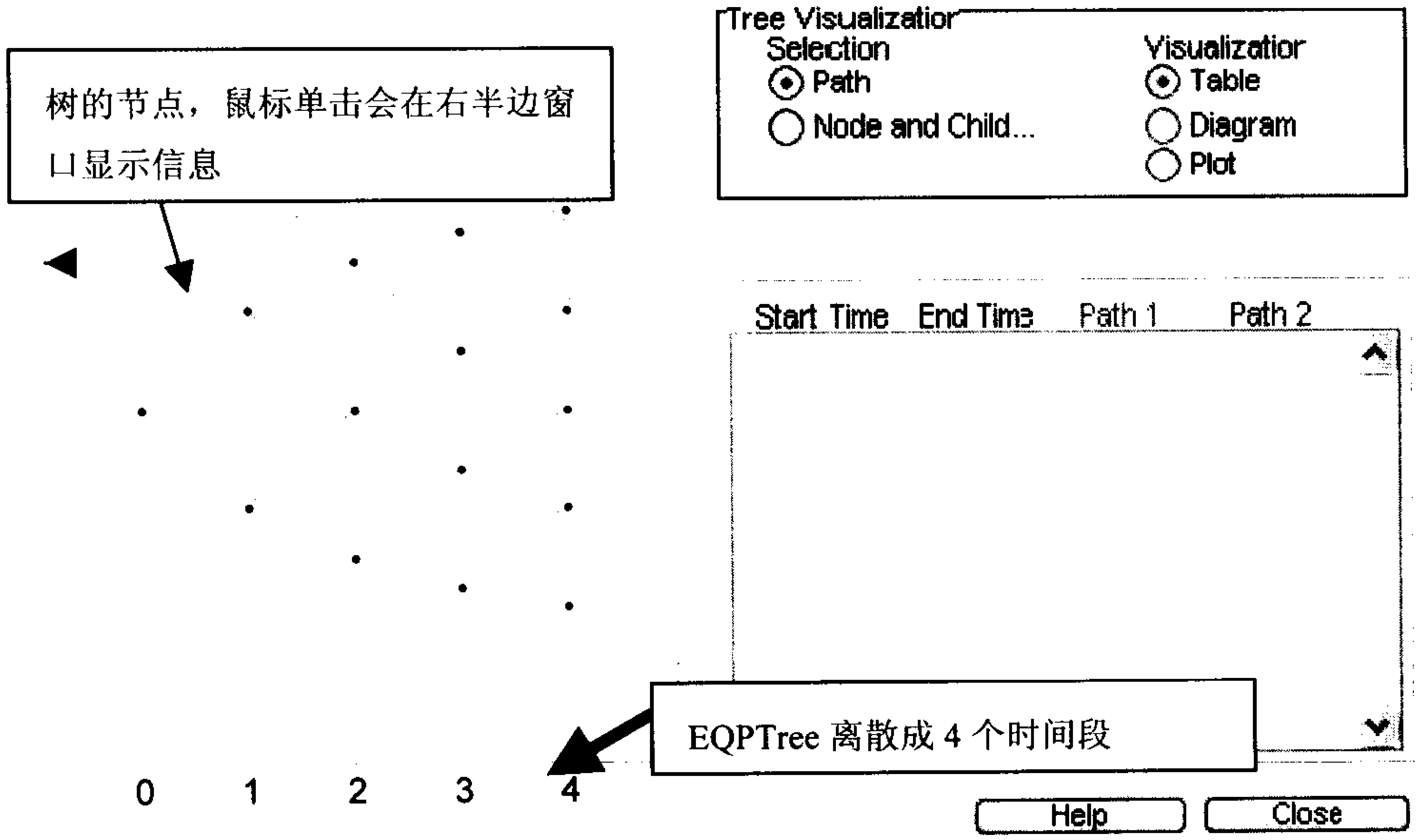


图 6.3　MATLAB 中二叉树示意图

右半边上方有两列单选按钮，其内容如下：

Path 表示按路径显示。

Node and Child 表示按本节点及下一个节点显示。

Table 表示用数表显示。

Diagram 表示用图表显示。

Plot 表示用图示法显示。

例如用鼠标分别依次单击最上面的节点，选择了一条路径，可以看到如图 6.4 所示的内容。

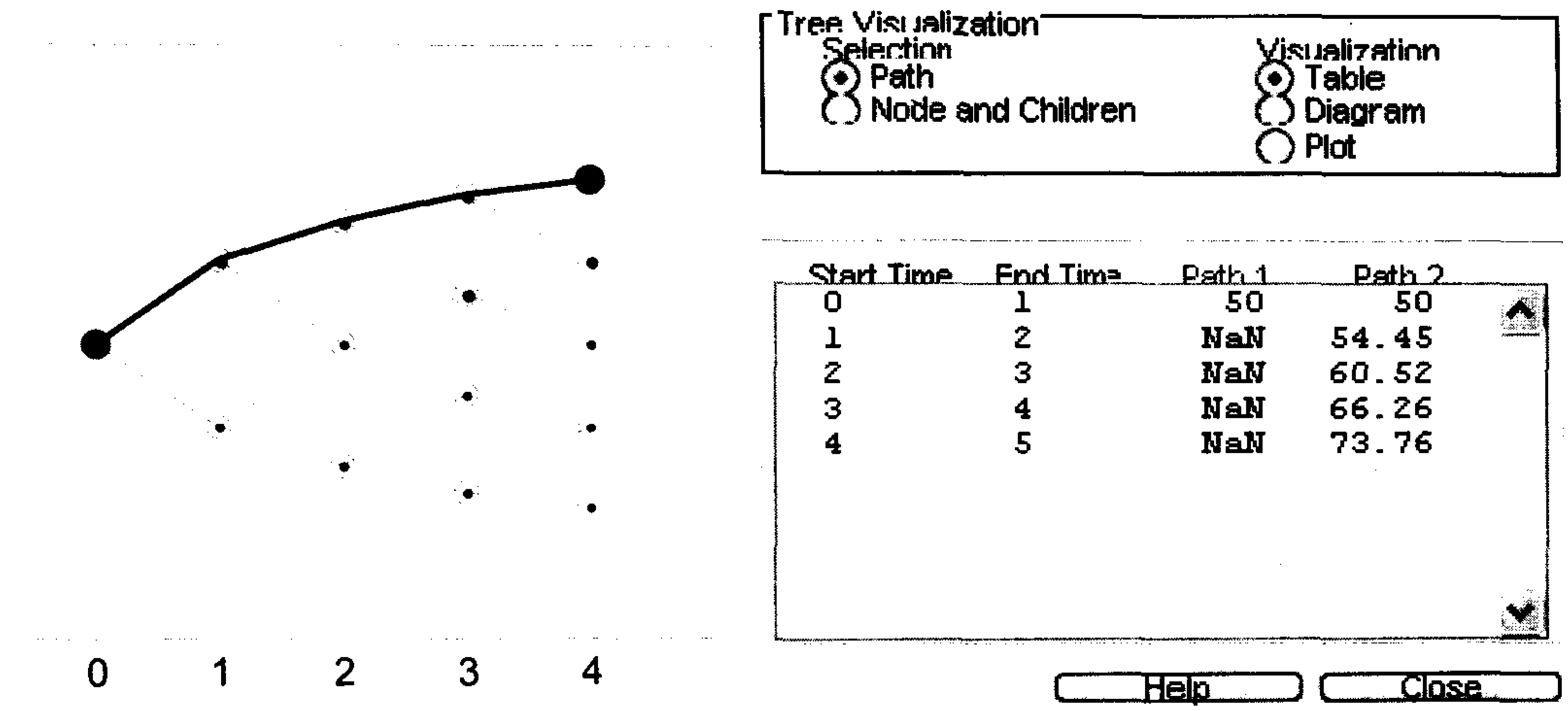


图 6.4　二叉树各路径节点的值

如果选中 Diagram 单选按钮，就会出现如图 6.5 所示的内容。

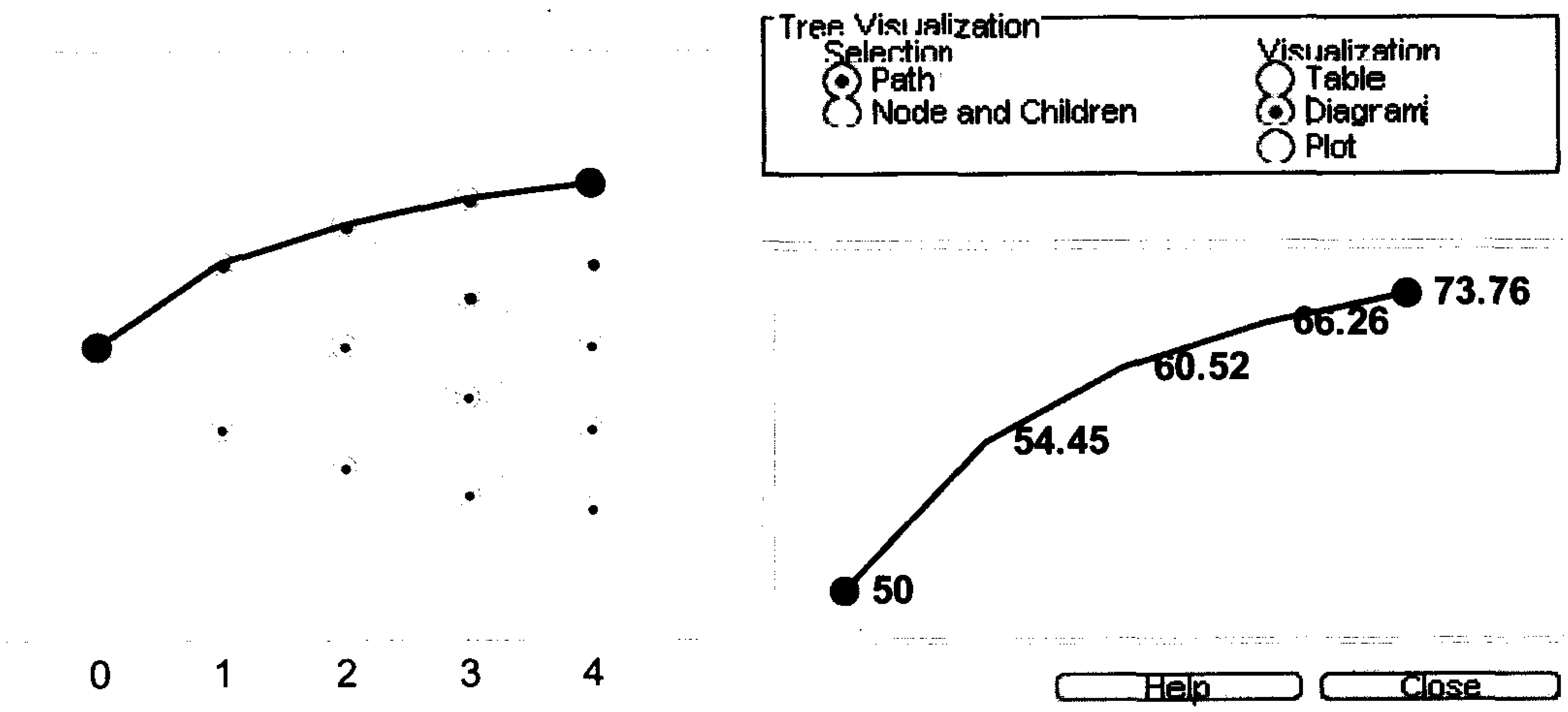


图 6.5　二叉树各节点的值

每个节点的数值标在路径上。

6.4.3　证券类衍生产品定价函数介绍

MATLAB 可以对亚式期权、障碍期权、复合期权和回望期权进行定价，方法为 CRR 模型与 EQP 模型两种，表 6.6 列出了各定价函数。

表 6.6 MATLAB 中的奇异期权定价函数表

期权名称	CRR 模型	EQP 模型
亚式期权(Asian Option)	asianbycrr	asianbyepq
障碍期权(Barrier Option)	barrierbycrr	barrierbyept
复合期权(Compound Option)	compoundbycrr	compoundbyepq
回望期权(Lookback Option)	lookbackbycrr	lookbackbyepq
欧式、美式、百慕大期权	optstockbycrr	optstockbyeqp

1. 亚式期权定价

1) CRR 模型对亚式期权定价

调用方式

```
Price = asianbycrr(CRRTree, OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, AvgType, AvgPrice, AvgDate)
```

输入参数

```
CRRTree        %CRR 型二叉树
OptSpec        %期权类型，如果是亚式看涨期权输入字符“Call”，如果是亚式看跌期权输
                入字符“Put”
Strike         %亚式期权执行价，如果是 NaN 表示执行价是浮动的
Settle         %结算日
ExerciseDates  %行权日期
AmericanOpt    %(Optional)如果 AmericanOpt=0，NaN，期权行权方式为美式，如果为 1，
                期权行权方式类似于欧式期权。默认值是欧式期权
AvgType        %(Optional)如果是算术平均输入字符“arithmetic”，默认值为算术平均，
                几何平均输入字符“geometric”
AvgPrice       %(Optional)计算标的资产的平均价，默认值为当前股价
AvgDate        %(Optional)开始计算平均价格的日期，默认值为结算日
```

输出参数

```
Price          %期权价格
```

下面我们调用 MATLAB 金融衍生工具箱中自带的 deriv 变量中的二叉树并进行定价计算，代码如下：

```
>> load deriv;
>> OptSpec = 'put';
>> Strike = NaN;
>> Settle = '01-Jan-2003';
>> ExerciseDates = '01-Jan-2004';
>> Price = asianbycrr(CRRTree, OptSpec, Strike, Settle,ExerciseDates)
```

```
Price =
   1.2177
```

2) EQP 模型对亚式期权定价

调用方式

```
Price = asianbyeqp(EQPTree, OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, AvgType, AvgPrice, AvgDate)
```

输入参数

同 asianbycrr 函数。

输出参数

同 asianbycrr 函数。

下面调用 MATLAB 金融衍生工具箱中自带的 deriv 变量中的二叉树并进行定价计算，代码如下：

```
>> load deriv;
>> OptSpec = 'put';
>> Strike = NaN;
>> Settle = '01-Jan-2003';
>> ExerciseDates = '01-Jan-2004';
>> Price = asianbyeqp(EQPTree, OptSpec, Strike, Settle, ExerciseDates)
   Price =
      1.2724
```

2. 障碍期权定价

1) CRR 模型对障碍期权定价

调用方式

```
Price= barrierbycrr(CRRTree, OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, BarrierSpec, Barrier, Rebate, Options)
```

输入参数

```
CRRTree          %CRR 型二叉树
OptSpec          %期权类型，如果是看涨期权输入字符“Call”，如果是看跌期权输入字符“Put”
Strike           %障碍期权执行价
Settle           %结算日
ExerciseDates    %行权日，对于欧式期权(AmericanOpt=0)，期权到期日即为行权日。对于美
                  式期权(AmericanOpt=1)而言，表示日期的期限
AmericanOpt      %(Optional)如果 AmericanOpt=0，NaN，期权行权方式为美式，如果为
                  AmericanOpt=1，表示期权行权方式类似于欧式期权。默认值是欧式期权
BarrierSpec      %障碍期权类型：
                 “UI”上涨入局期权(up-and-in option)
                 “UO”上涨出局期权(up-and-out option)
```

```
                    “DI”下跌入局期权(down-and-in option)
                    “DO”下跌出局期权(down-and-out option)
Barrier             %障碍期权的障碍值
Rebate              %(Optional)反馈金。对于敲出期权而言，在到达障碍时支付反馈金，缺失值
                     是 0
Options             %(Optional)由 derivset 函数构建的输出格式
```

输出参数

```
Price               %障碍期权的价格
```

【例 6-11】下面是利用 MATLAB 自带的数据进行演示。期权执行价是 105，结算日是 2003 年 1 月 1 日，行权日是 2006 年 1 月 1 日，期权行权方式为欧式看涨下跌入局，标的资产与无风险利率特征分别保存在 deriv 文件 CRRTree 与 OptSpec 中。

在 MATLAB 中执行如下命令：

```
>> load deriv;
>> OptSpec = 'Call';
>> Strike = 105;
>> Settle = '01-Jan-2003';
>> ExerciseDates = '01-Jan-2006';
>> AmericanOpt = 1;
>> BarrierSpec = 'UI';
>> Barrier = 102;
>> Price = barrierbycrr(CRRTree, OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, BarrierSpec, Barrier)
   Price =
        12.1272
```

2) EQP 模型对障碍期权定价

调用方式

```
[Price, PriceTree] = barrierbyeqp(EQPTree, OptSpec, Strike, ExerciseDates,
AmericanOpt, BarrierSpec, Barrier, Rebate, Options)
```

输入参数

同 barrierbyeqp 函数。

输出参数

同 barrierbyeqp 函数。

【例 6-12】我们对上一个例子，用 EQP 二叉树方法进行定价。

在 MATLAB 中执行如下命令：

```
>> load deriv;
>> OptSpec = 'Call';
>> Strike = 105;
```

```
>> Settle = '01-Jan-2003';
>> ExerciseDates = '01-Jan-2006';
>> AmericanOpt = 1;
>> BarrierSpec = 'UI';
>> Barrier = 102;
>> Price = barrierbyeqp(EQPTree, OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, BarrierSpec, Barrier)
Price =
          12.2632
```

3. 复合期权定价

1)　CRR 模型对复合期权定价

调用方式

```
Price=compoundbycrr(CRRTree, UOptSpec, UStrike, USettle, UExerciseDates,
UAmericanOpt, COptSpec, CStrike, CSettle, CExerciseDates, CAmericanOpt)
```

输入参数

CRRTree	%CRR 型期权二叉树
UOptSpec	%期权类型。UOptSpec=“call”时表示看涨期权，UOptSpec=“put”时表示看跌期权
UStrike	%执行价
USettle	%结算日期
UExerciseDates	%对于欧式期权(UAmericanOpt=0)表示行权日的期限，对于美式期权(UAmericanOpt=1)表示行权日的期限
UAmericanOpt	%标的期权行权方式，UAmericanOpt=0 表示欧式期权行权方式，
UAmericanOpt=1	%表示美式期权行权方式
COptSpec	%COptSpec=“call”复合期权是看涨期权，COptSpec=“put”表示复合期权是看跌期权
CStrike	%执行价
CSettle	%结算日
CExerciseDates	%行权日
CAmericanOpt	%当期权是欧式期权时 CAmericanOpt = 0；当期权是美式期权时 CAmericanOpt = 1

输出参数

price	%期权价格

2)　EQP 模型对复合期权定价

调用方式

```
[Price, PriceTree] = compoundbyeqp(EQPTree, UOptSpec, UStrike, USettle,
                     UExerciseDates, UAmericanOpt, COptSpec, CStrike,
                     CSettle, CExerciseDates, CAmericanOpt)
```

输入参数

同 compoundbycrr 函数。

输出参数

同 compoundbycrr 函数。

4. 回望期权定价

1) CRR 模型对回望期权定价

调用方式

```
[Price, PriceTree] = lookbackbycrr(CRRTree, OptSpec, Strike, Settle,
ExerciseDates, AmericanOpt)
```

输入参数

```
CRRTree        %CRR 型二叉树
OptSpec        %期权类型。UOptSpec=“call”时表示看涨期权，UOptSpec=“put”时表
                示看跌期权
Strike         %执行价，当执行价为浮动时取值为“NaN”
Settle         %结算日期
ExerciseDates  %期权行权日期，每列代表一个期权
AmericanOpt    %期权行权方式为美式(AmericanOpt=1)。AmericanOpt=0 时表示欧式期
                权；默认值是欧式期权
```

输出参数

```
Price          %回望期权价格
PriceTree      %价格树中每个节点的数据
```

2) EQP 模型对回望期权定价

调用方式

```
[Price, PriceTree] = lookbackbyeqp(EQPTree, OptSpec, Strike, Settle,
ExerciseDates, AmericanOpt)
```

输入参数

同 lookbackbycrr 函数。

输出参数

同 lookbackbycrr 函数。

5. 欧式(European option)、美式(American option)、百慕大期权(Bermuda option)定价

1) CRR 模型对欧式、美式、百慕大期权定价

调用方式

```
[Price, PriceTree] = optstockbycrr(CRRTree, OptSpec, Strike, Settle,
```

```
ExerciseDates, AmericanOpt)
```

输入参数

同 lookbackbycrr 函数。

输出参数

同 lookbackbycrr 函数。

2)　EQP 模型对欧式、美式、百慕大期权定价

调用方式

同 lookbackbycrr 函数。

输入参数

同 lookbackbycrr 函数。

输出参数

同 lookbackbycrr 函数。

6.4.4　证券类衍生产品输入格式

下面的定价方法是首先把产品特征用统一格式输入，然后调用树进行定价。

instadd 函数可以建立很多种衍生产品类型，是通用的定义函数，例如亚式期权、障碍期权、债券、戴帽期权、现金流、复合工具、固定收益工具、浮动收益工具、地板期权、回望期权、债券期权、掉期。MATLAB 中除了通用定义函数外，每个产品都有专门的定义函数，下面我们分别介绍这两种函数。

1．亚式期权(Asian instrument)输入格式

1)　利用 instadd 函数构造亚式期权输入格式

调用方式

```
InstSet = instadd('Asian', OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, AvgType, AvgPrice, AvgDate)
```

输入参数

```
OptSpec          %如果是看涨期权输入“Call”，如果是看跌期权输入“Put”
Strike           %执行价
Settle           %结算价
ExerciseDates    %行权日期
AmericanOpt      %美式期权
AvgType          %平均的方式
AvgPrice         %平均价
AvgDate          %平均日期
```

2) 利用 instasian 函数构造亚式期权格式

调用方式

```
InstSet = instasian(InstSet, OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, AvgType, AvgPrice, AvgDate)
```

输入参数

同 instadd 函数。

2. 障碍期权(Barrier instrument)输入格式

1) 利用 instadd 函数构造障碍期权输入格式

调用方式

```
InstSet = instadd('Barrier', OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt, BarrierType, Barrier, Rebate)
```

输入参数

```
OptSpec        %如果是看涨期权输入“Call”，看跌期权输入“Put”。
Strike         %执行价
Settle         %结算日
ExerciseDates  %行权日
AmericanOpt    %AmericanOpt = 0 表示欧式期权行权方式，AmericanOpt = 1 表示美式
                期权行权方式
BarrierType    %障碍期权类型。具体如下：
                “UI”  表示上涨入局期权(Up Knock In)
                “UO”  表示上涨出局期权(Up Knock Out)
                “DI”  表示下跌入局期权(Down Knock In)
                “DO”  表示下跌出局期权(Down Knock Out)
Barrier        %障碍期权的障碍值
Rebate         %(Optional)返回值
```

2) 利用 instbarrier 函数构造障碍期权输入格式

输入参数

同 instadd 函数。

3. 建立复合期权输入方式

1) 利用 instadd 函数构造复合期权输入方式

调用方式

```
InstSet = instadd('Compound', UOptSpec, UStrike, USettle, UExerciseDates,
UAmericanOpt,COptSpec, CStrike, CSettle, CExerciseDates, CAmericanOpt)
```

输入参数

```
UOptSpec          %标的期权类型，看涨期权输入“Call”，看跌期权输入“Put”
UStrike           %标的期权执行价
USettle           %标的期权结算日期
UExerciseDates    %标的期权行权日期
UAmericanOpt      %标的期权行权方式，UAmericanOpt=0 表示欧式期权行权方式，
                   UAmericanOpt=1 表示美式期权行权方式
COptSpec          %复合期权类型，看涨期权输入“Call”，看跌期权输入“Put”
CStrike           %复合期权执行价
CSettle           %复合期权结算日期
CExerciseDates    %复合期权行权日期
CAmericanOpt      %复合期权行权方式。CAmericanOpt=0 表示欧式期权行权方式，
                   CAmericanOpt=1 表示美式期权行权方式
```

2) 利用 instcompound 函数构造复合期权输入方式

输入参数

同 instadd 函数。

4. 回望期权(Lookback instrument)输入格式

1) 利用 instadd 函数构造回望期权输入格式

调用方式

```
InstSet = instadd('Lookback', OptSpec, Strike, Settle, ExerciseDates,
AmericanOpt)
```

输入参数

```
OptSpec          %如果是看涨期权输入“Call”，如果是看跌期权输入“Put”
Strike           %执行价
Settle           %结算日期
ExerciseDates    %行权日
AmericanOpt      %AmericanOpt = 0 表示欧式期权行权方式，AmericanOpt = 1 表示美式
                  期权行权方式
```

2) 利用 instlookback 函数构造回望期权输入格式

输入参数

同 instadd 函数。

5. 普通股票期权(Stock option instrument)输入格式

1) 利用 instadd 函数构造普通股票期权输入格式

调用方式

```
InstSet = instadd('OptStock', OptSpec, Strike, Settle, Maturity,
AmericanOpt)
```

输入参数

```
OptSpec        %如果是看涨期权输入“Call”，如果是看跌期权输入“Put”
Strike         %执行价
Settle         %结算日期
Maturity       %到期日
AmericanOpt    %AmericanOpt = 0表示欧式期权行权方式，AmericanOpt = 1表示美式
                期权行权方式
```

2) 利用 instoptstock 函数构造普通股票期权输入格式

输入参数

同 instadd 函数。

6.4.5 证券类衍生产品定价函数

1. 利用 EQP 模型对衍生产品定价

调用方式

```
[Price, PriceTree] = eqpprice(EQPTree, InstSet, Options)
```

输入参数

```
EQPTree       %EQP 价格树
InstSet       %期权类衍生产品特征
Options       %(Optional)控制期权
```

输出参数

同上。

下面我们利用 MATLAB 自带的数据对利率类产品进行定价，代码如下：

```
>> load deriv.mat; % 载入MATLAB自带的数据文件deriv.mat
>> EQPSubSet = instselect(EQPInstSet, 'OptSpec', 'put');% 选择所要定价的利率
产品
>> instdisp(EQPSubSet) %显示利率产品特征
Index Type     OptSpec Strike Settle       ExerciseDates  AmericanOpt Name
Quantity
1     OptStock put     105    01-Jan-2003   01-Jan-2006    0       Put1    5
Index Type  OptSpec Strike Settle        ExerciseDates AmericanOpt AvgType
2     Asian put      110    01-Jan-2003   01-Jan-2006    0      arithmetic
3     Asian put      110    01-Jan-2003   01-Jan-2007    0      arithmetic
AvgPrice AvgDate  Name    Quantity
NaN    NaN     Asian1  4
NaN    NaN     Asian2  6
```

下面我们调用 EQP 树对上述 3 种产品进行定价，代码如下：

```
>> [Price, PriceTree] = eqpprice(EQPTree, EQPSubSet)
Price =
   2.6375
   4.7444
   3.9178
PriceTree =
   FinObj: 'BinPriceTree'
   PTree: {[3x1 double]  [3x2 double]  [3x3 double]  [3x4 double]  [3x5
double]}
   tObs: [0 1 2 3 4]
   dObs: [731582 731947 732313 732678 733043]
```

上述 3 种产品的价格分别是 2.6375、4.7444、3.98178。下面我们调用 treeviewer 函数观看 PriceTree。

首先我们观看 EQPTree 的结构，代码如下：

```
>> treeviewer(PriceTree)
```

显示的价格树结构如图 6.6 所示。

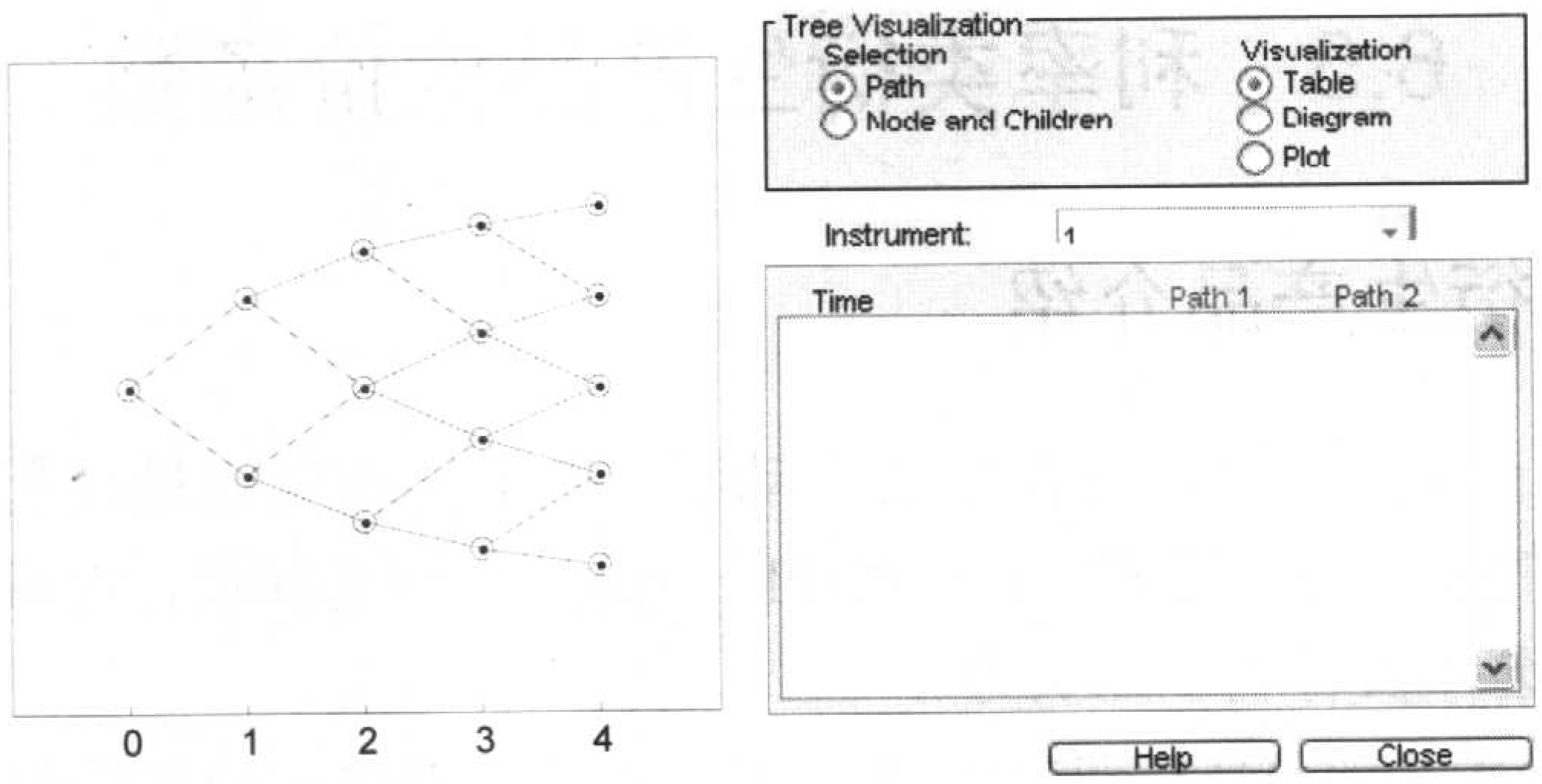


图 6.6　EQP 型二叉树结构

同样我们观察价格树 PriceTree 的现金流。PriceTree 含有 3 个产品价格树，在 Instrument 下拉列表框中可以选择其中一个产品的价格树，例如我们选择第 1 个产品的价格树。

如图 6.7 所示依次单击各个节点，每个节点上的现金流就会显示出来。

2. 利用 EQP 模型对衍生产品定价

调用方式

```
[Price, PriceTree] = crrprice(EQPTree, InstSet, Options)
```

输入参数

同 eqprice 函数。

输出参数

同 eqprice 函数。

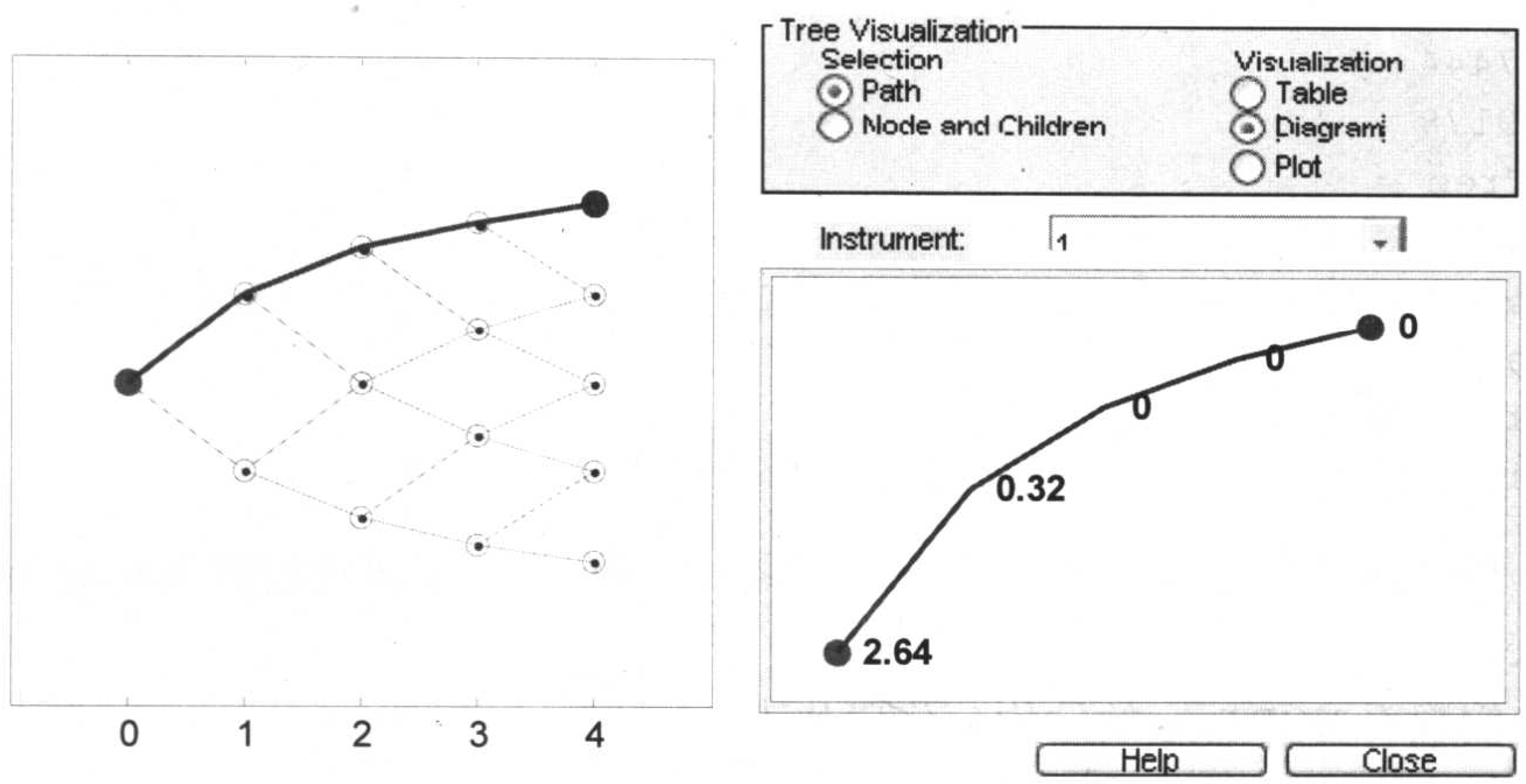


图 6.7　EQP 型二叉树各节点的现金流值

6.5　利率类衍生产品定价函数

6.5.1　利率类衍生产品介绍

利率顶(Cap)是一个选择权，它限制了浮动利率负债所支付的最高利率，很明显，这一选择权有利于债券的发行方。同理，利率底(Floor)也是一个选择权，它赋予浮动利率负债的最低支付利率，该选择权有利于购买方。

利率互换(Interest Swap)是指双方同意未来时期内，根据同种货币的相同名义本金交换现金流。交易一方的现金流根据浮动利率计算出来，另一方的现金流根据固定利率计算出来。在普通互换中，固定利率支付者，被称为买了互换(Buy the Swap)，它建立了互换多头；浮动利率一方建立了互换空头，支付浮动利率，获取固定收益，称为卖了互换(Sell the Swap)。

固定收益票据(Fixed-rate note)是一种长期债券，票息是固定的，而且要到期支付。

浮动利率票据(Floading-rate note)类似于债券，不同之处在于其利率每隔一段时间就要参照某个基准利率进行调整。

债券期权(Bond option)指买方可以在商定的某个日期以某个价格购买某一债券。

6.5.2　利率模型介绍

单因子利率模型就是将整条收益曲线看作单一变量函数，即短期利率函数。建立单因

子利率模型基于下列假设：当短期利率发生变化时(短期利率是无风险回报率)，不同期限债券的价格彼此相关。

1. Ho-Lee 模型二叉树定价

Ho-Lee 模型是由华裔学者 Thomas Ho 与 San-Bin Lee 创立，该模型假定利率(r)服从下面的形式：

$$\mathrm{d}r = \mu(t)\mathrm{d}t + \sigma \mathrm{d}w \tag{6.22}$$

其中，$\mu(t)$ 是 t 时刻利率趋势项，σ 是利率波动标准差。

Ho-Lee 模型中趋势变量 μ 是时间依赖型的，随着时间不同利率趋势也是不断变化的，例如第一个月趋势为 10 个基点，第二个月可能就变为 5 个基点。

在 Ho-Lee 模型中，利率变化遵循二项式结构，如图 6.8 所示。

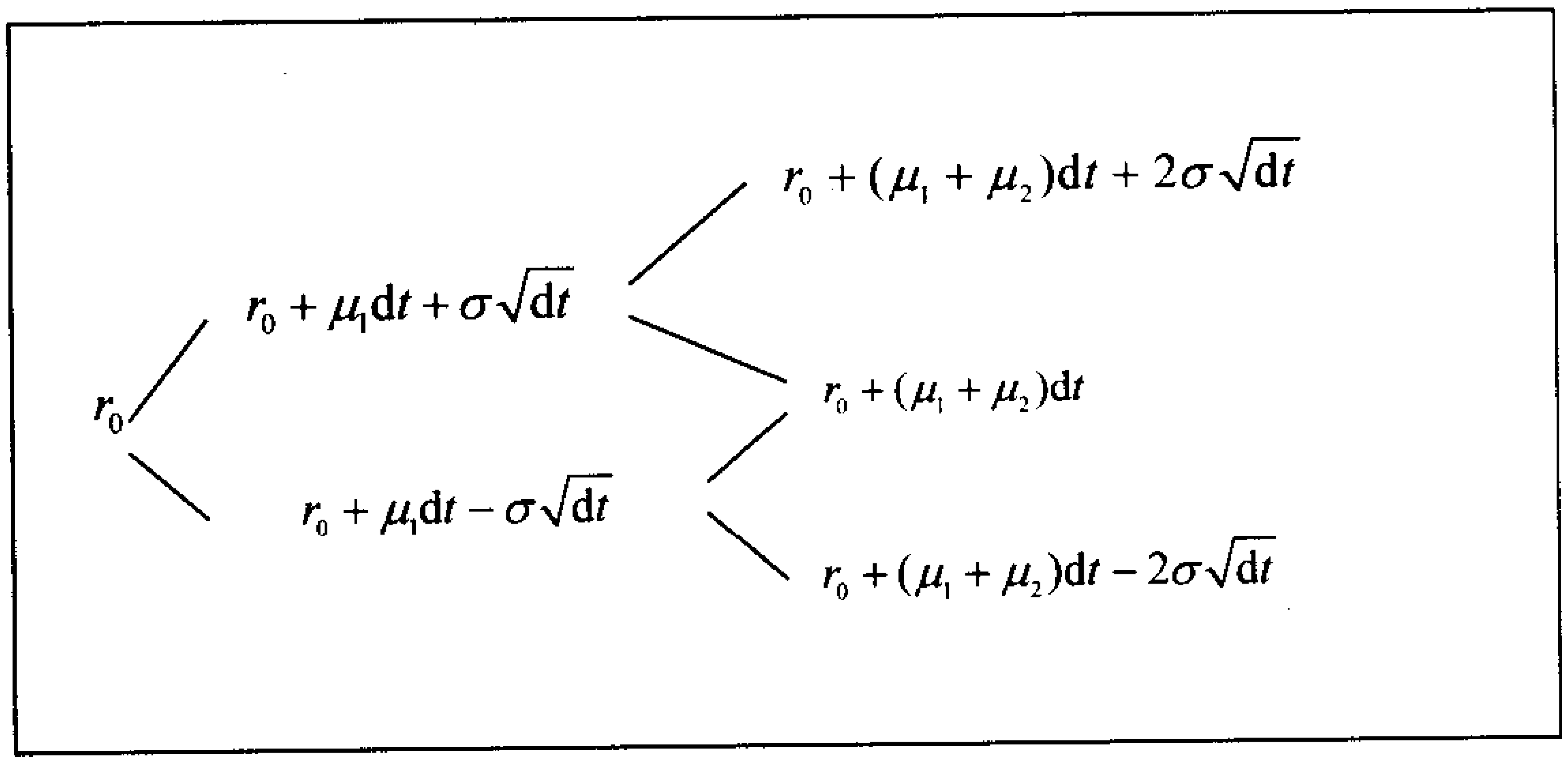


图 6.8　Ho-Lee 模型

该模型表明，下一期短期利率为本期利率加上某个常数与时间的乘积，再加上或者减去另一个常数乘以时间段的平方根，不论利率是上升还是下降，下一期利率都要加上 $\mu_1\mathrm{d}t$，这是短期利率趋势变化。

接下来的问题是如何确定 μ_1，μ_2 和 σ。

σ 数值可以通过历史波动率进行估计，也可以采用隐含波动率确定，为了确保定价模型不存在套利机会，模型参数 μ_1，μ_2 和 σ 必须能使零息债券定价与市场价格相一致。下面我们用一个例子来说明。

【例 6-13】假定即期利率为 6%，利率波动的年标准差为 0.5%，2 年期零息债券的市场价格为 88.58，3 年期零息债券的价格为 82.47，下面我们构造利率树，并且利用该利率树对相关产品进行定价。

首先，2 年期零息债券价格变化如图 6.9 所示。

求解下列方程：

$$\left(\frac{1}{2}\times\frac{100}{1+6\%+\mu_1+0.5\%}+\frac{1}{2}\times\frac{100}{1+6\%+\mu_1-0.5\%}\right)\div 1.06=88.58$$

求解得 $\mu_1=0.5\%$，这样形成利率二叉树图，如图 6.10 所示。

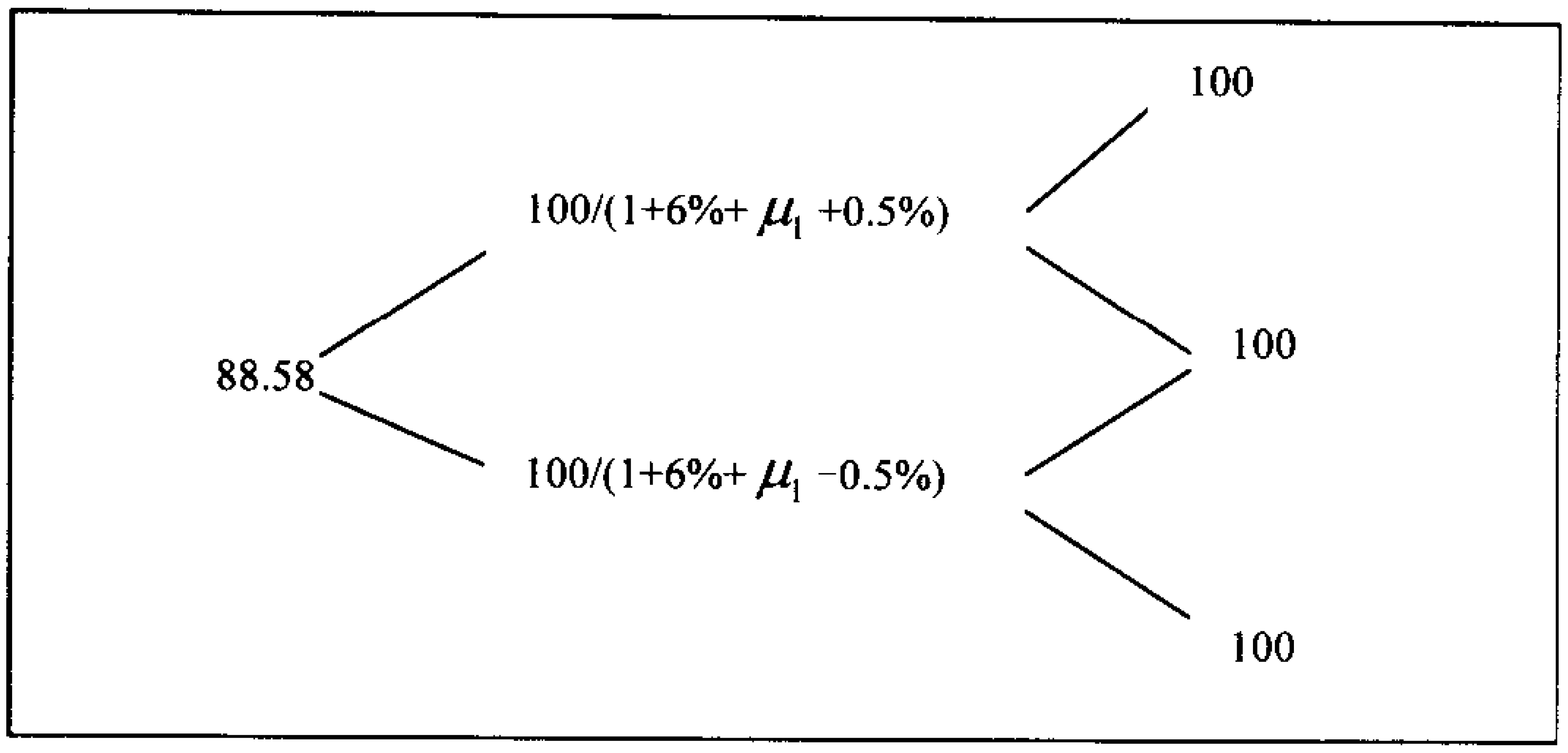


图 6.9　2 年期零息债券价格

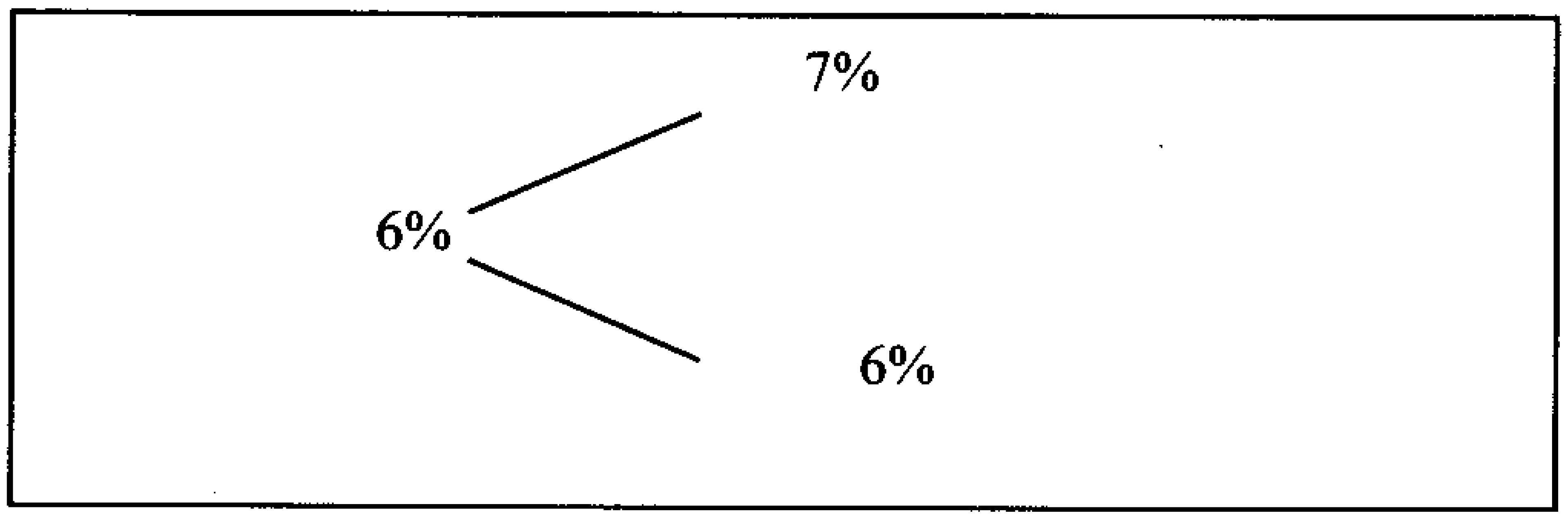


图 6.10　二叉树节点上的利率

将利率二叉树图延伸一年，利率二叉树如图 6.11 所示。

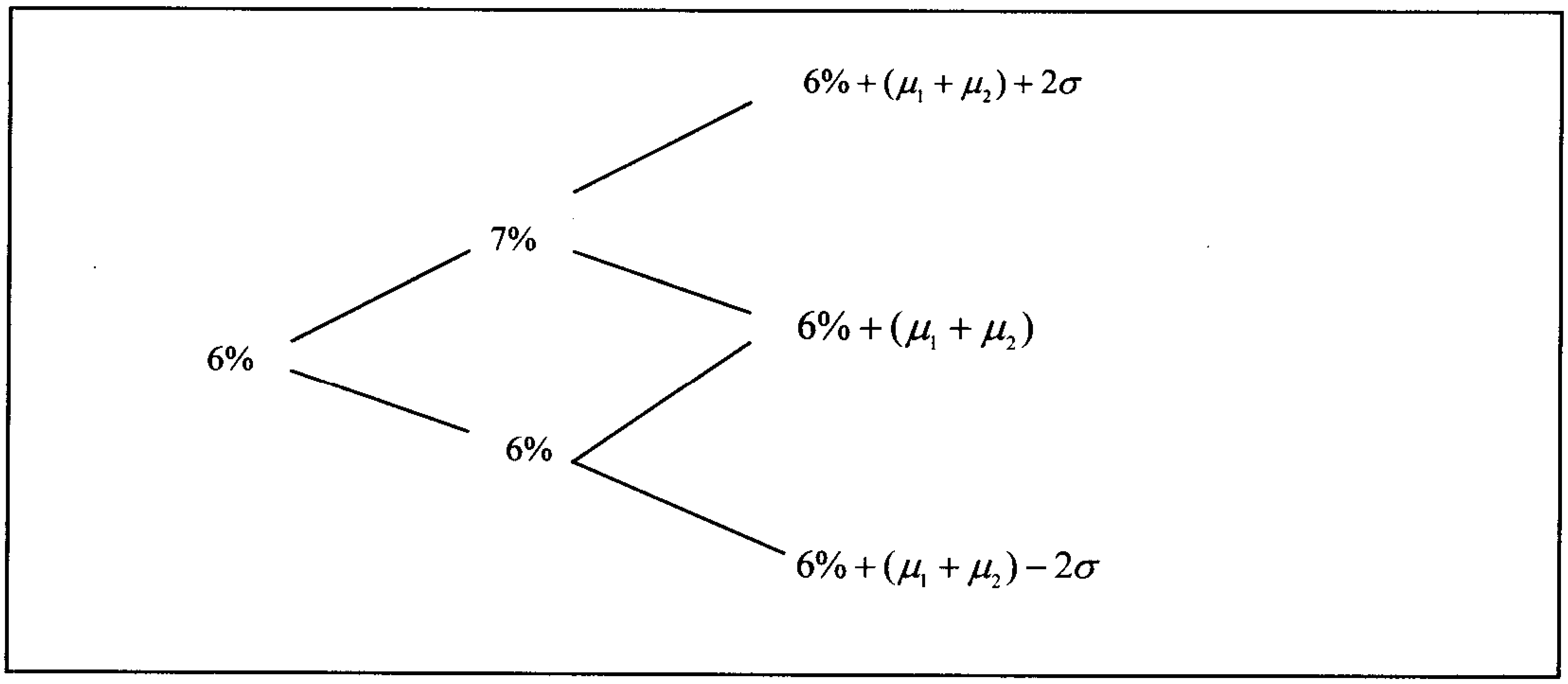


图 6.11　2 年期利率二叉树结构

由于 $\mu_1 = 0.5\%$， $\sigma = 0.5\%$，3 年期二叉树图如图 6.12 所示。

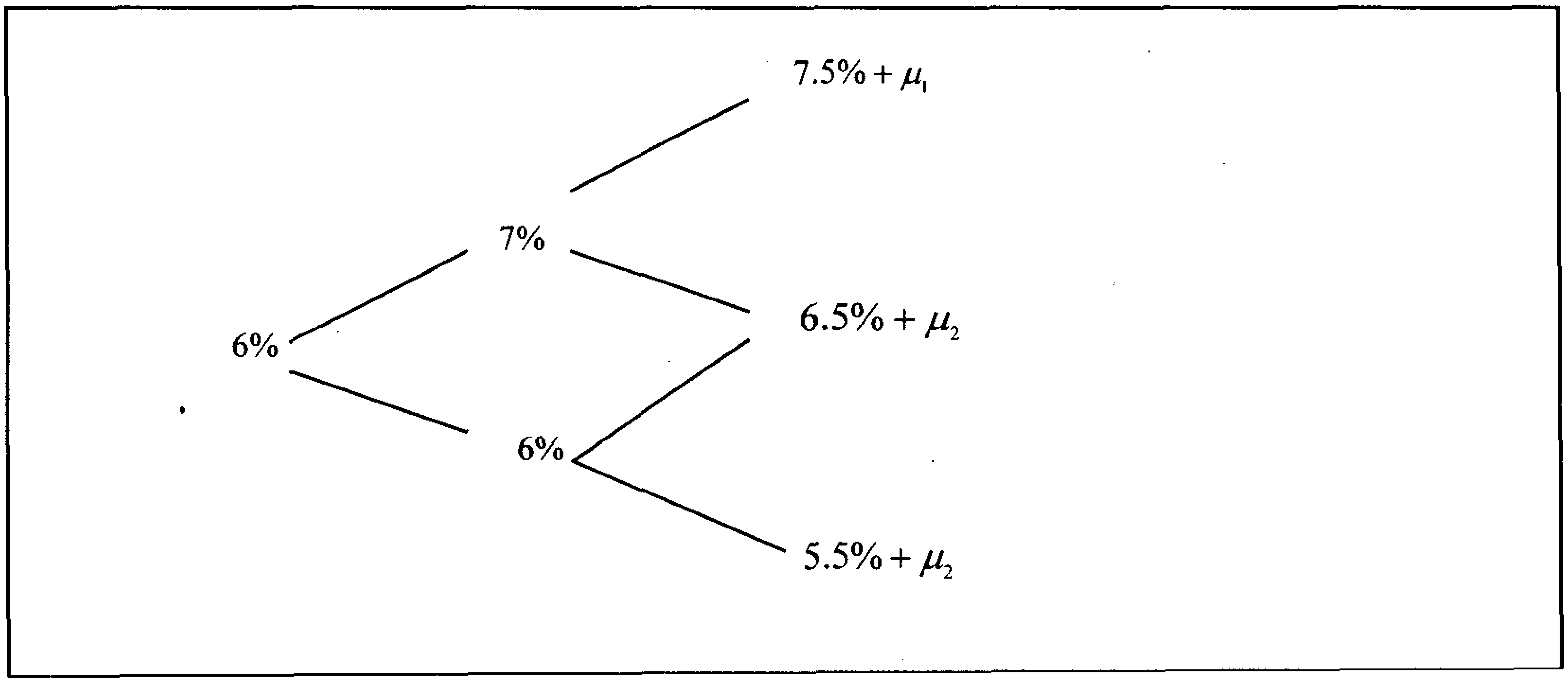


图 6.12　3 年期二叉树结构

假定 3 年期零息债券的市场价格为 88.47 元，那么 3 年期零息债券的二叉树图如图 6.13 所示。

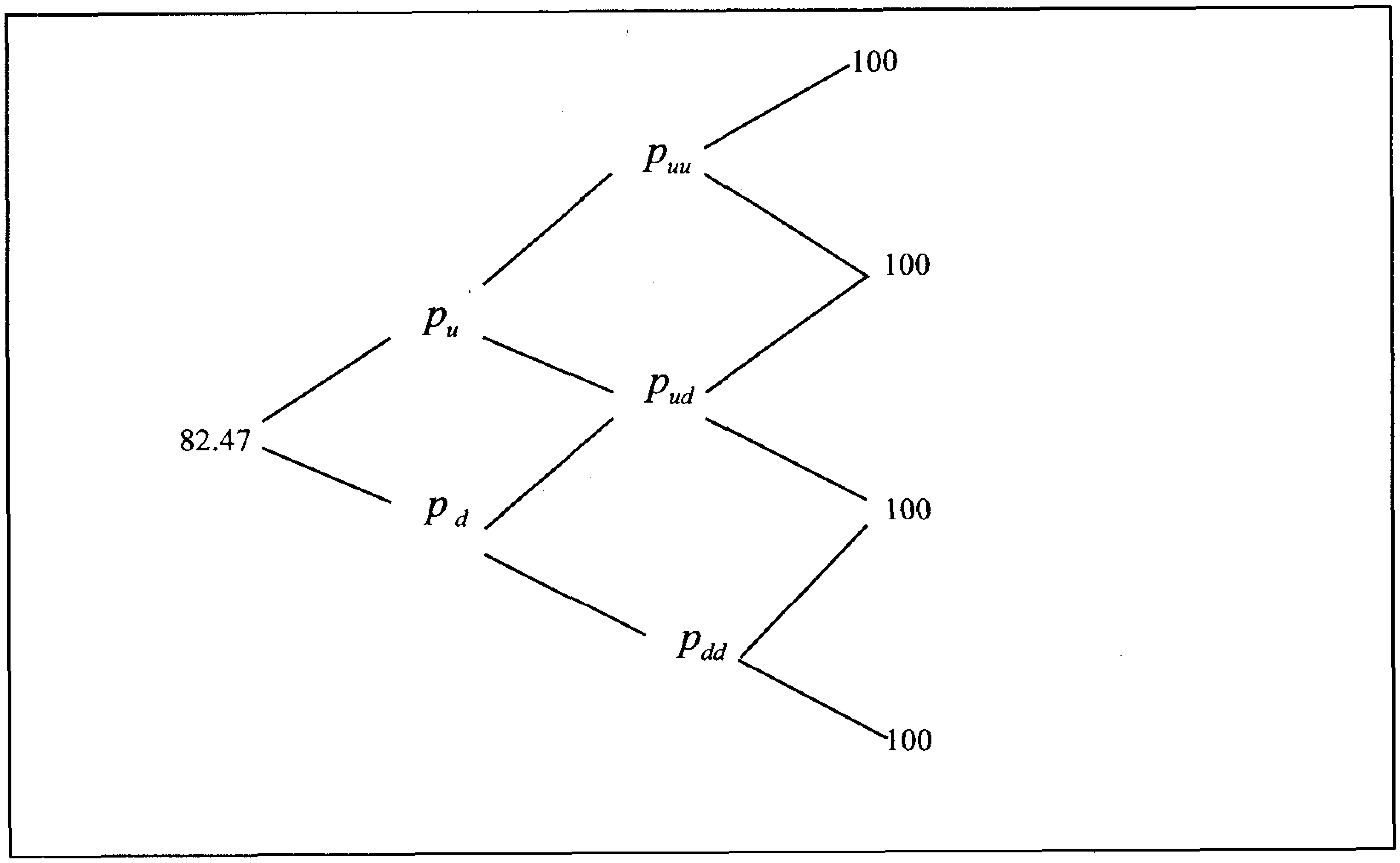


图 6.13　3 年期二叉树价格示意图

其中

$$p_{uu}=\frac{100}{1+7.5\%+\mu_2}$$

$$p_{ud}=\frac{100}{1+6.5\%}$$

$$p_{dd}=\frac{100}{1+5.5\%+\mu_2}$$

$$p_u=\frac{0.5p_{uu}+0.5p_{ud}}{1+7\%}$$

$$p_d=\frac{0.5p_{uu}+0.5p_{ud}}{1+6\%}$$

$$p=82.47=\frac{0.5p_u+0.5p_d}{1+6\%}$$

求解得 $\mu_2=0.6\%$。那么 3 年期利率树如图 6.14 所示。

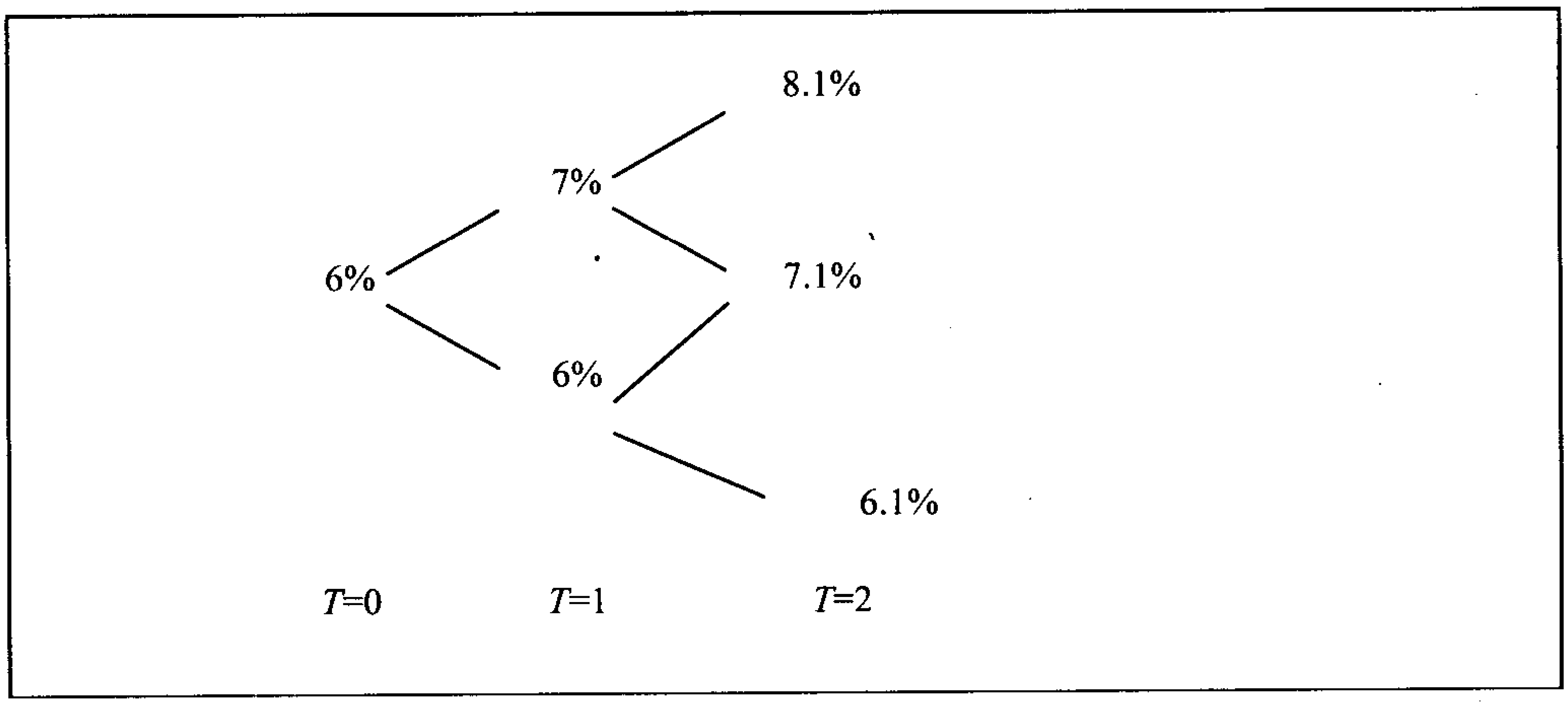


图 6.14　3 年期利率二叉树示意图

图 6.15 是利用利率树对 3 年期贴现利率为 5.25%的债券进行定价的示意图。

在 T=2 节点上，当短期利率为 8.1%时，该附息债权价值为 97.36，计算过程如下：

$$97.36=\frac{(100+5.25)\times 0.5+(100+5.25)\times 0.5}{1+8.1\%}$$

在 T=2 节点上，当短期利率为 7.1%时，该附息债权价值为 98.27，计算过程如下：

$$98.27=\frac{(100+5.25)\times 0.5+(100+5.25)\times 0.5}{1+7.1\%}$$

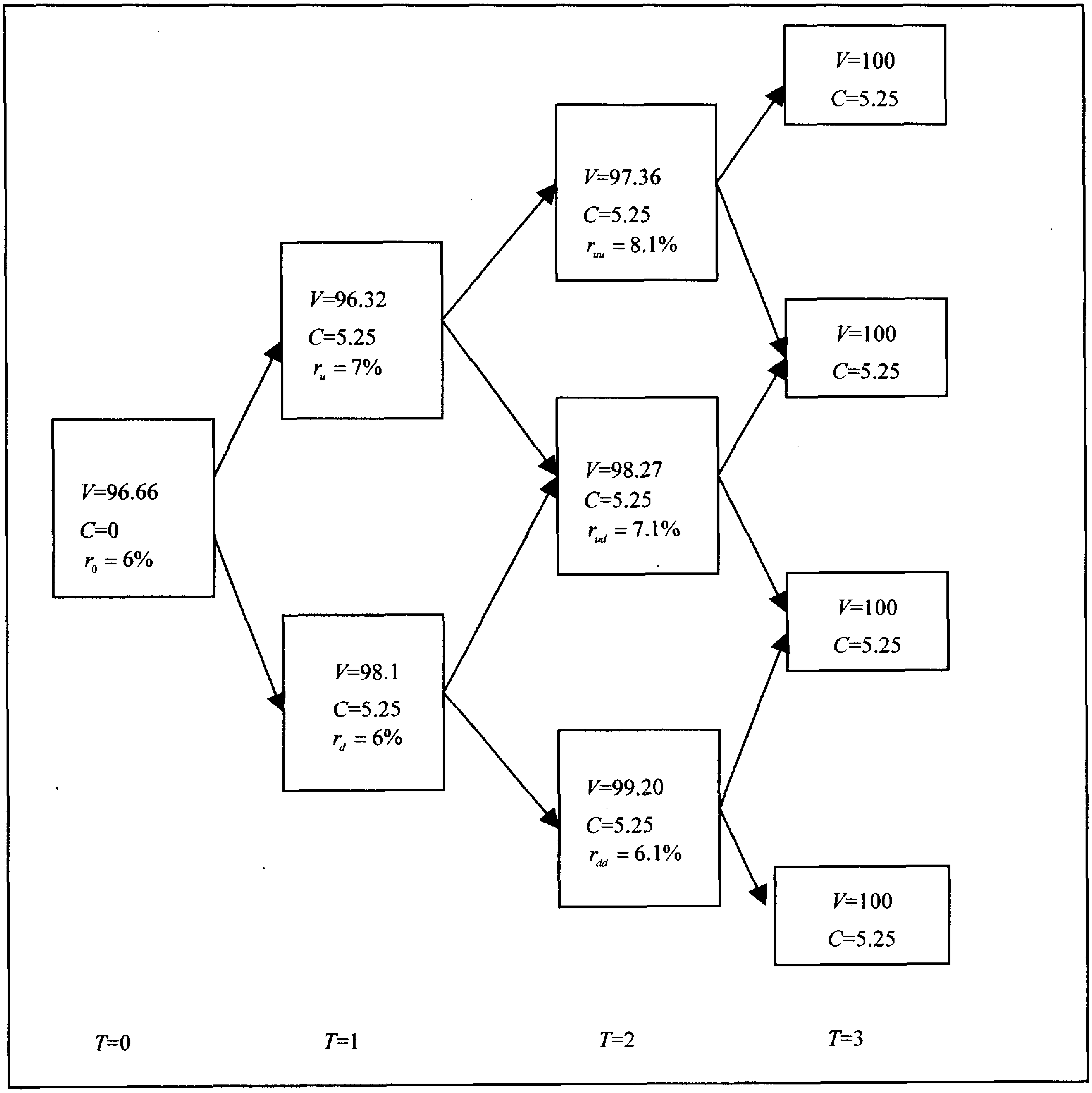


图 6.15　利率二叉树对衍生产品定价示意图

在 T=2 节点上，当短期利率为 6.1%时，该附息债权价值为 99.20，计算过程如下：

$$99.20=\frac{(100+5.25)\times 0.5+(100+5.25)\times 0.5}{1+6.1\%}$$

在 T=1 节点上，当短期利率为 7%时，该附息债权价值为 96.32，计算过程如下：

$$96.32=\frac{(97.36+5.25)\times 0.5+(99.20+5.25)\times 0.5}{1+7\%}$$

在 T=1 节点上，当短期利率为 6%时，该附息债权价值为 98.1，计算过程如下：

$$98.1 = \frac{(98.27 + 5.25) \times 0.5 + (99.20 + 5.25) \times 0.5}{1 + 6\%}$$

在 T=0 节点上，当短期利率为 6%时，该附息债权价值为 96.66，计算过程如下：

$$96.66 = \frac{(96.32 + 5.25) \times 0.5 + (98.1 + 5.25) \times 0.5}{1 + 6\%}$$

这样 3 年期附息债权的价格为 96.66。

2. Hull-White(1990)模型

Hull 和 White 在 1990 年探讨了 Vasicek(Vasicek 模型)与 Cox、Ingersoll 和 Ross 模型(CIR 模型)的扩展情况，并且提供了一个利率结构模型(HW 模型)。利率模型形式如下：

$$\mathrm{d}r = (\theta(t) - \alpha r)\mathrm{d}t + \sigma \mathrm{d}W \tag{6.23}$$

其中，r 是短期利率，α, σ 是常数，$\theta(t)$ 是一个与时间相关的函数。

此模型几乎可以涵盖所有无套利单因子模型，式(6.23)也可以写成

$$\mathrm{d}r = \alpha\left(\frac{\theta(t)}{\alpha} - r\right)\mathrm{d}t + \sigma \mathrm{d}W \tag{6.24}$$

3. Black-Karasinski(1991)模型

通常情况下，在建立短期利率模型时不希望出现零利率、负利率，BK 模型采用对数收益率，可以避免这种情况，其形式如下：

$$\mathrm{d}\ln(r) = (\theta(t) - k(t)\ln(r))\mathrm{d}t + \sigma(t)\mathrm{d}W \tag{6.25}$$

BK 模型中使用对数正态分布，对数正态分布由均值、方差刻画，而且均值与方差随时间变化而变化，我们进一步地可以将式(6.25)改写为

$$\mathrm{d}\ln(r) = \varphi(t)(\mu(t) - \ln(r))\mathrm{d}t + \sigma(t)\mathrm{d}W \tag{6.26}$$

其中，$\varphi(t)$ 表示均值回归速度，$\sigma(t)$ 表示短期波动率，$\mu(t)$ 表示瞬时利率平均值。

4. Black-Derman-Toy(1990)模型

Black、Derman 和 Toy 模型(简称 BDT 模型)假设瞬时利率为对数正态分布，模型中除了初始利率期限结构，也把波动率期限结构视为输入变量。此时由于利率波动率是时间的函数，使得利率动态变化具有均值回归特征。BDT 模型形式如下：

$$\mathrm{d}\ln(r) = \left(\theta(t) + \frac{\sigma'(t)}{\sigma(t)}\ln r\right)\mathrm{d}t + \sigma(t)\mathrm{d}W$$

如果假设 $u = \ln r$，可以得到下面等式：

$$\mathrm{d}u = \left(\theta(t) + \frac{\sigma'(t)}{\sigma(t)}u\right)\mathrm{d}t + \sigma(t)\mathrm{d}W$$

同样，如果在 BK 模型中设 $-\phi(t)=\dfrac{\sigma'(t)}{\sigma(t)}$，也可以得到上面的模型。

5. Heath-Jarrow-Morton(1992)模型

Heath、Jarrow 和 Morton(1992)提出了远期利率模型(简称 HJM 模型)，HJM 模型与其他模型的不同之处在于波动率与整个期间的远期利率曲线相关，他们假设实时远期利率 $f(t,T)$ 服从方程

$$\mathrm{d}f(t,T)=\alpha(t,T)\mathrm{d}t+\sum_{i=1}^{N}\sigma_i(t,T,f(t,T))\mathrm{d}W_i(t) \tag{6.27}$$

写成向量形式为

$$\mathrm{d}f=\alpha(t)\mathrm{d}t+\sigma(t)\mathrm{d}\boldsymbol{W} \tag{6.28}$$

这里，$\sigma(t,T)=(\sigma_1(t,T),\cdots,\sigma_N(t,T))$ 是一个 $1\times N$ 维向量，$\boldsymbol{W}=(w_1,w_2,\cdots,w_N)$ 是一个 $1\times N$ 维布朗运动，彼此之间互不相关。

HJM 模型属多因子模型，对利率期限结构的波动描述更为完整，但是该模型需要输入很多数据，增加了数据取得的成本。

6.5.3　利率类衍生产品输入格式

MATLAB 衍生类工具箱可以对利率类产品定价，包括现金流、浮动收益工具、地板期权、回望期权、债券期权等，但首先需要对这些金融工具的特征进行说明。下面是利率类衍生产品的输入格式。

1. 现金流(CashFlow Instrument)输入格式(参见 instcf)

1)　利用 instadd 函数构造现金流输入格式

调用方式

```
InstSet = instadd('CashFlow', CFlowAmounts, CFlowDates, Settle, Basis)
```

输入参数

```
CFlowAmounts     %现金流金额
CFlowDates       %现金流发生日期
Settle①          %结算日
Basis            %(Optional)计息期间的计算方法。0 = actual/actual (默认值)，1 =
                  30/360 (SIA)，2 = actual/360，3 = actual/365，4 = 30/360 (PSA)，
```

① 结清证券或商品交易往来账项的日期，公司债券和股票的结算日期，通常为交易日期(trade date)后的第 5 个营业日。

```
5 = 30 360 (ISDA), 6 = 30/360 (European), 7 = actual/365
(Japanese)
```

2) 利用 instcf 函数构造现金流输入格式

调用方式

```
InstSet = instcf(InstSet, CFlowAmounts, CFlowDates, Settle, Basis)
```

输入参数

同 instadd 函数。

【例 6-14】 有两个现金流产品，其现金流发放日期如表 6.7 所示。

表 6.7 各金融产品现金流

品种 1 现金流				
日期	2006 年 1 月 1 日	—	2008 年 1 月 1 日	2009 年 1 月 1 日
现金流	5	—	5.5	105
品种 2 现金流				
日期	2006 年 1 月 1 日	2006 年 12 月 23 日	2008 年 1 月 1 日	2011 年 9 月 28 日
现金流	5	0	6	105

结算日期为 2006 年 1 月 6 日，计息期间法则为 Actual/Actual，则这两种现金流产品计算方式如下：

```
>> CFlowAmounts =[5 NaN 5.5 105; 5 0 6 105];
>> CFlowDates = [732678, NaN, 733408,733774;732678, 733034, 733408, 734774];
>> Settle = '06-Jan-2006';
>> Basis=0;
>> InstSet = instadd('CashFlow', CFlowAmounts, CFlowDates, Settle, Basis)
InstSet =
        FinObj: 'Instruments'
    IndexTable: [1x1 struct]
          Type: {'CashFlow'}
     FieldName: {{4x1 cell}}
    FieldClass: {{4x1 cell}}
     FieldData: {{4x1 cell}}
>> instdisp(InstSet)    %显示 InstSet
Index   Type       CFlowAmounts
1     CashFlow   5          NaN          5.5            105
2     CashFlow   5           0            6            105

CFlowDates
01-Jan-2006  NaN          01-Jan-2008   01-Jan-2009
01-Jan-2006  23-Dec-2006    01-Jan-2008   28-Sep-2011
```

```
Settle         Basis
06-Jan-2006   0
     06-Jan-2006   0
```

2. 债券工具(Bond instrument)输入格式

1)　利用 instadd 函数构造债券输入格式

调用方式

```
InstSet = instadd('Bond', CouponRate, Settle, Maturity, Period, Basis,
EndMonthRule, IssueDate, FirstCouponDate, LastCouponDate, StartDate, Face)
```

输入参数

```
CouponRate       %票息率
Settle           %结算日
Maturity         %到期日
Period           %(Optional)计息期间
Basis            %(Optional) 应计天数计算方式
EndMonthRule     %(Optional)月末法则
IssueDate        %(Optional)发行日期
FirstCouponDate  %(Optional)第一次支付票息的日期
LastCouponDate   %(Optional)最后一次支付票息的日期，一般指距离到期日(Maturity)前一
                  次支付票息的日期
StartDate        %可以忽略
Face             %(Optional)面值，默认值为 100 元
```

2)　利用 instbond 函数构造债券输入格式

调用方式

```
InstSet = instbond(InstSet, CouponRate, Settle, Maturity, Period, Basis,
EndMonthRule, IssueDate, FirstCouponDate, LastCouponDate, StartDate, Face)
```

输入参数

同 instadd 函数。

3. 债券期权(Bond option)输入格式

1)　利用 instadd 函数构造债券期权输入格式

调用方式

```
InstSet = instadd('OptBond', BondIndex, OptSpec, Strike, ExerciseDates,
AmericanOpt)
```

输入参数

```
BondIndex        %债券数量
```

```
OptSpec          %如果是看涨期权输入“Call”，如果是看跌期权输入“Put”
Strike           %执行价格
ExerciseDates    %行权日期
AmericanOpt      %行权方式，0 表示欧式期权或者百慕大期权行权方式，1 表示美式期权行权
                  方式
```

2) 利用 instoptbond 函数构造债券期权输入格式

调用方式

```
InstSet = instoptbond(InstSet, BondIndex, OptSpec, Strike, ExerciseDates,
AmericanOpt)
```

输入参数

同 instadd 函数。

【例 6-15】 有一种债券期权为欧式看涨期权，执行价为 101，行权日期为 2006 年 1 月 1 日。其输入格式如下：

```
>> BondIndex=1
>> OptSpec='call'
>> Strike=101;
>> ExerciseDates=' 01-Jan-2006';
>>AmericanOpt=0;
>> InstSet = instadd('OptBond', BondIndex, OptSpec, Strike, ExerciseDates,
  AmericanOpt)
```

下面用 instdisp 函数显示 InstSet 中的内容，代码如下：

```
  >> instdisp(InstSet)
Index  Type    UnderInd  OptSpec Strike ExerciseDates   AmericanOpt
1      OptBond  1         call    101    01-Jan-2006     0
```

4. 固定收益票据(Fixed-rate note instrument)输入格式

1) 利用 instadd 函数构造固定收益票据输入格式

调用方式

```
InstSet = instadd('Fixed', CouponRate, Settle, Maturity, Reset, Basis,
Principal)
```

输入参数

```
CouponRate      %息票率
Settle          %结算日期
Maturity        %现金流的到期日
Reset           %(Optional)票息支付频率，默认值=1
Basis           %(Optional)应计天数法则
                 0 = actual/actual (default)
```

```
                 1 = 30/360 (SIA)
                 2 = actual/360
                 3 = actual/365
                 4 = 30/360 (PSA)
                 5 = 30 360 (ISDA)
                 6 = 30/360 (European),
Principal       %(Optional)本金，默认值为 100
```

2) 利用 instfixed 函数构造固定收益票据输入格式

调用方式

```
InstSet = instfixed(InstSet, CouponRate, Settle, Maturity, Reset, Basis,
Principal)
```

输入参数

同 instadd 函数。

输出参数

同 instadd 函数。

下面是一个固定收益票据的例子：

```
>> CouponRate = 0.10;
>> Settle = '01-Jan-2000';
>> Maturity = '01-Jan-2004';
>> Basis=0
>> Reset = 1;
>> Principal=100
>> InstSet = instadd('Fixed', CouponRate, Settle, Maturity, Reset, Basis,
Principal)
InstSet =
        FinObj: 'Instruments'
    IndexTable: [1x1 struct]
          Type: {'Fixed'}
     FieldName: {{6x1 cell}}
    FieldClass: {{6x1 cell}}
     FieldData: {{6x1 cell}}
```

用 instdisp 函数显示其内容，代码如下：

```
>> instdisp(InstSet)
Index Type  CouponRate Settle       Maturity      FixedReset Basis
Principal
1     Fixed 0.1        01-Jan-2000  01-Jan-2004   1          0     100
```

5. 帽子期权(Cap instrument)输入格式

1) 利用 instadd 函数构造帽子期权输入格式

调用方式

```
InstSet = instadd('Cap', Strike, Settle, Maturity, Reset, Basis, Principal)
```

输入参数

```
Strike          %帽子期权执行价
Settle          %结算日
Maturity        %帽子期权到期日
Reset           %(Optional)票息支付频率。默认值=1
Basis           %(Optional)应计天数法则
Principal       %(Optional)本金，默认值=100
```

2) 利用 instcap 函数构造帽子期权输入格式

调用方式

```
InstSet = instcap(InstSet, Strike, Settle, Maturity, Reset, Basis, Principal)
```

输入参数

同 instadd 函数。

输出参数

同 instadd 函数。

6. 地板期权(Floor instrument)输入格式

1) 利用 instadd 函数构造地板期权输入格式

调用方式

```
InstSet = instadd('Floor', Strike, Settle, Maturity, Reset, Basis, Principal)
```

输入参数

```
Strike          %地板期权的利率下界，输入格式为小数
Settle          %结算日期
Maturity        %到期日
Reset           %(Optional)票息支付频率。默认值=1
Basis           %(Optional)应计天数法则
Principal       %(Optional)本金，默认值=100
```

2) 利用 instfloor 函数构造地板期权输入格式

调用方式

```
InstSet = instfloor(InstSet, Strike, Settle, Maturity, Reset, Basis,
Principal)
```

输入参数

同 instadd 函数。

输出参数

同 instadd 函数。

7. 利率互换(Swap instrument)输入格式

1) 利用 instadd 函数构造利率互换输入格式

调用方式

```
InstSet = instadd('Swap', LegRate, Settle, Maturity, LegReset, Basis,
Principal, LegType)
```

输入参数

```
LegRate         %需要输入一个 N×2 矩阵，N 为资产数目，第一列是息票年收益率，第二列是
                 与参考利率的差
Settle          %结算日
Maturity        %到期日
LegReset        %(Optional) 需要输入一个 N×2 矩阵，N 为资产数目，每年支付利息次数，
                 默认值是[1,1]，表示每年互换 1 次
Basis           %(Optional) 应计天数计算方式
Principal       %(Optional)本金，默认值=100
LegType         %(Optional)双方的类型，1 为固定收益，0 为浮动收益
```

2) 利用 instswap 函数构造利率互换输入格式

调用方式

```
instSet=instswap(InstSet, LegRate, Settle, Maturity, LegReset, Basis,
Principal,LegType)
```

输入参数

同 instadd 函数。

输出参数

同 instadd 函数。

下面考虑一个固定利率与浮动利率互换的例子，每年互换一次，本金为 100，互换的一方为固定利率 15%，另一方为浮动利率加上 10 个基点。

在 MATLAB 中执行如下命令：

```
>> Settle = '01-Jan-2000';
>> Maturity = '01-Jan-2003';
>> Basis = 0;
>> Principal = 100;
>> LegRate = [0.15 10]; % [票息 利率价差]
```

```
>> LegType = [1 0]; % [固定收益 浮动收益]
>> LegReset = [1 1]; % 双方每年互换一次
>> InstSet=instadd('Swap',LegRate, Settle, Maturity, LegReset, Basis,
Principal,LegType)
InstSet =
        FinObj: 'Instruments'
    IndexTable: [1x1 struct]
          Type: {'Swap'}
     FieldName: {{7x1 cell}}
    FieldClass: {{7x1 cell}}
     FieldData: {{7x1 cell}}
```

下面用 instdisp 函数显示互换的内容，代码如下：

```
>> instdisp(InstSet)
Index Type LegRate    Settle       Maturity    LegReset Basis Principal
LegType
1     Swap [0.15 10]  01-Jan-2000  01-Jan-2003  [1  1]   0    100      [1  0]
```

8. 浮动收益票据(Floating-rate note instrument)输入格式

1) 利用 instadd 函数构造浮动收益票据输入格式

调用方式

```
InstSet = instadd('Float', Spread, Settle, Maturity, Reset, Basis,
Principal )
```

输入参数

```
Spread          %与参考利率的价差
Settle          %结算日
Maturity        %现金流到期日
Reset           %(Optional)票息支付的频率，默认值=1
Basis           %(Optional) 应计天数计算方式
Principal       %(Optional)本金，默认值=100
```

2) 利用 instfloat 函数构造浮动收益票据输入格式

调用方式

```
InstSet = instfloat (InstSet , Spread, Settle, Maturity, Reset, Basis,
Principal )
```

输入参数

同 instadd 函数。

输出参数

同 instadd 函数。

如果需要改变产品格式可以利用 instsetfield 函数，例如：

```
>> product=instadd('Fixed',0.04,'01-Jan-2001','01-Jan-2003');% 构建一个固定收益证券
>> instdisp(product)  % 显示变量内容
Index Type  CouponRate Settle        Maturity       FixedReset Basis
Principal
1     Fixed 0.04       01-Jan-2001    01-Jan-2003    1          0     100
>> product=instsetfield(product,'Type','Fixed','Basis',1);%更改 Basis 的内容
>> instdisp(product)  % 显示更改后的内容
Index Type  CouponRate Settle        Maturity       FixedReset Basis
Principal
1     Fixed 0.04       01-Jan-2001    01-Jan-2003    1          1     100
```

如果需要增加一个债券品种，可以执行如下命令：

```
>> product =instadd(product,'Bond',0.04,'01-Jan-2000','01-Jan-2004') %增加一个债券品种
>> instdisp(product)  % 显示增加品种后的内容
Index Type  CouponRate Settle        Maturity       FixedReset Basis
Principal
1     Fixed 0.04       01-Jan-2001    01-Jan-2003    1          1     100
Index Type CouponRate Settle        Maturity       Period Basis EndMonthRule
2     Bond 0.04       01-Jan-2000    01-Jan-2004    2      0     1
IssueDate FirstCouponDate LastCouponDate  StartDate Face
   NaN       NaN             NaN       NaN     100
```

6.5.4　利率树时间格式

MATLAB 中对利率型衍生产品的定价主要是通过树图来完成的，每个树图的建立又是通过调用一系列函数来完成的。表 6.8 是建立衍生产品树图需要调用的函数。

表 6.8　构造利率树图需要调用的函数①

树形图名称	输入参数	调用函数形式
HW 树 (hwtree 函数)	VolSpec	hwvolspec (ValuationDate, VolDates, VolCurve, AlphaDates, AlphaCurve, InterpMethod)
	RateSpec	Intenvset
	TimeSpec	hwtimespec(ValuationDate, Maturity, Compounding)

① 表中前四类函数主要对利率类衍生产品定价，CRR 型与 EQP 型树图主要对证券类衍生产品定价。二者用于不同的衍生产品定价，鉴于它们在树形图函数形式方面基本相同，故将它们列在一起。

续表

树形图名称	输入参数	调用函数形式
BDT 树 (bdttree 函数)	VolSpec	Bdtvolspec (ValuationDate, VolDates, VolCurve, InterpMethod)
	RateSpec	Intenvset
	TimeSpec	bdttimespec(ValuationDate, Maturity, Compounding)
BK 树 (bktree 函数)	VolSpec	bkvolspec(ValuationDate, VolDates, VolCurve, AlphaDates, AlphaCurve, InterpMethod)
	RateSpec	Intenvset
	TimeSpec	TimeSpec = bktimespec(ValuationDate, Maturity, Compounding)
HJM 树 (hjmtree 函数)	VolSpec	hjmvolspec(varargin)
	RateSpec	Intenvset
	TimeSpec	hjmtimespec(ValuationDate, Maturity, Compounding)
CRR 树 (crrtree 函数)	StockSpec	stockspec(Sigma,AssetPrice,DividendType,DividendAmount, DividendDates)
	RateSpec	Intenvset
	TimeSpec	crrtimespec(ValuationDate, Maturity, NumPeriods)
EQP 树 (epqtree 函数)	StockSpec	stockspec(Sigma,AssetPrice,DividendType,DividendAmount, DividendDates)
	RateSpec	Intenvset
	TimeSpec	TimeSpec = eqptimespec(ValuationDate, Maturity, NumPeriods)

1. 建立利率波动模型利率树

1) Hull-White 模型利率波动率格式

Hull-White 模型形式如下：

$$dr = (\theta(t) - \alpha r)dt + \sigma dZ$$

其中，r 是利率，$\theta(t)$ 是为了保持模型与初始利率期限结构一致而选择的函数，独立于 r，α 是常数，σ 为短期利率的瞬间标准差，Z 为票息率。

主要是确定 $\theta(t)$ 与 $\alpha(t)$，我们给出固定日期 $\theta(t)$ 与 $\alpha(t)$ 的值，但是 $\theta(t)$ 与 $\alpha(t)$ 的值是连续的，所以其他日期采用插值方法来确定。

调用方式

```
Volspec = hwvolspec(ValuationDate, VolDates, VolCurve, AlphaDates,
AlphaCurve, InterpMethod)
```

输入参数

```
ValuationDate      %评估日，利率树第一个起始日
VolDates           %波动率的日期
VolCurve           %收益率的波动率
AlphaDates         %均值反转日期
AlphaCurve         %正的均值回归值
InterpMethod       %插值方法，默认值是线性插值法
```

输出参数

```
Volspec            %利率波动率期限结构
```

下面建立 Hull-White 模型波动率格式，主要问题是确定 Hull-White 中的参数 $\sigma(t)$ 与 $\alpha(t)$，代码如下：

```
>> ValuationDate = '01-01-2003';
>> VolDates = ['12-31-2004'; '12-31-2005'; '12-31-2006'; '12-31-2007'];
>> VolCurve=[0.01 0.01 0.01 0.01];
>> AlphaDates = ['12-31-2004'; '12-31-2005'; '12-31-2006';
'12-31-2007';'12-31-2008'];
>> AlphaCurve = [0.1; 0.1; 0.1; 0.1; 0.1] ;
>> HWVolSpec = hwvolspec(ValuationDate, VolDates, VolCurve,AlphaDates,
AlphaCurve)
HWVolSpec =
              FinObj: 'HWVolSpec'
       ValuationDate: 731582
            VolDates: [4x1 double]
            VolCurve: [4x1 double]
          AlphaCurve: [5x1 double]
          AlphaDates: [5x1 double]
     VolInterpMethod: 'linear'
```

2)　BDT 模型利率波动率格式

调用方式

```
Volspec = bdtvolspec(ValuationDate, VolDates, VolCurve, InterpMethod)
```

输入参数

同 hwvolspec 函数。

输出参数

同 hwvolspec 函数。

3) Black-Karasinski 模型利率波动率格式

调用方式

```
Volspec = bkvolspec(ValuationDate, VolDates, VolCurve, AlphaDates,
AlphaCurve, InterpMethod)
```

输入参数

同 hwvolspec 函数。

输出参数

同 hwvolspec 函数。

4) HJM 模型利率波动率格式

HJM 属于多因子模型，波动率过程包含 Constant、Stationary、Exponential、Vasicek、Proportional 5 种类型，分别代表不同的模型，其形式如下：

因子模型	公式
Constant	$\sigma(t,T) = sigma\_0$
Stationary	$\sigma(t,T) = vol(T-t) = Vol(Term)$
Exponential	$\sigma(t,T) = sigma\_0 * \exp(-\lambda(T-t))$
Vasicek, Hull-White	$\sigma(t,T) = sigma\_0 * \exp(-Decay(T-t))$
Proportional	$\sigma(t,T) = Prot(T-t) * \max(SportRate(t), MaxSpot)$

调用方式根据因子的不同分为下面几种。

① 常数波动率

```
VolSpec = hjmvolspec('Constant', Sigma_0)
```

② 静态波动率模型

```
VolSpec = hjmvolspec('Stationary', CurveVol, CurveTerm)
```

③ 指数波动率模型

```
VolSpec = hjmvolspec('Exponential', Sigma_0, Lambda)
```

④ Vasicek、Hull-White 模型

```
VolSpec = hjmvolspec('Vasicek', Sigma_0, CurveDecay, CurveTerm)
```

⑤ 比例模型

```
VolSpec = hjmvolspec('Proportional', CurveProp, CurveTerm, MaxSpot)
```

输入参数

```
Sigma_0          %初始波动率
Lambda           %衰减因子
Term             %票息发放频率，Compouding=2，Term=1 表示半年发放一次票息
```

```
CurveVol          %各个样本点上的波动率
CurveTerm         %各个样本点上的频率
CurveDecay        %各个样本点上的衰减因子
CurveProp         %各个样本点上的比率
MaxSpot           %利率上限
```

下面是一个单因子 Proportional 模型波动率过程的例子，代码如下：

```
>> CurveProp = [0.11765; 0.08825; 0.06865];
>> CurveTerm = [1; 2; 3];
>> VolSpec = hjmvolspec('Proportional', CurveProp, CurveTerm, 1e6)
VolSpec =
          FinObj: 'HJMVolSpec'
    FactorModels: {'Proportional'}
      FactorArgs: {{1x3 cell}}
      SigmaShift: 0
      NumFactors: 1
       NumBranch: 2
         PBranch: [0.5000 0.5000]
     Fact2Branch: [-1 1]
```

考虑一个 3 因素模型，波动率过程为 2 个指数型和 1 个常数型，代码如下：

```
>> VolSpec = hjmvolspec('Exponential', 0.1, 1, 'Constant', 0.2)
VolSpec =
          FinObj: 'HJMVolSpec'
    FactorModels: {'Exponential'  'Constant'}
      FactorArgs: {{1x2 cell}  {1x1 cell}}
      SigmaShift: 0
      NumFactors: 2
       NumBranch: 3
         PBranch: [0.2500 0.2500 0.5000]
     Fact2Branch: [2x3 double]
```

2. 树图时间展开输入格式

1) Hull-White 模型树图时间展开输入格式

调用方式

```
TimeSpec = hwtimespec(ValuationDate, Maturity, Compounding)
```

输入参数

```
ValuationDate     %评估日，利率树起始日
Maturity          %现金流到期日，主要是 Hull-White 模型中时间段的每个离散点日期，到期
                   日必须是依次增加的，例如 Hull-White 模型考虑时间段是 2004 年 12 月
                   31 日至 2007 年 12 月 31 日，在建立价格树时离散成 3 个时间段，每个时间
```

```
                      段的节点分别是 2004 年 12 月 31 日、2005 年 12 月 31 日、2006 年 12 月
                      31 日、2007 年 12 月 31 日
Compounding          %(Optional)票息转换为贴现率方式，默认值为 2，Compounding 可以取 1，
                      2, 3, 4, 6, 12
```

对应贴现率计算方式如下

$$Disc = \left(1 + \frac{Z}{F}\right)^{-T}$$

其中，F 是计息频率，Z 是票息率，T 是时间长度，$Disc$ 为贴现率。

如果 Compounding=365，则贴现率计算如下

$$Disc = \left(1 + \frac{Z}{F}\right)^{-T}$$

如果 Compounding=-1，表示复利，则贴现率计算如下

$$Disc = \exp(-T * Z)$$

下面建立一个 Hull-White 模型时间展开输入格式，代码如下：

```
>> ValuationDate = 'Jan-1-2004';
>> Maturity = ['12-31-2004'; '12-31-2005'; '12-31-2006'; '12-31-2007'];
>> Compounding = 1;
>> TimeSpec = hwtimespec(ValuationDate, Maturity, Compounding)
TimeSpec =
          FinObj: 'HWTimeSpec'
   ValuationDate: 731947
        Maturity: [4x1 double]
     Compounding: 1
           Basis: 0
```

2) BDT 模型树图时间展开输入格式

调用方式

```
TimeSpec = bdttimespec(ValuationDate, Maturity, Compounding)
```

输入参数

同 hwtimespec 函数。

【例 6-16】 例如期权生效日为 2000 年 1 月 1 日，到期日为 2001 年 1 月 1 日、2002 年 1 月 1 日、2003 年 1 月 1 日、2004 年 1 月 1 日、2005 年 1 月 1 日，下面给出时间输入格式。

在 MATLAB 中执行如下命令：

```
>> Compounding = 1;
>> ValuationDate = '01-01-2000';
>> Maturity = ['01-01-2001'; '01-01-2002'; '01-01-2003';
          '01-01-2004'; '01-01-2005'];
```

```
>> TimeSpec = bdttimespec(ValuationDate, Maturity, Compounding)
TimeSpec =
          FinObj: 'BDTTimeSpec'
    ValuationDate: 730486
         Maturity: [5x1 double]
      Compounding: 1
            Basis: 0
     EndMonthRule: 1
```

3) BK 模型树图时间展开输入格式

调用方式

```
TimeSpec = bktimespec(ValuationDate, Maturity, Compounding)
```

输入参数

同 hwtimespec 函数。

4) HJM 模型树图时间展开输入格式

调用方式

```
TimeSpec = hjmtimespec(ValuationDate, Maturity, Compounding)
```

输入参数

同 hwtimespec 函数。

5) CRR 模型树图时间展开输入格式

调用方式

```
TimeSpec = crrtimespec(ValuationDate, Maturity, Compounding)
```

输入参数

同 hwtimespec 函数。

6) EQP 模型树图时间展开输入格式

调用方式

```
TimeSpec = eqptimespec(ValuationDate, Maturity, Compounding)
```

输入参数

同 hwtimespec 函数。

6.5.5 说明利率期限结构函数

MATLAB 金融工具箱为了更加方便地处理多种时间长短不一的债券利率，可以事先把债券的特征保存在一个结构数组中。在研究利率期限结构时，通常要考虑的因素是每年分红的频率、债券发行日期、债券到期日、债券年利率、研究时间段等。我们可以把这些特

征用结构变量来表示，不妨记为 RateSpec，具体用法见第 4 章。

6.5.6 建立利率树

1. HW 模型利率树

调用方式

```
HWTree = hwtree(VolSpec, RateSpec, TimeSpec)
```

输入参数

见前。

输出参数

见前。

2. BDT 模型利率树

调用方式

```
BDTTree = bdttree(VolSpec, RateSpec, TimeSpec)
```

输入参数

见前。

输出参数

见前。

3. BK 模型利率树

调用方式

```
BKTree = bktree(VolSpec, RateSpec, TimeSpec)
```

输入参数

见前。

输出参数

见前。

4. HJM 模型利率树

调用方式

```
HJMTree = hjmtree(VolSpec, RateSpec, TimeSpec)
```

输入参数

见前。

输出参数

见前。

6.5.7　利率产品定价

MATLAB 对利率类衍生产品定价的函数建立在利率树基础上，其核心思想是用利率树对金融产品进行定价。

MATLAB 工具箱中定价函数的调用方式如表 6.9 所示。

表 6.9　MATLAB 中利率类衍生产品定价函数[①]

模型名称	输入参数
HW 模型	hwprice(HWTree,InstSet,Options)
BK 模型	bkprice(BKTree, InstSet,Options)
BDT 模型	bdtprice(BDTTree, InstSet,Options)
HJM 模型	hjmprice(HJMTree, InstSet,Options)
EQP 模型	Eqpprice(EQPTree,InstSet,Options)
CRR 模型	Crrprice(CRRTree, InstSet,Options)

1．利用 HW 模型对衍生产品定价

调用方式

```
[Price, PriceTree] = hwprice(HWTree, InstSet, Options)
```

输入参数

HWTree	%见前
InstSet	%见前
Options	%(Optional)模型控制参数。由 derivset 函数构建，其调用方式为 Options = derivset(Options, '参数1', 参数1的值, …'参数4',参数4的值)，参数的具体内容如下： “Diagnostics”选择“on”表示给出诊断信息，“off”(默认值)表示不给出诊断信息 “Warnings”选择“on”(默认值)表示给出警告信息，“off”表示不给出警告信息 “ConstRate”选择“on”(默认值)表示树的各个时间段之间利率是常数，“off”表示不是常数 “BarrierMethod”选择“unenhanced”(默认值)表示在计算障碍期权时不进行误差修正，“interp”表示障碍期权各个节点用插值法修正

① MATLAB 中 EQP 模型与 CRR 模型对证券类的衍生产品适用，但不适用于利率类衍生产品。

【例 6-17】调用 MATLAB 自带的数据进行定价，deriv.mat 中包含了 Hull-White 模型的树形数据，以及 HWInstSet 变量中包含了需要定价的产品特征。下面对 HWInstSet 中的 3 种产品进行定价，执行如下命令：

```
>> load deriv.mat
>> % 下面从 deriv.mat 文件中调入 HWInstSet 变量，包含了 8 个金融工具。下面我们选取其中
特征为“Bond” 与“Cap”金融工具进行定价
>> HWSubSet = instselect(HWInstSet,'Type', {'Bond', 'Cap'}); % 重新选择利率
工具
>> instdisp(HWSubSet)  %观察选中的金融工具特征
Index  Type  CouponRate  Settle        Maturity      Period Basis   ...
1      Bond  0.04        01-Jan-2004   01-Jan-2007   1      0       ...
2      Bond  0.04        01-Jan-2004   01-Jan-2008   1      0       ...
Index Type Strike Settle        Maturity      CapReset   Basis  Principal ...
3     Cap  0.06   01-Jan-2004   01-Jan-2008   1          0      100    ...
>> [Price, PriceTree] = hwprice(HWTree, HWSubSet)
   Price =
       100.9188
       99.3296
       0.5837
PriceTree =
       FinObj: 'HWPriceTree'
       PTree: {[3x1 double]  [3x3 double]  [3x5 double]  [3x5 double]  [3x5
double]}
       AITree: {[3x1 double]  [3x3 double]  [3x5 double]  [3x5 double]  [3x5
double]}
       tObs: [0 1 2 3 4]
       Connect: {[2]  [2 3 4]  [2 2 3 4 4]}
       Probs: {[3x1 double]  [3x3 double]  [3x5 double]}
```

Price 变量给出了选中的 3 种金融工具价格分别为 100.9188、99.3296、0.5837。PriceTree 为现金流的树，可以用 treeviewer 函数观察 PriceTree 每个节点上的现金流。

2. 利用 BK 模型、BDT 模型、HJM 模型进行定价

方法同上。

思 考 题

1. 绘出宝钢权证的隐含波动率，并与真实波动率进行比较，试从行为金融学角度解释二者之间的差别。

2. 分析五粮液股票、看涨与看跌权证的价格走势，研究三者之间的套利关系。

3. 2007 年 2 月初都市股份与沪东重机股价连续翻番，市场非常关注，都市股份、海通证券公司借壳上市，沪东重机涉及非公开发行，而相比之下，权证波澜不惊，说明如何看待二者之间的收益差别，要求用 MATLAB 对收益进行分析比较。

4. 试分析权证和相关股票构成的组合市值与大盘之间的关系，研究如何实现为不同风险偏好的客户设计出不同的投资品种。

5. 例如 2007 年 2 月 9 日某个投资者看好一只股票，但是担心大盘继续下跌，试为该投资者设计出几种不同投资策略，分析其风险收益。为了应对市场竞争，进一步地分析通过承担一部分客户风险来留住客户的可行性。

6. 考虑为什么权证的价格与理论模型存在差别，如何看待这种差别。

7. 观察宝钢权证的价格时间序列，并分析其合理的投资区间，联系宝钢股份(600019)股价，分析上市公司管理层的经营思路影响权证价格途径。

8. 2007 年 3 月 2 日上海、深圳二市部分权证信息如表 6.10 所示。

表 6.10 沪、深二市部分权证信息

名 称	代 码	种 类	到 期 日	正 股 价	执 行 价	权证价格
海尔	580991	Put	2007-5-16	9.88	4.29	0.6
雅戈尔	580992	put	2007-5-21	13.23	4.09	0.64
雅戈尔	580006	Call	2007-5-17	13.23	3.66	10.4
新钢钒	038001	put	2007-5-3	6.19	3.16	1.32
新钢钒	031002	Call	2007-11-28	6.19	3.95	3.73
包钢	580995	put	2007-3-30	5.37	2.37	0.577
招行	580997	put	2007-9-1	15.64	5.48	0.428
万华	580993	put	2007-4-26	29.27	9.22	0.18
万华	580005	buy	2007-4-20	29.27	6.38	28.86
中集	038006	put	2007-11-19	23.19	8.87	0.998
五粮液	038004	put	2008-3-27	24.39	7.89	1.32
五粮液	030002	Call	2008-3-27	24.39	6.87	16.22
钾肥	038008	put	2007-6-25	26.79	15.1	1.427
茅台	580990	put	2007-5-29	97.57	30.3	0.857
中化	580011	call	2007-12-11	10.34	6.58	6.9
伊利	580009	call	2007-11-8	25.53	8	18.15

试给出每种权证的内在价值与时间价值，分析发行权证的风险与收益，如果同时卖出雅戈尔、新钢钒、万华、五粮液看涨与看跌期权，用图画出收益来。

第 7 章　有限差分法定价

有限差分方法是计算偏微分方程的有效工具。本章介绍了 3 种常见的差分方法，并且给出了相关程序，要求了解有限差分方法计算的基本原理，熟悉显示法计算欧式看涨期权价格。

7.1　有限差分法基本原理

偏微分方程在金融工程中占有重要位置，著名的 Black-Scholes 方程就是以二阶偏微分方程形式给出的。偏微分方程为求解复杂的金融衍生工具价格提供了有力手段，但是偏微分方程通常没有解析解，因此数值计算方法求解衍生工具价格就成为金融工程的一项基本功。求解金融衍生工具价格与求解通常偏微分方程的区别主要在于一般偏微分方程是给定初值求解终值，而衍生品定价问题是给定终值求初值，属于倒向随机偏微分方程求解。

有限差分方法的核心思想是对导数进行离散化，把偏微分方程转化为差分方程，然后利用迭代法求解。

有限差分方法根据对偏导数离散方法的不同，分为显式差分法[①]、隐式差分法[②]、内含差分法，下面分别进行介绍。

假设 $f_{i,j}$ 表示在 i 时刻股票价格为第 j 价位的期权价格，我们对 f 一阶导数进行如下差分：

$$\frac{\partial f}{\partial S}=\frac{f_{i,j+1}-f_{i,j}}{\delta S} \tag{7.1}$$

$$\frac{\partial f}{\partial t}=\frac{f_{i+1,j}-f_{i,j}}{\delta t} \tag{7.2}$$

上面的差分方法称为显式差分法。我们也可以做如下差分：

$$\frac{\partial f}{\partial S}=\frac{f_{i,j}-f_{i,j-1}}{\delta S} \tag{7.3}$$

① 又称向前差分法(Forward Different Approximation)。

② 又称向后差分法(Backward Different Approximation)。

$$\frac{\partial f}{\partial t}=\frac{f_{i,j}-f_{i-1,j}}{\delta t} \tag{7.4}$$

上面的差分方法称为隐式差分法。我们也可以对一阶导数做如下差分：

$$\frac{\partial f}{\partial S}=\frac{f_{i,j+1}-f_{i,j-1}}{2\delta S} \tag{7.5}$$

$$\frac{\partial f}{\partial t}=\frac{f_{i+1,j}-f_{i-1,j}}{2\delta t} \tag{7.6}$$

上面的差分方法称为内含差分法(Implicit Finite Method)。

对二阶微分方程，我们用如下方法进行差分：

$$\frac{\partial^2 f}{\partial S^2}=\left(\frac{f_{i,j+1}-f_{i,j}}{\delta S}-\frac{f_{i,j}-f_{i,j-1}}{\delta S}\right)\Big/\delta S \tag{7.7}$$

整理得

$$\frac{\partial^2 f}{\partial S^2}=\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\delta S^2}$$

以上就是偏微分方程的常见离散方法。

7.2 有限差分求解方法

7.2.1 显示法求解欧式看跌期权

下面利用显示法求解欧式看跌期权。对一阶偏导数、二阶导数离散方式如下

$$\frac{\partial f}{\partial t}=\frac{f_{i,j}-f_{i-1,j}}{\delta t} \tag{7.8}$$

$$\frac{\partial f}{\partial S}=\frac{f_{i,j+1}-f_{i,j-1}}{2\delta S} \tag{7.9}$$

$$\frac{\partial^2 f}{\partial S^2}=\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\delta S^2} \tag{7.10}$$

将式(7.8)、式(7.9)、式(7.10)代入 B-S 公式有

$$\frac{f_{i,j}-f_{i-1,j}}{\delta t}+rj\delta S\frac{f_{i,j+1}-f_{i,j-1}}{2\delta S}+\frac{1}{2}\sigma^2 j^2\delta S^2\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\delta S^2}=rf_{i,j} \tag{7.11}$$

经过整理可得

$$f_{i-1,j}=a_j^* f_{i,j-1}+b_j^* f_{i,j}+c_j^* f_{i,j+1} \qquad i=0,1,2,\cdots,N-1;\quad j=1,2,3,\cdots,M-1 \tag{7.12}$$

其中

$$a_j^* = \frac{1}{2}\delta t(\sigma^2 j^2 - rj)$$

$$b_j^* = 1 - \delta t(\sigma^2 j^2 + rj)$$

$$c_j^* = \frac{1}{2}\delta t(\sigma^2 j^2 + rj)$$

将式(7.12)写成矩阵形式

$$\begin{pmatrix} f_{N-1,M-1} \\ f_{N-1,M-2} \\ f_{N-1,M-3} \\ \vdots \\ f_{N-1,2} \\ f_{N-1,1} \end{pmatrix} = \begin{pmatrix} c_{M-1}^* & b_{M-1}^* & a_{M-1}^* & 0 & \cdots & 0 \\ 0 & c_{M-2}^* & b_{M-2}^* & a_{M-2}^* & \cdots & 0 \\ 0 & 0 & c_{M-3}^* & b_{M-3}^* & a_{M-3}^* & 0 \\ 0 & 0 & 0 & c_{M-4}^* & b_{M-4}^* & a_{M-4}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & c_1^* & b_1^* & a_1^* \end{pmatrix} \begin{pmatrix} f_{N,M} \\ f_{N,M-1} \\ f_{N,M-2} \\ \vdots \\ f_{N,1} \\ f_{N,0} \end{pmatrix}$$

如果记矩阵

$$\boldsymbol{L} = \begin{pmatrix} c_{M-1}^* & b_{M-1}^* & a_{M-1}^* & 0 & \cdots & 0 \\ 0 & c_{M-2}^* & b_{M-2}^* & a_{M-2}^* & \cdots & 0 \\ 0 & 0 & c_{M-3}^* & b_{M-3}^* & a_{M-3}^* & 0 \\ 0 & 0 & 0 & c_{M-4}^* & b_{M-4}^* & a_{M-4}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & c_1^* & b_1^* & a_1^* \end{pmatrix}_{(M-1)\times(M-1)}$$

则上式可以写为

$$\begin{pmatrix} f_{N-1,M} \\ f_{N-1,M-1} \\ f_{N-1,M-2} \\ \vdots \\ f_{N-1,1} \\ f_{N-1,0} \end{pmatrix} = L_{(M-1)\times(M-1)} \begin{pmatrix} f_{N,M} \\ f_{N,M-1} \\ f_{N,M-2} \\ \vdots \\ f_{N,1} \\ f_{N,0} \end{pmatrix}$$

也即

$$f^{N-1} = Lf^N$$

对于欧式看跌期权，其终值条件如下：

$$f(S,T) = \max\{K - S, 0\} \qquad \forall S > 0$$

下面我们考虑欧式看跌期权的边界条件，当股票价格 S_t 非常大时，看跌期权到期日价值为 0，$f(t, S_{\max}) = 0$；当股票价格为 0 时，$S_t = 0$，那么到期日支付价值为 K，贴现到 t 期

有 $f(t,0)=Ke^{-r(T-t)}$，边界条件可以写成下面形式

$$f_{i,M}=0 \qquad i=1,2,\cdots,N$$

$$f_{i,0}=Ke^{-r(N-i)\Delta t} \qquad i=0,1,2,\cdots,N$$

$$f_{N,j}=\max(K-j\Delta S,0) \qquad j=0,1,2,\cdots,M$$

【例 7-1】 已知股票价格为 50 元，欧式看跌期权执行价为 50 元，到期日为 5 个月，股票年波动率的标准差为 0.3，无风险利率为 5%，试用有限元方法求解期权价格。

下面是用 MATLAB 编写的有限元求解欧式期权的程序。

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%              显示法求解欧式看跌期权          %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s0=50;          %股价
k=50;           %执行价
r=0.1;          %无风险利率
T=5/12;         %存续期
sigma=0.3;      %股票波动率
Smax=100;       %确定股票价格最大价格
ds=2;           %确定股价离散步长
dt=5/1200;          %确定时间离散步长
M=round(Smax/ds);    %计算股价离散步数，对 Smax/ds 取整运算
ds=Smax/M;           %计算股价离散实际步长
N=round(T/dt);       %计算时间离散步数
dt=T/N;              %计算时间离散实际步长
matval=zeros(M+1,N+1);
ets=linspace(0,Smax,M+1);   % 将区间 [0Smax] 分成 M+1 段
Veti=0:N;
vetj=0:M;
% 建立偏微分方程边界条件
matval(:,N+1)=max(k-vetS,0);
matval(1,:)=k*exp(-r*dt*(N-veti));
matval(M+1,:)=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 确定叠代矩阵系数
a=0.5*dt*(sigma^2*vetj-r).*vetj;
b=1-dt*(sigma^2*vetj.^2+r);
c=0.5*dt*(sigma^2*vetj+r).*vetj;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L=zeros(M-1,M+1)
for i=2:M
%建立递推关系
L(i-1,i-1)=a(i); L(i-1,i)=b(i); L(i-1,i+1)=c(i);
end
```

```
for i=N:-1:1
matval(2:M,i)=L*matval(:,i+1);
end
%寻找期权价格进行插值①
jdown=floor(s0/ds);
jup =ceil(s0/ds);
if jdown= =jup
   price=matval(jdown+1,1)+(s0-jdown*ds)*(matval(jup+1,1)-
      matval(jup+1,1))/ds
end
```

运行结果如下：

```
price =
   2.8288
```

7.2.2 显示法求解美式看跌期权

这里我们采用与 7.2.1 小节不同的离散方式，显式差分离散方法如下：

$$\frac{\partial f}{\partial S}=\frac{f_{i+1,j+1}-f_{i+1,j-1}}{2\Delta S} \tag{7.13}$$

$$\frac{\partial^2 f}{\partial S^2}=\frac{f_{i+1,j+1}+f_{i+1,j-1}-2f_{i+1,j}}{\Delta S^2} \tag{7.14}$$

这样差分方程为

$$\frac{f_{i+1,j}-f_{i,j}}{\Delta t}+rj\Delta S\frac{f_{i+1,j+1}-f_{i+1,j-1}}{2\Delta S}+\frac{1}{2}\sigma^2 j^2\Delta S^2\frac{f_{i+1,j+1}+f_{i+1,j-1}}{\Delta S^2}=rf_{i,j}$$

整理得

$$f_{i,j}=a_j f_{i+1,j-1}+b_j f_{i+1,j}+c_j f_{i+1,j+1} \tag{7.15}$$

其中

$$a_j=\frac{1}{1+r\Delta t}\left(-\frac{1}{2}rj\Delta t+\frac{1}{2}\sigma^2 j^2\Delta t\right)$$

$$b_j=\frac{1}{1+r\Delta t}(1-\sigma^2 j^2\Delta t)$$

$$c_j=\frac{1}{1+r\Delta t}\left(\frac{1}{2}rj\Delta t+\frac{1}{2}\sigma^2 j^2\Delta t\right)$$

---

① 通过递推关系我们最后得到在 0 时刻的股价 S_0 与期权价格 f 之间的函数关系是 $f=f(S_0)$，该函数关系是离散的，所以我们利用线性插值法求解期权的价格 f。

考虑到边界条件

$$f_{i,0}=k$$
$$f_{i,M}=0$$
$$i=0,2,3,\cdots,N$$

如果记

$$\boldsymbol{L}=\begin{pmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \cdots & 0 \\ & & \ddots & & \\ \vdots & \vdots & a_{M-2} & b_{M-2} & c_{M-2} \\ 0 & 0 & \cdots & a_{M-1} & b_{M-1} \end{pmatrix}, \quad g=\begin{pmatrix} a_1 k \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

对于第 i 个时刻的现金流 $F_{i,j}$， $F_{i,j}=\max(k-j\Delta S,f_{i,j}) \quad i,j=1,2,3,\cdots,M-1$

记

$$F=(F_1,F_2,\cdots,F_{M-1})^{\tau}$$

则公式(7.15)有

$$F^i=L\,F^{i-1}+g$$

【例 7-2】已知股票价格为 50 元，美式看跌期权执行价为 50 元，到期日为 5 个月，股票年波动率的标准差为 0.4，无风险利率为 10%，试用显式差分法求解期权价格。

下面是用 MATLAB 编写的显式差分法求解美式看跌期权程序。

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  显式差分法求解美式看跌期权
%  时间：2006 年 4 月 12 日
%  最新修改时间：2006 年 7 月 6 日
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数说明：
%  s0      0 时刻股价
%  k       期权执行价
%  r       无风险利率
%  T       到期日
%  sigma   股票波动标准差
%  Smax    股票最大值
%  ds      股票离散步长
%  dt      时间离散步长
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                股票价格、时间初始化                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s0=50;k=50;r=0.1;sigma=0.4;T=5/12;
dt=T/10;ds=5;
Smax=100;
```

```
M=round(Smax/ds);    % 对 Smax/ds 取整运算
N=round(T/dt);
ds=Smax/M;           % 重新确定股票价格步长
dt=T/N;              % 重新确定时间步长
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
veti=1:N;
vetj=1:M;
a=1/(1+r*dt)*(-1/2*r*vetj*dt+1/2*sigma^2*vetj.^2*dt);
b=1/(1+r*dt)*(1-sigma^2*vetj.^2*dt);
c=1/(1+r*dt)*(1/2*r*vetj*dt+1/2*sigma^2*vetj.^2*dt);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L=zeros(M-1,M-1);
L(1,1)=b(1);L(1,2)=c(1);
L(M-1,M-2)=a(M-1);L(M-1,M-1)=b(M-1);
for j=2:M-2
   L(j,j-1)=a(j);L(j,j)=b(j);L(j,j+1)=c(j);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f1=zeros(M-1,N+1);
f1(:,N+1)=max(k-vetj(1:M-1)*ds,0);
f0=zeros(M-1,1);
f0(1,1)=a(1)*k ;
for i=N:-1:1
   f1(:,i)=L*f1(:,i+1)+f0;
   for j=1:M-1
      if f1(j,i)<=k-vetj(j)*ds;
        f1(j,i)=k-vetj(j)*ds ;
      end
   end
end
f2(1,1:N+1)=50;
f2(2:M,1:N+1)=f1;
f2(M+1,1:N+1)=0 ;
```

f2(i,j)表示在 i 时刻，j 价位的期权价格，结果如表 7.1 所示。

表 7.1 显示法计算美式看跌期权过程

股价	时间										
	10	9	8	7	6	5	4	3	2	1	0
100	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
95	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00
90	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00

续表

股价	时间										
	10	9	8	7	6	5	4	3	2	1	0
85	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00
80	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
75	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00
70	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
65	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00
60	10.28	10.20	10.13	10.06	10.01	10.00	10.00	10.00	10.00	10.00	10.00
55	6.76	6.61	6.47	6.31	6.15	5.96	5.75	5.50	5.24	5.00	5.00
50	4.26	4.08	3.89	3.68	3.44	3.18	2.87	2.53	2.07	1.56	0.00
45	2.59	2.39	2.21	1.99	1.77	1.50	1.24	0.90	0.59	0.00	0.00
40	1.48	1.37	1.16	1.02	0.81	0.65	0.42	0.27	0.00	0.00	0.00
35	0.91	0.68	0.63	0.44	0.37	0.21	0.14	0.00	0.00	0.00	0.00
30	0.32	0.46	0.23	0.25	0.10	0.09	0.00	0.00	0.00	0.00	0.00
25	0.45	0.06	0.20	0.04	0.06	0.00	0.00	0.00	0.00	0.00	0.00
20	−0.13	0.20	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.28	−0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	−0.11	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

股价 50 对应的期权价格为 4.26。

7.2.3　隐式法求解欧式看跌期权

隐式法一阶导数、二阶导数的离散方法如下：

$$\frac{\partial f}{\partial t}=\frac{f_{i+1,j}-f_{i,j}}{\delta t} \tag{7.16}$$

$$\frac{\partial f}{\partial S}=\frac{f_{i,j+1}-f_{i,j-1}}{2\delta S} \tag{7.17}$$

$$\frac{\partial^2 f}{\partial S^2}=\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\delta S^2} \tag{7.18}$$

把上面 3 个等式代入 B-S 公式有

$$\frac{f_{i+1,j}-f_{i,j}}{\delta t}+rj\delta S\frac{f_{i,j+1}-f_{i,j-1}}{2\delta S}+\frac{1}{2}\sigma^2 j^2\delta S^2\frac{f_{i,j+1}+f_{i,j-1}-2f_{i,j}}{\delta S^2}=rf_{i,j} \tag{7.19}$$

整理得

$$a_j f_{i,j-1}+b_j f_{i,j}+c_j f_{i,j+1}=f_{i+1,j} \tag{7.20}$$

其中

$$a_j=\frac{1}{2}rj\delta t-\frac{1}{2}\sigma^2 j^2\delta t \tag{7.21}$$

$$b_j=1+\sigma^2 j^2\delta t+r\delta t \tag{7.22}$$

$$c_j=-\frac{1}{2}rj\delta t-\frac{1}{2}\sigma^2 j^2\delta t \tag{7.23}$$

利用矩阵形式改写为

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & a_{M-2} & b_{M-2} & c_{M-2} \\ 0 & 0 & 0 & 0 & a_{M-1} & b_{M-1} \end{bmatrix}\begin{pmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \\ \vdots \\ f_{i,M-2} \\ f_{i,M-1} \end{pmatrix}=\begin{pmatrix} f_{i+1,1} \\ f_{i+1,2} \\ f_{i+1,3} \\ \vdots \\ f_{i+1,M-2} \\ f_{i+1,M-1} \end{pmatrix}-\begin{pmatrix} a_1 f_{i+1,0} \\ 0 \\ 0 \\ \vdots \\ 0 \\ c_{M-1} f_{i+1,M} \end{pmatrix}$$

记

$$\boldsymbol{L}=\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & a_{M-2} & b_{M-2} & c_{M-2} \\ 0 & 0 & 0 & 0 & a_{M-1} & b_{M-1} \end{bmatrix},\quad f^{(i)}=\begin{pmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \\ \vdots \\ f_{i,M-2} \\ f_{i,M-1} \end{pmatrix},\quad g=\begin{pmatrix} a_1 f_{i+1,0} \\ 0 \\ 0 \\ \vdots \\ 0 \\ c_{M-1} f_{i+1,M} \end{pmatrix}$$

则有

$$Lf^{(i)}=f^{(i+1)}-g$$

【例 7-3】已知股票价格为 50 元，欧式看跌期权执行价为 50 元，到期日为 5 个月，股票年波动率的标准差为 0.4，无风险利率为 10%，试用有限元方法求解期权价格。

编写程序如下：

```
s0=50;k=50;T=5/12;sigma=0.4;r=0.1;
Smax=100;ds=0.5;dt=5/2400;  %确定时间与价格离散步长
M=round(Smax/ds);
ds=Smax/M;
N=round(T/dt);
```

```
dt=T/N;
matval=zeros(M+1,N+1);
vetS=linspace(0,Smax,M+1);
veti=0:N;
vetj=0:M;
%*********************************%
%              建立边界条件             %
%*********************************%
matval(:,N+1)=max(k-vetS,0);
matval(1,:)  =k*exp(-r*dt*(N-veti));
matval(M+1,:)=0;
%*********************************%
%             建立三对角矩阵            %
%*********************************%
a=0.5*(r*dt*vetj-sigma^2*dt*vetj.^2);
b=1+sigma^2*dt*vetj.^2+r*dt;
c=-0.5*(r*dt*vetj+sigma^2*dt*vetj.^2);
coeff=diag(a(3:M),-1)+diag(b(2:M))+diag(c(2:M-1),1);
[L,U]=lu(coeff);  % 对矩阵 coeff 进行 LU 分解
%********************************%
%              求解线性方程           %
%********************************%
aux=zeros(M-1,1);
for i=N:-1:1
    aux(1)=-a(2)*matval(1,i);
    matval(2:M,i)=U\(L\(matval(2:M,i+1)+aux));
end
jdown=floor(s0/ds);
jup=ceil(s0/ds)
if jdown==jup
    price=matval(jdown+1,1)
else
    price=matval(jdown+1,1)+(s0-jdown*ds)*(matval(jup+1,1)-
           matval(jup+1,1))/ds
end
```

运行结果如下：

```
>>Euputimpl
price =
    4.0718
```

7.2.4 隐式法求解美式看跌期权

同 7.2.2 小节中的方法一样，根据式式(7.20)、式(7.21)、式(7.22)和式(7.23)，下面我们考虑美式看跌期权的边界条件。

T时刻看跌期权到期现金流为$\max(K-S_T,0)$，其中S_T为T时刻的股票价格，因此

$$f_{N,j}=\max(K-j\Delta S,0) \qquad j=0,1,2,3,\cdots,M \tag{7.24}$$

当股票价格为 0 时，看跌期权价格为K，因此

$$f_{i,0}=K \qquad i=0,1,2,3,\cdots,N \tag{7.25}$$

当股票价格趋于无穷大时，看跌期权价格趋于 0，因此我们有近似值

$$f_{i,M}=0 \qquad i=0,1,2,3,\cdots,N$$

下面考虑美式期权提前执行条件，当计算$f_{i,j}$时，有

$$f_{i,j}=\max(f_{i,j},k-j\Delta S) \tag{7.26}$$

注意式(7.26)右边$f_{i,j}$是递推公式结果，左边才是i时刻j价位的期权价格。

【例 7-4】中的美式看跌期权的隐式法求解程序如下。

MATLAB 程序如下：

```
function  amoption(s0,k,rf,sigma,T,dt,ds,Smax)
%%%%%%%%%%%%%%%%%%%%%%%%%%
%  隐式法求解美式看跌期权
%  时间：2006 年 4 月 12 日
%  最新修改时间：2006 年 7 月 6 日
%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数说明：
%  s0      0 时刻股价
%  k       期权的执行价
%  r       无风险利率
%  T       到期日
%  sigma   股票波动的标准差
%  Smax    股票最大值
%  ds      股票价格离散的步长
%  dt      时间离散的步长
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                  初始化                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=round(Smax/ds);
N=round(T/dt);
ds=Smax/M;                          %  重新确定股票价格步长
dt=T/N;                             %  确定时间的步长
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
for j=1:M
  a(j)=0.5*rf*j*dt-0.5*sigma^2*j^2*dt
  b(j)=1+sigma^2*j^2*dt+rf*dt
  c(j)=-0.5*rf*j*dt-0.5*sigma^2*j^2*dt
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%
 L=zeros(M-1,M-1);
 L(1,1)=b(1);L(1,2)=c(1);                        % 边界条件
 L(M-1,M-2)=a(M-1),L(M-1,M-1)=b(M-1);           % 边界条件
  for j=2:M-2
  L(j,j-1)=a(j);L(j,j)=b(j);L(j,j+1)=c(j)
  end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  for j=1:M-1
     f(j,N+1)=max(k-j*ds,0)
  end
for i=N:-1:1
  F(1)=f(1,i+1)-a(1)*k;F(2:M-1)=f(2:M-1,i+1)      % 终值条件
  f(1:M-1,i)=L^(-1)*F'
     for j=1:M-1                              %判断是否行权
        if f(j,i)<k-j*ds
           f(j,i)=k-j*ds
        end
      end
```

将例 7-4 的参数代入即可，计算结果如表 7.2 所示。

表 7.2　隐式法求解美式看跌期权过程

股价	时间										
	10	9	8	7	6	5	4	3	2	1	0
100	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
95	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00
90	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00
85	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00
80	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
75	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00
70	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
65	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00
60	10.15	10.10	10.05	10.01	10.00	10.00	10.00	10.00	10.00	10.00	10.00
55	6.58	6.44	6.29	6.13	5.96	5.77	5.57	5.36	5.17	5.02	5.00

续表

股价	时间										
	10	9	8	7	6	5	4	3	2	1	0
50	4.07	3.88	3.67	3.45	3.19	2.91	2.57	2.17	1.66	0.99	0.00
45	2.43	2.24	2.05	1.83	1.61	1.36	1.09	0.81	0.51	0.22	0.00
40	1.42	1.27	1.11	0.95	0.78	0.62	0.45	0.30	0.16	0.05	0.00
35	0.82	0.71	0.60	0.49	0.38	0.28	0.19	0.11	0.05	0.02	0.00
30	0.47	0.39	0.32	0.25	0.18	0.13	0.08	0.04	0.02	0.00	0.00
25	0.27	0.22	0.17	0.13	0.09	0.06	0.03	0.02	0.01	0.00	0.00
20	0.16	0.12	0.09	0.07	0.04	0.03	0.02	0.01	0.00	0.00	0.00
15	0.09	0.07	0.05	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00
10	0.05	0.04	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00
5	0.02	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

从表 7.2 可以看出，股票价格 50 对应的期权价格为 4.07。

7.2.5 Crank-Nicolson 法求解欧式障碍期权

Crank-Nicholson 方法是内含有限差分与外推有限差分的平均值，是比较精确的一种数值解法。隐式有限差分方程为

$$f_{i,j} = a_j f_{i-1,j-1} + b_j f_{i-1,j} + c_j f_{i-1,j+1} \tag{7.27}$$

外推有限差分方程为

$$f_{i-1,j} = a_j^* f_{i,j-1} + b_j^* f_{i,j} + c_j^* f_{i,j+1} \tag{7.28}$$

Crank－Nicholson 方法将式(7.27)、式(7.28)这两个方程相加，得

$$f_{i,j} + f_{i-1,j} = a_j f_{i-1,j-1} + b_j f_{i-1,j} + c_j f_{i-1,j+1} + a_j^* f_{i,j-1} + b_j^* f_{i,j} + c_j^* f_{i,j+1}$$

整理得

$$-\alpha_j f_{i-1,j-1} + (1-\beta_j) f_{i-1,j} - \gamma_{i-1,j+1} = \alpha_j f_{i,j-1} + (1+\beta_j) f_{i,j} + \gamma_j f_{i,j+1}$$

其中

$$\alpha_j = \frac{\Delta t}{4}(\sigma^2 j^2 - rj)$$

$$\beta_j = -\frac{\Delta t}{2}(\sigma^2 j^2 + r)$$

$$\gamma_j = \frac{\Delta t}{4}(\sigma^2 j^2 + rj)$$

方程的矩阵形式为

$$M_1 f_{i-1} = M_2 f_i$$

其中

$$M_1 = \begin{pmatrix} 1-\beta_1 & -\gamma_1 & 0 & 0 & 0 & 0 \\ -\alpha_2 & 1-\beta_2 & -\gamma_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & -\alpha_{M-2} & 1-\beta_{M-2} & -\gamma_{M-2} \\ 0 & 0 & 0 & 0 & -\alpha_{M-1} & 1-\beta_{M-1} \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 1+\beta_1 & \gamma_1 & 0 & 0 & 0 & 0 \\ \alpha_2 & 1+\beta_2 & \gamma_2 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \alpha_{M-2} & 1+\beta_{M-2} & \gamma_{M-2} \\ 0 & 0 & 0 & 0 & \alpha_{M-1} & 1+\beta_{M-1} \end{pmatrix}$$

$$f^i = (f_{i,1}, f_{i,2}, \cdots, f_{i,M-1})^T$$

我们考虑欧式下跌出局看跌期权(down-and-out put option)，障碍值为 S_b， $S \geqslant S_b$，障碍期权的现金流如下：

$$f(t, S\max) = 0\text{，}\quad f(t, S_b) = 0$$

【例 7-5】已知股票价格为 50 元，欧式看跌期权执行价为 50 元，到期日为 5 个月，股票年波动率的标准差为 0.4，无风险利率为 10%，期权障碍值是 40，试用有限元方法求解欧式障碍期权价格。

在 MATLAB 中编写程序如下：

```
clc;clear;
s0=50;k=50;r=0.1;T=5/12;sigma=0.4;sb=40;  % sb 是障碍值
Smax=100;ds=0.5;dt=5/2400;
M=round((Smax-sb)/ds);
ds=(Smax-sb)/M;
N=round(T/dt);
dt=T/N;
matval=zeros(M+1,N+1);
vetS=linspace(sb,Smax,M+1)';
veti=0:N;
vetj=vetS/ds;
%  建立边界条件
matval(:,N+1)=max(k-vetS,0);
```

```
matval(1,:)  =0;
matval(M+1,:)=0;
% 建立三对角矩阵
alpha=0.25*dt*(sigma^2*vetj.^2-r*vetj);
beta=-dt*0.5*(sigma^2*vetj.^2+r);
gama=0.25*dt*(sigma^2*vetj.^2+r*vetj);
M1=-diag(alpha(3:M),-1)+diag(1-beta(2:M))-diag(gama(2:M-1),1);
[L,U]=lu(M1);
M2=diag(alpha(3:M),-1)+diag(1+beta(2:M))+diag(gama(2:M-1),1);
for i=N:-1:1
   matval(2:M,i)=U\(L\(M2*matval(2:M,i+1)));
end
jdown=floor((s0-sb)/ds);
jup=ceil((s0-sb)/ds)
if jdown==jup
   price=matval(jdown+1,1)
   else

price=matval(jdown+1,1)+(s0-sb-jdown*ds)*(matval(jup+1,1)-matval(jup+1,1
))/ds
end
```

运行结果为：

```
price =
   0.5414
```

【例 7-6】已知股票价格为 50 元，美式看跌期权执行价为 50 元，到期日为 5 个月，股票年波动率的标准差为 0.4，无风险利率为 10%，试用有限元方法求解期权价格。

Crank—Nicholson 方法求解美式看跌期权的 MATLAB 程序如下：

```
clear
s0=50,k=50,r=0.1,T=5/12,sigma=0.4,
dt=T/10;ds=5;
Smax=100;
%amoption(s0,k,rf,sigma,T,dt,ds,Smax)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Crank—Nicolson 法求解美式看跌期权   %
%  时间：2006 年 4 月 12 日            %
%  最新修改时间：2006 年 7 月 6 日      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数说明：
%  s0     0 时刻股价
%  x      期权的执行价
```

```
% r       无风险利率
% T       到期日
% sigma  股票波动的标准差
% Smax    股票最大值
% ds      股票的步长
% dt      时间的步长
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           股票价格、时间初始化          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=round(Smax/ds);
N=round(T/dt);
ds=Smax/M;                  % 重新确定股票的价格步长
dt=T/N;                     % 确定时间的步长
%%%%%%%%%%%%%%%%%%%%%%%%%%%
vetj=1:M-1;
vetj=vetj';
alpha= dt/4*(sigma^2*vetj.^2-r*vetj);
beta =-dt/2*(sigma^2*vetj.^2+r);
gama = dt/4*(sigma^2*vetj.^2+r*vetj);
%%%%%%%%%%%%%%%%%%%%%%%%%%%
H=diag(alpha(2:M-1),-1)+diag(beta(1:M-1))+diag(gama(1:M-2),1);
M1=eye(M-1)-H;M2=eye(M-1)+H;
%%%%%%%%%%%%%%%%%%%%%%%%%%%
f=zeros(M+1,N+1);
f(1,:) =k;                        % 边界条件
f(M+1,:) =0;
f(2:M,N+1)=max(k-vetj*ds,0);          % 初始值
%%%%%%%%%%%%%%%%%%%%%%%%%%%
g=zeros(1,M-1)';g(1)=2*k*alpha(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=N:-1:1
f(2:M,i)=M1^(-1)*[M2*f(2:M,i+1)+g];
f(2:M,i)=max(f(2:M,i),k-vetj*ds);
end
```

$f(j,i)$ 为 i 时刻，j 价位的美式期权价格。计算结果如表 7.3 所示。

表 7.3　显示法求解美式看跌期权价格表

股价	时间										
	10	9	8	7	6	5	4	3	2	1	0
100	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
95	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00	45.00

续表

股价	时间										
	10	9	8	7	6	5	4	3	2	1	0
90	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00
85	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00	35.00
80	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
75	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00
70	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
65	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00	15.00
60	10.21	10.14	10.08	10.03	10.00	10.00	10.00	10.00	10.00	10.00	10.00
55	6.67	6.52	6.37	6.21	6.04	5.84	5.63	5.40	5.17	5.00	5.00
50	4.16	3.98	3.78	3.56	3.31	3.04	2.72	2.34	1.87	1.19	0.00
45	2.50	2.32	2.12	1.91	1.68	1.43	1.15	0.84	0.50	0.16	0.00
40	1.46	1.30	1.14	0.97	0.80	0.62	0.44	0.27	0.12	0.03	0.00
35	0.83	0.71	0.59	0.48	0.36	0.25	0.16	0.08	0.03	0.00	0.00
30	0.47	0.38	0.30	0.23	0.16	0.10	0.05	0.02	0.01	0.00	0.00
25	0.26	0.20	0.15	0.11	0.07	0.04	0.02	0.01	0.00	0.00	0.00
20	0.14	0.11	0.08	0.05	0.03	0.02	0.01	0.00	0.00	0.00	0.00
15	0.08	0.05	0.04	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00
10	0.04	0.03	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
5	0.02	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

从表 7.3 可以看出，股价为 50 元对应的期权价格为 4.16。

思 考 题

已知股票价格为 50 元，执行价为 52 元，到期日为 6 个月，股票年波动率的标准差为 0.3，无风险利率为 5%，试用有限元方法求解期权价格，并和 B-S 公式的计算结果相比较。

第 8 章　蒙特卡洛模拟金融衍生产品定价

本章介绍蒙特卡洛模拟期权定价的内容，要求读者掌握随机数生成方式，了解蒙特卡洛定价就是模拟风险中性测度下标的资产的运动过程，学会蒙特卡洛方法模拟欧式期权定价，掌握提高模拟精度的常用方法。

8.1　随机模拟基本原理

1977 年，菲力浦·伯耶勒(Phelim Boyle)提出了模拟方法求解金融资产定价问题，其想法是假设资产价格分布是随机波动，如果知道了这个波动过程，就可以通过随机模拟不同的路径，每做完一次模拟，就产生了一个最终资产价值，再进行若干次这样的过程，那么所得到的结果就是一个最终的资产价值分布，从这个分布中我们可以得到期望的资产价格。

8.1.1　随机数生成函数

1. 均匀分布随机数生成函数

MATLAB 中的 unidrnd 函数可以生成 1 到 N 的均匀分布随机数。

调用方式

```
R = unidrnd(N)
R = unidrnd(N,m)
R = unidrnd(N,m,n)
```

输入参数

```
N                    %所要生成的随机数个数，在 1～N之间
m                    %确定输出随机矩阵 R 的行数
n                    %确定输出随机矩阵 R 的列数
```

输出参数

```
R                    %随机数矩阵
```

2. 生成服从连续均匀分布的随机数

如果需要生成服从连续分布的随机数，则需要调用 unifrnd 函数，其调用方式如下。

调用方式 1

```
R=unifrnd(A,B)
```

生成位于 A、B 之间的一个随机数。

调用方式 2

```
R=unifrnd(A,B,m)
```

生成位于 A、B 之间的一个随机数。$\boldsymbol{m}$ 为 1×2 向量，分别为 $\boldsymbol{R}$ 的行数与列数。

调用方式 3

```
R=unifrnd(A,B,m,n)
```

生成位于 A、B 之间的一个随机数。$\boldsymbol{m}$、$\boldsymbol{n}$ 分别为输出矩阵 $\boldsymbol{R}$ 的行数与列数。

如生成一个 0 到 1 之间的随机数，代码如下：

```
>> unifrnd(0,1)
  ans =
         0.4565
```

下面介绍两种方法生成 1 到 2 之间的随机矩阵 $\boldsymbol{K}$，$\boldsymbol{K}$ 为 5 行 6 列矩阵。

方法 1

```
>> unifrnd(1,2,[5,6])
ans =
    1.0185    1.9218    1.9169    1.8132    1.6038    1.4451
    1.8214    1.7382    1.4103    1.0099    1.2722    1.9318
    1.4447    1.1763    1.8936    1.1389    1.1988    1.4660
    1.6154    1.4057    1.0579    1.2028    1.0153    1.4186
    1.7919    1.9355    1.3529    1.1987    1.7468    1.0000
```

方法 2

```
>> unifrnd(1,2,5,6)
ans =
    1.5252    1.6813    1.4289    1.3028    1.8600    1.8216
    1.2026    1.3795    1.3046    1.5417    1.8537    1.6449
    1.6721    1.8318    1.1897    1.1509    1.5936    1.8180
    1.8381    1.5028    1.1934    1.6979    1.4966    1.6602
    1.0196    1.7095    1.6822    1.3784    1.8998    1.3420
```

8.1.2 生成正态分布随机数

调用方式

```
R=normrnd(mu,sigma)
```

```
R=normrnd(mu,sigma,m)
R=normrnd(mu,sigma,m,n)
```

输入参数

```
mu              %正态分布的均值
sigma           %正态分布的方差
m               %随机矩阵 R 的行数
n               %随机矩阵 R 的列数
```

如需生成均值为 0，方差为 1 的正态分布随机数，可以执行下面命令：

```
>> normrnd(0,1)
ans =
   -0.4326
```

下面用两种方法生成均值为 0，方差为 1 的正态分布矩阵，矩阵为 5 行 6 列。

方法 1

```
>> normrnd(0,1,[5 6])
ans =
   -1.6656    1.1892    0.7258    1.0668   -1.3362    1.2540
    0.1253   -0.0376   -0.5883    0.0593    0.7143   -1.5937
    0.2877    0.3273    2.1832   -0.0956    1.6236   -1.4410
   -1.1465    0.1746   -0.1364   -0.8323   -0.6918    0.5711
    1.1909   -0.1867    0.1139    0.2944    0.8580   -0.3999
```

方法 2

```
>> normrnd(0,1,5, 6)
ans =
    0.6900    1.1908    0.2573    0.2193    0.6145    0.3803
    0.8156   -1.2025   -1.0565   -0.9219    0.5077   -1.0091
    0.7119   -0.0198    1.4151   -2.1707    1.6924   -0.0195
    1.2902   -0.1567   -0.8051   -0.0592    0.5913   -0.0482
    0.6686   -1.6041    0.5287   -1.0106   -0.6436    0.0000
```

8.1.3　特定分布随机数发生器

MATLAB 中有统一格式的随机数发生器，函数名称为 random，可生成许多服从不同分布的随机数。

调用方式

```
y=random('name',A1,A2,A3,m,n)
```

输入参数

```
name                %表明随机数类型，具体如表 8.1 所示
```

表 8.1　生成特定分布的随机数函数参数表

	贝塔	二项分布	卡方	指数分布	F 分布	伽码	对数正态
分布	beta	Binomial	Chisquare	Exponential	F	Gamma	Lognormal
简写	beta	bino	chi2	exp	f	gam	logn

	均匀分布	泊松分布	T 分布	正态分布	非中心 F 分布	非中心 T 分布
分布	Uniform	Poisson	T	Normal	Noncentral F	Noncentral T
简写	unif	poiss	t	norm	ncf	nct

下面用 random 函数生成正态分布随机数，正态分布的均值为 0，方差为 1，随机矩阵为 3 行 2 列，代码如下：

```
>> a=random('Normal',0,1,3,2)
a =
   -0.4326    0.2877
   -1.6656   -1.1465
    0.1253    1.1909
```

8.1.4　蒙特卡洛模拟方差削减技术

蒙特卡洛模拟精度与模拟次数密切相关，模拟次数越高其精度越高，但是次数增加又会增加计算量。实践表明减少模拟方差可以提高稳定性，减少模拟次数。有很多种方法可以减少方差，如对偶变量技术、控制变量技术、分层抽样、矩匹配、条件蒙特卡洛模拟等，但最简单并且应用最为广泛的是对偶变量技术与控制变量技术。

对偶变量技术就是先随机抽样得到一组数据，然后以此为基础构造出另一组对偶变量。下面以正态分布为例介绍对偶变量技术，首先从正态分布变量中随机抽取 N 个样本值，分别为 $Z_i(i=1,2,3,\cdots,N)$，由此可以得到 N 个模拟值 $C_i(i=1,2,3,\cdots,N)$，那么衍生证券蒙特卡洛模拟估计值为

$$\hat{C}=\frac{1}{N}\sum_i C_i$$

以 $Z_i(i=1,2,3,\cdots,N)$ 为基础，构造对偶随机数 $\tilde{Z}_i=-Z_i$，$\tilde{Z}_i$ 是与 $Z_i(i=1,2,3,\cdots,N)$ 相互对偶的随机数，由正态分布性质可知，$\tilde{Z}_i=-Z_i(i=1,2,3,\cdots,N)$，也是服从正态分布，由对偶随机数生成的估计值为

$$\tilde{C}=\frac{1}{N}\sum_i \tilde{C}_i$$

对 $\tilde{C}$ 和 $\hat{C}$ 取平均得到新的估计值

$$C=\frac{1}{2}(\tilde{C}+\hat{C})=\frac{1}{N}\sum_i\left(\frac{\hat{C}_i+\tilde{C}_i}{2}\right)$$

如果随机抽样的样本 Z_i($i=1,2,3,\cdots,N$)模拟得到的估计值比较小，那么与之对偶的随机抽样样本 $\tilde{Z}_i=-Z_i(i=1,2,3,\cdots,N)$ 得到的估计值可能会偏大，二者的平均值就可能会接近真实值。如果 $\mathrm{cov}(\hat{C}_i,\tilde{C}_i)\leqslant 0$ ，那么

$$\mathrm{var}\left(\frac{\hat{C}_i+\tilde{C}_i}{2}\right)=\frac{1}{2}\mathrm{var}(\hat{C}_i)+\frac{1}{2}\mathrm{cov}(\hat{C}_i,\tilde{C}_i)\leqslant\frac{1}{2}\mathrm{var}(\hat{C}_i)$$

从上面的不等式可以看出，利用对偶技术可以增加估计稳定性，提高了估计精确度。

8.1.5　随机模拟控制变量技术

控制变量技术就是将与所估计的未知变量密切相关的另一个已知量的真实值和估计值之间的差异作为控制量，以提高估计精度。在定价实践中，将这两种衍生证券用相同的随机抽样样本和时间间隔，实施同样的蒙特卡洛模拟过程，能够得到两个模拟估计值，以第二种衍生证券真实值与估计值之间的差异作为控制变量，最后得到第一种衍生证券的蒙特卡洛估计值。

假定 V_1 是需要估计的第一种衍生证券的价值，V_2 是价值容易估计的第二种衍生证券的价值，第一种证券与第二种证券相似，而 $\hat{V}_1$ 和 $\hat{V}_2$ 分别是第一种衍生证券和第二种衍生证券在同样的随机抽样样本的蒙特卡洛估计值，那么利用控制变量技术得到第一种衍生证券的价格估计值为

$$\hat{V}_1^{C1}=\hat{V}_1+(V_2-\hat{V}_2)$$

这里 $V_2-\hat{V}_2$ 就是控制变量，它实际上是第二种衍生证券的蒙特卡洛模拟的估计误差，且上述方程的方差之间的关系为

$$\mathrm{var}(\hat{V}_1^{C1})=\mathrm{var}(\hat{V}_1)+\mathrm{var}(\hat{V}_2)+2\,\mathrm{cov}(\hat{V}_1,\hat{V}_2)$$

如果 $\mathrm{var}(\hat{V}_2)<2\,\mathrm{cov}(\hat{V}_1,\hat{V}_2)$ ，一定有

$$\mathrm{var}(\hat{V}_1^{C1})<\mathrm{var}(\hat{V}_1)$$

因此，当两种衍生证券的协方差很大时，或者当两种衍生证券的价格高度相关时，上述关系是成立的，两种衍生证券的正相关性越强，估计效率越理想。然而从实际应用的角度看，这种控制变量技术的应用十分有限，因此，下面是更一般的控制变量技术，其控制变量的形式为

$$\hat{V}_1^{\beta} = \hat{V}_1 + \beta(V_2 - \hat{V}_2)$$

方差为

$$\mathrm{var}(\hat{V}_1^{\beta}) = \mathrm{var}(\hat{V}_1) + \beta^2 \mathrm{var}(\hat{V}_2) - 2\beta \mathrm{cov}(\hat{V}_1, \hat{V}_2)$$

这是关于控制变量系数 β 的二次三项式，下面的目标是能够找到特殊的 β 使方差 $\mathrm{var}(\hat{V}_1^{\beta})$ 最小，这时只要取 $\beta^* = \dfrac{\mathrm{cov}(\hat{V}_1, \hat{V}_2)}{\mathrm{var}(\hat{V}_2)}$ 就可以保证方差 $\mathrm{var}(\hat{V}_1^{\beta})$ 最小，这种控制变量技术的缺点是 β^* 需要提前知道协方差 $\mathrm{cov}(\hat{V}_1, \hat{V}_2)$ 的信息，而这一般需要靠经验实现。

8.2 蒙特卡洛方法模拟期权定价

8.2.1 蒙特卡洛方法模拟欧式期权定价

在期权计算中，我们可以利用风险中性的方法计算期权的价格。风险中性定价形式如下：

$$f = \mathrm{e}^{-rT} \hat{E}(f_T)$$

其中，f 是期权的价格，f_T 是到期日 T 的现金流，$\hat{E}$ 是风险中性测度。

如果知道了风险中性测度就可以模拟全路径，也可模拟终端价格，例如计算障碍期权等路径依赖型期权时可以模拟全路径，而欧式期权可模拟终端价格。

如果标的资产服从几何布朗运动

$$\mathrm{d}S = \mu S \mathrm{d}t + \sigma S \mathrm{d}W$$

那么风险中性定价的关键在于寻找风险中性测度，对于几何布朗运动，可以证明风险中性测度下，标的资产运动过程如下：

$$S_T = S_0 \exp\left[\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}\varepsilon\right]$$

对于欧式看涨期权，到期日欧式看涨期权现金流如下：

$$\max\{0, S(0)\mathrm{e}^{(r-\sigma^2/2)T+\sigma\sqrt{T}\varepsilon} - K\}$$

其中，K 是执行价，r 是无风险利率，σ 是标准差，ε 是正态分布的随机变量。

对到期日的现金流用无风险利率贴现，就可以知道期权的价格。

【例 8-1】假设股票价格服从几何布朗运动，股票现在价格 $S(0)$=50，欧式期权执行价 K=52，无风险利率 r=0.1，股票波动的标准差 *sigma*=0.4，期权的到期日 T=5/12，试用蒙特卡洛模拟方法计算该期权价格。

下面用 MATLAB 编写一个子程序 blsmc 进行计算。

```
function eucall=blsmc(s0,K,r,T,sigma,Nu)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    蒙特卡洛方法计算欧式看涨期权的价格        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 输入参数
% s0        股票价格
% K         执行价
% r         无风险利率
% sigma     股票波动的标准差
% Nu        模拟的次数
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 输出参数
% eucall    欧式看涨期权的价格
% varprice  模拟期权价格的方差
% ci        95%概率保证的期权价格区间
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
randn('seed',0);    % 定义随机数发生器种子是 0，这样可以保证每次模拟的结果相同
nuT=(r-0.5*sigma^2)*T;
sit=sigma*sqrt(T);
discpayoff=exp(-r*T)*max(0,s0*exp(nuT+sit*randn(Nu,1))-K);  % 期权到期时的
现金流
[eucall,varprice,ci]=normfit(discpayoff)
```

调用子程序可得到欧式看涨期权价格。

```
>> blsmc(50,52,0.1,5/12,0.4,1000)
eucall =
         5.4445
varprice =
         9.1361
ci =
         4.8776
         6.0115
```

从上面的结果可以看到，蒙特卡洛模拟得到的期权价格为 5.4445，样本正态拟合的方差为 9.1361，95%的置信区间为[4.8776,6.0115]，模拟波动的区间还是很大的。

我们用了 normfit 函数对模拟的结果用正态分布函数进行拟合，这不是必需的，主要是为了考察模拟结果的稳定性，如果不需要考察结果是否稳定，也可直接对模拟的结果求均值，此时可将最后一句改为 price=mean(discpayoff)。欧式期权的公式解如下：

```
>> blsprice(50,52,0.1,5/12,0.4)
ans =
   5.1911
```

公式解 5.1911 和模拟值 5.4445 二者之间还是存在较大的差距，增加模拟的次数为 10 000 次时结果如下：

```
>> blsmc(50,52,0.1,5/12,0.4,10000)
eucall =
            5.1338
varprice =
            8.9335
ci =
            4.9587
            5.3089
```

模拟结果为 5.1338，可以看到期权模拟精度有了显著提高，95%概率保证的置信区间为 [4.9587,5.3089]，置信区间较 10 000 次时大大缩小，模拟可靠性显著增加。

下面用对偶方法计算欧式看涨期权的价格。

```
function eucall=blsmc(s0,K,r,T,sigma,Nu)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  蒙特卡洛方法计算欧式看涨期权的价格 (对偶法)  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数:
%  s0         股票价格
%  K          期权执行价
%  r          无风险利率
%  sigma      股票波动的标准差
%  Nu         模拟的次数
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输出参数
%  eucall     欧式看涨期权的价格
%  varprice   模拟的现金流的方差
%  ci         95%概率保证的期权价格区间
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
randn('seed',0);  % 定义随机数发生器的种子是 0，这样可以保证每次模拟的结果相同
nuT=(r-0.5*sigma^2)*T;
sit=sigma*sqrt(T);
rand=randn(Nu,1);
discpayoff=exp(-r*T)*max(0,s0*exp(nuT+sit*rand)-K);% 期权到期时的现金流
discpayoff1=exp(-r*T)*max(0,s0*exp(nuT+sit*-rand)-K);
[eucall,varprice,ci]=normfit([discpayoff;discpayoff1])
```

结果如下：

```
>> blsmc(50,52,0.1,5/12,0.4,10000)
eucall =
```

```
              5.2094
varprice =
              8.9025
ci =
              5.0860
              5.3328
```

模拟的结果为 5.2094，离精确值 5.1911 非常接近，说明对偶技术还是非常有效的。

8.2.2　蒙特卡洛方法模拟障碍期权定价

障碍期权是特殊形式的期权，例如确定一个障碍值 S_b，在期权的存续期内有可能超过该价格，也有可能低于该价格，对于敲出期权而言，如果在期权的存续期内标的资产价格触及障碍值 S_b 时，期权合同可以提前终止执行；相反地对于敲入期权而言，如果标的资产价格触及障碍值 S_b 时，期权合同开始生效。注意障碍值 S_b 可以低于标的资产现在的价格 S_0，也可以高于 S_0。如果 $S_b > S_0$，称为上涨期权，反之称为下跌期权。

对于下跌敲出看跌期权，该期权首先是看跌期权，股票价格是 S_0，执行价是 K，买入看跌期权就首先保证以执行价 K 卖掉股票，下跌敲出障碍期权相当于在看跌期权的基础上附加提前终止执行的条款，内容是当股票价格触及障碍值 S_b 时看跌期权就提前终止执行。因为该期权对于卖方有利，所以其价格应低于看跌期权的价格。

下面考虑下跌敲入看跌期权，同样地该期权首先是看跌期权，下跌敲入期权相当于在看跌期权的基础上附加何时生效的条款，内容是当股票的价格触及障碍值 S_b 时，看跌期权开始生效，综合地看，标准的看跌期权合同可以拆分为两份产品，分别是下跌敲出看跌期权与下跌敲入看跌期权，用公式表示如下

$$P = P_{di} + P_{do}$$

其中，P 是标准看跌期权价格，P_{di} 与 P_{do} 分别表示下跌敲入看跌期权与下跌敲出看跌期权的价格。如果下跌敲出看跌期权提前终止时卖方补偿一些费用给买方，上述公式表示的平价关系就不再有效。

当障碍值确定时，障碍期权存在公式解[①]，其形式如下：

$$P = K\mathrm{e}^{-rT}\{N(d_4) - N(d_2) - a[N(d_7) - N(d_5)]\}$$
$$-S_0\{N(d_3) - N(d_1) - b[N(d_8) - N(d_6)]\}$$

其中，S_0 是股票价格，S_b 是障碍值，K 是看跌期权执行价，T 是存续期，r 是无风险利率，σ 是波动率的标准差，其他参数如下：

① 见 P.Wilmot Quantitative Finance．Wily, Chichester, West Sussex, England, 2000。

$$a=\left(\frac{S_b}{S_0}\right)^{-1+2r/\sigma^2}$$

$$b=\left(\frac{S_b}{S_0}\right)^{1+2r/\sigma^2}$$

以及

$$d_1=\frac{\ln(S_0/K)+(r+\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2=\frac{\ln(S_0/K)+(r-\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_3=\frac{\ln(S_0/S_b)+(r+\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_4=\frac{\ln(S_0/S_b)+(r-\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_5=\frac{\ln(S_0/S_b)-(r-\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_6=\frac{\ln(S_0/S_b)-(r+\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_7=\frac{\ln(SK/S_b^2)-(r-\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_8=\frac{\ln(SK/S_b^2)-(r+\sigma^2/2)T}{\sigma\sqrt{T}}$$

利用上面的公式编写下跌敲出障碍期权价格的程序如下：

```
function   P=DownOutPut(S0,K,r,T,sigma,Sb)
a=(Sb/S0)^(-1+2*r/sigma^2);
b=(Sb/S0)^( 1+2*r/sigma^2);
d1=(log(S0/K)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2=(log(S0/K)+(r-sigma^2/2)*T)/(sigma*sqrt(T));
d3=(log(S0/Sb)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d4=(log(S0/Sb)+(r-sigma^2/2)*T)/(sigma*sqrt(T));
d5=(log(S0/Sb)-(r-sigma^2/2)*T)/(sigma*sqrt(T));
d6=(log(S0/Sb)-(r+sigma^2/2)*T)/(sigma*sqrt(T));
d7=(log(S0*K/Sb^2)-(r-sigma^2/2)*T)/(sigma*sqrt(T));
d8=(log(S0*K/Sb^2)-(r+sigma^2/2)*T)/(sigma*sqrt(T));
P=K*exp(-r*T)*(normcdf(d4)-normcdf(d2)-a*(normcdf(d7)-normcdf(d5)))...
        -S0*(normcdf(d3)-normcdf(d1)-b*(normcdf(d8)-normcdf(d6)));
```

【例 8-2】我们考虑一个欧式看跌股票期权。股票的价格为 50，看跌期权执行价为 50，

无风险利率为 0.1，时间为 5 个月，股票年波动率的标准差为 0.4。

首先公式的解如下：

```
>> [Call,Put]=blsprice(50,50,0.1,5/12,0.4)
Call =

   6.1165
Put =
   4.0760
```

看跌期权价格为 4.076。对于上述看跌期权，进一步地，我们考虑障碍值 S_b 等于 40 时下跌敲出期权的价格：

```
>> P=DownOutPut(50,50,0.1,5/12,0.4,40)
P =
   0.5424
```

由于该下跌敲出看跌期权提供的条件过于优厚，买方承担了大量风险，作为回报，其价格较看跌期权便宜许多。

下面用蒙特卡洛方法模拟下跌敲出看跌期权价格，在模拟中我们给出了模拟次数为 NRepl，每次模拟时间分为 NSteps 步离散，障碍值为变量 S_b，其现金流如下：

当 $S_t < S_b$ 时，$CashFlow = 0$。

我们可以先模拟路径，然后让大于 S_b 路径的现金流为 0，程序如下：

```
% DOPutMC (s0,k,r,T,sigma,sb,NSteps,NRepl)
function [P,aux,CI]=DOPutMC(s0,k,r,T,sigma,sb,NSteps,NRepl)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      利用蒙特卡洛方法对欧式下跌敲出期权定价      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数
%      s0        股票价格
%      k         执行价
%      r         无风险利率
%      T         期权续存期
%      sigma     股价波动的标准差
%      sb        障碍值
%      NSteps    时间离散数目
%      NRepl     路径数目
%  输出参数
%      P         下跌敲出看跌期权价格
%      CI        蒙特卡洛模拟的方差
%      NCrossed    95%置信度的价格区间
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        用蒙特卡洛方法模拟风险中性下股价路径         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dt=T/NSteps;
nudt=(r-0.5*sigma^2)*dt;
sidt=sigma*sqrt(dt);
randn('seed',0);
rand=randn(NRepl,NSteps);
rand1=nudt+sidt*rand;
rand2=cumsum(rand1,2);  % 沿列方向逐列累加
path=s0*exp(rand2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              利用路径进行定价                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
payoff=zeros(NRepl,1);  % 设定现金流初值为 0
for i=1:NRepl
   ax=path(i,:);
   if min(ax)<sb
      payoff(i)=0;  %如果路径中的任意一点价格低于障碍值，现金流为 0
   else
      payoff(i)=max(0,k-ax(NSteps));
   end
end
[P,aux,CI]=normfit(exp(-r*T)*payoff) ; % P 为期权价格
```

运行程序：

```
>> s0=50;k=50;r=0.1;T=5/12;sigma=0.4;sb=40;NSteps=600;NRepl=10000;
>> [P,aux,CI]=DOPutMC(s0,k,r,T,sigma,sb,NSteps,NRepl)
P =
   0.5921
aux =
   1.6687
CI =
   0.5594
   0.6248
```

该期权模拟的价格为 0.5921，和前面的公式解 0.5424 尚存在差距，增加模拟次数，代码如下：

```
>> s0=50;k=50;r=0.1;T=5/12;sigma=0.4;sb=40;NSteps=600;NRepl=20000;
>> [P,aux,CI]=DOPutMC(s0,k,r,T,sigma,sb,NSteps,NRepl)
P =
   0.5597
aux =
```

```
    1.6022
CI =
    0.5375
    0.5819
```

结果和公式解比较接近，如果将存续期改为 2/12 年，再考察其价格变化，代码如下：

```
>> s0=50;k=50;r=0.1;T=2/12;sigma=0.4;sb=40;NSteps=60;NRepl=50000;
>> [P,aux,CI]=DOPutMC(s0,k,r,T,sigma,sb,NSteps,NRepl)
P =
    1.3527
aux =
    2.3664
CI =
    1.3319
    1.3734
```

从运行结果可知，期权价格为 1.3527，模拟期权方差为 2.3664，期权 95%置信区间为 [1.3319,1.3734]，障碍期权价格较上面的结果增加了很多。

8.2.3　蒙特卡洛方法模拟亚式期权定价

亚式期权是一种路径依赖型期权，它的收益函数依赖于期权存续期内标的资产的平均价格。平均价格分算术平均和几何平均两种，对于离散算术平均价格定义为

$$A_{da}=\frac{1}{n}\sum_{i=1}^{n}S(t_i) \tag{8.1}$$

其中 $t_i(i=1,2\cdots n)$ 是离散时间样本点。

离散几何平均价格定义为

$$A_{dg}=\left[\prod_{i=1}^{n}S(t_i)\right]^{1/n} \tag{8.2}$$

亚式看涨期权到期现金流为

$$\max\left\{\frac{1}{N}\sum_{i=1}^{N}S(t_i)-K,0\right\}\quad,\quad t_i=i\delta t,\ \delta t=T/N$$

其中，K 是执行价，$S(t_i)$ 是 t_i 时刻的股价。

【例 8-3】股票价格为 50，亚式看涨期权执行价为 50，存续期为 5 个月，期权到期现金流是每月均价与执行价之差，股票波动率的标准差为 0.4，无风险利率为 0.1，下面用蒙特卡洛方法计算该亚式期权价格。

该期权定价程序为：

```
% AsianMC.m
```

```
% function [P,CI]=AsianMC(s0,k,r,T,sigma,NSteps,NRepl)
s0=50;k=50;r=0.1;T=5/12;sigma=0.4;NSteps=5;NR7epl=50000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     利用蒙特卡洛方法对亚式期权定价          %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数
%      s0        股票价格
%      k         亚式看涨期权执行价
%      r         无风险利率
%      T         期权续存期
%      sigma     股价波动的标准差
%      NSteps    时间离散数目
%      NRepl     模拟路径数目
%  输出参数
%      P         亚式期权价格
%      CI        蒙特卡洛模拟的方差
%      NCrossed  95%置信度的价格区间
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          用蒙特卡洛方法模拟股价路径             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dt=T/NSteps;
nudt=(r-0.5*sigma^2)*dt;
sidt=sigma*sqrt(dt);
randn('seed',0);
rand=randn(NRepl,NSteps);
rand1=nudt+sidt*rand;
rand2=cumsum(rand1,2);  % 沿列方向逐列累加
path=s0*exp(rand2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            利用路径进行定价             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
payoff=zeros(NRepl,1);  % 设定现金流初值为 0
for i=1:NRepl
  payoff(i)=max(0,mean(path(i,:))-k);
end
[P,aux,CI]=normfit(exp(-r*T)*payoff)
```

运行后得到:

```
P =
    3.9622
aux =
    5.9669
CI =
```

```
3.9099
4.0145
```

这是一个比较粗糙的估计，我们可以用控制变量(Control Variables)技术提高估计精度。构造

$$Y = \sum_{i=0}^{N} S(t_i)$$

显然它和收益函数是相关的，而

$$E[Y] = E\left[\sum_{i=0}^{N} S(t_i)\right] = \sum_{i=0}^{N} E[S(i\delta t)]$$
$$= \sum_{i=0}^{N} S(0)\mathrm{e}^{ir\delta t} = S(0)\sum_{i=0}^{N} [\mathrm{e}^{r\delta t}]^i = S(0)\frac{1-\mathrm{e}^{r(N+1)\delta t}}{1-\mathrm{e}^{r\delta t}}$$

($E[S(t)/S(0)] = \mathrm{e}^{rt}$)，这样用控制变量法可以将前面程序改为：

```
% AsianMCCV.m
function [P,CI]=AsianMCCV(S0,X,r,T,sigma,NSamples,NRepl,NPilot)
TryPath=AssetPaths1(S0,r,sigma,T,NSamples,NPilot); %生成路径
StockSum=sum(TryPath,2);
PP=mean(TryPath(:,2:(NSamples+1)),2);
TryPayoff=exp(-r*T)*max(0,PP-X);
MatCov=cov(StockSum,TryPayoff);
c=-MatCov(1,2)/var(StockSum);
dt=T/NSamples;
ExpSum=S0*(1-exp((NSamples+1)*r*dt))/(1-exp(r*dt));
% MC run
ControlVars=zeros(NRepl,1);
for i=1:NRepl
   StockPath=AssetPaths1(S0,r,sigma,T,NSamples,1);
   Payoff(i)=exp(-r*T)*max(0,mean(StockPath(2:(NSamples+1)))-X);
   ControlVars(i)=Payoff(i)+c*(sum(StockPath)-ExpSum);
end
[P,aux,CI]=normfit(ControlVars);
function Spaths=AssetPaths1(S0,mu,sigma,T,NSteps,NRepl)
dt=T/NSteps;
nudt=(mu-0.5*sigma^2)*dt;
sidt=sigma*sqrt(dt);
Increments=nudt+sidt*randn(NRepl,NSteps);
LogPaths=cumsum([log(S0)*ones(NRepl,1),Increments],2)
Spaths=exp(LogPaths);
```

下面是控制变量模拟的结果：

```
>> randn('seed',0)
```

```
>> [P,CI]=AsianMCCV(50,50,0.1,5/12,0.4,5,45000,50000)
P =
   3.9809
CI =
   3.9581
   4.0036
```

根据前面蒙特卡洛模拟的结果有：

```
>> randn('seed',0)
>> [P,CI]=AsianMC(50,50,0.1,5/12,0.4,5,50000)
P =
   3.9618
CI =
   3.9100
   4.0136
```

可以看出控制变量使得估计的置信区间缩小了。

8.2.4 蒙特卡洛模拟经验等价鞅测度

等价鞅理论是金融衍生品定价的重要方法，衍生品价格就是在等价鞅测度下对衍生品现金流用无风险利率贴现，可以用下面方程表示：

$$S_0 = \mathrm{e}^{-rT} E^Q[S_T | F_0]$$

其中，S_0 表示股票价格，r 为无风险利率，F_0 表示 0 时刻信息，$E^Q[\cdot|F_0]$ 表示等价鞅测度下算子。

Duan 和 Simonate(1998)提出了乘数调整法，该方法可以保证等价鞅性质，当标的资产服从几何布朗运动时，等价鞅下的资产运动满足下列条件：

$$S_T = S_0 \exp\left[\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}\varepsilon\right]$$

其中 $\varepsilon \sim N(0,1)$，这样模拟的第 i 条路径为

$$S_{i,T} = S_0 \exp\left[\left(r - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}\varepsilon_{i,T}\right]$$

这时我们可以构造一个等价鞅测度

$$S_{i,t}^*(i) = S_0 \exp(rT) \frac{\hat{S}_{i,T}}{\sum \hat{S}_{i,T} / M}$$

$S^*(\cdot)$ 测度实际上在第 i 条路径的权重上乘以 $S_0 \exp(rT)$，当样本容量趋于无穷大时有

$$e^{(-rT)}\frac{1}{M}\sum_{i=1}^{M}S_{i,t}^{*}=S_0$$

$S_{i,t}^{*}(i)$ 是风险中性测度下的新样本。

对于例 8-1 我们重新编写程序如下：

```
% function eucall=blsmc(s0,K,r,T,sigma,Nu)
s0=50;K=52;r=0.1;T=5/12;sigma=0.4;Nu=1000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     蒙特卡洛经验等价鞅方法计算欧式看涨期权价格      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输入参数
%  s0       股票价格
%  K        执行价
%  r        无风险利率
%  sigma    股票波动的标准差
%  Nu       模拟次数
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  输出参数
%  eucall     欧式看涨期权价格
%  varprice 模拟现金流的方差
%  ci         95%概率保证的期权价格区间
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 定义随机数发生器种子是 0，这样可以保证每次模拟的结果相同
randn('seed',0);
nuT=(r-0.5*sigma^2)*T;
sit=sigma*sqrt(T);
% 期权到期时的现金流
discpayoff=exp(-r*T)*max(0,s0*exp(nuT+sit*randn(Nu,1))-K);
disp('蒙特卡洛模拟结果')
[eucall,varprice,ci]=normfit(discpayoff)
SM=s0*exp(nuT+sit*randn(Nu,1));
SM=s0*exp(r*T)*SM/mean(SM);
S1=max(0,SM-K);
disp('风险中性下欧式看涨期权结果')
[Emscall,varprice,ci]=normfit(S1)
disp('欧式看涨期权解析解')
blsprice(50,52,0.1,5/12,0.4)
```

运行结果如下：

```
蒙特卡洛模拟结果
eucall =
    5.4445
varprice =
```

```
    9.1361
ci =
    4.8776
    6.0115
风险中性下欧式看涨期权结果
Emscall =
    5.4320
varprice =
    9.8321
ci =
    4.8219
    6.0422
欧式看涨期权解析解
ans =
    5.1911
```

从结果可以看出，经验等价鞅定价误差较单电蒙特卡洛模拟稍有改进。

思 考 题

1．宝钢权证是我国第一家权证，观察宝钢权证哪一段时间价格和 B-S 模型相同，并分析原因，并用蒙特卡洛方法进行模拟，观察其预测的效果，并用行为金融理论解释。

2．试分析蒙特卡洛方法求解欧式看跌期权的具体过程。

3．我国证券市场正在尝试建立投资者保护资金，用以对冲投资者损失，实际是对于投资损失提供补偿的一种保险产品。试分析如果建立一家保险公司，接受每个投资者的投保，补偿一定额度损失，用蒙特卡洛方法衡量将来 5、10、20、60 个交易日的损失分布，并对将来的损失计提适度准备金，分析其可行性，进一步可考虑该种产品与权证产品有何不同。

4．权证产品并不完全是根据风险定价，我国的权证产品是股权分置改革的产物，第一个股票权证是宝钢看涨权证(代码 580000)，2005 年 8 月 22 日上市流通，2006 年 8 月 23 日到期，该权证行权价为 4.2 元，上市公司为了回避行权，打压股价。此外权证也可成为收购手段，通过行权增持公司股份，这些都为权证定价增加了新的内容，试考虑在蒙特卡洛方法对衍生产品定价时如何考虑这些因素。

第 9 章　金融数据可视化技术

本章主要介绍了 MATLAB 中常见的绘图函数，要求读者能根据不同的需求选择合适的绘图函数，熟悉对坐标轴、图形线条的修改，学会改变坐标轴刻度以满足实际需要，了解绘图函数参数的赋值方法。

9.1　图形对象、对象句柄和句柄图形结构

金融数据可视化能使金融研究人员直接感受到数据背后的内在本质，是人们研究科学、认识世界所不可缺少的手段。MATLAB 不仅在数值计算方面是一款优秀的科技应用软件，在数据可视化方面也具有上佳表现。

MATLAB 具有二维、三维乃至四维图形表现能力，可以从线型、边界面、色彩、渲染、光线、视角等方面把数据特征表现出来。MATLAB 视图化功能是建立在一组“图形对象”基础之上，“图形对象”的核心是图形句柄(Granhics Handle)操作。

最常用的是高层绘图指令，该指令简单明了容易掌握。本章介绍高层绘图指令,内容按“前易后难”次序安排。

9.1.1　MATLAB 中图形图像基本内容

句柄图形是构建 MATLAB 图形图像系统最重要的元素，属于 MATLAB 图形系统中的底层系统(Low-Level)部分，它所支持的指令可以直接创建线、文字、网线、曲面以及用户界面，也是学习图像用户界面 GUI(Graphical User Interfaces)技术的入门。MATLAB 中的图形都可以用句柄函数进行修改。MATLAB 7.1 对句柄图形对象的结构和元素做了较大调整，增加了部分图形对象，并对图形对象和场景对象重新进行了分类。

MATLAB 分底层与高层两个层次的绘图指令。

(1)　底层(Low-level)绘图指令：直接对句柄进行操作。

底层绘图指令控制和表现数据图形的能力比高层绘图指令强，优点是灵活多变，缺点在于较难掌握。

(2)　高层(High-level)绘图指令：建立在底层绘图指令上的绘图指令。

1. 图形对象

MATLAB 中一幅图形是由众多图形对象构成的，包括根对象(root)、图形对象(figure)、轴对象(axes)、用户控件(uicontrol)、用户菜单(uimenu)、用户界面说明菜单(uicontextmenu)、图像(image)、光线(light)、线条(line)、面片(patch)、矩形对象(rectangle)、曲面(surface)、文本(text)等，MATLAB 中各种图像对象的关系如图 9.1 所示。

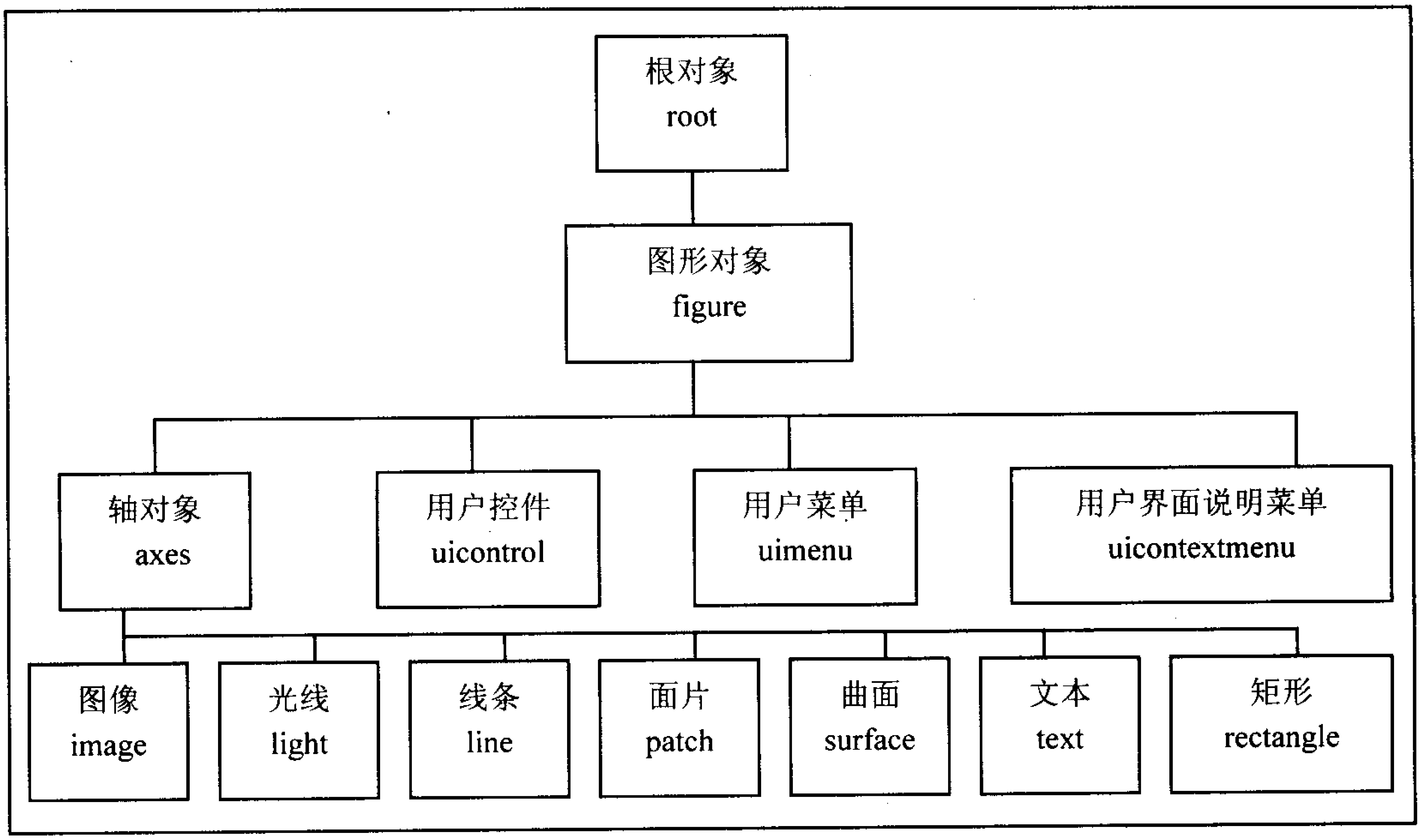


图 9.1　句柄图形对象结构树

图形对象分为父对象与子对象，父对象是子对象的上一级，父对象影响其所有子对象。句柄图像内容如表 9.1 所示。

表 9.1　句柄图像内容注解

对　象	父　代	属　性
Root	—	所有图像都是根对象的子对象，根对象句柄值为 0
Figure	Root	屏幕上窗口是一个图像对象，对象句柄值是个整数
Axes	Figure	轴对象在窗口定义一个图像区域，可以用来描述子对象的位置与方向
Uicontrol	Figure	用户界面控制，响应鼠标动作
Uimenu	Figure	创建一个菜单窗口，通过菜单可以控制程序

续表

对　象	父　代	属　性
uicontextmenu	Figure	创建一个图形对象快捷菜单，点击时出现菜单
Image	Axes	用当前色彩创建一个图像
Light	Axes	定义多边形或者曲面的光照
Line	Axes	用 plot、plot3、contour、contour3 等创建的简单图像
Patch	Axes	创建曲片对象
Rectangle	Axes	创建矩形、椭圆方框，可以填入数据
Surface	Axes	定义 4 个角曲面，形成一个个小网格
Text	Axes	图像中输入文字

2. 句柄对象

MATLAB 中的句柄图形系统以面向对象为基础，一幅图的每一个组成部分都是一个对象，每一个对象都有一系列句柄相对应，并且通过修改句柄值可以达到修改图像的目的。根屏幕的句柄值总是 0，图形窗口值总是一个整数，其他图像句柄总是一个浮点数，在使用时，一般先把句柄值赋给一个变量后再调用。为查找方便一般都用大写字母给句柄命名，相应的对象可以在“h”后面加上描述符，便于识别，句柄可以是单值也可以是矩阵。

MATLAB 中获得句柄的方式有两种：直接由高级绘图函数和句柄函数获得，下面分别介绍。

绘图函数句柄内容如表 9.2 所示。

表 9.2　绘图函数内容

函 数 名	功　能	调用方式
figure	创建图形对象	h=figure(n)，创建句柄为 n 的图像窗口，n 为整数
axes	创建轴对象	h=axes(‘position’,[left,bottom,width,height])
uicontrol	用户界面控制	h=unicontrol(‘property’,value)，property 为指定界面的控制类型，value 为其值
uimenu	用户菜单控制	h=uimenu(‘property’,value)，property 为指定窗口上方的菜单形式，value 为其值
image	创建并显示图像对象	h=image(x)，x 为图像数据矩阵
line	画线	h=line(x,y,z)，绘制 x、y、z 确定的直线，如果不指定 z，则在 x-y 平面上画线
patch	创建并显示多边形	h=patch(‘facess’,fac,‘vertices’,vert)，fac 为定义多边形的顶点序号矩阵，vert 为顶点矩阵

续表

函数名	功　能	调用方式
surface	创建空间曲面	h=surface(x,y,z,c)，x，y，z 为三维曲面上的点，c 为色彩矩阵
text	创建文本	h=text(x,y,‘string’)，x,y 为文本坐标，string 为文本内容

利用句柄函数可以得到现有的图形句柄，也可以查找已经创建的图像对象，并且可以赋值给其他变量，还可以作为输入变量使用，下面分别是这几个函数的调用格式。

(1) gco：返回当前对象的句柄。

除了命令 uimenus 之外，“当前对象”为最后用鼠标单击的对象，若鼠标没有单击到一图形对象之下的子对象，则该图形对象为“当前对象”，系统会把当前图形对象的句柄存放于图形的属性 Currentobject 中，图形窗口中的“当前对象”是正在执行的对象，例如：

```
h = gco %  返回当前对象的句柄给 h
h = gco(figure_handle) % 返回指定窗口 figure_handle 中当前对象值
```

(2) gcf：获得当前图形窗口的句柄。

h = gcf 返回当前图形窗口句柄，一般为命令 plot、title 与 surf 等得到的结果，若不存在图形窗口，则系统自动地生成一个，并返回它的句柄，若图形窗口不存在时，也不创建新的，则输入“get(0,'CurrentFigure')”。

(3) gca：获得当前轴对象的句柄。

h=gca 返回当前图形窗口中坐标轴的句柄，若坐标轴不存在，则系统自动生成一坐标轴同时返回它的句柄，用户如果想查看上面结果，可以输入“get(gcf, CurrentAxes)”，当前坐标轴为用户创建坐标轴以下子对象的目的地。有许多图形命令可以在当前坐标轴中画出图形对象，如 plot、text、surf 等，如果改变了当前窗口，相应地改变了当前坐标轴。

(4) clf：清除当前图形窗口。

该命令在命令窗口中执行与在回调程序中执行效果是一样的，它不能区别由 callback 设置的属性 HandleVisibility，也就是说，当它在一回调程序中执行时，命令 clf 仅仅删除属性 HandleVisibility 为 on 的图形对象。clf reset 无条件地清除当前图形窗口中的所有图形对象，且重新设置图形窗口属性为缺省值，除了属性 Position、Units、PaperPosition、PaperUnits。

(5) close：删除指定的图形窗口。

close(h) 表示删除由句柄 h 指定的图形窗口，若 h 为一向量或矩阵，则 close 全部删除每一分量的指定图形句柄，例如：

```
close name          %删除指定名字 name 的窗口
close all              %删除所有没有隐藏的图形
```

```
close all hidden %删除所有隐藏的图形
status=close(...)  %若成功地删除了指定对象则返回 status=1，否则返回 0
```

(6) get：获取对象属性。

get(h) 返回由句柄 h 指定图形对象的所有属性及相应的当前属性值，get(h,'PropertyName') 返回由句柄 h 指定的图形对象属性 PropertyName 的属性值。<m-by-n value cell array> = get(h,<property cell array>) 返回由 m 个图形对象 n 个属性值组成的 m×n 阶单元数组，其中 m=length(h)，且 n 为指定的属性单元数组<property cell of array>中包含属性名的个数。

a = get(h) 返回一结构，域名为该对象属性名，结构域名值为相应属性的当前值。h 必须为标量，若用户没有指定输出参量，则系统将信息显示于屏幕。

a = get(0,'Factory') 返回所有能由用户设置属性的缺省定义值。输出参量 a 为一结构数组，该结构的域名为对象属性名，域名值为相应属性的当前值，若用户没有指定输出参量，则系统将信息显示于屏幕。

a = get(0,'FactoryObjectTypePropertyName') 返回指定对象属性的缺省属性值。输入参量 FactoryObjectTypePropertyName 为一关键字，由字符 Factory 与对象类型(如 Figure)还有属性名(如 Color)组成，如 FactoryFigureColor。

a = get(h,'Default') 返回由句柄 h 指定的对象的所有缺省属性值。输出参量 a 为一结构数组，该结构的域名为缺省值对应的属性名。若用户没有指定输出参量，则系统将该结构信息显示于屏幕。

a = get(h,'DefaultObjectTypePropertyName') 返回对象类型的指定属性的缺省属性值。输入参量 DefaultObjectTypePropertyName 为一关键字，由字符 Default 与对象类型名(例如 Figure)还有具体属性名(例如 Color)组成，如 DefaultFigureColor。

(7) set：设置对象属性。

set(H,'PropertyName',PropertyValue,…)表示用属性值 PropertyValue 设置用参量 H 标识的对象(一个或多个)属性名 PropertyName (一个或多个)。H 可以为一句柄向量。在这种情形下，命令 set 可以设置所有对象的属性值。

set(H,a)表示用指定属性值设置由 H 标识的对象属性。其中 a 为一结构数组，该结构数组的域名为对象属性名，域名值为相应属性名的属性值。

set(H,pn,pv)表示对由 H 指定的所有对象中，对单元数组属性值名为 pn 的赋值为 pv。

set(H,pn,<m-by-n cell array>)表示对于每 m 个图形对象设置 n 个属性值，其中 m=length(H)，n 为包含属性名的单元数组 pn 中包含的属性名个数。即允许用户对每一对象指定的属性设置不同属性值。

a=set(h) 返回句柄 h 中允许用户设置的属性名与可能的属性值。输出参量 a 为一结构数组，其域名为对象属性名，域名值为相应属性名对应的属性值。若没有指定输出参量 a，则

系统自动将信息显示于屏幕。h 必须为标量。

a= set(0,'Factory') 返回那些用户可以设置缺省值的所有对象属性名，同时显示可能的属性值。输出参量 a 为一结构数组，其域名为对象的属性名，域名值为相应属性名对应的属性值，若没有指定输出参量 a，则系统自动将信息显示于屏幕。

a=set(0,'FactoryObjectTypePropertyName') 返回指定根对象(0)类型中指定的属性名 ObjectTypePropertyName 的所有可能属性值。输入参量是由固定关键字 Factory、对象类型(如 axes)与属性名(如 position 等)组成。

a=set(h,'Default') 返回由 h 标记的对象的缺省设置值，其中 h 必须是标量。

a=set(h,'DefaultObjectTypePropertyName') 返回指定对象 h 类型中指定属性名 ObjectTypePropertyName 的所有可能的属性值。输入参量由固定关键字 Factory、对象类型(如 axes)与属性名(如 position 等)组成。

(8) Delete：删除当前图像。

3. 绘制可视化二维图形

步骤如表 9.3 所示。

表 9.3 绘图步骤

步骤		典型指令
1	数据准备； 确定变量的范围； 产生自变量的采样向量； 计算相应函数值	T=pi*(0:100)/100; Y=sin(t).*sin(9*t);
2	选定图形窗位置； 缺省时打开第一个窗口，当前窗口、子图； 可用指令指定图形窗号与子图	figure(1) %指定 1 号图形窗 subplot(1,2,1) %指定 1 号子图
3	调用(高层)绘图指令； 线条、色彩、数据点型	plot(t,Y, 'b-') %用蓝色实线画曲线
4	设置坐标轴范围与刻度、坐标分格线	axis([0,pi,-1,1]) %设置坐标轴范围 Grid on %坐标分格
5	图形注释； 图名、坐标名、图例、文字说明	title('调制波形') %图名 xlabel('t');ylabel('9t');
6	图像精细修饰； 利用对象属性设置； 利用图形窗工具条设置	set(h, 'MarkerSize',10) %数据点大小

续表

	步　骤	典型指令
7	打印； 图像窗中打印菜单； 利用图形后台处理软件打印	采用图形选项或按键打印最简捷 Print –dps2 %专业打印

9.1.2　金融时间序列基本绘图函数

1. 收益率与频率直方图

【例 9-1】浦发银行(600000)从 2006 年 5 月 24 日～2006 年 7 月 24 日每个交易日的收盘价保存在变量 price 中，现画出其对数收益率直方图，代码如下：

```
>> ret=price2ret(price);  % price2ret 函数把股票价格转换为对数收益率
>> bar(ret)
>> xlabel('天  数');ylabel('收益率');
>> title('浦发银行(600000)对数收益率直方图');
```

图 9.2 是浦发银行的对数收益率直方图。

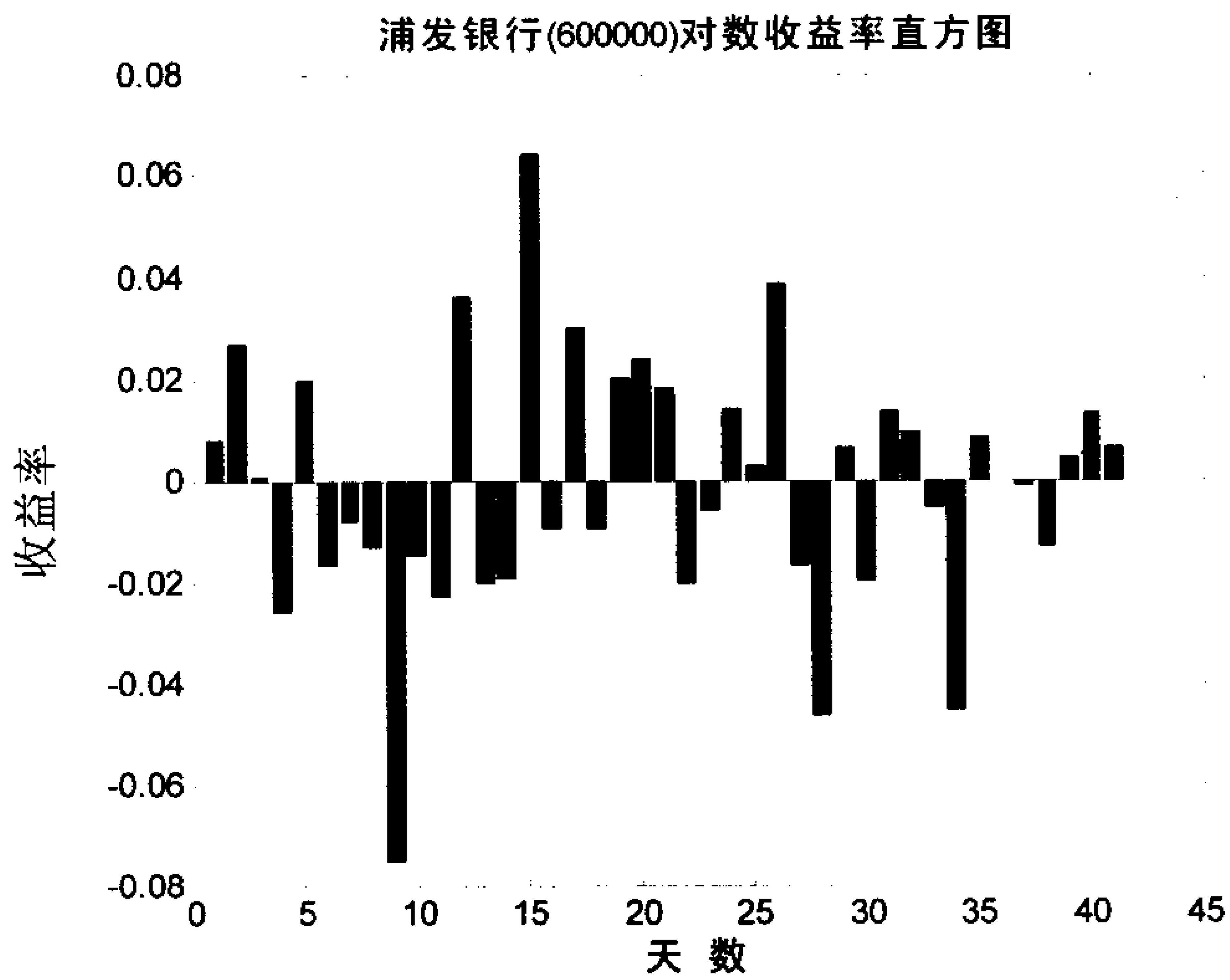


图 9.2　浦发银行对数收益率直方图

【例 9-2】根据浦发银行数据，作出收益率三维直方图，代码如下：

```
>> bar3(ret)
>> title('浦发银行(600000)对数收益率三维图');
>> xlabel('天  数');ylabel('频  率');
```

图 9.3 是浦发银行对数收益率三维直方图。

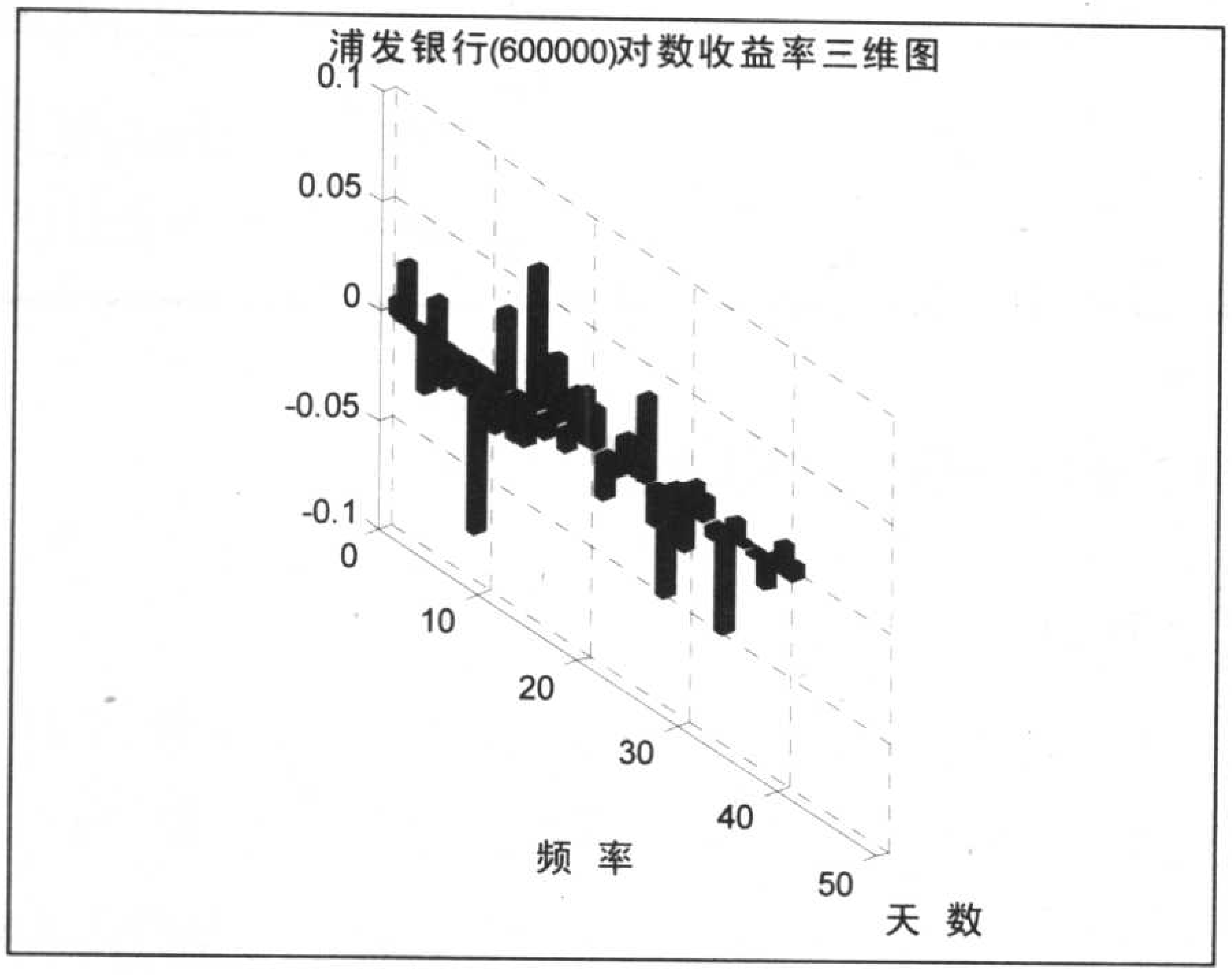


图 9.3　浦发银行对数收益率三维图

【例 9-3】 用上例中浦发银行数据，我们画出其对数收益率频率直方图，代码如下：

```
>> hist(ret);  %画频率直方图
>> title('浦发银行(600000)对数收益率直方图');
>> xlabel('收益率');ylabel('频  率');
```

浦发银行对数收益率频率统计图如图 9.4 所示。

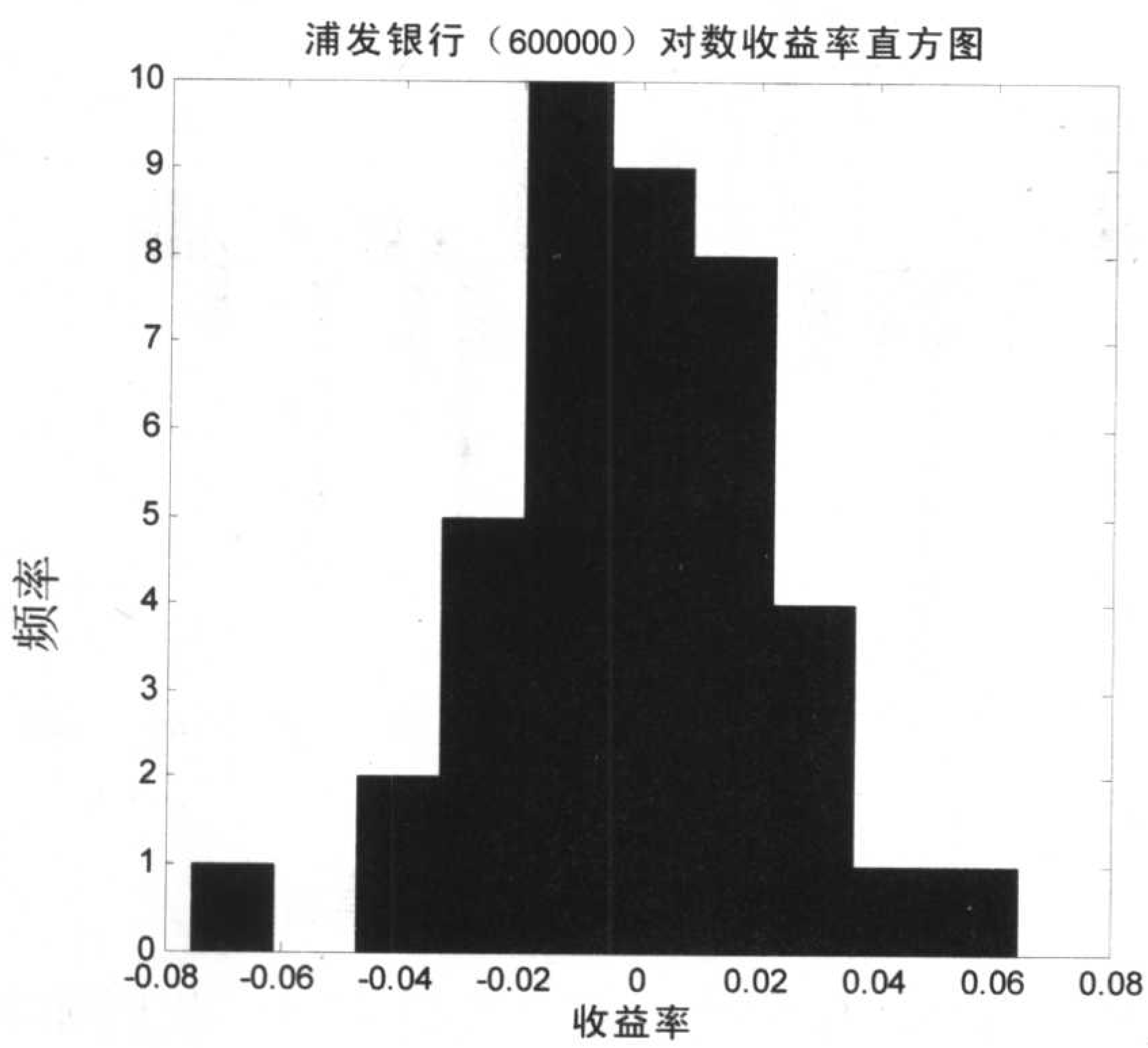


图 9.4　浦发银行对数收益频率直方图

从图 9.4 可以看出股票收益率有肥尾趋势，说明股票收益率风险集中在尾部，这也和通常的金融时间序列肥尾特征相吻合。

2. 双坐标图

在金融时间序列中常常需要把两组数据画在同一张图中，例如两只股票的股价画在同一幅图上，Plotyy 函数可以实现该功能。plotyy 函数是绘制双 Y 轴曲线函数，下面用一个例子加以说明。

上证指数与沪深 300 指数及其收益率如表 9.4 所示，现在画出收益率双坐标图。

表 9.4 上证指数和沪深 300 指数及其收益率表

	上证指数	沪深 300	上证指数收益	沪深 300 收益
2006-7-19	1645.16	1336.64	−0.02321	−0.02678
2006-7-20	1655.12	1345.19	0.006054	0.006397
2006-7-21	1665.33	1356.03	0.006169	0.008058
2006-7-24	1665.94	1358.12	0.000366	0.001541
2006-7-25	1685.46	1374.17	0.011717	0.011818
2006-7-26	1686.65	1371.30	0.000706	−0.00209
2006-7-27	1675.17	1355.55	−0.00681	−0.01149
2006-7-28	1662.03	1341.39	−0.00784	−0.01045
2006-7-31	1612.73	1294.33	−0.02966	−0.03508

在 MTALAB 中执行如下命令：

```
>>
a=[-0.02321;0.006054;0.006169;0.000366;0.011717;0.000706;-0.00681;-0.007
84;…
-0.02966];
>>
b=[-0.02678;0.006397;0.008058;0.001541;0.011818;-0.00209;-0.01149;-0.010
45;…
-0.03508];
>> [ax,h1,h2]=plotyy(1:9,a,1:9,b,'plot')
ax =
  153.0023  155.0027
h1 =
  154.0066
h2 =
  156.0046
>>grid on
```

```
>>% 标注左边 Y 轴
>> set(get(ax(1),'Ylabel'),'string','上证指数收益率');
>>% 标注右边 Y 轴
>> set(get(ax(2),'Ylabel'),'string','沪深 300 收益率');
>>box on  % 为图形加边框
>>set(h1,'LineStyle',':');
```

图 9.5 是上证指数与沪深 300 指数收益率双坐标图。

注意，legend 函数不能使用于标识双 Y 轴图，只能用线条特征加以区分。

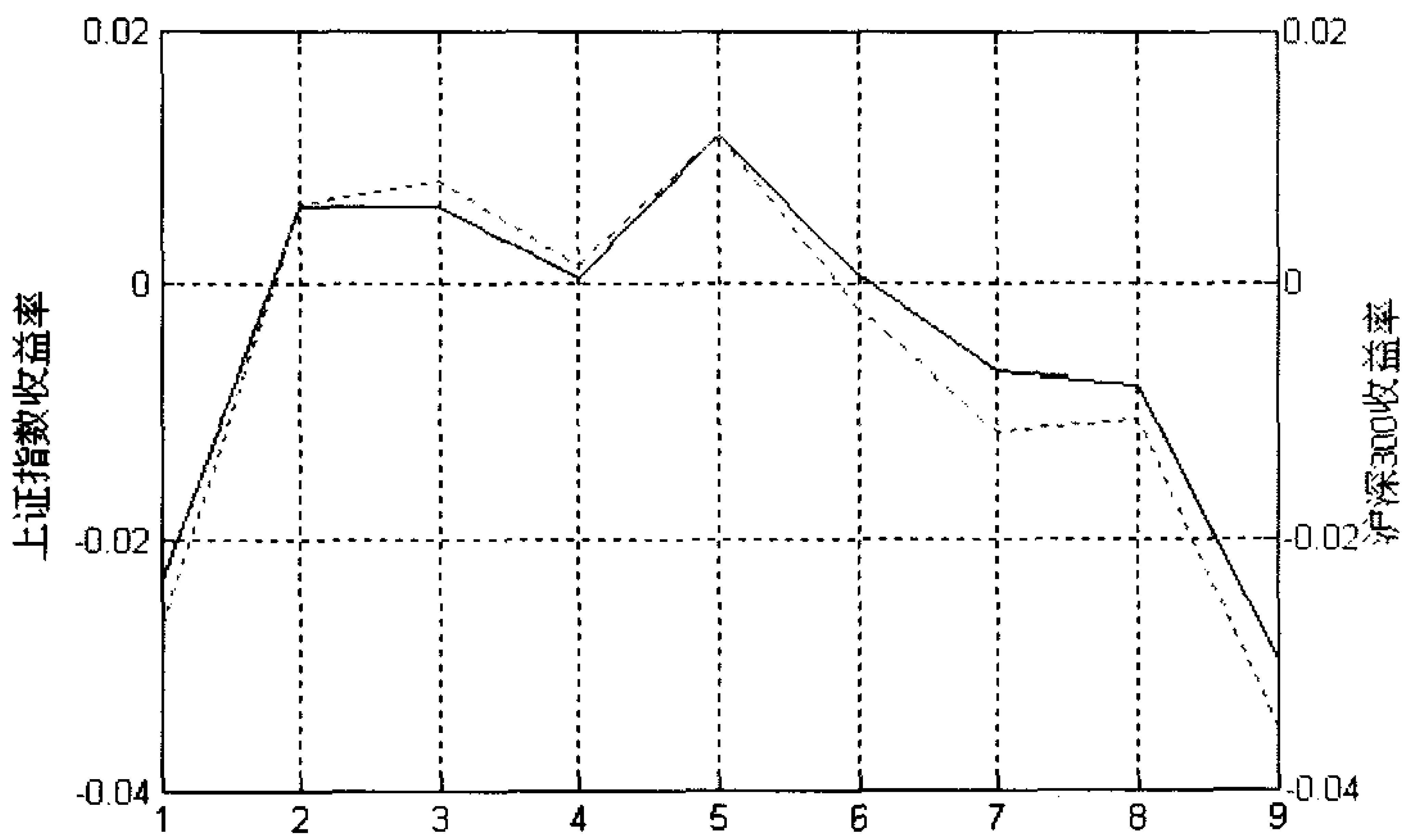


图 9.5　上证指数与沪深 300 指数收益率双坐标图

3. 折线图

【例 9-4】根据浦发银行股价数据，作出股价对数收益率折线图，代码如下：

```
>> stairs(ret,'b');
>> xlabel('天  数');ylabel('频  率');
>> title('浦发银行(600000)对数收益率折线图');
```

图 9.6 是浦发银行对数收益率折线图。

4. 茎叶图

【例 9-5】根据浦发银行股价数据，作对数收益率茎叶图，代码如下：

```
>> stem(ret)
>> xlabel('天  数');ylabel('频  率');
>> title('浦发银行(600000)对数收益率茎叶图');
```

浦发银行对数收益率茎叶图如图 9.7 所示。

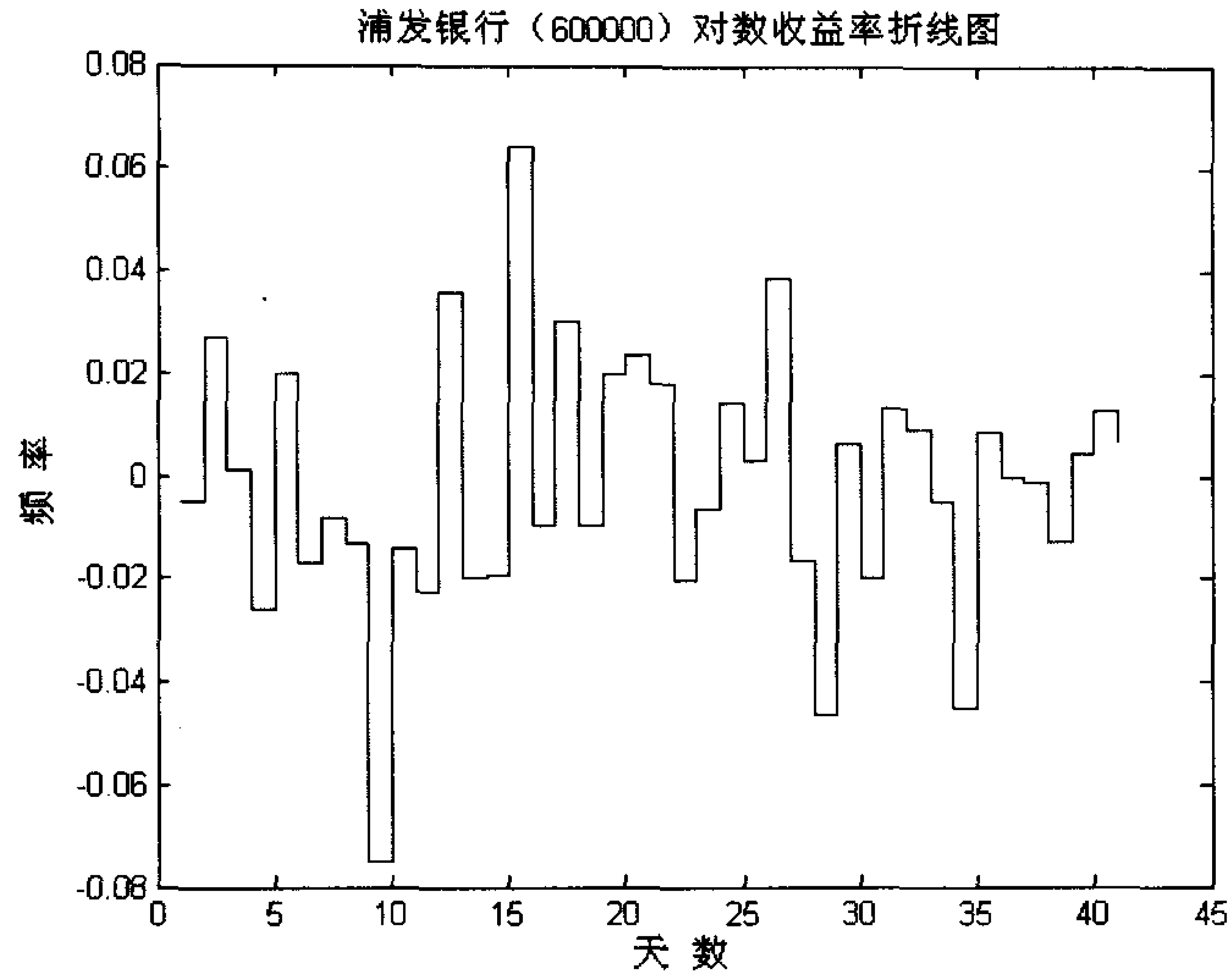


图 9.6　浦发银行对数收益率折线图

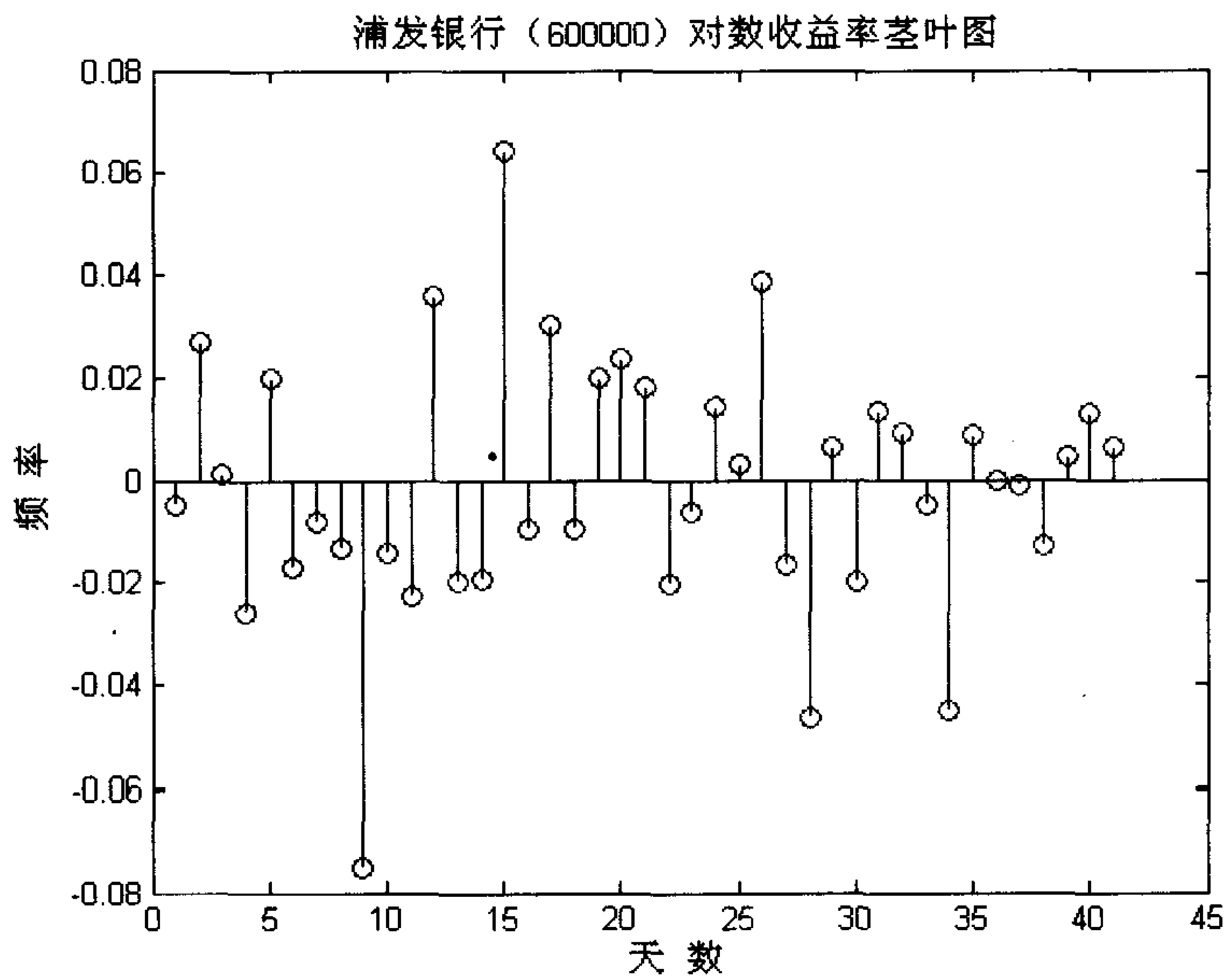


图 9.7　浦发银行对数收益率茎叶图

5. 误差棒形图

误差棒为数据置信水平或者为沿着曲线的偏差。一般沿着一曲线画误差棒形图，调用函数为 errorbar。

调用方式 1

```
errorbar(Y,E)
```

输入参数

Y　　　　　%画出向量 Y，同时显示在向量 Y 每一元素之上的误差棒，误差棒为 E(i)在曲线 Y 上面与下面的距离，所以误差棒长度为 2*E(i)

调用方式 2

```
errorbar(X,Y,E)
```

输入参数

X,Y,E　　　　　%必须为同型参量

若同为向量，则画出带长度为 2*E(i)、对称误差棒于曲线点(X(i),Y(i))之处；若同为矩阵，则画出的带长度为 E(i,j)、对称误差棒于曲面点(X(i,j),Y(i,j))之处。

调用方式 3

```
errorbar(X,Y,L,U)
```

输入参数

X,Y,L,U　　　　　%函数中 X，Y，L，U 必须为同型参量

若同为向量，则在点(X(i),Y(i))处画出向下长为 L(i)，向上长为 U(i)的误差棒；若同为矩阵，则在点(X(i,j),Y(i,j))处画出向下长为 L(i,j)，向上长为 U(i,j)的误差棒。

errorbar(…,LineSpec) 用 LineSpec 指定的线型、标记符、颜色等画出误差棒；h= errorbar(…) 返回线图形对象的句柄向量给 h，例如：

```
>> X = 0:pi/10:pi;
>> Y = exp(X).*sin(X);
>> E = std(Y)*ones(size(X));
>> errorbar(X,Y,E)
```

结果如图 9.8 所示。

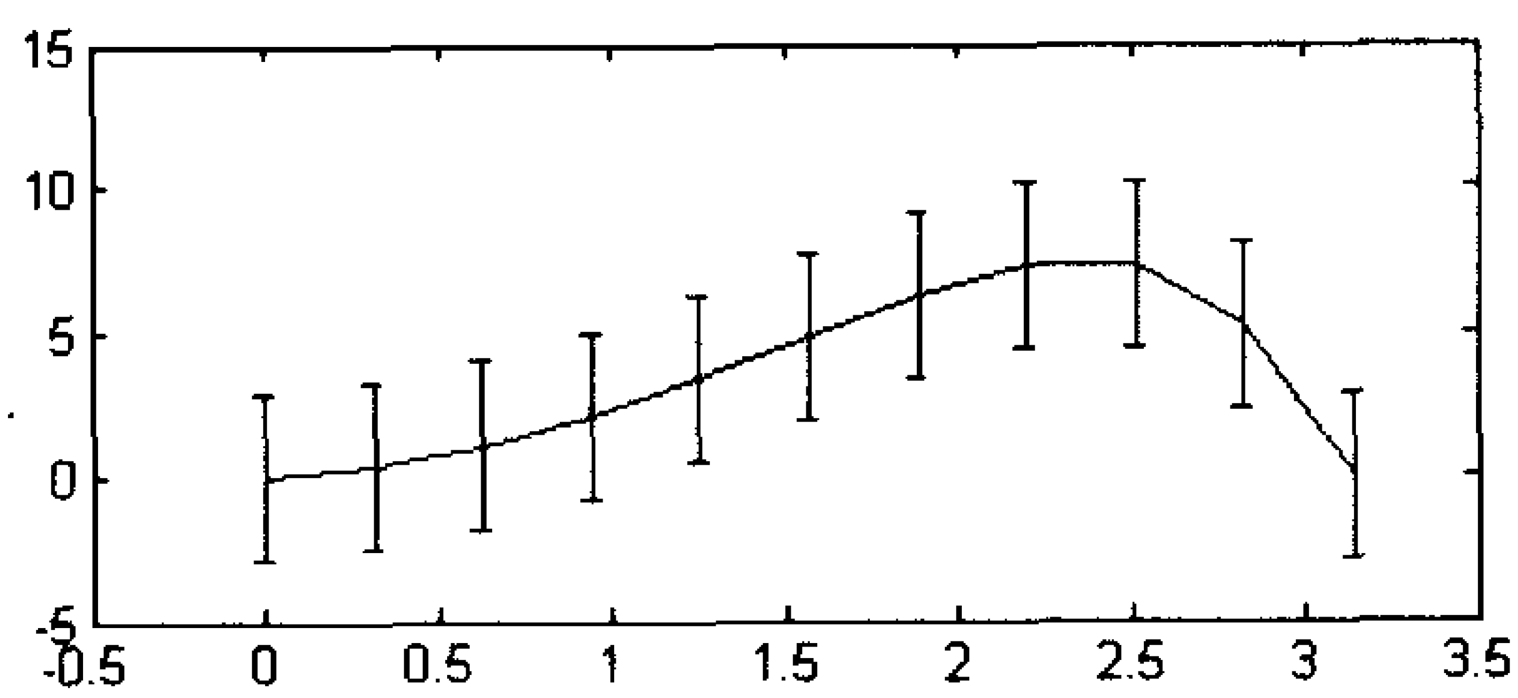


图 9.8　MATLAB 画出的误差图

6. 多图叠加

为了达到预期效果，有时需要将多种图像叠加到一幅图中。下面把浦发银行对数收益率折线图与茎叶图叠加在一起，图形叠加主要靠 hold on 命令完成的。

在 MATLAB 中执行如下命令：

```
>> stairs(ret,'b');
>> xlabel('天  数');ylabel('频  率');
>> title('浦发银行(600000)对数收益率折线图');
>> hold on %保持前一幅图不变
>> stem(ret)
```

结果如图 9.9 所示。

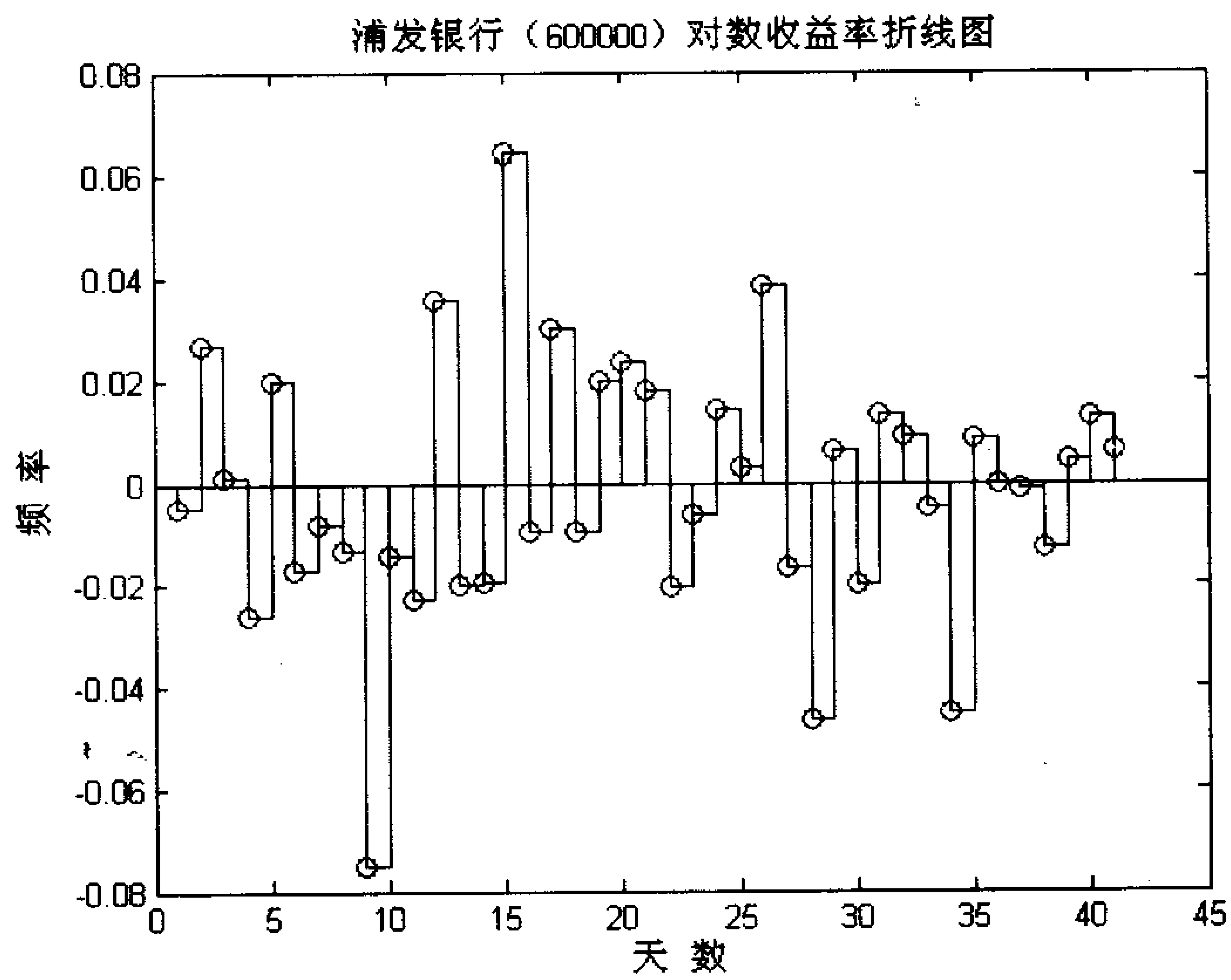


图 9.9 浦发银行对数收益率复合图

7. 并列作图

有时为了比较图与图之间的区别，需要把多个图并列起来做对比，可以调用函数 subplot(m,n,k)，把多幅图排成 m 行 n 列子图，k 表示其中从左到右编号的第 k 幅子图。下面我们生成两组随机数，分别保存在数组 a, b 中，然后将这两组数分别作图，代码如下：

```
>> rand('seed',0); %将随机数种子初值设为 0
>> a=rand(10,1);b=rand(10,1) ; % 生成两组随机数，分别保存在数组 a, b 中
>> subplot(1,2,1),plot(a),title('子图 1' ); %将 2 幅子图排成 1 行 2 列，第一幅是数
组 a 折线图
```

```
>>subplot(1,2,2),stairs(b); title('子图2' ); % 第二幅是数组b折线图。
```

结果如图 9.10 所示。

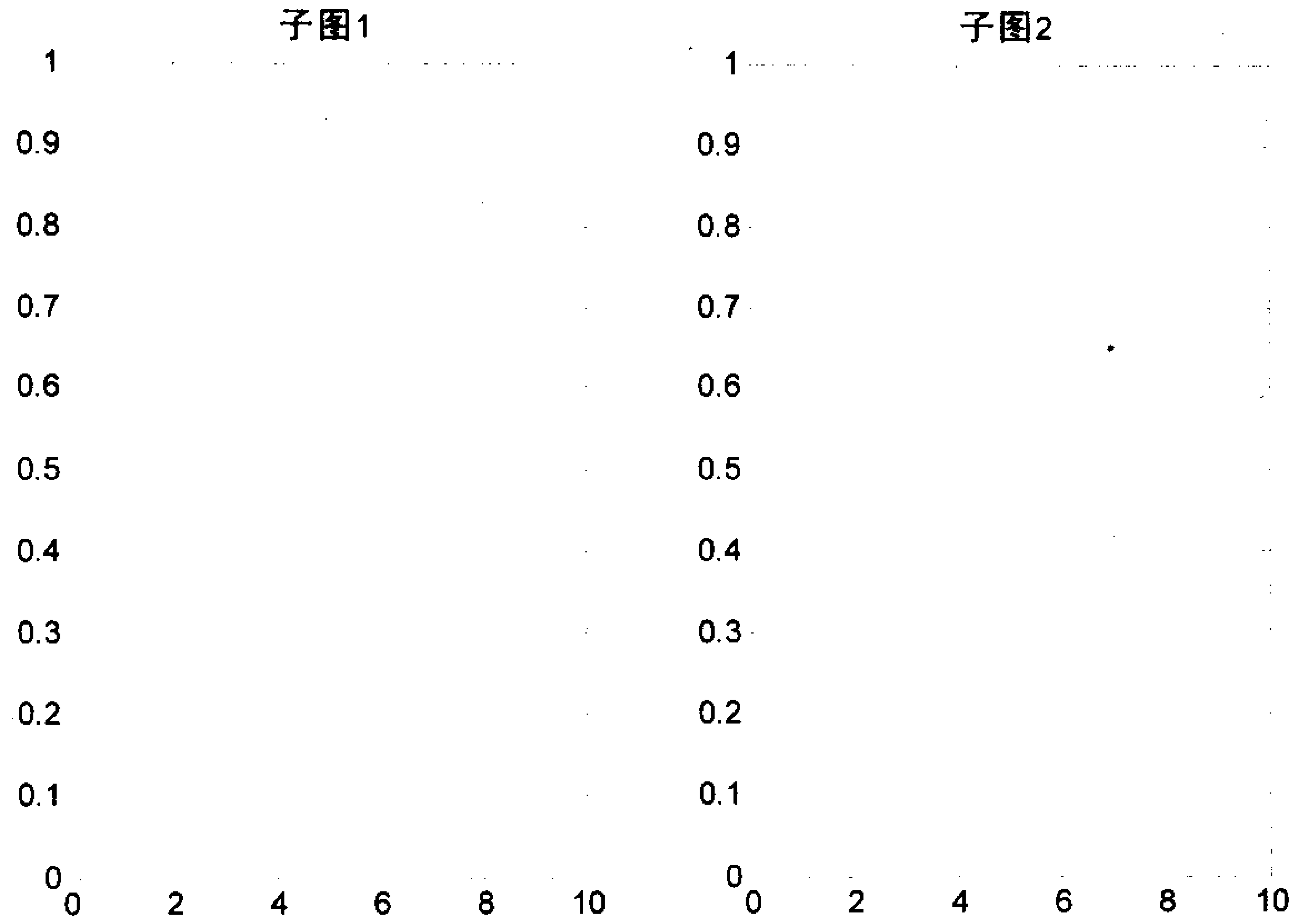


图 9.10 并列子图

9.1.3 修改金融时间序列作图

金融数据一般为时间序列数据，在绘图时通常要把 X 轴坐标上的刻度由整数改为日期，下面用 3 种方法改变坐标刻度。

【例 9-6】根据 G 上港(600018)2006 年 4 月 7 日至 5 月 22 日收盘价做出股价图，然后更改坐标。

首先将 Excel 中的时间、股价数据传给 MATLAB 中的 a1 变量，注意日期数据变成了整数。Excel 中的起始时间原点与 MATLAB 中的起始时间原点相差 693 960 天，需要在 MATLAB 中加上该整数，转化为 MATLAB 中的日期。

在 MATLAB 中执行如下命令：

```
>> aa=fints(a1(:,1)+693960,a1(:,2))  % 将数据转化为MATLAB中的时间序列格式
aa =
    desc:  (none)
    freq:  Unknown (0)
    'dates:  (27)'    'series1:  (27)'
    '07-Apr-2006'     [       10.2400]
    '10-Apr-2006'     [       10.4600]
    '11-Apr-2006'     [       10.7500]
```

```
    '12-Apr-2006'      [        10.5000]
    '13-Apr-2006'      [        10.2400]
    '14-Apr-2006'      [        10.4600]
    '17-Apr-2006'      [        10.5600]
    '18-Apr-2006'      [        10.4900]
    '19-Apr-2006'      [        10.5400]
    '20-Apr-2006'      [        10.3400]
    '21-Apr-2006'      [        10.2500]
    '24-Apr-2006'      [        10.1100]
    '25-Apr-2006'      [        10.0400]
    '26-Apr-2006'      [        10.6000]
    '27-Apr-2006'      [        10.6700]
    '28-Apr-2006'      [        10.6600]
    '08-May-2006'      [        10.9800]
    '09-May-2006'      [        11.3100]
    '10-May-2006'      [        12.1300]
    '11-May-2006'      [        12.1500]
    '12-May-2006'      [        13.4000]
    '15-May-2006'      [        13.6100]
    '16-May-2006'      [        12.8500]
    '17-May-2006'      [        13.2000]
    '18-May-2006'      [        13.0300]
    '19-May-2006'      [        13.3000]
    '22-May-2006'      [        13.9700]
>> plot(aa) ; % 调用 plot 函数绘图
>> title('G 上港 2006 年 4 月 7 日—5 月 22 日股价图')
>> ylabel('股价')
>> xlabel('时间')
```

图 9.11 为 G 上港股价图。

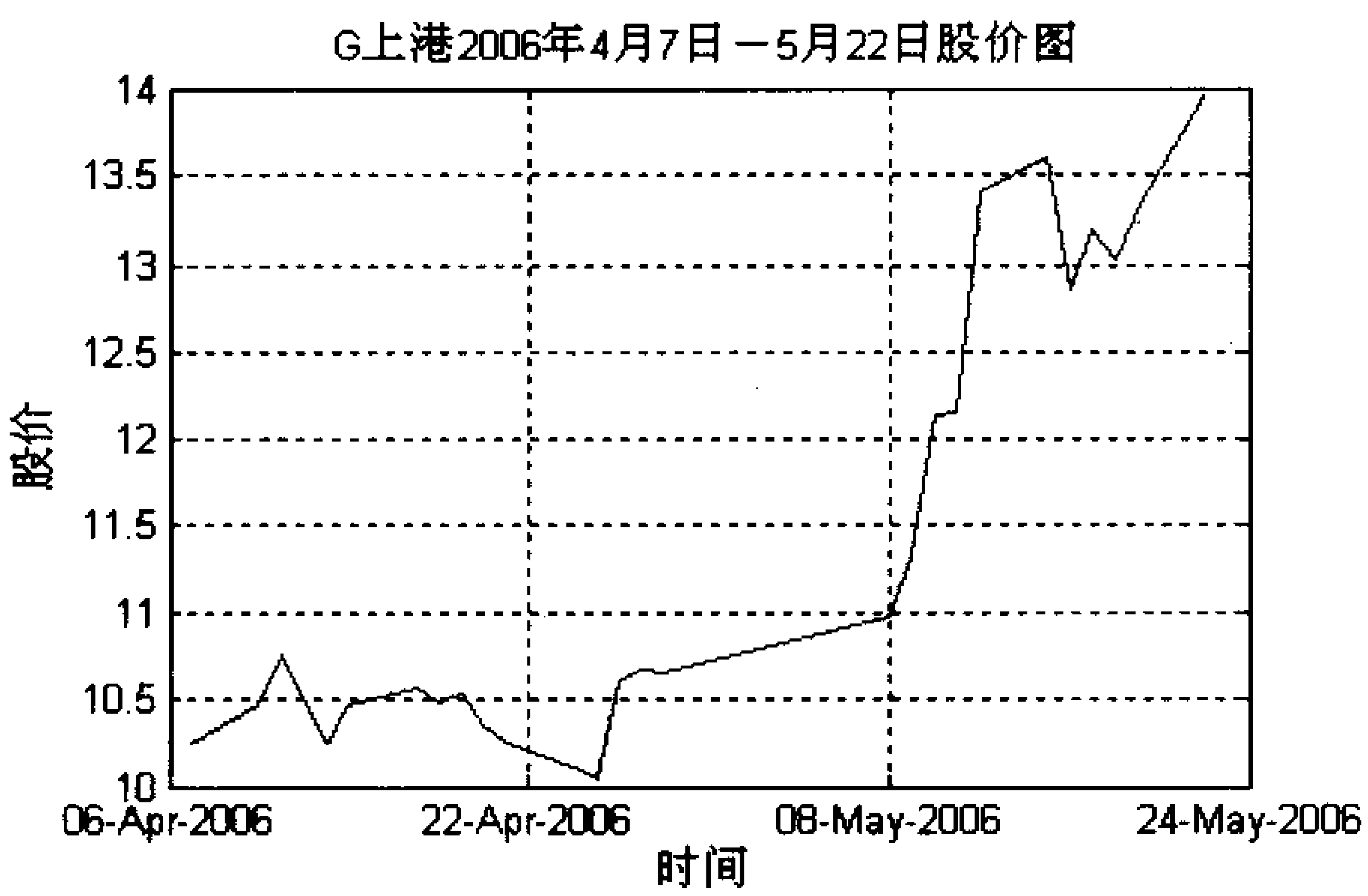


图 9.11　G 上港股价图

下面用两种方法更改坐标刻度名称。

方法 1 手工修改坐标刻度

打开 Edit 菜单，选择 Axes Properties 选项，单击 Ticks 按钮，弹出 Edit Axes Ticks 对话框，如图 9.12 所示，在该对话框的 Labels 列表框中修改坐标刻度。

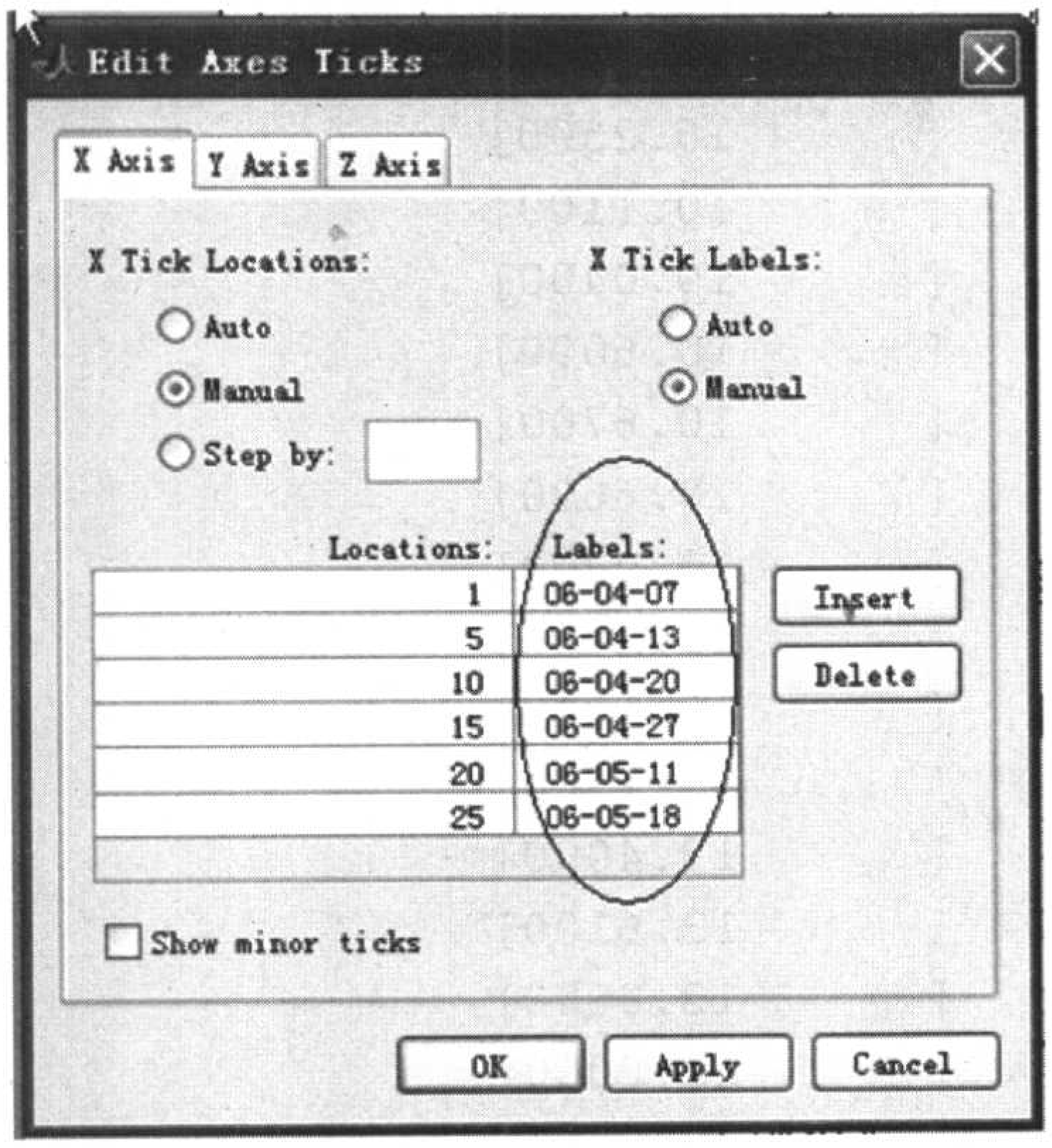


图 9.12 手工修改坐标

关闭对话框就可以看到修改后的股价图，如图 9.13 所示。

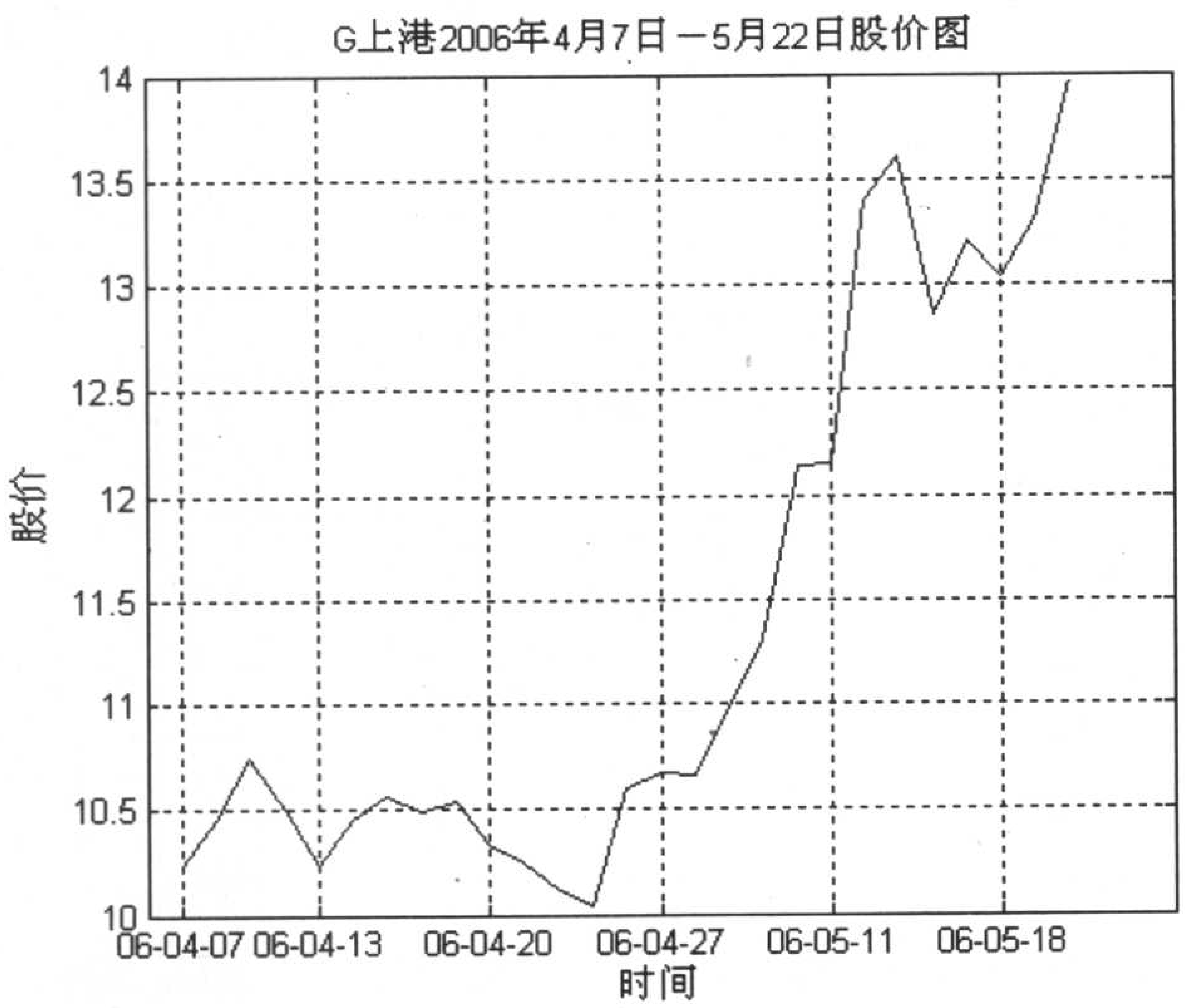


图 9.13 修改后的股价图

从中可以看出，坐标轴刻度已变为日期刻度。

方法 2 利用 set 函数更改图像坐标刻度

在 MATLAB 中执行如下命令：

```
>> plot(a1(:,2))
>>set(gca,'XTick',[1  5 10 15 20 25 30]);%确定刻度位置
>> set(gca,'XTickLabel',{'06-04-07';'06-04-13';'06-04-20';'06-04-27';…
'06-05-11';'06-05-18';'06-05-25'}) ;  % 确定坐标刻度名称
>> xlabel('时间'),%定义 X 轴名称
>> ylabel('股价'),%定义 Y 轴名称
>> title('G 上港 2006 年 4 月 7 日—5 月 22 日股价图'),%定义股价图名称
```

结果如图 9.14 所示。

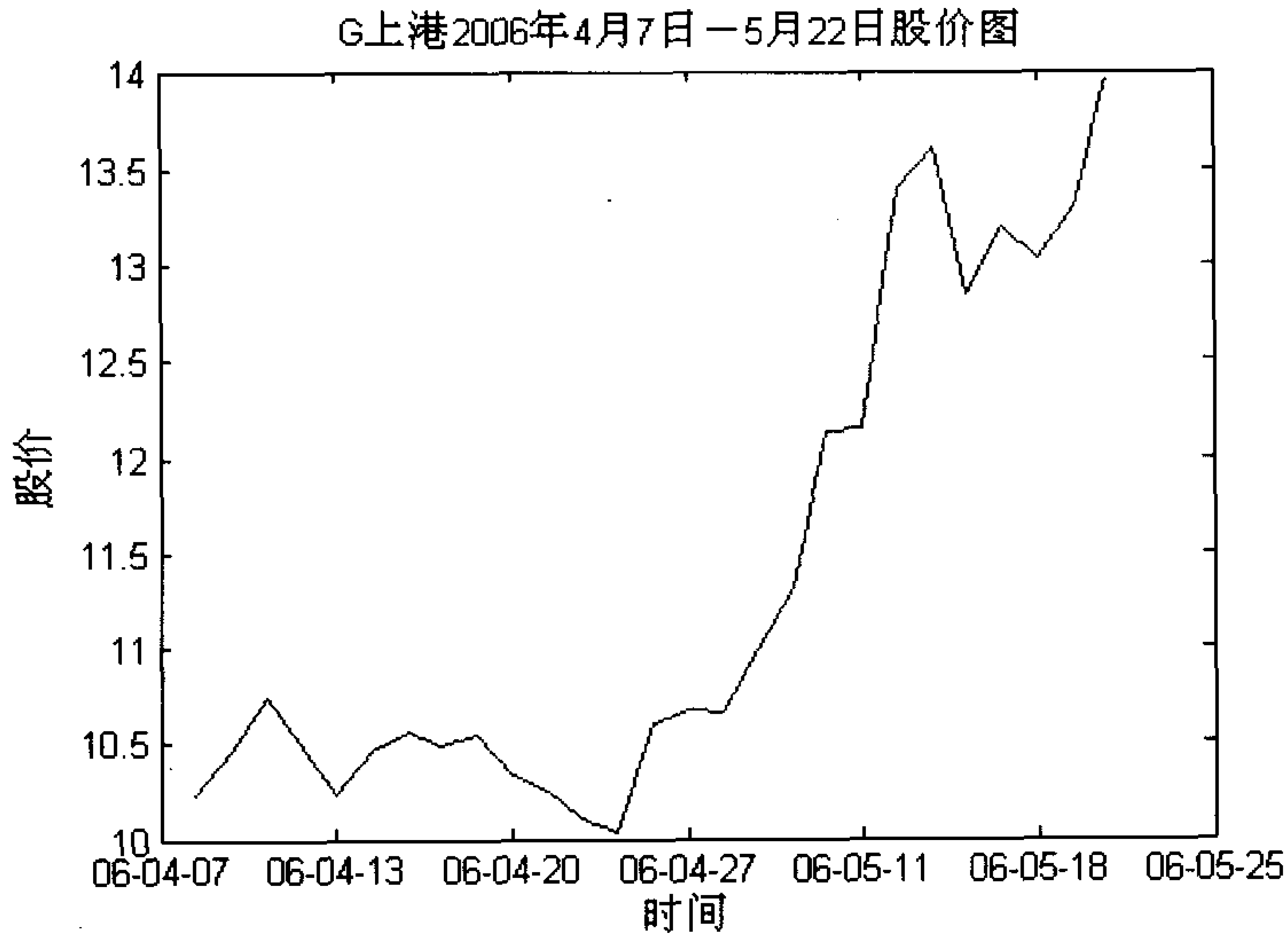


图 9.14 set 函数修改后的股价图

9.2 金融时间序列精确绘图

有时图像要求比较高，需要执行许多命令才能完成绘图任务，这就需要编写脚本文件来完成。下面的例子同样用 G 上港股票的数据画出股价图，要求给出标题、坐标轴的名称，X 轴的刻度为日期，背景添加网格线。

```
>>% plot times.m
>>load qq
>>plot(a1(:,2),'r*','markersize',20);
```

```
>>hold on;
>>plot(al(:,2))
>>axis([1 27 10,14])
>>xlabel('时间');
>>ylabel('股价');
>>title('G上港2006年4月7日—5月22日股价图','Color','r')
>>set(gca,'XTick',[1,5,10,15,20,25,30]);% 标出X轴的刻度位置
>>set(gca,'XTickLabel',{'06-04-07';'06-04-13';'06-04-20';'06-04-27';...
>>'06-05-11';'06-05-18';'06-05-25'});% 更改图像X轴的坐标刻度
>>grid on;  % 为图像背景添加网格线，grid off可以消除图像中的网格线
```

下面对程序中的命令进行讲解。

注释 1：程序第 3 句，plot 函数可以对线条的类型和大小进行定义，其注释如图 9.15 所示。

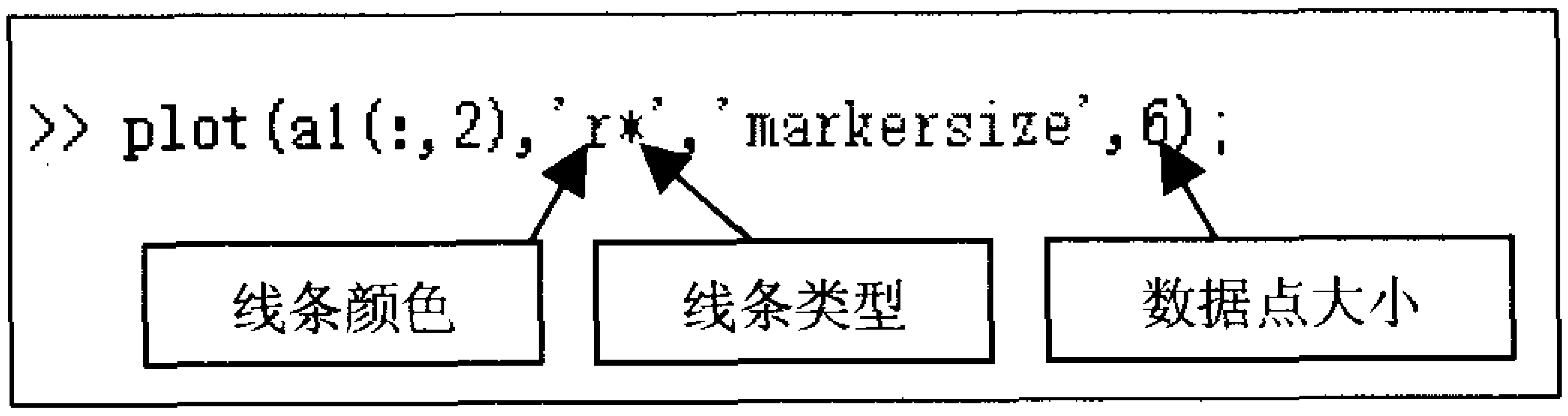


图 9.15　plot 函数内容

表 9.5、表 9.6 分别为 plot 函数定义线条特征的具体内容。

表 9.5　线条的形状与颜色

线条形状	符号	-		:		-.		--	
	含义	实线		虚线		点划线		双划线	
线条颜色	符号	R	G	b	k	y	w	m	c
	含义	红色	绿色	蓝线	黑色	黄色	白色	品红	青

表 9.6　线条类型

符　号	含　义	符　号	含　义
.	实心黑点	d	菱形符号
+	十字符	h	六角形符号
*	八线符	o	空心圆点
^	朝上三角形	p	五角星符
<	朝左三角形	s	方块符
>	朝右三角形	x	叉字符
V	朝下三角形		

注释 2：程序第 7 句、第 8 句中 xlabel、ylabel 函数的功能是给 X、Y 轴贴上标签。

xlabel('string')的功能是给当前轴对象中的 X 轴贴标签。注意：若再次执行 xlabel 命令则新标签会覆盖旧标签。

xlabel(fname)是先执行函数 fname，其返回一个字符串，然后在 X 轴旁边显示出来。

xlabel(…,'PropertyName',PropertyValue,…)指定轴对象中要控制的属性名和要改变的属性值。

h = xlabel(…)、h = ylabel(…)返回作为标签 text 对象句柄。

注释 3：程序第 10 句，该行语句作用主要是在 X 轴上标出刻度位置，如图 9.16 所示。

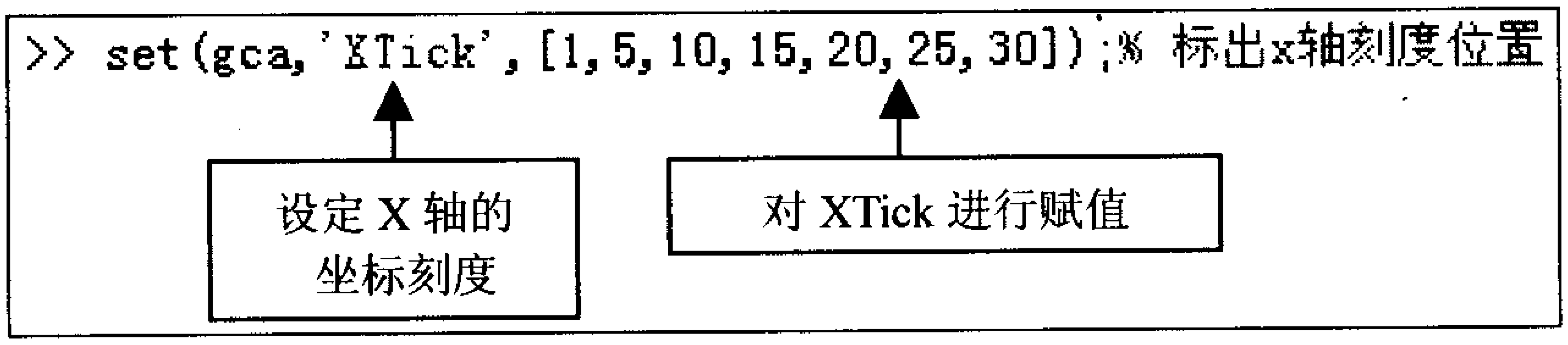


图 9.16　set 函数内容

注释 4：程序第 11 句中的 set 函数。

该语句主要可以对 X 轴坐标刻度进行赋值，由于原来图例中 X 轴的刻度是整数，为了直观起见需要转化为日期，这样可以明确看到每个交易日的股票收盘价，set 函数可以对刻度的位置及其标签进行定义，具体如图 9.17 所示。

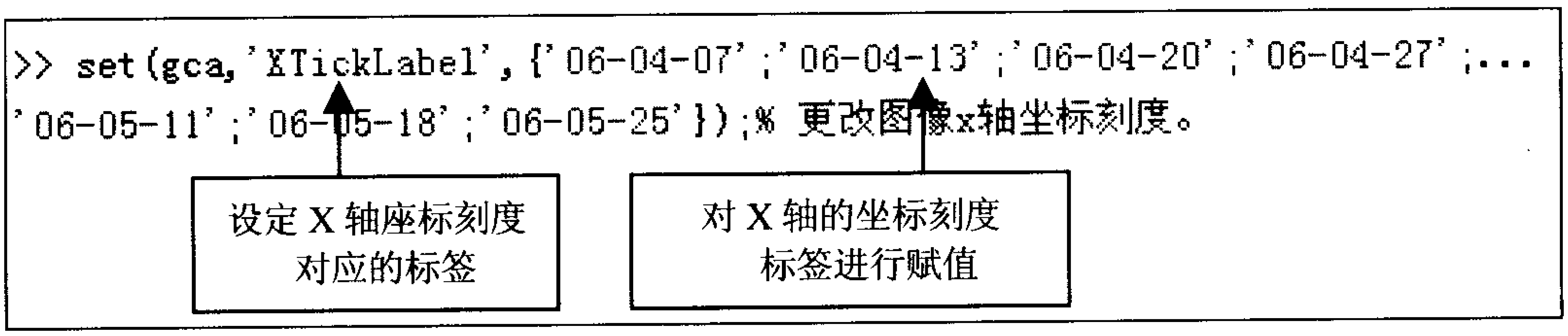


图 9.17　set 函数修改坐标刻度

注释 5：程序第 13 句中的 grid 函数，功能是给图像背景加网格线。

grid 函数的功能是给二维或三维图形坐标面增加分隔线，该命令会对当前坐标轴 Xgrid、Ygrid，Zgrid 的属性有影响。

```
grid on             %功能是给当前坐标轴增加分隔线
grid off            %从当前坐标轴中去掉分隔线
grid                %转换分隔线的显示与否的状态
grid(axes_handle,on|off)    %对指定坐标轴 axes_handle 是否显示分隔线
```

注释 6：程序第 6 句，axis 函数的功能是定义坐标轴刻度与外在显示，其内容如表 9.7 所示。

表 9.7 axis 函数内容

axis([xmin xmax ymin ymax])	以 xmin、xmax 设定横轴的下限及上限，以 ymin、ymax 设定纵轴的下限及上限
axis auto	横轴及纵轴依照数据大小的上下限来订定，横轴及纵轴比例是 4∶3
axis square	设置当前图形为正方形(或立方体形)，系统将调整 X 轴、Y 轴与 Z 轴，使它们有相同的长度，同时相应地自动调整数据单位之间的增加量
axis equal	横轴及纵轴比例是 1∶1
axis image	效果与命令 axis equal 相同，只是图形区域刚好紧紧包围图像数据
axis xy	使用笛卡儿坐标系(默认)：坐标原点在左下角、横坐标(X 轴)的值从左到右增加，纵坐标(Y 轴)的值从下到上增加
axis ij	使用矩阵坐标系：坐标原点在左上角、横坐标(J 轴)的值从左到右增加，纵坐标(I 轴)的值从上到下增加
axis normal	以预设值画纵轴及横轴，自动调整坐标轴的纵横比，还有用于填充图形区域的、显示于坐标轴上的数据单位的纵横比
axis off	将纵轴及横轴取消
axis on	恢复纵轴及横轴
axis fill	用于将坐标轴的取值范围分别设置为绘图所用数据在相应方向上的最大、最小值
axis tight	把坐标轴的范围定为数据的范围，即坐标轴中没有多余的部分

表 9.8 显示了由上面命令设置的坐标轴属性。

表 9.8 axis 函数的属性

命令 / 坐标轴属性	axis equal	axis normal	axis square	axis tightequal
DataAspectRatioMode	[1 1 1]	没有设置	没有设置	[1 1 1]
PlotBoxAspectRatio	manual	auto	auto	Manual
PlotBoxAspectRatioMode	[3 4 4]	没有设置	[1 1 1]	Auto
Stretch-to-fill	禁止	可行	禁止	禁止

```
axis vis3d      %冻结坐标系此时的状态，以便进行旋转
axis off        %关闭所用坐标轴上的标记、栅格和单位标记，但保留由 text 和 gtext 设置
                 的对象
axis on         %显示坐标轴上的标记、单位和栅格
```

```
[mode,visibility,direction] = axis('state')    %返回当前坐标轴设置属性的三个字符串，每个字符串的含义如表 9.9 所示
```

表 9.9　坐标轴属性注解

输出参量	返回字符串	说　明
Mode	‘auto’或 ‘manual’	若 XLimMode、YlimMode 与 ZlimMode 都设置为 auto，则 mode 为 auto；若 XLimMode、YlimMode 与 ZlimMode 都设置为 manual，则 mode 为 manual
Visibility	‘on’或‘off’	
Direction	‘xy’或‘ij’	

程序运行结果如图 9.18 所示。

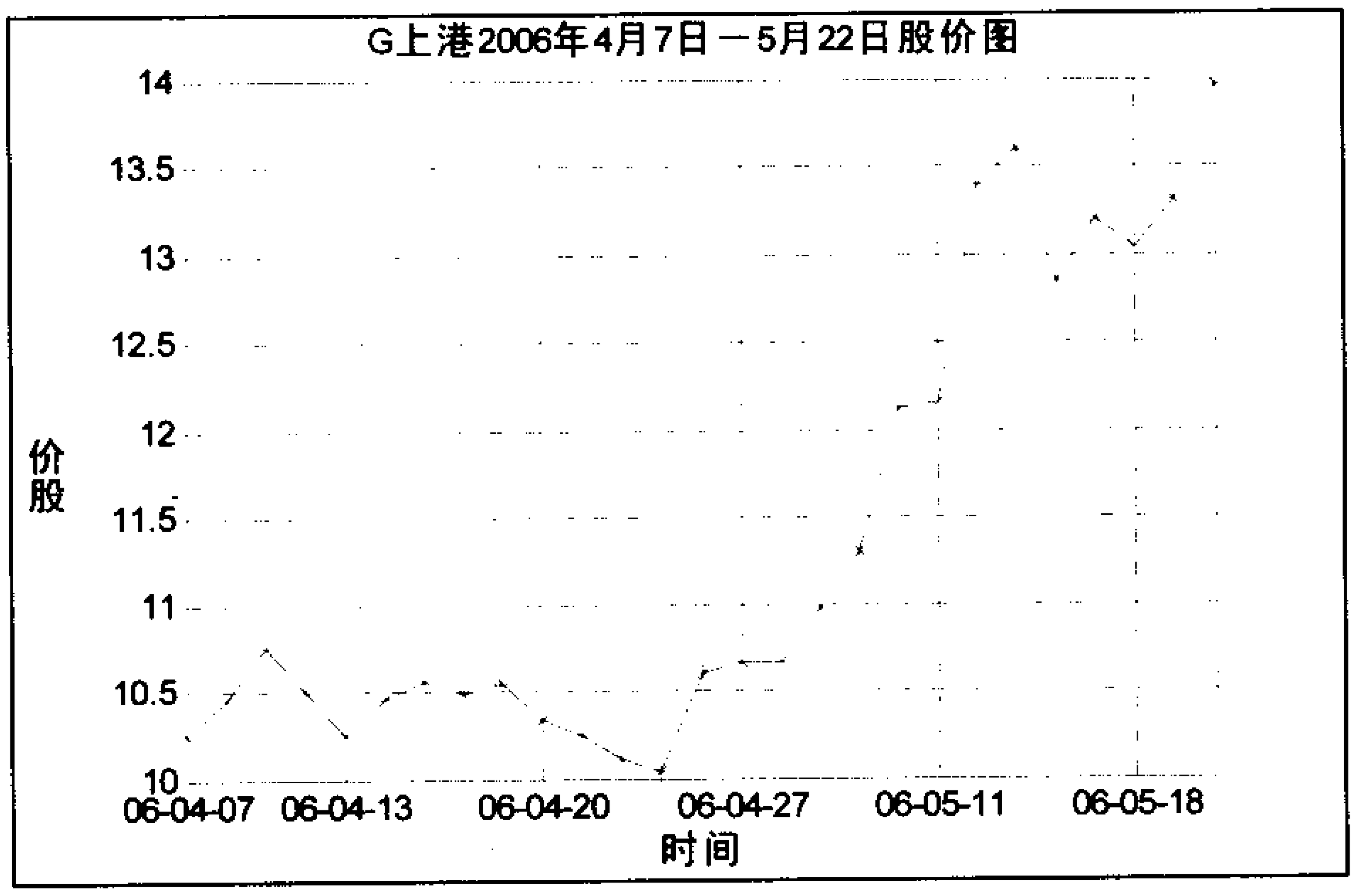


图 9.18　G 上港股价图

下面仍然用 G 上港的股价的数据绘图，要求在图像中找出股价最低点并且标记出来，取消坐标轴，图像框。

在 MATLAB 中执行如下命令：

```
>>load qq
>>plot(a1(:,2),'r*','markersize',5);
>>hold on;
>>plot(a1(:,2))
>>axis([1 27 10,14])
>>xlabel('时间');
>>ylabel('股价');
>>title('G上港2006年4月7日—5月22日股价图','Color','r')
>>set(gca,'XTick',[1,5,10,15,20,25,30]);    % 标出 X 轴的刻度位置
>>set(gca,'XTickLabel',{'06-04-07';'06-04-13';'06-04-20';'06-04-27';...
>>'06-05-11';'06-05-18';'06-05-25'});    % 更改图像 X 轴坐标刻度
```

```
>>grid on;                    % 为图像背景添加网格线
>>legend('G 上港股价') ;
>>str{1}='\fontname{隶书}最低价格';
>>min1=min(a1(:,2));
>>n1=find(a1(:,2)==min1)
>>text(n1,min1,str{1})      % 在图像中添加文本
>>box off
>>axis off
>>legend off
```

程序运行结果如图 9.19 所示。

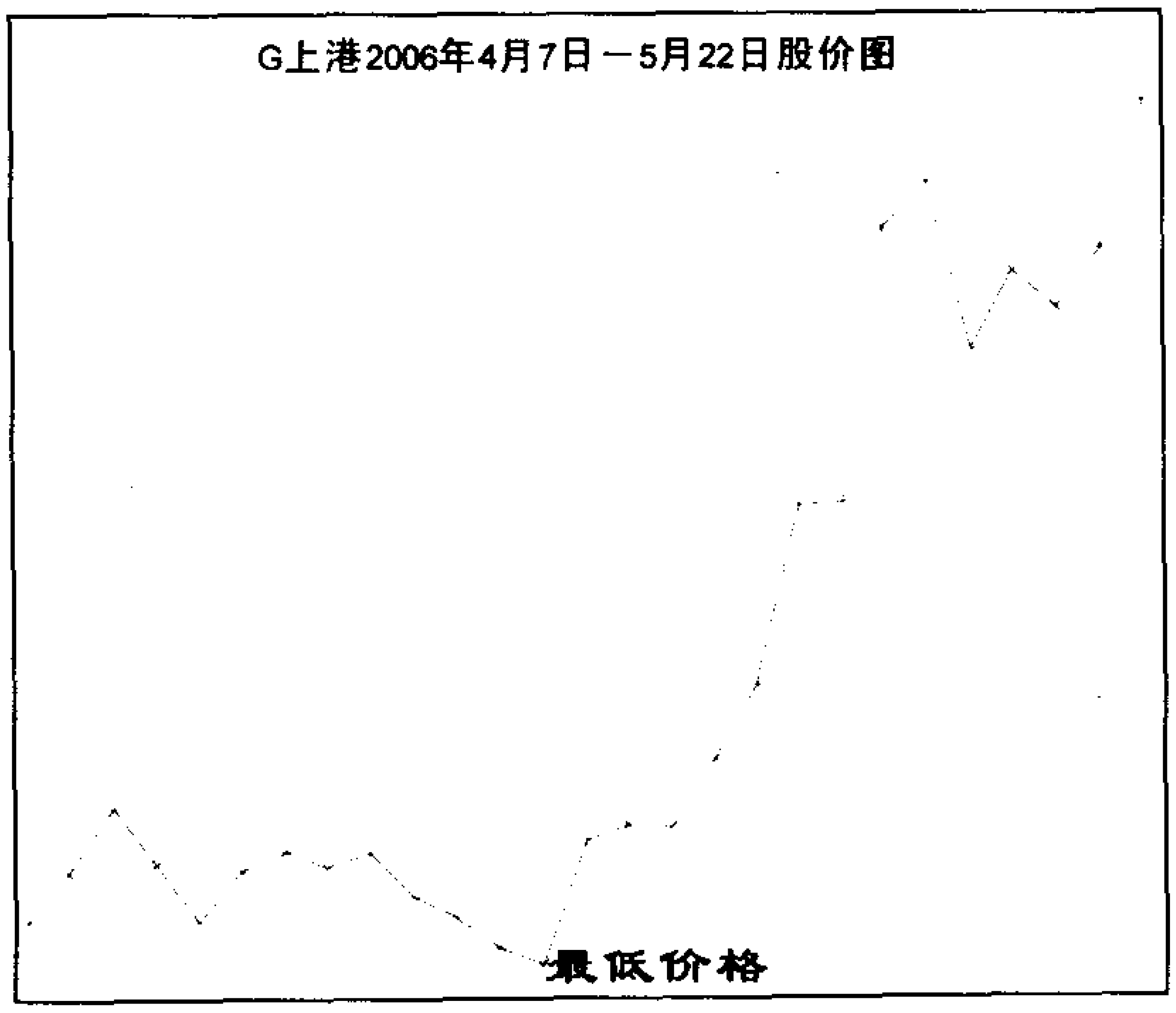


图 9.19　取消坐标轴的 G 上港股价图

思　考　题

1. 选择一家上市公司，绘出其收益图，要求标出标题、日期刻度，用醒目的符号标出最低点、最高点。

2. 绘出一家上市公司的收益率折线图。

第 10 章 MATLAB 和其他软件数据连接

数据连接是金融实务的基本内容，通过本章的学习，要求熟悉 MATLAB 与 Excel 之间的多种数据连接技术，学会从 Yahoo 网站获取国外债券市场与股票市场的实时数据与历史数据，学会 MATLAB 与 Access 等数据库数据的连接方式，了解在 Word 环境下启动 MATLAB 的方法。

MATLAB 提供了文件读写函数，这些函数是 MATLAB 语言的一部分，并不需要专门的工具箱支持。MATLAB 文件读写函数分为高级函数和低级函数，高级函数调用方法简单，使用方便，但是灵活性差，仅仅适用于一般格式文件类型；低级函数比较烦琐，但是灵活性好。

MATLAB 应用程序接口(Application Program Interface，API)所涵盖的内容非常广泛，可以和多种语言实现数据连接，如 C++、Excel、Word、Spss、Visual Basic，而且可以与 Java 等网络编程语言实现数据连接。鉴于市面上已经有许多书籍介绍 MATLAB 与 C 语言、Visual Basic、Java 软件的数据连接，这里就不再赘述，下面主要介绍与 Excel、Word 数据连接。

10.1 MATLAB 和 Excel 数据连接

Excel 和 MATLAB 在数据显示和数值计算上各有优势，Excel 是商业运用最广泛的工具，非常直观，但是数值编程比较差，而 MATLAB 可以弥补这一点，有时在程序开发上需要将两者结合起来，实现两者之间的优势互补，为此 MATLAB 提供了 Excel Link 和 MATLAB Builder For Excel 两个连接工具，实现 MATLAB、Excel、VBA 之间的混合编程。

Excel Link 是一个在 Windows 环境下实现 MATLAB 和 Excel 互相连接的插件，通过实现 Excel 和 MATLAB 的连接，用户可以在 MATLAB 工作区中使用 Excel 宏编写程序，也可以在 Excel 中使用 MATLAB 的数据处理和图形图像处理功能进行相关操作，同时保证两个工作环境数据连接和同步。使用 Excel Link 时不必离开 Excel 环境，直接在 Excel 工作区调用 MATLAB 变量即可。

10.1.1 MATLAB 和 Excel 接口安装

MATLAB 和 Excel 接口连接步骤如下：

(1) 打开 Excel 软件，打开【工具】菜单，选择【加载宏】选项，弹出【加载宏】对话框。

(2) 在【加载宏】对话框中单击【浏览】按钮，在 MATLAB\toolbox\exlink 文件夹下选中 Excllink 文件，单击【确定】按钮。

(3) 重新回到【加载宏】对话框，选中 Excel Link 2.2 for use with MATLAB 复选框，如图 10.1 所示。

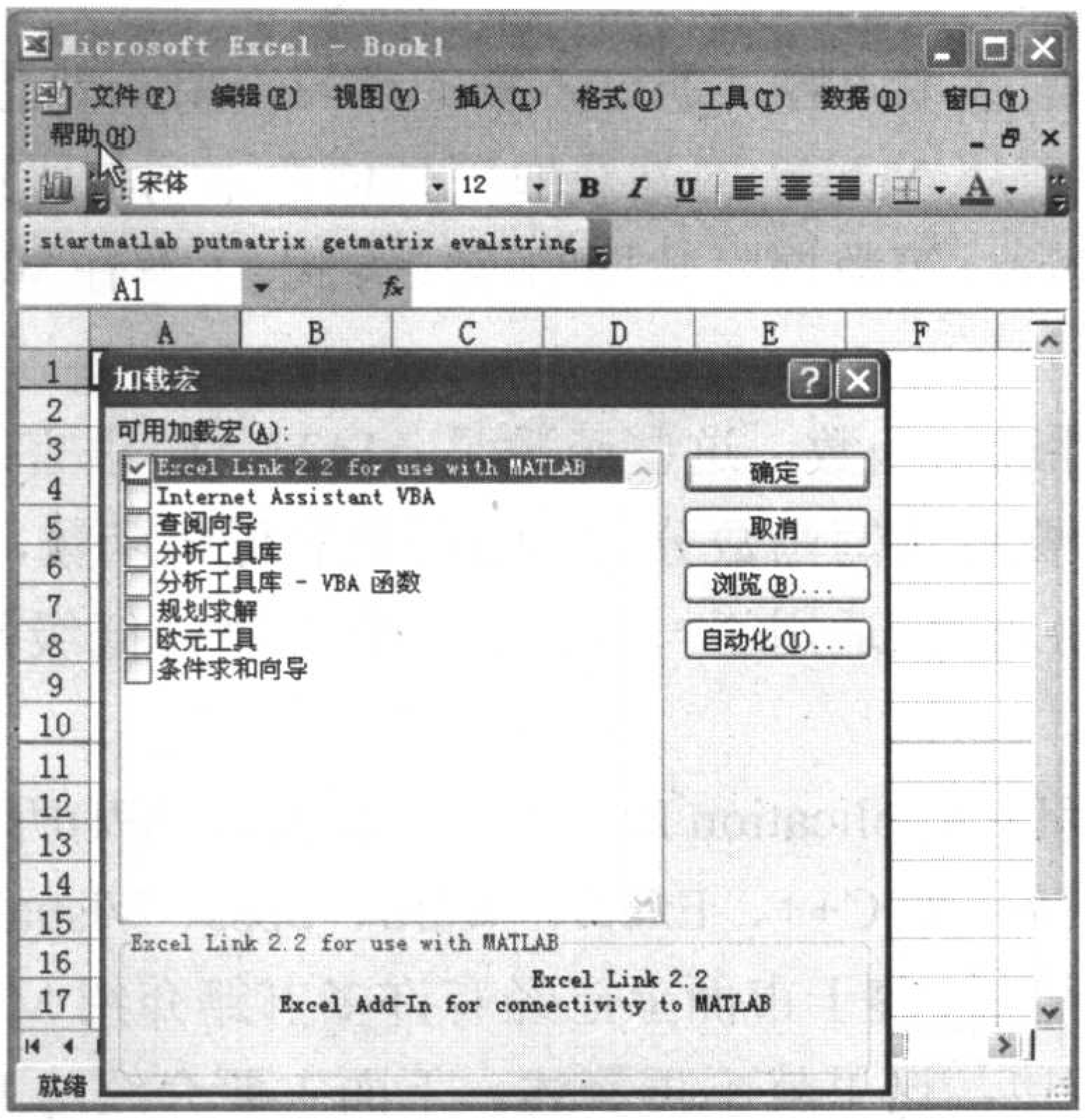


图 10.1 Excel【工具】菜单下加载宏

此时发现(MATLAB 命令窗口)已经加在 Excel 任务窗口中，单击 startmatlab 按钮就会弹出 MATLAB Command Window 窗口，如图 10.2 所示。

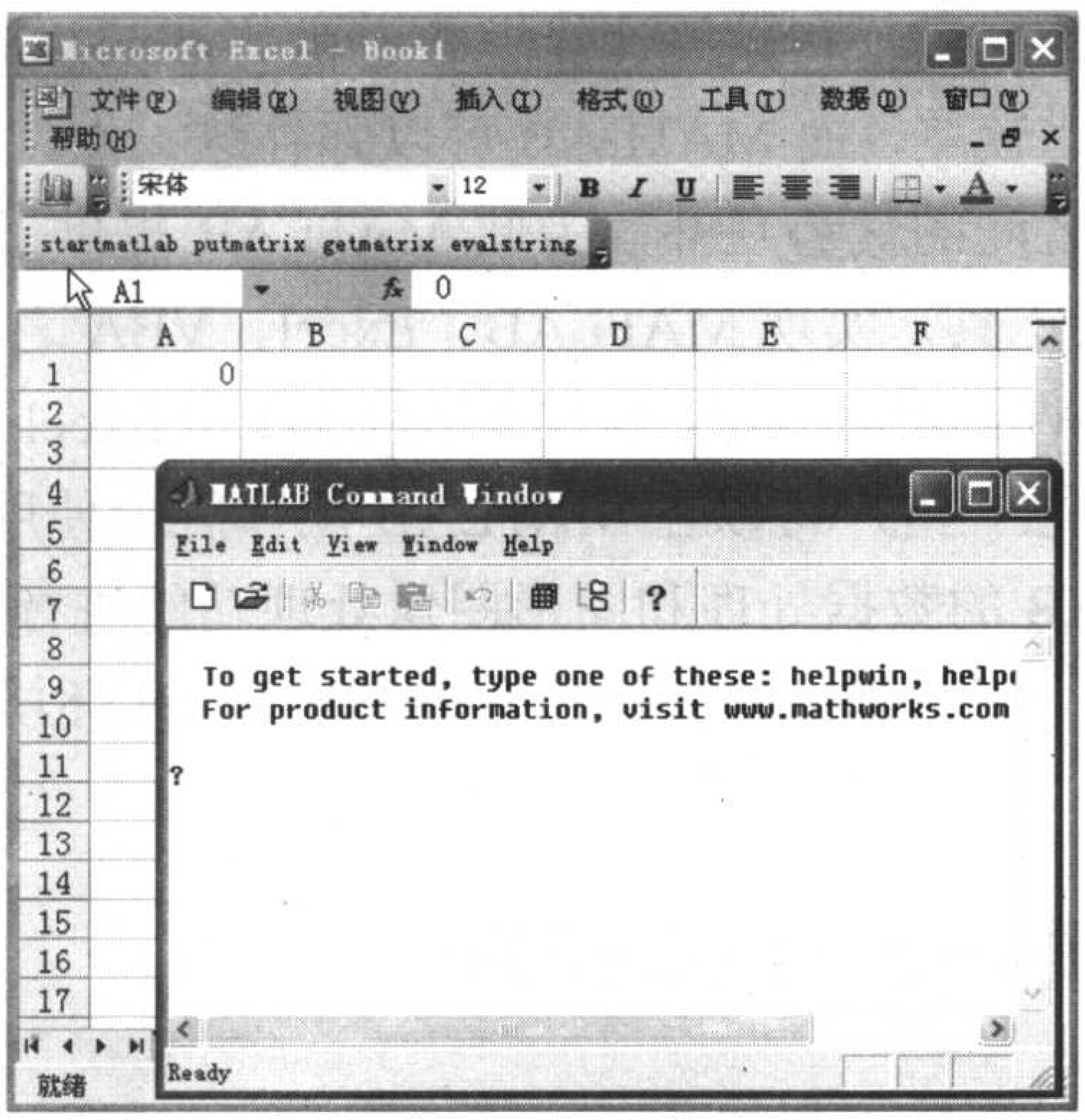


图 10.2 加载宏后的界面

MATLAB Command Window 窗口中的提示符变为“？”。图 10.3 所示是 MATLAB 命令窗口，窗口中的 4 个按钮 startmatlab、putmatrix、getmatrix、evalstring 用以实现 Excel 中的数据与 MATLAB 实时交换。

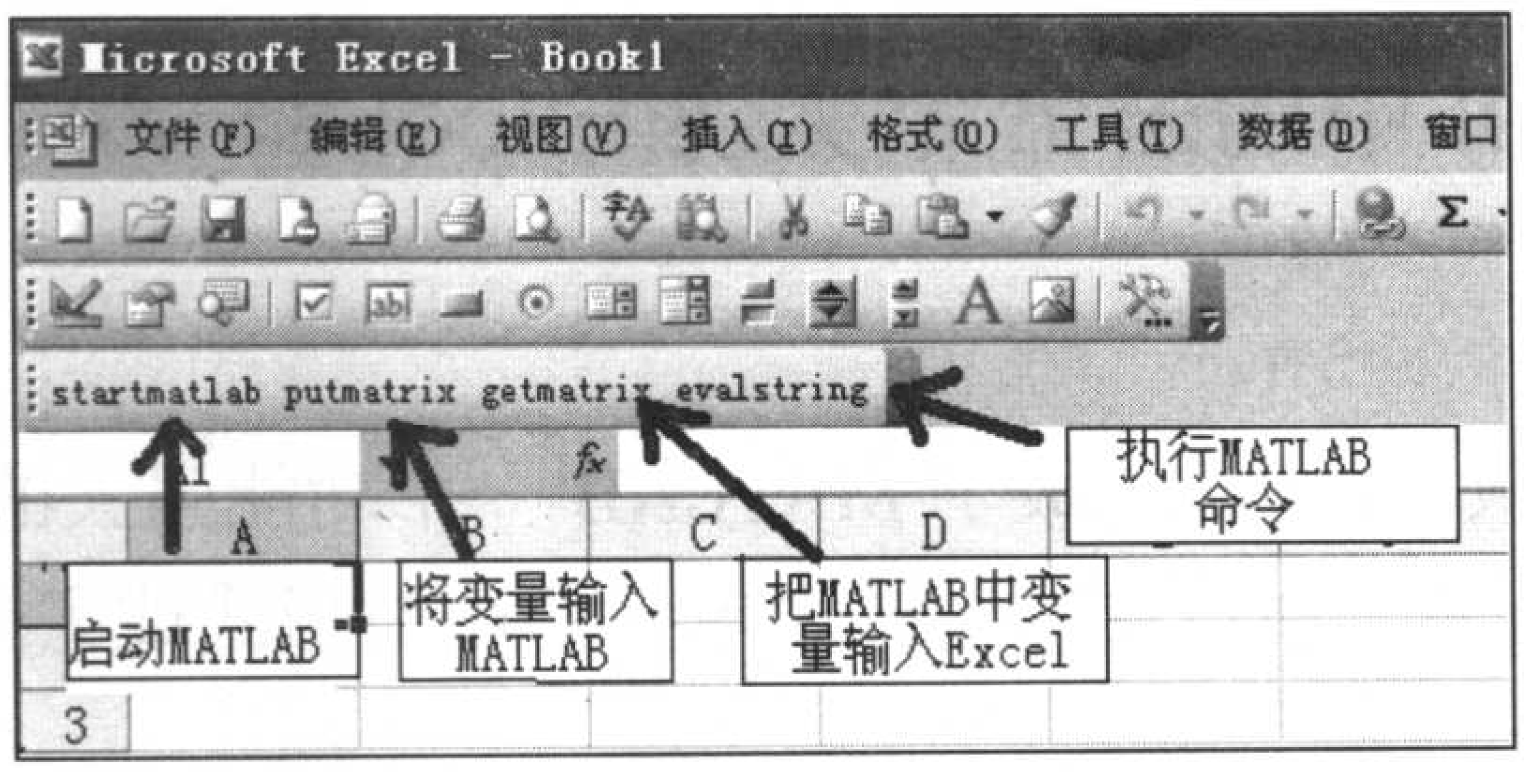


图 10.3　Excel 中的 MATLAB 命令菜单

10.1.2　MATLAB 自动启动和 Excel 连接

在桌面上的 MATLAB 7 小图标下单击鼠标右键，在弹出快捷菜单中选择【属性】命令，弹出【MATLAB 7.0 属性】对话框，在对话框的【目标】文本框的“C:\MATLAB7\bin\win32\MATLAB.exe”后面加入“/automation”，具体如图 10.4 所示。

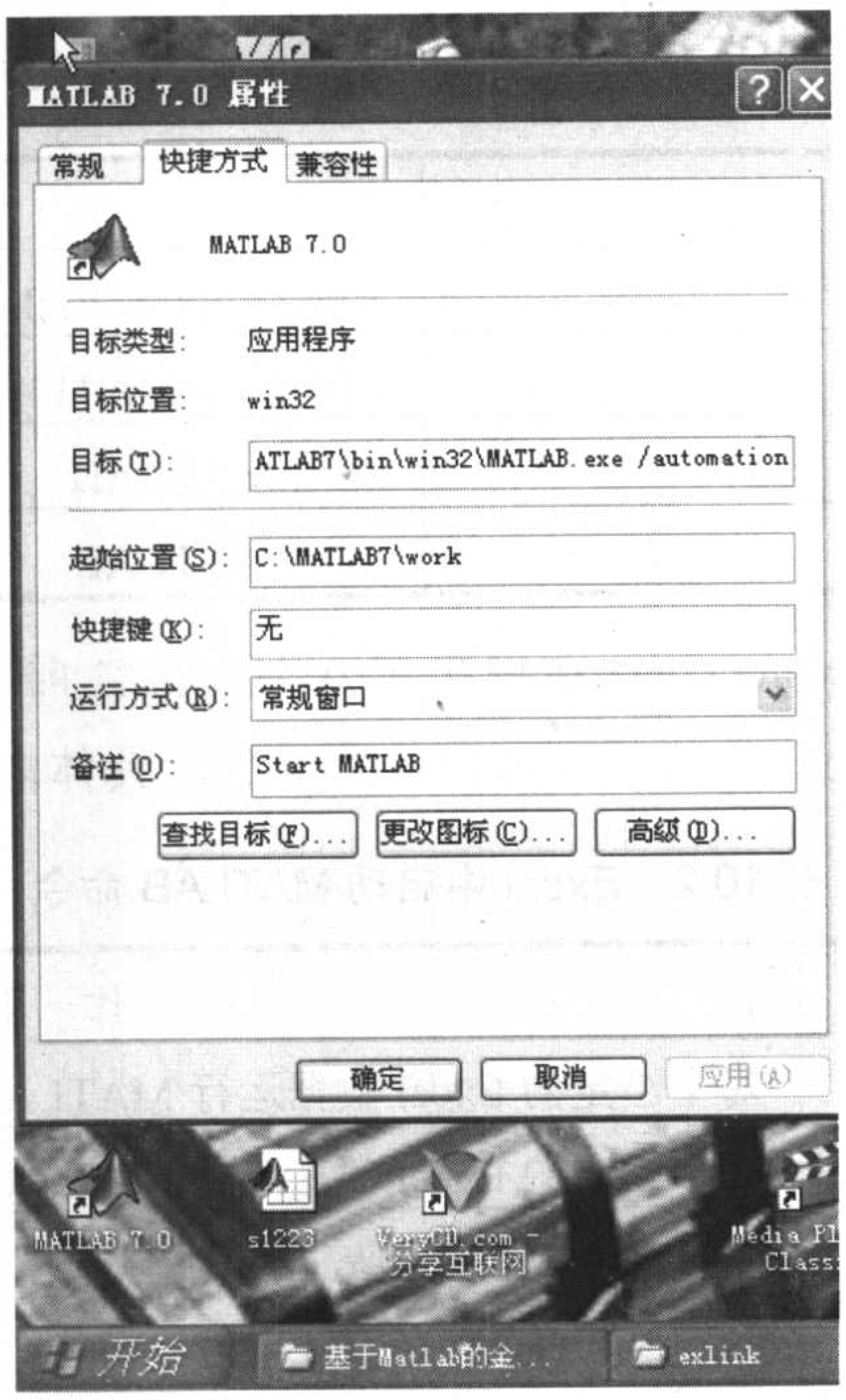


图 10.4　MATLAB 和 Excel 自动连接

这样单击 MATLAB 快捷键就自动出现与 Excel 连接的窗口，如果要关闭 MATLAB Command Window 窗口，同样选择 Excel 中的【工具】菜单，在【加载宏】对话框中取消选择 Excel Link 2.2 for use with MATLAB 复选框，就取消了 Excel 与 MATLAB 的连接。

10.1.3 利用 Excel 中的宏命令实现 Excel 和 MATLAB 数据连接

1. 连接管理函数

通过 Excel Link 函数，Excel 就成了 MATLAB 一个功能强大的数据存储和应用终端，使用户不必脱离 Excel 环境，而只要以数据表单元的函数形式或宏命令来使用 MATLAB 提供的相关处理函数。

Excel Link 提供了 4 个连接函数，分别是 MATLABinit、MLAutoStart、MLOpen 和 MLClose，其中，MATLABinit 只能以宏命令方式运行，其他命令均可以作为数据单元函数或者宏命令执行。

MLAutoStart 用于设置自启动，系统默认设置是在每一次启动 Excel 时自动启动 MATLAB，如果选择手动启动，就需要 MATLABinit 初始化 Excel Link 并且启动 MATLAB。MLClose 在保持 Excel 继续运行状态下用来终止 MATLAB 运行。MLOpen 或者 MATLABinit 在原来 Excel 进程中重新启动 MATLAB。具体内容如表 10.1 所示。

表 10.1 MATLAB 中启动 Excel 命令

函 数	作 用
MATLABinit	初始化 Excel Link，启动 MATLAB
MLAutoStart	自动启动 MATLAB
MLClose	终止 MATLAB
MLOpen	启动 MATLAB

除了上述 4 个连接函数，MATLAB 还提供了 9 个数据管理函数，实现 Excel 和 MATLAB 之间的数据连接，并可在 Excel 中执行 MATLAB 命令，具体内容如表 10.2 所示。

表 10.2 Excel 中启动 MATLAB 命令

函 数	作 用
MATLABfcn	对于给定的 Excel 数据运行 MATLAB 命令
MATLABsub	对于给定的 Excel 数据运行 MATLAB 命令，并指定输出位置
MLEDeleteMatrix	删除 MATLAB 矩阵
MLEvalString	执行 MATLAB 命令
MLGetMatrix	向 Excel 数据表写入 MATLAB 数据

续表

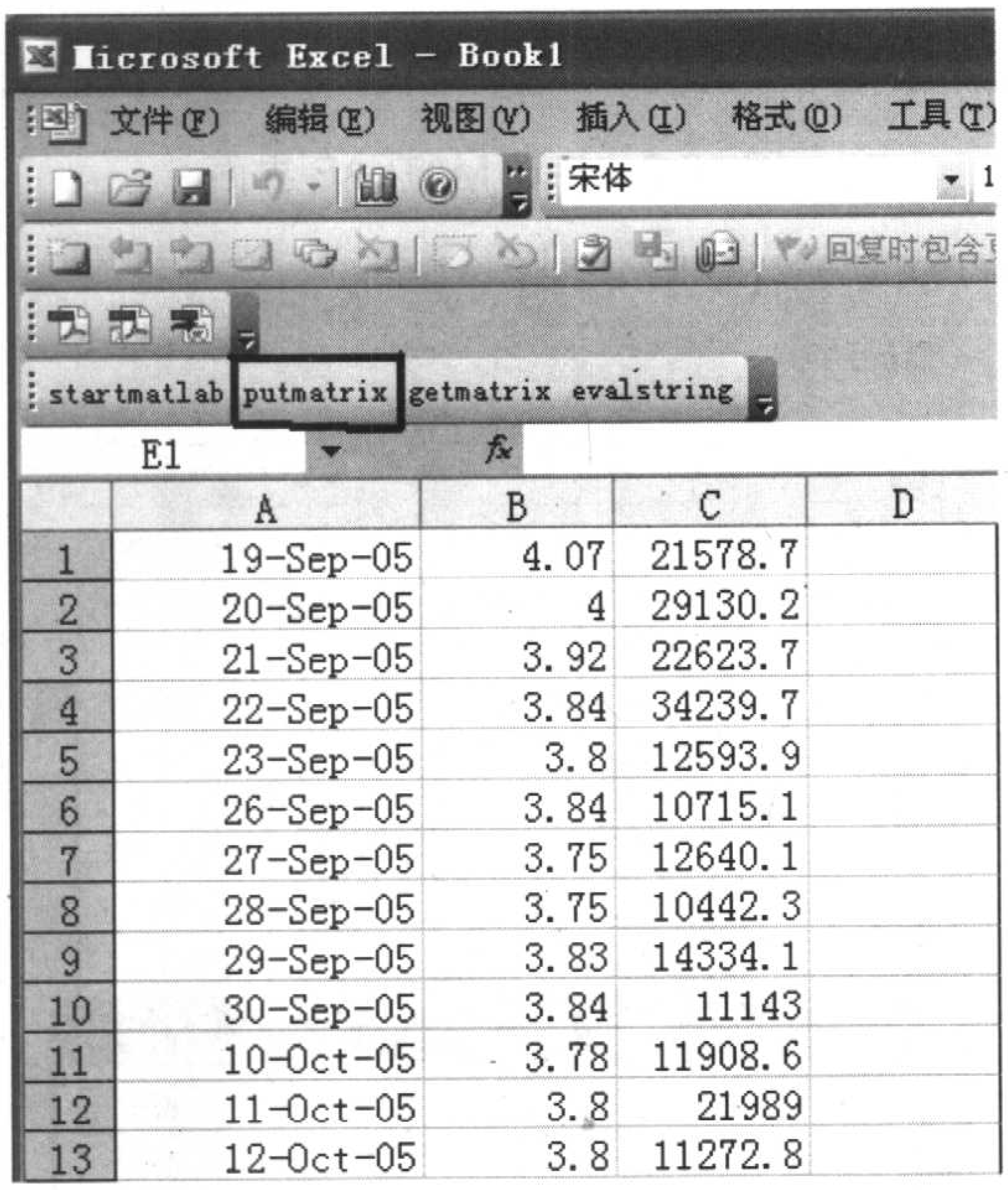
函　数	作　用
MLAppendMatrix	向 MATLAB 工作区添加 Excel 数据
MLPutMatrix	用 Excel 数据表创建或覆盖 MATLAB 数据
MLPutVar	用 Excel 数据表 VBA 创建或覆盖 MATLAB 数据
MLGetVar	向 Excel 数据表 VBA 写 MATLAB 数据内容

用户能以数据表单元函数形式或者宏命令形式调用除 MLGetVar 和 MLPutVar 以外的所有数据处理函数，MLGetVar 和 MLPutVar 只能以宏命令形式被调用。

2. 利用 Excel 直接把数据输入 MATLAB 中

例如我们把 Excel 中厦门建发(600153)从 2005 年 9 月 19 日到 2006 年 6 月 19 日的股价数据录入 MATLAB 中，保存变量名为 xmjf_600153，具体步骤如下：

(1) Excel 中厦门建发的股价数据分别为日期、收盘价、成交量，具体如图 10.5 所示。



	A	B	C	D
1	19-Sep-05	4.07	21578.7	
2	20-Sep-05	4	29130.2	
3	21-Sep-05	3.92	22623.7	
4	22-Sep-05	3.84	34239.7	
5	23-Sep-05	3.8	12593.9	
6	26-Sep-05	3.84	10715.1	
7	27-Sep-05	3.75	12640.1	
8	28-Sep-05	3.75	10442.3	
9	29-Sep-05	3.83	14334.1	
10	30-Sep-05	3.84	11143	
11	10-Oct-05	3.78	11908.6	
12	11-Oct-05	3.8	21989	
13	12-Oct-05	3.8	11272.8	

图 10.5　Excel 中厦门建发的股价数据

(2) 单击 startmatlab 启动 MATLAB，出现界面如图 10.6 所示。

(3) 单击 putmatrix 后将会弹出一个对话框，如图 10.7 所示。

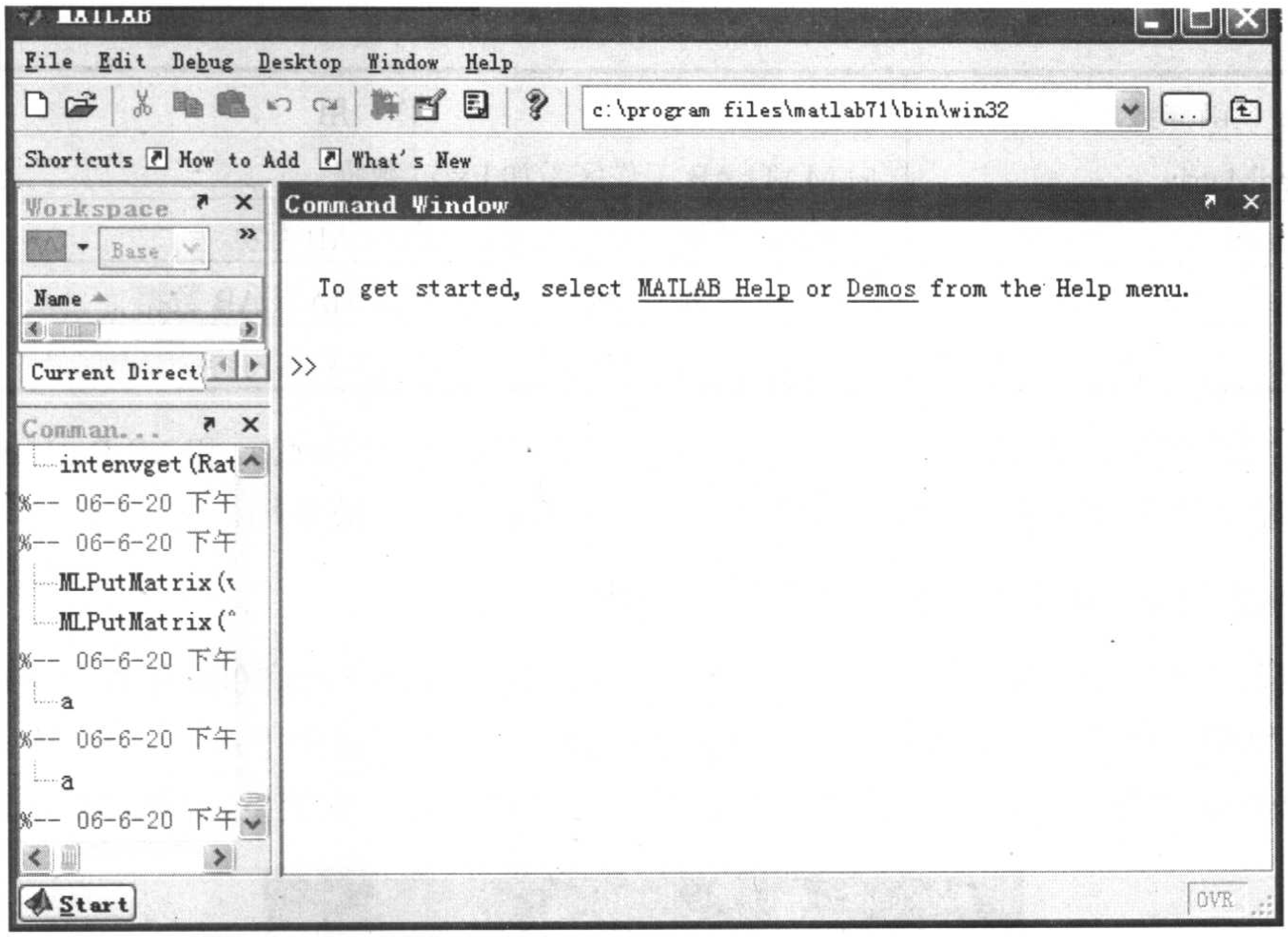


图 10.6　启动 MATLAB 界面

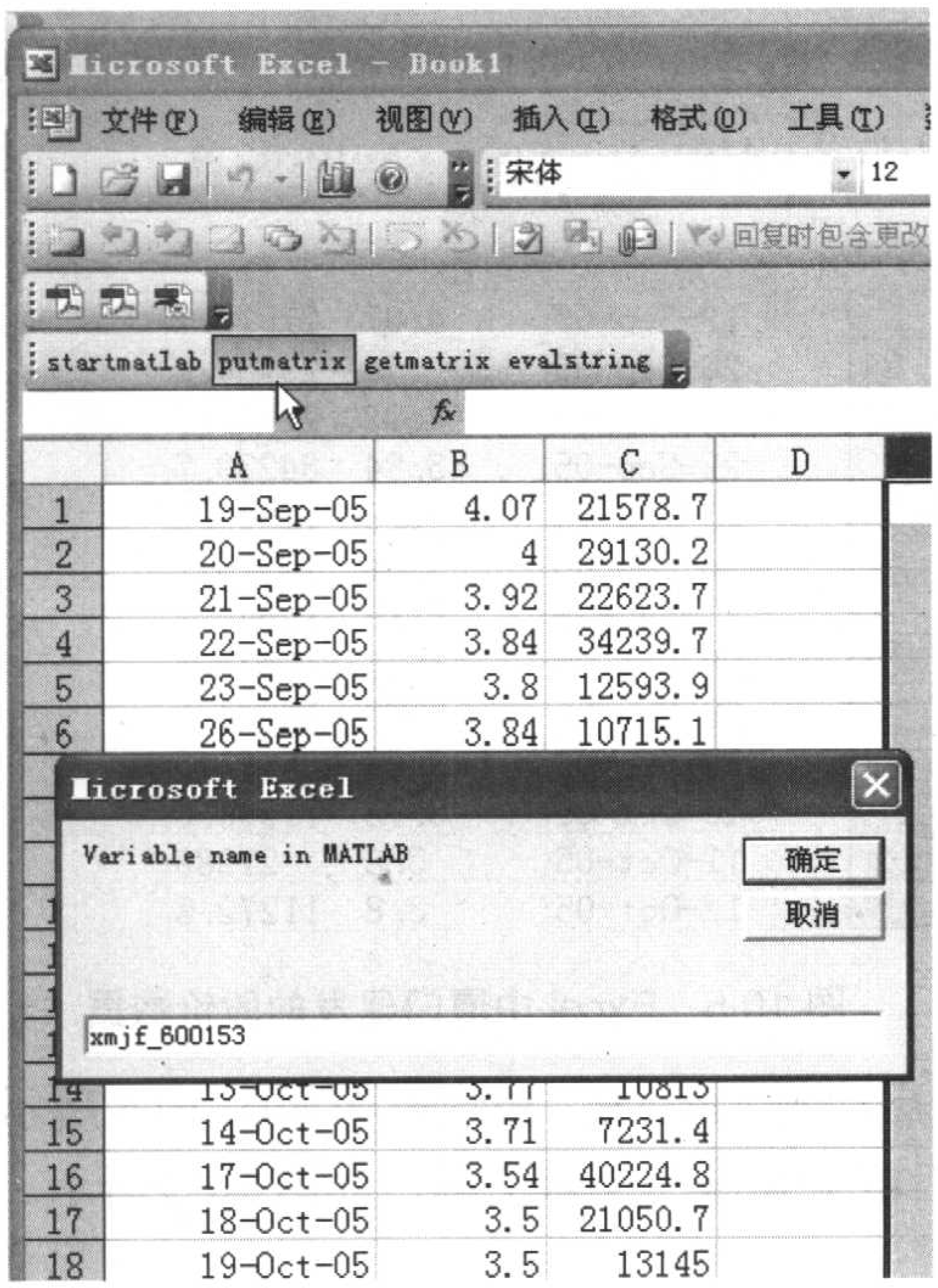


图 10.7　为股价在 MATLAB 中的数组命名

根据对话框中的提示将 Excel 中的数据区用鼠标选定，传输到 MATLAB 中名为 xmjf_600153 的变量，然后单击【确定】按钮，则 MATLAB 中就会出现 xmjf_600153 变量，Excel 中的数表被储存在变量 xmjf_600153 中，MATLAB 工作区内容如图 10.8 所示。

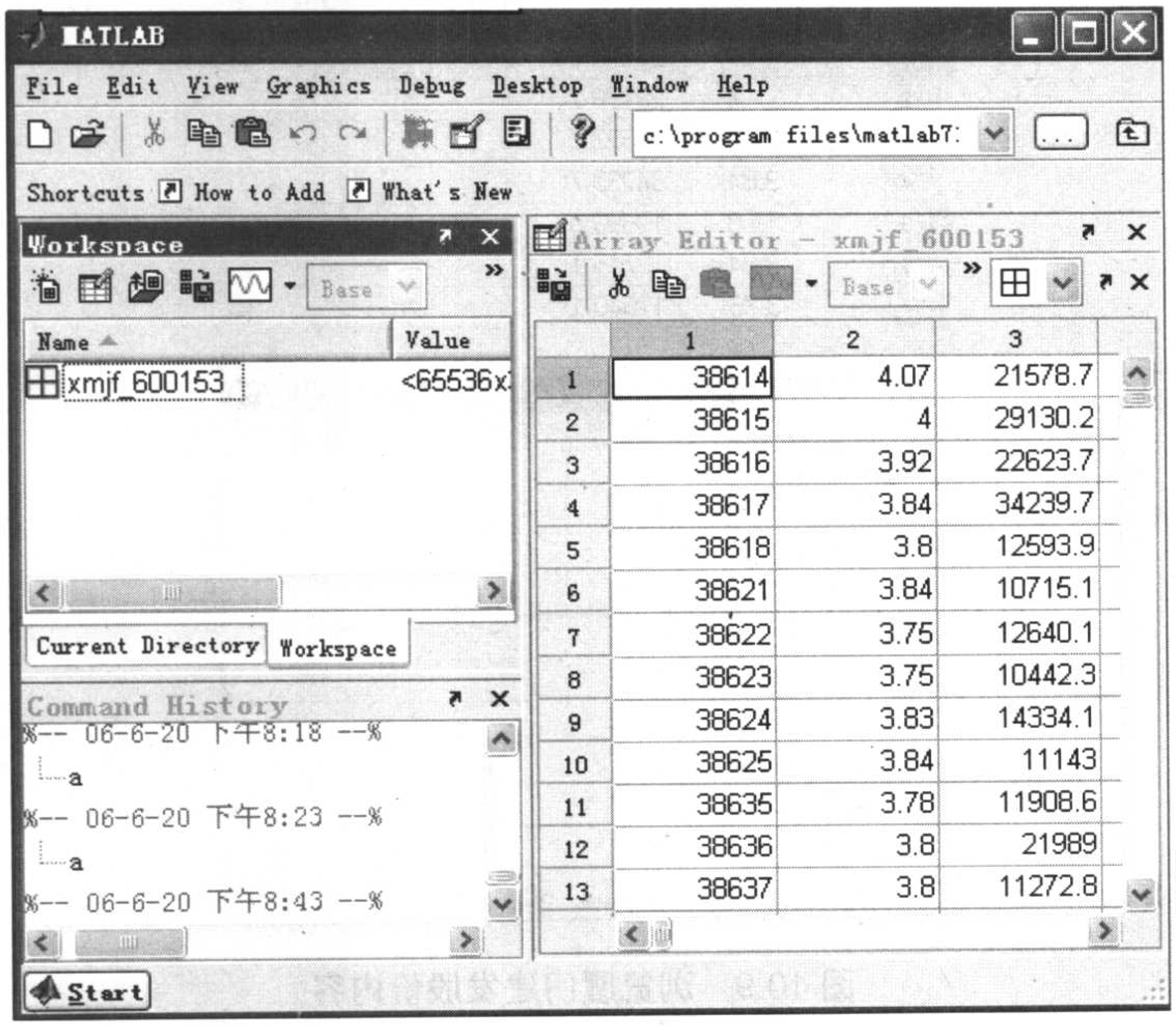


图 10.8　厦门建发在 MATLAB 中的变量

变量 xmjf_600153 第一列是整数，38614 表示 2006 年 6 月 19 日距 1900 年 1 月 1 日为 38 614 天，MATLAB 中的时间起始为 0000 年 1 月 1 日，和 Excel 中的时间相差 693 960 天。

3. 利用 MATLAB 中的 xlsread 函数读入 Excel 数据文件

把厦门建发的收盘价数据存入 Excel 中，命名为文件 book1.xls，下面在 MATLAB 中将其读出来，因为文件中有日期型数据，这样我们将数据分成两个部分：xmjf_date 为单元数据储存日期，xmjf_600153 储存收盘价、成交量，执行如下命令：

```
>> [xmjf_600153,xmjf_date]=xlsread('book1.xls');
```

图 10.9 所示为 MATLAB 浏览数据的结果。

实际上也可以将 MATLAB 中的数据读入 Excel 中，其函数是 xlswrite，其调用方式如下。

调用方式

```
xlswrite('filename', M)
xlswrite('filename', M, sheet)
xlswrite('filename', M, 'range')
```

```
xlswrite('filename', M, sheet, 'range')
```

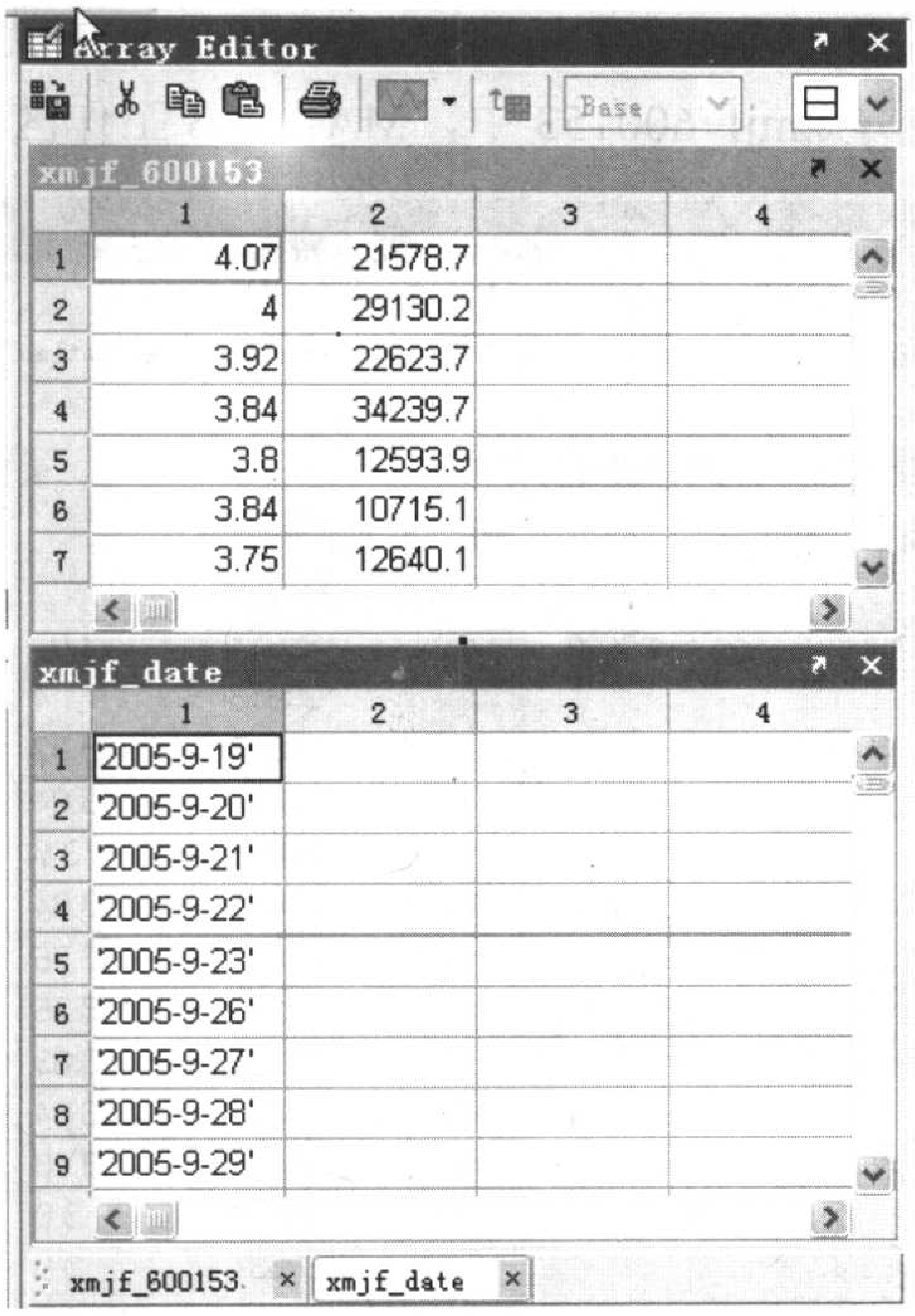


图 10.9　浏览厦门建发股价内容

输入参数

```
filename          %Excel 文件名
M                 %MATLAB 中的变量
sheet             %Excel 中的工作簿
range             %Excel 中工作簿的数据区域
```

下面在 MATLAB 中创建一个魔方矩阵，然后将其写入 Excel 中，代码如下：

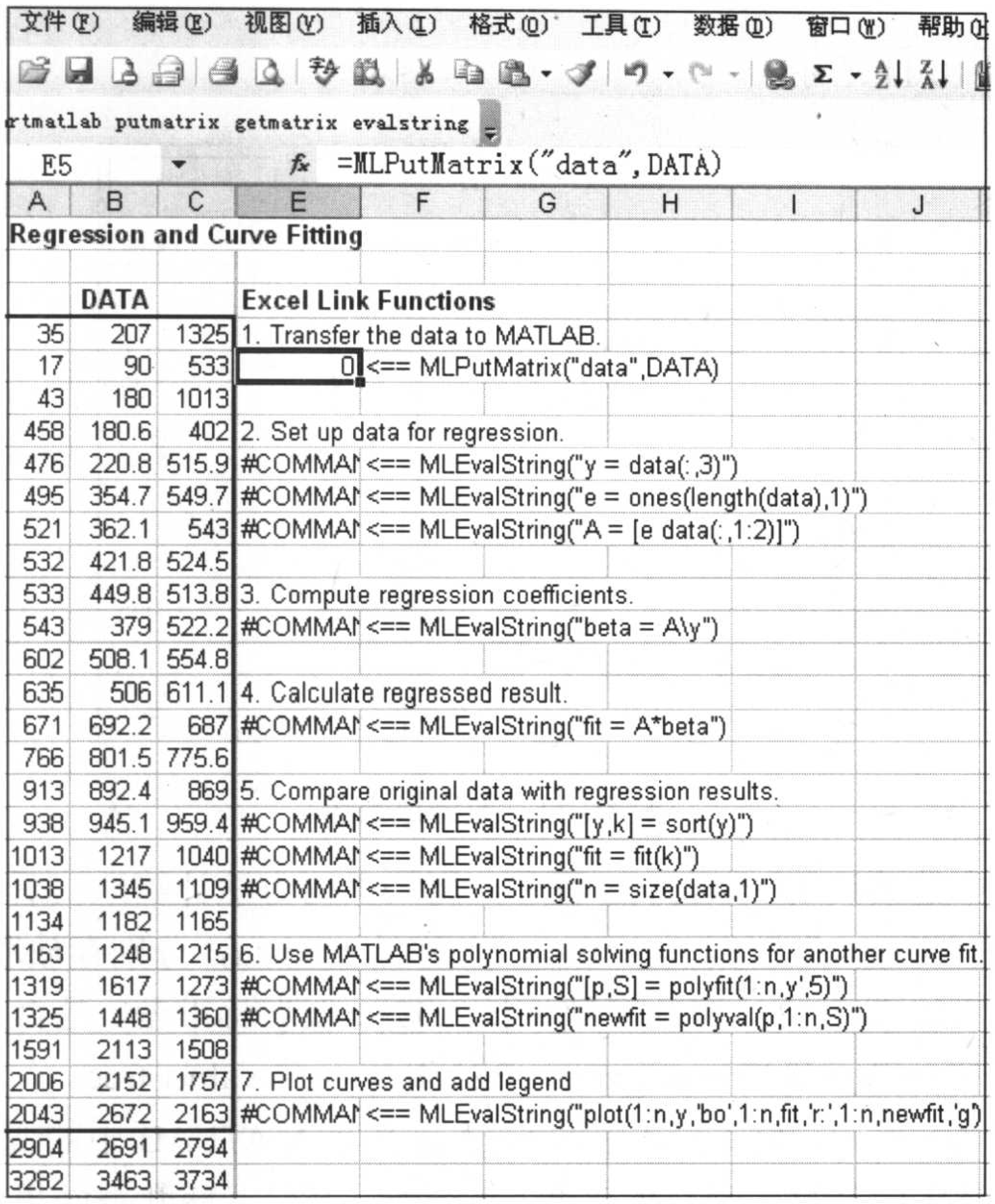
```
>> M=magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
>>%将数据写入 Excel 文件 abc 的工作簿 Sheet1 中，位置从 A2 到 C4
>> xlswrite('abc',M,'Sheet1','A2:C4') %将数据写入
```

打开 abc.xls，可以看到数据已经写入了文件 abc.xls 中。

4.　MATLAB 中自带的 Excel 文件介绍

【例 10-2】打开 MATLAB toolbox 中 exlink 目录下的 ExliSam.xls 文件，如图 10.10

所示。



A	B	C	E	F
Regression and Curve Fitting				
	DATA		Excel Link Functions	
35	207	1325	1. Transfer the data to MATLAB.	
17	90	533	0	<== MLPutMatrix("data",DATA)
43	180	1013		
458	180.6	402	2. Set up data for regression.	
476	220.8	515.9	#COMMAN	<== MLEvalString("y = data(:,3)")
495	354.7	549.7	#COMMAN	<== MLEvalString("e = ones(length(data),1)")
521	362.1	543	#COMMAN	<== MLEvalString("A = [e data(:,1:2)]")
532	421.8	524.5		
533	449.8	513.8	3. Compute regression coefficients.	
543	379	522.2	#COMMAN	<== MLEvalString("beta = A\y")
602	508.1	554.8		
635	506	611.1	4. Calculate regressed result.	
671	692.2	687	#COMMAN	<== MLEvalString("fit = A*beta")
766	801.5	775.6		
913	892.4	869	5. Compare original data with regression results.	
938	945.1	959.4	#COMMAN	<== MLEvalString("[y,k] = sort(y)")
1013	1217	1040	#COMMAN	<== MLEvalString("fit = fit(k)")
1038	1345	1109	#COMMAN	<== MLEvalString("n = size(data,1)")
1134	1182	1165		
1163	1248	1215	6. Use MATLAB's polynomial solving functions for another curve fit.	
1319	1617	1273	#COMMAN	<== MLEvalString("[p,S] = polyfit(1:n,y',5)")
1325	1448	1360	#COMMAN	<== MLEvalString("newfit = polyval(p,1:n,S)")
1591	2113	1508		
2006	2152	1757	7. Plot curves and add legend	
2043	2672	2163	#COMMAN	<== MLEvalString("plot(1:n,y,'bo',1:n,fit,'r:',1:n,newfit,'g')
2904	2691	2794		
3282	3463	3734		

图 10.10　Excel 数据拟合

下面分别介绍文件内容，打开 Sheet1，依次运行 7 个步骤。

第 1 步：将 Excel 中的 DATA 数据传给 MATLAB 工作区中的变量 data，变量 data 储存了 3 个变量、25 个观察值。

第 2 步：在 Excel 中执行 MATLAB 命令。

```
MLEvalString("y = data(:,3)")      %将变量 data 中的第三列作为因变量
MLEvalString("e = ones(length(data),1)")
MLEvalString("A = [e data(:,1:2)]")
```

第 3 步：计算回归系数。

第 4 步：计算回归结果。

第 5 步：将原结果和回归结果相比较。

第 6 步：用 MATLAB 中的多项式进行拟合。

第 7 步：对拟合结果作图。

```
MLEvalString("plot(1:n,y,'bo',1:n,fit,'r:',1:n,newfit,'g');
legend('data','fit','newfit')")
```

运行结果如图 10.11 所示。

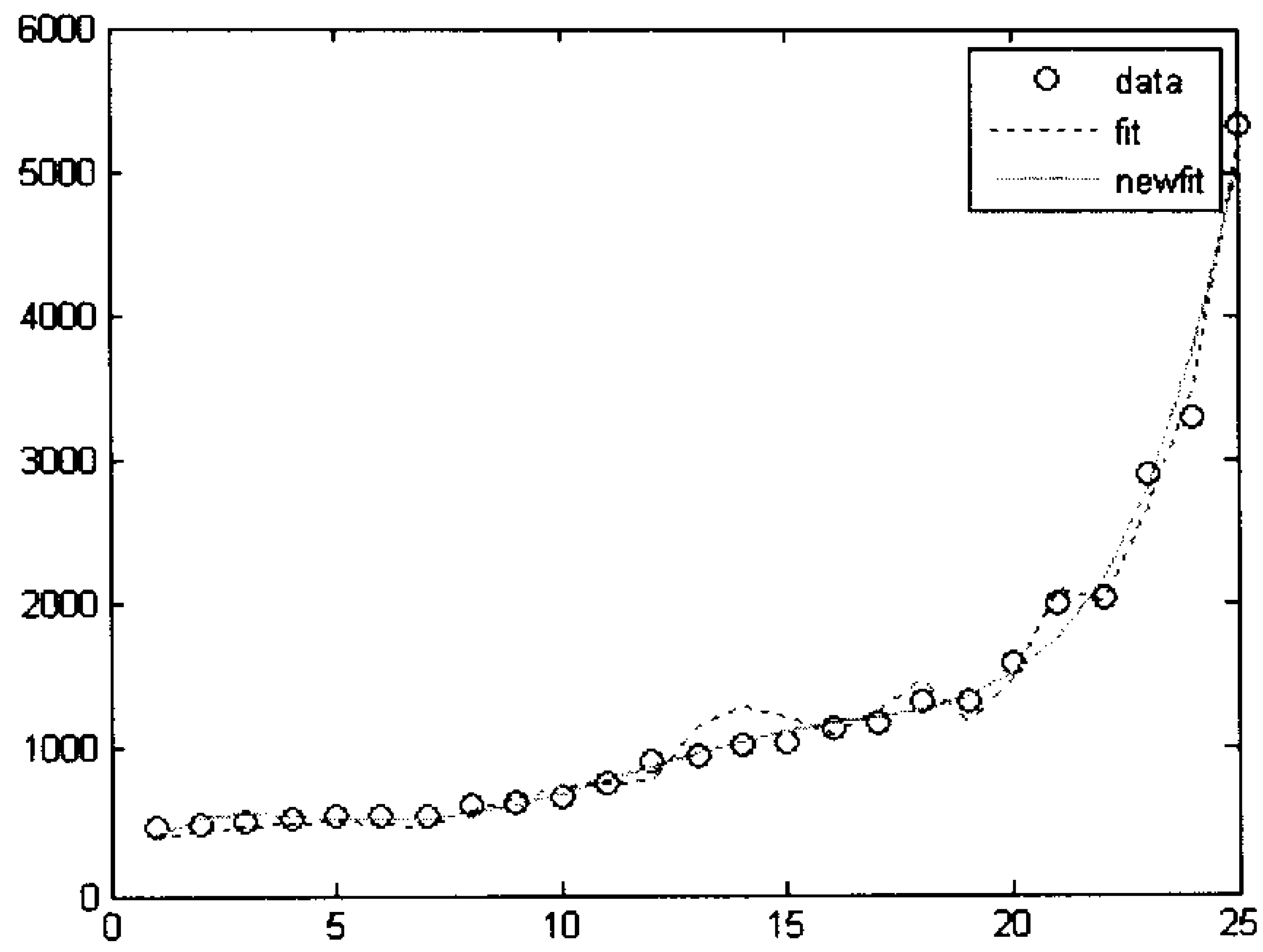


图 10.11　数据拟合图

下面在 Excel 中继续调用 MATLAB 程序，用二叉树方法计算欧式看跌期权价格。

第 1 步：打开“Sheet4”数据 B4：B10，名称是 bindata，分别存放股票价格、行权价、无风险利率、期权存续期(5/12)、时间离散步长(1/12)、股票标准差、是否为欧式(美式)期权，B15 开始存放二叉树各节点价格 asset_tree，B23 开始存放 value_tree。

第 2 步：激活 D5 单元，将 Excel 中的 bindata 传到 MATLAB 中的变量 b 中。激活 D12 单元，调用 MATLAB 中计算二叉树的函数 binprice，其二叉树数据结果保存在变量 p 中，价格保存在变量 o 中。激活 D11、D12 单元，分别保存在 MATLAB 的变量 asset_tree 与 value_tree 中，注意 B15 与 B23 中分别是 asset_tree 与 value_tree 的起点。

下面计算资产组合有效前沿。

第 1 步：打开 Sheet5，激活 A15，将 MATLAB 中的 F3：G3 数据传到 MATLAB 中的变量 labels。激活 B4：D9 数据，并将之传到 MATLAB 中，命名为变量 retseries。

第 2 步：激活 A15，根据输入的时间序列计算收益率与方差。激活 A20，执行 MATLAB 中的命令。

第 3 步：将变量输出到 Excel 中，风险(risk)输出到 F4，期望收益输出到 G4，资产权重输出到 H4 中。

第 4 步：调用 MATLAB 函数，绘制资产有效前沿。绘制的资产组合有效前沿如图 10.12 所示。

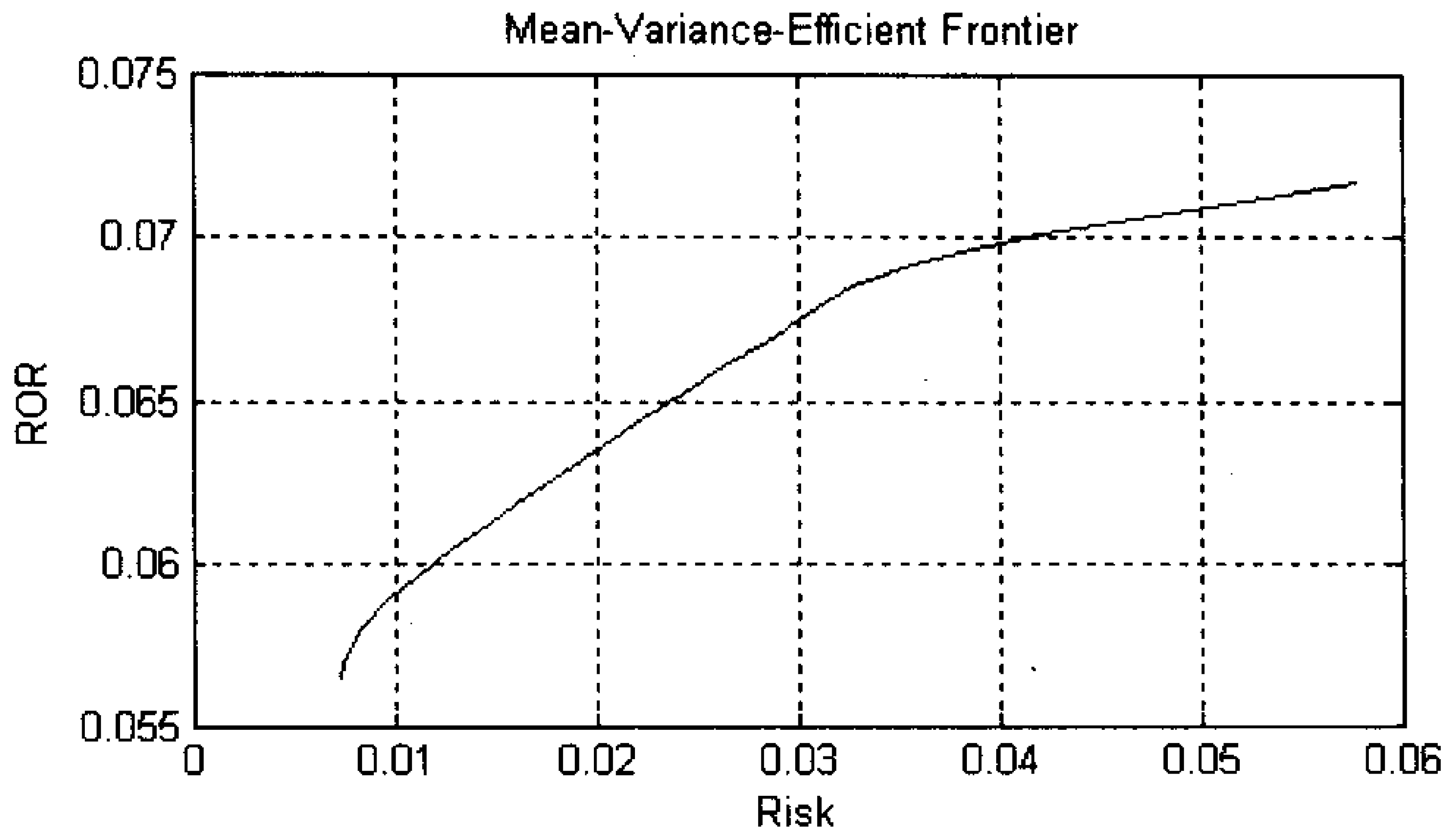


图 10.12　资产组合有效前沿

下面计算债券现金流和现金流之间的时间映射。

第 1 步：打开 Sheet6，分别将 Excel 中的数据传到 MATLAB 中。

激活 A18，将 Excel 中的到期日变量 maturity 传给 mATLAB 中的变量 maturity，激活 A19，将 Excel 中的债券利率传到 MATLAB 中的变量 cpnrate；激活 A20，将 Excel 中的债券结算日传给 MATLAB 中的变量 sd。

第 2 步：调用 MATLAB 中的现金和时间映射函数。

执行 A23，将 Excel 中的日期型数据转换为对应 MATLAB 中的日期；执行 A24，将债券转换为现金流。

第 3 步：调用 MATLAB 中将日期型数据转换为单元数据的函数。

第 4 步：将现金流数据传到 Excel 中。

第 5 步：画出现金流图，如图 10.13 所示。

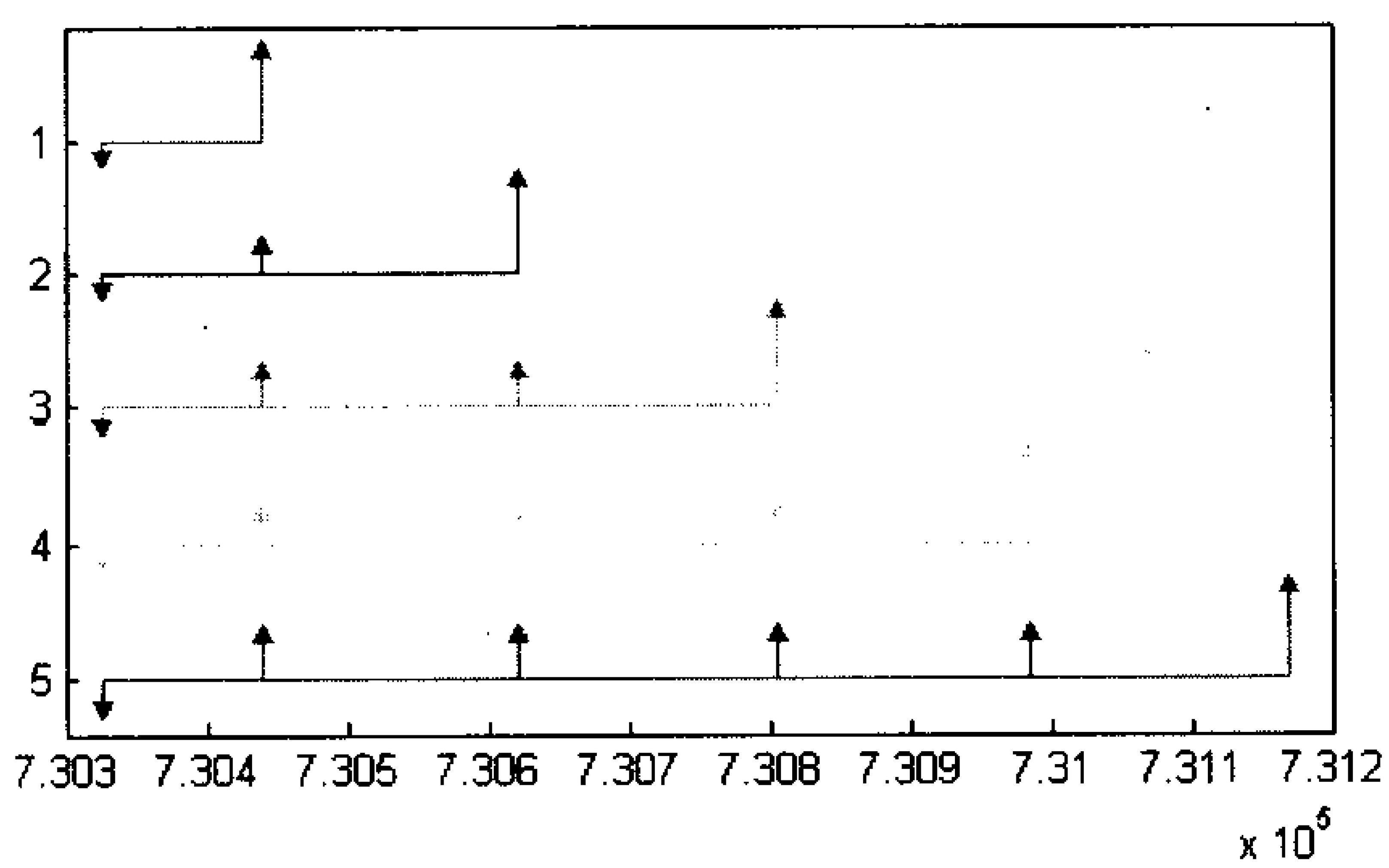


图 10.13　现金流图

5. 简易方法将 Excel 数据导入 MATLAB

首先在 Excel 中建立一个数据文件，如图 10.14 所示。

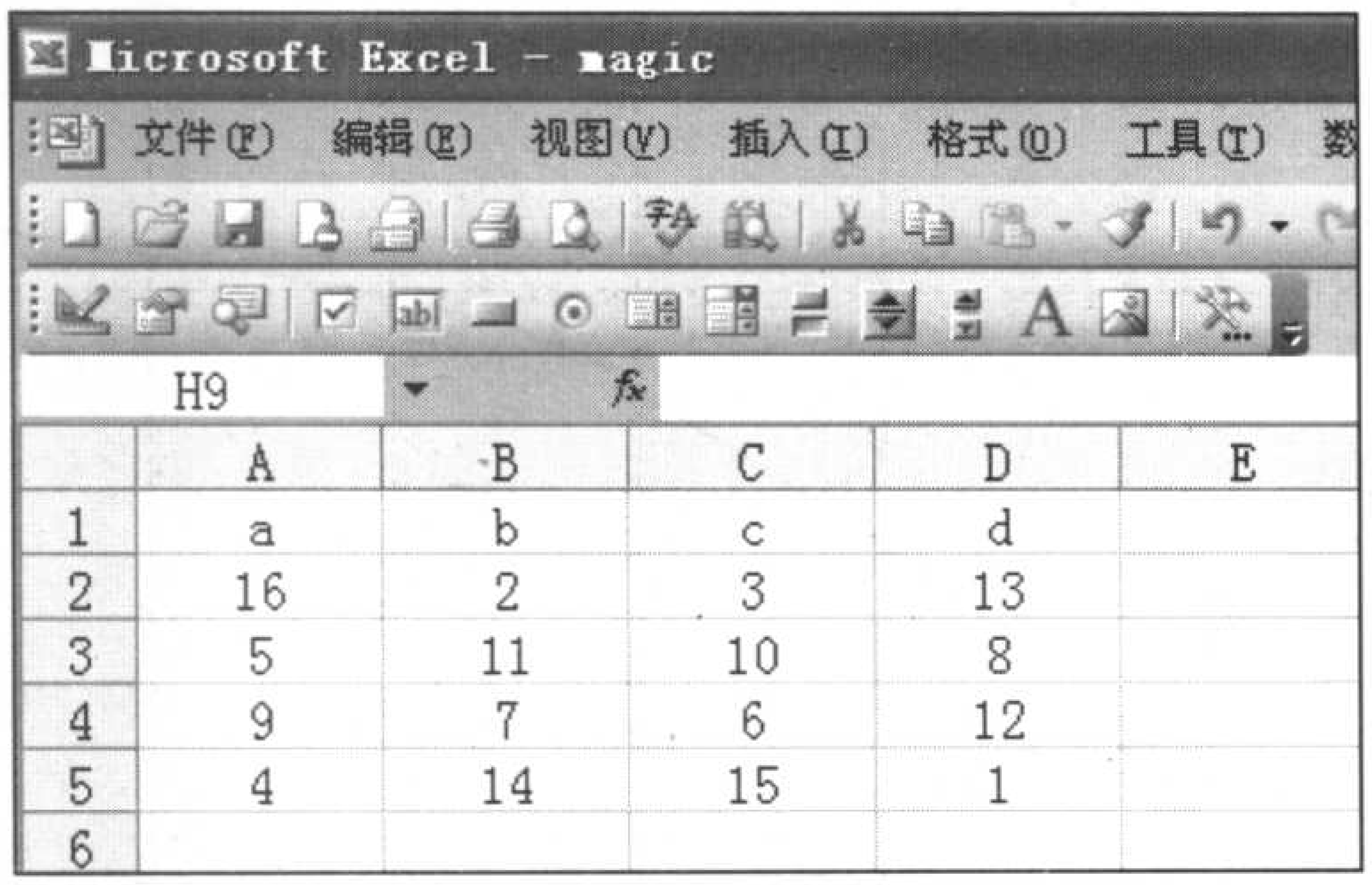


图 10.14　在 Excel 中建立数据文件

然后在 MATLAB 的文件目录浏览器下找到 magic 文件，如图 10.15 所示。

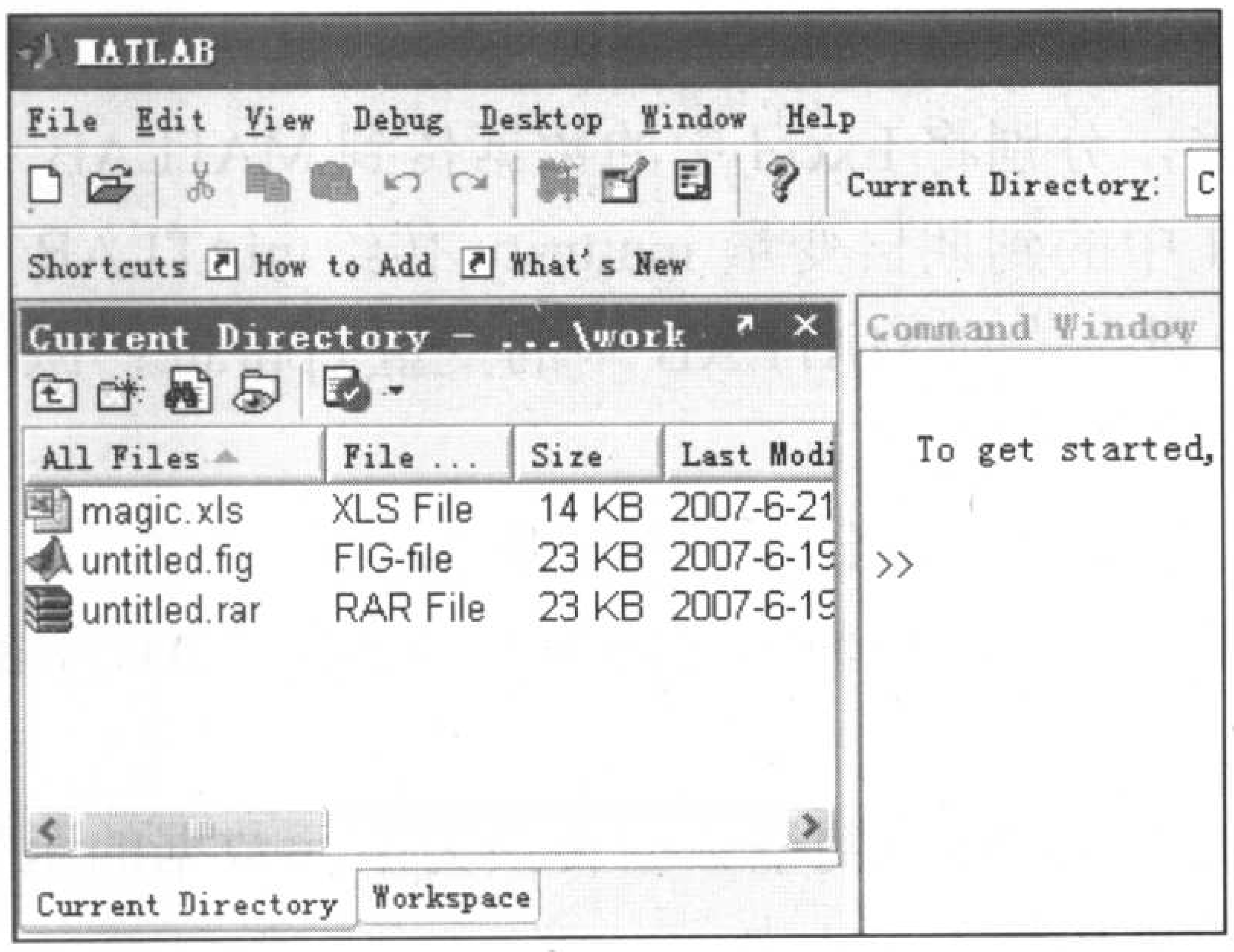


图 10.15　在 MATLAB 中调入 magic.xls 文件

选中 magic 文件，右击，在 File 菜单中选择 Import Data 命令，如图 10.16 所示，弹出如图 10.17 所示的对话框。

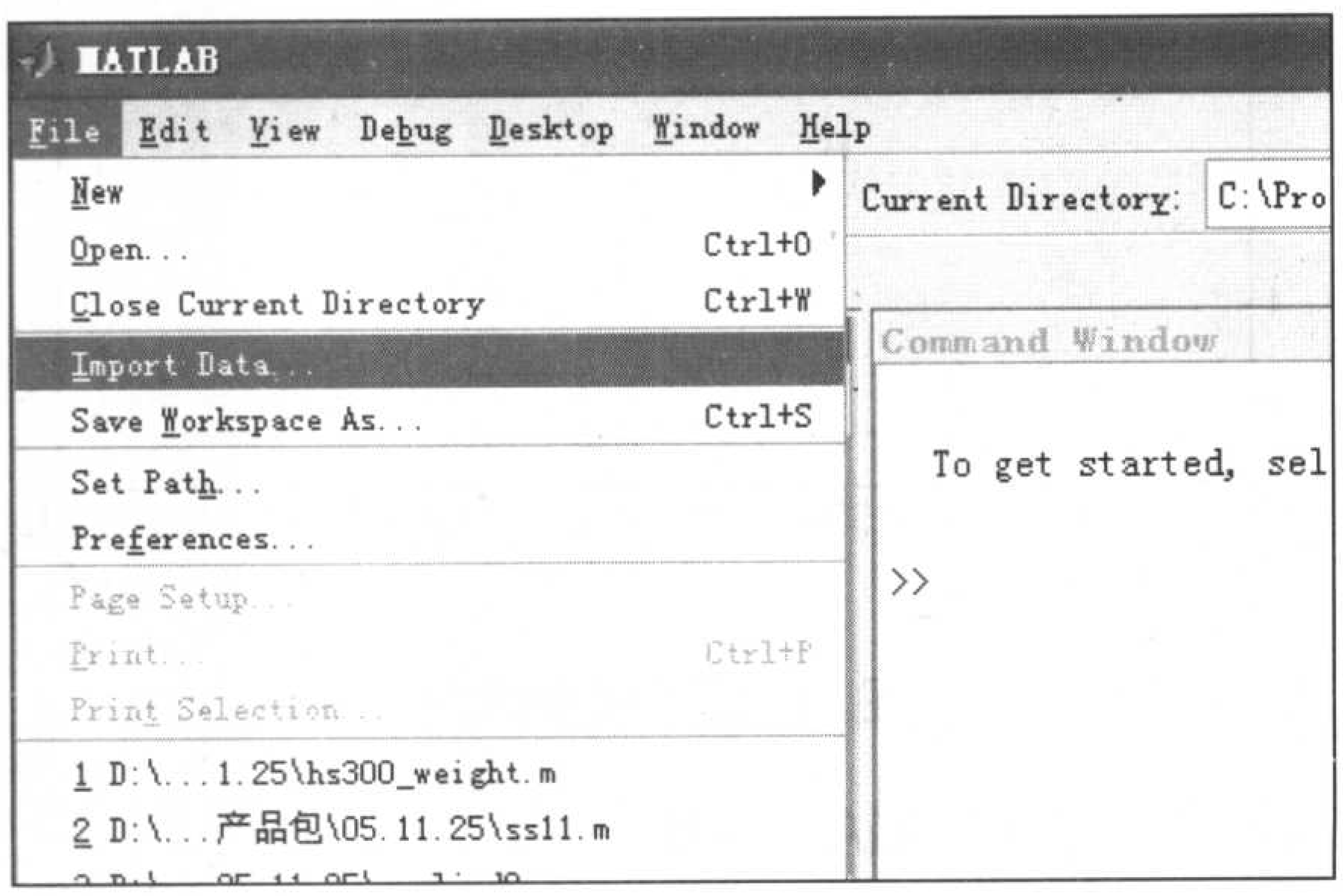


图 10.16　选择 Import Data(数据输入)命令

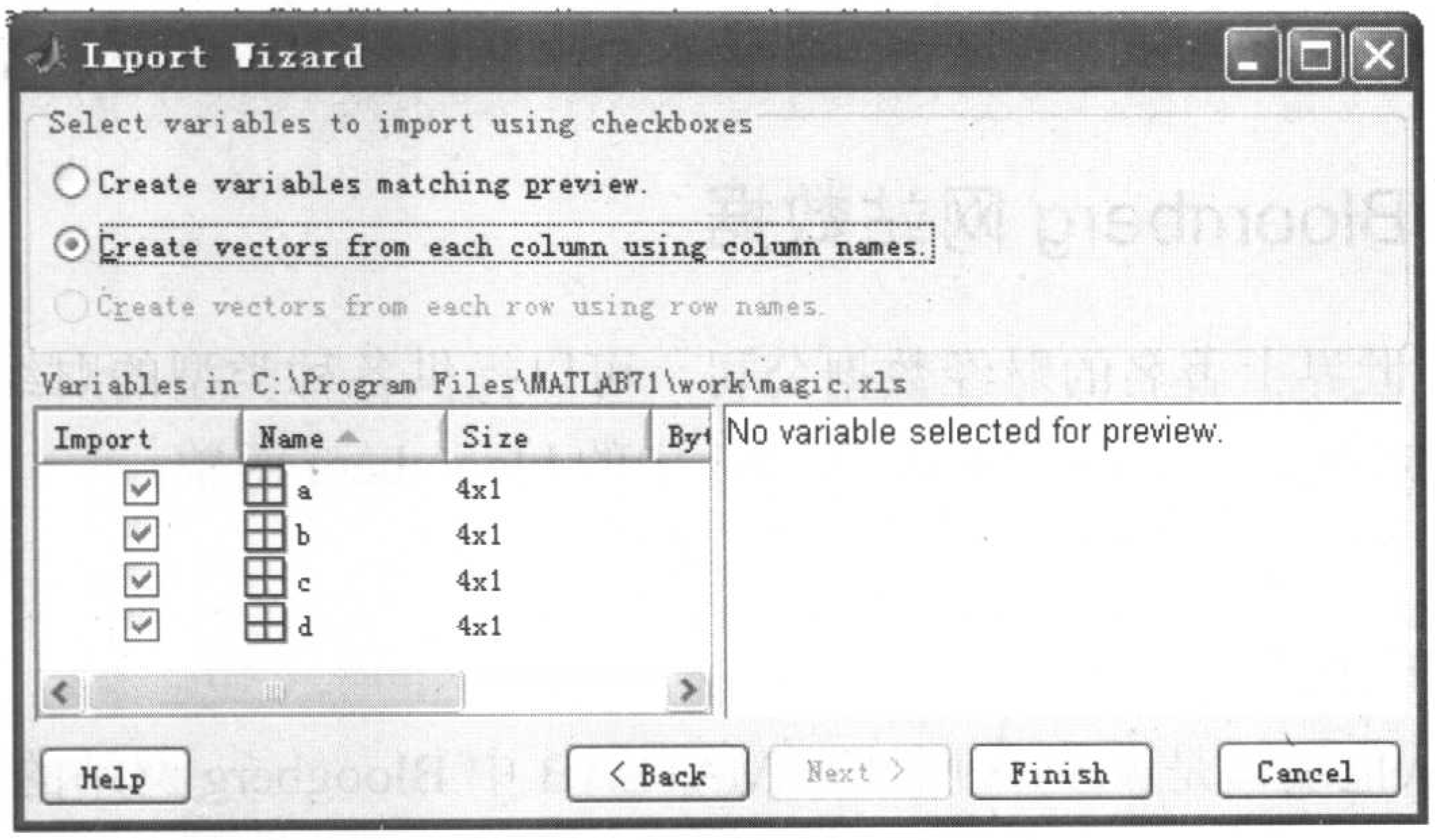


图 10.17　选择需要的变量

上面的单选按钮表示按类型保存或数据与字符串分开保存。下面的复选框表示每列作为一个变量，第一字母为变量名，现在选择 a、b、c、d 作为变量名，只需在其前面的复选框中打“√”即可，在工作区中可以浏览到这 4 个变量，如图 10.18 所示。

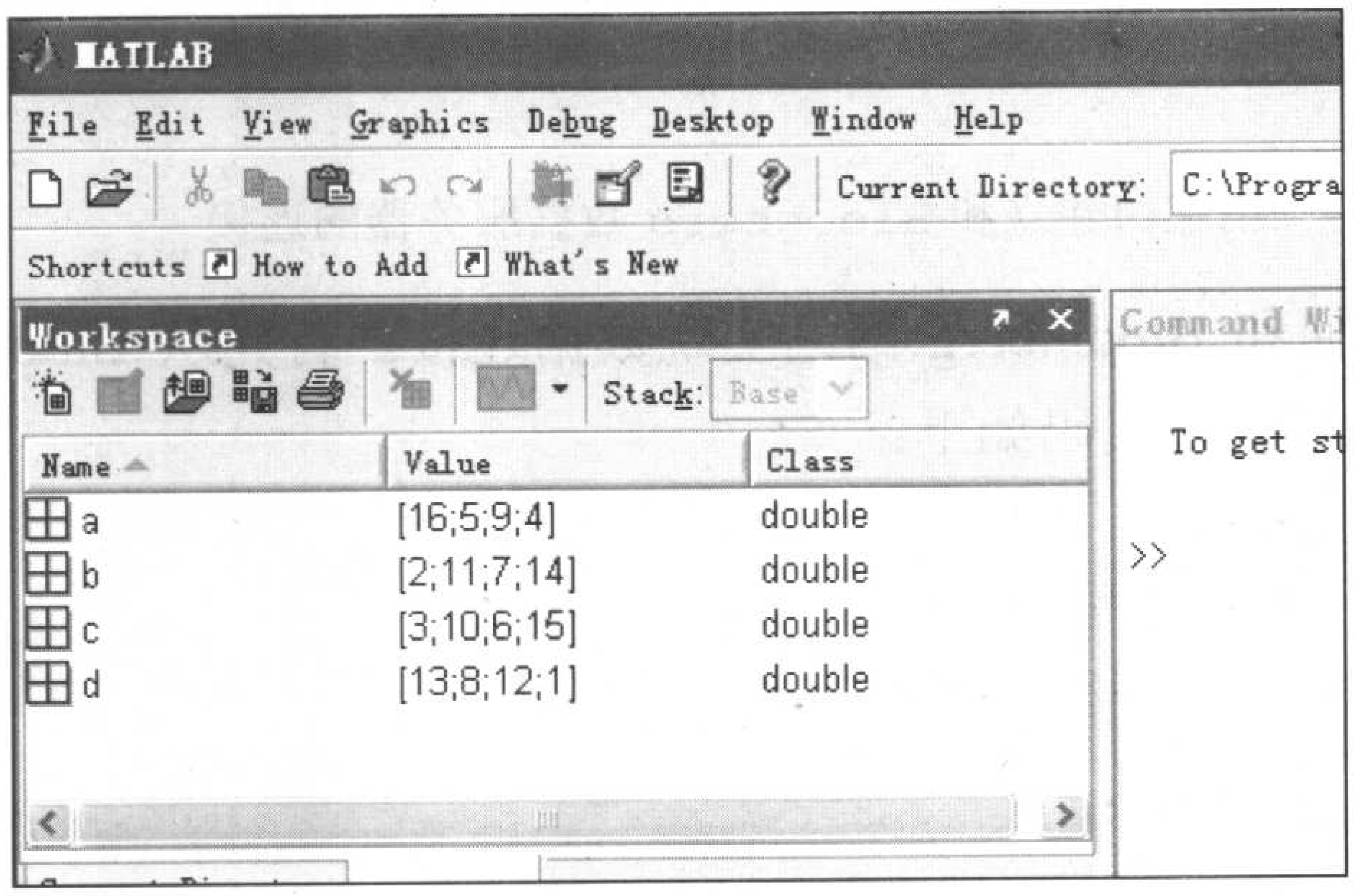


图 10.18　浏览选择的变量

10.2　MATLAB 与财经网站数据连接

MATLAB 中带有 Datafeed 工具箱(Datafeed Toolbox)，该工具箱可以实现 MATLAB 与财经网站的数据连接，从财经网站上下载财经数据，财经网站包括 Bloomberg、FactSet、Hyperfeed、Idc、Yahoo 等著名专业网站。

如果要和 Bloomberg、FactSet、 Byperfeed、Idc 等财经数据公司网站实现数据交流，必须安装客户端 API 插件，如果安装不正确，则得不到财经数据，和 FactSet FAST 建立数

据连接还需要得到该网站许可，具体可查看该公司网站 http://www.factset.com。

10.2.1 获得 Bloomberg 网站数据

Bloomberg 是世界上著名的财经数据公司，可以提供各种类型的财经数据，如果要和 Bloomberg 网站建立连接，需要调用 MATLAB 中的 bloomberg 函数。

调用方式 1

```
Connect = bloomberg
```

MATLAB 默认值端口是 8194，网址是 MATLAB 中 Bloogberg 网站的默认网址。

调用方式 2

```
connect = bloomberg(PortNumber, IPAddress)
```

输入参数

```
PortNumber      %机器端口
IPAddress       %字符串，财经网站 Internet 网址
```

输出参数

```
connect         %用端口建立和 Bloogberg 数据服务器的连接
```

下面举例说明如何和 Bloomberg 网站实现数据连接。首先从 Bloomberg 公司网站上下载 API 软件，并完成安装，代码如下：

```
>> c = bloomberg
  c =
  connection: 20135448
     ipaddress: []
     port: 8194
```

如果要查看端口号、版本等信息可以调用 get 函数，代码如下：

```
>>p = get(c,{'Port';'Version'}) %查看连接 c 中的端口号、版本
p =
        port: 8194
        version: 1.9000
```

端口号为8194，版本为1.9。

进一步地接收 Bloogberg 数据就要使用连接函数 fetch。

调用方式

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'HEADER', 'Flag','Ident')
```

```
data = fetch(Connect, 'Security', 'GETDATA', 'Fields', 'Override', 'Ident',
'Values')
data = fetch(Connect, 'Security', 'TIMESERIES', 'Date', 'Minutes',
'TickField')
data = fetch(Connect, 'Security', 'HISTORY, 'Fields', 'FromDate', 'ToDate',
        'Period', 'Currency', 'Ident')
ticker = fetch(Connect, 'SearchString', 'LOOKUP', 'Market')
data = fetch(Connect, 'Security', 'REALTIME', 'Fields', MATLABProg)
data = fetch(Connect, Security, 'STOP')
```

输入参数

Connect	%bloogberg 函数输出值
Security	%Bloogberg 网站服务器中的股票名称，是一个字符串，也可以是一个或多个股票名称构成的单元变量
Flag	%默认值是最近买入价、卖出价、成交价。如 Flag=TODAY，表示今天的价格
Currency	%(Optional)货币的回报，相关货币名称在@bloomberg/bbfields.mat 中。默认值为[]
Ident	%(Optional)证券类型标识符，证券类型保存在数据文件 bbfields.mat 中，默认值为[]
Fields	%字段名，具体内容见 bbfields.mat，默认值为[]
Override	%(Optional)Override 字段名清单，字符串或者单元变量，默认值为[]
Values	%(Optional)保存 Override 的值
Date	%证券时间序列日期
Minutes	%(Optional)间隔分钟
TickField	%(Optional) 字符串或者数值，例如 TickField = 'Trade'或者 TickField = 1 将返回交易数据，用函数 dftool('ticktypes')将返回交易日字段
FromDate	%历史数据开始时间
ToDate	%历史数据结束时间
Period	% Optional)数据类型，'d' 日数据 (默认值)，'w' 周数据，'m' 月数据，'q'季度数据，'y'年数据
Currency	%(Optional) 货币名称，在文件 @bloomberg/bbfields.mat 中列出了名称
Market	%金融市场名称，Comdty 表示商品市场，Corp 表示公司债，Curncy 表示货币，E quity 表示证券市场，Govt 表示政府债，Index 表示指数，M-Mkt 表示货币市场证券，Mtge 表示抵押证券，Muni 表示市政债券市场，Pfd 表示优先股市场

如果需要获得证券文件头，即可在 Command 窗口中执行如下命令：

```
>>D = fetch(C,'ABC US Equity')
```

如果需要获得 ABC 股票开盘价、收盘价，可在 Command 窗口中执行如下命令：

```
>>D = fetch(C,'ABC US Equity','GETDATA',{'Last_Price';'Open'})
```

如果需要获得 ABC 股票当日的时间序列，可在 Command 窗口中执行如下命令：

```
>>D = fetch(C,'ABC US Equity','TIMESERIES', now)
```

如果需要获取证券历史月收盘价，时间从 1999 年 8 月 1 日到 2000 年 9 月 30 日，可在 Command 窗口中执行如下命令：

```
>>D = fetch(C,'ABC US Equity','HISTORY','Last_Price','8/01/99',
'9/30/00','m')
```

如果需要获得 IBM 股票 1999 年 11 月 16 日的时间序列，可在 Command 窗口中执行如下命令：

```
>>data = fetch(c1, 'IBM US Equity', 'TIMESERIES', '11/16/99')
```

结果如下：

```
31.00 730440.31 130.00 1000.00
32.00 730440.31 130.00 200.00
32.00 730440.35 129.50 10000.00
31.00 730440.35 129.50 100.00
32.00 730440.35 129.50 100.00
1.00 730440.56 129.25 4000.00
31.00 730440.56 129.38 1500.00
32.00 730440.56 129.50 500.00
1.00 730440.56 129.63 5000.00
31.00 730440.56 129.63 400.00
32.00 730440.56 129.63 200.00
1.00 730440.56 129.69 5000.00
31.00 730440.56 129.69 500.00
32.00 730440.56 129.69 500.00
31.00 730440.56 129.75 100.00
32.00 730440.56 130.00 100.00
1.00 730440.56 130.00 5000.00
1.00 730440.56 129.88 5000.00
31.00 730440.56 129.88 300.00
```

第一列是 Flag 值，第二列是时间序列，第三列是成交价格，第四列是成交量。

如果需要获取 IBM 股票历史上的收盘价，时间是 1999 年 7 月 15 日到 8 月 2 日，可在 Command 窗口中执行如下命令：

```
>>data = fetch(c1, 'IBM US Equity', 'HISTORY', 'Last_Price', '07/15/99',
'08/02/99')
data =
730316.00 136.31
730317.00 136.25
```

```
730320.00 134.63
730321.00 128.25
730322.00 129.00
730323.00 123.88
730324.00 124.81
730327.00 123.00
730328.00 126.25
730329.00 128.38
730330.00 125.38
730331.00 125.69
730334.00 122.25
```

第一列是日期，第二列是收盘价。

确定是否已经和 Bloomberg 网站建立连接，可以执行如下命令：

```
>>x = isconnection(Connect)
```

如果返回值为 1 表示已经连接上，0 表示没有连接上。

如果想关掉和 MATLAB 的连接，可以执行如下命令：

```
>>close(Connect)
```

10.2.2 获得 Yahoo 网站数据

1. 建立和 Yahoo 网站服务器连接

调用方式

```
Connect = yahoo
Connect = yahoo('URL', 'IPAddress', PortNumber)
```

输入参数

```
URL            %Yahoo 网站网址
IPAddress      %代理服务器 IP 地址
PortNumber     %代理服务器端口
```

建立和 Yahoo 网站的连接可以执行如下命令：

```
>>Connect = yahoo('http://quote.yahoo.com','111.222.33.444',5678)
```

2. 获得 Yahoo 网站数据

调用方式

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Date')
```

```
data = fetch(Connect, 'Security', 'Fields', 'Date')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')
```

输入参数

```
Connect        %yahoo 函数输出值
Security       %证券名称，如 IBM 股票可以执行“IBM”，Yahoo 网站每次只能输入一只股票
Data           %日期
FromData       %证券历史数据开始日期
ToData         %证券历史数据终止日期
Period         %选取数据类型，“d”表示日数据，“w”表示周数据，“m”表示月数据
Fields         %字符串，例如 Symbol 表示证券名称，Last 表示上一个交易日数据，Data
                表示日期，Time 表示时间，Open 表示开盘价，High 表示最高价，Low 表示
                最低价，Volume 表示成交量
```

例如和 Yahoo 网站进行连接可以执行如下命令：

```
>> Connect = yahoo
Connect =
     url: 'http://quote.yahoo.com'
     ip: []
     port: []
```

如果要进一步查询可口可乐 2000 年 4 月 6 日的收盘价，可以执行如下命令：

```
>>ClosePrice = fetch(Connect,'ko','Close','Apr 6 00')
ClosePrice =
        730582   45.75
```

从上面可以看出可口可乐收盘价为 45.75 美元。

如果今天是 2006 年 11 月 6 日，如果需要查看 IBM 股票上一个交易日的收盘价，可以执行如下命令：

```
>> ClosePrice = fetch(Connect,'IBM','last')
ClosePrice =
      Last: 91.4100
```

可以知道 IBM 股票上一个交易日的收盘价为 91.41 美元。

如果查看 MATLAB 是否与网站建立了连接，可以执行如下命令：

```
>> isconnection(Connect)
ans =
         1
```

返回值是 1，说明已经建立连接，如果返回值是 0，则说明没有建立连接。如果需要取消一个连接可以执行如下命令：

```
>> close(Connect)
```

MATLAB 中用 get 函数查看连接性质，如果查看连接网址，可以执行如下命令：

```
>> get(Connect,'url')
ans =
http://quote.yahoo.com
```

则显示连接网址是 http://quote.yahoo.com。

10.2.3　获取 FactSet 网站数据

调用方式

```
Connect = factset('UserName','SerialNumber','Password', CustomerID')
```

输入参数

```
UserName        %用户注册名
SerialNumber    %用户序列号
Password        %密码
CustomerID      %FactSet 用户 ID
```

如果要获取 Connect 内容可以调用 get 函数。

调用方式

```
value = get(Connect, 'Property Name')
```

输入参数

```
PropertyName    %Connect 的相关属性，属性包括 user、serial、password、cid
```

具体代码如下：

```
>> Connect = factset('Fast_User','1234','Fast_Pass','userid')
>> h = get(Connect)
  h=
       user: 'Fast_User'
     serial: '1234'
   password: 'Fast_Pass'
        cid: 'userid'
```

判断 MATLAB 有没有和 FactSet FAST 建立联系可以执行如下命令：

```
>>x = isconnection(Connect)
```

返回值是 1 表示已经连接，如果是 0 表示没有连接。

如果要关闭和 FactSet FAST 之间的联系可以执行如下命令：

```
>>close(Connect)
```

10.2.4 获取 Hyperfeed 中的数据

调用方式

```
data = fetch(Connect, 'Security')
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Date')
data = fetch(Connect, 'Security', 'Fields', 'Date')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'FromDate', 'ToDate', 'Period')
```

输入参数

```
Connect        %MATLAB 中 hyperfeed 函数的输出值
Security       %证券名称
Fields         %字符串或单元变量，其取值为 Symbol、LastDate、Time、Change、
                OpenHigh、Low、Volume
Date           %日期字符串，或含有日期字符串
FromDate       %开始日期
ToDate         %最后日期
Period         %股票数据类型，可以选择的有 d（日数据）、w（周数据）、m（月数据）、v（单
                个数据）
```

如要获得 Coca Cola 公司 2000 年 4 月 6 日的收盘价，可以执行如下命令：

```
>> c = hyperfeed('History');
>> ClosePrice = fetch(c,'ko','Close','Apr 6 00')
ClosePrice =
              730582.00        45.75
```

可知 2000 年 4 月 6 日可口可乐的收盘价是 45.75 美元。

10.2.5 获得 FT 网站的数据

调用方式

```
data = fetch(Connect, 'Security', 'Fields')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate')
data = fetch(Connect, 'Security', 'Fields', 'FromDate', 'ToDate', 'Period')
data = fetch(Connect,'','GUILookup','GUICategory')
```

输入参数

同前。

输出参数

同前。

10.2.6　MATLAB 和财经网站数据接口 GUI

MATLAB 软件为了和财经类数据网站更好地进行数据交流，方便使用者研究，开发了 GUI 进行数据交换。

若要打开 GUI，可在 Command 窗口中执行如下命令：

```
>> dftool
```

打开的 GUI 界面如图 10.19 所示。

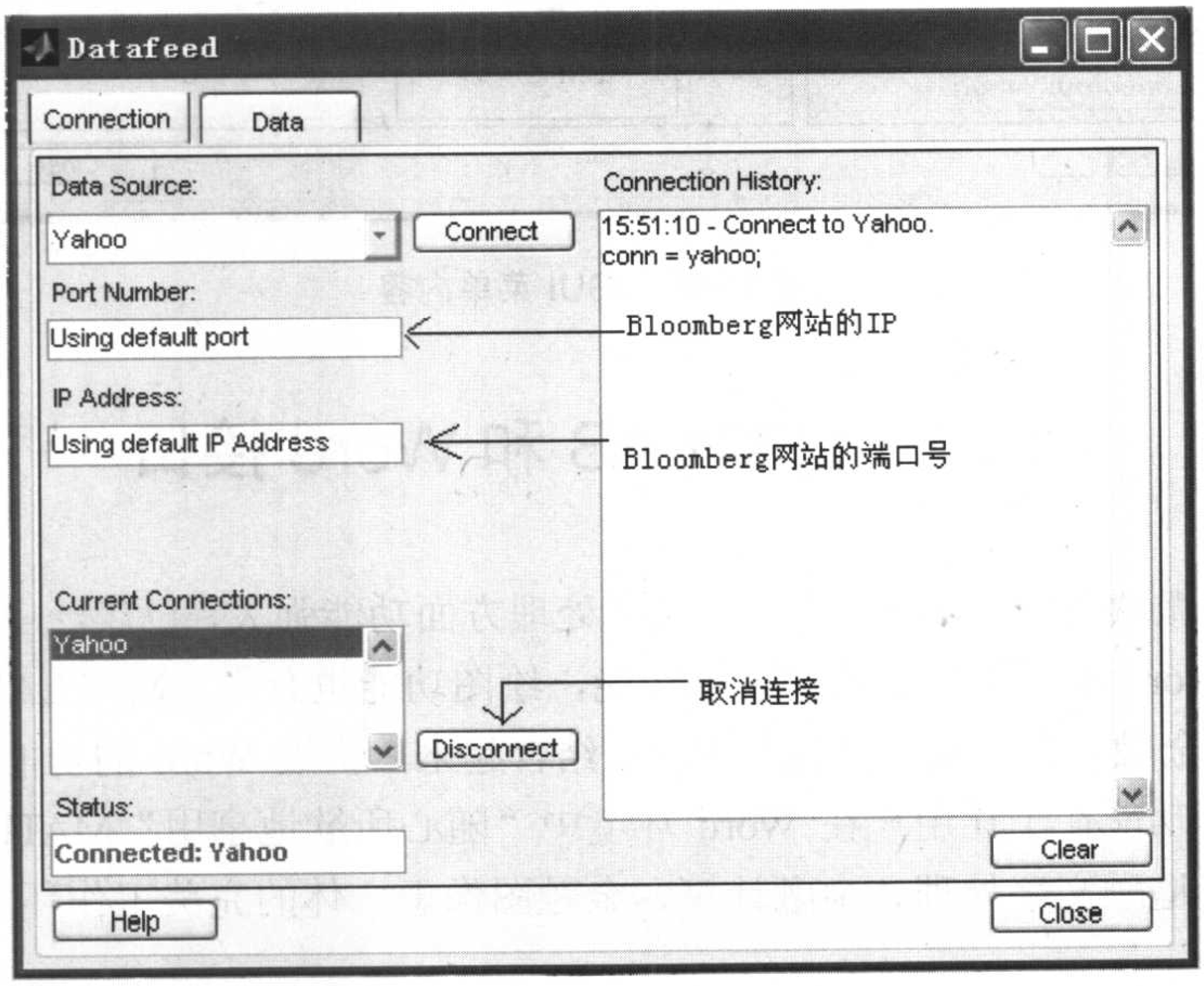


图 10.19　财经数据 GUI 界面

切换到 Data 选项卡，如图 10.20 所示。

在 Command 窗口中执行如下命令：

```
>> ibm
732984   91.41
732983   91.68
732982   91.8
```

可以知道变量 ibm 中保存了 IBM 股票的收盘价。

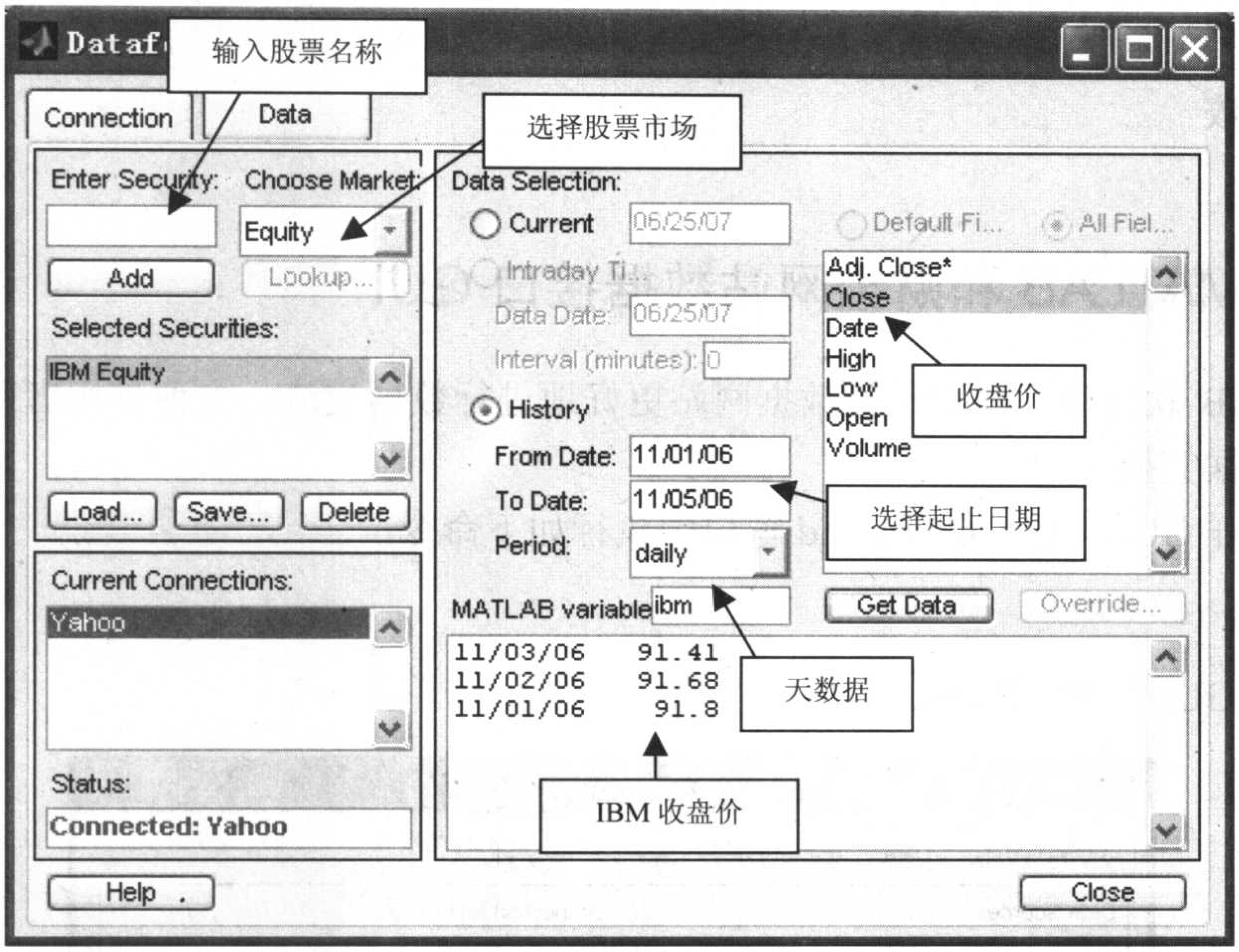


图 10.20　GUI 菜单内容

10.3　MATLAB 和 Word 接口

众所周知，微软公司的 Word 软件在文字处理方面功能强大，已广泛应用于科学研究的各个领域。但 Word 本身不具备数据运算功能，绘图功能也有限。MATLAB 中的 Notebook 把 MATLAB 的金融计算和绘图功能与 Word 结合起来，扩展 Word 的功能，使二者能协调一致地工作，其功能在于让用户在 Word 环境中“随心所欲地享用”MATLAB 的浩瀚科技资源，为用户营造融文字处理、金融计算、金融图像于一体的完美工作环境。

10.3.1　启动 Notebook

1. 从 MATLAB 中启动 Notebook

在 MATLAB 中直接执行如下命令启动 Notebook：

```
>> notebook -setup
Welcome to the utility for setting up the MATLAB Notebook
for interfacing MATLAB to Microsoft Word
Setup complete
```

上述信息表明 Notebook 已经安装完成。

在 MATLAB 中执行如下命令：

```
>> notebook
Warning: MATLAB is now an automation server
```

MTALAB 提示已经启动 Notebook，并且出现如图 10.21 所示的界面。

图 10.21　Notebook 界面

2. 从 Word 中启动 Notebook

方法一是直接打开 MATLAB71\notebook\pc 文件夹下的 M-book 文件，直接进入编辑状态。

方法二是在 Word 菜单中选择【工具】|【模板和加载项】命令，然后打开【模板】选项卡，单击【选用】按钮，选中 M-book 即可。

10.3.2　创建和运行 Word 中的计算区

计算区通常由 Word 文本和细胞(群)混合的若干段落构成，它被当作 Word 的一个“节”处理。MATLAB 提供计算区功能目的在于使得计算区的计算操作与电子文档其他部分独立，

不管计算区中包含多少细胞群，只需一次操作就可以对所有细胞群进行计算。

计算区操作独立于 Word 文档其他部分，但是所有计算都是通过同一个 MATLAB“引擎”实现的，生成的变量放在 MATLAB 的同一个工作区中。

计算区可用于实时运行电子演讲稿中的 MATLAB 指令，而不用切换到 MATLAB 状态。

10.4 MATLAB 与 ActiveX 接口

10.4.1 ActiveX 基本介绍

ActiveX 是一种基于 Microsoft Windows 系统组件的集成协议，通过 ActiveX，开发商和终端客户就把来自不同商家的 ActiveX 组件集成在自己的应用程序中，这样可以提高开发效率，避免在一些低层次上花费更多时间。

ActiveX 是面向对象技术的集合，这些技术都有的共同基础是“组件对象模型”(Component Object Model)，简称 COM。

MATLAB 支持两种 ActiveX，分别是 ActiveX 控件封装集成和 ActiveX 自动化，ActiveX 控件(Control)是指那些可视的、能编程的集成于 ActiveX 容器(Containor)的应用组件，最常见的是 Microsoft Internet Explorer 和 Web Browser Control。

ActiveX 自动化(Automation)使得 MATLAB 控制、受控于其他软件，当 MATLAB 受控于其他软件时，MATLAB 就变成了自动化服务器(Automation Server)；当 MATLAB 控制其他软件时，就表现为自动化客户(Automation Client)。MATLAB 自动化服务器包括在 MATLAB 中执行指令、与 MATLAB 交换数据。

MATLAB 自动化客户功能仅是 MATLAB ActiveX 控件封装集成功能的子集，所有 ActiveX 控件都是 ActiveX 自动化服务器，其内容如表 10.3 所示。

表 10.3 ActiveX 控件

函　数	功　能
activecontrol	创建 ActiveX 控件对象，建立客户自动化支持
activeserver	创建 ActiveX 自动化服务器
delete	删除 ActiveX 对象
get	从接口获得属性值
invoke	激活(或显示)接口
move	在父窗口上移动对象或改变其大小
propedit	要求控件显示其内建属性页
release	释放接口

续表

函 数	功 能
send	返回事件列表
set	设置接口属性

【例 10-3】运行自动化服务器基本命令。打开 Excel 默认界面，增添工作簿(Workbook)，激活当前页(Worksheet)，实现 MATLAB 与 Excel 之间的数据传输，保存 Excel 数据。

在 MATLAB 中执行如下命令：

```
% 打开一个 Excel 服务器
>> Excel = actxserver('Excel.Application');
>> set(Excel, 'Visible', 1); %开启页面可见
% 加入一个新工作簿句柄
>> Workbooks = Excel.Workbooks;
>> Workbook = invoke(Workbooks, 'Add'); %产生新空白工作簿
% 激活第二张表
>> Sheets = Excel.ActiveWorkBook.Sheets;
>> sheet2 = get(Sheets, 'Item', 2); %
>> invoke(sheet2, 'Activate'); % 激活第二张表
% 激活工作表服务器
>> Activesheet = Excel.Activesheet;
% 把 Mtalab 中的数据输到 Excel 中
>> A = [1 2 4; 3 4 7];
% 矩阵 A 为 2 行 3 列，所以在 Excel 中“A1: C2”工作区可以保存数据
>> ActivesheetRange = get(Activesheet,'Range','A1:C2');
>> set(ActivesheetRange, 'Value', A);
% 取得一个区间，由于保存数据
>> Range = get(Activesheet, 'Range', 'A1:C2');
>> B = Range.value;
% 转换为双精度数据
>> B = reshape([B{:}], size(B));
% 保存工作簿
>> invoke(Workbook, 'SaveAs', 'myfile.xls'); %保存到新工作簿
>> Workbook.Saved = 1;
>> invoke(Workbook, 'Close');
% 退出 Excel
>> invoke(Excel, 'Quit');
% delete(Excel); % 结束整个过程
```

【例 10-4】用二叉树方法计算欧式看涨期权，然后输出到 Excel 中。

在 MATLAB 中执行如下命令：

```
% function binomial()
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  利用 MATLAB 二叉树计算欧式看涨期权并将结果输出到 Excel 中  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s0=52;k=50;r=0.1;T=5/12;dt=1/12;sigma=0.5;class=1;
bb=[s0,k,r,T,dt,sigma];bb=bb';
[Price, Option] = binprice(s0, k, r, T, dt, sigma,class)
aa=cell(6,1);% 定义单元数组
aa(1)={'股价'};aa(2)={'执行价'};aa(3)={'无风险利率'};
aa(4)={'波动率'};aa(5)={'续存期'};aa(6)={'时间步长'};
a1='欧式看涨期权的参数'
a2='欧式看涨期权的二叉树结构'
a3='欧式看涨期权的在二叉树节点的价格'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          激活 Excel 服务器          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Excel = actxserver('Excel.Application');
set(Excel, 'Visible', 1);
Workbooks = Excel.Workbooks;
Workbook = invoke(Workbooks, 'Add');
Sheets = Excel.ActiveWorkBook.Sheets;
sheet1 = get(Sheets, 'Item', 1); %
invoke(sheet1, 'Activate'); % 激活第一张表
Activesheet = Excel.Activesheet;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   把 MATLAB 的数据传递给 Excel        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ActivesheetRange=get(Activesheet,'Range','A1'); %
set(ActivesheetRange, 'Value', a1);
ActivesheetRange=get(Activesheet,'Range','A3:A8');
set(ActivesheetRange, 'Value', aa);
ActivesheetRange=get(Activesheet,'Range','B3:B8');
set(ActivesheetRange, 'Value', bb);
ActivesheetRange=get(Activesheet,'Range','F1');
set(ActivesheetRange, 'Value', a2);
ActivesheetRange=get(Activesheet,'Range','D2:I7');
set(ActivesheetRange,'Value',Price);
ActivesheetRange=get(Activesheet,'Range','E10');
set(ActivesheetRange,'Value',a3);
ActivesheetRange=get(Activesheet,'Range','D11:I16');
set(ActivesheetRange,'Value',Option);
% 保存工作簿
invoke(Workbook, 'SaveAs', 'myfile.xls');
invoke(Excel, 'Quit');
```

```
delete(Excel); % 结束整个过程
```

运行结果如图 10.22 所示。

	A	B	C	D	E	F	G	H
1	欧式看涨期权的参数				欧式看涨期权的二叉树结构			
2			52.00	60.07	69.40	80.18	92.63	107.01
3	股价	52.00	0.00	45.01	52.00	60.07	69.40	80.18
4	执行价	50.00	0.00	0.00	38.96	45.01	52.00	60.07
5	无风险利率	0.10	0.00	0.00	0.00	33.72	38.96	45.01
6	波动率	0.42	0.00	0.00	0.00	0.00	29.19	33.72
7	续存期	0.08	0.00	0.00	0.00	0.00	0.00	25.27
8	时间步长	0.50						
9								
10				欧式看涨期权的在二叉树节点的价格				
11			8.90	14.00	21.27	31.01	43.04	57.01
12			0.00	4.09	7.16	12.16	19.82	30.18
13			0.00	0.00	1.18	2.41	4.92	10.07
14			0.00	0.00	0.00	0.00	0.00	0.00
15			0.00	0.00	0.00	0.00	0.00	0.00
16			0.00	0.00	0.00	0.00	0.00	0.00
17								
18								

图 10.22　利用 ActiveX 将欧式期权二叉树输出到 Excel 中

10.4.2　MATLAB ActiveX 自动化服务器

通过 MATLAB ActiveX 自动化服务器，用户可以在自己的应用程序中执行 MATLAB 命令，并可以与 MATLAB 工作区间进行数据连接。当将 MATLAB 作为服务器使用时，首先要查阅希望使用自动化服务器的应用程序文档，了解如何在控制器中开启自动化服务器。MATLAB ActiveX 对象在系统注册表中的定义名称 ProgID，通常 ProgID 自动化服务器取下面两个名字之一：MATLAB.Application(将启动的 MATLAB 自动化服务器作为共享服务器)和 MATLAB.Application.Single(将启动的 MATLAB 自动化服务器作为专用服务器)。

10.5　MATLAB 与 Access 数据连接

10.5.1　Access 数据库介绍

Access 是 Office 办公套件的一个重要组成部分，刚开始时微软公司是将 Access 单独作为一个产品进行销售，后来微软发现如果将 Access 捆绑在 Office 中一起发售，将带来更加可观的利润，于是第一次将 Access 捆绑到 Office 97 中，成为 Office 套件中的重要一员，现在它已经成为 Office 办公套件中不可缺少的部件，自从 1992 年开始销售以来，Access 已经卖出了超过 6000 万份，现在它已经成为世界上流行的桌面数据库管理系统。

后来微软公司通过大量地改进，将 Access 新版本的功能变得更加强大，不管是处理公司的客户订单数据，管理自己的个人通讯录，还是大量科研数据的记录和处理，人们都可以利用它来解决大量数据的管理工作。

Access 功能这么强，那使用起来会不会很麻烦呢？随着版本升级，Access 使用也变得越来越容易，过去很烦琐的工作现在只需几个很简单的步骤就可以高质量地完成了。

10.5.2 MATLAB 与 Access 数据连接

下面以一个例子说明如何将 Access 中的数据导入到 MATLAB 中，其他数据库如 SQL、Dbase、Foxpro 等数据库也可以类似地和 MATLAB 进行数据连接。本例数据连接是通过 ODBC(Open DataBase Connectivity，开放式数据库连接)完成的。ODBC 是最传统的远程数据库连接方法。我们先建立了一个 Access 文件，文件名为 db1.mdb，如图 10.23 所示。

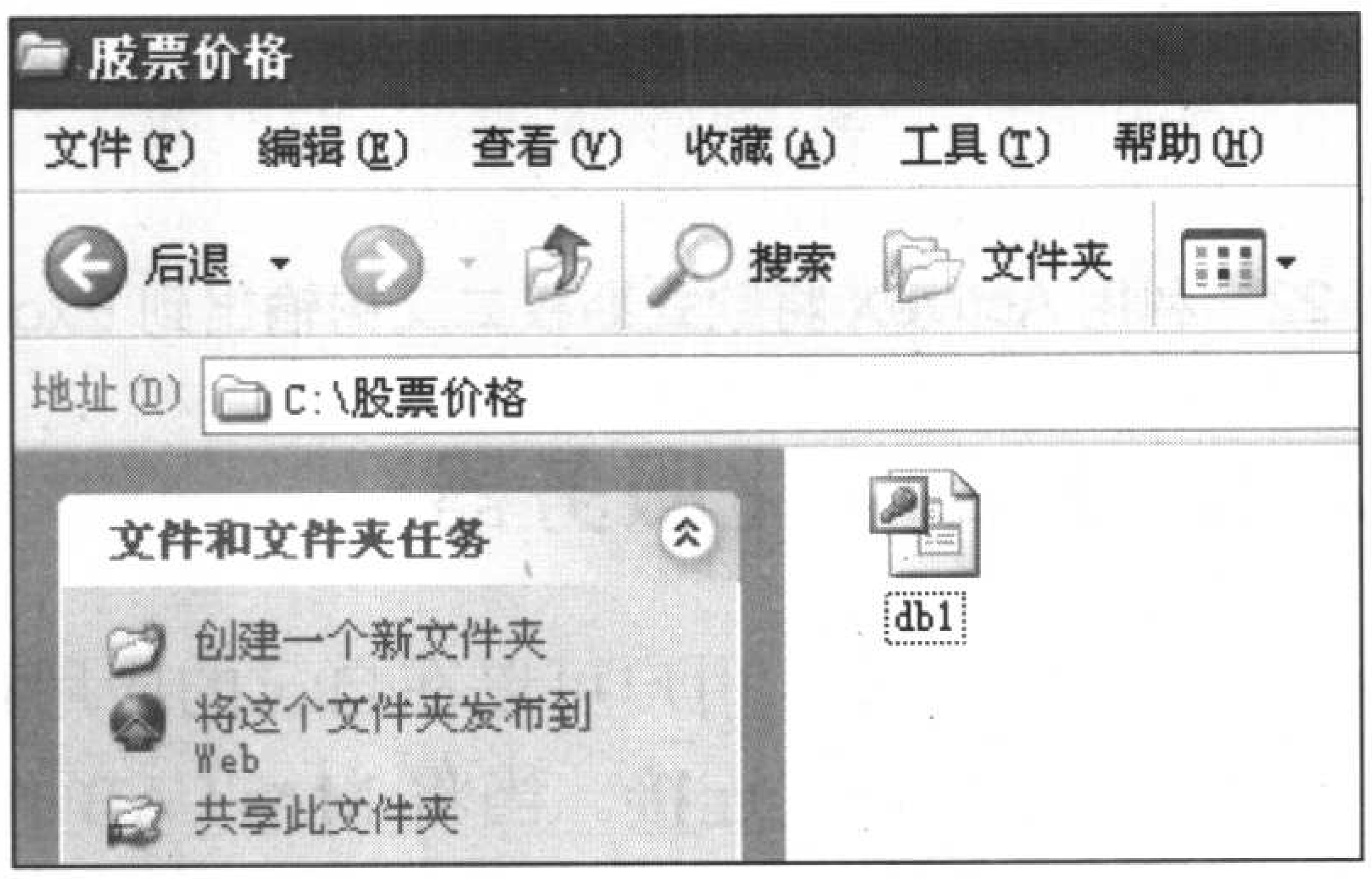


图 10.23 Access 中的文件

下面浏览其内容，如图 10.24 所示。

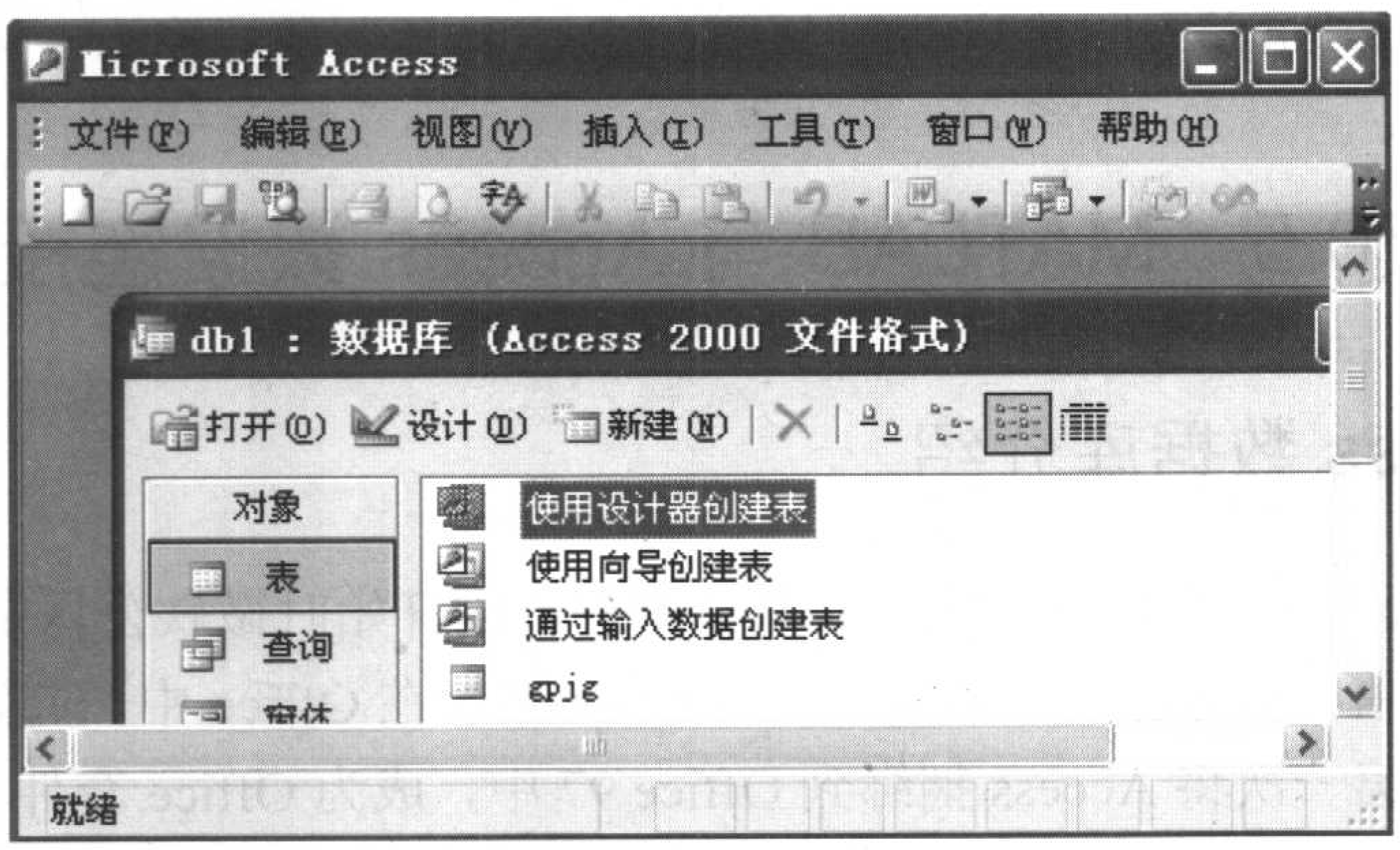


图 10.24 浏览文件内容

文件中只有一个工作簿 gpjg，退出 Access，打开 MATLAB，在 Command 窗口中输入命令，如图 10.25 所示。

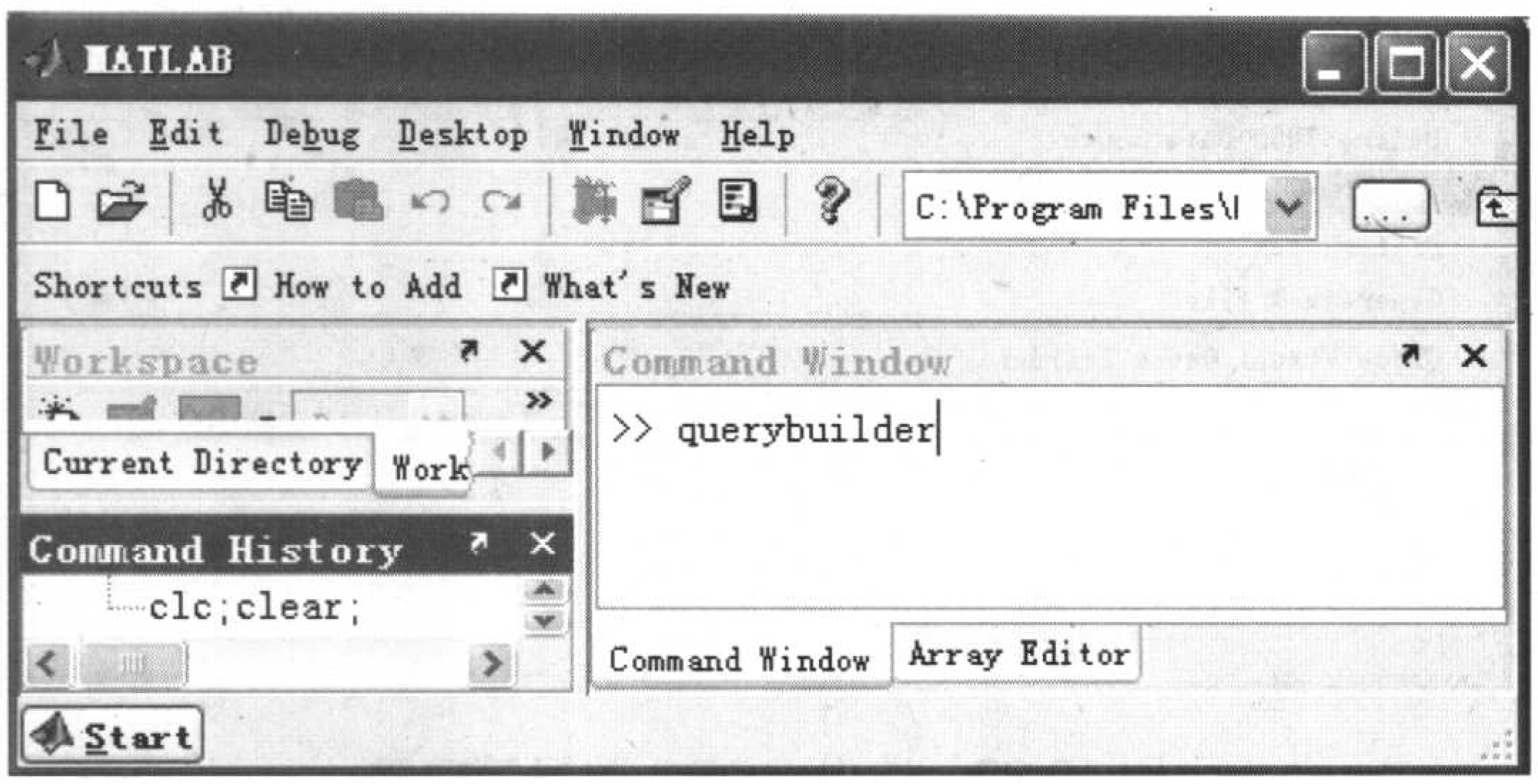


图 10.25　在 MATLAB 窗口中调用数据连接 GUI

弹出如图 10.26 所示的 Visual Query Builder 窗口。

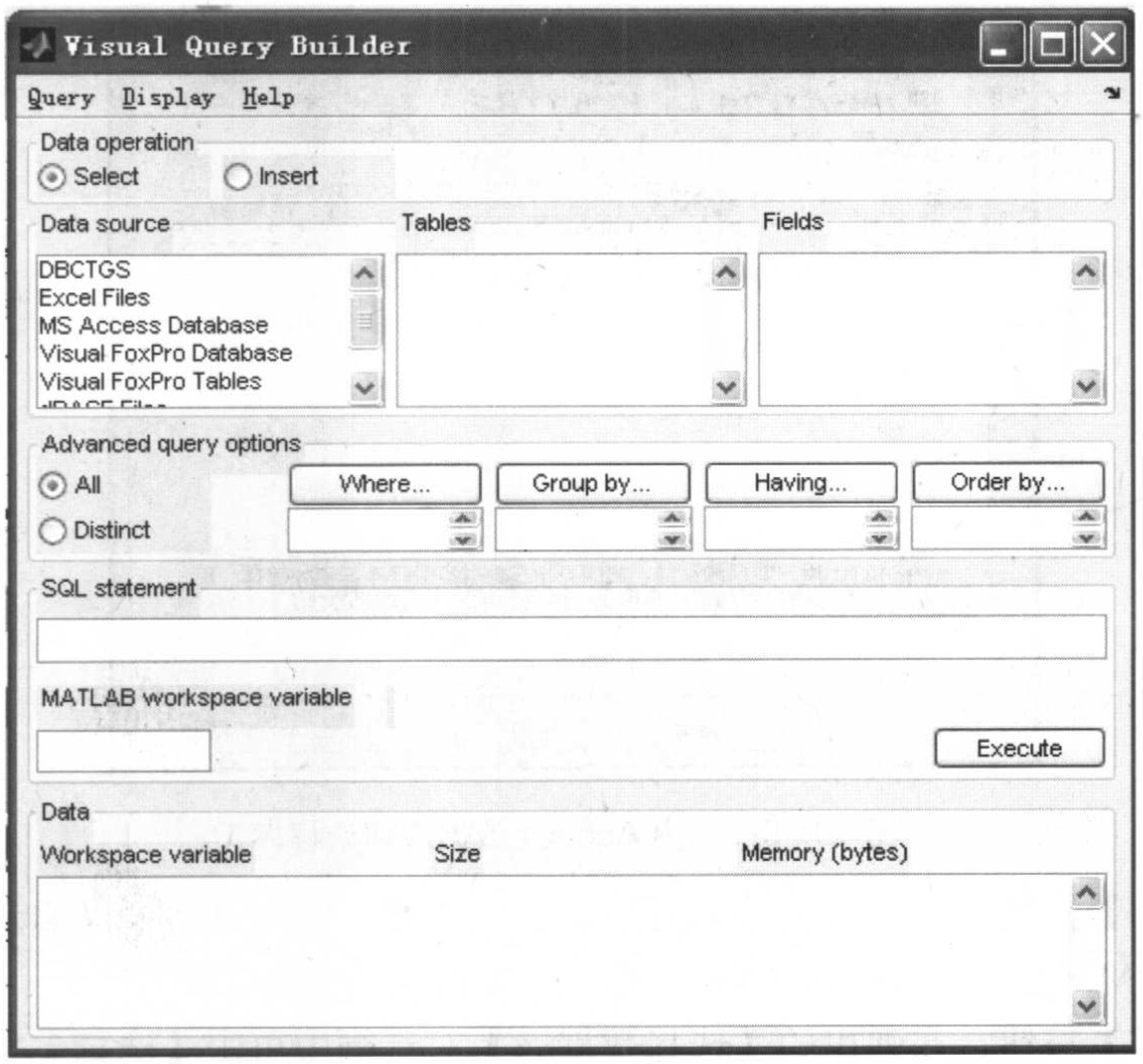


图 10.26　打开 Visual Query Builder 窗口

打开 Query 菜单单击 Define ODBC Data Source 选项，如图 10.27 所示。

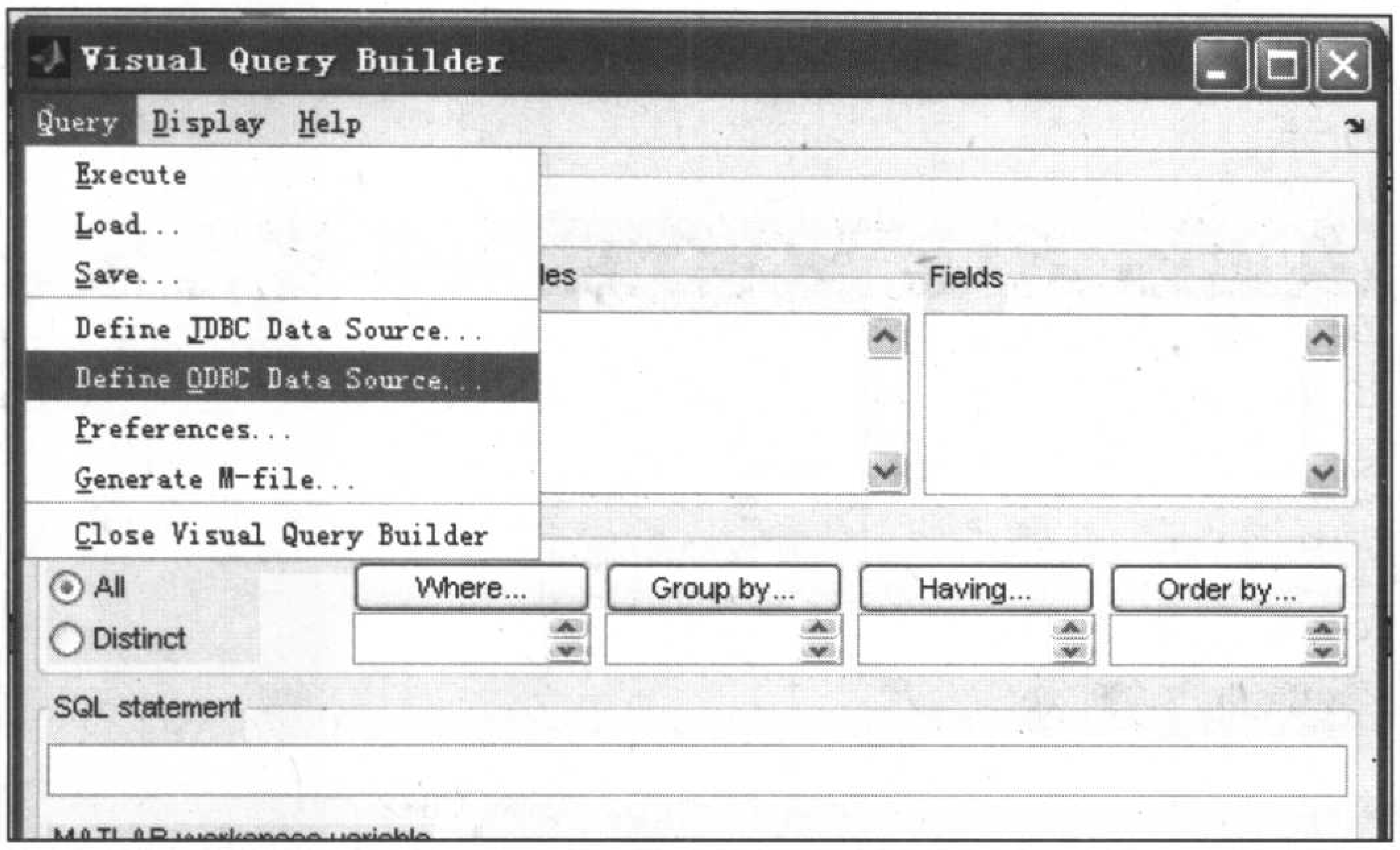


图 10.27　选中 ODBC 数据管理器

选择 Define ODBC Data Sourse 选项，弹出【ODBC 数据源管理器】对话框，如图 10.28 所示，在对话框【用户数据源】列表框中选择 MS Access Database 选项。

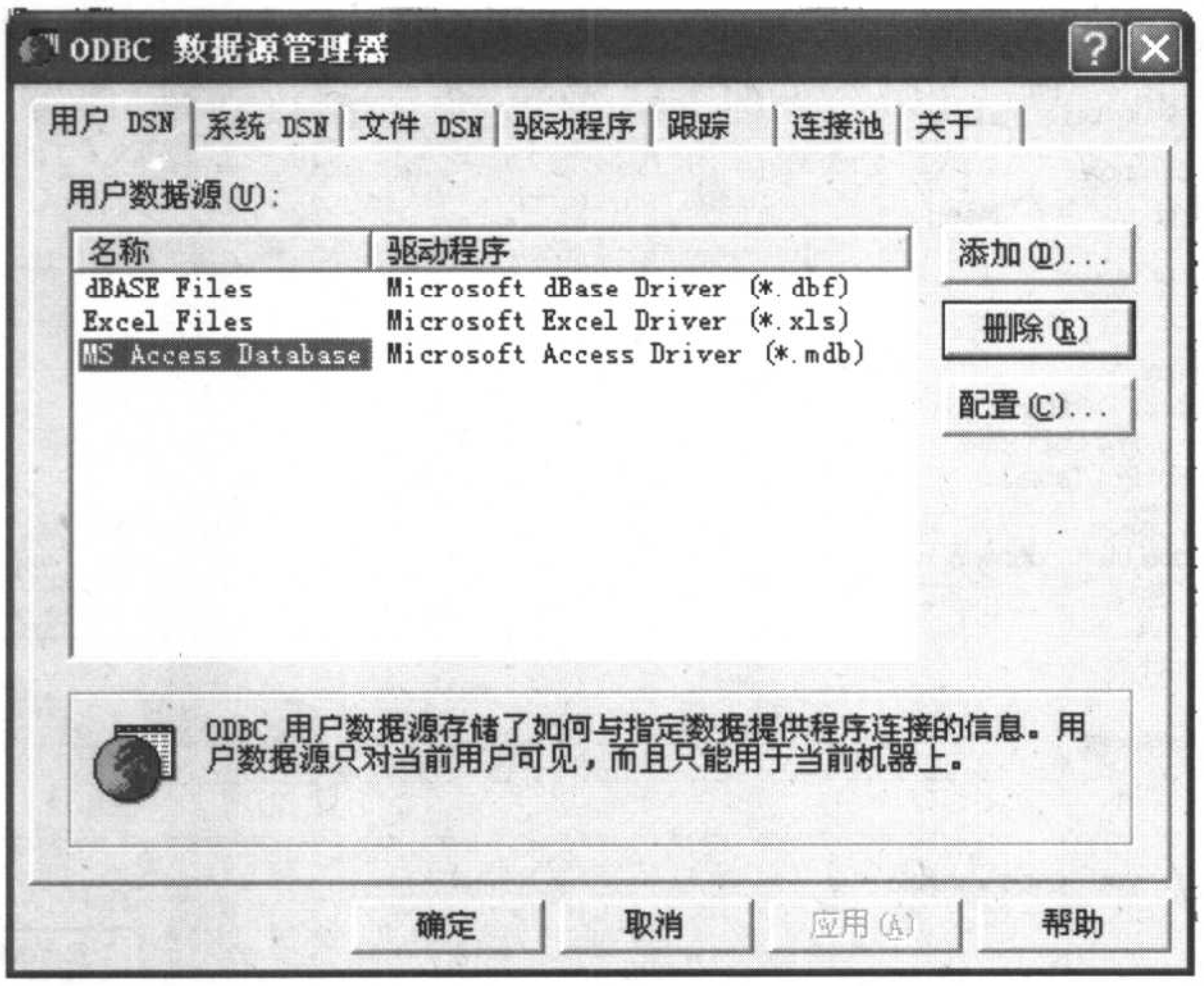


图 10.28　选中 Access 数据库驱动程序

单击【添加】按钮，弹出【创建新数据源】对话框，在对话框的【名称】下拉列表框中选择 Drier do Microsoft Access(*mdb)选项，如图 10.29 所示。

单击【完成】按钮，在弹出的【选择数据库】对话框中单击【选择】按钮，弹出【选择数据库】对话框，根据对话框中的提示，找到“C:\股票价格\db1.mdb”文件，如图 10.30 所示。

单击【确定】按钮，在弹出的如图 10.31 所示的【ODBC Microsoft Access 安装】对话框中输入文件 db1.mdb 中的 gpjg 工作簿。

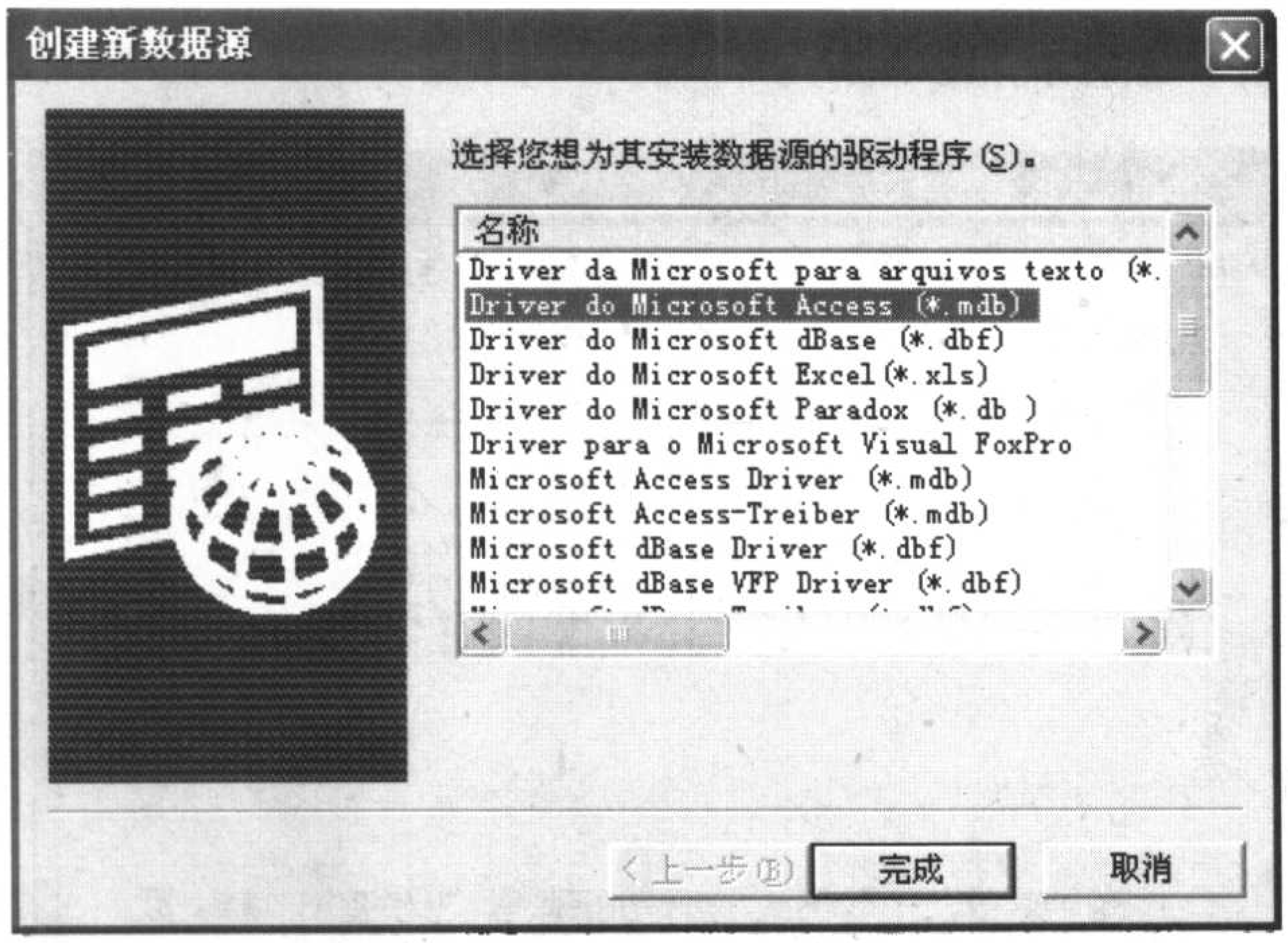


图 10.29　选中为数据安装的驱动程序

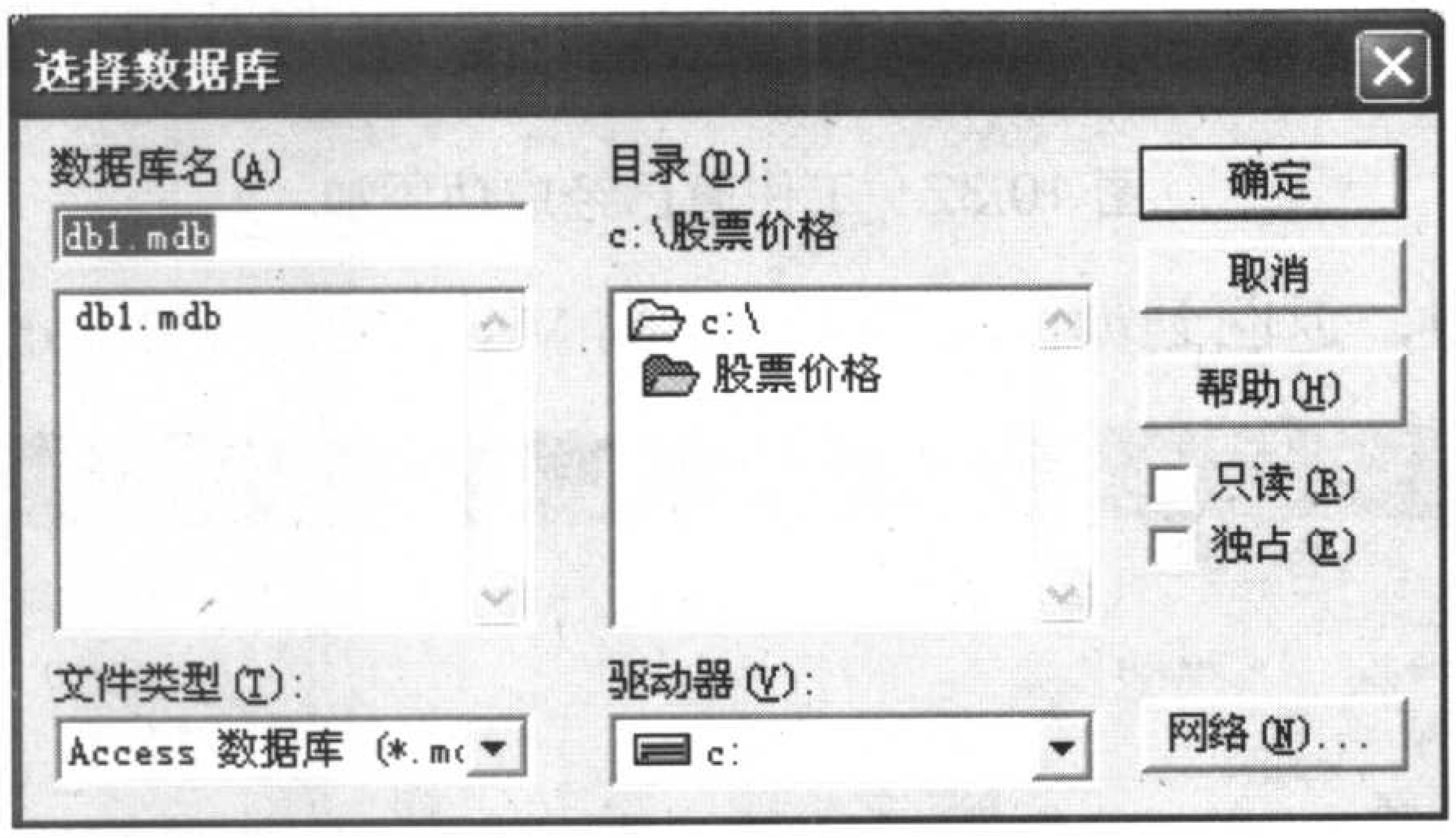


图 10.30　浏览目标文件

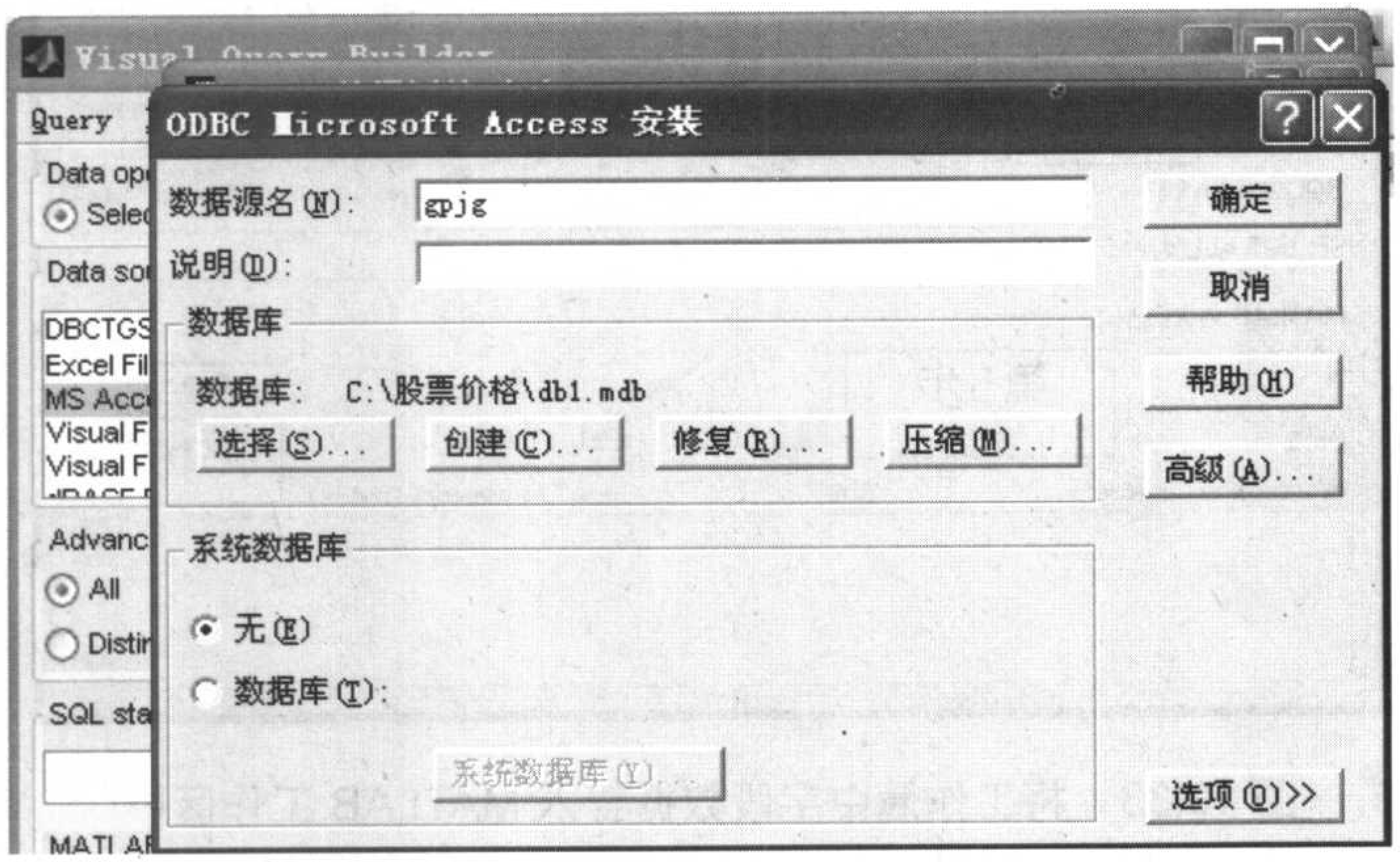


图 10.31　浏览目标文件中的工作簿

单击【确定】按钮返回【ODBC 数据源管理器】对话框，可以看到 ODBC 数据管理器

中加载了 gpjg 工作簿，如图 10.32 所示。

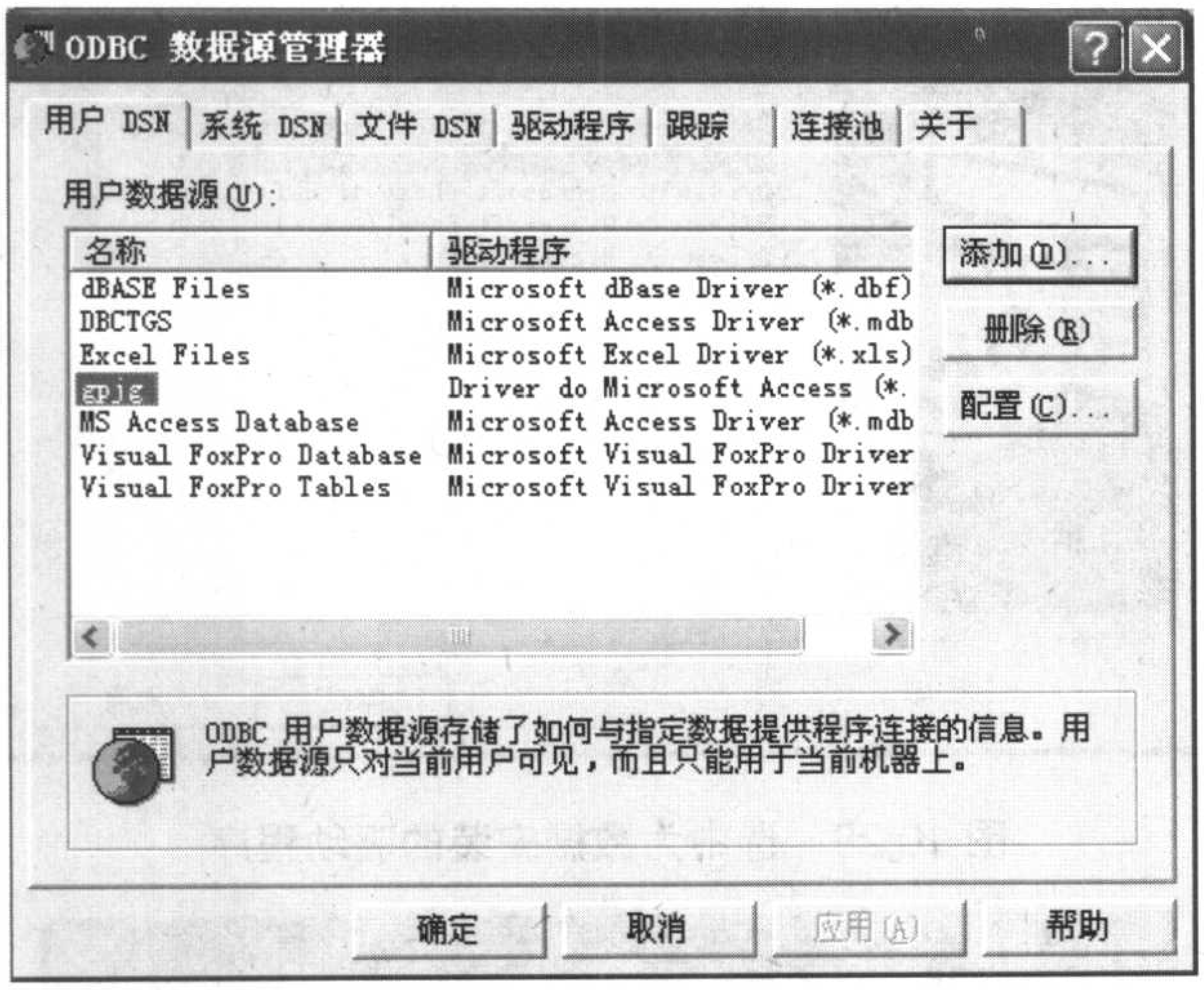


图 10.32 工作簿已经成功添加

单击【确定】按钮，返回到如图 10.33 所示的 Visual Query Builder 对话框。

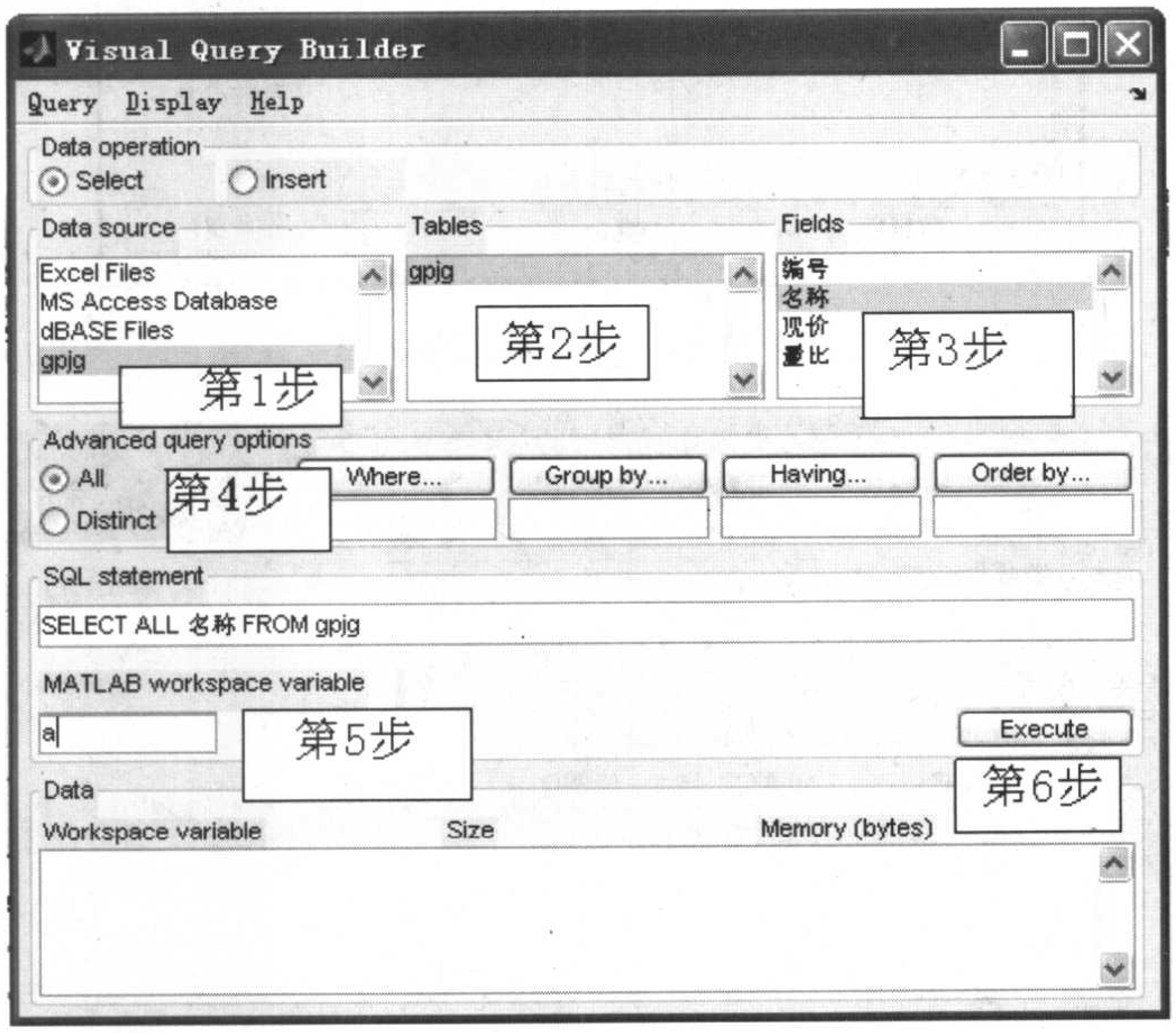


图 10.33 将工作簿中字段数据导入 MATLAB 工作区中

按照图 10.33 的次序分别单击相关按钮，下面介绍每一步的内容。

第 1 步：在 Data Source 列表框中选中 gpjg 工作簿作为数据来源。

第 2 步：在 Table 列表框中选中 gpjg 作为当前工作簿。

第 3 步：Fields 列表框中字段名，我们选择现价。

第 4 步：对字段名下的数据进行选择，这里选择 All，将现价字段下的数据全部选中。

第 5 步：填入在 MATLAB 中的变量 a，用以保存上面选中的数据。

第 6 步：单击 Excute 按钮，这样字段“现价”下的数据全部导入 MATLAB 的变量 a 中。

浏览 MATLAB 工作区可以看到变量 a，注意 a 是单元数据，如图 10.34 所示。

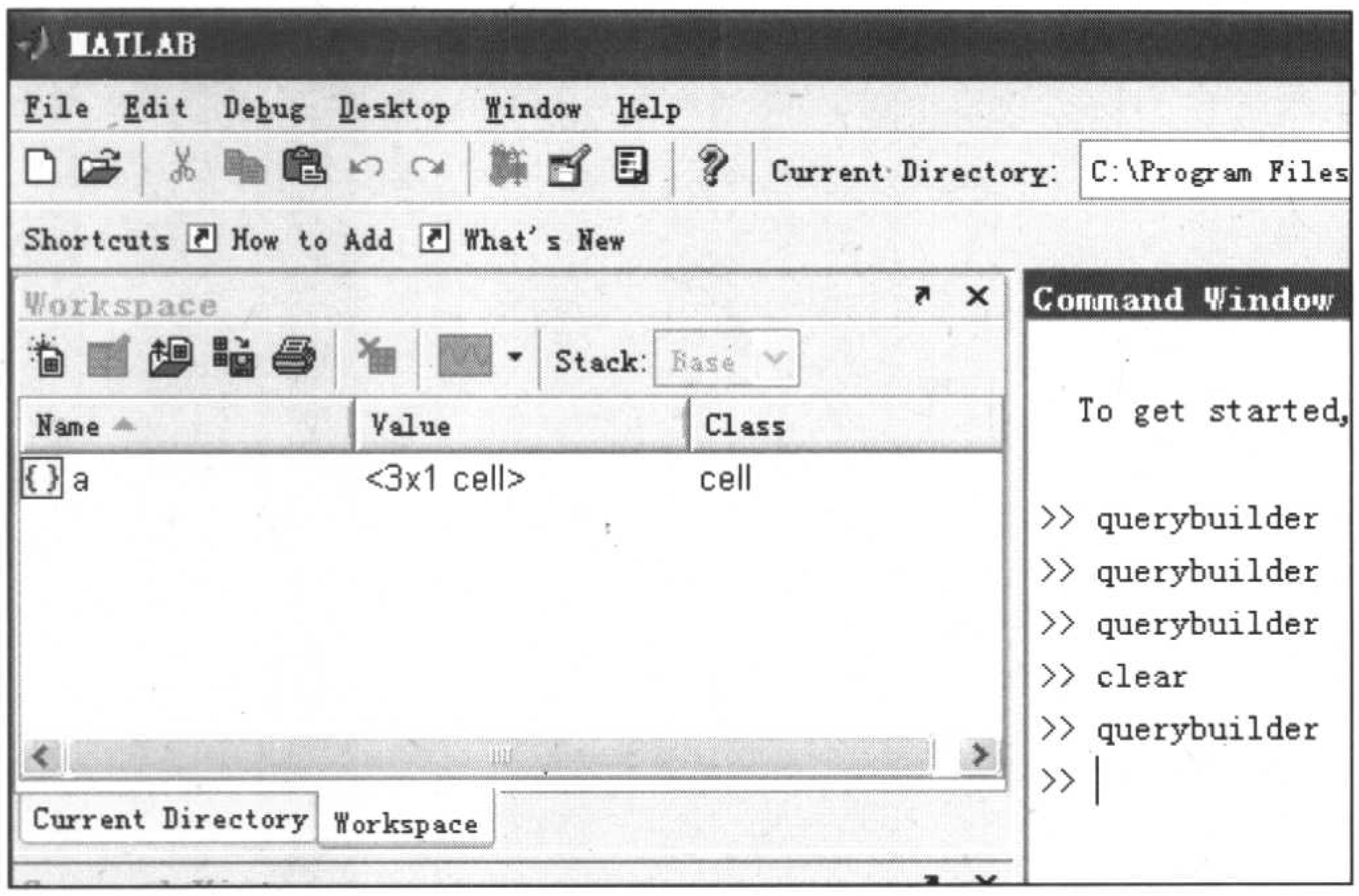


图 10.34　浏览工作区中的变量

浏览工作区中的变量 a，发现变量 a 中的内容就是字段“现价”下的数据，这样就完成了数据的导入。

思　考　题

1. 利用 MATLAB 自带的连接工具，试从 Yahoo 网站上获得微软公司股票的价格并和用友软件收益率与 Var 进行比较。

2. 利用 MATLAB 自带的连接工具，从 Yahoo 网站上获得 10 年期美国国债的收益率，分析和股票市场的互动性。

3. 利用 MATLAB 自带的连接工具，从相关网站获得 Nasdaq 与 S&P500，分析其和上证指数的相关性。

前言
目录