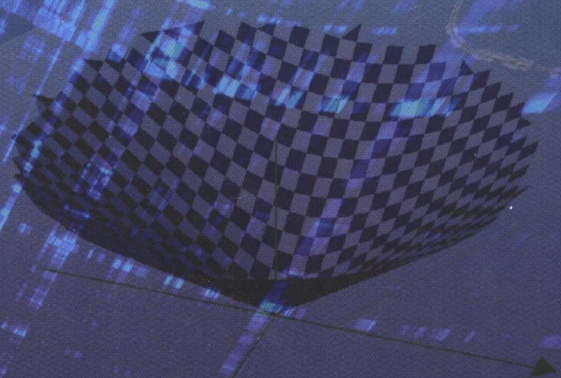
普通高等教育“十五”国家级规划教材

数学实验

刘琼荪　龚　劬　何中市　傅　鹂　任善强　编著



高等教育出版社

ISBN 7-04-014409-3

9 787040 144093 >

定价 18.70 元

普通高等教育“十五”国家级规划教材

数学实验

刘琼荪　龚劬　何中市　傅鹂　任善强　编著

高等教育出版社

内容简介

本书是普通高等教育“十五”国家级规划教材，书中通过“问题→数学模型→数学方法→软件求解→思考分析→实际操练”这种有效的教学模式，让读者充分体验数学实验的奥妙。全书共十三章，涵盖数值计算、数理统计、优化方法和图论网络等具有极大实用价值的数学模型类别。本书的特点是通俗易懂，趣味性强，遍布书中的“想”、“做”、“注意”和“提示”四种图标，令人耳目一新，兴致勃勃。本书在内容和编排上的精心设计，极大地帮助读者培养观察问题、分析问题、解决问题的实际能力，引导读者达到一种全新的境界。

本书可作为高等院校理工科各专业本科生、研究生、教师以及各行业工程技术人员的教材或参考书，也适合于作为上述各类人士的自学读本。

图书在版编目(CIP)数据

数学实验/刘琼荪等编著. —北京：高等教育出版社，2004.7

普通高等教育“十五”国家级规划教材

ISBN 7-04-014409-3

Ⅰ.数… Ⅱ.刘… Ⅲ.高等数学-实验-高等学校-教材 Ⅳ.O13-33

中国版本图书馆 CIP 数据核字(2004)第 046662 号

出版发行	高等教育出版社	**购书热线**	010-64054588
社　　址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
总　　机	010-82028899		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京星月印刷厂		
开　　本	787×960　1/16	**版　　次**	2004 年 7 月第 1 版
印　　张	16	**印　　次**	2004 年 7 月第 1 次印刷
字　　数	290 000	**定　　价**	18.70 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

前　言

这本书名为“数学实验”，但是书中并不讨论“数学实验”的定义，不回答“数学实验”是什么的问题。

那有什么要紧的吗？你做过物理实验、化学实验吗？做过！好！那请你现在回答物理实验或者化学实验的定义……请别想了，说不准定义，并没有妨碍你物理实验全优，化学实验及格。

“数学实验”你知道吗，听说过吗？知道！很好，那么这“前言”的下面部分你就更可以立即闭上眼睛跳过了！

既然根本没有听说过“数学实验”，那么建议你就像对待物理实验、化学实验一样，不管它是什么，做了再说！

做物理实验的时候，你可能会用到滑轮、弹簧、电池组电路什么的，根据实验课题的不同需要一大堆东西；做化学实验的时候，你可能会用到试管、烧杯、各种原料等等又是一大堆东西。做“数学实验”呢？你只需要计算机！简洁多了吧！

这是当代信息技术、计算技术的蓬勃发展所带来的新境界！

因此，当今越来越趋向于尽可能用“数学实验”这种虚拟的实验代替常规的真实的实验，如物理实验、化学实验、生物实验、医学实验，甚至有可能渗透到心理学、社会学实验等等。现实中的核爆模拟实验以及幻想中的《黑客帝国》就是这种趋势的登峰造极。

“数学实验”应该是不同学科领域新的共同手段。

“数学实验”的一个首要环节是“数学建模”，然后是数学方法/算法。正因为如此，读完本书后，如果你觉得这书更像一本“数学建模”，也不足为奇。其实，“数学实验”的内涵人们还没有统一的认识，这也是此书不给出“数学实验”定义的另一个原因。

还是留给你自己去探索、体验，去认识一个新天地吧！

本书第1、6、7、8章由刘琼荪编写，第11、12、13章及附录A由龚劬编写，第2、3、4、5章由何中市编写，第9、10章由傅鹂编写，任善强对全书进行了全面的审阅和修改，并提出了许多合理化的建议。

全体作者

2004年1月

郑重声明

策划编辑　李艳馥
责任编辑　蒋　青
封面设计　李卫青
责任绘图　尹　莉
版式设计　马静如
责任校对　王　雨
责任印制　孔　源

目　录

第 1 章

如何用数学解决实际问题

数学是一种很美的语言. 伽利略曾说:“自然界的伟大的书是用数学语言写成的”. 希望你能从一些数学的探索和数学的应用中获得乐趣.

——作者

数学是与人类文明并存共同发展的,它是一种语言,是一种交流和认识世界的方法,是一种将自然、社会运动现象法则化、简约化的工具. 数学作为一门研究现实世界数量关系和空间形式的科学,在它产生和发展的历史长河中,一直是和人们生活的实际需要密切相关的. 随着计算机的发展,数学渗入各行各业,得到了广泛的应用,直接为社会创造价值,已经成为一种关键的、普遍的、适用的技术. 历史已经证明,国家的繁荣昌盛,关键在于高新技术的发达和经济管理的高效率. 高新技术的基础是应用科学,而应用科学的基础是数学. 数学给予人们的不只是知识,更重要的是能力,这种能力包括直观思维、逻辑思维、精确计算和准确判断等. 因此数学在提高民族的科学文化素质中处于极为重要的地位,掌握数学的概念、计算和解决问题的能力对一个真正有文化的人来说是至关重要的.

§1.1 什么是数学模型

模型是人们十分熟悉的概念,如玩具电动模型、机械运动模型等是人们熟悉的实物模型. 而数学模型(Mathematical Model)还没有一个人们公认的确切的定义,如相关信息资源 2 所述:对于一个特定对象,为了一个特定的目标,根据特有的内在规律,做出一些必要的简化假设,运用适当的数学工具,得到的一个数学结构. 这里的“特定对象”是为了解决某个实际问题而提出的;“特定的目的”是指当研究一个特定对象时要达到的目的,如分析、预测、控制、决策等;“数学结构”可以是数学关系式,也可以是程序、图、表等. 数学建模(Mathematical Modeling)则是指建立数学模型的全过程,包括问题分析、模型建立、求解、结果检验和

应用等.

为了更好地理解数学模型和数学建模的概念,我们列举一个实际建模例子.

例 1.1 新产品的销售量变化规律.

一种新产品进入市场以后,产品的销售量一般会经过"先增后逐渐平稳略有下降"的一个过程,这称为产品的生命周期.怎样使用数学模型来描述新产品的销售量的变化过程呢?

1. 问题分析

当一个新产品进入市场时,其有关信息的传播有两个途径:一是经营者或厂家进行广泛的广告宣传,消费者亲眼看到广告或亲耳听到消息,这是来自消费者以外的信息;二是当一部分消费者购买了该产品之后,经过使用对该产品有了认识,向其周围的人们进行宣传,这称之为来自消费者内部的信息.正是这两方面的信息促使消费者去购买该商品.

2. 合理假设

- 设 N 为潜在的消费者人数,$x(t)$ 为 t 时刻购买了该产品的人数,并且认为变量 $x(t)$ 随时间变化是连续的;
- 购买者增量 Δx 由两部分组成,一是由外部信息导致消费者增加,其增量记为 Δx_1.二是由内部信息导致消费者增加,记为 Δx_2;
- 由外部信息导致消费者增量与未购买者人数成正比,即

$$\Delta x_1 = k_1(N - x(t))\Delta t \quad (k_1 > 0 \text{ 为比例系数}); \tag{1.1}$$

- 由内部信息导致购买者增量与已购买者人数和未购买者人数之积成正比,即

$$\Delta x_2 = k_2 x(t) \cdot (N - x(t))\Delta t \quad (k_2 > 0 \text{ 为比例系数}). \tag{1.2}$$

3. 建立数学模型

由上述(1.1)(1.2)式,再由假设知,Δx 由 Δx_1 和 Δx_2 两部分组成,得到:

$$\Delta x = k_1(N - x(t))\Delta t + k_2 x(t)(N - x(t))\Delta t. \tag{1.3}$$

(1.3)式两端同除以 Δt,并令 $\Delta t \to 0$,得到微分方程模型如下:

$$\frac{dx}{dt} = (N - x(t))(k_1 + k_2 x(t)),\text{并且},x(0) = 0. \tag{1.4}$$

4. 模型求解

为分析产品销售量 $x(t)$ 随时间 t 的变化情况,对微分方程(1.4)求解,得:

$$x(t) = N \cdot \frac{1 - e^{-(k_1 + k_2 N)t}}{1 + \frac{k_2 N}{k_1} e^{-(k_1 + k_2 N)t}}. \tag{1.5}$$

该问题解含有未知参数 k_1, k_2, N,如何确定它们呢?其方法是采样,收集某产品从推向市场以来其销售情况的统计数据,根据数据分析,用最小二乘法将模型中

的未知参数辨识出来. 例如,使用某产品一段时期的销售统计数据,对模型(1.4)中的参数进行参数辨识,得到 $k_1 = 0.02, k_2 = 0.035, N = 10^4$,通过计算,具体分析出某产品的销售量与时间的关系如下图所示:

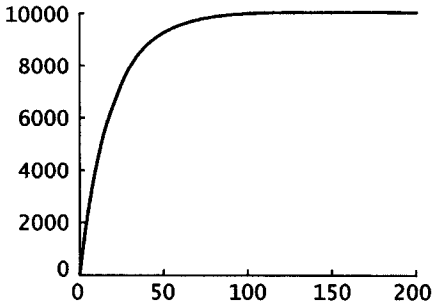


图 1.1 产品销售量曲线

图 1.1 的曲线基本反映出产品的生命周期——先增后逐渐平稳略有下降.

5. 模型检验

通过对实际问题分析,我们建立了数学模型(1.4),并得到解曲线(1.5)或图形 1.1,它们是否能应用于实际? 能否预测产品在今后一段时期内的销售情况? 关键取决于对模型的检验.

我们使用某产品一段时期的销售统计数据,将这些实测数据代入模型(1.5)中,如果实测数据与理论数据(模型中对应值)之差的平方和(定义为误差平方和)很小,则称该模型通过了检验. 如图 1.2 所示,符号"。"表示实测数据,而曲线 $x(t)$ 是理论曲线. 实际观测数据在理论曲线附近波动,显而易见,其波动幅度较小. 再计算误差平方和的值,如果该值相对于原始数据比较小,我们称该模型通过了检验,这时可以运用该模型进行预测了

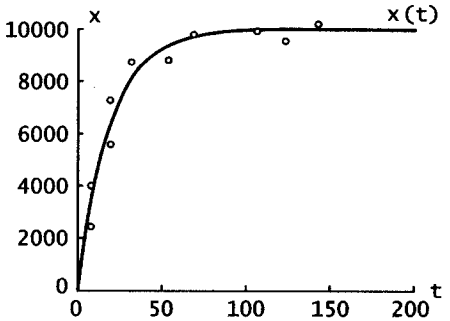


图 1.2 产品销售量曲线

以上反映了一个数学建模的全过程.

§1.2 数学模型的分类

为了有效地、系统地研究数学建模方法,需要将数学模型加以分类,在各类模型中找出数学建模方法的共性,便于初学者学习和尽快掌握数学建模方法. 一般地,对数学模型按两种方法进行分类,一是按应用领域划分,如人口模型、交通模型、经济预测模型、数量经济模型、金融模型、生态模型、环境模型、企业规划模型等等;二是按数学方法进行划分,如方程模型、微分方程模型、图论模型、网络模型、概率模型、统计模型、优化模型、最优控制模型等等. 或者更笼统地划分,分成连续系统模型和离散系统模型、确定性模型和随机性模型等几大类. 从理论研究的角度看,对数学模型的分类是有必要的. 但不管数学模型按什么方式进行分类,建立数学模型的主要目的是能有效地解决实际问题和预测未来.

学习数学模型或数学建模的第一步关键是掌握数学建模的基本方法和步骤.

§1.3 数学建模的基本方法和步骤

我们可能面对的实际问题是多种多样、错综复杂的,如果解决实际问题的目的不同,分析问题和解决问题的方法也不同,建立的数学模型也不相同. 一般地,解决实际问题的数学建模方法大致可分为两类:机理分析法和测试分析法. 所谓机理分析是根据对客观事物特性的认识,通过对现实对象各个因素之间的因果关系的分析,找出反映内部机理的数量规律,建立的数学模型常有明确的物理意义. 例如,试建立细菌繁殖过程的数学模型. 该问题显然涉及几个因素,一是需要预测任意时刻 t 某种细菌的数量 $A(t)$,即细菌的数量与时间构成的函数关系;二是细菌的繁殖速度 $v(t)$ 也与时间有关,同时它与细菌的数量成一定的关系. 由机理分析法,可以直接分析得到如下关系:$v(t_0)=\lim\limits_{t\to t_0}\dfrac{A(t)-A(t_0)}{t-t_0}$. 机理分析法的特点就是某实际问题涉及的各个因素之间有比较明确的物理意义下的因果关系. 但对于大多数实际问题,要认识其内部机理是很困难的,甚至没法确定研究对象与哪些因素有关,这时需要用到第二种建模方法——测试分析法. 例如,某种疾病的诊断问题,首先需要分析该疾病有什么样的症状,该疾病与各种症状具有什么关系呢? 有时很难表达成一个数学关系式. 但可以收集患者的各种数据,

通过数据分析建立该疾病与症状之间的关系模型，从而达到正确诊断该疾病的目的. 因此，测试分析法的特点是通过对系统输出的测试来认识系统的输入－输出规律，建立尽可能与这一规律相吻合的数学模型. 另外，许多实际问题也常常将两种方法结合起来，用机理分析法建立模型的结构，用测试分析法确定模型的参数.

数学建模要经过哪些步骤并没有固定的模式，建模的步骤往往与问题的性质和建模的目的有关. 我们用一个简单的示意图来描述，如图 1.3.

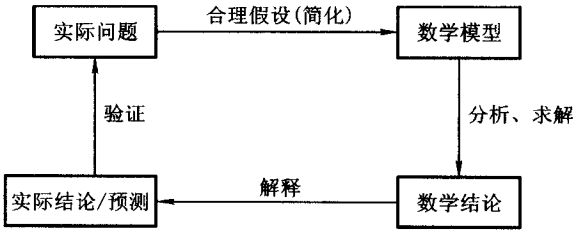


图 1.3　数学建模过程的示意图

上述步骤只是数学建模过程的一个大致地描述，实际建模时可以灵活应用. 如例 1.1，为了分析某产品销售量的变化规律，其数学建模的过程经过了分析→假设→模型→求解→检验→应用. 并且在建立数学模型的过程中，既用到了机理分析法，又用到了测试分析法.

数学建模是以解决实际问题为对象，并不局限于用一种数学方法，也不拘泥于使用哪一种计算软件工具，在建模过程中提倡百花齐放、开阔思路. 常常对于同一问题有多种建模方法，较难分辨优劣. 另一方面，应该尽可能用简明、巧妙的方法完成数学建模，使模型不仅有很好的应用价值，亦有很好的推广价值和理论价值.

在建模的过程中还应当注意模型的解释和检验. 因为数学模型往往只是对现实对象的某种近似，模型是否成功，是否能反映所研究的实际问题，需要通过对模型的解释检验才能发现其合理性，才能达到应用的目的. 如果发现用模型的解去解释实际问题尚有一定的距离，则需要修改假设和模型，使模型更加完善，更能反映实际情况.

总之，数学建模需要有对现实对象的敏锐的洞察力，有对问题分析的高度的抽象力，有对数学工具的熟练的把握力，再加上不时迸发的创造力. 有了这些能力，只要坚持数学建模实践，就一定能够提高数学建模的水平.

更多的相关信息资源

1 Frank R. Giordano, Maurice D. Weir and William P. Fox. *A First Course in Mathematical Modeling*. 影印版. 北京:机械工业出版社, Cole, a division of Thomson Learning, Inc,2003

2 叶其孝. 大学生数学建模竞赛辅导教材. 长沙:湖南教育出版社,1993

3 吴翊,吴孟达,成礼智. 数学建模的理论与实践. 长沙:国防科技大学出版社,1999

4 傅鹂,龚劬,刘琼荪,何中市. 数学实验. 北京:科学出版社,2000

5 萧树铁,姜启源,何青,高立. 数学实验. 北京:高等教育出版社,1999

6 周义仓,赫孝良. 数学建模实验. 西安:西安交通大学出版社,1999

7 潭永基,俞文鉞. 数学模型. 上海:复旦大学出版社,1996

8 么焕民,孙秀梅,孟凡友. 数学建模(上、下册). 哈尔滨:哈尔滨工业大学出版社,2003

9 唐焕文,贺明峰. 数学模型引论. 北京:高等教育出版社,2001

10 王守愚. 思维与创造. 北京:气象出版社,2000

第 2 章

飞机如何定价

——方程求解

很多的科学问题都可以转化为数学问题，很多的数学问题都可以转化为解方程问题，可见方程求解之重要. 无论在理论分析上还是在计算方法上，方程求解都非常复杂，然而其思想和所用的工具却十分简单：迭代、数列和图形.

——作者

§2.1 竞争中的飞机制造业

在近 90 年的历史中，波音这个世界最大的飞机制造商已经有三次因喷气机设计的创新而迎来了它的巨大飞跃，波音 707 是第一架横跨大西洋的喷气机；波音 747 是第一架巨型的喷气式客机；而波音 777 是第一架长途旅行双发动机的客机. 为使波音公司在与其竞争对手空中客车公司的角逐中继续保持领先地位，工程师们正在设计一种全新的喷气客机——波音 7E7，该飞机的设计目标是成为世界飞机市场上最有效率和最为经济的机型.

1970 年法、德两国分别成立了空中客车公司，开始了漫长的向波音挑战的旅程. 而后，1971 年西班牙加入，1979 年英国也加入. 在此后长达 25 年的时间里，尽管四国政府不断给予财政补贴，空中客车公司工程技术人员全力以赴，新型飞机一架接一架上天，但在波音公司竞争性定价策略的压力下，公司仍一直亏损. 直到 1 300 多架空中客车交付使用后，波音公司才正视空中客车公司的存在，不再继续使用低价倾销策略，逼迫空中客车公司退出竞争. 因为那时波音公司的账单上首次出现了亏损，它不得不意识到有欧洲各国政府力量支持的空中客车公司是不可战胜的. 于是，两家公司握手言和，共同提高飞机价格，共同垄断市场.

波音和空中客车这两家厂商的最大客户，全球最大的高科技商用飞机租赁公司——国际租赁金融公司的主席斯蒂文·艾德华兹说：“波音应该如空中客车一样，在定价和技术上跃过对手”.

价格:作为市场调节的杠杆,厂家采用何种定价策略,将是决定其生死存亡的重要因素.从波音公司与空中客车公司的竞争过程中,一味的低价是行不通的,竞争与联合、由市场调节的科学合理的定价策略成为必然.我们的问题是:某飞机制造商对其研发的一种新型客机如何定价?

§2.2 飞机的定价策略

传统观念中,价格被认为是管理者决定的变量,而销售量是市场决定的变量.另一种观念与此相反,把销售量作为管理者决定的变量,而价格是由市场决定的变量.事实上,他们都是为了相同的目标——利润最大.

2.2.1 问题分析

飞机定价策略涉及诸多因素,这里主要考虑以下因素:

飞机的制造成本、公司的生产能力、飞机的销售数量与价格、竞争对手的行为与市场占有率等因素.

2.2.2 假设及模型

价格记为 p,根据实际情况,飞机的几大制造商共同垄断市场,并在价格上形成联合,具体假设如下:

1) 单一型号:为了研究方便,假设只有一种型号飞机;

2) 价格决定总销售量:设该型号飞机全球销售量为 N. N 应该受到诸多因素的影响,假设其中价格是最主要的因素.根据市场历史的销售规律和需求曲线,假设该公司销售部门预测得到

$$N = N(p) = -78p^2 + 655p + 125;$$

3) 市场占有率:既然在价格上形成联合,即几大公司在价格的变化是同步的,因此,不同定价不会影响各自的市场占有率(但是会影响飞机的市场需求总量).假设市场占有率是常数,记为 h;

4) 飞机生产能力:假设公司具有足够生产能力,能够满足订单要求,销售量记为 x.既然可以预测该型号飞机全球销售量,结合公司的市场占有率,可以得到

$$x = h \times N(p);$$

5) 制造成本:根据产品分析部门的估计,制造成本为

$$C(x) = 50 + 1.5x + 8x^{\frac{3}{4}};$$

6）利润：假设利润等于销售收入减去成本，利润函数为

$$R(p)=px-C(x);$$

7）最佳定价策略：利润 $R(p)$ 最大.

由以上分析及假设得到波音公司飞机最佳定价策略的数学模型如下：

$$\mathrm{Max}R(p)=px-C(x),$$

其中 $N(p)=-78p^2+655p+125$，$x=h\times N(p)$，$C(x)=50+1.5x+8x^{\frac{3}{4}}$，$p,x,N\geqslant 0$. 即：

$$\mathrm{Max}\ R(p)=(p-1.5)h\times(-78p^2+655p+125)-50-8h^{\frac{3}{4}}\times$$
$$(-78p^2+655p+125)^{\frac{3}{4}}.$$

这是一个典型的无约束非线性优化模型，可以转化为求目标函数 $R(p)$ 的驻点，也就是其导函数 $R'(p)$ 的零点：

$$R'(p)=0,$$

即

$$h\times(-78p^2+655p+125)+(p-1.5)h\times(-156p+655)-$$
$$6h^{\frac{3}{4}}\times(-78p^2+655p+125)^{-\frac{1}{4}}(-156p+655)=0$$

因此，模型的求解转化为非线性方程的求解.

§2.3 方程数值求解方法

无论是探索未知世界、还是改造已知世界，都涉及变量的取值及优化，从而几乎无一不归结为解方程. 方程是很多工程和科学工作的发动机. 若干世纪以来，工程师和科学家们花了大量的时间用于求解方程：线性的或非线性的，单个方程或方程组的，代数的或超越的等等. 求解方法主要有寻求精确解的解析法、近似解的数值解法.

2.3.1 非线性方程

方程分为代数（多项式函数）方程和超越方程，一次代数方程称为线性方程，其余称为非线性方程. 从中学到大学，对数学课程来讲，解方程可谓是家常便饭. 对于解方程，我们掌握了多少呢？

二次方程的求根公式能倒背如流，三次方程求解公式呢？四次方程有求根公式吗？如果有，你又能够记住多少？

注意　早在19世纪初，伽罗瓦等人就已证明了高于四次的代数方程不存在类似于二次方程的求根公式.

在数学的教科书上，虽然介绍了方程求解的因式分解法，但问题远不止这些. 对一般非线性方程，是否有一套行之有效的求解方法？

考虑一般方程 $f(x)=0$，可能难以甚至根本无法找到精确的解析解，但是我们可以去找近似解，只要满足我们期望的精度要求. 本章的目的就在于探讨能够求解任意一个方程全部解（根）或者部分指定范围的近似解的数值方法.

2.3.2　非线性方程求解数值方法

考虑求方程 $f(x)=0$ 的解.

试做　对于方程 $\tan x - x = 0$，试求其非负实数根，并对所得到的结果进行分析.

注意　上述问题中，考虑将方程 $\tan x - x = 0$ 换为 $e^{-\tan^2 x} = 0$，结论如何？

通过对以上问题的计算实践，可以发现：单一方法，比如迭代法求解方程，在方程有根的情况下，迭代算法可能找到根，也可能找不到根；而在方程无解的情况下，可能找出了“根”（增根）. 因此，对于方程求解，常常需要结合具体方程（函数）的性质，从解析的、几何图形的、数值的角度加以分析，需要结合多种不同求解方法.

1. 图形放大法

由于计算机的广泛应用，可以非常方便地作出函数 $f(x)$ 的图形（曲线），找出曲线与 x 轴的交点的横坐标值就可求出 $f(x)=0$ 的近似根，即函数 $f(x)$ 的零点. 这些值虽然粗糙但直观，有多少个根、在何范围，一目了然. 并且还可以借助于图形局部放大功能，将根逐步定位得更加精细. 因此，如果你拥有一台计算机，那么图形的方法将是分析方程根的性态最简捷直观的方法. 不过，不要总想得到根的精确值.

例 2.1　求方程 $x^5 + 2x^2 + 4 = 0$ 所有的根及其大致分布范围.

解　画出函数 $f(x) = x^5 + 2x^2 + 4$ 的图形如图 2.1，可知 $f(x)=0$ 有多少个实数根以及大致分布情况. 用图形放大法，有选择性地逐步放大局部图形（有根区间的函数曲线），直至找到满意的根，如图 2.2. 由图 2.2 可知方程的根在 -1.55 与 -1.5 之间.

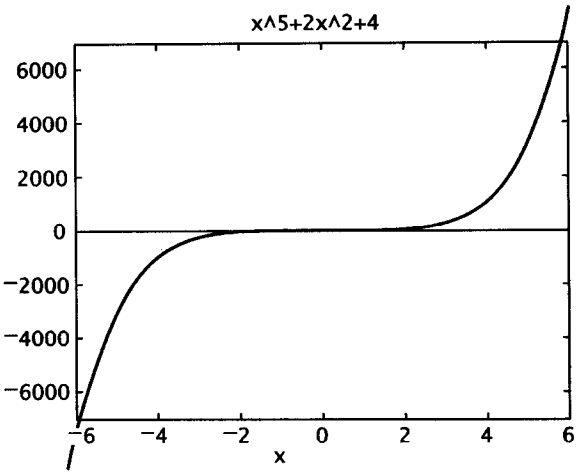


图 2.1　函数 $f(x)=x^5+2x^2+4$ 的图形

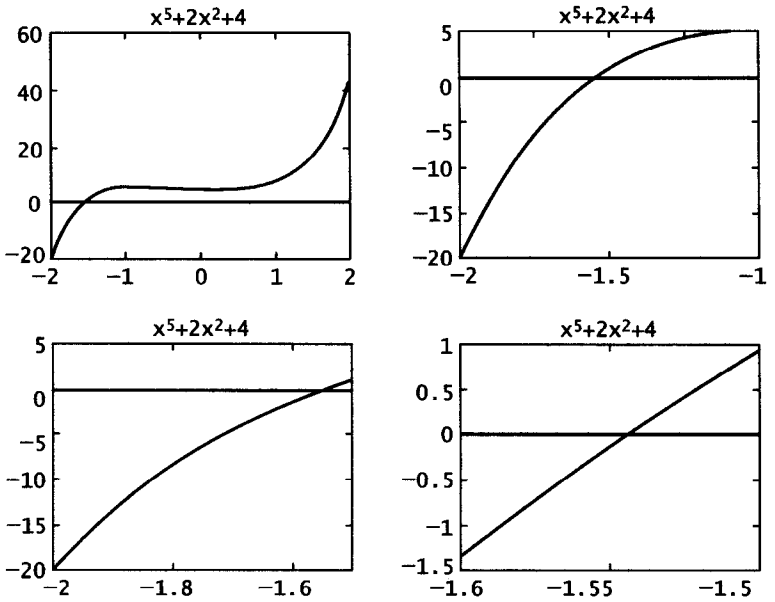


图 2.2　图形放大法求函数 $f(x)=x^5+2x^2+4$ 的零点

用图形放大法求方程 $x\sin x=1$ 的根. 注意观察结果. 它有多少个根?

将上式变形为 $\sin x=\dfrac{1}{x}$,可以画出两条函数曲线 $y=\sin x$ 与 $y=1/x$,两条曲线的交点的横坐标 x 就是原方程的根.

2. 数值迭代逼近法

利用图形的方法或连续函数的零点存在性定理,可以确定 $f(x)$ 在某一区间内有根,比如 $x^5+2x^2+4=0$ 在 $(-1.55,-1.5)$ 内有根,如果需要更精确地定位这个根,图形的方法就不如数值迭代逼近的方法有效. 下面讨论方程求根的几个数值迭代方法,分为两类:区间迭代与点迭代.

区间迭代如对分法、黄金分割法等,点迭代有简单迭代法、单点割线、两点割线法以及使用导数值的牛顿切线法等等.

1) 区间迭代法

对分法 是重复应用零点存在性定理,将区间一分为二,其中一个区间至少包含一个根,选择其中的一个有根区间,区间减少为原来的一半. 下一步再二分减半,如此迭代逐步缩短区间,直至最终区间长度达到满意的精度为止.

黄金分割法 与对分法本质上一致,只不过每次区间缩短的比例不是一半,而是$\dfrac{\sqrt{5}-1}{2}\approx 0.618$(黄金分割比例).

2) 点迭代法

简单迭代法 是基于构造与方程 $f(x)=0$ 等价的方程 $\varphi(x)=x$,得到迭代函数 $\varphi(x)$,然后求迭代函数 $\varphi(x)$ 的不动点. 相应的迭代公式如下

$$x_{n+1}=\varphi(x_n).$$

注意到 $\varphi(x)$ 形式不惟一,不同形式的迭代函数,产生的迭代结果可能差异很大. 如何构造函数 $\varphi(x)$,才能保证迭代算法的收敛性? 对此,有如下结论.

定理 设 $\varphi(x)$ 在 $[a,b]$ 上连续,且满足

① 对任意的 $x\in[a,b]$,$\varphi(x)\in[a,b]$;

② 存在常数 $L\in[0,1]$,使得对于任意的 $x,y\in[a,b]$,

$$|\varphi(x)-\varphi(y)|\leqslant L|x-y|,$$

则对任意初值 $x_0\in[a,b]$,迭代过程 $x_{n+1}=\varphi(x_n)$ 产生的序列 $\{x_n\}$ 收敛到 $\varphi(x)$ 在 $[a,b]$ 上的不动点.

下面介绍几个经典的迭代方法,它们是根据 $f(x)$ 的几何性质与解析性质,从不同角度构造出的迭代公式

$$x_{n+1}=\varphi(x_n).$$

1）对于方程 $\tan x=x$，用迭代算法求其非负实数根，考虑不同迭代函数.

2）用非迭代方法求解 $\tan x=x$，并对所得到的结果进行分析.

单点割线法 是给定初值 x_0（点 P_0）及当前值 x_n（点 P_n），将曲线 $y=f(x)$ 上当前点和初始点构成的割线（代替曲线）与 x 轴的交点的横坐标，作为本次迭代的输出 x_{n+1}，也为下次迭代的输入. 其迭代公式为

$$x_{n+1}=x_n-\frac{f(x_n)}{f(x_n)-f(x_0)}(x_n-x_0).$$

两点割线法 是给定初值 x_0 及 x_1（两个点），将曲线上最新的相邻（按迭代先后为序）两点的割线（代替曲线）与 x 轴的交点的横坐标，作为本次迭代输出. 其迭代公式如下：

$$x_{n+1}=x_n-\frac{f(x_n)}{f(x_n)-f(x_{n-1})}(x_n-x_{n-1}).$$

牛顿切线法 是给定初值 x_0（点 P_0），将过曲线上当前点的切线（代替曲线 $y=f(x)$）与 x 轴的交点的横坐标作为本次迭代的输出 x_{n+1}. 其迭代公式如下：

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}.$$

迭代函数 $\varphi(x)$ 的构造决定了迭代算法的收敛性及收敛速度.

用前面给出的收敛性定理，分析前面介绍的三种点迭代算法的收敛性.

2.3.3 线性方程组求解的数值方法

对于线性方程组

$$\begin{aligned}&a_{11}x_1+\cdots+a_{1n}x_n=b_1,\\&\cdots\cdots\cdots\cdots\\&a_{m1}x_1+\cdots+a_{mn}x_n=b_m\end{aligned}$$

可以写成矩阵的形式

$$\boldsymbol{AX}=\boldsymbol{b},$$

由线性代数知识可知，关于其解的理论如下：

解可能出现三种情形：无解、有惟一解和有无穷多组解. 这主要取决于系数

矩阵 A 的秩与增广矩阵$(A|b)$的秩是否相等、秩与变量个数是否相等,具体地:

若 $R(A)\neq R(A|b)$,则 $AX=b$ 无解;

若 $R(A)=R(A|b)=n$(n 为变量个数),则 $AX=b$ 有惟一解;

若 $R(A)=R(A|b)<n$,则 $AX=b$ 有无穷多组解.

求解方法大概可以划分为两类:直接消去方法(这主要是指高斯消去法)、迭代算法.

关于线性方程组的消去法和迭代算法,简介如下:

1) 直接消去法　理论上,经过有限次算术运算能够求出方程组的精确解,实际上由于计算存在舍入误差,因此,可能得到的只是近似解.

高斯消去法可以用 LU 分解来表示. 如果方程组的系数矩阵 A 满足一定条件,则 A 可以分解为一个下三角阵 L 与一个上三角阵 U 的乘积

$$A=LU,$$

于是由 $AX=b$ 得到

$$X=A^{-1}b=(LU)^{-1}b=U^{-1}L^{-1}b.$$

注意:上三角阵与下三角阵的逆矩阵很容易求得,因此,这是一种很好的思想.

2) 迭代法　常用的有雅可比迭代法和高斯 - 塞德尔迭代法等.

借鉴求解非线性方程的迭代算法思想,将 $AX=b$ 等价变形为(形式不惟一)

$$X=BX+f,$$

由此构造迭代公式

$$X^{(n+1)}=BX^{(n)}+f,\quad n=0,1,\cdots.$$

可以证明:在 B 满足谱半径 $\rho(B)<1$ 的条件下,当 $n\to\infty$ 时,$X^{(n)}$ 收敛,且其极限 X^* 就是原方程组的解.

雅可比迭代法　设 $A=L+D+U$(矩阵分解:D 为对角矩阵,L 为下三角矩阵,U 为上三角矩阵),由 $AX=b$ 得到

$$(L+D+U)X=b,X=D^{-1}[-(L+U)X+b]=BX+f,$$

其中 $B=-D^{-1}(L+U),f=D^{-1}b.$

高斯 - 塞德尔迭代法　将上述分解式子 $(L+D+U)X=b$ 换一种写法:

$$(L+D)X=-UX+b,X=(D+L)^{-1}(-UX+b)=BX+f,$$

其中 $B=-(D+L)^{-1}U,f=(D+L)^{-1}b.$

用以上两种不同的迭代算法求解以下线性方程组:

$$\begin{bmatrix} 5 & -1 & 0 & 0 \\ -1 & 5 & -1 & 0 \\ 0 & -1 & 5 & -1 \\ 0 & 0 & -1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 4.3 \\ 3.8 \\ 3.1 \\ 4.9 \end{bmatrix}$$

2.3.4 非线性方程组求解的数值方法

非线性方程组的一般形式如下：

$$\begin{aligned} &f_1(x_1,\cdots,x_n)=0,\\ &\cdots\cdots\cdots\\ &f_n(x_1,\cdots,x_n)=0, \end{aligned}$$

也可以写成向量形式

$$\boldsymbol{F}(\boldsymbol{x})=0$$

其中向量 $\boldsymbol{x}\in\mathbf{R}^n$，$\boldsymbol{F}(\boldsymbol{x})$ 为 n 维向量函数. 如果方程组中所有函数都是变量的线性函数，则称之为线性方程组，它是一类特殊的方程组，有其独特的求解方法（前面已经介绍）. 对于非线性方程组，可以借助非线性方程迭代算法，构造求解非线性方程组的简单迭代算法. 这里主要介绍解非线性方程组的牛顿迭代法.

考虑非线性方程组的向量形式：$\boldsymbol{F}(\boldsymbol{x})=0$，其中 $\boldsymbol{x}$ 和 $\boldsymbol{F}(\boldsymbol{x})$ 都为 m 维向量或向量函数.

若用 $\boldsymbol{x}^{(n)}$ 表示第 n 次迭代产生的输出（向量），通过将 $\boldsymbol{F}(\boldsymbol{x})$ 在当前点 $\boldsymbol{x}^{(n)}$ 处展开成一阶泰勒展式，去掉余项，并用 $\boldsymbol{x}^{(n+1)}$ 代替 $\boldsymbol{x}$，就得到求解方程组的牛顿迭代公式：

$$\boldsymbol{x}^{(n+1)}=\boldsymbol{x}^{(n)}-[\boldsymbol{F}'(\boldsymbol{x}^{(n)})]^{-1}\boldsymbol{F}(\boldsymbol{x}^{(n)}).$$

事实上，这是求解非线性方程的牛顿迭代公式的推广.

注意 牛顿迭代公式要求函数 $\boldsymbol{F}(\boldsymbol{x})$ 在 $\boldsymbol{x}^{(n)}$ 处的雅可比矩阵 $\boldsymbol{F}'(\boldsymbol{x}^{(n)})$ 可逆.

牛顿法中，每次迭代都需要计算雅可比矩阵 $F'(x^{(n)})$ 的逆矩阵. 当函数较为复杂或阶数较高时，给计算带来相当不便，因此出现了各种基于牛顿法的修正方法，如拟牛顿法等.

§2.4 飞机的最优价格

在最优定价策略下，价格 p 就是函数

$$\begin{aligned} &h\times(-78p^2+655p+125)+(p-1.5)h\times(-156p+655)-\\ &6h^{\frac{3}{4}}\times(-78p^2+655p+125)^{-\frac{1}{4}}(-156p+655) \end{aligned}$$

的零点.

我们可以采用如下几种方法求函数的零点：

图形法或直接利用 MATLAB 函数 fzero( )、迭代法等.

2.4.1 图形法

我们首先采用图形的方法求函数的零点. 用 MATLAB 作出(利润函数的导数)函数曲线,确定出函数的零点至少有两个:分别落在 1 到 2 之间与 6 到 7 之间. 再用图形放大法,进一步找出零点的近似值(如图 2.3 和图 2.4)分别为:

$$p_1 \approx 1.177; \quad p_2 \approx 6.286.$$

事实上,$p_1 \approx 1.177$ 为利润函数的极小值点,$p_2 \approx 6.286$ 为利润函数的极大值点. 得到利润函数的最优值为 $R = 1\ 780.833\ 6$,如图 2.5.

相应得 MATLAB 程序如下(作函数曲线图的基本程序):

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % 利润导数的定义:
    function ld = LirunD(p)
    h = 0.5;
    n = -78 * p^2 + 655 * p + 125;nd = -156 * p + 655;
    ld = h * n + (p - 1.5) * h * nd - 6 * (h * n)^(3/4)/n * nd;
    % 图形放大过程:
    subplot(2,2,1);hold on ;a = 0;b = 8;
    fplot(['LirunD',0],[a,b]);fplot('0',[a,b],'k:');
    title('利润导数曲线 R'(p)')

    subplot(2,2,2);hold on;a = 1;b = 2;
    fplot(['LirunD',0],[a,b]);fplot('0',[a,b],'k:');
    title('利润导数曲线 R'(p)')

    subplot(2,2,3);hold on;a = 1.1;b = 1.25;
    fplot(['LirunD',0],[a,b]);fplot('0',[a,b],'k:');
    title('利润导数曲线 R'(p)')

    subplot(2,2,4);hold on;a = 1.17;b = 1.18;
    fplot(['LirunD',0],[a,b]);fplot('0',[a,b],'k:');
    title('利润导数曲线 R'(p)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

注意:

1) 根据图形的具体情况,不断修改上面程序中的区间端点参数 a,b 的取值(第一条语句),就可以不断地放大图形,将函数的零点范围限制得越来越小,直至找出满意的近似根;

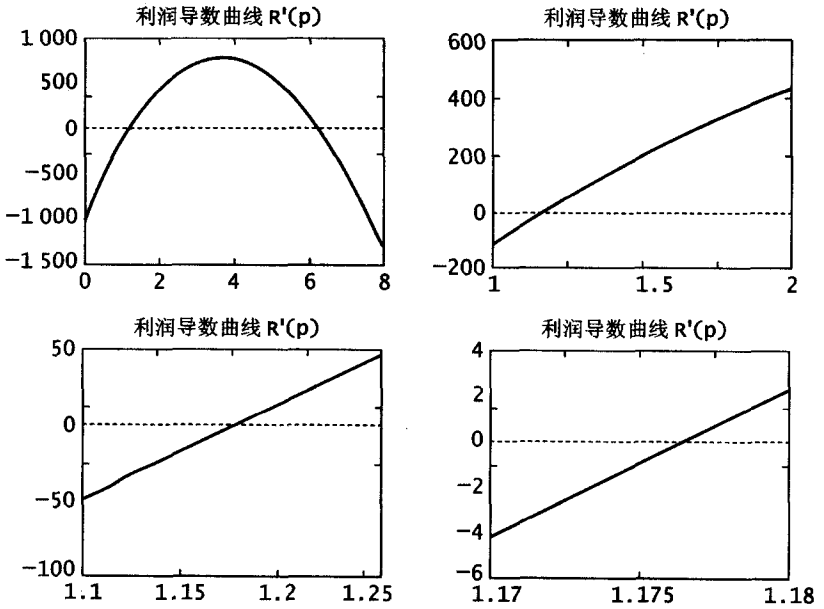


图 2.3　图形放大法求利润函数导数的零点之一

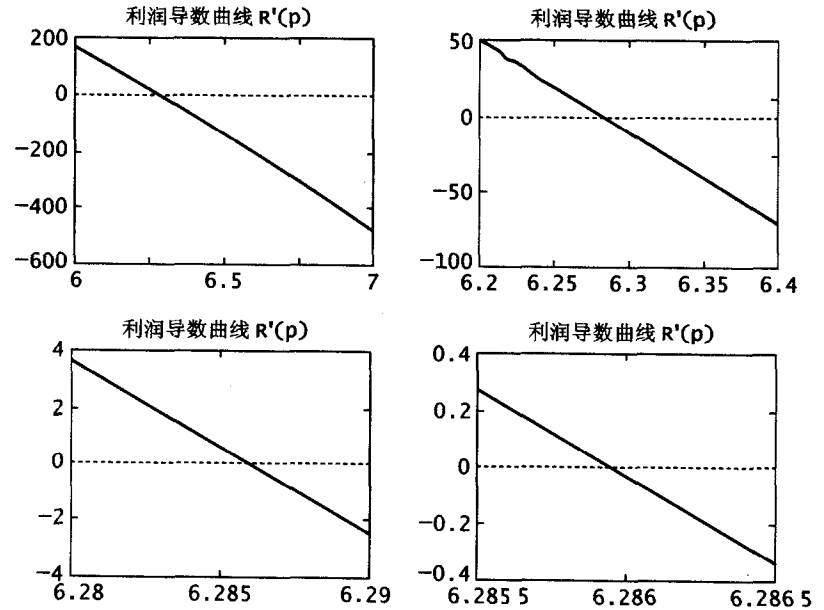


图 2.4　图形放大法求利润函数导数的零点之二

2）以上市场占有率 $h=0.5$；对于市场占有率 h 的其他取值，可以类似地进行，只需要修改程序中函数定义中的参数 h 的值；

3）也可以直接用 MATLAB 中求函数的零点的内部函数 fzero，求得利润导函数的零点如下：

```
fzero('LirunD',3)  % ans = 1.1764
fzero('LirunD',4)  % ans = 6.2860
```

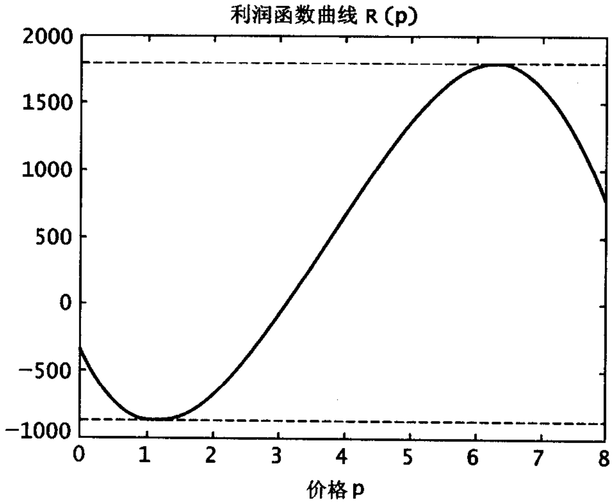


图 2.5　利润函数曲线

2.4.2　迭代法

前面采用图形法，求利润函数导数的零点．下面介绍用对分法在区间[a,b]内寻找函数零点的 MATLAB 程序 M 文件：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % 在区间[a,b]内寻找函数零点,利润导函数有 m 文件 LirunD 定义
  a=0;b=4;fa=feval('LirunD',a);fb=feval('LirunD',b);
  time=0;% 迭代次数
  while(time<100)
    h=(a+b)/2;fh=feval('LirunD',h);
    if (fh==0)
      break
      elseif(fa*fh<0)
        b=h;fb=fh;
      else
        a=h;fa=fh;
```

```
    end
    time = time + 1;
end
p = (a + b)/2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

运行该程序,其输出结果为:$p = 1.1764$. 因为利润导数函数有两个零点,将程序中区间的端点 a,b 改为“a = 4;b = 8”,则得到另外一个零点 $p = 6.2860$.

2.4.3 飞机如何定价问题的解答

前面三种方法都可以得到利润函数导数的两个零点:$p = 1.1764$ 和 $p = 6.2860$,如图 2.5 所示. 经过分析与图形直观表示发现,$p = 1.1764$ 为利润函数的极小值点,$p = 6.2860$ 为利润函数的极大值点,并且为最大值点. 因此,飞机的最优定价为 $p = 6.2860$.

飞机如何定价问题的解为:价格 $p = 6.2860$,相应的最佳利润为 $R = 1780.8336$.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 利润函数的定义:
function l = Lirun(p)
h = 0.5;
n = h * ( - 78 * p^2 + 655 * p + 125);
r = p * n;c = 50 + 1.5 * n + 8 * n^(3/4);
l = r - c;
% 利润函数曲线:
fplot('Lirun',[0,8]);hold on;
f1 = feval('Lirun',1.177);f2 = feval('Lirun',6.286);
x = 0:0.1:8;n = size(x);
plot(x,f1 * ones(n),'k:');plot(x,f2 * ones(n),'k:');
xlabel('价格 p');title('利润函数曲线 R(p)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

想

1) 在模型假设中,除二次函数以外,还可以用其他哪些符合市场规律的曲线来预测飞机的销售量?

2) 模型的 6 条假设中,哪些不太合理,应该如何修改?

做

1) 计算 h 取其他值时的最优价格,并进行比较.

2) 该问题本身是一个最值问题,直接采用求最优值的函数 fminsearch(fun,x0) 求解.

§2.5 操　　练

操练一　油价与船速的优化问题

油价的上涨，将对大型海船确定合理的航行速度产生影响，以优化航行收入. 直观地，油耗的多少直接影响船速的快慢，因而直接影响航行时间的长短，进而影响支付船员人工费用数量. 过去有一些经验表明：(1) 油耗正比于船速的立方；(2) 最省油航速的基础上改变 20% 的速度，则引起 50% 的油耗的变化. 作为一个例子：某中型海船，每天油耗 40 t. 若减少 20% 的航速，可省油 50%，即每天耗油 20 t. 油价 250 美元/吨，由此每天减少耗油费用 5 000 美元，而航行时间的增加将对船员支付的费用的增加，如何最优化？

算例：航程 $L = 1\ 536$ n mile(海里)，标准最省油航速 20 kn(节)，油耗每天 50 t，航行时间 8 天. 最低航速 10 kn，本次航行总收入为 84 600 美元. 油价 250 美元/吨，日固定开支 1 000 美元. 试确定最佳航速(1 n mile = 1 852 m，1 kn = 1 n mile/h = (1 852/3 600) m/s).

操练二　航空公司的预订票策略

在激烈的市场竞争中，航空公司为争取更多的客源而开展的一项优质服务项目是预订票业务. 公司承诺，预先订购机票的乘客如果未能按时前来登机，可以乘坐下一班机或退票，无需附加任何费用.

设飞机容量为 n，若公司限制只预订 n 张机票，那么由于总会有一些订了机票的乘客不能按时前来登机，致使飞机因不满员飞行而利润降低，甚至亏本. 如果不限制订票数量，则当持票按时前来登机的乘客超过飞机容量时，将会引起那些不能登机的乘客(以下称被挤掉者)的抱怨，导致公司声誉受损和一定的经济损失(如付给赔偿金). 这样，综合考虑公司的经济利益和社会声誉，必然存在一个恰当的预订票数量的限额.

假设已经知道飞行费用(可设与乘客人数无关)、机票价格(一般乘客占飞机满座的 50% ~60% 时不亏本，由飞行费用可确定价格)、每位被挤掉者的赔偿金等数据，以及由统计资料估计的每位乘客不按时前来登机的概率(不妨认为乘客间是相互独立的)，试建立一个数学模型，综合考虑公司经济利益(飞行费用、赔偿金与机票收入等)和社会声誉(被挤掉者不要太多，被挤掉的概率不要太大等)，确定最佳的预订票数量.

1) 对上述飞机容量、费用、迟到概率等参数给出一些具体数据，按你的模型计算，对结果进行分析；

2) 对模型进行改进，如增设某类旅客(学生、旅游者)的减价票，迟到则机

票作废等.

更多的相关信息资源

1 萧树铁主编,姜启源,何青,高立编著. 数学实验. 北京:高等教育出版社,1999

2 姜启源. 数学模型. 第2版. 北京:高等教育出版社,1993

3 傅鹂,龚劬,刘琼荪,何中市. 数学实验. 北京:科学出版社,2000

4 任善强,雷鸣. 数学模型. 重庆:重庆大学出版社,1996

5 D. Quinney. An Introduction to the Numerical Solution of Differential Equations. New York:John Wiley & Sons Inc. ,1985

6 G. Fulford. Modelling with differential and difference equations. Cambridge:Cambridge University Press,1997

7 Peitgen, Heinz-Otto. Chaos and fractals: new frontiers of science. New York: Springer-Verlag,1992

8 何良材,何中市. 经济应用数学. 第2版. 重庆:重庆大学出版社,1999

第3章

收敛与混沌

——迭代

迭代是方程求解的几乎惟一方法. 对于一个给定的迭代算法，它产生迭代序列的变化行为除了收敛之外，还可能具有十分复杂、古怪的现象：周期性变化、分岔和混沌等. 对于混沌，虽然目前尚未给出一个统一的定义，但是迭代序列的渐近行为研究已经成为动力系统的热点，其研究结果不仅对科学计算具有重要的意义，而且已经引起其他领域科学家（如哲学家、物理学家、经济学家、生物学家等）的诸多兴趣.

——作者

许多实际问题的求解，大都采用迭代算法. 由于计算机以重复性运算见长，故为研究迭代算法提供了极为合适的手段. 例如非线性方程或函数的迭代将会产生一系列数列，数列或者收敛或者不收敛，如何判断一个数列是否收敛？在高等数学的课程中，我们学到了一些判定定理和准则，如柯西准则，单调有界准则、夹逼准则等等. 如何将理论（这些定理、准则）与实际相结合，判定一个给定的迭代数列的收敛性？这可能不是一个简单的问题. 例如由式 $x_{n+1}=3x_n(1-x_n)$，$x_0=0.4$ 迭代产生的数列 $\{x_n\}$ 收敛与否？

本章将通过计算机实践，以图形的方式探索迭代数列的收敛性. 期望能够为解决某些实际问题提供一些有益的解决途径.

§3.1 不动点与迭代

3.1.1 什么是迭代

科学实验就是反复观察、试验、总结、发现规律. 古人云：“路遥知马力，日久

见人心”,这句话既表达了一种哲学思想,也揭示出一个重要的数学方法. 其实质是通过反复操作或观察某种现象,总结出一些隐藏而又深刻的规律,在数学上这正是迭代的思想.

迭代是一个非常好的从实践中来到实践中去的典型例子. 例如,计算器的操作:当你使用计算器时,你是否无意中按过计数器上的某个键,如根号键 $\sqrt{\ }$. 输入 2,再重复按 $\sqrt{\ }$ 键,这样,你会得到一个数列:

$$1.414\,213\,562, 1.189\,207\,115, 1.090\,507\,733,$$
$$1.044\,273\,782, 1.021\,897\,149, 1.010\,889\,286, \cdots$$

数列中的第一项既是第一次操作的输出,也是第二次重复操作的输入,这种将输出作为新的输入的重复计算过程,实质上就是迭代. 下面给出迭代的一个数学描述:

任意给定一个输入值 x,由一个函数表达式 f 得到一个输出 $f(x)$;再将 $f(x)$ 作为新的输入 x,得到下一个输出 $f(x)$;…,这种对某个函数规则 f 反复将输出作为新输入的重复执行过程就称为**迭代**. 数学表示为:

$$x_{n+1} = f(x_n), n = 0, 1, \cdots,$$

函数 $f(x)$ 称为迭代函数,数列 $\{x_n\}$ 称为迭代数列,x_0 称为迭代初值(启动值). 所以,迭代就是将输出作为输入的简单重复计算过程.

迭代既是一种问题求解的重要的数值方法,又是一种典型的数学变换.

1) 迭代与重复实验有何区别?

2) 迭代的结果是否一定是数值?

3) 是否存在一个平面图形,其面积有限而周长无穷?

Koch 雪片的生成　初始阶段:一个面积为 1 的等边三角形;第 1 阶段:将三角形每条边中间 1/3 部分的换成一个等边三角形;第 $k+1$ 阶段:将每一个三角形的每条边中间 1/3 部分的换成一个等边三角形,…. 观察图形的变化规律,并分析该平面图形的周长和面积.

重复地执行既定方式,关键在于迭代的规则,即迭代函数 $f(x)$;当然也可能依赖于初值的选取. 无穷迭代的趋势可能稳定(收敛),也可能不稳定(发散). 如前面提到的开方问题 $f(x) = \sqrt{x}$,倒数问题 $f(x) = 1/x$. $f(x) = \sqrt{x}$ 对任何正数初值,迭代生产的数列是收敛的(极限是 1),就称迭代是**收敛**的. 而 $f(x) = 1/x$ 对除 0 和 ±1 以外的任何初值,迭代产生的数列是跳跃(不收敛)的,因此迭代是**发散**的.

3.1.2 不动点

对于迭代函数 $f(x)=\sqrt{x}$，将 $x=1$ 作为输入，迭代输出值也为 1，也就是迭代输出将不会再改变. $x=1$ 是函数迭代不动(不变化)的点，称为不动点. 一般地有：

定义 对函数 $f(x)$，如果存在点 u，使 $f(u)=u$，则称 u 为 $f(x)$ 的一个**不动点**.

一个函数的不动点可能不惟一，在函数不同的不动点附近迭代的情况可能有很大区别. 例如 $x=0$ 与 $x=1$ 都是 $f(x)=\sqrt{x}$ 的不动点，但是，在 $x=1$ 和 $x=0$ 附近，对 $\sqrt{x}$ 的迭代情况大不一样. 在 $x=1$ 附近的任何初值 $x_0(>0)$，迭代过程都收敛于该不动点 $x=1$，称此不动点为**吸引的**. 在 $x=0$ 附近的任何初值 $x_0(>0)$，迭代过程都不会收敛于不动点 $x=0$，称此不动点为**排斥的**.

§3.2 图示迭代数列

迭代作为一种特殊的数学变换，在数学理论上，众多复杂的函数或规律可以用迭代甚至是很简单的迭代来研究. 对于迭代函数 $f(x)=ax+b$，因为它是线性函数，因此称为**线性迭代**，否则称为**非线性迭代**，如二次函数迭代等.

因为线性迭代比较简单，但是非线性迭代则不然. 本节仅以二次函数迭代为例，讨论揭示迭代数列的规律性的图形方法. 将会碰到某些全新的现象和问题，甚至可以导出像分岔、混沌这样的古怪现象.

我们讨论的二次函数如下

$$f(x)=ax(1-x),$$

其中参数 a 在 0 和 4 之间取值. 该函数称为逻辑斯谛(Logistic)函数，其他二次函数与逻辑斯谛函数具有类似的性质.

该二次函数定义的迭代过程如下：

$$x_n=ax_{n-1}(1-x_{n-1}),n=1,2,3,\cdots,x_0\in(0,1).$$

该二次函数存在两个不动点 $x=0$ 与 $x=1-1/a$. 显然，对于以不动点为初始值的迭代都将收敛于相应的不动点.

1) 对于参数 $a=3$，以上二次函数迭代是否收敛？

2) 可以用图形的方式研究迭代数列的收敛性，能否借助声音来研究迭代数列的变换规律？

1）验证数列 $x_n=3x_{n-1}(1-x_{n-1})$，$n=1,2,\cdots$ 收敛或不收敛.

2）判断逻辑斯谛函数的两个不动点 $x=0$ 与 $x=1-1/a$ 是吸引的还是排斥的？

逻辑斯谛函数产生的迭代数列收敛性如下：

若 $a\in(0,1]$，则迭代对于任何除不动点以外的初值都是收敛的，并收敛于不动点 $x=0$.

若 $a\in[1,3)$，则迭代对于任何除不动点以外的初值都是收敛的，并收敛于不动点 $x=1-1/a$.

若 $a\in[3,4]$，则迭代的收敛性非常难以想像！随着参数 a 取值的增大，迭代数列会出现诸如收敛、周期振荡、分岔、混沌之类的从有规律到无规律，又从无规律到有规律等等的非常复杂的、有趣的、甚至是怪异的现象.

对于 a 的某个取值，当 n 充分大时，如果迭代数列在 k 个值 $x_1,x_2,\cdots,x_k$ 之间周期性地来回振荡，则称该迭代数列为 k - 周期.

对于 $a\in[3,4]$ 不同的取值，尽量挖掘迭代数列 $x_n=ax_{n-1}(1-x_{n-1})$，$n=1,2,\cdots$ 的不同变化模式. 体会迭代数列周期性的变化规律：有 2 - 周期、3 - 周期、5 - 周期吗？如果有，a 取何值时出现？

下面主要介绍几种刻画迭代数列变化规律的图形方法.

3.2.1 线性联结图

联结相邻迭代点列的折线所形成的图形，就称为线性联结图. 即联接两个点 (n,x_n) 与 $(n+1,x_{n+1})$ 的折线构成的图形，如图 3.1 所示.

为了更好地揭示迭代数列在无穷远处的变化规律，可以去掉迭代数列前面若干项，如前面 10 000 项，而只显示从第 10 001 项开始的一段迭代数列.

图 3.2 就是（去掉前面 10 000 项）显示从 10 001 项开始的 20 项的结果，其中四幅图分别对应参数 a 的四个不同取值：$a=3.2,3.5,3.564\ 4,3.828\ 4$. 从该图中，能够看出迭代数列的变化模式吗？

$a=3.2$ 时，迭代数列呈现 2 - 周期振荡；$a=3.5$ 时，呈现 4 - 周期振荡；$a=3.564\ 4$ 时，呈现的是 4 - 周期还是 8 - 周期呢？$a=3.828\ 4$ 时，是 2 - 周期还是 3 - 周期呢？

仔细观察，逐步放大，大胆尝试，小心求证.

如果发现线性联结图尚不够得出结论，可以尝试采用新的图示方法.

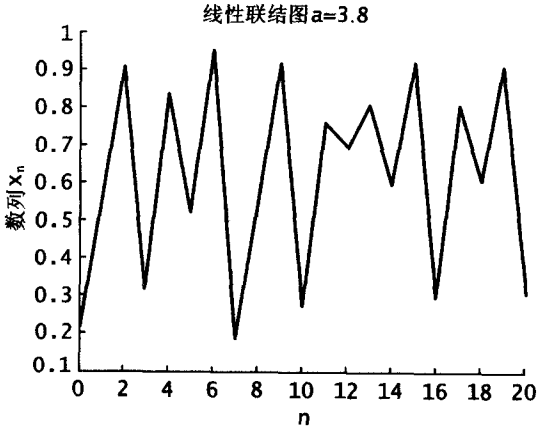


图 3.1　线性联结图：$x_n = ax_{n-1}(1 - x_{n-1})$

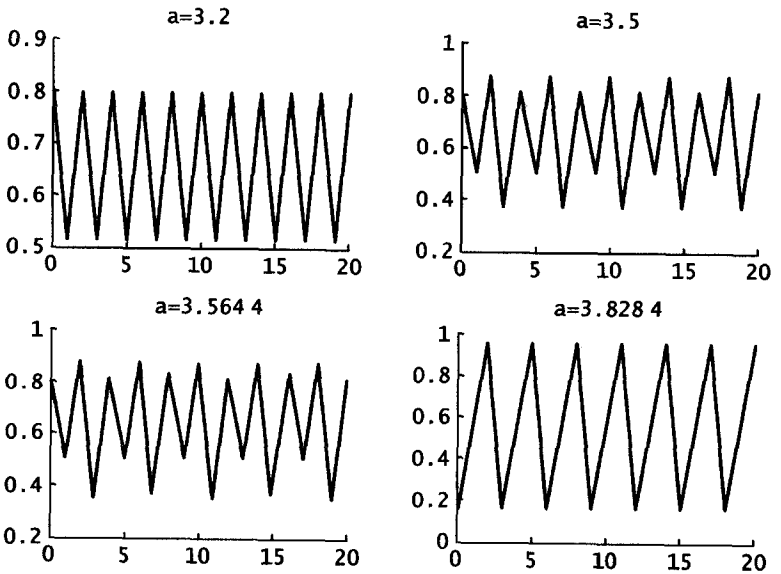


图 3.2　线性联结图：$x_n = ax_{n-1}(1 - x_{n-1})$

3.2.2　蛛网图

在平面直角坐标系 xOy 里画出直线 $y = x$ 与曲线 $y = f(x)$，函数 $f(x)$ 表示的迭代过程：

$$x_{n+1} = f(x_n), n = 0, 1, \cdots, \text{初值 } x_0.$$

迭代产生的点列记为$\{x_n\}$，则可以在图中表示$\{x_n\}$如下：

第一步：过直线$y=x$上点A_n($y=x_n$与$y=x$的交点)，作垂线与$y=f(x)$相交于B_n点；

第二步：过点B_n作水平线与$y=x$相交于A_{n+1}点；

第三步：A_{n+1}点的横坐标就是x_{n+1}，再以为开始点(即用x_{n+1}代替x_n)，返回到第一步，进行迭代.

经过第一步到第三步的迭代，重复地作垂直投影与水平投影，如此得到的折线图$A_nB_nA_{n+1}B_{n+1}\cdots$，形如一个阶梯或一个蜘蛛网，因此称之为蜘蛛网图，简称蛛网图. 如图 3.3 所示.

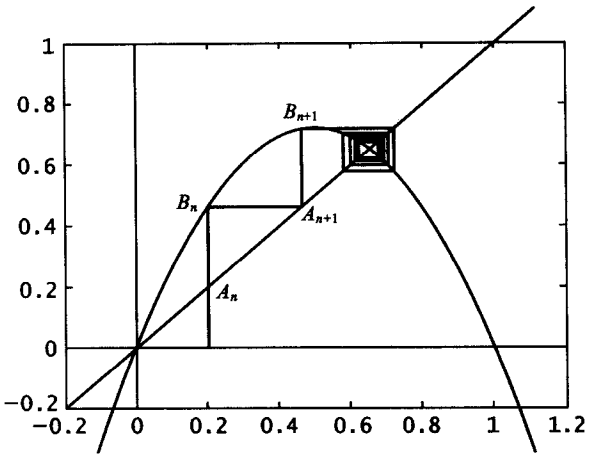


图 3.3　蛛网图

图 3.4 就是(去掉前面 10 000 项)显示从 10 001 项开始的 20 项的蛛网图，图 3.2(线性联结图)表示的是相同的四个迭代数列. 由图 3.4 的四幅图能够比较清楚地看出迭代数列的变化模式：

$a=3.2$时呈现 2 - 周期，$a=3.5$时呈现 4 - 周期，$a=3.564\ 4$时，呈现的是 8 - 周期而不是 4 - 周期，$a=3.828\ 4$时，呈现的是 3 - 周期而不是 2 - 周期.

3.2.3　费根鲍姆图

为研究二次函数$f(x)=ax(1-x)$不同参数a的取值对迭代的影响，将迭代产生的数列$\{x_n\}$的一部分画在同一平面直角坐标系中：参数a为横坐标，迭代产生的数列为纵坐标. 具体地，给定一个固定的初值x_0，如$x_0=0.3$，对一个a值，选取$\{x_n\}$的若干连续的一段数列，为更好地揭示规律，不妨去掉前面的若干(规律不明显的)项，比如 10 000 项，将从第 10 001 项开始的数列分别作为纵坐标，

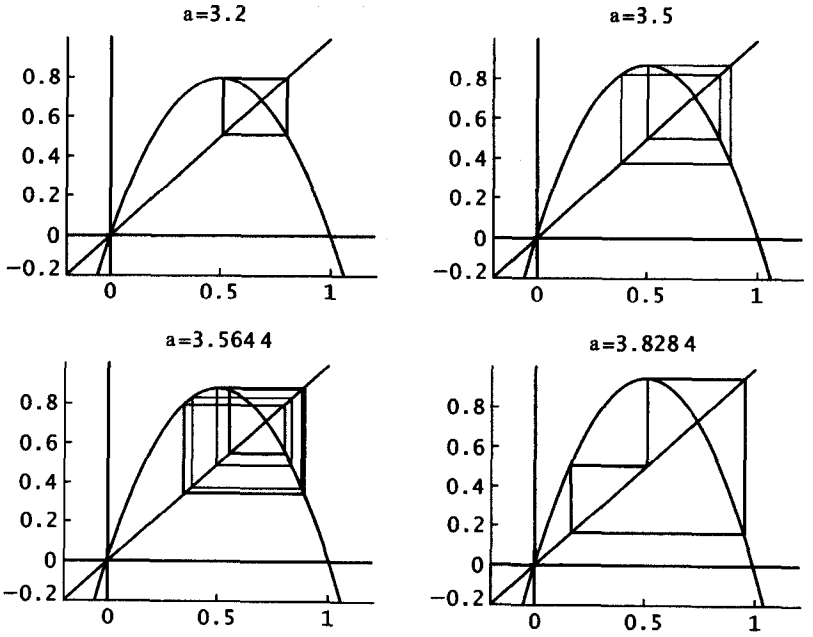


图 3.4　蛛网图:$f(x)=ax(1-x)$

打印在该直角坐标系中的一条铅直线 $x=a$ 上. 然后对下一个 a 值,如法炮制. 这样得到的图,就称为费根鲍姆(Feigenbaum)图. 如图 3.5 分别给出了 $a=2.9$, 3.2,3.5,3.564 4,3.7,3.828 4 时二次函数 $f(x)=ax(1-x)$ 的费根鲍姆图,迭代初始值为 0.2.

图 3.5 中直接输出前 1 000 项的结果,图中表示的周期性情况不太清楚.

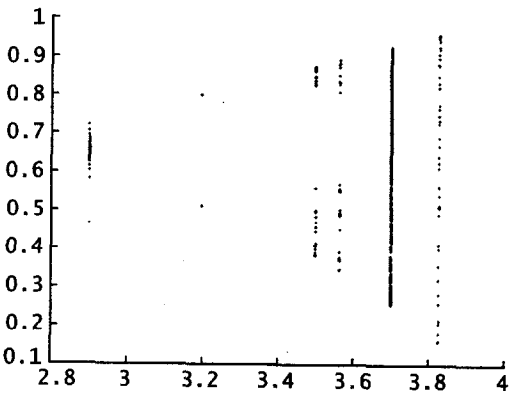


图 3.5　费根鲍姆图:$f(x)=ax(1-x)$

图 3.6 给出了去掉迭代产生的前 10 000 项以后的 1 000 项的结果，非常明显地刻画了参数 $a=2.9,3.2,3.5,3.564\,4,3.7,3.828\,4$ 的这 6 个不同取值时的规律：分别为收敛，2 - 周期，4 - 周期，8 - 周期，混沌，3 - 周期.

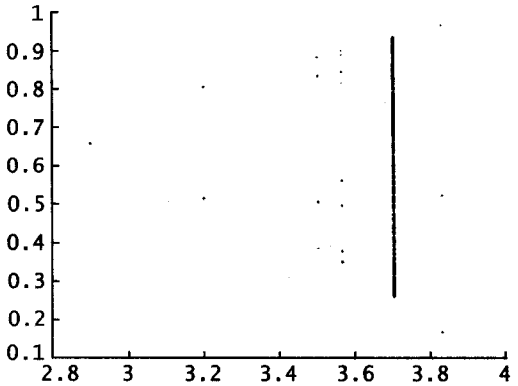


图 3.6 费根鲍姆图

本节介绍的三种数列表示的图形方法优劣何在？

§3.3 分岔与混沌

3.3.1 倍周期

利用上面的工具可以发现二次函数迭代对参数 a 的敏感性. 对 a 在 3 到 4 之间，除不动点之外的初始值，迭代都是不收敛的.

当 $a\in(3,1+\sqrt{6}]$，迭代数列将在如下两个值 x_{21}，x_{22} 之间来回振荡，即 2 - 周期.

$$x_{21}=\frac{a+1+\sqrt{a^2-2a-3}}{2a},x_{22}=\frac{a+1-\sqrt{a^2-2a-3}}{2a}.$$

当 $a\in(1+\sqrt{6},3.544\,09]$，迭代数列将在四个值 x_{41}，x_{42}，x_{43}，x_{44} 之间来回振荡，即 4 - 周期，为前一个阶段周期的两倍，称为倍周期(Period Doubling)现象，由于该倍周期现象的周期基数为 2，因此又称为倍 2 - 周期现象. 可见

倍 2 - 周期的分裂行为的周期依次是：2，4，8，16，32，64，….

倍 3 - 周期的分裂行为的周期依次是：3，6，12，24，48，96，….

倍 5 - 周期的分裂行为的周期依次是：5，10，20，40，80，160，….

对 $a \in (1+\sqrt{6}, 3.544\ 09]$，迭代数列将在四个值 $x_{41}, x_{42}, x_{43}, x_{44}$ 之间来回振荡，即 4 - 周期，你能够找出这四个值吗？

3.3.2 分岔

以上这些由 2 - 周期到 4 - 周期，由 4 - 周期到 8 - 周期，8 - 周期到 16 - 周期，…，等等的倍周期分裂行为就是所谓的**分岔**(Bifurcation). 随着参数 a 的取值的增加，分岔频率逐渐加快，亦即保持同一周期的参数取值范围减少. 费根鲍姆通过计算机数值计算发现，频率增加的比例趋近于一个常数，这个常数就是 $f = 4.669\ 201\ 609\cdots$，被称为**费根鲍姆常数**. 图 3.7 给出了参数 a 在 $[2,4]$ 上等距离(步长为 0.02)取值时的费根鲍姆图，图中是去掉迭代的前 10 000 项后的 1 000项的结果. 图 3.7 非常直观地揭示了：

1) 迭代序列周期性变化规律；

2) 随着 a 的取值的增加，出现倍 2 - 周期分裂行为，即分岔现象；

3) 倍 2 - 周期分裂的频率不断加快.

如果设 2^k - 周期的持续长度(参数 a 取值范围)为 d_k，则有

$$\lim_{k\to\infty}\frac{d_k}{d_{k+1}} = f = 4.669\ 201\ 609\cdots(\text{费根鲍姆常数})$$

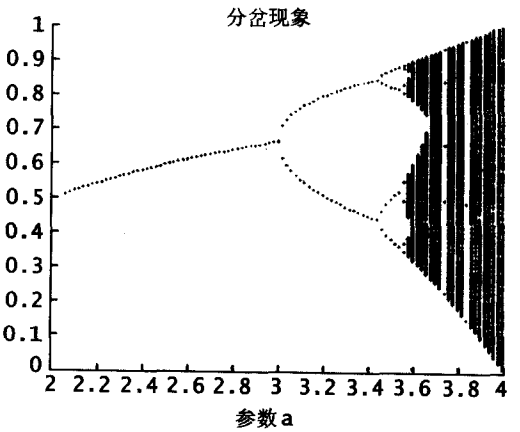


图 3.7 费根鲍姆图：$a \in [2,4]$

1) 4 - 周期在 $a = 3.544\ 09$ 处分裂为 8 - 周期，8 - 周期将在何处分裂为 16 - 周期？

2) 图 3.7 中 $a = 3.8$ 左右的密集点阵中出现了两条奇特的缝隙(分别含 5 个和 3 个点)，又有什么现象会在这里发生呢？

3.3.3 混沌

当 $a=3.569\ 945\ 672$ 时，以上的倍 2 - 周期分裂行为（分岔）终止，迭代进入没有周期性规律的模式，并且，迭代数列非常敏感地依赖于初始值 x_0 的选取，这种不规则性与不可预测性，就是所谓的混沌（Chaos）. 混沌与分岔相伴发生，分岔是混沌出现的早期现象. 例如 $a=3.7$ 的迭代行为就是如此.

通过实验还可以发现，当 $a=3.7$ 时，迭代对初值的高度敏感性：无论两个初值如何靠近，经过迭代其结果（两个数列）将是各行其道，这体现了混沌的不可预测性.

1）对于参数 $a\in(3.569\ 945\ 672,4)$，二次函数迭代是否总是出现混沌行为？

2）是否会出现类似倍 2 - 周期的分岔行为的倍 3 - 周期的分岔行为？

1）对于 $a=1+2\sqrt{2}$，实验说明迭代数列 $x_n=ax_{n-1}(1-x_{n-1})$，$n=1,2,\cdots$ 的变化规律.

2）通过计算机实验，发现更多的倍周期分岔行为，如倍 3 - 周期、倍 5 - 周期、倍 7 - 周期现象等等.

§3.4 二元函数迭代

前面讨论的是基于一元函数的迭代，下面将简单讨论由两个二元函数联合产生的迭代过程的收敛性. 将会碰到既不是周期振荡、又非混沌现象的有轨振荡现象. 由此引发人们对小到微粒的布朗运动，大到现代化文明城市的地铁运行等等的无穷遐想.

有两个二元函数 $f(x,y)$，$g(x,y)$，构造迭代过程如下：

$$x_{n+1}=f(x_n,y_n),$$

$$y_{n+1}=g(x_n,y_n),$$

由此产生的数列 $\{x_n,y_n\}$，可以看成两个数列 $\{x_n\}$，$\{y_n\}$. 因此完全可以用一元迭代数列的收敛性来考察二元迭代过程的收敛性. 即分别考察数列 $\{x_n\}$，$\{y_n\}$ 的收敛性，所以完全可以应用前面介绍的一元函数迭代数列的图形方法.

若 $u=f(u,v)$，$v=g(u,v)$，则称 (u,v) 为二元迭代函数 (f,g) 的**不动点**.

下面要介绍几个迭代数列：

高斯算术几何平均数列(不含参数)：对于不同迭代初始值(a,b)，迭代都收敛，但是收敛的极限值高度依赖于初始值.

旋转数列(含参数)：对于不同参数值a. 迭代数列不是吸引到原点$(0,0)$，就是趋近于无穷大.

海伦数列(含参数)：对于不同参数值a，迭代几乎都不收敛，考虑到迭代初始值的不同，迭代数列将出现不同现象：周期性振荡、有轨振荡、无穷发散等等.

3.4.1 高斯算术几何平均数列

例如：由如下两个二元函数

$$f(x,y)=\frac{1}{2}(x+y),g(x,y)=\sqrt{xy}$$

产生的迭代数列：

$$x_{n+1}=\frac{1}{2}(x_n+y_n),$$

$$y_{n+1}=\sqrt{x_n y_n},n=0,1,\cdots$$

这就是有名的**高斯算术几何平均数列**.

该二元函数迭代有无穷多个不动点：$(x,x),x\geqslant 0$.

可以证明：

结论 1 高斯算术几何平均数列对于任何非负初值$x_0=a,y_0=b$都是收敛的，并且两个数列收敛的极限相同，但其极限值，记为$M(a,b)$，却依赖于初值a，b.

图 3.8 是用一对线性联结图来分别表示二元迭代数列(x_n,y_n)的两个一元数列$\{x_n\}$，$\{y_n\}$的收敛性，图中给出了 10 个不同迭代初始值产生的前 10 个点的线性联结图. 由此图可以直观地看出，对于每一个迭代初值，两个数列都是收敛的，从而二元迭代是收敛的，但是其收敛极限高度依赖于迭代初值.

高斯早在他 22 岁时就发现并证明了无理数π、椭圆积分与$M(a,b)$存在密切的关系. 具体如下：

结论 2 $M(a,b)=\frac{\pi}{2}\cdot\frac{1}{G}$,

其中G为椭圆积分$G=\int_0^{\pi/2}\frac{1}{\sqrt{a^2\cos^2 x+b^2\sin^2 x}}\mathrm{d}x$.

显然可以利用以上公式计算椭圆积分.

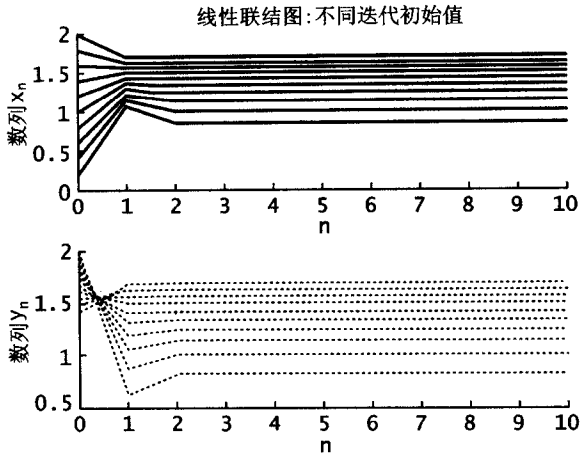


图 3.8　二元迭代的线性联结图:高斯算术几何平均数列

1）如何从理论上证明高斯以上发现，即 $M(a,b)=\frac{\pi}{2}\cdot\frac{1}{G}$?

2）能否用数学软件 Mathematica 或 MATLAB 验证高斯的以上结论?

3）如何画出 $M(a,b)$ 的 3 维图形:$M(a,b)$ 随 a,b 变化的情况?

利用结论 2 计算椭圆积分:$G=\int_0^{\pi/2}\frac{1}{\sqrt{2\cos^2 x+\sin^2 x}}\mathrm{d}x$. 对此积分，还有其他计算方法吗? 如果有，将计算结果与上面的计算结果进行比较，有何发现?

对高斯数列略做修改，得到所谓的 **Borchardt 数列**

$$x_{n+1}=\frac{1}{2}(x_n+y_n),y_{n+1}=\sqrt{x_{n+1}y_n},n=0,1,\cdots.$$

1）该数列收敛性如何? 是否依赖于初值 x_0,y_0 的选取? 两个极限关系如何?

2）记 $m(a,b)$ 为 $x_0=a,y_0=b$ 时该数列的极限，讨论 $m(a,b)$ 与 $M(a,b)$ 及 π 的关系.

3）能否用一个图形将二元数列直接表示出来?

3.4.2 旋转数列

对于如下二元函数迭代数列：

$$x_{n+1}=a(x_n-\sqrt{3}y_n),\quad y_{n+1}=a(\sqrt{3}x_n+y_n),n=0,1,\cdots$$

参数 $a\in[0,+\infty)$. 该二元函数迭代只有一个不动点为(0,0).

可以证明：对于参数 $a\in[0,0.5)$，迭代数列 (x_n,y_n) 对于任何初值 (x_0,y_0) 都收敛于不动点(0,0). 图 3.9 给出参数 $a=1/3$，由 10 个不同初始值产生的迭代数列的前 10 项的线性联结图. 可以看出，对于这 10 个初始值，迭代都是收敛的.

对于参数 $a\in[0.5,+\infty)$，迭代数列 (x_n,y_n) 对于任何初值 (x_0,y_0) 都不收敛.

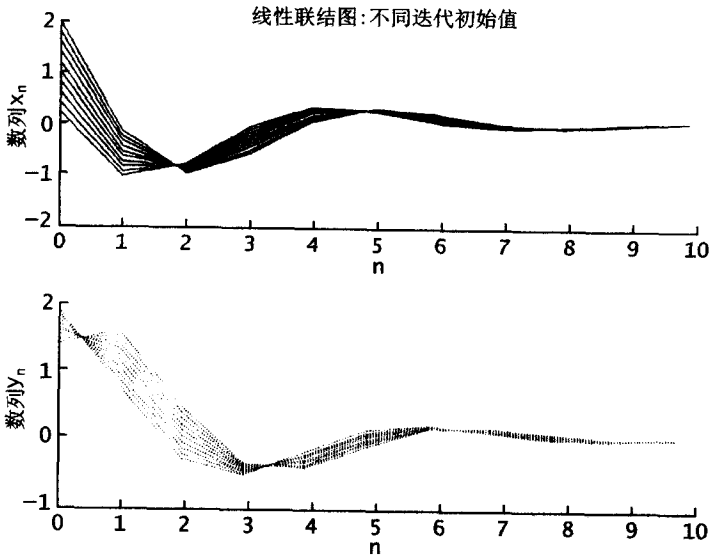


图 3.9　二元迭代的线性联结图：旋转数列

1）用分析的方法证明旋转数列的收敛性结论.

2）考察旋转数列的收敛速度与初始值选取的相关性？

3.4.3 海伦数列

如下数列就是由海伦(Helen)映射产生的迭代数列，称为海伦数列：

$$x_{n+1} = x_n \cos a - (y_n - x_n^2) \sin a,$$

$$y_{n+1} = x_n \sin a + (y_n - x_n^2) \cos a, n = 0, 1, \cdots$$

该二元函数迭代有两个不动点,其中之一为(0,0).

对于不同参数 a,迭代数列的收敛性如何呢? 是否有类似于逻辑斯谛迭代的周期性振荡、分岔与混沌现象存在呢?

对于参数 $a = 1.4$,迭代的收敛性如何呢? 实验发现:对于除不动点以外的任何初值(x_0, y_0),产生的迭代数列(x_n, y_n)都不收敛. 数列变化呈现两种模式:有界振荡或者趋于无穷大. 但是没有发现周期性振荡现象.

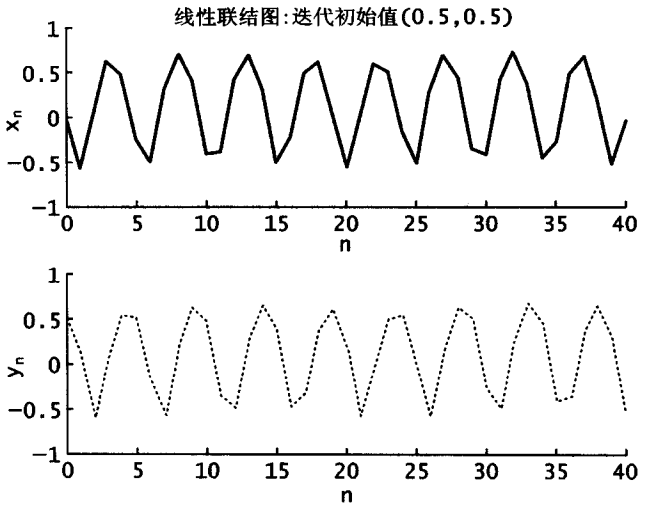


图 3.10　二元迭代的线性联结图:海伦数列

图 3.10 是海伦数列当参数 $a = 1.4$ 时的线性联结图,图中去掉了迭代产生的前 10 000 项. 可以看出迭代不收敛,但是是否呈现周期性振荡呢? 由该线性联结图难以确定.

图 3.11 给出了海伦数列当 $a = 1.4$ 时的 2 - 维散点图:将迭代产生的二维数列(x_n, y_n)直接描绘在平面直角坐标系 xOy 中. 图 3.11 含四幅图,它们是由迭代初始值为(0.5,0.5),去掉迭代产生的前 10 000 项之后的 40 项、80 项、400 项、4 000 项对应的散点图. 也看不出周期性振荡规律,但是其迭代是有界.

非常有趣的是海伦迭代数列沿着一条封闭的光滑曲线振荡! 称这条封闭曲线为振荡的轨道,因此,海伦迭代数列呈现有轨振荡现象,轨道的形状与初始值的选取有关,如图 3.11 和图 3.12.

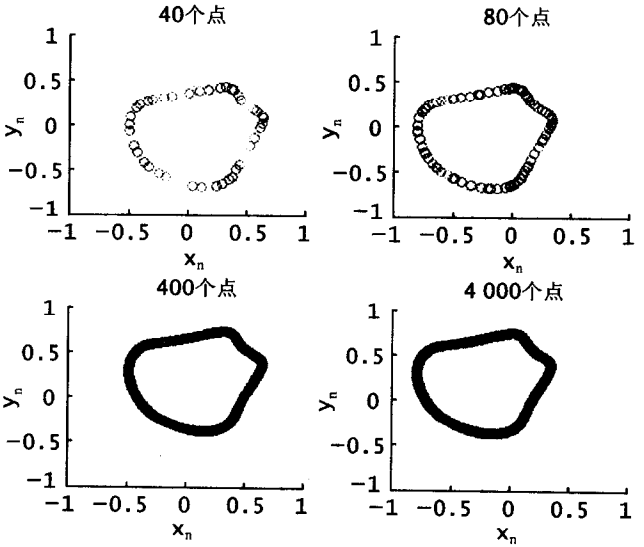


图 3.11　散点图:海伦数列,参数 $a=1.4$,迭代初始值为(0.5,0.5)

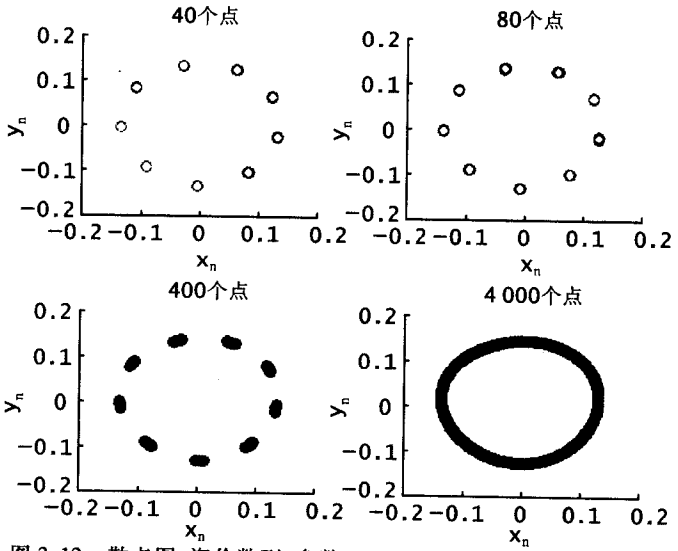


图 3.12　散点图:海伦数列,参数 $a=1.4$,迭代初始值为(0.1,0.1)

1）对于参数 $a=1.4$,海伦数列对于不同初始值可能出现有界振荡和无穷发散现象.对于参数的其他取值,迭代数列是否会出现周期性振荡现象?

2）能否找出有轨振荡的轨道方程?

3）迭代数列的有轨振荡与城市环城地铁运行有何联系?

4）迭代数列的混沌现象与微粒的布朗运动有何联系?

§3.5 操　　练

操练一　迭代与分岔

对于非线性函数 $f(x)=ax(1-x)$ 的迭代：

1）对于参数 a 分别取值于[1,4],[3,4],[3.828 4,4],作出费根鲍姆图.

2）观察其倍 2 - 周期的分裂现象，尽可能多地给出分裂出现的参数取值.

3）观察其倍 3 - 周期现象，并总结类似倍 2 - 周期的规律.

4）观察其倍 5 - 周期现象.

注意：选取同一个迭代初值，去掉前面若干项；将参数 a 的取值间距尽量地减小，以便于发现和总结规律.

操练二　迭代与分形

1）对于非线性函数 $f(z)=z^2-1$，在复数平面上迭代过程：作出其迭代有界的初值点集，就是所谓的 Julia 集.

注意：迭代产生的（复数）数列可能有界，也可能无界，这完全依赖于迭代初值的选取. 初值可以在整个复平面上任意选取. 我们可以根据初值产生的迭代数列有界与否，将复平面上的点划分为两类：其中之一称为“迭代有界初值点集”，用实心黑点代表这些点，观察其几何形状，特别是其边缘的几何性质.

2）对于非线性函数 $f(z)=z^2+c$，参数 c 在复平面取值. 对于每一个复数平面上的参数值，迭代产生的 Julia 集（迭代有界的初值点集）可能连通，也可能不连通. 其中 Julia 集连通的参数取值的集合，就是所谓的 Mandelbrot 集.

具体地：对参数 c 的一个取值，例如 $c=-1$，可以得到一个 Julia 集，这个点集可能连通，也可能不连通. 由于参数 c 的取值范围是整个复数平面，因此，参数 c 取值的复平面就可以根据迭代有界初值点集连通与否划分为两类. 其中之一称为“Julia 集连通的参数点集”，用实心黑点表示这些点，观察其几何形状，特别是其边缘的几何性质.

提示：快速确定 Mandelbrot 集的方法：对于一个 c，如果迭代对初值 $z=0$ 产生的迭代数列是有界的，那么这个 c 就是属于 Mandelbrot 集的.

操练三　迭代与混沌

使用牛顿方法求解非线性方程，自然希望找到好的初始点，能够快速地收敛到某个特定的根. 根是一个**吸引子**，相应有一个该吸引子的**控制域**（收敛到该吸引子的初值范围）. 利用计算机可以很方便地作出所有的吸引子与其控制域的图形，加上那些不是任何一个控制域的点，就构成了整个初值空间. 这样的图形，不妨称为**初值空间谱图**.

1）对于用牛顿方法，只求实数根，作出初值空间谱图，当然应该是一维的.

2）对于用牛顿方法求所有根，也就是包括复数根，初值可以是任何一个复数，其初值空间谱图是二维的. 试作出其初值空间谱图.

3）对问题2，作出彩色的初值空间谱图. 注：用彩色替代黑色的点的方法是：不同的（吸引子的）控制域用不同颜色，并用颜色的暗淡（不同强度）表示收敛速度（指牛顿算法以此点为初值的迭代过程的收敛快慢程度）. 可以先想一想，这个图色彩形状如何？五彩缤纷、千奇百怪、还是平淡无奇，很难想到，除非你自己亲自动手！

注意：可以采用如下两个方程进行实践：

$$(1)\ z^4 - 1 = 0; (2)\ z^3 - 1 = 0.$$

更多的相关信息资源

1 COMAP. Principles and practice of mathematics. 中译本. 北京：高等教育出版社，Springer-Verlag，1997

2 傅鹂，龚劬，刘琼荪，何中市. 数学实验. 北京：科学出版社，2000

3 姜启源. 数学模型. 第二版. 北京：高等教育出版社，1993

4 萧树铁主编，姜启源，何青，高立编著. 数学实验. 北京：高等教育出版社，1999

5 任善强，雷鸣. 数学模型. 重庆：重庆大学出版社，1996

6 D. Quinney. An Introduction to the Numerical Solution of Differential Equations. New York：John Wiley & Sons Inc.，1985

7 G. Fulford. Modelling with differential and difference equations. Cambridge：Cambridge University Press，1997

8 Peitgen, Heinz-Otto. Chaos and fractals：new frontiers of science. New York：Springer-Verlag，1992

9 何良材，何中市. 经济应用数学. 第二版. 重庆：重庆大学出版社，1999

第 4 章

种群数量的状态转移

——微分方程

刻画世界千变万化的规律，微分方程是最有力的工具；求解微分方程特征值问题，数值迭代为最直接的方法；表示数值迭代结果与定性分析结论，几何图形乃最直观的方式，它们的结合在哪里？

——作者

一个国家或地区人口的数量，一个区域内生物链中物种的数量，一个网络系统元件的运行状态等等，他们都在发生状态转移. 这种状态转移可能与很多因素有关，如历史状态及时间间隔等系统内部的因素、控制变量等外部因素，集中体现为变化率（也就是某个物理量的导数）. 从而需要根据物理的、非物理的原理或规律，作出适当的假设，建立变量、变量导数之间的一种平衡关系，也就是微分方程模型. 微分方程是研究变化规律的有利工具，在科技、工程、经济管理、生态、环境、人口等各个领域中有着广泛的应用.

如何求解微分方程？高等数学介绍了很多求微分方程解析解的方法，但实际问题是绝大多数微分方程难以求出、甚至根本求不出解析解，这就需要求助于其他解决方法，如数值解法、图形方法乃至微分方程的定性分析方法.

本章将对人口变化、动物种群变迁、网络系统的可靠性分析，介绍微分方程（组）的模型建立、数值解和图形解等方法，并用 MATLAB 几何直观地展示各种求解方法的求解结果.

§4.1 人口问题

在 18 世纪末，英国人马尔萨斯（Malthus，1766—1834）在他出版的一本专著中，对人口数量的增长趋势进行了模拟，提出了人口的指数增长模式，导致最后会出现人口数量超过地球的承受容量，即人口爆炸问题. 虽然马尔萨斯人口模型忽略了一些人口增长的重要因素，但是他为以后人口模型的改进提供了基础. 下

面将介绍著名的马尔萨斯人口模型及其改进.

马尔萨斯认为:单位时间内,人口的出生数量和死亡数量与人口总数成比例,即人口出生率和死亡率都是常数,因此人口的净增长率为常数.

设时刻 t 的人口数量为 $p(t)$,人口出生率为 b,死亡率为 d,则有:

$$p(t+\Delta t)=p(t)+bp(t)\Delta t-dp(t)\Delta t,$$

从而

$$\frac{\mathrm{d}p(t)}{\mathrm{d}t}=bp(t)-dp(t)=kp(t),$$

其中:$k=b-d$ 称为净增长率(常数).因此马尔萨斯人口模型如下:

$$\frac{\mathrm{d}p(t)}{\mathrm{d}t}=kp(t),\quad p(t_0)=p_0.$$

利用高等数学的知识可以得出该微分方程初值问题的解析解:

$$p(t)=p_0\mathrm{e}^{k(t-t_0)},\tag{4.1}$$

这就是马尔萨斯的人口指数增长模型.为了对该模型进行验证,这里采用某国家的人口历史数据,见表 4.1:

表 4.1　1790—2000 年间某国的人口记录

年份	1790	1800	1810	1820	1830	1840	1850	1860	1870	1880	1890
人口/10^3	3 929	5 308	7 240	9 638	12 866	17 069	23 192	31 433	38 558	50 156	62 948
年份	1900	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000
人口/10^3	75 995	91 972	105 711	122 755	131 669	150 697	179 323	203 212	226 505	248 710	281 416

首先需要确定参数 k:因为 $\ln\dfrac{p(t)}{p_0}=k(t-t_0)$,所以

$$k=\frac{\ln(p(t)/p_0)}{t-t_0}.$$

利用 1790 年和 1840 年的数据计算得出:

$$k=\frac{\ln(17\ 069/3\ 929)}{1\ 840-1\ 790}=0.029\ 4.$$

由此预测,1850 年的人口数量为 22 898 000,误差为 1%,1900 年的人口数量为 99 476 000,误差为 31%,2000 年的人口数量为 1 877 463 000,误差为 567%,2050 年将达到 80 多个亿?我们不得不承认该模型对于长期预测不合理.

1)马尔萨斯人口模型为什么不能适合长期预测?

2)如何改进马尔萨斯人口模型,使其适合长期预测?

3)如果改进后的模型根本求不出解析解怎么办?

§4.2 微分方程的数值解法

前面提到的马尔萨斯人口模型及其改进,都涉及微分方程及微分方程求解.这里只讨论一阶微分方程初值问题:

$$\frac{dy}{dx}=f(x,y),y(x_0)=y_0 \tag{4.2}$$

或一阶微分方程组初值问题:

$$\dot{x}(t)=f(t,x),x(t_0)=x_0, \tag{4.3}$$

其中 $x(t)$ 为向量函数,t 为自变量.

如何有效地求解微分方程?虽然用解析法能够求解一些特殊类型的微分方程,但对于绝大多数微分方程的求解却难以胜任.事实上,在实际微分方程(组)初值问题求解问题中,数值逼近法,又称为数值解法是一种非常有效的方法.

数值解法的基本原理如下:

引入自变量取值点列 $\{x_n\}$,定义 $h_n=x_n-x_{n-1}$,称 h_n 为步长,常用等间距步长(h_n 与 n 无关,不妨记为 h),假设精确解为 $\{y(x_n)\}$.为了寻求 $y(x_n)$ 的近似值 y_n,根据一定的原理,结合当前得到的近似解,近似地表示该点或前一点的导数值,由此推出计算 y_n 的迭代公式.

注意 数值解法一般只能得到微分方程的近似解 $\{y_n\}$.

4.2.1 欧拉方法

欧拉方法是一种简单的求解初值问题的数值逼近方法,其基本思路为:

对于方程(4.2),在小区间 $[x_n,x_{n+1}]$ 上,用差商 $\frac{y(x_{n+1})-y(x_n)}{h}$ 代替导数 y',用左端点 x_n 替换右端函数 $f(x,y)$ 中的 x,得到方程(4.2)的近似表达式如下:

$$y(x_{n+1})\approx y(x_n)+hf(x_n,y(x_n)),$$

设 $y(x_n)\approx y_n$,则 $y(x_{n+1})$ 的近似值为

$$y_{n+1}=y_n+hf(x_n,y_n),n=0,1,\cdots, \tag{4.4}$$

其中初始点为 (x_0,y_0).公式(4.4)被称为**显式欧拉公式**,也称为**向前欧拉公式**.

若在小区间 $[x_n,x_{n+1}]$ 上用差商代替导数 y' 后,用右端点 x_{n+1} 替换右端函数 $f(x,y)$ 中的 x,考虑到 $y(x_n)\approx y_n$ 可得另一个欧拉公式如下:

$$y_{n+1}=y_n+hf(x_{n+1},y_{n+1}),n=0,1,\cdots \tag{4.5}$$

称为**隐式欧拉公式**，也称**向后欧拉公式**. 这是一个非线性方程，无法直接计算 y_{n+1}.

向前欧拉法简单易于计算，但精度却不高，收敛速度慢. 向后欧拉法与向前欧拉法在计算精度、收敛速度方面相同，但计算起来比较困难.

如果将向前和向后欧拉公式(4.4)和(4.5)加以平均，则得到

$$y_{n+1}=y_n+\frac{1}{2}h[f(x_n,y_n)+f(x_{n+1},y_{n+1})],n=0,1,\cdots, \tag{4.6}$$

(4.6)式被称为**梯形公式**. 该法的计算精度均比向前和向后欧拉法要高，而且收敛速度快. 但迭代计算与向后欧拉法一样繁，因此产生了改进欧拉方法.

改进欧拉方法是将迭代过程简化为两步：先由向前欧拉公式(4.4)算出 y_{n+1} 的预测值 $\bar{y}_{n+1}$；再将它代入梯形公式(4.6)式的右端，作为校正，即

$$\bar{y}_{n+1}=y_n+hf(x_n,y_n),$$

$$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},\bar{y}_{n+1})],n=1,2,\cdots,$$

称为**改进欧拉公式**，它还可写作：

$$\begin{cases} y_{n+1}=y_n+\dfrac{h}{2}(k_1+k_2), \\ k_1=f(x_n,y_n), \\ k_2=f(x_{n+1},y_n+hk_1), \end{cases} \tag{4.7}$$

人们常用的是向前欧拉公式(4.4)和改进欧拉公式(4.7). 并且欧拉法可以推广到求解微分方程组(4.3)的情形.

现在用上面介绍的系列欧拉方法求解以下微分方程初值问题：

$$y'=-y+x+1,y(0)=1.$$

要求：取步长 $h=0.1$ 和 0.001. 分别用三种数值解法求解，并结合其精确解，对求解误差进行分析比较.

首先用解析法得到其精确解 $y=x+e^{-x}$.

其次用数值解法(取 $h=0.1$)

向前欧拉法 迭代公式为 $y_{n+1}=0.9y_n+0.1x_n+0.1,n=0,1,\cdots,y_0=1.$

向后欧拉法 迭代公式为 $y_{n+1}=y_n+0.1(-y_{n+1}+x_{n+1}+1)$，将它变形为

$$y_{n+1}=(y_n+0.1x_n+0.11)/1.1,n=0,1,\cdots,y_0=1.$$

梯形法 将隐式梯形公式转化为显示迭代公式

$$y_{n+1}=(0.95y_n+0.1x_n+0.105)/1.05,n=0,1,\cdots,y_0=1.$$

计算结果如表 4.2，当 $h=0.001$ 时的计算结果如表 4.3 所示.

表 4.2　数值解，步长 $h=0.1$

x_n	精确解	向前欧拉法	向后欧拉法	梯形法
0	1	1	1	1
0.1	1.004 8	1	1.009 1	1.004 8
0.2	1.018 7	1.010 0	1.026 4	1.018 6
0.3	1.040 8	1.029 0	1.051 3	1.040 6
0.4	1.070 3	1.056 1	1.083 0	1.070 1
0.5	1.106 5	1.090 5	1.120 9	1.106 3
0.6	1.148 8	1.131 4	1.164 5	1.148 5
0.7	1.196 6	1.178 3	1.213 2	1.196 3
0.8	1.249 3	1.230 5	1.266 5	1.249 0
0.9	1.306 6	1.287 4	1.324 1	1.306 3
1	1.367 9	1.348 7	1.385 5	1.367 3

表 4.3　数值解，步长 $h=0.001$

x_n	精确解	向前欧拉法	向后欧拉法	梯形法
0	1	1	1	1
0.1	1.004 8	1.004 8	1.004 9	1.004 8
0.2	1.018 7	1.018 6	1.018 8	1.018 7
0.3	1.040 8	1.040 7	1.040 9	1.040 8
0.4	1.070 3	1.070 2	1.070 5	1.070 3
0.5	1.106 5	1.106 4	1.106 7	1.106 5
0.6	1.148 8	1.148 6	1.149 0	1.148 8
0.7	1.196 6	1.196 4	1.196 8	1.196 6
0.8	1.249 3	1.249 1	1.249 5	1.249 3
0.9	1.306 6	1.306 4	1.306 8	1.306 6
1	1.367 9	1.367 7	1.368 1	1.367 9

计算结果表明，当步长 $h=0.1$ 时，它们的前两位有效数字是精确的；当步长 $h=0.001$ 时，它们的前四位有效数字是精确的. 说明在迭代中，步长 h 越小，计算结果越精确，并且迭代离开初始点越远，误差越大. 另外，梯形法显然优于向前、向后欧拉法.

4.2.2　龙格 - 库塔方法

龙格 - 库塔方法（简称 R - K 方法）是利用泰勒展式将 $y(x+h)$ 在 x 处展开，并取其前面若干项来近似 $y(x+h)$ 而得到公式

$$y(x+h)\approx y(x)+h\varphi(x,y(x),h).$$

如果 $y(x_n)\approx y_n$，则 $y(x_{n+1})$ 的近似值为：

$$y_{n+1}=y_n+h\varphi(x_n,y_n,h),n=0,1,\cdots.$$

若

$$y(x+h)-[y(x)+h\varphi(x,y(x),h)]=o(h^{p+1}),$$

则称以上迭代公式为 p - 阶公式，p 的大小反映了截断误差的高低，高阶高精度. 要得到一个 p - 阶公式，关键在于如何选取 $\varphi(x,y(x),h)$ 使之满足阶的要求.

常用的 R - K 公式有 2 - 阶、3 - 阶、4 - 阶公式如下：

2 - 阶公式

中点公式

$$y_{n+1}=y_n+k_2,k_2=hf\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}k_1\right),k_1=hf(x_n,y_n).$$

改进欧拉公式

$$y_{n+1}=y_n+\frac{1}{2}(k_1+k_2),k_1=hf(x_n,y_n),k_2=hf(x_n+h,y_n+k_1).$$

3 - 阶公式

$$y_{n+1}=y_n+\frac{1}{6}(k_1+4k_2+k_3),$$

$$k_1=hf(x_n,y_n),k_2=hf\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}k_1\right),$$

$$k_3=hf(x_n+h,y_n-k_1+2k_2).$$

4 - 阶公式

$$y_{n+1}=y_n+\frac{1}{6}(k_1+2k_2+2k_3+k_4),$$

$$k_1=hf(x_n,y_n),k_2=hf\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}k_1\right),$$

$$k_3=hf\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}k_2\right),k_4=hf(x_n+h,y_n+k_3).$$

以上关于微分方程的数值求解方法，可以推广到微分方程组(4.3)的求解. 在 MATLAB 软件中含有数值求解的系统函数，其实现原理就是龙格 - 库塔方法，我们将在 4.4 节中介绍.

§4.3 微分方程图解法

前面介绍的数值解法适应面广，却要损失精度，并且只能得到一些离散点处

解的近似值. 使人感觉到"只见树木,不见森林",即不能展示全局的、具有趋势性的结果. 能否有一种方法兼顾二者之所长,将微分方程解的全局信息形象化、直观化,也许只有图形—图解法能解决这个问题.

4.3.1 斜率场表示微分方程的解

对于一阶微分方程$\frac{dy}{dx}=f(x,y)$,如果给出平面上任意一点(x,y),就能够确定出解$y=f(x)$在该点(x,y)处的斜率$f(x,y)$. 从图像上看,给出平面上的一系列点,通过每一点(x_0,y_0),可以画出一条通过点(x_0,y_0)、斜率为$f(x_0,y_0)$的短直线. 这样的短直线布满整个坐标平面,形成的图形就称为**斜率场**或**方向场**.

另一方面若能求出微分方程的通解

$$y=\int g(x,y(x))\,dx+C,$$

将得到一簇积分曲线,对不同的初始值,对应于不同的解曲线. 在这些曲线簇中任意取出一条,进行如下处理:

1）曲线分小段(**分**)　将曲线分割成若干段首尾相连的弧线段;

2）以直代曲线(**替**)　将每一段弧线用短的直线段代替,这条短直线的斜率方向是曲线段左端点处的斜率;

3）相邻不相连(**短,断**)　因为直线段足够地短,使相邻直线段不相交,就可以得到微分方程的斜率场.

也就是说,由微分方程的解曲线可以得到斜率场,反之,由斜率场可以隐约看出解曲线的走向,可以将斜率场看作一组路标,它指明了每一点处所要走的方向. 若要得到初值问题的解曲线就从平面上初始点出发沿该点所指明的方向前进,按方向一段接一段地走下去,……,将会弯弯曲曲地走出一条"路". 从几何角度上讲,微分方程求解本身就意味着找解曲线,在图形上显示这样的解既全面又直观.

想　能否不使用积分运算求解微分方程,而直接得到斜率场图形? 用什么方法呢?

用斜率场求解下列微分方程$y'=\sin x\sin y$的MATLAB程序如下:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    clear all;       syms x y
    f = sin(x) * sin(y);
    a = 18;b = 18;
    x0 = -9;y0 = -9;
```

```
m = 40;n = 40;h1 = a/m;h2 = b/n;
hold on
for i = 1:m
    x = x0 + (i - 1) * h1;
    for j = 1:n
        y = y0 + (j - 1) * h2;d = eval(f);y1 = y + 2/3 * h1 * d;
        if abs(y1 - y) > 2/3 * h2
            x1 = x + 1/d * h2 * 2/3;
              plot([x,x1],[y,y + h2 * 2/3],'k')
        else
              plot([x,x + h1 * 2/3],[y,y1],'k')
        end
    end
end
title('斜率场:dy/dx = sinx * siny');xlabel('x');ylabel('y');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

输出斜率场图如图 4.1

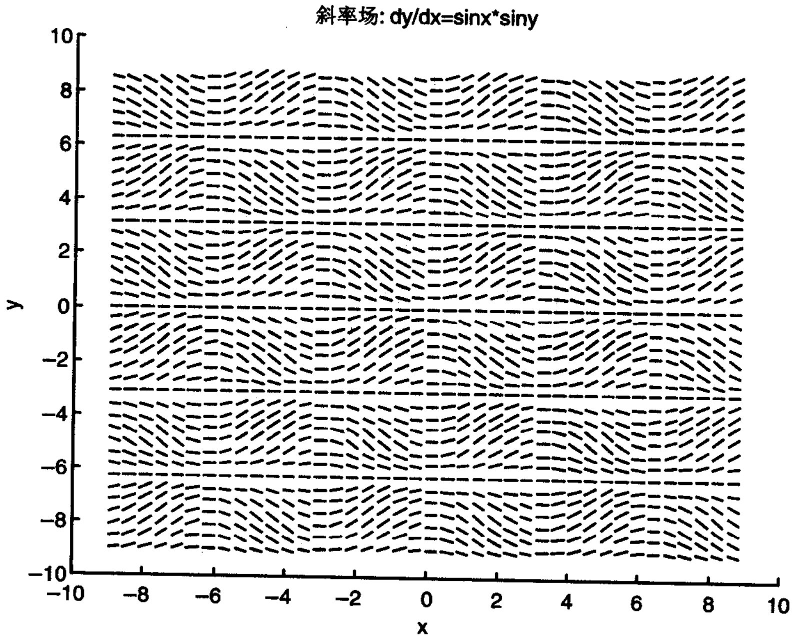


图 4.1　斜率场表示的微分方程的解

1）图 4.1 是否可以看出该微分方程的通解或者积分曲线簇的形状？

2）是否能够找出满足初值条件 $y(-8)=2$ 的那条积分曲线？

3）能否将斜率场用于求解微分方程组？如何应用？

斜率场的另一个应用是对那些不能用初等函数表示的积分，可作出它们的积分曲线.

4.3.2 相平面轨迹表示微分方程的解

斜率场能够非常直观地表示一元微分方程的特解及通解. 对于两个未知函数的微分方程组，比如

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=y+x-x(x^2+y^2),$$

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t}=y-x-y(x^2+y^2),$$

$$x(t_0)=x_0,y(t_0)=y_0.$$

能否直观地表示出其解呢？

利用微分方程的数值解法（比如 4 - 阶 R - K 方法），可以得到其数值解：$(x(t),y(t))$ 在 t 取离散值 T（可以看成列向量）时的列向量 $\boldsymbol{X},\boldsymbol{Y}$；然后分别独立地作出函数 $x(t)$ 和 $y(t)$ 的曲线，如图 4.2，其初值条件为(5,5).

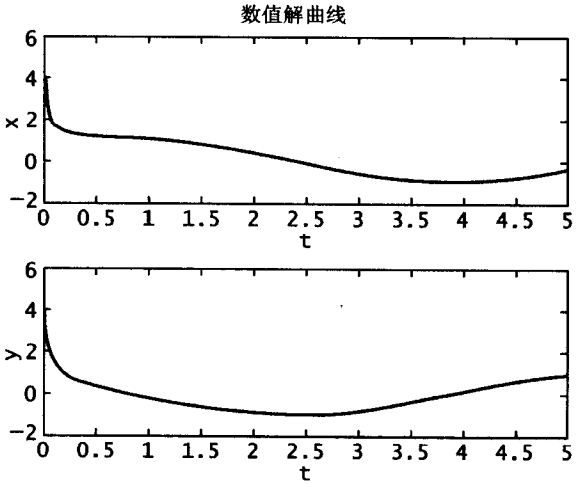


图 4.2 数值解曲线

从这个图中难以直观地发现两个函数的相对变化趋势，为此，人们设法将

$x(t)$和$y(t)$的曲线组合在一起. 如果撇开自变量的取值 T,直接利用 $\boldsymbol{X},\boldsymbol{Y}$ 的分量作为坐标(需保持由 T 建立的对应关系),就可以在 xOy 平面上画出解的轨迹,称为**相平面轨迹图**,如图 4.3.

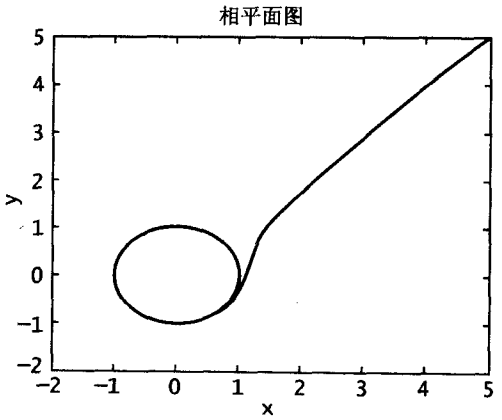


图 4.3　通过数值解得到的相平面轨迹图

利用斜率场作图　将以上方程组的自变量 t 消去,得到的 y 与 x 之间的函数关系为

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\frac{y-x-y(x^2+y^2)}{y+x-x(x^2+y^2)},$$

它是一个一阶微分方程. 由前一小节,在平面直角坐标系中可以作出斜率场图形如图 4.4.

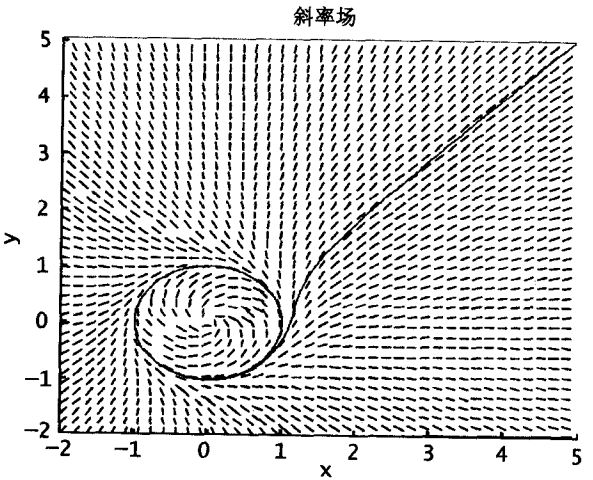


图 4.4　通过斜率场得到的相平面轨迹图

1）随着时间的推移，相平面轨迹图中轨迹的走向按逆时针还是顺时针方向？

2）两种方法作出的相平面轨迹图形在形式上似乎有一些差别，原因何在？本质上是否一致？

§4.4 MATLAB 软件求解

前面已经介绍了微分方程(组)的求解方法——数值解法和图解法. 这些方法计算工作量大，需要借助于计算机实现. 下面简单介绍 MATLAB 软件在此领域的应用.

4.4.1 解析解

求微分方程(组)的解析解的 MATLAB 命令

```
dsolve('eqn1','eqn2',…,'x')
```

其中'eqni'表示第 i 个方程，'x'表示微分方程(组)中的自变量，默认时自变量为 t. 举例如下：

求微分方程 $dy/dx = 1 + y^2$ 的通解

输入：

```
dsolve('Dy = 1 + y^2')
```

输出：

```
ans = tan(t - C1)
```

求微分方程 $dy/dx = 1 + y^2, y(0) = 1$ 的特解

输入：

```
dsolve('Dy = 1 + y^2','y(0) = 1','x')
```

输出：

```
ans = tan(x + 1/4 * pi)
```

求解二阶微分方程

$$x^2y'' + xy' + (x^2 - n^2)y = 0, y(\pi/2) = 2, y'(\pi/2) = -2/\pi, n = 1/2$$

输入：

```
dsolve('D2y + (1/x) * Dy + (1 - (1/2)^2/x^2) * y = 0','y(pi/2) = 2,Dy(pi/2) =
-2/pi','x')
```

输出：

```
ans =2^(1/2)*pi^(1/2)*sin(x)/x^(1/2)
```

化简输出结果，输入：

```
pretty(ans)
```

计算结果为：$$y=\sqrt{\frac{2\pi}{x}}\sin x.$$

求解微分方程组 $df/dx=3f+4g;dg/dx=-4f+3g.$

求通解

输入：

```
[f,g]=dsolve('Df=3*f+4*g','Dg=4*f+3*g')
```

输出：

```
ans
f = 1/2*(C1*exp(-8*t)+C1+C2-C2*exp(-8*t))*exp(7*t)
g = -1/2*(-C1+C1*exp(-8*t)-C2*exp(-8*t)-C2)*exp(7*t)
```

求特解

输入：

```
[f,g]=dsolve('Df=3*f+4*g','Dg=4*f+3*g','f(0)=0,g(0)=1')
```

输出：

```
ans
f = 1/2*exp(7*t)-1/2*exp(-t)
g = 1/2*exp(-t)+1/2*exp(7*t)
```

4.4.2 数值解

设微分方程的形式为 $y'=f(t,y)$，其中 t 为自变量，y 为因变量（变量 y 可以是向量，例如微分方程组）.

在 MATLAB 6.1 版本中，用 2 阶（3 阶）龙格－库塔公式和 4 阶（5 阶）龙格－库塔公式的程序分别为

```
[t,y]=ode23('F',ts,y0,options)
[t,y]=ode45('F',ts,y0,options)
```

其中 F 是由微分方程（组）写成的 M 文件名，输入 ts 的取法有几种，当 ts=[t0,tf]，t0，tf 分别表示自变量的初值和终值，若 ts=[t0,t1,t2,…,tf]，则输出在指定时刻 t0，t1，t2，…，tf 处给出；对于等步长时用 ts=t0：k：tf；则输出在区间[t0,tf]的等分点给出.y0 为函数的初值；options 用于设定误差限（可以缺省，缺省时设定相对误差是 10^{-3}，绝对误差 10^{-6}）. 程序为

```
options=odeset('reltol',rt,'abstol',at)
```

这里的 rt 和 at 分别为设定的相对误差和绝对误差.

[t,y]为输出矩阵,分别表示自变量 t 和因变量 y 的取值.

ode23 是微分方程组数值解的低阶方法,ode45 为较高阶方法,与 ode23 类似.另外还有一些其他方法,如求解非线性微分方程组的可变阶方法 ode113.举例如下:

求微分方程 $x^2y''+xy'+(x^2-n^2)y=0$ 的数值解

令 $y_1=y, y_2=y_1'$,则将二阶微分方程:

$$x^2y''+xy'+(x^2-n^2)y=0$$

转化为一阶微分方程组: $y_1'=y_2$,

$$y_2'=-y_2/x+((n/x)^2-1)y_1,$$

当 $n=1/2, y_1(\pi/2)=2, y_2(\pi/2)=-2/\pi$ 时,用 MATLAB 求解如下:

首先建立 M 文件函数

```
function f=eqs2(x,y)
f=[y(2);-y(2)/x+((1/2)^2/x^2-1)*y(1)];
```

输入 M 文件:

```
[x,y]=ode23('eqs2',[pi/2,pi],[2,-2/pi])
plot(x,y(:,1),x,y(:,2),'r'),
xlabel('x');gtext('y1'),gtext('y2')
```

将输出结果作图,如图 4.5 所示.图中曲线 y1 就是该二阶微分方程的解.

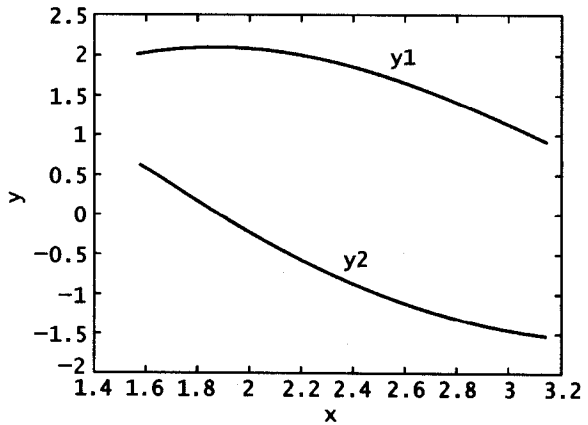


图 4.5 数值解图形

§4.5 微分方程的应用

4.5.1 逻辑斯谛人口模型

针对马尔萨斯人口模型的不足,1837 年荷兰生物数学家 Verhulst(1804—1849)提出了如下改进:

由于资源的限制,人口存在最大值(极限)M. 因此,人口增长率不应该是常数,假设增长率 k 是随着人口数量接近 M 而线性递减:

$$k = r(M - p),$$

从而得到改进后的人口模型为

$$\frac{\mathrm{d}p(t)}{\mathrm{d}t} = kp = r(M - p)p, p(t_0) = p_0,$$

称以上改进模型为逻辑斯谛增长模型.

可以求得,该人口模型的解为:

$$p(t) = \frac{Mp_0}{p_0 + (M - p_0)\mathrm{e}^{-rM(t - t_0)}},$$

这个函数就是著名的逻辑斯谛曲线.

1) 如何得到逻辑斯谛人口模型中的参数 M?

2) 逻辑斯谛人口模型是否适合人口长期预测?

估计参数 M 的方法之一:利用第 6 章介绍的曲线拟合.

4.5.2 弱肉强食的双种群模型

生活在同一环境中的两种动物,存在各种各样的生存模式,弱肉强食就是其中典型的一种行为. 设想在一个海岛上,居住着野兔和狐狸,狐狸吃兔子,兔子吃青草. 青草无限,兔子大量繁殖;兔多则狐狸容易捕食,狐狸数量增加,导致兔子减少. 进而狐狸将减少. 形成兔子狐狸数量交替增减,无休无止地循环,形成生态平衡. 著名的意大利数学家 Volterra 建立了如下微分方程组模型:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = r \cdot x(t) - a \cdot x(t) \cdot y(t),$$

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = -s \cdot y(t) + b \cdot x(t) \cdot y(t),$$

其中 $x(t)$, $y(t)$ 分别表示 t 时刻兔子、狐狸的数量. 模型解释为：对于兔子，项 $rx(t)$ 表示兔子的繁殖数量，项 $ax(t)y(t)$ 表示兔子的减少与兔子狐狸相遇次数成比例. 对于狐狸，项 $-sy(t)$ 表示同类竞争造成死亡的数量，项 $bx(t)y(t)$ 表示狐狸的增加与兔子狐狸相遇次数成比例.

考虑以下三种方法作出其相平面轨迹图.

对于以上二元微分方程组，当然可以采用数值解法. 下面将数值解与图形结合的方法给出该生态系统中狐兔数量变化的轨迹.

1. 利用数值解作图

直接用微分方程组的数值解法求出数值解 $(x(t), y(t))$，再画由数值解产生的系统的相平面轨迹，如图 4.6.

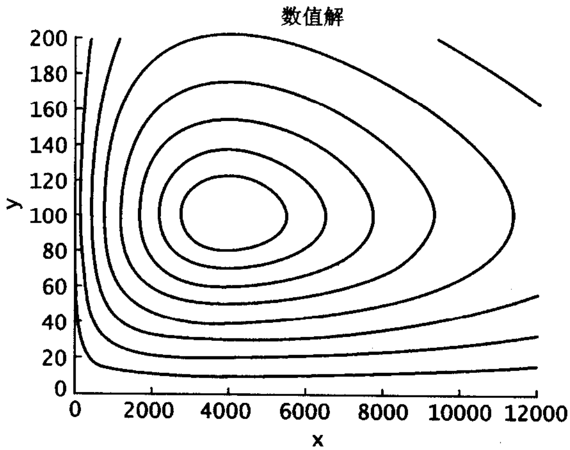


图 4.6 通过数值解得到的相平面轨迹图

注意：为了得到数值解，必须给出适当的初值条件. 同时，为了得到多条相平面轨迹，便于和其他方式得到的相平面轨迹图进行比较. 这里我们给出了系列初值下的解及其相平面轨迹图，图 4.6 中，具体的初值包括 $[x(0), y(0)] = [1, 0.3]$, $[1, 0.5]$, …, $[1, 1.5]$ 等. 相应的 MATLAB 程序如下：

首先，建立 M 函数文件(fish. m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  function xdot = fish(t,x)
  r = 2; s = 0.8;a = 0.02;b = 0.0002;
  xdot = [r * x(1) - a * x(1) * x(2); - s * x(2) + b * x(1) * x(2)];
  % 由数值解作相平面轨迹图
  hold on
  for k = 1:8
```

```
    ts =1:0.01:8;x0 =[4000,10 * k];
    [t,x] =ode45('fish',ts,x0);
    plot(x(:,1),x(:,2),'k')
  end
  xlabel('x');ylabel('y');title('数值解')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2. 利用斜率场作图

将以上方程组的自变量 t 消去,得到的 y 与 x 之间的函数关系为

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\frac{-sy+bxy}{rx-axy},$$

它是一个一阶微分方程. 由前一小节,在平面直角坐标系中可以作出斜率场图形如图 4.7(当参数是 $r=2,s=0.8,a=0.02,b=0.000\ 2$ 时的情形).

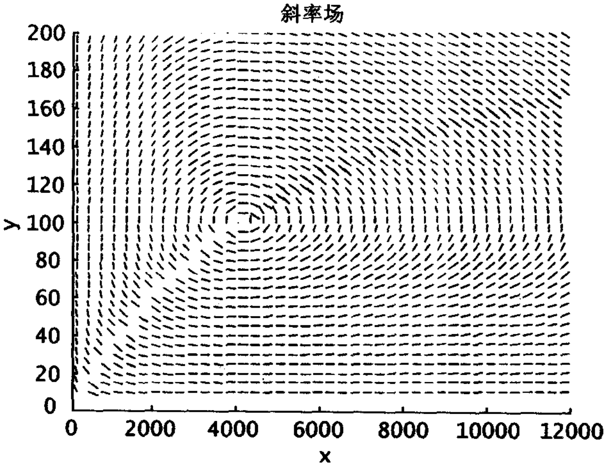


图 4.7　通过斜率场得到的相平面轨迹图

3. 利用等值线作图

将以上微分方程通过分离变量方法得到其通解为

$$r\ln y-ay+s\ln x-bx=K,$$

其中 K 为任意常数. 如果给定了初始条件 (x_0,y_0),则 K 为定值. 即

$$K=r\ln y_0-ay_0+s\ln x_0-bx_0.$$

反之,对于任何一个给定值 K,也就确定了一个特解. 因此,微分方程组的通解实质上是如下二元函数的等值线

$$f(x,y)=r\ln y-ay+s\ln x-bx.$$

下面给出当参数取值 $r=2,s=0.8,a=0.02,b=0.000\ 2$ 时,由等值线作出的相平面轨迹图,如图 4.8. MATLAB 程序如下:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x0 =100;x1 =12000;y0 =10;y1 =200;
[x,y] =meshgrid(x0:10:x1,y0:1:y1);
r =2; s =0.8;a =0.02;  b =0.0002;
z =r * log(y) -a * y +s * log(x) -b * x;
contour(x,y,z,20,'k');
axis([0 x1 0 y1]);  xlabel('x');ylabel('y');
title('等值线')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

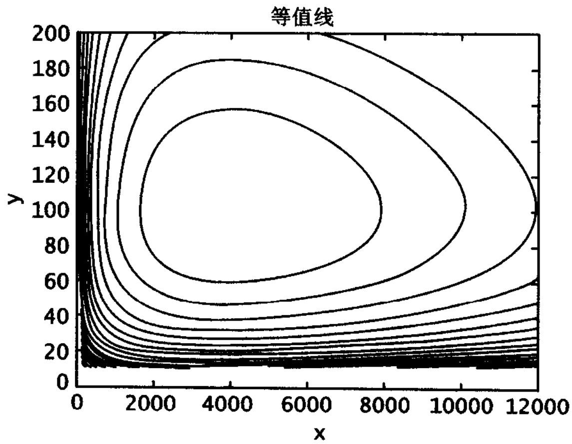


图 4.8　通过等值线得到的相平面轨迹图

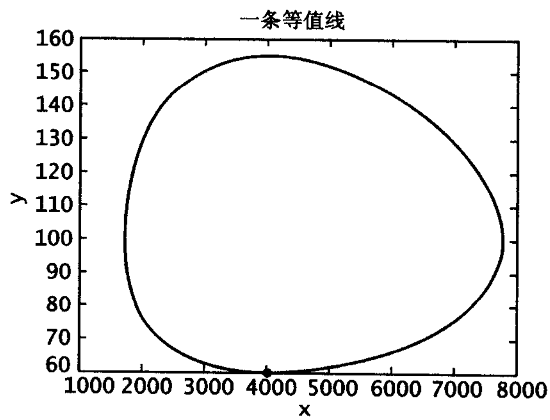


图 4.9　初始值为 x = 4000;y = 60 对应的相平面轨迹

从图 4.9 中可以看出,如果初始参数值为 x0 = 4 000 兔子,y0 = 60 狐狸,其

两种动物的种群数量变化如图中那条封闭曲线.

1）随着时间的推移，在相平面轨迹图 4.8 中，轨迹的走向是按逆时针还是顺时针方向？

2）如果某生态系统中有三种弱肉强食的动物，相应的种群模型为三元微分方程，如何用图形表示其解的轨迹？

4.5.3 计算机网络可靠性分析问题

随着计算机通信网络系统特别是 Internet 网络日益广泛的应用，提高系统的可靠性，意义重大. 研究和分析具有实用性的高可靠计算机通信网络系统，是国际上非常活跃的一个研究方向.

1. 问题及假设

我们将研究的计算机通信网络系统称为**无冗余的防火墙协议系统**. 计算机随时可能发生三个事件——无故障 、间歇故障和永久故障. 因此，计算机一般处于三种运行状态：无故障工作、带故障工作和不工作. 这三种状态之间的转移过程如图 4.10 所示. 要求建立该系统的状态转移模型，并进行可靠性分析.

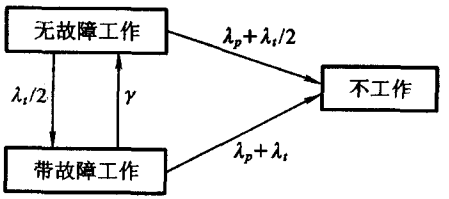


图 4.10 无冗余的防火墙协议系统状态转移过程

2. 分析与模型

该问题属于状态转移问题，利用马尔可夫状态转移原理，用 $P_1(t)$，$P_2(t)$ 和 $P_3(t)$ 分别表示系统处于无故障工作、带故障工作和不工作三种状态的概率，则有以下状态转移方程组

$$\frac{dP_1(t)}{dt} = -(\lambda_p+\lambda_t)P_1(t)+\gamma P_2(t),$$

$$\frac{dP_2(t)}{dt} = (\lambda_t/2)P_1(t)-(\gamma+\lambda_p+\lambda_t)P_2(t),$$

$$\frac{dP_3(t)}{dt} = (\lambda_p+\lambda_t/2)P_1(t)+(\lambda_p+\lambda_t)P_2(t).$$

初始条件 $[P_1(0),P_2(0),P_3(0)]=[1,0,0]$，参数取值 $\lambda_p: 10^{-5}\sim 10^{-4}$，

λ_t:10^{-4} ~ 10^{-3},γ:0.01 ~ 0.1. 这是一个带参数的微分方程组模型.

3. 模型求解

采用数值解法求解这个带参数的微分方程组模型. 假定模型中的参数取下限值,即 $\lambda_p = 10^{-5}$,$\lambda_t = 10^{-4}$,$\gamma = 0.01$. 数值解的 MATLAB 程序如下:

首先编写 M 文件(eqs3.m)

```
function xdot = eqs3(t,p,flag,lp,lt,gm)
a = [ -(lp + lt) gm 0;lt/2 -(gm + lp + lt) 0; lp + lt/2 lp + lt 0];
p = [p(1);p(2);p(3)];
xdot = a * p;
```

在工作空间执行以下程序:

```
ts = [0 10000];p0 = [1;0;0];lp = 10^( -5);lt = 10^( -4);gm = 0.01;
[t,p] = ode23('eqs0',ts,p0,[],lp,lt,gm);
plot(t,1 -p(:,3));
xlabel('时间 t(小时)');ylabel('可靠度 R(t)');
title('参数取值 lp = 0.00001;lt = 0.0001;gm = 0.01');grid
```

输出结果如图 4.11.

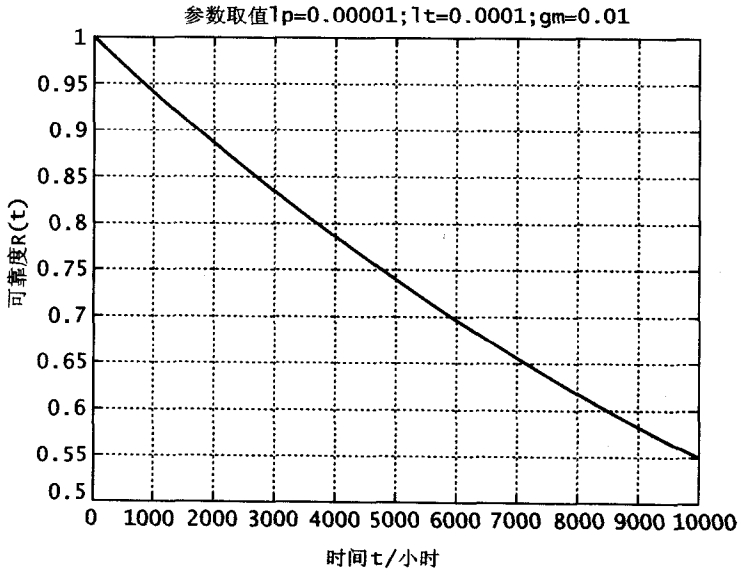


图 4.11 系统正常运行的可靠度曲线

计算结果表明:在时间 t = 1 000 小时情况下,[$P_1(t)$ $P_2(t)$ $P_3(t)$] = [0.936 9 0.004 7 0.058 4]. 显示系统工作的概率为 94.18%. 从图 4.11 中还可

以看出系统可靠性变化更加细致的情况，何时可靠度达到99%，何时达到98%等等.

当参数取值在以下范围 $\lambda_p: 10^{-5} \sim 10^{-4}$，$\lambda_t: 10^{-4} \sim 10^{-3}$，$\gamma$: 0.01 ~ 0.1 内变化时，系统的可靠性如何变化？

§4.6 操　　练

操练一　盐水的混合问题

一个圆柱形的容器，内装350 L的均匀混合的盐水溶液. 如果纯水以14 L/s的速度从容器顶部流入，同时，容器内的混合的盐水以10.5 L/s的速度从容器底部流出. 开始时，容器内盐的含量为7 kg. 求经过时间 t s后容器内盐的含量.

操练二　遗传模型

孟德尔(Mendel)第一定律：配子的基因是从其父倍的两个基因型中随机地选择的.

实际应用中，将比例作为概率：$P_k(\mathrm{A}) = \mathrm{Prob}\{\mathrm{AA}\ 或\ \mathrm{Aa}\}$；$P_k(\mathrm{a}) = \mathrm{Prob}\{\mathrm{aa}\}$，并记 $X_k = P_k(\mathrm{a})$. 得到如下遗传模型：

1) 致死基因遗传模型：$X_{k+1} = \dfrac{X_k}{1+X_k}$. 讨论 X_k 的变化趋势.

2) 自然选择基因遗传模型：$X_{k+1} = \dfrac{(\beta-1)X_k^2 + X_k}{1+(\beta-1)X_k^2}$，其中 $\beta = r_1/r_2$. r_1 和 r_2 分别表示在总人口数量中，新生儿基因为(AA 或 Aa)和(aa)所占的比例. 对不同的 β 取值，讨论 X_k 的变化趋势，选取初值：$X_0 = 0.9$.

3) 突变基因遗传模型：$X_{k+1} = (1-\mu)X_k + \mu$，其中 μ 为 A 突变为 a 的概率(比例一般为：$10^{-6} \sim 10^{-5}$). 对不同的 μ 讨论 X_k 的变化情况？考虑初值 $X_0 = 0.1$.

操练三　老鼠觅食

有一个连续的很多个小老鼠笼子(正方形)，它们首尾相连. 在其前后两边的中央都开有一个洞，可供老鼠自由进出. 并在右边放置鼠粮，左边未放鼠粮. 老鼠在笼子里面只能够沿着笼子边沿(正方形的四条边)沿左边或从右边向前通过. 沿左边则吃不到鼠粮，只有沿右边才能够吃到鼠粮. 在每个鼠笼子里，老鼠随机地选择左右之一向前行进.

1) 奖励型：如果老鼠沿右边吃到鼠粮后，则下次将毫不犹豫地沿右边，如果沿左边未吃到鼠粮，则下次将以 $1-\alpha$ 的概率向左.

2）奖惩兼顾型：如果向右吃到鼠粮后，则下次向右的概率为 $1-\beta$；如果向左未吃到鼠粮，则下次向左的概率为 $1-\alpha$.

就这两种情况，分别建立并求解老鼠在第 n 次进入鼠笼子时向右能够吃到鼠粮的概率. 并考察其无穷趋势.

当然有兴趣的读者可以考虑学习型：设想老鼠具有一定的学习记忆能力，并将其在模型中反映出来.

更多的相关信息资源

1 Frank R. Giordano, Maurice D. Weir, William O. Fox. A first course in mathematical modeling. 3rd edition. 影印版. 北京：机械工业出版社，Thomson Learning, Inc., 2003

2 傅鹂，龚劬，刘琼荪，何中市. 数学实验. 北京：科学出版社，2000

3 COMAP. Principles and practice of mathematics. 中译本. 北京：高等教育出版社，Springer-Verlag, 1997

4 萧树铁主编，姜启源，何青，高立编著. 数学实验. 北京：高等教育出版社，1999

5 姜启源. 数学模型. 第二版. 北京：高等教育出版社，1993

6 John H. Mathews, Kurtis D. Fink. Numerical methods using Matlab. 3rd edition. 英文版. 北京：电子工业出版社，Pearson Education，2002

7 李尚志. 数学建模竞赛教程. 南京：江苏教育出版社，1996

8 D. Quinney. An Introduction to the Numerical Solution of Differential Equations. New York: John Wiley & Sons Inc., 1985

9 G. Fulford. Modelling with differential and difference equations. Cambridge: Cambridge University Press, 1997

10 何良材，何中市. 经济应用数学. 第二版. 重庆：重庆大学出版社，1999

第 5 章
水塔用水量的估计
——插值

对数量变化关系的研究，有两种相反的方式：数学更多地分析由一个函数表达式到数据的变化，因而可以得到很多结果；现实则通常是由数据到数据或表达式，一般需要用很多数据才能得到一些我们所需的数据或表达式. 插值与拟合从不同的角度出发，将数据到函数、函数到数据的研究方式成功地结合起来.

——作者

在工程实践和科学实验中，常常需要从一组实验观测数据 (x_i, y_i)，$i = 0, 1, \cdots, n$，揭示自变量 x 与因变量 y 之间的关系，一般可以用一个近似的函数关系式 $y = f(x)$ 来表示. 函数 $f(x)$ 的产生方式因观测数据与要求的不同而异，通常可以采用两种方法：曲线拟合和插值. 拟合主要是考虑到观测数据受随机误差的影响，寻求整体误差最小、较好反映观测数据的近似函数，并不保证所得到的函数一定满足 $y_i = f(x_i)$. 插值则要求函数在每个观测点处一定要满足 $y_i = f(x_i)$. 拟合的方法将在下一章讨论，本章主要介绍插值方法.

插值函数一般是已知函数的线性组合或者称为加权平均. 插值在工程实践和科学实验中有着非常广泛而又十分重要的应用，例如，信息技术中的图像重建，图像放大中为避免图像扭曲失真的插值补点，建筑工程的外观设计，化学工程实验数据与模型的分析，天文观测数据、地理信息数据的处理（如天气预报）以及社会经济现象的统计分析等等.

§5.1 水塔用水量问题

某居民区的民用自来水是由一个圆柱形的水塔供给，水塔高 12.2 m，直径 17.4 m. 水塔是由水泵根据水塔内水位高低自动加水，一般每天水泵工作两次. 现在需要了解该居民区用水规律与水泵的工作功率. 按照设计，当水塔的水位降

至最低水位，约 8.2 m 时，水泵自动启动加水；当水位升高到一个最高水位，约 10.8 m 时，水泵停止工作.

可以考虑采用用水率（单位时间的用水量）来反映用水规律，并通过间隔一段时间测量水塔里的水位来估算用水率. 表 5.1 是某一天的测量记录数据，测量了 28 个时刻，但是由于其中有 4 个时刻遇到水泵正在向水塔供水，而无水位记录（表 5.1 中用符号//表示）.

试建立合适的数学模型，推算任意时刻的用水率，一天的总用水量和水泵工作功率.

表 5.1　原始数据（单位：时间/h，水塔中水位/m）

时间 t	0	0.921	1.843	2.949	3.871	4.978	5.900
水位	9.677	9.479	9.308	9.125	8.982	8.814	8.686
时间 t	7.006	7.928	8.967	9.981 1	10.925	10.954	12.032
水位	8.525	8.388	8.220	//	//	10.820	10.500
时间 t	12.954	13.875	14.982	15.903	16.826	17.931	19.037
水位	10.210	9.936	9.653	9.409	9.180	8.921	8.662
时间 t	19.959	20.839	22.015	22.958	23.880	24.986	25.908
水位	8.433	8.220	//	10.820	10.591	10.354	10.180

该问题的关键在于确定用水率函数，即单位时间内用水体积，记为 $f(t)$，又称水流速度. 如果能够通过测量数据，产生若干个时刻的用水率，也就是 $f(t)$ 在若干个点的函数值，则 $f(t)$ 的计算问题就可以转化为插值或拟合问题.

还有其他解决该问题的思路吗？如何实现？是否更好？

§5.2　插值算法

对于插值问题，有不同的方法可以构造插值函数，从而形成了不同插值方法. 直观地分析，分段线性插值自然，三次样条插值光滑，但是它们的函数表达式复杂，拉格朗日多项式插值函数形式表示简单. 究竟谁优谁劣？下面将分别加以介绍.

5.2.1　拉格朗日多项式插值

如果用一个次数不超过 n 次的多项式函数 $L(x)$ 作为插值函数 $p(x)$，即有

$$L(x_i) = y_i, i = 0, 1, \cdots, n.$$

可以证明,存在惟一的多项式函数满足以上插值条件. 可以用不同的方法构造出这个函数,拉格朗日插值法提出用 $n+1$ 个被称为拉格朗日插值基函数的 n 次多项式的线性组合来表示 $L(x)$,并记得到的插值函数为 $L_n(x)$:

$$L_n(x) = \sum_{i=0}^{n} l_i(x) y_i,$$

其中 $l_i(x) = \dfrac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}, i = 0, 1, \cdots n$ 就是拉格朗日插值基函数. 容易证明,

$$l_i(x_j) = \delta_{ij}, j = 0, 1, \cdots, n, i = 0, 1, \cdots, n,$$

从而 $L_n(x_i) = y_i, i = 0, 1, \cdots, n$ 满足插值条件.

还可以从其他角度出发,构造出插值多项式,如牛顿(Newton)插值公式.

拉格朗日插值方法最大优点是函数具有很好的解析性质(无穷次可微),但是它也存在固有的缺点:可能出现严重的振荡现象,并且多项式函数的系数依赖于观察数据.

一般而言,插值节点越多,插值的效果应该越好. 但是,对于拉格朗日多项式插值,情况并非如此. 例如:

假设用于插值的 $n+1$ 个数据由函数 $f(x) = e^{-x^2}, x \in [-5, 5]$(称为被插函数)产生,当插值节点个数 $n+1$(n 为插值多项式的阶数)增大时,其插值效果变差. 图 5.1 中给出了分别由 $n=6$ 和 $n=10$ 的插值结果. 随着 n 的增大,插值函数在接近区间端点处出现了剧烈的振荡现象,从而产生很大的误差. 这种现象称为龙格(Runge)振荡现象.

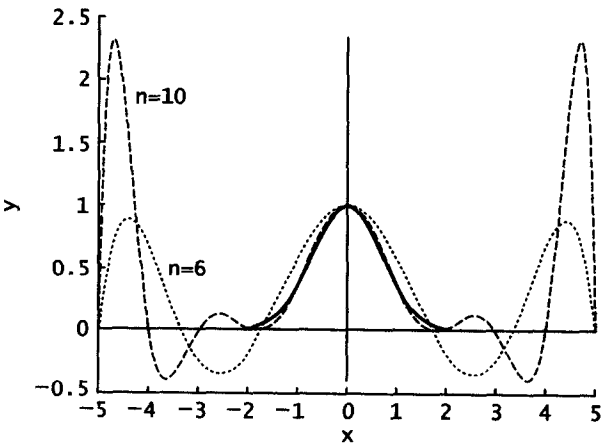


图 5.1 拉格朗日插值多项式的龙格振荡现象

拉格朗日多项式插值的另一个缺点是:多项式函数的系数依赖于观察数据的例子如表 5.2 给出的三个插值问题. 得到的插值多项式(4 次多项式)的五个系数如表 5.3:

表 5.2　待插值的数据(三个插值问题的数据:相差微小)

x_i	0.2	0.3	0.4	0.6	0.9
Case1: y_i	2.753 6	3.241 1	3.801 6	5.153 6	7.867 1
Case2: y_i	2.754	3.241	3.802	5.154	7.867
Case3: y_i	2.753 6	3.241 1	3.891 6	5.153 6	7.867 1

表 5.3　插值的结果:多项式的系数(三个插值问题的结果:大相径庭)

x_i	a_0	a_1	a_2	a_3	a_4
Case1: y_i	2	3	4	-1	1
Case2: y_i	2.012 3	2.878 1	4.415 9	-1.571 4	1.269 8
Case3: y_i	3.458 0	-13.200 0	64.750 0	-91.000 0	46.000 0

事实上,以上三种情况的关系是:Case2 是将 Case1 的数据 y_i 的最后一位数字进行四舍五入的结果,Case3 只是修改了 Case1 中一个数据的第三位有效数(由 3.8$\underline{0}$1 6 改为 3.8$\underline{9}$1 6),其他没有改变. 可以说这些都是非常微小的数据变化,结果导致插值多项式的系数大相径庭,如表 5.3.

导致拉格朗日多项式插值如上缺点在于:多项式的次数高阶尚不一定是好事. 克服这些缺点的办法之一,就是采用低次插值,即分段低次插值. 其中最简单的是分段线性插值,即在每个子区间上作通过两个端点的线性插值.

5.2.2　分段线性插值

分段线性插值的提法如下:

问题　给定一组观察数据$(x_i,y_i)(i=0,1,\cdots,n)$,其中 $x_0,x_1,\cdots,x_n$ 互异.

要求　求一个分段(共 n 段)线性函数 $q(x)$,使其满足:$q(x_i)=y_i,i=0,1,\cdots,n$.

根据直线的点斜式方程变形得到 $q(x)$ 在第 i 段$[x_{i-1},x_i]$上的表达式为

$$q(x)=\frac{x-x_i}{x_{i-1}-x_i}y_{i-1}+\frac{x-x_{i-1}}{x_i-x_{i-1}}y_i,x_{i-1}\leqslant x\leqslant x_i,i=1,2,\cdots,n.$$

可以证明,分段线性插值具有良好的收敛性,即 $\lim\limits_{n\to\infty}q(x)=f(x)$,其中 $f(x)$

为被插值函数.

分段线性插值在计算插值时,只用到前后两个相邻节点的函数值,计算量小. 在对函数表作插值计算时,经常用到这种插值方法.

例 5.1 求标准正态分布函数值 $\Phi(2.345\ 678\ 9)$.

由标准正态分布函数值表可以查表得到:

$$\Phi(2.34)=0.990\ 36,\qquad \Phi(2.35)=0.990\ 61.$$

采用分段线性插值计算 $\Phi(2.345\ 678\ 9)$. 取区间 $[x_{i-1},x_i]=[2.34,2.35]$, 被插值函数 $f(x)=\Phi(x)$, 则

$y_{i-1}=\Phi(x_{i-1})=\Phi(2.34)=0.990\ 36$; $y_i=\Phi(x_i)=\Phi(2.35)=0.990\ 61$,

利用如上分段线性插值公式得到

$$\Phi(2.345\ 678\ 9)=q(2.345\ 678\ 9)=0.990\ 5.$$

分段线性插值的函数曲线连续但不光滑(不可导),如果要求用光滑的曲线作为插值函数,可以考虑分段二次多项式函数吗?

对插值函数的要求可以根据需要进行改变,由此产生了不同的插值方法,分段三次埃尔米特插值就是其一.

在插值问题中,如果除了在插值节点的函数值给定以外,还要求在节点的导数值为给定值,这时插值问题变为:

问题 给定一组观察数据 $(x_i,y_i)(i=0,1,\cdots,n)$ 及 $y'_0,y'_1,\cdots,y'_n$.

要求 求一个分段(共 n 段)多项式函数 $q(x)$,使其满足:

$$q(x_i)=y_i,q'(x_i)=y'_i,i=0,1,\cdots,n.$$

这个问题可以用分段三次埃尔米特插值来解决.

相当于在每一小段上应满足四个条件(方程),可以确定四个待定参数. 三次多项式正好有四个系数,所以可以考虑用三次多项式函数作为插值函数,这就是**分段三次埃尔米特插值**,与分段线性插值一起都称为分段多项式插值.

1) 如何简单地产生分段三次埃尔米特插值公式?

2) 分段三次埃尔米特插值与分段线性插值的曲线光滑程度有何差别?

3) 在上面问题中,如果只要求节点的导数存在,则分段多项式的次数为多少?

5.2.3 三次样条插值

上面介绍的分段线性插值，其总体光滑程度不够. 在数学上，光滑程度的定量描述是：函数（曲线）的 k 阶导数存在且连续，则称该曲线具有 k 阶光滑性. 自然，光滑性阶数越高其曲线光滑程度越好. 于是，分段线性插值具有零阶光滑性，也就是不光滑；分段三次埃尔米特插值具有一阶光滑性. 仅有这些光滑程度，在工程设计和机械加工等实际中是不够的. 提高分段函数如多项式函数的次数，可望提高整体曲线的光滑程度. 但是，是否存在较低次多项式达到较高阶光滑性的方法？三次样条插值就是一个很好的例子.

样条曲线本身就来源于飞机、船舶等外形曲线设计中所用的绘图工具. 在工程实际中，要求这样的曲线应该具有连续的曲率，也就是连续的二阶导数. 值得注意的是分段插值曲线的光滑性关键在于段与段之间的衔接点（节点）处的光滑性.

三次样条函数 记为 $S(x)$，它是定义在区间 $[a,b]$ 上的函数，满足

1）$S(x)$ 在每一个小区间 $[x_{i-1},x_i]$ 上是一个三次多项式函数；

2）在整个区间 $[a,b]$ 上，其二阶导数存在且连续，即在每个节点处的二阶导数连续.

三次样条插值问题的提法 给定函数 $f(x)$ 在 $n+1$ 个节点 $x_0,x_1,\cdots,x_n$ 处的函数值为 $y_0,y_1,\cdots,y_n$，求一个三次样条函数 $S(x)$，使其满足

$$S(x_i)=y_i,\qquad i=0,1,\cdots,n.$$

如何确定三次样条函数在每一个小区间上的三次多项式函数的系数呢？它是一个比较复杂的问题，这里只介绍确定系数的思想.

分段线性插值在每一段的线性函数的两个参数，是由两个方程（两个端点处的函数值为给定值）惟一确定；对于三次样条插值呢，每一个区间上的三次函数有四个参数，而在该区间上由两个端点的函数值为给定值只能够产生两个方程，仅此不足以惟一确定四个参数. 注意到三次样条函数对整体光滑性要求，其二阶导数存在且连续，从全局的角度上考虑参数个数与方程个数的关系如下：

参数：每个小段上 4 个，n 个小段共计 $4n$ 个.

方程：每个小段上由给定函数值得到 2 个，n 个小段共计 $2n$ 个；光滑性要求每一个内部节点的一阶、二阶导数连续，得出其一阶、二阶左右导数必相等，因此，每个节点产生 2 个方程，共计 $2(n-1)$ 个.

现在得到了 $4n-2$ 个方程，还差两个. 为此，常用的方法是对边界节点除函数值外附加要求，这就是所谓的**边界条件**. 需要两个，正好左右两个端点各一个. 常用如下三类边界条件.

m 边界条件 $S'(x_0)=m_0, S'(x_n)=m_n$，即两个边界节点的一阶导数值为给定值 m_0, m_n.

M 边界条件 $S''(x_0)=M_0, S''(x_n)=M_n$，即两个边界节点的二阶导数值为给定值 M_0, M_n.

特别地，当 M_0 和 M_n 都为零时，称为**自然边界条件**.

周期性边界条件 $S'(x_0)=S'(x_n); S''(x_0)=S''(x_n)$.

以上分析说明，理论上三次样条插值函数是确定的，具体如何操作，可以查阅有关文献.

本节介绍的是一维插值方法：分段线性插值，三次样条插值，拉格朗日多项式插值等，这些可以推广到高维插值. 常用高维插值方法包括最邻近插值(Proximal)，线性插值(Linear)，三次样条插值(Cubic Spline)，二次多项式插值(Quadratic Polynomial)等等，可以参考相应文献.

§5.3 水塔用水量的计算

为了解决水塔用水量的计算，我们采用一维插值方法. 首先需要对问题作合理的假定和分析.

5.3.1 问题分析

1. 假设

1) 水塔中水流量是时间的连续光滑函数，与水泵工作与否是无关的，并忽略水位高度对水流速的影响.

2) 水泵工作与否完全取决于水塔内水位的高度.

3) 水塔为标准圆柱体.

考虑到假设 2)，结合表 5.1 中具体数据，推断得出：

4) 水泵第一次供水时间段为[8.967, 10.954]，第二次供水时间段为[20.839, 22.958].

2. 体积计算

水塔是一个圆柱体，体积为 $V=\frac{\pi}{4}D^2h$，其中 D 为底面直径，h 为水位高度. 近似地取 $\pi=3.141\ 592\ 654$，得到不同时刻水塔中水的体积如表 5.4.

表 5.4　水塔中水的体积(单位:时间/h,体积/m^3)

时间	0	0.921	1.843	2.949	3.871	4.978	5.900
体积	2 294	2 247	2 206	2 163	2 129	2 089	2 059
时间	7.006	7.928	8.967	9.981 1	10.925	10.954	12.032
体积	2 020	1 988	1 948	//	//	2 564	2 489
时间	12.954	13.875	14.982	15.903	16.826	17.931	19.037
体积	2 420	2 355	2 288	2 230	2 176	2 114	2 053
时间	19.959	20.839	22.015	22.958	23.880	24.986	25.908
体积	1 999	1 948	//	2 564	2 510	2 454	2 413

3. 水流速度的估算

水流速度应该是水塔中水的体积对时间的导数(微商),由于没有水的体积关于时间的函数表达式,而只有一组离散的函数值(表 5.4).因此考虑用差商代替微商,这也是离散反映连续的常用思想.为提高精度,采用二阶差商,即 $f(t_i) = -\nabla^2 v_i$.

具体地,因为所有数据被水泵两次工作分割成三组数据,对每组数据的中间数据采用中心差商,前后两组数据不能够采用中心差商,改用向前或向后差商,由此得到的水塔中水的流速数据如表 5.5.

中心差商公式

$$\nabla^2 v_i = \frac{-v_{i+2} + 8v_{i+1} - 8v_{i-1} + v_{i-2}}{12(t_{i+1} - t_i)}.$$

向前和向后差商公式:

$$\nabla^2 v_i = \frac{-v_{i+2} + 4v_{i+1} - 3v_i}{2(t_{i+1} - t_i)}, \nabla^2 v_i = \frac{3v_i - 4v_{i-1} + v_{i-2}}{2(t_i - t_{i-1})}.$$

表 5.5　水塔中水的流速(单位:时间/h,流速/$m^3 \cdot h^{-1}$)

时间	0	0.921	1.843	2.949	3.871	4.978	5.900
流速	54.516	42.320	38.085	41.679	33.297	37.814	30.748
时间	7.006	7.928	8.967	9.981 1	10.925	10.954	12.032
流速	38.455	32.122	41.718	//	//	73.686	76.434
时间	12.954	13.875	14.982	15.903	16.826	17.931	19.037
流速	71.686	60.190	68.333	59.217	52.011	56.626	63.023
时间	19.959	20.839	22.015	22.958	23.880	24.986	25.908
流速	54.859	55.439	//	57.602	57.766	51.891	36.464

1）在估算流速的时候，为什么要分成三段分别处理？

2）就一段估算流速合理吗？还有其他估算流速的方法吗？

5.3.2 模型建立与求解

我们采用 MATLAB 软件直接计算一维插值问题. 在 MATLAB 软件中给出了一维插值函数 interp1()，其调用格式如下：

```
yi = interp1(x,y,xi,'method')
```

其中 x,y 为插值点，y_i 为在被插值点 x_i 处的插值结果. 'method'表示采用的插值方法，MATLAB 提供的插值方法有几种：'nearest'表示最邻近插值；'linear'表示线性插值；'spline'表示三次样条插值；'cubic'表示三次插值. 缺省时表示线性插值.

注意：所有的插值方法都要求 x 单调，并且 x_i 不能够超过 x 的范围.

还有其他的插值函数，如 interp1q，interpft 等.

根据已知数据可以作出水的流速数据散点图，如图 5.2 所示. 代码如下：

```
plot(t,r,'b+');title('流速散点图');
xlabel('时间/小时');    ylabel('流速/(立方米/小时)')
```

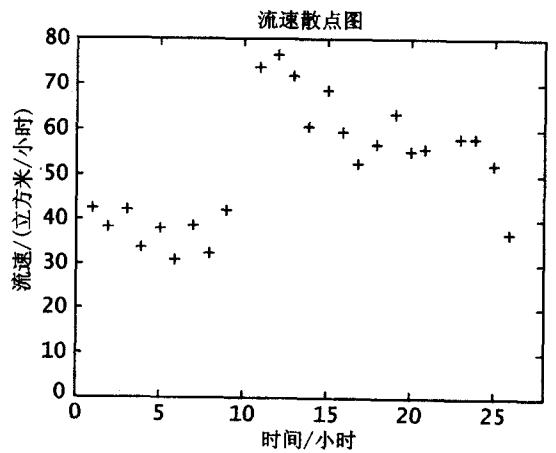


图 5.2 水塔内水流速散点图

问题已经转变为根据流速 $f(t)$ 的一个函数值表，产生函数 $f(t)$ 在整个区间（二十四小时）上的函数或函数值，插值和拟合是两种最常用的方法. 如果建立

拟合模型,需要根据散点图的趋势,选择适当的拟合函数形式.如果采用插值模型,可以考虑分段线性插值,三次样条插值等等.

思考

1) 如果采用多项式函数作拟合,选择几次多项式函数?

2) 多项式函数拟合模型与多项式插值模型,思想有何异同,结果有何差异?

通过对不同插值方法的比较,结合假设,考虑到流速应该是时间的连续光滑函数.下面采用三次样条插值模型,计算用水率函数,如图 5.3 所示.

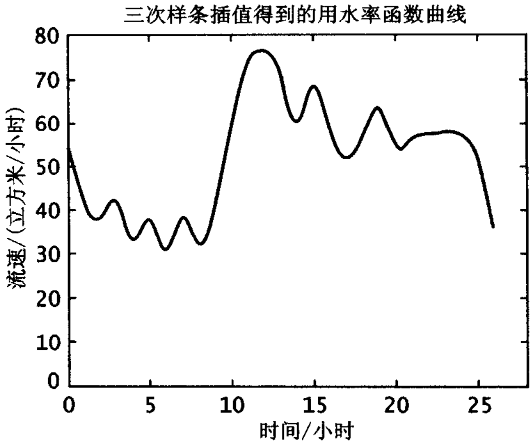


图 5.3 三次样条插值模型计算得到的用水率函数曲线

首先由三次样条插值计算得到用水率函数 $f(t)$,如图 5.3 所示. MATLAB 程序如下:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    clear x y1 ys
    areas = trapz(t,r);    x0 = t;y0 = r;
    [1,n] = size(x0);d1 = x0(n) - x0(1);
    x = x0(1):1/3600:x0(n);% 被插值点
    ys = interp1(x0,y0,x,'spline');% 样条插值输出
    flows = trapz(x,ys)% 样条插值时,下边界所成曲边梯形的面积
    plot(x,ys);title('三次样条插值得到的用水率函数曲线');
    xlabel('时间/小时');ylabel('流速/(立方米/小时)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

如何计算一天 24 小时的总用水量呢？既可以直接由前面通过二阶差商得到的水的流速数据(表 5.5)，用梯形公式进行数值积分得到的结果为 1 250.3 m^3，也可以用三次样条插值模型得到的函数 $f(t)$ 在时间区间[0,24]上积分得到的结果为 1 257.3 m^3. 二者之间的绝对误差仅为 7 m^3，相对误差不到百分之一(0.56%)，说明两种方法的一致性好.

5.3.3 模型的检验

下面从几个侧面对模型进行检验：

用不同时刻作为起始点，计算一天 24 小时的总用水量：使用插值模型得到的用水率函数，在长度为 24 小时的时间区间上进行积分，所得到的结果相差无几，如表 5.6 所示.

表 5.6 由插值模型计算出不同起点的 24 小时总用水量(用水量单位/m^3)

起始点	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75
用水量	1 257.3	1 258.5	1 260.3	1 262.6	1 265.1	1 267.6	1 269.6	1 270.7

分三段(水泵未工作)的实际用水量与由模型推算的用水量的差异很小. 如表 5.7，最大相对误差不超过 4%，三段总计误差仅为 0.2%.

表 5.7 分三段的实际用水量与模型用水量的比较

	实际用水量	模型用水量	绝对误差	相对误差
第一段 [0,8.967]	345.368 2	338.051 1	7.317 1	2.1%
第二段 [10.954,20.839]	616.316 1	620.834 1	4.518 0	0.7%
第三段 [22.958,25.908]	151.730 8	156.600 0	4.869 2	3.2%
三段总计	1 113.415 1	1 115.485 2	2.070 1	0.2%

注：模型用水量为用水率函数 $f(t)$ 的积分，实际用水量为水塔内水体积之差.

两次充水期间，水泵充水量的差异相差约三个立方米，不到 0.5%. 水泵充水量 = 充水后的水量 + 充水期间的流出量 - 充水前的水量.

第一次：水泵充水量 = 2 564 + 117.446 3 - 1 948 = 733.446 3(m^3)；

第二次：水泵充水量 = 2 564 + 120.705 2 - 194 8 = 736.705 2(m^3).

用水高峰的比较：实际用水高峰与模型对应的用水高峰之间相差无几. 实际用水高峰：近似地用差商最大值点为 $t = 11$，即上午 11 点钟左右. 模型得到的用

水高峰:由模型得到的用水率函数依次在单位区间上积分或者直接找出用水率函数$f(t)$的最大值点,它们都在$t=11$左右.有兴趣的读者可以精确计算一下.

§5.4 二维插值的应用

MATLAB给出的高维插值函数为interpN( ),其中N可以为2,3,….例如$N=2$为二维插值,调用格式为

```
zi = interp2(x,y,z,xi,yi,'method')
```

其中x,y,z为插值节点,z_i为被插值点(x_i,y_i)处的插值结果.'method'表示采用的插值方法:'nearest'表示最邻近插值;'linear'表示线性插值;'cubic'表示三次插值.缺省时表示线性插值.所有的插值方法都要求x和y是单调的网格,x和y可以是等距的也可以是不等距的.

气旋变化情况的可视化

表5.8是气象学家测量得到的气象资料,它们分别表示在南半球地区按不同纬度、不同月份的平均气旋数据.根据这些数据,绘制出气旋分布曲面图形.

表5.8 南半球地区按不同纬度、不同月份的平均气旋数据

	0~10	10~20	20~30	30~40	40~50	50~60	60~70	70~80	80~90
1月	2.4	18.7	20.8	22.1	37.3	48.2	25.6	5.3	0.3
2月	1.6	21.4	18.5	20.1	28.8	36.6	24.2	5.3	0
3月	2.4	16.2	18.2	20.5	27.8	35.5	25.5	5.4	0
4月	3.2	9.2	16.6	25.1	37.2	40	24.6	4.9	0.3
5月	1.0	2.8	12.9	29.2	40.3	37.6	21.1	4.9	0
6月	0.5	1.7	10.1	32.6	41.7	35.4	22.2	7.1	0
7月	0.4	1.4	8.3	33.0	46.2	35	20.2	5.3	0.1
8月	0.2	2.4	11.2	31.0	39.9	34.7	21.2	7.3	0.2
9月	0.5	5.8	12.5	28.6	35.9	35.7	22.6	7	0.3
10月	0.8	9.2	21.1	32.0	40.3	39.5	28.5	8.6	0
11月	2.4	10.3	23.9	28.1	38.2	40	25.3	6.3	0.1
12月	3.6	16	25.5	25.6	43.4	41.9	24.3	6.6	0.3

下面用二维三次插值方法,可以得到不同月份按纬度变化的气旋值(插值

结果),然后再作出其可视化图形如图 5.4. MATLAB 程序如下:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    y=5:10:85;x=1:12;
    z=[2.4  18.7  20.8  22.1  37.3  48.2  25.6  5.3  0.3;…;
    3.6  16  25.5  25.6  43.4  41.9  24.3  6.6  0.3]';
    [xi,yi]=meshgrid(1:12,5:1:85);zi=interp2(x,y,z,xi,yi,'cubic');
    mesh(xi,yi,zi);
    xlabel('月份'),ylabel('纬度'),zlabel('气旋'),
    axis([0  12  0  90  0  50])
    title('南半球气旋可视化图形')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

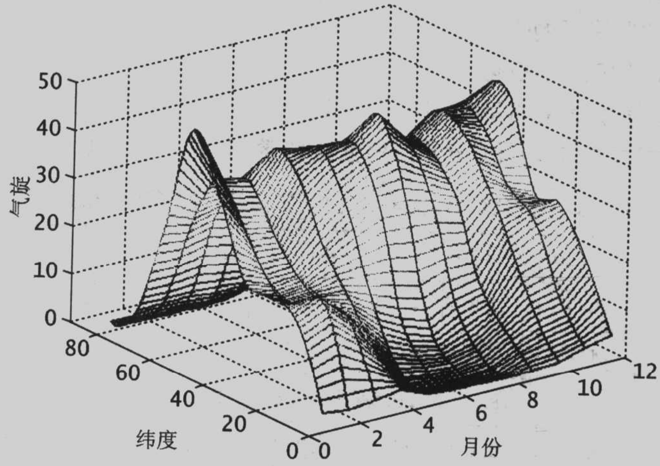


图 5.4　南半球气旋变化的可视化图形

§5.5　操　　练

操练一　确定地球与金星之间的距离

天文学家在 1914 年 8 月份的 7 次观测中,测得地球与金星之间距离(单位/m),并取其常用对数值与日期的一组历史数据如表 5.9 所示,试推断何时金星与地球的距离(m)的对数值为 9.935 179 9?

表 5.9　地球与金星之间距离的对数值与日期的数据

日期	18	20	22	24
距离对数	9.961 772 4	9.954 364 5	9.946 806 9	9.939 095 0
日期	26	28	30	
距离对数	9.931 224 5	9.923 191 5	9.914 992 5	

操练二　机动车刹车问题

表 5.10 是一组机动车的刹车距离 d 与速度 v 的测试数据(速度单位为 mile/h,距离单位为 ft(英尺)).利用三次样条插值分析刹车距离与速度的关系,为了使刹车距离限制在 328 ft 以内,行驶速度必须限制在多少 mile/h 之内(1 ft = 0.304 8 m,1 mile = 160 9 m).

表 5.10　机动车刹车距离与速度的测试数据

v	20	25	30	35	40	45	50	55	60	65	70	75	80
d	42	56	73.5	91.5	116	142.5	173	209.5	248	292.5	343	401	464

操练三　城市居民生活节奏变化

城市人口增加将如何影响居民生活节奏. Marc 和 Helen Bornstein 对分布在世界各地的 15 个城市作了调查,测试了每个城市居民在其居住城市大街上步行 50 ft 的平均速度,记录数据如表 5.11. 试用插值方法确定插值函数,并对结果进行分析.

表 5.11　城市人口与居民步行速度的数据

人口数量	314 948	1 092 759	5 491	49 375	1 340 000	365	2 500	7 820
步行速度	4.81	5.88	3.31	4.90	5.62	2.76	2.27	3.85
人口数量	867 023	14 000	23 700	70 700	304 500	138 000	2 602 000	
步行速度	5.21	3.7	3.27	4.31	4.42	4.39	5.05	

操练四　山区地貌图

在某山区(平面区域(0,280 0) × (0,240 0)内,单位:m)测得一些地点的高度(单位:m)如表 5.12 所示,试作出该山区的地貌图和等高线图.

表 5.12　某山区一些地点的高度

2 400	1 430	1 450	1 470	1 320	1 280	1 200	1 080	940
2 000	1 450	1 480	1 500	1 550	1 510	1 430	1 300	1 200
1 600	1 460	1 500	1 550	1 600	1 550	1 600	1 600	1 600
1 200	1 370	1 500	1 200	1 100	1 550	1 600	1 550	1 380
800	1 270	1 500	1 200	1 100	1 350	1 450	1 200	1 150
400	1 230	1 390	1 500	1 500	1 400	900	1 100	1 060
0	1 180	1 320	1 450	1 420	1 400	1 300	700	900
Y/X	0	400	800	1 200	1 600	2 000	2 400	2 800

更多的相关信息资源

1　Frank R. Giordano, Maurice D. Weir, William O. Fox. A first course in mathematical modeling. 3rd edition. 影印版. 北京:机械工业出版社,Thomson Learning, Inc., 2003

2　傅鹂,龚劬,刘琼荪,何中市. 数学实验. 北京:科学出版社,2000

3　COMAP. Principles and practice of mathematics. 中译本. 北京:高等教育出版社,Springer-Verlag, 1997

4　萧树铁主编,姜启源,何青,高立编著. 数学实验. 北京:高等教育出版社,1999

5　姜启源. 数学模型. 第二版. 北京:高等教育出版社,1993

6　John H. Mathews, Kurtis D. Fink. Numerical methods using Matlab. 3rd edition. 英文版. 北京:电子工业出版社,Pearson Education,2002

7　李尚志. 数学建模竞赛教程. 南京:江苏教育出版社,1996

8　D. Quinney. An Introduction to the Numerical Solution of Differential Equations. New York: John Wiley & Sons Inc., 1985

9　G. Fulford. Modeling with differential and difference equations. Cambridge: Cambridge University Press, 1997

10　李庆扬,王能超,易大义. 数值分析. 武汉:华中理工大学出版社,1987

第 6 章

医用薄膜渗透率的确定

——曲线拟合

一个实际问题往往用机理分析或凭经验可以建立经验公式，但在经验公式中，如果含有一些未知参数，如何确定呢？这就需要采样，通过获得的测量数据，建立一种规则从而确定经验公式中的未知参数，这就是数据拟合思想.

——作者

在生产实践和科学实验中，经常会遇到大量的各种不同类型的数据(data). 这些数据提供了有用的信息，它可以帮助我们认识事物的内在规律、研究事物之间的关系等. 例如，我们欲分析汽车的刹车距离(汽车从使用刹车到完全停止惯性行驶的距离)与刹车速度之间的关系，如果根据机理分析，刹车距离与汽车的刹车速度的平方成正比，即 $d = Cv^2$，其中 d 表示刹车距离，v 表示当使用刹车时汽车的速度，C 是未知常数，如何确定 C? 通常，我们进行多次重复测试，获得刹车距离与速度的数据 (d_i, v_i)，$i = 1, 2, \cdots, n$，由此建立某种规则去确定未知常数 C，这时，模型 $d = Cv^2$ 才能完全确定.

根据一组二维数据，即平面上的若干点，要求确定一个一元函数 $y = f(x)$，即曲线，使这些点与曲线总体来说尽量接近，这就是数据拟合曲线的思想，简称为曲线拟合(fitting a curve). “拟合”即不要求所作的曲线完全通过所有的数据点，只是要求所得的近似曲线能反映数据的基本趋势.

怎样从给定的二维数据出发，寻找一个简单合理的函数来拟合给定的一组看上去杂乱无章的数据，正是本章要讨论的主要内容.

§6.1 医用薄膜的渗透率

某种医用薄膜有允许一种物质的分子穿透它，从高浓度的溶液向低浓度的溶液扩散的功能，在试制时需测定薄膜被这种分子穿透的能力. 测定方法如下：

用面积为 S 的薄膜将容器分成体积分别为 V_A，V_B 的两部分，在两部分中分别注满该物质的两种不同浓度的溶液如图 6.1. 此时该物质分子就会从高浓度溶液穿过薄膜向低浓度溶液扩散. 通过单位面积薄膜分子扩散的速度与薄膜两侧溶液的浓度差成正比，比例系数 K 表征了薄膜被该物质分子穿透的能力，称为渗透率. 定时测量容器中薄膜某一侧的溶液浓度值，以此确定 K 的数值.

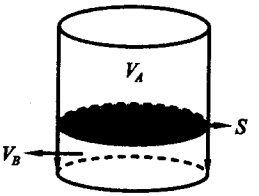


图 6.1　圆柱体容器被薄膜截面 S 阻隔

§6.2　确定医用薄膜渗透率的数学模型

该问题涉及化学知识，即浓度的变化规律，可用多种方式建立数学模型.

6.2.1　问题假设

1）薄膜两侧的溶液始终是均匀的，即在任何时刻薄膜两侧的每一处溶液的浓度都是相同的.

2）当薄膜两侧的溶液浓度不一致时，物质的分子穿透薄膜总是从高浓度溶液向低浓度溶液扩散.

3）通过单位面积薄膜分子扩散的速度与薄膜两侧溶液的浓度差成正比.

4）薄膜是双向同性的，即物质从薄膜的任何一侧向另一侧渗透的性能是相同的.

是否还有其他的假设？

不同的假设应该具有不同形式的数学模型.

注：符号说明

1）$C_A(t)$，$C_B(t)$ 表示 t 时刻薄膜两侧溶液的浓度；

2）α_A、α_B 表示初始时刻两侧溶液的浓度（单位：mg/cm^3）；

3）K 表示薄膜渗透率；

4）V_A，V_B 表示由薄膜阻隔的容器两侧的体积.

6.2.2　问题分析

解决问题的思路：首先通过机理分析寻找某一侧浓度随时间变化的函数关系 $C_A(t)$ 或 $C_B(t)$，其中可能含有待定参数，如薄膜渗透率 K；然后根据一组测量

值(t_i, C_i), $i=1,2,\cdots,n$ 去确定模型中的待定系数.

考察时段$[t, t+\Delta t]$薄膜两侧容器中该物质质量的变化. 以容器 A 侧为例,在该时段物质质量的增加量:$V_A C_A(t+\Delta t) - V_A C_A(t)$,另一方面从 B 侧渗透至 A 侧的该物质的质量为:$SK[C_B(t) - C_A(t)]\Delta t$. 由质量守恒定律,两者应该相等,于是有

$$V_A C_A(t+\Delta t) - V_A C_A(t) = SK[C_B(t) - C_A(t)]\Delta t.$$

两边除以 Δt,令 $\Delta t \to 0$ 并整理得

$$\frac{dC_A(t)}{dt} = \frac{SK}{V_A}[C_B(t) - C_A(t)]. \tag{6.1}$$

且注意到整个容器的溶液中含有该物质的质量应该不变,即有下式成立:

$$V_A C_A(t) + V_B C_B(t) = V_A \alpha_A + V_B \alpha_B,$$

两边除以 V_A并整理得 $C_A(t) = \alpha_A + \frac{V_B}{V_A}\alpha_B - \frac{V_B}{V_A}C_B(t)$.

代入(6.1)得

$$\frac{dC_B(t)}{dt} + SK\left(\frac{1}{V_A} + \frac{1}{V_B}\right)C_B(t) = SK\left(\frac{\alpha_A}{V_B} + \frac{\alpha_B}{V_A}\right),$$

再利用初始条件 $C_B(0) = \alpha_B$,

解出 $$C_B(t) = \frac{\alpha_A V_A + \alpha_B V_B}{V_A + V_B} + \frac{V_A(\alpha_B - \alpha_A)}{V_A + V_B} e^{-SK\left(\frac{1}{V_A} + \frac{1}{V_B}\right)t}.$$

6.2.3 数学模型

问题归结为利用 C_B在时刻 t_j的测量数据 C_j $(j=1,2,\cdots,N)$来辨识参数 K 和 α_A, α_B,对应的数学模型变为使函数

$$E(K, \alpha_A, \alpha_B) = \sum_{j=1}^{N} (C_B(t_j) - C_j)^2$$

达到最小.

令 $$a = \frac{\alpha_A V_A + \alpha_B V_B}{V_A + V_B}, b = \frac{V_A(\alpha_B - \alpha_A)}{V_A + V_B},$$

问题转化为如下最优化问题:

$$\min_{K,a,b} E(K,a,b) = \sum_{j=1}^{n} [a + be^{-SK\left(\frac{1}{V_A} + \frac{1}{V_B}\right)t_j} - C_j]^2.$$

实际上,该目标函数是误差平方和,确定参数 K, a, b,使得误差平方和达最小.

如果对该问题所用的机理分析背景知识一无所知，只知道一组测量数据，用什么样的浓度函数 $C=f(t)$ 来拟合所给出的数据 (t_i,C_i)，$i=1,2,\cdots,n$？

对给定的数据作散点图，以便观察浓度函数 $C=f(t)$ 的函数形式.

§6.3 一元最小二乘法简介

给定平面上的点 (x_i,y_i)，$i=1,2,\cdots,n$，进行曲线拟合有多种方法，最小二乘法是解决曲线拟合最常用的一种方法. 最小二乘法的提法是：

$$\text{求 } f(x),\text{使 } \delta=\sum_{i=1}^{n}\delta_i^2=\sum_{i=1}^{n}[f(x_i)-y_i]^2 \text{达到最小.}$$

如图 6.2 所示，其中 δ_i 为点 (x_i,y_i) 与曲线 $y=f(x)$ 的距离. 曲线拟合的实际含义是寻求一个函数 $y=f(x)$，使 $f(x)$ 在某种准则下与所有数据点最为接近，即曲线拟合得最好. 最小二乘准则就是使所有散点到曲线的距离平方和最小. 拟合时选用一定的拟合函数 $f(x)$ 的形式，设拟合函数可由一些简单的“基函数”(例如幂函数，三角函数等等) $\varphi_0(x),\varphi_1(x),\cdots,\varphi_m(x)$ 来线性表示：

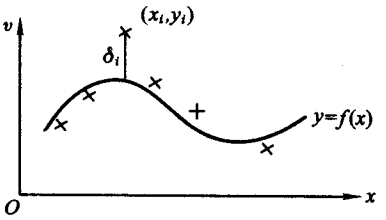


图 6.2 曲线拟合示意图

$$f(x)=c_0\varphi_0(x)+c_1\varphi_1(x)+\cdots+c_m\varphi_m(x).$$

现在要确定系数 $c_0,c_1,\cdots,c_m$，使 δ 达到极小. 为此，将 $f(x)$ 的表达式代入 δ 中，δ 就成为 $c_0,c_1,\cdots,c_m$ 的函数，为使 δ 达到极小，可令 δ 对 c_i 的偏导数等于零，于是得到 $m+1$ 个方程组，从中求解出 c_i. 通常取基函数为 $1,x,x^2,x^3,\cdots,x^m$，这时拟合函数 $f(x)$ 为多项式函数. 当 $m=1$ 时，$f(x)=a+bx$，称为一元线性拟合函数，它是曲线拟合最简单的形式.

已知一组数据 (x_i,y_i)，$i=1,2,\cdots,n$，选择什么样的函数 $f(x)$ 呢？一是根据机理分析来确定函数形式，二是根据散点图直观判断函数 $f(x)$ 的形式，常用的一元曲线拟合函数有双曲线 $f(x)=a+b/x$，指数曲线 $f(x)=ae^{bx}$，多项式 $f(x)=$

$a_m x^m + \cdots + a_1 x + a_0$等，对于这些曲线，可以通过变量代换转化为线性函数.

1）拟合为什么要用最小二乘准则？还有其他准则吗？

2）用其他准则得到的关于待定系数 $c_0, c_1, \cdots, c_m$ 的方程是怎样的？

§6.4 用曲线拟合方法确定医用薄膜渗透率

由§6.2节所述的医用薄膜渗透率 K 的最小二乘法参数辨识模型：

$$\min_{K,a,b} E(K,a,b) = \sum_{j=1}^{n} [a + be^{-SK(\frac{1}{V_A}+\frac{1}{V_B})t_j} - C_j]^2. \tag{6.2}$$

假设给定一个算例：$V_A = V_B = 1\ 000\ \text{cm}^3$，$S = 10\ \text{cm}^2$，对容器的 B 部分溶液浓度的测试结果如表6.1.

表6.1 容器的B部分溶液浓度的测试

t_j/s	100	200	300	400	500	600	700	800	900	1 000
$C_j(\times 10^{-5})$	4.54	4.99	5.35	5.65	5.90	6.10	6.26	6.39	6.50	6.59

其中 C_j 的单位：mg/cm^3.

可以将式(6.2)简化为 $\min_{K,a,b} E(K,a,b) = \sum_{j=1}^{10} [a + be^{-20K \cdot t_j} - C_j]^2$.

用 MATLAB 软件进行计算.

1）编写 M 文件(nongdu.m)

```
function f = nongdu(x,tdata)
f = x(1) + x(2) * exp( -0.02 * x(3) * tdata);
```

其中 x(1) = a; x(2) = b; x(3) = K;

2）编写程序(nihel.m)

```
tdata = linspace(100,1000,100);
cdata = [4.54,4.99,5.35,5.65,5.90,6.10,6.26,6.39,6.50,6.59];
x0 = [0.2,0.05,0.05];% 任意选取
 x = lsqcurvefit('nongdu',x0,tdata,cdata)
```

3）输出结果：

```
x =0.007   -0.003  0.1012
```

即表示 $a=0.007, b=-0.003, K=0.1012$,

并且求出　$\alpha_B=0.004\ mg/cm^3, \alpha_A=0.01\ mg/cm^3$.

拟合效果如何？需要进行误差分析，主要计算误差平方和，即公式(6.2)的值. 另外，可以作出拟合函数曲线和数据点的图形，直观分析拟合效果，如图6.3所示.

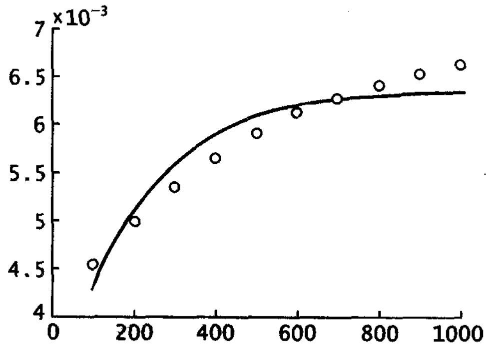


图 6.3　曲线拟合的效果图

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nihe2.m
tdata = linspace(100,1000,100);
cdata = 1e-05.*[454 499 535 565 590 610 626 639 650 659];
x0 = [0.2,0.05,0.05];
[x,resnorm,residual] = lsqcurvefit('nongdu',x0,tdata,cdata)
t = linspace(100,1000,100);
c = nongdu(x,t);
plot(tdata,cdata,'o',t,c)
%求拟合在各节点的误差平方和
c1 = nongdu(x,tdata);
e = c1 - cdata;
e1 = sum(e.*e)
输出结果如下：
e1 = 3.5604e-007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

计算结果表明，总的误差平方和为 3.5604×10^{-7}，几乎为零. 说明该问题的

拟合效果很好.

注意　在 MATLAB 软件中，语句 lsqcurvefit( )不仅可以做曲线拟合，还可以做曲面拟合.

1）还有其他的拟合方法吗？

2）拟合的效果如何评价？

另外，MATLAB 软件还提供了多项式函数似合的语句：

```
a=polyfit (xdata,ydata,n)
```

其中 n 表示多项式的最高阶数，xdata,ydata 为将要拟合的数据，它是用数组的方式输入. 输出参数 a 为拟合多项式 $y = a_1 x^n + \cdots + a_n x + a_{n+1}$ 的系数 $a = [a_1, \cdots, a_n, a_{n+1}]$.

多项式在 x 处的拟合值 y 可用下面程序计算：

```
y=polyval(a,x)
```

例如，考虑载重汽车的刹车距离与刹车速度之间的函数关系，现测得一组观测值如表 6.2：

表 6.2　载重汽车的刹车距离与刹车速度的观测值

序号	1	2	3	4	5	6	7	8	9	10	11	12	13
v	20	25	30	35	40	45	50	55	60	65	70	75	80
d	42	56	73.5	91.5	116	142.5	173	209.5	248	292.5	343	401	464

首先，画散点图如图 6.4.

通过实测数据的散点图观察，不妨假设刹车距离 d 与速度 v 之间的函数结构为二次多项式：$d = a_2 v^2 + a_1 v + a_0$. 关键是通过观测数据辨识未知参数 $a = [a_2, a_1, a_0]$. MATLAB 程序如下：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  (dxsnh.m)
  v=[20 25 30 35 40 45 50 55 60 65 70 75 80];
  d=[42 56 73.5 91.5 116 142.5 173 209.5 248 292.5 343 401 464];
  a=polyfit(v,d,2)
  d1=polyval(a,v);
  plot(v,d,'ro',v,d1,'b'.);
```

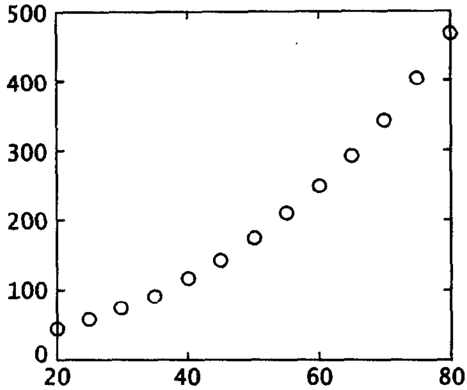


图 6.4　刹车距离 d 与刹车速度 v 之间的实测数据

输出结果:

```
a =    0.0886   -1.9701  50.0594
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

即拟合多项式为:$d=0.0886v^2-1.9701v+50.0594$.

注意:通过实测数据的散点图观察,还可以假设刹车距离 d 与速度 v 之间的函数关系为:$d=a\sqrt{v}$. 通过观测数据辨识未知参数 a. 编写 MATLAB 程序如下:

```
v=[20 25 30 35 40 45 50 55 60 65 70 75 80];
d=[42 56 73.5 91.5 116 142.5 173 209.5 248 292.5 343 401 464];
v=sqrt(v);
a=polyfit(v,d,1)
d1=polyval(a,v);
plot(v,d,'ro',v,d1,'b'.);box off,
```

输出结果:

```
a =    91.4428   -430.1865
```

即拟合函数为 $d=91.4428\sqrt{v}-430.1865$.

两个模型:1) $d=0.0886v^2-1.9701v+50.0594$;

2) $d=91.4428\sqrt{v}-430.1865$,

哪一个拟合效果最佳?残差平方和的值越小,反映拟合效果越好. 比较两个模型的残差平方和,模型 1)、2)的残差平方和分别记为 e_1,e_2,计算得:$e_1=9.6863$,$e_2=140.8012$,显然,模型 1)的拟合效果较好. 反映拟合效果的散点图见图 6.5 和图 6.6.

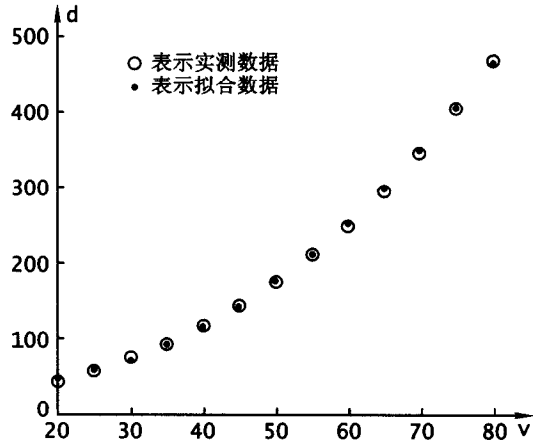


图 6.5　模型 1)刹车距离 d 的拟合数据与实测数据的比较

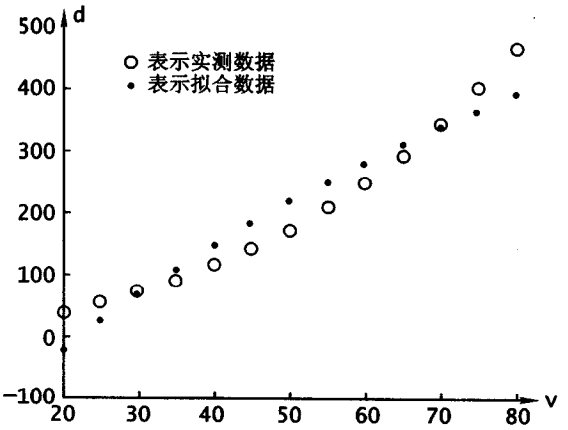


图 6.6　模型 2)刹车距离 d 的拟合数据与实测数据的比较

曲线拟合与曲线插值有什么区别呢？

曲线拟合和曲线插值这两种方法都可以预测数据间的取值，选择哪种方法主要取决于建模者的态度. 如果建模者单纯地只是希望预测两点之间的值，则可以选择插值. 但如果欲预测观测值以外的点，或者还有其他目的，则可能选择拟合方法. 例如，简单的线性插值与拟合的图形效果如图 6.7 所示.

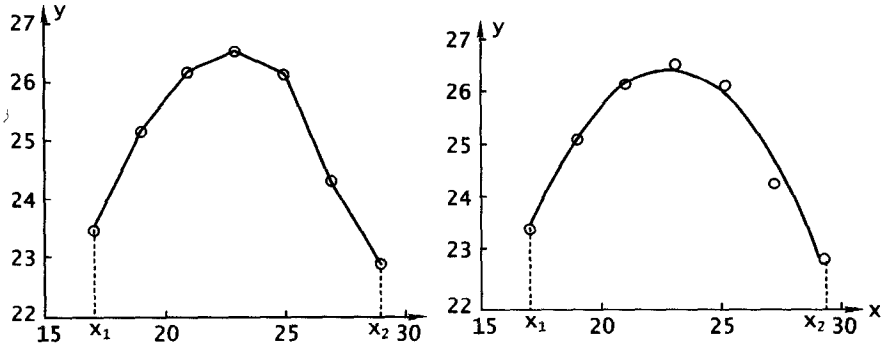


图 6.7　曲线插值与曲线拟合图

由图可知,插值曲线经过所有的观测数据点,而拟合曲线不一定经过所有的观测数据点.

§6.5　简介曲面拟合

实际问题中可能遇到曲面拟合问题,可以将一元最小二乘方法的有关概念和结论推广到多元最小二乘方法. 已知 m 个自变量$(x_1,\cdots,x_m)$和一个因变量 y 的一组观测值$(x_{1i},x_{2i},\cdots,x_{mi},y_i)$,$i=1,2,\cdots,n$,要确定函数 $y=f(x_1,\cdots,x_m)$,使得

$$\min J=\sum_{i=1}^{n}[f(x_{1i},x_{2i},\cdots,x_{mi})-y_i]^2. \tag{6.3}$$

一般地,首先通过机理分析或数据的直观判断,去确定函数$f(x_1,x_2,\cdots,x_m)$结构,假定函数中含有未知参数 $a_1,\cdots,a_k$,然后,通过最小二乘原理具体确定参数 $a_1,\cdots,a_k$. 下面我们给出一个实际例子.

经济增长模型

增加生产、发展经济所依靠的主要因素有增加投资、增加劳动力以及技术革新等,在研究国民经济产值与这些因素的数量关系时,由于技术水平不像资金、劳动力那样容易定量化,作为初步的模型,可认为技术水平不变,只讨论产值和资金、劳动力之间的关系. 在科学技术发展不快时,如资本主义经济发展的前期,这种模型是有意义的.

用 Q,K,L 分别表示产值、资金、劳动力,要寻求的数量关系 $Q(K,L)$. 经过简化假设与分析,在经济学中,推导出一个著名的 Cobb - Douglas 生产函数:

$$Q(K,L) = aK^{\alpha}L^{\beta}, \qquad 0 < \alpha,\beta < 1, \tag{6.4}$$

式中 α,β,a 要由经济统计数据确定. 现有美国马萨诸塞州 1900—1926 年上述三个经济指数的统计数据,如表 6.3,试用数据拟合的最小二乘法,辨识出式(6.4)中的参数 α,β,a.

表 6.3　美国马萨诸塞州 1900—1926 年三个经济指数的统计数据

t	Q	K	L	t	Q	K	L
1900	1.05	1.04	1.05	1914	2.01	3.24	1.65
1901	1.18	1.06	1.08	1915	2.00	3.24	1.62
1902	1.29	1.16	1.18	1916	2.09	3.61	1.86
1903	1.30	1.22	1.22	1917	1.96	4.10	1.93
1904	1.30	1.27	1.17	1918	2.20	4.36	1.96
1905	1.42	1.37	1.30	1919	2.12	4.77	1.95
1906	1.50	1.44	1.39	1920	2.16	4.75	1.90
1907	1.52	1.53	1.47	1921	2.08	4.54	1.58
1908	1.46	1.57	1.31	1922	2.24	4.54	1.67
1909	1.60	2.05	1.43	1923	2.56	4.58	1.82
1910	1.69	2.51	1.58	1924	2.34	4.58	1.60
1911	1.81	2.63	1.59	1925	2.45	4.58	1.61
1912	1.93	2.74	1.66	1926	2.58	4.54	1.64
1913	1.95	2.82	1.68				

该问题有两个自变量 K,L 和一个因变量 Q,已知函数结构,如公式(6.4)所示,根据表 6.3 中给定的数据 (K_i,L_i,Q_i), $i=1,2,\cdots,27$,确定未知函数 α,β,a. 其最小二乘准则如下:

$$\min_{a,\alpha,\beta} \sum_{i=1}^{27} (aK_i^{\alpha}L_i^{\beta} - Q_i)^2.$$

显然这是多元函数的参数辨识问题.

我们对该模型进行参数辨识,其 MATLAB 程序如下:

1) 建立 M 函数文件 jinjizz. m

```
function Q = jingjizz(x,y)
Q = x(1) * (y(1,:).^x(2)). * (y(2,:).^x(3));
```

其中　x(1) = a;x(2) = α;x(3) = β;

2) 建立运行文件 qumiannihe. m

```
Q = [1.05 1.18 1.29 1.30 1.30 1.42 1.50 1.52 1.46 1.60 1.69 1.81 1.93
   1.95 2.01 2.00 2.09 1.96 2.20 2.12 2.16 2.08 2.24 2.56 2.34 2.45
```

```
    2.58];
y=[1.04 1.06 1.16 1.22 1.27 1.37 1.44 1.53 1.57 2.05 2.51 2.63 2.74
   2.82 3.24 3.24 3.61 4.10 4.36 4.77 4.75 4.54 4.54 4.58 4.58 4.58
   4.54;1.05 1.08 1.18 1.22 1.17 1.30 1.39 1.47 1.31 1.43 1.58 1.59
   1.66 1.68 1.65 1.62 1.86 1.93 1.96 1.95 1.90 1.58 1.67 1.82 1.60
   1.61 1.64];
x0=[0.1,0.1,0.2];
x=lsqcurvefit('jingjizz',x0,y,Q)
```

计算结果：

```
x=1.2246    0.4612    -0.1277
```

于是公式(6.4)变为

$$Q(K,L)=1.224\,6K^{0.461\,2}L^{-0.127\,7},$$

这就是产值 Q 随资金 K、劳动力 L 的变化规律. 拟合曲面图形如下：

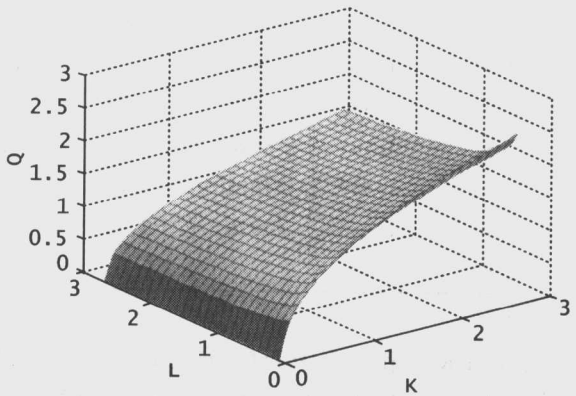


图 6.8 (K,L,Q)拟合曲面图

在上面程序中，最后一句改写为如下语句：

```
[x,resnorm,residual]=lsqcurvefit('jingjizz',x0,y,Q)
```

得到如下结果：

```
x=  1.2239    0.4610    -0.1259
resnorm=  0.4230  % 表示残差平方和
residual=……      % 表示各对应残差值
```

§6.6 操　　练

操练一　Malthus 人口指数增长模型

从 1790—1980 年间美国每隔 10 年的人口记录如表 6.4：

表 6.4　1790—1980 年间美国每隔 10 年的人口记录

年 份	1790	1800	1810	1820	1830	1840	1850
人口($\times 10^6$)	3.9	5.3	7.2	9.6	12.9	17.1	23.2
年 份	1860	1870	1880	1890	1900	1910	1920
人口($\times 10^6$)	31.4	38.6	50.2	62.9	76.0	92.0	106.5
年 份	1930	1940	1950	1960	1970	1980	
人口($\times 10^6$)	123.2	131.7	150.7	179.3	204.0	226.5	

用以上数据检验马尔萨斯(Malthus)人口指数增长模型，根据检验结果进一步讨论马尔萨斯人口模型的改进.

提示

Malthus 模型的基本假设是：人口的增长率为常数，记为 r. 记时刻 t 的人口为 $x(t)$(即 $x(t)$ 为模型的状态变量)且初始时刻的人口为 x_0，于是得到如下微分方程：

$$\begin{cases}\dfrac{dx}{dt}=rx,\\ x(0)=x_0.\end{cases}$$

需要先求微分方程的解，再用数据拟合求出模型中的参数.

操练二　旧车价格预测

某年美国旧车价格的调查资料如表 6.5，其中 x_i 表示轿车的使用年数，y_i 表示相应的平均价格. 试分析用什么形式的曲线来拟合上述的数据，并预测使用 4.5 年后轿车的平均价格大致为多少？

表 6.5　美国旧车价格的调查数据

x_i	1	2	3	4	5	6	7	8	9	10
y_i	2 615	1 943	1 494	1 087	765	538	484	290	226	204

更多的相关信息资源

1 李尚志等. 数学建模竞赛教程. 南京:江苏教育出版社,1996

2 谭永基等编著. 数学模型. 上海:复旦大学出版社,1998

3 Frank R. Giordano, Maurice D. Weir and William P. Fox. *A First Course in Mathematical Modeling*. 3rd edition. 影印版. 北京:机械工业出版社,Thomson Learning, Inc., 2003

4 傅鹂,龚劬,刘琼荪,何中市. 数学实验. 北京:科学出版社,2000

5 范金城,梅长林等编著. 数据分析. 北京:科学出版社,2002

6 George Casella, Roger L. Berger. Statistical Inference. 2rd Edition. 影印版. 北京:机械工业出版社, Press by Thomson Learning, 2002

7 James M. Lattin, J. Douglas Carroll, Paul E. Green. Analyzing Multivariate Data. 影印版. 北京:机械工业出版社,Press by Thomson Learning, 2002

第 7 章

怎样让医院的服务工作做得更好

——回归分析

面对错综复杂的各种关系，我们可能作许多的猜测、猜想和假设，但从理论上对关系的描述却很难抓住要点，因而需要在统计数据中去挖掘信息，发现本质.

——作者

汽车的重量和它消耗一加仑汽油所行驶的平均里程间有什么关系？受教育程度与收入之间是否存在着某种关系？等等. 统计学家所做的许多工作都是关注一个变量(或多个变量)是否影响另一个变量. 回归分析(regression analysis)是用统计数据寻求变量间关系的近似表达式的一种方法，并利用所得公式进行统计描述、分析和推断，解决预测、控制和优化问题.

§7.1 一份有趣的社会调查

某医院管理部门希望了解病人对医院服务工作的满意程度 Y 和病人的年龄 X_1、病情的严重程度 X_2 和忧虑程度 X_3 之间的关系，他们随机挑选了 23 位病人，对医院的服务工作进行综合打分(百分制)，同时也调查了这 23 位病人的简单情况，调查结果如下表：

表 7.1 某医院病人对医院的服务工作综合打分

i	1	2	3	4	5	6	7	8	9	10	11	12
x_{i1}	50	36	40	41	28	49	42	45	52	29	29	43
x_{i2}	51	46	48	44	43	54	50	48	62	50	48	53
x_{i3}	2.3	2.3	2.2	1.8	1.8	2.9	2.2	2.4	2.9	2.1	2.4	2.4
y_i	48	57	66	70	89	36	46	54	26	77	89	67

续表

i	13	14	15	16	17	18	19	20	21	22	23
x_{i1}	38	34	53	36	33	29	33	55	29	44	43
x_{i2}	55	51	54	49	56	46	49	51	52	58	50
x_{i3}	2.2	2.3	2.2	2.0	2.5	1.9	2.1	2.4	2.3	2.9	2.3
y_i	47	51	57	66	79	88	60	49	77	52	60

这些病人对医院的评价是否真实可信？是否年龄偏大的、病情严重的病人对医院的服务工作不满意？因此需要分析病人的年龄 X_1、病情的严重程度 X_2 和忧虑程度 X_3 与病人对医院服务工作的满意程度 Y 之间的关系.

§7.2 如何定量分析病人与医院之间的关系

病人有其自身的研究指标，如病人的年龄、病状、病情的严重程度指标等，医院里各个部门有服务状况. 欲分析 §7.1 叙述的变量 X_1、X_2、X_3 与 Y 之间的关系，需要建立数学模型. 究竟变量 X_1、X_2、X_3 与 Y 之间存在什么样的关系呢？

客观现象之间总是普遍联系和相互依存，反映这些联系的数量关系可分为两类，一类是确定性关系，另一类是不确定性关系，也称为相关关系. 对确定性关系，可用函数来描述它们，其特点是，当一个或几个变量的值取定时，相应的另一个变量的值就能完全确定. 而当一个或几个变量的值给定时，相应的另一个变量的值不能完全确定，它在一定范围内变化时，称变量之间的这种关系为不确定性关系或相关关系. 例如，人的身高与体重之间的关系、空气污染程度与人口寿命之间的关系、广告投入费用与销售量之间的关系等都可以理解为不确定的关系. 变量之间的确定性关系或不确定性关系不是永恒不变的，在一定条件下可以相互转化. 对具有确定性关系的变量，由于观测误差的存在，其表现形式也具有某种不确定性；对具有不确定性关系的变量，当我们深刻认识了它们内部之间相互联系和变化规律时，不确定性关系就可能转化成确定性关系. 对确定性关系常用数学分析的理论与方法研究，对不确定性关系，一般用概率统计的理论与方法研究，回归分析就是其中一种常用的方法.

我们将不确定性关系中作为影响因素的变量称为自变量或解释变量，用 X 表示，受 X 取值影响的变量称为因变量，用 Y 表示. 如病人的年龄 X_1、病情的严重程度 X_2 和忧虑程度 X_3 为解释变量，病人对医院服务工作的满意程度 Y 为因变量. 一般地，X_1，X_2，X_3 和 Y 都可能是随机变量，但回归分析一般假定自变量为标量，我们用 x_1，x_2，x_3 来表示，并称 x_1，x_2，x_3 是可控制变量，即它的取值是可以事

先给定的，Y 是可观测的随机变量. 其回归模型为：

$$\begin{cases} Y = f(x_1, x_2, x_3) + \varepsilon, \\ \varepsilon \sim N(0, \sigma^2), \end{cases} \tag{7.1}$$

其中偏差 $\varepsilon = Y - f(x_1, x_2, x_3)$ 可能有两个原因所致，一是观测过程中由于对 Y 的测量误差导致偏差，另一个是除了自变量 x_1, x_2, x_3 以外，还有其他未考虑到的影响因素而导致偏差.

倘若知道了 $y = f(x_1, x_2, x_3)$，则可以从数量上掌握 x_1, x_2, x_3 与 Y 之间复杂关系的大趋势，就可以利用这种趋势研究回归模型的预测问题和控制问题. 这就是回归分析处理不确定性关系的基本思想.

模型(7.1)中含有两层含义：第一，平均偏差为零，即 $E\varepsilon = 0$；第二，偏差 ε 呈现的统计规律是正态分布. 这是人为假定的，原因是这种假定便于进行统计推断，但这种假定的合理性有待进一步论证.

若在模型(7.1)的基础上，进一步假定 Y 与 x_1, x_2, x_3 是线性函数关系，则回归模型为

$$\begin{cases} Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon, \\ \varepsilon \sim N(0, \sigma^2), \end{cases} \tag{7.2}$$

称模型(7.2)为线性回归模型. 其中 $\beta_i, i = 0, 1, 2, 3$ 为未知参数，需要由统计数据去确定.

在实际问题中，理论回归函数 $f(x_1, x_2, x_3)$ 一般总是未知的，回归分析的任务就是根据 x_1, x_2, x_3 的值和 Y 的观测值去估计这个函数以及讨论与此有关的种种统计推断问题，所用方法在相当大的程度上取决于回归模型的假定. 对 $y = f(x_1, x_2, x_3)$ 的数学形式无特殊假定的回归分析称为“非参数回归”，对已知 $y = f(x_1, x_2, x_3)$ 的数学形式，只是其中的若干个参数未知的回归分析称为“参数回归”，这是目前研究最多、应用最多的情形.

需要解决的基本问题是：

1）如何根据抽样信息确定回归函数类型及其参数的估计量；

2）如何判断 x_1, x_2, x_3 与 Y 之间的关系是否密切；

3）变量 x_1, x_2, x_3 是否都对指标 Y 影响显著？

4）如何应用回归分析进行预测或控制.

线性回归(linear regression)是应用上最重要、理论上较完善的回归分析方法.

§7.3 回归分析

7.3.1 线性回归模型

回归分析是基于观测数据建立变量间适当的依赖关系，以分析数据的内在规律，并可用于预测、控制等问题.

线性回归模型：

$$\begin{cases} Y=\beta_0+\beta_1 x_1+\cdots+\beta_m x_m+\varepsilon, \\ \varepsilon\sim N(0,\sigma^2), \end{cases} \tag{7.3}$$

该模型具有两个基本假定：

1）有 m 个自变量 $x_1,\cdots,x_m$，它们与因变量 Y 构成线性关系；

2）偏差 ε 的数学期望为 0，方差为 σ^2，并且服从正态分布.

特殊情形，若只有一个自变量 x，一个因变量 Y，模型(7.3)变为如下形式：

$$\begin{cases} Y=\beta_0+\beta_1 x+\varepsilon, \\ \varepsilon\sim N(0,\sigma^2) \end{cases} \quad \text{或} \quad Y\sim N(\beta_0+\beta_1 x,\sigma^2), \tag{7.4}$$

称模型(7.4)为一元线性回归模型.

1. 关于回归系数 $\beta_0,\beta_1,\cdots,\beta_m$ 的确定

根据观测数据 $(x_{i1},x_{i2},\cdots,x_{im},y_i)$，$i=1,2,\cdots,n$，采用最小二乘法确定回归系数 $\beta_0,\beta_1,\cdots,\beta_m$. 观测数据应满足模型(7.3)，即

$$\begin{aligned} y_1&=\beta_0+\beta_1 x_{11}+\cdots+\beta_m x_{1m}+\varepsilon_1, \\ &\cdots\cdots\cdots \\ y_n&=\beta_0+\beta_1 x_{n1}+\cdots+\beta_m x_{nm}+\varepsilon_n, \end{aligned}$$

$$\varepsilon_i\sim N(0,\sigma^2),\operatorname{Cov}(\varepsilon_i,\varepsilon_j)=0,i\neq j,i,j=1,2,\cdots,n,$$

建立优化目标函数：

$$\min Q(\beta_0,\beta_1,\cdots,\beta_m)=\sum_{i=1}^{n}\varepsilon_i^2=\sum_{i=1}^{n}[y_i-(\beta_0+\beta_1 x_{i1}+\cdots+\beta_m x_{im})]^2.$$

为此，确定回归模型中的参数 β_i，$i=0,1,\cdots,m$，转化为求解一个优化问题.

令

$$\frac{\partial Q(\beta_0,\beta_1,\cdots,\beta_m)}{\partial\beta_i}=0,i=0,1,\cdots,m, \tag{7.5}$$

式(7.5)是 $m+1$ 元线性方程组，经过整理，得正则方程组：$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\beta}$，其中

$$\boldsymbol{X}=\begin{bmatrix}1 & x_{11} & \cdots & x_{1m}\\ 1 & x_{21} & \cdots & x_{2m}\\ \vdots & \vdots & & \vdots\\ 1 & x_{n1} & \cdots & x_{nm}\end{bmatrix},\boldsymbol{Y}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix},\boldsymbol{\beta}=\begin{bmatrix}\beta_0\\ \beta_1\\ \vdots\\ \beta_m\end{bmatrix} \tag{7.6}$$

只要$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$存在，可以求解正则方程，得到的解记为$\hat{\boldsymbol{\beta}}$，它是样本$(x_{i1},x_{i2},\cdots,x_{im},y_i)$，$i=1,2,\cdots,n$的函数，称$\hat{\boldsymbol{\beta}}$为参数$\boldsymbol{\beta}$的估计量，即$\hat{\boldsymbol{\beta}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$. 定义回归方程如下：

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1x_1+\cdots+\hat{\beta}_mx_m.$$

所求的回归方程是否有意义，也就是说Y与$x_1,\cdots,x_m$之间是否存在显著的线性关系，还需要对回归方程进行检验.

2. 回归模型的检验

“Y与$x_1,\cdots,x_m$之间不存在线性关系”等价于检验如下的数学问题：

$$H_0:\beta_0=\beta_1=\cdots=\beta_m=0,$$

主要有F检验法. 理论上可以证明平方和分解公式：

$$\sum_{i=1}^{n}(y_i-\bar{y})^2=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2,$$

记为

$$SST=SSE+SSR,$$

其中称SSE为残差平方和，自由度为$n-m-1$，它反映了除Y与$x_1,\cdots,x_m$之间以外的因素引起的数据$y_1,\cdots,y_n$的波动. 若$SSE=0$，则每个观测值可由线性关系精确拟合，SSE越大，反映观测值与线性拟合值间的偏差越大. 称SSR为回归平方和，自由度为m，它反映了线性拟合值与它们的平均值的总偏差. 若$SSR=0$，反映了$\hat{y}_i=\bar{y}$，即不随$x_1,\cdots,x_m$的变化而变化，说明H_0成立. 相反，若SSR越大，反映Y与$x_1,\cdots,x_m$之间的线性越显著. SST为总的离差平方和.

构造F检验统计量，当H_0为真时，可证明：

$$F=\frac{SSR/m}{SSE/(n-m-1)}\sim F(m,n-m-1), \tag{7.7}$$

拒绝域为$\{F>F_{1-\alpha}(m,n-m-1)\}$.

将观测数据代入(7.7)中的F统计量中，计算F值，若F值大于查表值$F_{1-\alpha}(m,n-m-1)$，则拒绝H_0，认为Y与$x_1,\cdots,x_m$之间存在显著的线性回归关系.

除了使用F统计量检验线性回归关系的显著性以外，还有两个评价指标，一个是相关系数r，另一个是p值.

样本复相关系数：

$$\hat{r}=\sqrt{\frac{\sum_{i=1}^{n}\left(\hat{y}_i-\bar{y}\right)^2}{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}}=\sqrt{\frac{SSR}{SST}} \tag{7.8}$$

主要用于评价 Y 与 $X_1,\cdots,X_m$ 之间的线性关系的强度. 特别地,对 Y 与 X 之间的线性相关性的评价指标——样本相关系数:

$$\hat{r}=\frac{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2}\cdot\sqrt{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}}, \tag{7.9}$$

其中 $\hat{r}$ 是理论相关系数 r 的一个点估计值,$0\leqslant|\hat{r}|\leqslant 1$. 同理分析,当 $|\hat{r}|$ 接近于 1 时,说明 Y 与 $X_1,\cdots,X_m$ 之间的线性关系显著;当 $|\hat{r}|$ 接近零时,表明 Y 与 $X_1,\cdots,X_m$ 之间的线性关系不明显,随机因素起主要作用,或者 Y 与 $X_1,\cdots,X_m$ 之间可能存在着非线性的关系.

检验的 p 值:

$$p=P(F>F_0\mid H_0\text{成立}),$$

其中 F_0 为检验统计量 F 的观测值. 对于给定的显著性水平 α,其检验准则为:

$$\begin{cases}\text{若 } p<\alpha, & \text{拒绝 } H_0,\\ \text{若 } p\geqslant\alpha, & \text{接受 } H_0.\end{cases} \tag{7.10}$$

使用 r 值或 p 值的检验方法其特点是不需要相应分布的分位数表,而直接根据 r 值与 1 的比较、p 值与 α 的比较便可判断是拒绝 H_0 还是接受 H_0. 这种判断方法在 SAS、SPSS、MATLAB 等软件中经常使用.

3. 回归系数的检验以及最优回归方程的确定

回归线性关系显著并不意味着每个变量 $X_i,i=1,2,\cdots,m$ 对 Y 的影响都显著,可能其中的某个或某些对 Y 的影响不显著. 一般我们总希望从回归方程中剔除那些对 Y 的影响不显著的自变量,从而建立一个较为简单有效的回归方程,以便于实际使用,这就需要对每一个自变量作考察. "某个自变量 X_i 对 Y 无影响"等价于检验假设

$$H_0:\beta_i=0\quad(i=1,\cdots,m),H_1:\beta_i\neq 0.$$

可以证明:$\hat{\beta}_i\sim N(\beta_i,c_{ii}\sigma^2)(i=1,\cdots,m)$,其中 c_{ii} 为矩阵 $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 的对角线上第 i 个元素,我们构造检验统计量:

$$T_i=\frac{\hat{\beta}_i/\sqrt{c_{ii}}}{\sqrt{SSE/(n-k-1)}}\sim t(n-m-1)\quad(i=1,2,\cdots,m). \tag{7.11}$$

检验准则是:当 $|T_i|>t_{1-\alpha/2}(n-m-1)$,则拒绝 H_0. 并且还可用(7.11)的结果求

95% 的置信区间. 若检验结果是接受 H_0，则可以说明自变量 X_i 对因变量 Y 的影响比较小，可以将该变量 X_i 从已经建立起来的回归模型中剔除. 实际上，该统计检验结果成为剔除哪些自变量的一个重要依据.

许多实际问题往往涉及大量的自变量，当回归函数的类型选定为线性函数时，一个重要的问题就是自变量的选取问题. 由于包含较多自变量的模型拟合的计算量大，不便于利用拟合的模型对实际问题作解释，况且理论上可以证明预报值的方差随自变量数目的增加而增加. 因此，在实际应用中，希望拟合这样一个模型，它既能较好的反映问题的本质，又包含尽可能少的自变量. 这两个方面的一个适当折中就是回归方程的选取问题，其基本思想是在一定的准则下选取对因变量影响较为显著的自变量，建立一个既合理又简单实用的回归模型.

选取回归方程的回归变量方法主要有穷举法、逐步回归法等，如 SAS、SPSS、MATLAB 软件中较多使用的是逐步回归法(step-wise regression). 逐步回归的基本思想是：将回归变量一个一个地选入，选入的条件是偏回归平方和显著得大；每选入一个新变量后，对已选入的各变量逐个进行显著性检验，并剔除不显著变量. 如此反复选入、检验、剔除，直至无法选入和无法剔除为止.

4. 回归模型应用

建立回归模型主要目的在于应用，主要有两方面的应用——预测(prediction)与控制(control)，下面简单介绍预测的基本原理.

给定一组新的观测值$(x_{10}, x_{20}, \cdots, x_{m0})$，利用回归方程可得因变量 Y 的预测值：

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_{10} + \cdots + \hat{\beta}_m x_{m0},$$

$\hat{y}_0$ 实际上是对应于$(x_{10}, x_{20}, \cdots, x_{m0})$的 Y 的一个点估计. 但在实际应用中，更感兴趣的是给出 Y 的真值 y_0 的区间估计，具体公式略.

7.3.2 非线性回归模型

$$\begin{cases} Y = f(x_1, x_2, \cdots, x_m) + \varepsilon, \\ \varepsilon \sim N(0, \sigma^2). \end{cases}$$

若函数 $f(x_1, x_2, \cdots, x_m)$ 为非线性函数形式，可以想像对该模型进行统计分析较为困难. 若已知函数形式如指数函数、分式函数等，则问题转换为确定某些参数问题，在 MATLAB 软件中只需进行参数的曲线拟合即可. 关于这方面的统计推断我们不详细叙述.

§7.4 病人对医院的评价如何

1. 模型假设

1）病人有三项指标：年龄 X_1、病情的严重程度 X_2 和忧虑程度 X_3，它们构成模型的回归自变量；

2）因变量是病人对医院服务工作的满意程度 Y；

3）自变量 X_1，X_2，X_3 与因变量 Y 之间具有显著的线性关系，且考虑（X_1，X_2，X_3）固定取几组值；

4）实际观测值与估计值之间的偏差均值为 0，方差为 σ^2，并且实际观测值的统计规律为正态分布.

2. 回归模型

$$\begin{cases} Y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\varepsilon, \\ \varepsilon\sim N(0,\sigma^2). \end{cases} \tag{7.12}$$

3. 使用 MATLAB 软件编程计算

计算步骤：

1）输入数据

```
A=[50 36 40 41 28 49 42 45 52 29 29 43 38 34 53 36 33 29 33 55 29 44 43;
   51 46 48 44 43 54 50 48 62 50 48 53 55 51 54 49 56 46 49 51 52 58 50;
   2.3 2.3 2.2 1.8 1.8 2.9 2.2 2.4 2.9 2.1 2.4 2.4 2.2 2.3 2.2 2.0 2.5 1.
   9 2.1 2.4 2.3 2.9 2.3];
a=ones(23,1);
X=[a,A'];% 23×4 阶矩阵,其结构见公式(7.6).
Y=[48 57 66 70 89 36 46 54 26 77 89 67 47 51 57 66 79 88 60 49 77 52 60]';
Alpha=0.05;
```

2）MATLAB 调用格式：[b,bint,r,rint,stats]=regress(Y,X,alpha)

3）输出结果

```
b=
    162.8759
     -1.2103
     -0.6659
     -8.6130
bint=
     108.9268  216.8250
      -1.8413   -0.5794
      -2.3843   1.0525
```

```
        -34.2343   17.0082
    r = -0.5888   -11.8628   2.4490   1.5504   4.1504   -6.6336   -13.7986
  -1.7768   -7.6754   0.6060   13.8581   12.1321   -14.3103   -16.9539
13.1785   -3.4490   14.8879   7.2197   -12.2187   7.3241   3.6604   5.9784
2.2730
    rint =
        -21.1751   19.9974
        -30.8642    7.1386
        ... ...
    MSE = r'*r/(n-m-1) =105.8729
       stats =   0.6727    13.0145    0.0001
```

4. 结果分析

1）回归模型中参数 $\beta_0,\beta_1,\beta_2,\beta_3,\sigma^2$ 的估计值

由以上输出结果知，回归模型(7.12)中的参数分别是

$$\hat{\beta}_0 = 162.875\,9,\hat{\beta}_1 = -1.210\,3,\hat{\beta}_2 = -0.665\,9,\hat{\beta}_3 = -8.613\,0,$$

$$\hat{\sigma}^2 = MSE = 105.872\,9,$$

回归方程：$\hat{y} = 162.875\,9 - 1.210\,3x_1 - 0.665\,9x_2 - 8.613\,0x_3$.

2）模型检验

需要检验 $H_0:\beta_1 = \beta_2 = \beta_3 = 0$.

a. F 检验法

关于 F 统计量的计算公式：

$$F = \frac{SSR/m}{SSE/(n-m-1)}.$$

由 3)输出结果，F 值从 stats 中读取，$F = 13.014\,5$，另一方面，查 F 分布表，$F_{0.95}(3,19) = 3.13$，显然 $F > F_{0.95}(3,19)$，根据 F 检验准则知，拒绝 H_0，即认为 X_1,X_2,X_3 与 Y 的线性关系显著.

b. 相关系数 r 的评价

stats 中的第一个数据就是相关系数 r 的平方，即 $r^2 = 0.672\,7$，则 $|r| = 0.820\,2$，一般地，相关系数绝对值在 0.8 ~ 1 范围内，可判断回归自变量与因变量具有较强的线性相关性.

c. p 值检验

stats 中的第三个数据就是 p 值，即 $p = 0.000\,1$，显然满足 $p < \alpha = 0.05$，同样说明回归自变量 X_1,X_2,X_3 与因变量 Y 的线性关系显著.

以上使用三种统计推断方法推断的结果是一致的，都认为自变量 X_1,X_2,X_3 与因变量 Y 的线性关系显著. 说明以上模型假设和回归模型能够基本反映 X_1,X_2,X_3 与 Y 的关系.

3）残差分析

在拟合一个回归模型之前，人们并不能肯定这个模型适合于所给数据. 诸如对回归函数的线性假设、误差的正态性和同方差性假设等，都有可能不适合于所给数据. 拟合模型以后，再进一步考察模型对所给数据的适用性，也是十分重要的一个环节. 如果拟合的模型不能较好的反映数据的特点，就必须对模型作必要的修正或者对数据作某些处理. 在这一方面，残差分析起着十分重要的作用.

残差 $e_i = y_i - \hat{y}_i (i = 1, \cdots, n)$ 是 Y 的各观测值 y_i 与回归方程所对应得到的拟合值 $\hat{y}_i$ 之差. 如果模型正确，则可将 e_i 近似看作第 i 次的测量误差，真正的测量误差是 $\varepsilon_i = y_i - E(y_i)$ $(i = 1, \cdots, n)$ 是未知的. 在回归分析中，我们经常假定 $\varepsilon_i (i = 1, \cdots, n)$ 是独立同正态分布的随机变量，有零均值和常值方差 σ^2. 因此，若拟合的回归模型适合于所给数据，那么残差 e_i 应基本上反映未知误差 ε_i 的这些特性. 这种利用残差的特征反过来考察原模型的合理性就是残差分析的基本思想.

a. 残差向量正态性的图形检验

使用 MATLAB 程序进行线性回归分析，输出的结果参见程序，其中 r，rint 分别表示残差向量和残差向量的区间估计，即 $e_i = r_i$ $(i = 1, 2, \cdots, 23)$. 利用残差向量 r 和 MATLAB 语句：normplot(r)，得到如下图形：

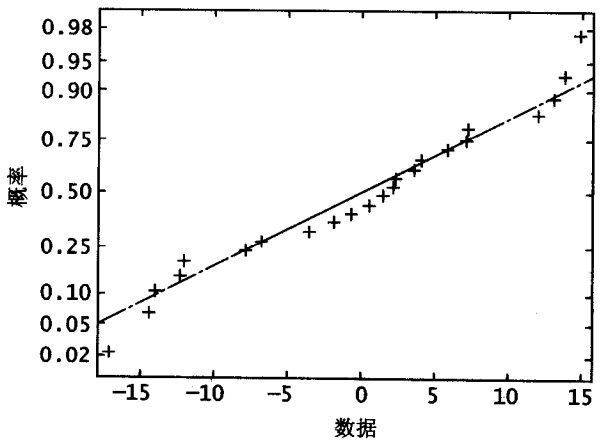


图 7.1　残差向量的正态性检验图

理论上可以证明，若 $e_i = r_i (i = 1, 2, \cdots, 23)$ 是来自正态分布总体的样本，则点“+”呈现的散点应在一条直线上. 从以上图形可知，误差的正态性假设是合理的.

b. 残差图分析

残差图是指以残差为纵坐标，以任何其他指定的量为横坐标的散点图. 主要包括：Ⅰ）横坐标为观测时间或观测值序号；Ⅱ）横坐标为 Y 的拟合值 $\hat{Y}$；Ⅲ）横坐标为某个自变量 X_j $(j=1,2,\cdots,m)$ 的观测值. 通过考察残差图，可以对奇异点进行分析，还可以对误差的等方差性以及对回归函数中是否应包含其他的自变量、自变量的高次项及交叉乘积项等问题给出直观的检验.

① 时序残差图

利用残差向量 r，残差的区间估计值 rint 和 MATLAB 语句：rcoplot(r, rint)，得到如 7.2 所示的图形：

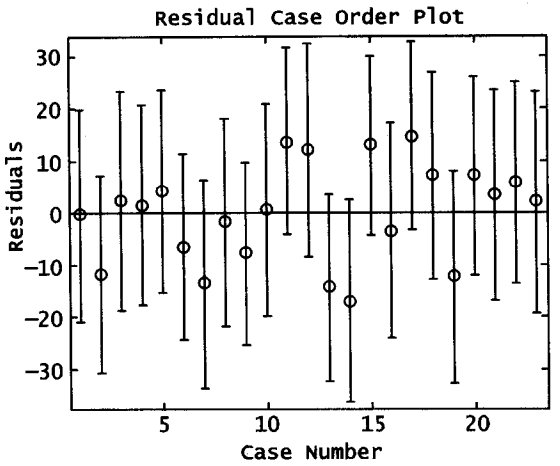


图 7.2　时序残差图

以观测值序号为横坐标，残差为纵坐标所得到的散点图称为时序残差图. 在图 7.2 中，符号“∘”表示某观测值的残差值，以符号“∘”为中心上下的一条直线表示某残差值 r_i 的波动范围. 拟合较好的模型的时序残差图中的残差值应落在以“$y=0$”为中轴线的带状区域内，且无明显的趋势. 如果存在某个残差值偏离中轴线很远，则判断这个点是奇异点，应删除. 删除奇异点以后，应重新建立回归方程. 图 7.2 说明数据没有奇异点，并且建立的线性回归模型比较适合于样本数据.

一般地，图 7.3(a) 显示模型拟合数据效果比较好；图 7.3(b) 说明回归函数中应包含时间的二次项作为自变量；图 7.3(d) 表示回归函数中应包含时间的线性项；图 7.3(c) 表明误差方差随时间而增大，即等方差的假定是不合理的.

② 其他横坐标的残差图

图7.4(a) 是以拟合值 $\hat{Y}$ 为横坐标的残差图，图7.4(b) 是以自变量 x_1 为

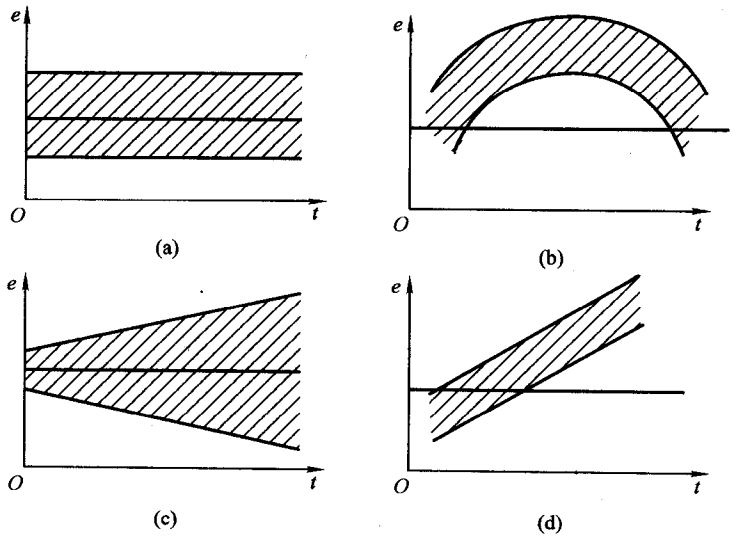


图 7.3　时序残差示意图

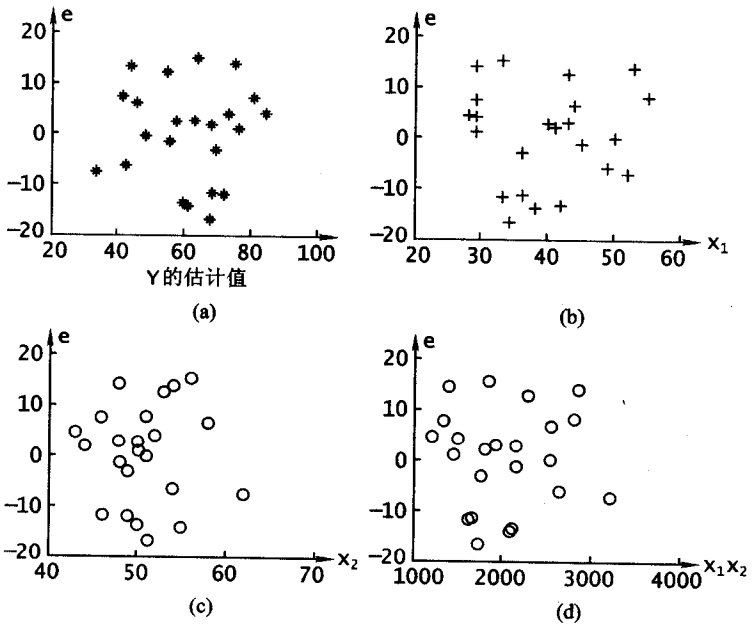


图 7.4　其他横坐标的残差图

横坐标的残差图，图 7.4(c)和图 7.4(d)分别是以横坐标为 x_2，x_1x_2 的残差图. —

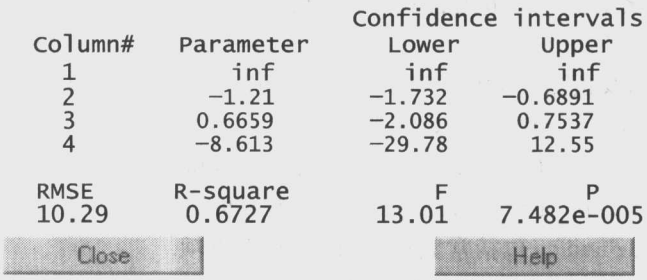
般它们出现的形状类似于图 7.3 的四种，如果出现图 7.3 中的图(a)的形状，则说明数据适合于拟合的模型. 否则，数据不适合于拟合的模型，要修改所建立的回归模型.

对病人与医院之间关系的采样数据，采用 X_1, X_2, X_3 的线性回归模型拟合，由图 7.4 各残差图可知，它们没有明显的趋势性变化，是比较满意的形式.

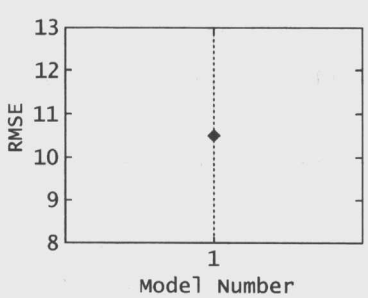
4) 最优回归方程的选取

主要通过筛选自变量，得到一个最佳的回归方程，常用的方法是逐步回归法. MATLAB 软件使用格式是：`stepwise(X,Y,inmodel,alpha)`，其中 `X,Y` 表示自变量和因变量的数据矩阵，`inmodel` 表示包含在初始模型中的矩阵 $\boldsymbol{X}$ 的列号所组成的向量.

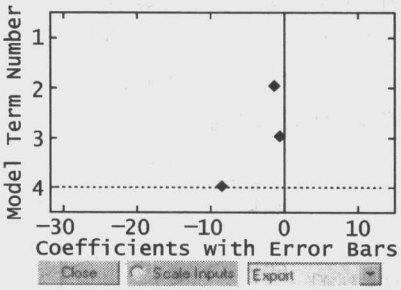
例如，该问题输入命令：`stepwise(X,Y,[2,3,4],0.1)`，将得到三个图形如下：



(a) 逐步回归诊断表



(b) 历史记载图



(c) 逐步回归图

图 7.5

图 7.5(a)显示回归及方差分析的各种信息，有回归方程的系数、系数的区间估计值、均方误差、复相关系数平方、F 统计量值和 p 值. 图 7.5(b)表示均方误差历史记载图，当去掉某个自变量 X_1，只需要将鼠标移动到图 7.5(a)中 X_1 下点击其系数，则图 7.5(b)将会显示在这种情况下均方误差的值，而评判标准是：均方误差越小越好. 图 7.5(c)显示的是回归系数的误差条状图.

通过对病人与医院之间关系模型的逐步回归分析知，最佳的回归方程是所有自变量 X_1, X_2, X_3 都在回归模型中，并且是线性函数关系，即前面计算得到的回归方程就是最佳的回归方程.

5）回归模型应用

如果有一个新病人，其病人特征是：年龄 $x_1 = 53$，病情的严重程度 $x_2 = 60$，忧虑程度 $x_3 = 2.5$，问该病人如何评价医院的服务质量？评判分数在什么范围内？

利用最佳回归方程：$\hat{y} = 162.875\ 9 - 1.210\ 3x_1 - 0.665\ 9x_2 - 8.613\ 0x_3$，将病人的特征信息代入方程，将得到 Y 的点估计值.

MATLAB 程序：x0 = [1,53,60,2.5]; y0 = b' * x0'

计算结果：y0 = 37.242 1

即该病人对医院的服务工作进行综合打分是 37 分. 但这只是一个近似值，近似程度如何？应该给出一个范围，即求出预测区间.

使用 MATLAB 语句：rstool(X,Y,'inmodel',alpha)，可以拟合二次响应曲面回归模型以及预测的交互式界面. 其中'inmodel'表示四种模型选择方案：

linear（缺省）：包括常数以及线性项；

interaction：包括常数项、线性项和交叉乘积项；

quadratic：在“interaction”的内容上再增加平方项；

purequadratic：包括常数项、线性项和平方项.

结合该问题，只要输入 MATLAB 程序：rstool(X,Y)，将得到一个交互式的界面图 7.6，它是默认为线性回归模型的形式. 在图 7.6 中，有三个窗口，分别输入 $x_1 = 53, x_2 = 60, x_3 = 2.5$，则图形左侧显示数据：37.242 1 ± 22.480 8，它既是点 $x_0 = [1,53,60,2.5]$ 处的预测区间，即 [14.761 3, 59.722 9]，也是点估计 $y_0 = 37.242\ 1$ 值的活动范围.

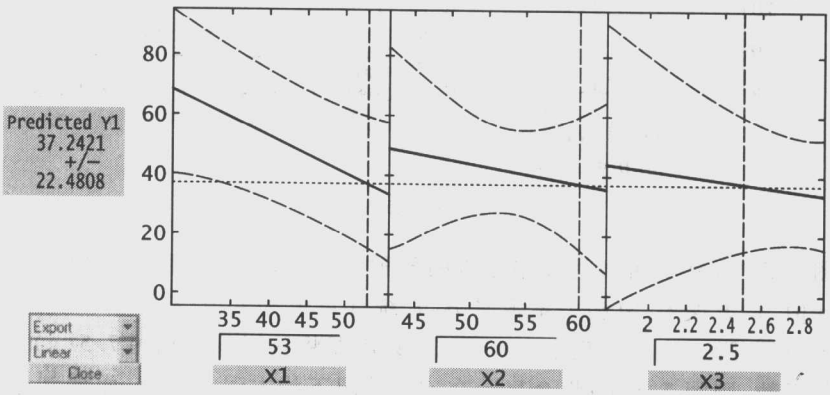


图 7.6　二次曲面交互界面图

§7.5 简介非线性回归分析

如果病人与医院之间的关系不是线性关系，而是非线性关系，这时如何分析和计算？非线性回归模型是回归函数关于未知参数具有非线性结构的回归模型. 模型的拟合一般很困难，通常首先需要猜测未知的初始值，然后反复迭代. 每次迭代都会修正当前的估计值，直至算法收敛为止. 本节以一个例子为例，简单介绍使用 MATLAB 软件分析和计算非线性回归问题.

非线性回归模型：

$$\begin{cases} Y = f(x_1, x_2, \cdots, x_m) + \varepsilon, \\ \varepsilon \sim N(0, \sigma^2). \end{cases}$$

1. 问题

为了研究三种化学物质：氢、n - 戊烷和异戊烷与某物质的反应速度 Y(%) 之间的关系，测定得到了表 7.2 所示的数据. 试建立非线性回归模型，并进行统计分析.

表 7.2 氢、n - 戊烷和异戊烷与某物质的反应速度数据

序号	氢 $x1$	n - 戊烷 $x2$	异戊烷 $x3$	反应速度 Y
1	470	300	10	8.55
2	285	80	10	3.79
3	470	300	120	4.82
4	470	80	120	0.02
5	470	80	10	2.75
6	100	190	10	14.39
7	470	80	65	2.54
8	100	190	65	4.35
9	100	300	54	13.00
10	100	300	120	8.5
11	100	80	120	0.05
12	285	300	10	11.32
13	285	190	120	3.13

2. 假设及建模

假定该问题需要建立一个非线性回归模型. 有以下两种情况:

1) 在各因素与指标(因变量)之间的信息"一无所知"的情况下,假设模型 $Y=f(x_1,x_2,x_3)+\varepsilon$ 中的函数 f 是多项式形式,这是常用的一种方法. 最简单的情形是二次多项式,即假设模型为:

$$\begin{aligned} y=&b_0+b_1x_1+b_2x_2+b_3x_3+ && \text{(linear terms)}\\ &b_{12}x_1x_2+b_{13}x_1x_3+b_{23}x_2x_3+ \quad (*) && \text{(interaction terms)}\\ &b_{11}x_1^2+b_{22}x_2^2+b_{33}x_3^2+\varepsilon, && \text{(quadratic terms)}\\ &\varepsilon\sim N(0,\sigma^2), \end{aligned}$$

称该模型为**二次曲面模型**(quadratic surface model)或二次多项式模型.

2) 假定由实际问题背景分析知经验公式为:

$$Y=\frac{\beta_1x_2-x_3/\beta_5}{1+\beta_2x_1+\beta_3x_2+\beta_4x_3}+\varepsilon,$$

称该模型为**非线性模型**(nonlinear model).

注意 尽管在两种假设下所建立的模型不同,但解决问题的目标是一致的,需要分别辨识模型中出现的参数 b_i,b_{ij},β_i 的估计值,并对指标 Y 进行预测.

下面分别对两种模型进行参数辨识和统计分析.

3. MATLAB 实现

1) 二次多项式(曲面)拟合与预测,首先输入数据并存储,便于今后访问. MATLAB 的用法是使用命令 rstool. 它产生一个交互式画面,并输出有关信息,具体用法是

```
rstool(X,y,'model',alpha)
```

其中 X、y 分别是自变量的数据矩阵($n\times m$)和因变量的数据向量($n\times 1$). alpha为显著性水平 α(缺省时设定为 0.05). 'model'表示使用什么模型,如使用二次曲面模型,则输入 quadratic. 一般有四种模型选择:

linear(缺省): $y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m$,

purequadratic: $y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{j=1}^{m}\beta_j^*x_j^2$,

interaction: $y=\beta_0+\beta_1x_1+\cdots+\beta_mx_m+\sum_{j,k=1}^{m}\beta_{jk}x_jx_k\ j\neq k$,

quadratic(完全二次,参见公式(*))

根据表 7.2 中的数据,我们计算出在各种模型情况下的多项式中各项的系数如表 7.3 和交互式画面图 7.7.

表 7.3　在各模型下的参数估计表

Full Quadratic	Linear	Pure Quadratic	Interactions
6.174 7(c)	5.818 4	4.591 2	5.819 2
−0.040 3(x_1)	−0.010 9	−0.037 7	−0.011 6
0.097 5(x_2)	0.033 2	0.101 0	0.049 3
−0.137 5(x_3)	−0.048 1	−0.095 0	−0.083 4
0.000 0(x_1x_2)		0.000 0	0.000 0
0.000 1(x_1x_3)		−0.000 2	0.000 1
0.000 0(x_2x_3)		0.000 3	0.000 0
0.000 1(x_1^2)			
−0.000 1(x_2^2)			
0.000 4(x_3^2)			

关于残差分析:(在 Full Quadratic 下)

总的残差平方和:　rmse = 0.182 0

还可以计算各种情况下的残差量.(residuals)

图 7.7 是一个交互式的图形界面.给出了三幅图形,它们分别表示独立变量 x_i(另两个变量取固定值)与 Y 的拟合曲线关系.每个独立变量的取值可以在每个坐标图形下的编辑内容框中确定,只需在框中键入一个新的值或拖曳 3 个纵轴线到新的位置来改变独立变量的取值.

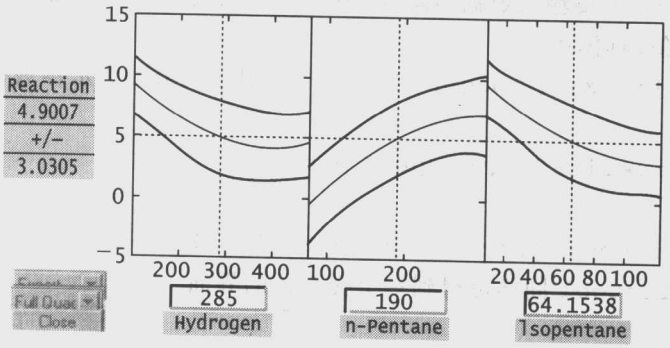


图 7.7　二次曲面交互式界面图

图 7.7 的左下方有两个下拉式菜单，一个是输出菜单，用以向 MATLAB 工作区传送数据，包括 beta(回归系数)，rmse(剩余标准差)，residuals(残差)．另一个是选项菜单，用以选择以下四个模型：

Full Quadratic：包含完整的二次曲面的各项．

Linear：只有常数和一次项．

Pure Quadratic：含有常数、线性和平方项．

Interactions：含有常数、线性和交叉项．

当前所举的例子仅用了三个独立变量，而命令 rstool 还可以适用于任意多个独立变量．但交叉项可能会受到输入容量的限制．

实际问题中应如何选择模型呢？选择模型的标准是什么？

2）一般的非线性模型．例如

$$Y=\frac{\beta_1 x_2-x_3/\beta_5}{1+\beta_2 x_1+\beta_3 x_2+\beta_4 x_3}+\varepsilon.$$

若仅仅估计各参数 β_i，这个问题就是曲线拟合的问题，可以使用 curvefit 命令．但如何同时进行参数估计和统计分析呢？

首先建立 M 文件函数(hougen．m)

```
function yhat = hougen(b,x)
yhat = (b(1) * x(2) - x(3) / b(5)) . / (1 + b(2) * x(1) + b(3) * x(2) + b(4) * x(3));
```

指令 nlinfit 需要输入数据矩阵 X、因变量数据向量 y、函数形式和要辨识的参数的初始值．具体用法：

```
nlinfit(X,y,'hougen',beta)
```

其中 beta 表示参数初值．

或

```
curvefit(X,y,'hougen',beta)
```

或

```
nlintool(X,y,'hougen',beta,0.01)
```

其中命令 nlintool 同样可得到一个交互式的界面．

MATLAB 已经保存有数据文件 reaction．mat，首先用命令 load 将数据调出．请你使用以上命令进行练习．

§7.6 操　练

操练一　某类研究学者的年薪与相关因素分析

对于工薪阶层的人群关心年薪与哪些因素有关，以此可制定出他们自己的奋斗目标.

某科学基金会希望估计从事某研究学者的年薪 Y 与他们的研究成果(论文、著作等)的质量指标 X_1、从事研究工作的时间 X_2、能成功获得资助的指标 X_3 之间的关系，为此按一定的试验设计方法调查了 24 位研究学者，得到如下数据：

表 7.4　从事某种研究的学者的相关指标数据

i	1	2	3	4	5	6	7	8	9	10	11	12
x_{i1}	3.5	5.3	5.1	5.8	4.2	6.0	6.8	5.5	3.1	7.2	4.5	4.9
x_{i2}	9	20	18	33	31	13	25	30	5	47	25	11
x_{i3}	6.1	6.4	7.4	6.7	7.5	5.9	6.0	4.0	5.8	8.3	5.0	6.4
y_i	33.2	40.3	38.7	46.8	41.4	37.5	39.0	40.7	30.1	52.9	38.2	31.8
i	13	14	15	16	17	18	19	20	21	22	23	24
x_{i1}	8.0	6.5	6.6	3.7	6.2	7.0	4.0	4.5	5.9	5.6	4.8	3.9
x_{i2}	23	35	39	21	7	40	35	23	33	27	34	15
x_{i3}	7.6	7.0	5.0	4.4	5.5	7.0	6.0	3.5	4.9	4.3	8.0	5.0
y_i	43.3	44.1	42.5	33.6	34.2	48.0	38.0	35.9	40.4	36.8	45.2	35.1

试建立 Y 和 X_1, X_2, X_3 之间关系的数学模型，并对该模型进行各种统计分析，能得到一个什么样的结论？

操练二　人们对某种品牌食品的评价

为了研究人们对某种品牌食品的喜爱程度 Y 和该食品的水分含量 X_1 和甜度 X_2 的关系，进行了一个完全随机化设计的小规模试验，得到下列数据：

表 7.5　某品牌食品的水分含量、甜度和人们的喜爱程度数据

i	1	2	3	4	5	6	7	8
x_{i1}	4	4	4	4	6	6	6	6
x_{i2}	2	4	2	4	2	4	2	4
y_i	64	73	61	76	72	80	71	83
i	9	10	11	12	13	14	15	16
x_{i1}	8	8	8	8	10	10	10	10
x_{i2}	2	4	2	4	2	4	2	4
y_i	83	89	86	93	88	95	94	100

试建立线性回归拟合模型，对 $\hat{\beta}_1$ 如何解释？并做进一步地分析

1）求出残差向量，分别做出残差关于拟合值 $\hat{Y}, X_1, X_2$ 以及 X_1X_2 的残差图及残差的正态图，具体分析并予以评述.

2）对 ε_i 给出合理的假设，给出一组新的数据观测值 $x_0 = (5,4)$，给出 Y 的预报值和 99% 的置信区间.

3）拟合 Y 关于 X_1 的一元线性回归模型，与二元线性回归模型作比较，由此得出什么结论？

更多的相关信息资源

1 范金城，梅长林等编著. 数据分析. 北京：科学出版社，2002

2 李涛，贺勇军等. MATLAB 工具箱应用指南——应用数学篇. 北京：电子工业出版社，2001

3 George Casella, Roger L. Berger. Statistical Inference. 2nd Edition. 影印版. 北京：机械工业出版社，Thomson Learning，2002

4 James M. Lattin, J. Douglas Carroll, Paul E. Green. Analyzing Multivariate Data. 影印版. 北京：机械工业出版社，Thomson Learning，2002

第 8 章

海港系统卸载货物的计算机模拟

设想,你到某大型的超市里购物,当然期望占用较少的时间购买到你所需要的物品.假设购物占用时间主要考虑收银台前排队等候,请问,应设置多少个收银台,使每个顾客平均等待时间不超过 3 分钟? 怎样建立该问题的数学模型? 这自然想到运用计算机模拟.计算机模拟就是在一定的假设下,利用数学运算模拟某个系统随时间的推进中系统的各种状态转移过程,当然这需要在计算机上完成模拟.因此建立数学模型,不一定只是涉及一个数学表达式,可以是某段时间的一个流程.

——作者

计算机科学技术的迅猛发展,给许多学科带来了巨大的影响.计算机不但使问题的求解变得更加方便、快捷和精确,而且使得解决实际问题的领域更加广泛.计算机适合于解决那些规模大、难以解析化以及具有不确定因素的数学模型.对某些实际问题人们不可能实地观察系统的运行行为,自然想到计算机的模拟.例如,考察某幢办公楼每天早晨上班高峰时期电梯服务系统,如何科学地评价电梯的服务质量? 我们可以随意提出一些合理化的建议,如两部电梯分别对奇、偶楼层服务,或上班高峰时期增加一部快速电梯的运行等.如果提出了三种方案,进一步问哪一种方案最优? 在一个时期内,实地检验各种方案的优劣,顾客肯定会产生厌烦情绪,这就必须借助计算机模拟该电梯在各种方案下的运行状况.又如大城市的交通道路车流量自动控制、各种运输网络设计、大、中型工厂的物流管理等问题,都需要使用计算机模拟技术.计算机模拟(computer simulation)是数学建模过程中较为重要的一类方法.

§8.1 海港系统的卸载货物问题

考虑一个中小规模的海港，拥有专门为货船卸载货物的设备. 假设在任何时刻只允许一艘船卸载货物，船仅为了卸载货物而停靠该港口，且假定连续两艘船先后到达港口的间隔时间范围是[15,145](单位:min)，并且是随机的. 每艘船需要的卸载货物时间依赖于船的型号和装载量，其卸载时间的变化区间为[45,90](单位:min). 提出如下问题:

1) 每艘船在港口的平均停留时间和最长停留时间是多少?

2) 定义一艘船的等待时间为船只到达港口时间到开始卸载货物时间，问每艘船的平均等待时间和最长等待时间是多少?

3) 试确定系统卸载设备的空闲率(或使用率).

§8.2 海港系统的卸载货物过程分析

海港系统的状态变量有船只在港停留时间、等待卸载货物时间、服务时间、船只数和卸载货物的设备数等. 系统有时间的状态转移过程. 针对海港系统的卸载货物问题，我们借助计算机模拟海港系统的实际运行状况. 首先假定:

1) 每艘船可能在任意时刻到达港口;

2) 连续两艘船到达港口的间隔时间服从区间[15,145]上的均匀分布;

3) 每艘船的卸载时间也服从[45,90]区间上的均匀分布;

4) 该港口只考虑卸载货物这一活动，不考虑其他活动;

5) 只考虑从零时刻起到最后一艘船离港的总的运行时间 T;

6) 每艘船卸完货物后立刻离开港口;

7) 在[0,T]时间范围内，考虑评价系统指标体系——卸载设备的使用率、货船的平均等待时间等.

为了讨论港口的实际卸载货物的过程，不妨假定某天有五艘船任意时刻到达该港口，根据历史记载，其数据如下:

船只	1	2	3	4	5
船只间隔到达时间	20	30	15	120	25
每艘船的卸载时间	55	45	60	75	80

根据以上数据，我们画一个简图，表示计算机模拟海港系统卸载货物的过程.

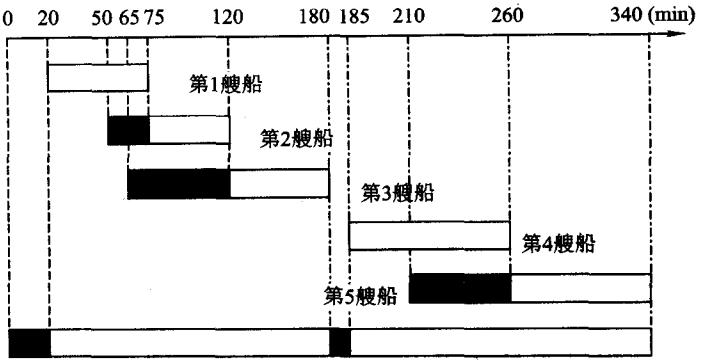


图 8.1　船只到达港口以及卸载货物过程的示意图

列表计算如下：

表 8.1　港口某天的历史数据

序号	到港时刻	开始服务时刻	排队长度	等待时间	服务时间	停留时间	设备空闲时间
1	20	20	0	0	55	55	20
2	50	75	1	25	45	70	0
3	65	120	2	55	60	115	0
4	185	185	0	0	75	75	5
5	210	260	1	50	80	130	0
合计				130	315	445	25
平均				26	63	89	

由此得出：$T = 340(\text{min})$，卸载设备的使用率为：$\frac{340-25}{340} = 92.65\%$；5 艘货船的平均等待时间为：$\frac{1}{5}(0+25+55+0+50) = 26(\text{min})$.

这是一个离散系统的模拟问题. 描述离散系统的一个重要概念是“事件”. 事件就是引起系统状态发生变化的行为，该系统的行为是由事件来驱动的. 如海港系统中，每艘船先后到达港口是一类事件，这时系统的状态——设备卸载货物（即服务）可能从闲变到忙，或排队的顾客人数发生变化. 一艘船接受服务完毕

后离开港口也定义为一类事件，这时系统状态可能由忙变成闲.

如何由计算机引起各类“事件”的发生？需要引入“随机数”概念以及蒙特卡罗(Monte Carlo)方法的基本思想.

§8.3 蒙特卡罗模拟思想

如何确定一艘潜水艇所受到的各种拉力？构造原型模型显然是不可能的，但可以构造一个仿真模型去模拟实际潜水艇的各种行为. 这种模拟实际模型行为的方法被称为 Monte Carlo 模拟，该方法的特点是它需要借助计算机来完成.

8.3.1 模拟确定性模型——任意曲边梯形面积的近似计算

本节主要介绍使用 Monte Carlo 模拟方法去模拟确定行为的模型——一条非负曲线与坐标轴所围成的面积.

假设曲线 $y=f(x)$，$a\leqslant x\leqslant b$，满足 $0\leqslant f(x)\leqslant M$.

如图 8.2 所示，在长为 $b-a$、宽为 M 的矩形内任选一点 $P(x,y)$，可以通过计算机产生随机数 x 和 y，满足 $a\leqslant x\leqslant b$，$0\leqslant y\leqslant M$，一旦点 P 选定，可以判断点 P 是在曲线 $y=f(x)$ 的上方还是下方？即 y 是否满足 $0\leqslant y\leqslant f(x)$？如果 y 满足上述不等式，则通过记数器记录这样的点. 程序设置两个计数器，一是记录满足 $0\leqslant y\leqslant f(x)$ 的点数，另一个计数器记录总的试验次数，则使用 Monte Carlo 模拟技术的近似计算公式：

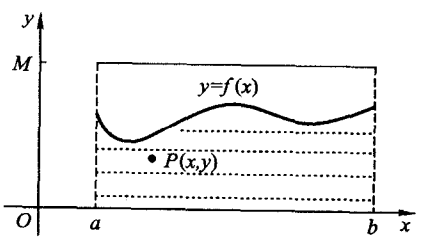


图 8.2 曲边梯形

$$\frac{\text{曲边面积}}{\text{矩形面积}}\approx\frac{\text{曲线 } y=f(x) \text{ 下方 } P \text{ 的点数}}{\text{发生在矩形内的总点数}},$$

即

$$\text{曲边面积}\approx\frac{\text{曲线 } y=f(x) \text{ 下方 } P \text{ 的点数}}{\text{发生在矩形内的总点数}}\times\text{矩形面积}.$$

Monte Carlo 模拟曲边梯形面积算法：

1) 赋初值：曲线下方点数 $m=0$；输入模拟所需总的试验次数 n.

2) 对 $i=1,2,\cdots,n$，执行以下三步：

Ⅰ. 产生随机数 x_i,y_i，满足 $a<x_i<b$，$0<y_i<M$；

Ⅱ. 计算 $f(x_i)$;

Ⅲ. 如果 $y_i \leqslant f(x_i)$,则 $m = m + 1$,否则,转Ⅱ.

3) 计算曲边面积的近似值:$S = M \times (b - a) \times m/n$.

例 8.1 用上述方法模拟曲边 $y = \sin x, 0 \leqslant x \leqslant \pi$ 与轴 $y = 0$ 所围成的面积,其中 $0 \leqslant \sin x < 2$.

用 MATLAB 编程如下:(mj. m)

```
n = N;
m = 0;
for i = 1:n
    x(i) = unifrnd(0,pi,1,1);
    y(i) = unifrnd(0,2,1,1);
    f = sin(x(i));
    if y(i) < = f
        m = m + 1;
    else
    end
end
S = 2 * pi * m/n
```

输入 N,执行 mj,可以得到如下结果:

表 8.2 计算机模拟曲边梯形面积数据

实验次数 N	曲边面积 S	实验次数 N	曲边面积 S
100	2.073 5	2 000	2.117 4
200	2.010 6	4 000	1.947 8
300	2.178 2	6 000	1.983 4
400	1.932 1	8 000	1.993 3
500	2.136 3	10 000	2.023 2
600	2.115 3	20 000	1.984 7
700	2.055 5	30 000	1.988 0
800	2.065 6	40 000	1.994 9
1 000	1.935 2	50 000	2.016 8

曲边 $y = \sin x, 0 \leqslant x \leqslant \pi$ 与轴 $y = 0$ 所围成的面积实际精确值为 2. 由表8.2可知,当实验次数 N 很大时,曲边面积的近似值越来越接近于精确值.

注意 即使实验次数 N 很大，曲边面积 S 的误差还是比较显著.

试一试 使用 Monte Carlo 模拟技术寻找曲面 $x^2+y^2+z^2\leqslant 1, x>0, y>0, z>0$ 所围成的体积.

8.3.2 模拟概率模型

我们在学习概率论时，肯定会学习概率的统计定义，即设有一个随机事件 A，如果事件 A 在 n 次重复试验中出现了 r 次，则称 $\frac{r}{n}$ 为事件 A 在 n 次试验中出现的频率. 显然，$0\leqslant\frac{r}{n}\leqslant 1$. 如果重复试验次数 n 不断增加，事件 A 的频率 $\frac{r}{n}$ 将会围绕某个数值 $p\in[0,1]$ 摆动，且随实验次数 n 增大，有 $\frac{r}{n}\approx p$，则定义事件 A 的概率为 $P(A)=p$，并且称 p 为事件 A 的统计概率.

例如，抛掷一枚均匀硬币的试验历史上曾有很多著名的数学家做过，实验表明均匀硬币的其中一面出现的频率随试验次数 n 的增加将接近 0.5. 能否用 Monte Carlo 模拟方法与计算机结合模拟该试验？

定义一个函数：

$$f(x)=\begin{cases}\text{正面}, & 0\leqslant x\leqslant 0.5,\\ \text{反面}, & 0.5<x\leqslant 1.\end{cases}\tag{8.1}$$

实际上，该函数 $f(x)$ 表明均匀硬币由区间 $[0,1]$ 上的数字产生正、反面的结果，设置试验次数 n，事件 $A=$“硬币正面”，则事件 A 的频率：

$$\frac{\text{事件 } A \text{ 出现的次数 } r}{n}.$$

根据公式(8.1)，使用 Monte Carlo 模拟抛掷均匀硬币试验，其 MATLAB 程序如下：(gailv.m)

输入总的试验次数 n；

```
N=n;
r=0;
for i=1:N
    x=rand;
```

```
    if 0 < =x&x < =0.5
        r =r+1;
    else
    end
end
r
PA =r/N
```

对各种试验次数 n,运行上述程序,得到如下结果:

表 8.3 计算机模拟抛掷均匀硬币试验数据

试验次数 n	正面出现次数 r	频率 PA
100	49	0.49
200	108	0.54
500	238	0.476
1 000	498	0.498
5 000	2 475	0.495
10 000	4 991	0.499
50 000	24 999	0.50

结果表明:随试验次数 n 增大,显示 $\frac{r}{n}\approx 0.5$,即频率接近事件的概率.

使用 Monte Carlo 模拟技术,模拟掷一枚均匀骰子的试验.

8.3.3 随机数的产生

在上两节,我们知道 Monte Carlo 模拟算法的关键步骤需要产生随机数(random number). 随机数有各种应用,如赌博问题、大系统优化问题、道路交通控制问题等模拟都需要使用随机数.

计算机上产生的随机数是按照确定的算法产生的,它遵循一定的规律,显然不是真正随机的,因此我们将这种随机数叫做伪随机数(pseudorandom number). 只要伪随机数能通过一系列的统计检验,就可以把它们当作真正的随机数放心地使用,而不会引起太大的误差.

产生均匀分布的伪随机数的常用方法有平方取中法、线性同余法和广义同余法等. 由于目前计算机上常用的高级语言(如 C、Pascal、Fortran 等)都有产生均匀分布随机数的系统函数,我们可以直接使用而不必关心其实现原理,在此不作专门介绍. 首先介绍 MATLAB 软件中产生区间[0,1]上的均匀分布随机数的系统函数 $r=\text{rand}(n)$,它产生 $n\times n$ 阶均匀随机矩阵,一般地 $r=\text{rand}(m,n)$ 产生 $m\times n$ 阶均匀随机矩阵,特别地,$r=\text{rand}$ 将产生一个随机数.

下面简单介绍如何由均匀随机数以及某种算法产生其他分布随机数的方法. 用 r_1、r_2、…表示独立同分布于 $U[0,1]$ 的随机数列.

1. 直接抽样法

设连续型随机变量 X 有分布函数

$$F(x)=\int_{-\infty}^{x} f(t)\,\mathrm{d}t,\quad \text{其中 } f(t) \text{ 是 } X \text{ 的密度函数.}$$

因为 $0<F(x)<1$,令 $r=F(x)$,若函数 $F(x)$ 的反函数存在,那么对随机变量 X 的抽样可由公式 $x=F^{-1}(r)$ 产生.

例 8.2 设 X 服从参数为 λ 的指数分布. 其密度函数为

$$f(x)=\begin{cases}\lambda \mathrm{e}^{-\lambda x}, x\geqslant 0,\\ 0, x<0.\end{cases}$$

解 首先求出 X 的分布函数:
$$F(x)=\begin{cases}1-\mathrm{e}^{-\lambda x}, x\geqslant 0,\\ 0, x<0.\end{cases}$$

当 $x\geqslant 0$ 时,令 $r=1-\mathrm{e}^{-\lambda x}$,可解出:$x=-\dfrac{1}{\lambda}\ln(1-r)$ $(r\neq 0)$.

在 MATLAB 下输入:

```
r = rand(1,5);
x = -(1/3) * log(1 - r)
```

输出结果为:

```
x = 0.0171   0.4882   0.1665   0.2405   0.0384
```

它们遵从参数为 3 的指数分布,称它们是服从指数分布(3)的随机数.

对于一些具有特殊分布(如两点分布、二项分布)的随机变量还可采用特殊的方法来处理. 按如下步骤实现:

1) 置 $k=0$;$i=1$;

2) 产生随机数 r;

3) 判断 $r<p$;若是,则令 $k=k+1$;否则,转 4);

4) 判断 $i<N$;若是,则令 $i=i+1$,转 2);否则,输出样本 $X=k$.

做 使用 MATLAB 语言编程实现上述步骤.

例 8.3 一般离散分布随机数的产生:设 X 服从如下离散分布:

表 8.4 X 服从的离散分布

x	2	4	6	8
$P(x)$	0.1	0.4	0.3	0.2

构造累积分布函数如下:

$$F(x)=\begin{cases}0, & x\leqslant 2,\\ 0.1, & 2<x\leqslant 4,\\ 0.5, & 4<x\leqslant 6,\\ 0.8, & 6<x\leqslant 8,\\ 1, & 8<x.\end{cases}$$

根据反函数的定义知,设随机数 $r\in[0,1]$,对应的转换关系如下表所示:

表 8.5 r 与 x 的对应关系

r	[0,0.1)	[0.1,0.5)	[0.5,0.8)	[0.8,1]
x	2	4	6	8

MATLAB 程序:(lisanrnd.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  r = rand(10); y = [];
  for i = 1:10
    if 0 < = r(i)&r(i) < 0.1
      y(i) = 2;
    elseif 0.1 < = r(i)&r(i) < 0.5
      y(i) = 4;
    elseif 0.5 < = r(i)&r(i) < 0.8
      y(i) = 6;
    else
      y(i) = 8;
    end
  end
```

```
y    % 产生随机变量数组
```

输出结果:

```
y =8    4    6    4    8    6    4    2    8    4
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

直接抽样法的特点是直观、方便. 但有时函数 $F(x)$ 的反函数不存在时,就不能使用这种方法,例如正态分布.

2. 近似抽样法

近似抽样方法的步骤是设一组独立同分布的随机变量 $r_1, r_2, \cdots, r_n, \cdots$,利用中心极限定理得

$$\frac{\frac{1}{n}\sum_{i=1}^{n} r_i - Er_i}{\sqrt{Dr_i/n}} \sim N(0,1), \text{当 } n \to \infty \text{ 时}.$$

设 $r_1, r_2, \cdots$ 为 $[0,1]$ 区间上的均匀随机数列,则 $Er_i = 1/2, Dr_i = 1/12$,从而有:当 n 充分大时,

$$\frac{\frac{1}{n}\sum_{i=1}^{n} r_i - \frac{1}{2}}{\sqrt{\frac{1}{12n}}} \sim N(0,1).$$

常用的是 $n = 12$ 的情形,令 $u_{12} = \dfrac{\frac{1}{n}\sum_{i=1}^{n} r_i - \frac{1}{2}}{\sqrt{\frac{1}{12n}}}$,化简得 $u_{12} = \sum_{i=1}^{12} r_i - 6$,其中 u_{12} 是标准正态分布的随机数,即由 12 个区间 $[0,1]$ 上均匀随机数产生一个标准正态随机数. 如果还要产生一般正态分布的随机数,则只需用变换 $z = \sigma u + a$.

参考程序如下:

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
x =[ ];
for i =1:8
n =12;
r =rand(1,n);
x(i) =sum(r) -6;
end
x
```

计算结果:

```
x =   0.6229    -1.9478    0.1422    -1.0502    0.4634    -1.4441
      -1.1871   -2.0751
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

另外，还有一种利用二维正态变换可以产生标准正态分布的随机数，其变换如下：

$$\begin{cases} x = \sqrt{-2\ln r_1}\cos 2\pi r_2, \\ y = \sqrt{-2\ln r_1}\sin 2\pi r_2, \end{cases} \text{当 } r_1 \neq 0 \text{ 时.}$$

3. MATLAB 中各种常见分布下产生随机数的命令

在 MATLAB 软件中，可以直接产生满足各种分布的随机数，便于编程实现.

表 8.6　MATLAB 中产生几种常见分布下的随机数的语句

常见的分布函数	MATLAB 语句
均匀分布 $U[0,1]$	r = rand(m,n)
均匀分布 $U[a,b]$	r = unifrnd(a,b,m,n)
指数分布　$\Gamma(1,\lambda)$	r = exprnd(λ,m,n)
正态分布 $N(\mu,\sigma)$	r = normrnd(mu,sigma,m,n)
二项分布 $B(n,p)$	r = binornd(n,p,m,n1)
泊松分布 $P(\lambda)$	r = poissrnd(λ,m,n)

注：以上语句均产生 $m \times n$ 的矩阵.

为了熟悉 MATLAB 语句，请使用以上命令进行练习.

§8.4　海港系统卸载货物的模拟

前面已经分析了海港系统卸载货物属于离散系统的模拟问题.

离散系统（discrete system）是指系统状态只在有限的时间点或可数的时间点上有随机事件驱动的系统. 例如排队系统（queue system），显然状态量的变化只是在离散的随机时间点上发生. 假设离散系统状态的变化是在一个时间点上瞬间完成的.

为了模拟离散系统，必须设置一个**模拟时钟**（simulate clock），它将时间从一个时刻向另一个时刻进行推进，并且可随时反映系统时间的当前值. 其中，模拟时间推进方式有两种——**下次事件推进法和均匀间隔时间推进法**，常用的是下次事件推进法. 其过程是：置模拟时钟的初值为 0. 跳到第一个事件发生的时刻，计算系统的状态，产生未来事件并加入到队列中去；跳到下一事件，计算系统状

态,……,重复这一过程直到满足某个终止条件为止.

海港系统卸载货物的模型一般用流程框图来描述,然后编制程序模拟在一定时间范围内或一定数量船只的系统运行的活动过程.

符号说明:

$between_i$:第 $i-1$ 艘船与第 i 艘船先后到达港口的间隔时间(服从均匀分布 $U[15,145]$,单位:min).

$arrive_i$:第 i 艘船到达港口的时间.

$unload_i$:第 i 艘船卸载货物所需时间(服从均匀分布 $U[45,90]$,单位:min).

$start_i$:第 i 艘船开始卸载时间.

$idle_i$:第 i 艘船开始卸载之前港口设备的空闲时间.

$wait_i$:第 i 艘船到达港口与开始卸载货物之间的等待时间.

$finish_i$:第 i 艘船卸载货物完成时刻.

$harbor_i$:第 i 艘船在港口停留时间.

需要输出系统指标:

$Hartime$:每艘船停留港口的平均时间.

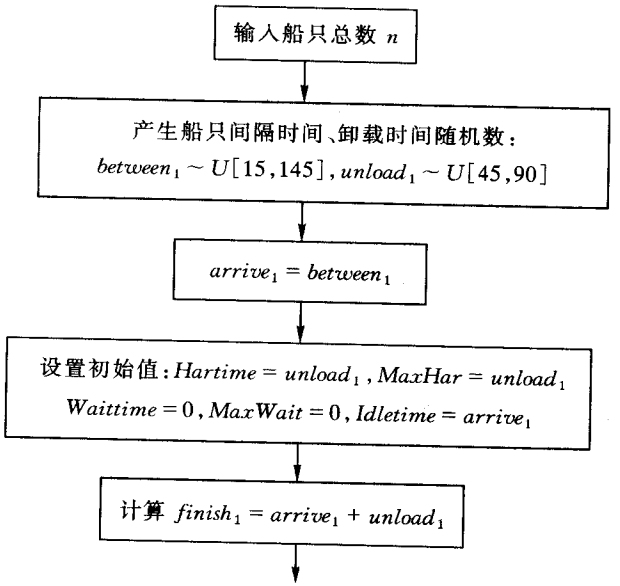
$MaxHar$:船只停留港口的最长时间.

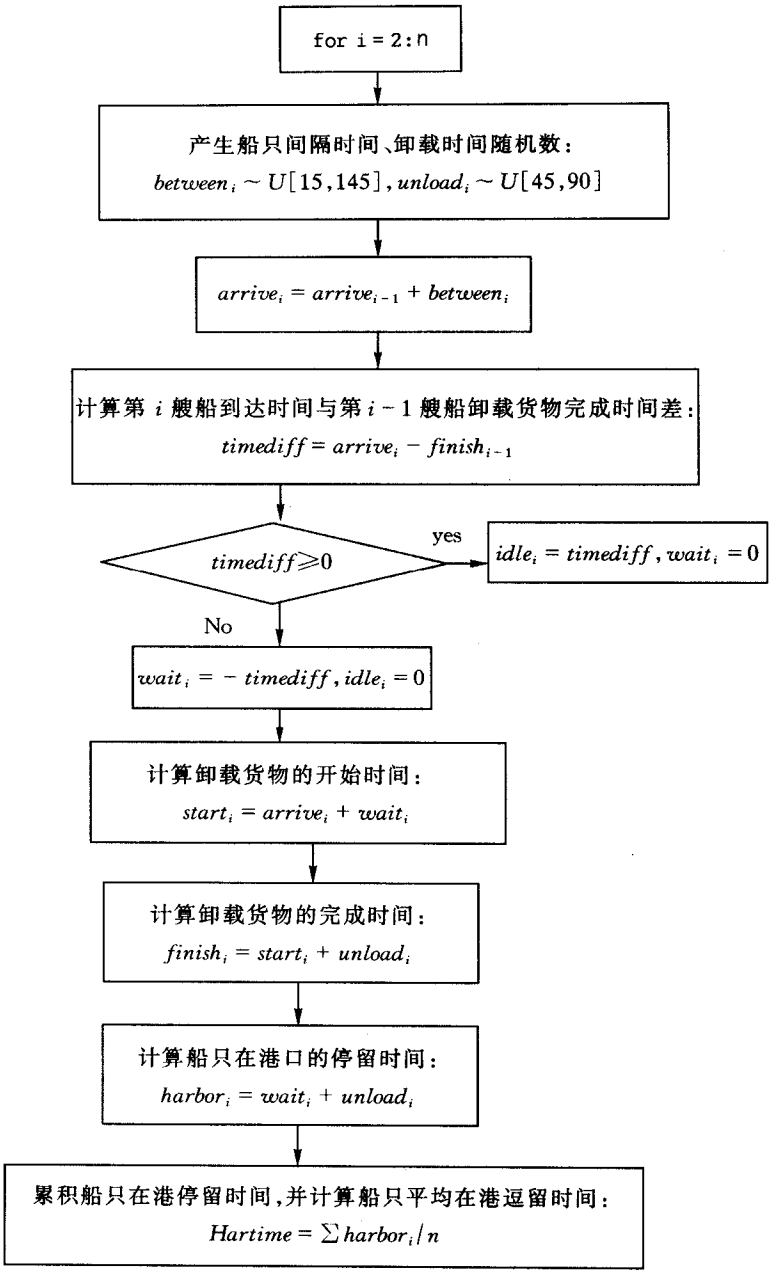
$Waittime$:每艘船只平均等待时间.

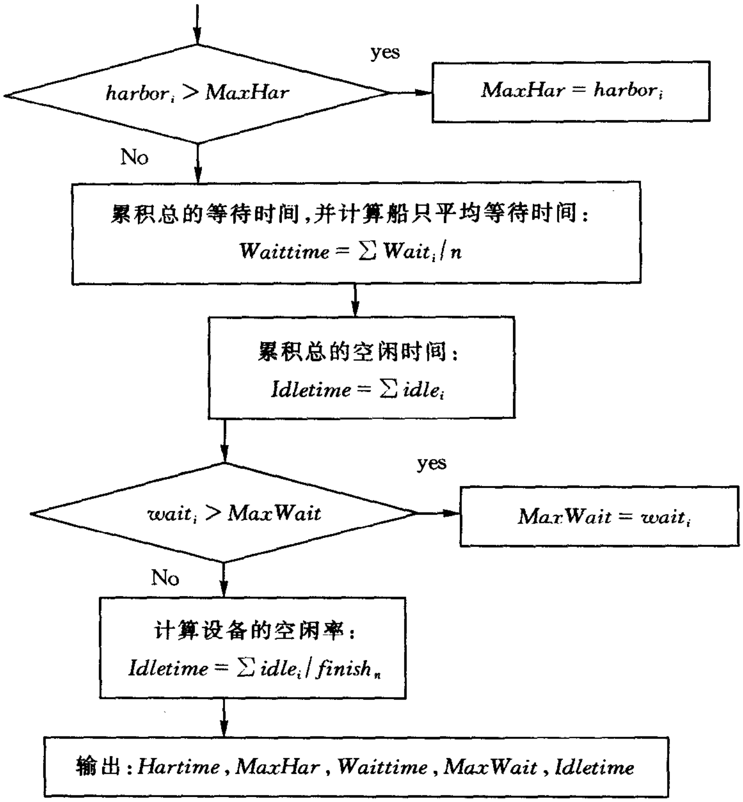
$MaxWait$:船只最长等待时间.

$Idletime$:卸载设备总的空闲率.

流程框图如下:




for i = 2:n
产生船只间隔时间、卸载时间随机数：
$between_i \sim U[15,145]$, $unload_i \sim U[45,90]$
$arrive_i = arrive_{i-1} + between_i$
计算第 i 艘船到达时间与第 $i-1$ 艘船卸载货物完成时间差：
$timediff = arrive_i - finish_{i-1}$
$timediff \geqslant 0$
yes
$idle_i = timediff$, $wait_i = 0$
No
$wait_i = -timediff$, $idle_i = 0$
计算卸载货物的开始时间：
$start_i = arrive_i + wait_i$
计算卸载货物的完成时间：
$finish_i = start_i + unload_i$
计算船只在港口的停留时间：
$harbor_i = wait_i + unload_i$
累积船只在港停留时间，并计算船只平均在港逗留时间：
$Hartime = \sum harbor_i / n$



根据上述框图,使用 MATLAB 语言编程如下:(czxz. m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   n = N;wait = [ ];idle = [ ];harbor = [ ];
   between(1) = unifrnd(15,145,1,1);
   unload(1) = unifrnd(45,90,1,1);
   arrive(1) = between(1);
   Hartime = unload(1);MaxHar = unload(1);
   Waittime = 0;Max Wait = 0;Idletime = arrive(1);
   finish(1) = arrive(1) + unload(1);
   for i = 2:n
       between(i) = unifrnd(15,145,1,1);
       unload(i) = unifrnd(45,90,1,1);
       arrive(i) = arrive(i -1) + between(i);
       timediff = arrive(i) - finish(i -1);
       if timediff > =0
```

```
            idle(i) = timediff;wait(i) = 0;
        else
            idle(i) = 0;wait(i) = -timediff;
        end
        start(i) = arrive(i) + wait(i);
        finish(i) = start(i) + unload(i);
        harbor(i) = wait(i) + unload(i);
        if harbor(i) > MaxHar
            MaxHar = harbor(i);
        else
        end
        if wait(i) > Max Wait
            Max Wait = wait(i);
        else
        end
    end
    F = finish(n);
    Hartime = sum(harbor)/n
    Waittime = sum(wait)/n
    Idletime = sum(idle)/F
    MaxHar,Max Wait,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

当输入船只数 $N=100$，根据上述程序模拟 8 次，得到如下结果：

表 8.7　程序模拟结果

实验次数	*Hartime*	*MaxHar*	*Waittime*	*MaxWait*	*Idletime*
1	111	254	43	168	0.12
2	97	238	30	171	0.16
3	93	194	29	142	0.17
4	88	202	22	147	0.22
5	89	202	22	119	0.22
6	96	295	29	232	0.19
7	110	284	44	204	0.16
8	97	217	29	143	0.16

注：表中数据按四舍五入取整，单位：min.

如果港口的卸载机器更换了许多新的设备或增加了设备的数量，由此为每艘船卸载货物的时间提速了，其服务时间由原来的 45 ~ 90 min 减到 35 ~ 75 min，

这时货船的平均等待时间减少，设备的空闲率增加. 计算结果见下表 8.8.

表 8.8　设备改进后计算机程序模拟结果($N=100$)

实验次数	*Hartime*	*MaxHar*	*Waittime*	*MaxWait*	*Idletime*
1	63	114	9	57	0.30
2	59	103	4	45	0.36
3	68	139	13	92	0.27
4	69	182	15	119	0.30
5	69	158	14	88	0.33
6	66	141	11	83	0.30
7	65	121	12	82	0.33
8	63	151	10	102	0.37

注：表中数据按四舍五入取整，单位：min.

1）如果船只到达港口的密度增加，即相邻两只船到达港口的间隔时间由原来的 15 ~ 145 min 变为 10 ~ 90 min，其他假设条件不变，你将得到什么结果？如何解释？

2）如果相邻两艘船到达港口的间隔时间不是服从均匀分布 $U[15,145]$，而是平均间隔时间为 60 min 的指数分布，你又会得到什么结论？

如果相邻两艘船到达港口的间隔时间以及在港设备卸载货物的服务时间不服从 15 ~ 145 和 45 ~ 90 min 的均匀分布. 为了更好地模拟实际海港系统，需要收集该系统的历史数据，比如人们对到达港口的 1 200 艘船只进行了调查，得到如下数据：

表 8.9　港口卸载货物的基本数据

间隔时间/min	频率	卸载时间/min	频率
15 ~ 24	0.009	45 ~ 49	0.017
25 ~ 34	0.029	50 ~ 54	0.045
35 ~ 44	0.035	55 ~ 59	0.095
45 ~ 54	0.051	60 ~ 64	0.086
55 ~ 64	0.090	65 ~ 69	0.130

续表

间隔时间/min	频率	卸载时间/min	频率
65 ~ 74	0.161	70 ~ 74	0.185
75 ~ 84	0.200	75 ~ 79	0.208
85 ~ 94	0.172	80 ~ 84	0.143
95 ~ 104	0.125	85 ~ 90	0.091
105 ~ 114	0.071		
115 ~ 124	0.037		
125 ~ 134	0.017		
135 ~ 145	0.003		

根据表 8.9 可以画累积频率直方图如下：

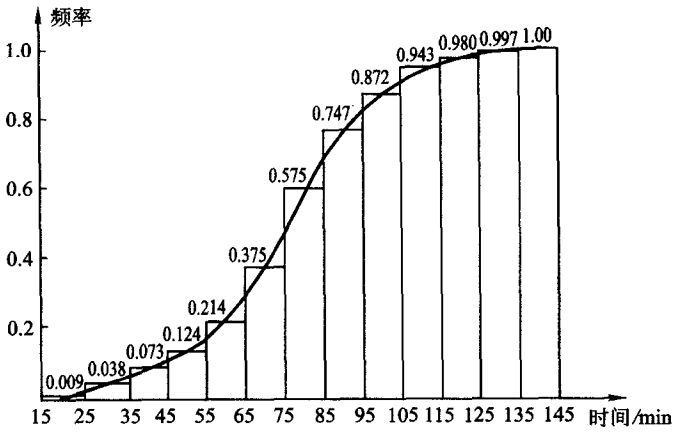


图 8.3　相邻两艘船只到达港口的间隔时间

根据图 8.3 和图 8.4，可以编程产生随机数. 首先，利用累积直方图，作分段线性插值折线. 例如在图 8.3 中，区间[15,25]和[25,35]的中点分别是 20 和 30，在直角坐标(时间，频率)下，用直线连接平面点(20,0.009)，(30,0.038)，可以得到直线方程：$x = 344.8y + 16.896\ 6$，$0.009 \leqslant y < 0.038$，当 y 取[0.009,0.038)上的均匀随机数时，通过线性方程即可产生间隔时间的随机数. 关于分段线性插值折线的解析表达式如下表 8.10 和表 8.11.

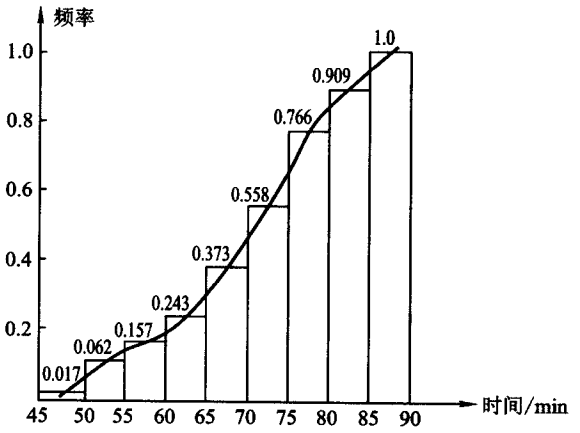


图 8.4　设备卸载货物时间

表 8.10　相邻两艘船只到达港口的间隔时间关于分段线性插值折线的解析表达式

产生均匀随机数的区间	对应时间范围	线性插值函数
$0 \leqslant y < 0.009$	$15 \leqslant x < 20$	$x = 555.6y + 15.000\ 0$
$0.009 \leqslant y < 0.038$	$20 \leqslant x < 30$	$x = 344.8y + 16.896\ 6$
$0.038 \leqslant y < 0.073$	$30 \leqslant x < 40$	$x = 285.7y + 19.142\ 9$
$0.073 \leqslant y < 0.124$	$40 \leqslant x < 50$	$x = 196.1y + 25.686\ 3$
$0.124 \leqslant y < 0.214$	$50 \leqslant x < 60$	$x = 111.1y + 36.222\ 2$
$0.214 \leqslant y < 0.375$	$60 \leqslant x < 70$	$x = 62.1y + 46.708\ 0$
$0.375 \leqslant y < 0.575$	$70 \leqslant x < 80$	$x = 50.0y + 51.25$
$0.575 \leqslant y < 0.747$	$80 \leqslant x < 90$	$x = 58.1y + 46.569\ 8$
$0.747 \leqslant y < 0.872$	$90 \leqslant x < 100$	$x = 80.0y + 30.240\ 0$
$0.872 \leqslant y < 0.943$	$100 \leqslant x < 110$	$x = 140.8y - 22.816\ 9$
$0.943 \leqslant y < 0.980$	$110 \leqslant x < 120$	$x = 270.3y - 144.864\ 9$
$0.980 \leqslant y < 0.997$	$120 \leqslant x < 130$	$x = 588.2y - 456.47$
$0.997 \leqslant y \leqslant 1.00$	$130 \leqslant x \leqslant 145$	$x = 5\ 000y - 4\ 855$

表 8.11　设备卸载货物时间关于分段线性插值折线的解析表达式

产生均匀随机数的区间	对应时间范围	线性插值函数
$0 \leq y < 0.017$	$45 \leq x < 47.5$	$x = 147y + 45.000$
$0.017 \leq y < 0.062$	$47.5 \leq x < 52.5$	$x = 111y + 45.611$
$0.062 \leq y < 0.157$	$52.5 \leq x < 57.5$	$x = 53y + 49.237$
$0.157 \leq y < 0.243$	$57.5 \leq x < 62.5$	$x = 58y + 48.372$
$0.243 \leq y < 0.373$	$62.5 \leq x < 67.5$	$x = 38.46y + 53.154$
$0.373 \leq y < 0.558$	$67.5 \leq x < 72.5$	$x = 27y + 57.419$
$0.558 \leq y < 0.766$	$72.5 \leq x < 77.5$	$x = 24y + 59.087$
$0.766 \leq y < 0.909$	$77.5 \leq x < 82.5$	$x = 35y + 50.717$
$0.909 \leq y < 1.0$	$82.5 \leq x < 90$	$x = 82.41y + 7.582$

根据表8.10和8.11可以产生间隔时间和服务时间的随机数，使用MATLAB编程如下：(jge.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    a = rand;
    if 0 < = a&a < 0.009
        b = 555.6 * a + 15;
    elseif 0.009 < = a&a < 0.038
        b = 334.8 * a + 16.8966;
    elseif 0.038 < = a&a < 0.073
        b = 285.7 * a + 19.1429;
    elseif 0.073 < = a&a < 0.124
        b = 196.1 * a + 25.6863;
    elseif 0.124 < = a&a < 0.214
        b = 111.1 * a + 36.2222;
    elseif 0.214 < = a&a < 0.375
        b = 62.1 * a + 46.7080;
    elseif 0.375 < = a&a < 0.575
        b = 50.0 * a + 51.25;
    elseif 0.575 < = a&a < 0.747
        b = 58.1 * a + 46.5698;
    elseif 0.747 < = a&a < 0.872
        b = 80.0 * a + 30.24;
    elseif 0.872 < = a&a < 0.943
```

```
        b=140.8*a-22.8169;
    elseif 0.943<=a&a<0.980
        b=270.3*a-144.8649;
    elseif 0.980<=a&a<0.997
        b=588.2*a-456.47;
    else 0.997<=a&a<=1
        b=5000*a-4855;
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

同理,产生服务时间随机数,其程序 fwu. m. 模拟港口运行情况的主程序修改为:(czxzl. m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    n=N;wait=[];idle=[];harbor=[];
    jge;
    fwu;
    between(1)=b;
    unload(1)=c;
    arrive(1)=between(1);
    Hartime=unload(1);MaxHar=unload(1);
    Waittime=0;Max Wait=0;Idletime=arrive(1);
    finish(1)=arrive(1)+unload(1);
    for i=2:n
        jge;
        fwu;
        between(i)=b;
        unload(i)=c;
        arrive(i)=arrive(i-1)+between(i);
        timediff=arrive(i)-finish(i-1);
        if timediff>=0
            idle(i)=timediff;wait(i)=0;
        else
            idle(i)=0;wait(i)=-timediff;
        end
        start(i)=arrive(i)+wait(i);
        finish(i)=start(i)+unload(i);
        harbor(i)=wait(i)+unload(i);
        if harbor(i)>MaxHar
            MaxHar=harbor(i);
```

```
        else
        end
        if wait(i) >Max Wait
            Max Wait =wait(i);
        else
        end
    end
    F =finish(n);
    Hartime =sum(harbor)/n
    Waittime =sum(wait)/n
    Idletime =sum(idle)/F
    MaxHar,Max Wait,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

对 $N=100$,运行结果如表 8.12.

表 8.12 模拟海港系统的计算机运行结果($N=100$)

实验次数	*Hartime*	*MaxHar*	*Waittime*	*MaxWait*	*Idletime*
1	108	191	38	112	0.05
2	105	271	36	192	0.07
3	102	229	32	159	0.09
4	88	163	19	109	0.09
5	87	145	20	91	0.11
6	93	174	24	94	0.11

注:表中数据按四舍五入取整,单位:min.

1) 如果港口可以同时卸载多艘船只的货物,将是一个多窗口的服务问题,如何进行模拟?

2) 如果海港系统不仅仅考虑卸载货物问题,如还考虑装载货物问题,或卸载货物以后再装载等问题,又应该如何分析?

3) 何谓连续系统? 如何进行连续系统模拟?

§8.5 连续系统的计算机模拟

状态随着时间连续变化的系统，称为连续系统(continuous system). 对连续系统的计算机模拟是近似地获取系统状态在一些离散时刻点上的数值，在一定假设条件下，利用数学运算模拟系统的运行过程. 连续系统模型一般是微分方程，它在数值模拟中最基本的算法是数值积分算法. 例如有一系统可用微分方程来描述

$$\frac{dy}{dt}=f(t,y).$$

已知输出量 y 的初始条件，$y(t_0)=y_0$，现在要求出输出量 y 随时间变化的过程 $y(t)$. 最直观的想法是：首先将时间离散化，令 $h_k=t_{k+1}-t_k$，称为第 k 步的计算步距(一般是等间距的)，然后按以下算法计算状态变量 $y(t)$ 在各时刻 t_{k+1} 上的近似值

$$y(t_{k+1})\approx y_{k+1}=y_k+f(t_k,y_k)(t_{k+1}-t_k),$$

其中初始点 (t_0,y_0)，$k=1,2,\cdots$. 按照这种作法即可求出整个 $y(t)$ 的曲线. 这种最简单的数值积分算法称为欧拉法. 除此之外，还有其他一些算法.

因此，连续系统模拟方法是：首先确定系统的连续状态变量，然后将它在时间上进行离散化处理，并由此模拟系统的运行状态.

例如，导弹跟踪问题，某军一导弹基地发现正北方向 120 km 处海面上有敌艇一艘以 90 km/h 的速度向正东方向行驶. 该基地立即发射导弹跟踪追击敌艇，导弹速率为 450 km/h，自动导航系统使导弹在任一时刻都能对准敌艇.

1) 试问导弹在何时何处击中敌艇？

2) 如果当基地发射导弹的同时，敌艇立即由仪器发觉. 假定敌艇为高速快艇，它即刻以 135 km/h 的速度向与导弹方向垂直的方向逃逸，问导弹何时何地击中敌艇？

微分方程建模：

$$\begin{cases}\dfrac{dy}{dx}=\tan\alpha=\dfrac{120-y}{90t-x},\\[2ex]\sqrt{\left(\dfrac{dy}{dt}\right)^2+\left(\dfrac{dx}{dt}\right)^2}=450,\end{cases}$$

其中，$(x(t),y(t))$ 为随时间 t 连续变化的状态变量(平面坐标，即运动轨迹). 使用计算机模拟，假设将时间离散化，时间步长为 τ，并且，$x(t_k)\approx x_k$，$y(t_k)\approx y_k$，$k=1,2,\cdots$，其状态转移方程为：

$$\begin{cases} x_{k+1} = x_k + 450\tau \times \dfrac{90k\tau - x_k}{\sqrt{(90k\tau - x_k)^2 + (120 - y_k)^2}}, \\ y_{k+1} = y_k + 450\tau \times \dfrac{120 - y_k}{\sqrt{(90k\tau - x_k)^2 + (120 - y_k)^2}}. \end{cases}$$

MATLAB 程序如下:(simul.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    function [num,y_j,L,T] = simul(x,y,t,eps)
    k = 0;
    while k < 1 000
      p = 90 * k * t - x;
      q = 120 - y;
      d1 = p/(p^2 + q^2)^0.5;
      d2 = q/(p^2 + q^2)^0.5;
      x = x + 450 * t * d1;
      y = y + 450 * t * d2;
      if (abs(q) < eps)
        break;
      end
        k = k + 1;
    end
    num = k;
    L = x;
    y_j = y;
    T = L/90;
    运行结果:
    Num = 2;y_j = 131.2155;
    L = 25.2645(km);T = 0.2807(h)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

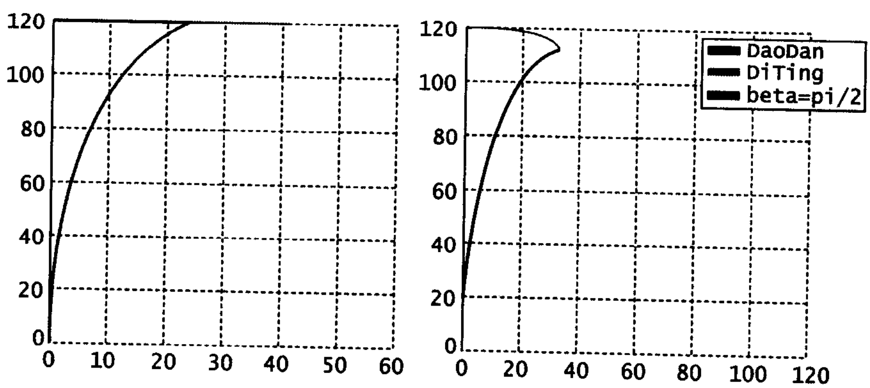
事实上,精确结果为: L = 25(km);T = 0.2778(h).

另外,还可以使用图形结果直观动态地表现追逐过程,MATLAB 程序如下:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x = [0,0];                    (huatu1.m)
    axis('equal');
    axis([0,140,0,140]);grid on
    hold on
    t = 0.001;% 步长
```

```
while (abs(x(2) -120) >0.1)   % 终止条件( < )
  for k =1:280
  p =90 * k * t - x(1);
  q =120 - x(2);
  d1 =p/(p^2 +q^2)^0.5;
  d2 =q/(p^2 +q^2)^0.5;
  x(1) =x(1) +450 * t * d1;
  x(2) =x(2) +450 * t * d2;
  x1(1) =90 * k * t;
  x1(2) =120;
  h1 =line('color',[0,0.2,0.4],'linewidth',2);
  h2 =line('color',[0,0.6,0.9],'linewidth',3);
  set(h1,'xdata', x1(1),'ydata',x1(2));
  set(h2,'xdata', x(1),'ydata',x(2));
  end
end  hold on
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

运行结果如下：



(a) (b)

图 8.5 导弹与敌艇追逐路线图

图 8.5(b)是问题 2)的解，只需程序稍稍修改即可.

§8.6 操　　练

操练一　设备的可靠性问题

一设备上配有三个相同的轴承，每个轴承正常工作寿命为随机变量，其概率分布如下：

表 8.13　轴承工作寿命

寿命/h	1 000	1 100	1 200	1 300	1 400	1 500	1 600	1 700	1 800	1 900
概　率	0.10	0.13	0.25	0.13	0.09	0.12	0.02	0.06	0.05	0.05

假定任何一个轴承损坏都可以使设备停止工作. 从有轴承损坏、设备停止工作到检修工到达、开始更换部件为止，我们称它为一个延迟时间. 假定延迟时间也是随机变量，其概率分布如下表：

表 8.14　延 迟 时 间

延迟时间/min	5	10	15
概　率	0.6	0.3	0.1

假定设备停工时会产生一定的损失费用，主要有三项损失费：单位时间损失费 5 元/分，检修工的工时费 12 元/小时，轴承的成本费 16 元/个. 另外，更换一个轴承所需时间是 20 min，同时更换两个轴承所需时间是 30 min，同时更换三个轴承所需时间是 40 min.

现在，设想有两种方案：一是损坏一个部件更换一个部件，另一个是一旦有轴承损坏就全部更换. 试通过计算机模拟对提出的两种方案做出合理的评价.

操练二　订货策略问题

在物资的供应过程中，由于到货与销售不可能做到同步、同量，故总要保持一定的库存储备. 如果库存过多，就会造成积压浪费以及保管费用的上升；如果库存过少，会造成缺货. 如何选择库存和订货策略，就是一个需要研究的问题. 现要研究以下问题：

某自行车商店的仓库管理人员采取一种简单的订货策略，当库存降低到 P 辆自行车时就向厂家订货 Q 辆，如果某一天的需求量超过了库存量，商店就有销售损失和信誉损失，但如果库存量过多，将会导致资金积压和保管费增加. 若现在已有如表 8.15 中的五种库存策略，试比较、选择一种策略以使花费最少. 已知该问题的条件：

1）从发出订货到收到货物需隔3天；

2）每辆自行车保管费为0.75元/天，每辆自行车的缺货损失为1.80元/天，每次的订货费为75元；

3）每天自行车的需求量服从0到99之间的均匀分布；

4）原始库存为115辆，并假设第一天没有发出订货.

表8.15　某自行车商店的五种库存策略

方案编号	1	2	3	4	5
重新订货量 P 辆	125	125	150	175	175
重新订货量 Q 辆	150	250	250	250	300

更多的相关信息资源

1　Frank R. Giordano, Maurice D. Weir and William P. Fox. *A First Course in Mathematical Modeling*. 影印版. 北京：机械工业出版社，Cole, a division of Thomson Learning, Inc.，2003

2　周义仓，赫孝良. 数学建模实验. 西安：西安交通大学出版社，2000

3　熊光楞，肖田元，张燕云. 连续系统仿真与离散事件系统仿真. 北京：清华大学出版社，1999

4　傅鹂，龚劬，刘琼荪，何中市. 数学实验. 北京：科学出版社，2000

第9章

在简约的世界里使收益最大

——线性规划

长期以来，人们总是追求一种有秩序的简约的境界，那就是清晰、确定、简单，尽管这些后来都被一一无情打破. 一个清晰、确定、简单的世界是一个易测易控的世界. 确实，在这种世界里，人们如鱼得水，征服一切，有效地使他们的利益最大化. 具有这种特性的极端就是一种确定性的线性世界，一种最易于优化的世界，让人感觉特别美好：在人们充分意识到非线性的奥妙之前.

——作者

§9.1 华尔街公司的投资选择

美国华尔街的金融投资商们绝不会把钱存在银行里，更不会揣在兜儿里，他们的使命是“钱生钱”，而且要求尽可能多尽可能快. 他们要投资. 怎么进行投资最好？华尔街的金融投资商们很精明，由于一个古老的说法，“不要把所有的鸡蛋放在一个篮子里”，他们通常考虑“组合投资”. 怎样组合投资？下面给出我们的例子.

组合投资问题 桫椤树公司考虑投资四种债券，可用于投资的资金为1 000万美元. 每种债券的年收益率期望值、最低值以及每种债券的持续期（债券的持续期是其对利率的敏感性的度量，持续期越长，受利率影响越大）由下表给定：

表9.1 四种债券的情况

债券	年收益率期望值/%	最低年收益率值/%	持续期/年
债券1	13	6	3
债券2	8	8	4
债券3	12	10	7
债券4	14	9	9

桤椤树公司希望其债券组合投资的年收益率期望值达到最大，并且满足下列要求：

1）债券组合投资的年收益率最低值至少为8%；

2）债券组合投资的平均持续期不超过6年（组合债券的平均持续期=各债券按投资比例的加权平均）；

3）根据分散投资原理（“不要把所有的鸡蛋放在一个篮子里”），每种债券的投资额度不能超过总投资额度的40%.

那么，每种债券应该投资多少呢？

§9.2 组合投资决策

面对每种债券应该投资多少这个问题，如果你就是桤椤树公司的决策者，你从何处入手？

解决任何问题，第一步都是要把问题理解和描述清楚.“每种债券应该投资多少”这个问题本身已经隐含了该问题描述的第一个方面：你要做的决策就是给每种债券分配一个投资额.所谓决策. 可以说就是选择，往往是对多个因素的选择.比如，在桤椤树公司的这个组合投资问题中，每种债券的投资额就是一个需要选择的因素.决策问题描述的第一个方面就是决策的各个构成因素.

桤椤树公司用于这次投资的总额为1 000万美元，同时还对收益率、平均持续期以及考虑风险分散等给出限制条件.这是决策问题描述的第二个方面：各种限制条件.任何事情往往都有若干前提条件或实际限制.比如，不可能不顾安全，不顾风险；不可能不计成本，不计代价；不可能不受自然规律的制约等等.

作为决策者，当然不会忘记，应使桤椤树公司债券组合投资的年收益率期望值达到最大.这是问题描述的第三个方面，也就是目标.

上面给出了问题描述的初步分析.在决策问题以及许多其他问题中，往往需要用数学方式来描述问题，以便于用数学方法来有效地解决问题.问题的数学描述称为数学模型（有时简称模型，只要上下文明白），获得数学模型的过程就叫数学建模（有时简称建模）.下面我们一起来试着建立桤椤树公司的组合投资决策问题的数学模型（参见§9.1），它是债券组合投资模型的一个简化版本.

设4种债券的投资额分别为 x_1, x_2, x_3, x_4，单位为万美元.

投资总额为1 000万美元：$x_1 + x_2 + x_3 + x_4 = 1\,000$.

债券组合投资的年收益率最低值至少为8%（参见表9.1）：$(0.06x_1 + 0.08x_2 + 0.10x_3 + 0.09x_4)/(x_1 + x_2 + x_3 + x_4) \geqslant 0.08$，即 $2x_1 - 2x_3 - x_4 \leqslant 0$.

债券组合投资的平均持续期不超过6年（参见表9.1，并注意组合债券的平

均持续期等于各债券按投资额加权平均)：$(3x_1+4x_2+7x_3+9x_4)/(x_1+x_2+x_3+x_4)\leqslant 6$，即 $-3x_1-2x_2+x_3+3x_4\leqslant 0$.

每种债券的投资额度不能超过总投资额度的40%(体现了分散投资原理)：$x_1\leqslant 1\ 000\times 40\%=400$，同样 $x_2\leqslant 400, x_3\leqslant 400, x_4\leqslant 400$.

还有没有漏掉的限制？投资额为负是不可能的，所以必须有 $x_1\geqslant 0, x_2\geqslant 0, x_3\geqslant 0, x_4\geqslant 0$. 初学者最容易犯的错误往往就是“忘掉”了单个因素取值范围这种最简单的限制.

最后是目标的表示，债券组合投资的年收益率期望值达到最大(参见表9.1)：$0.13x_1+0.08x_2+0.12x_3+0.14x_4$ 达到最大.

综合起来，就得到下面的数学模型：

求 x_1, x_2, x_3, x_4，使 $0.13x_1+0.08x_2+0.12x_3+0.14x_4$ 达到最大，并满足：

$$x_1+x_2+x_3+x_4=1\ 000,$$

$$2x_1-2x_3-x_4\leqslant 0,$$

$$-3x_1-2x_2+x_3+3x_4\leqslant 0,$$

$$0\leqslant x_1\leqslant 400,$$

$$0\leqslant x_2\leqslant 400,$$

$$0\leqslant x_3\leqslant 400,$$

$$0\leqslant x_4\leqslant 400.$$

现在，桫椤树公司的组合投资决策问题变成了究竟 x_1, x_2, x_3, x_4 各等于多少？

§9.3 线性规划——在平直世界中获取最大利益

桫椤树公司和许多华尔街的公司一样，使用线性规划(linear programming)模型来选择理想的组合投资方案.

前一节中，我们已经把桫椤树公司的债券组合投资问题变成了一个数学问题，这个数学模型就是一个线性规划模型.

线性规划属于最优化(optimization)范畴，例如“最优决策”或“最优设计”. 最优化问题是企业管理、科技研发和工程设计中常见的问题. 要表述一个最优化问题(即建立数学模型)，应明确三个要素：

决策变量(decision variable) 它们是决策者(你)所控制的那些数量，它们取什么数值需要决策者来决策，最优化问题的求解就是找出决策变量的最优取值. 例如，在债券组合投资问题中，所考虑的各债券的投资额就是决策变量.

约束条件(constraint condition)　它们是决策变量在现实世界中所受到的限制或者决策者规定的一些强制性要求,正如前一节已经提到的,不可能不顾安全,不顾风险;不可能不计成本,不计代价;不可能不受自然规律的制约等等. 在债券组合投资问题中,比如投资总额是固定的,每种债券的投资额不能为负等. 约束条件通常是一组关于决策变量的等式和不等式.

目标函数(objective function)　它代表决策者希望对其进行优化的那个指标,根据不同情形,可能希望最大,也可能希望最小. 例如,把目标定为效益,自然要求最大;如果目标是风险之类,当然就要求最小. 目标函数是决策变量的函数. 在债券组合投资例子中,目标是债券组合投资的年收益率期望值达到最大.

1) 在实际中,约束条件和目标函数哪个更重要?

2) 如果有多个目标怎么办? 比如"花钱要少又要买好东西".

3) "尽量少花钱"和"花钱的最高限额"分别应该放在目标函数里还是约束条件里?

有了上面几个概念,我们就可以来探讨线性规划模型. 由于这本书不是面向数学理论研究的,而是面向各专业实际应用,所以我们只给出线性规划模型的两种实用形式. 在更多的相关信息资源 3 中,你还可以找到线性规划模型的一般形式和标准形式,各种形式之间都是可以相互转化的. 但无论如何,你都必须明白向量/矩阵表示法.

线性规划模型中,约束条件和目标函数中只含有决策变量的线性函数.

1) 线性规划模型实用形式一

$$\begin{aligned} &\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}. \\ &\text{s. t.} \quad \boldsymbol{A}_1\boldsymbol{x}=\boldsymbol{b}_1, \\ &\qquad \boldsymbol{A}_2\boldsymbol{x}\leqslant\boldsymbol{b}_2, \\ &\qquad \boldsymbol{L}\leqslant\boldsymbol{x}\leqslant\boldsymbol{U}. \end{aligned}$$

其中,$\boldsymbol{x}=[x_1\ x_2\cdots\ x_j\cdots\ x_n]^{\mathrm{T}}$是决策变量(向量);$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 为变量的下界(可为负无穷)和上界(可为正无穷). min 即 minimization 表示"极小化"(如果是"极大化"用 max 即 maximization 表示);s. t. 即 subject to 表示"使…满足".

2) 线性规划模型实用形式二

$$\begin{aligned} &\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}. \\ &\text{s. t.} \quad \boldsymbol{A}_1\boldsymbol{x}=\boldsymbol{b}_1, \end{aligned}$$

$$\boldsymbol{b}_2 \leqslant \boldsymbol{A}_2\boldsymbol{x} \leqslant \boldsymbol{b}_3,$$

$$\boldsymbol{L} \leqslant \boldsymbol{x} \leqslant \boldsymbol{U}.$$

其中 $\boldsymbol{b}_2$ 和 $\boldsymbol{L}$ 可为负无穷，$\boldsymbol{b}_3$ 和 $\boldsymbol{U}$ 可为正无穷.

提示 “简洁”和“明了”基本上不可分割. 简明是数学的本质之一，它对数学功不可没. 向量/矩阵表式法其实不过是数学中的一种简明形式而已，但其奥妙也恰恰在于此.

在选择软件求解线性规划模型时，最好选择可以支持上述两种形式的软件. 如果你自己开发程序(例如，你需要线性规划算法作为你的软件项目中的一个模块，而又没有找到合适的现成的情形)，可以考虑根据需要提供对这两种形式的支持.

想 1) 如果你所用的线性规划软件不支持上述实用形式，你就必须在使用该软件求解之前把你的问题转化成它所支持的形式.

2) 如何将目标函数求极大转化为求极小?

3) 如何将 $\boldsymbol{b}_2 \leqslant \boldsymbol{A}_2\boldsymbol{x} \leqslant \boldsymbol{b}_3$ 转化为 $\boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}$?

根据上面的规范，§9.3 中得出的桫椤树公司的组合投资决策问题的数学模型就可以写成：

$$\begin{aligned}
&\min -0.13x_1 - 0.08x_2 - 0.12x_3 - 0.14x_4.\\
&\text{s.t.}\quad x_1 + x_2 + x_3 + x_4 = 1\,000,\\
&\qquad 2x_1 - 2x_3 - x_4 \leqslant 0,\\
&\qquad -3x_1 - 2x_2 + x_3 + 3x_4 \leqslant 0,\\
&\qquad 0 \leqslant x_1 \leqslant 400,\\
&\qquad 0 \leqslant x_2 \leqslant 400,\\
&\qquad 0 \leqslant x_3 \leqslant 400,\\
&\qquad 0 \leqslant x_4 \leqslant 400,
\end{aligned}$$

或者：

$$\begin{aligned}
&\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}.\\
&\text{s.t.}\quad \boldsymbol{A}_1\boldsymbol{x} = \boldsymbol{b}_1,\\
&\qquad \boldsymbol{A}_2\boldsymbol{x} \leqslant \boldsymbol{b}_2,\\
&\qquad \boldsymbol{L} \leqslant \boldsymbol{x} \leqslant \boldsymbol{U}.
\end{aligned}$$

其中，$\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4]^{\mathrm{T}}$ 是决策向量，代表 4 种债券的投资额；$\boldsymbol{c} = [-0.13$

$-0.08\ -0.12\ -0.14]^{\mathrm{T}}$; $\boldsymbol{A}_1=[1\ 1\ 1\ 1]$; $\boldsymbol{b}_1=1\ 000$; $\boldsymbol{A}_2=\begin{bmatrix}2 & 0 & -2 & -1\\ -3 & -2 & 1 & 3\end{bmatrix}$; $\boldsymbol{b}_2=[0\ 0]^{\mathrm{T}}$; $\boldsymbol{L}=[0\ 0\ 0\ 0]^{\mathrm{T}}$; $\boldsymbol{U}=[400\ 400\ 400\ 400]^{\mathrm{T}}$.

到此为止,桫椤树公司的债券组合投资问题的数学模型已经写成了非常规范的形式,尽管还没有给出问题的答案. 建立模型以后,通常要做的事包括分析、求解、设计、验证等. 其实,有了数学模型以后,在很多情况下,说问题已经解决了一大半应该是不为过的. 对于线性规划而言,主要是求解问题. 许多复杂数学问题的求解,常常需要用算法进行求解,算法这个概念是非常重要的.

在介绍线性规划的一些算法之前,有必要弄清有关线性规划问题解的若干概念,这是因为,尽管你并不从事线性规划本身的研究,但你必须对所用的软件工具的输出结果有清晰的认识和理解,为此,至少应掌握以下的基本概念.

1) 可行解或可行点(feasible solution):满足约束条件的决策向量 $\boldsymbol{x}\in\boldsymbol{R}^n$,其中 $\boldsymbol{R}^n$ 表示 n 维欧氏空间. 换句话说,是满足约束条件中所有等式和不等式的解.

2) 可行域或可行集:全部可行解构成的集合. 它是 n 维欧氏空间 $\boldsymbol{R}^n$ 中的点集,而且是一个"凸多面体". 如果线性规划问题无解,即不存在任何可行解,那么可行集就是空集.

3) 最优解:使目标函数达到最优值(最大值或最小值,并且有界)的可行解,这也就是线性规划问题的解. 最优解不一定惟一.

4) 无界解:若求极大化时目标函数在可行域中无上界,若求极小化时目标函数在可行域中无下界. 这是一种特殊情况.

1) 一个线性规划的解仅有三种可能的情况:最优解、无解和无界解.

2) 无界解是不合实际的,它通常意味着所建立的模型有问题或没有正确地把模型输入计算机.

1) 无解的情况可能符合实际吗?

2) 有无界解时可行域是无界的. 反过来,如果可行域是有界的,则意味着什么?

从计算复杂性的观点,找出可行解并不比找出最优解更容易,事实上两者的难度本质上相同!(两种问题可以互相转化).

线性规划中有这么一个定理:线性规划问题若有最优解,则最优解必是作为可行域的凸多面体的某个顶点(对一般凸集而言称为极点).

有了上面的一些结论后,我们就可以介绍线性规划算法.下面介绍几种具有代表性的线性规划算法.

提示

1)既然本书的读者大多仅对实际应用感兴趣而不是理论研究,只需大致了解某些线性规划算法的背景就够了.重要的是会用现成的软件工具.

2)现实中出现的线性规划问题可能是几百、几千、几万以至几十万、上百万个决策变量/约束条件这样的规模.好在当今的计算工具(硬件+软件)能够对付.

---

1)单纯形法和修正单纯形法

单纯形法(simplex method)从可行域的一个顶点到相邻的使目标函数值严格下降的另一个顶点的迭代直到达到最优点.它是非多项式时间算法.但在概率平均意义下不仅时间复杂性是多项式的,并且是线性时间复杂性,这就解释了为什么它在实践中的高效性.这是一个在实践中久经考验的算法,是用得最多的算法,至今仍然是好的选择.许多线性规划软件中都实现了此算法.

2)椭球算法

求解整系数严格线性不等式组的算法.由于求解线性规划模型与求解整系数严格线性不等式组是等价的,因此椭球算法是被看作线性规划算法的.它是多项式时间算法,但在实践中的效果不佳,因此实际中不宜选用.但是,它的重大意义在于,由于此算法的发现,终于消除了人们长期以来关于"线性规划会是NP问题吗?"的疑虑(关于NP问题请见§9.5).

3)Karmarkar算法

由在Bell实验室效力的Karmarkar发明的轰动一时的算法,其特点是穿越可行域内部而不是边界一步步地达到最优点,属于所谓内点法(interior-point methods)或障碍法(barrier methods).它源于非线性规划中的障碍法.与椭球算法一样,它也是多项式时间算法,其不同之处在于,它不仅具有理论的优势,同时在实践中以优异的性能超过了单纯形算法,因而在其被提出的时候引起了全世界的轰动.但当时Karmarkar迟迟不公开算法的某些细节,而Bell实验室以此算法为基础的商用软件标价百万美元.

后面的事情,就是要用线性规划算法/软件来求解问题了.

§9.4 用线性规划软件求解组合投资问题

相信大部分读者都乐意用现成的软件来求解问题. 当今的线性规划算法/软件有两类. 第一类就是线性规划算法的实现,其功能是输入线性规划模型中的系数($\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}$ 等)及相关信息,输出最优解和一些附加信息. 另一类称为"建模系统"(modeling systems),它不仅具有通常的求解功能,更能让人们方便地描述原始问题并对其解进行分析. 这类软件以非常自然的问题描述作为输入(采用所谓建模语言 modeling language),然后自动转换成算法所需要的形式进行求解,并以自然的形式输出.

1) 商业软件是要付费的,除了所谓演示版(demo versions)或"学生"版("Student" versions)外.

2) 那些全免费(或对科研教学免费,对商业使用不免费)的软件也许功能有限或者不特别可靠,但还是很有用的.

终于到了要给出桫椤树公司的债券组合投资问题答案的时候了! 为了方便,这里把 §9.3 中列出的模型复制到这里:

$$\begin{aligned} &\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}.\\ &\text{s.t.}\quad \boldsymbol{A}_1\boldsymbol{x}=\boldsymbol{b}_1,\\ &\qquad \boldsymbol{A}_2\boldsymbol{x}\leqslant\boldsymbol{b}_2,\\ &\qquad \boldsymbol{L}\leqslant\boldsymbol{x}\leqslant\boldsymbol{U}. \end{aligned}$$

其中,$\boldsymbol{x}=[x_1\ x_2\ x_3\ x_4]^{\mathrm{T}}$ 是决策向量,代表 4 种债券的投资额;$\boldsymbol{c}=[-0.13\ -0.08\ -0.12\ -0.14]^{\mathrm{T}}$;$\boldsymbol{A}_1=[1\ 1\ 1\ 1]$;$\boldsymbol{b}_1=1\ 000$;$\boldsymbol{A}_2=\begin{bmatrix}2 & 0 & -2 & -1\\ -3 & -2 & 1 & 3\end{bmatrix}$;$\boldsymbol{b}_2=[0\ 0]^{\mathrm{T}}$;$\boldsymbol{L}=[0\ 0\ 0\ 0]^{\mathrm{T}}$;$\boldsymbol{U}=[400\ 400\ 400\ 400]^{\mathrm{T}}$.

用什么软件呢? 就用 MATLAB 吧! MATLAB 是一款优秀的通用计算和仿真设计软件,包括线性规划和非线性规划模型的求解.

1. MATLAB 中的线性规划有关函数(MATLAB6.1)

在 MATLAB 中,同一个函数有多种形式,求解线性规划的函数 LINPROG 也是如此. 这里向你推荐 LINPROG 中最实用的形式:

```
[x,fmin] = LINPROG(c, A, b, Aeq, beq, xL, xU)
```

它求解下列线性规划模型：

$$\min f = c^{T}x.$$

s. t.

$$Ax \leqslant b,$$

$$Aeq = beq,$$

$$xL \leqslant x \leqslant xU.$$

2. 求解杪椤树公司问题的 MATLAB 程序

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  BondsLinprog.m

  min -0.13*x1 -0.08*x2 -0.12*x3 -0.14*x4
  s.t.       x1       +x2        +x3        +x4    =1000
          2*x1                 -2*x3        -x4    <=0
         -3*x1     -2*x2         +x3      +3*x4    <=0
         0 <=x1 <=400
         0 <=x2 <=400
         0 <=x3 <=400
         0 <=x4 <=400
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  echo off;
  close all hidden;
  fclose('all');
  clear;
  clc;
  format short;

  c=[-0.13;-0.08;-0.12;-0.14]
  A=[2 0 -2 -1;-3 -2 1 3]
  b=[0;0]
  Aeq=[1 1 1 1]
  beq=[1000]
  xL=zeros(4,1)
  xU=400*ones(4,1)

  [x,fmin]=LINPROG(c,A,b,Aeq,beq,xL,xU);
  Bond1=x(1)
  Bond2=x(2)
```

```
Bond3 = x(3)
Bond4 = x(4)
ReturnExpectation = - fmin
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3. 程序运行结果

```
Optimization terminated successfully.

Bond1 =
  400.0000
Bond2 =
  6.1446e-010
Bond3 =
  300.0000
Bond4 =
  300.0000

ReturnExpectation =
  130.0000
```

4. 桫椤树公司问题的解答

债券 1 投资 400 万美元;债券 2 投资 0 万美元(不投资债券 2);债券 3 投资 300 万美元;债券 4 投资 300 万美元. 此组合投资决策满足所有要求并使得投资年收益率期望值达到最大为 130 万美元.

修改程序使之:仅仅求一个可行解并验证解的可行性即行.

---

§9.5 如果决策变量只能取整数怎么办

在桫椤树公司的债券组合投资问题中,决策变量是几种看好的债券的投资额度,在实际中这些决策变量取整数(单位用美元)应该更合理一些,尽管给出实数结果也无大碍. 然而,在另外一些情形,比如一个家电厂商,电视机应该生产多少台,电冰箱应该生产多少台,空调应该生产多少台等,如果给出实数的决策结果,就显得不合理了.

加上决策变量只能取整数这样的限定,线性规划就变成了**整数线性规划**(integer linear programming),有时简称整数规划(integer programming). 相对于整数线性规划,前述线性规划可称为普通线性规划(ordinary linear programming).

整数线性规划是不是更简单了？回答通常会出乎初学者的意料：它比普通线性规划无论在理论上还是实践中都要难得多！

1）“整数线性规划比普通的线性规划要难得多”这一点，并非是用直觉就易于理解的. 其实，变量只能取整数，这相当于加上了一类很强的约束条件，因此通常使求解的难度大大增加而不是难度减小！

2）整数线性规划是组合问题（combinatorial problems），两者都属于所谓“NP - 完全问题”（NP - Complete Problem），即未找到多项式时间算法的问题，并且人们相信对此类问题根本不存在多项式时间算法，计算量将随问题规模（变量数、约束数等）按指数增长.

整数线性规划的困难在于当问题的规模（变量与约束的个数）增大时，计算量将爆炸性（即按指数规律）增加. 下面简介一些常用求解方法.

1）穷举法：变量数（n）大时不可行.

2）舍入凑整法：变量数（n）大时同样不可行，因为与非整数最优点相邻的整数个数为 2^n，故此时等效于求一个 0 - 1 规划的穷举法.

3）分支定界法：比较可行.

4）割平面法：比较可行.

1）在实践中，不必把整数规划与普通规划分得太清. 许多实际问题中，建模本身就包含了一些不确定因素，同时常常也允许近似的或粗略的结果. 不必太在意严格和完美.

2）我们发现舍入凑整法在实践中经常是有用的！如枞椤树公司的投资问题中，若要各投资额为整数（单位用美元），用舍入凑整法是没有问题的！

§9.6 操　　练

操练一　动物饲料配置

奇克兹公司专门饲养并出售一种家禽. 为了让家禽健康而快速生长，同时降低成本，奇克兹公司进行了大量实验，发现该种家禽的生长主要依赖于饲料中的三种营养成分，即蛋白质、矿物质和维生素. 实验测定结果表明，每只该种家禽每

天至少需要 200 g 蛋白质、10 g 矿物质和 30 mg 维生素. 奇克兹公司能得到的供应饲料有五种,每种饲料每 kg 所含的营养成分如表 9.2,每种饲料每 kg 的成本如表 9.3. 奇克兹公司希望找出满足动物营养需要而成本又最低的混合饲料配置.

表 9.2　每种饲料每 kg 所含的营养成分

饲料	蛋白质/g	矿物质/g	维生素/mg
1	0.30	0.10	0.05
2	2.00	0.05	0.10
3	1.00	0.02	0.02
4	0.60	0.20	0.20
5	1.80	0.05	0.08

表 9.3　每种饲料每 kg 的成本

饲料	1	2	3	4	5
成本/元	0.2	0.7	0.4	0.3	0.5

1) 决策变量: $x_j(j=1,2,3,4,5)$ 为每天所需混合饲料中第 j 种饲料的 kg 数.

2) 约束条件:

蛋白质每天至少 200 g: $0.30x_1+2.00x_2+1.00x_3+0.60x_4+1.80x_5\geqslant 200$,

矿物质每天至少 10 g: $0.10x_1+0.05x_2+0.02x_3+0.20x_4+0.05x_5\geqslant 10$,

维生素每天至少 30 mg: $0.05x_1+0.10x_2+0.02x_3+0.20x_4+0.08x_5\geqslant 30$,

每种饲料的 kg 数不能为负: $x_j\geqslant 0(j=1,2,3,4,5)$.

3) 目标函数:混合饲料的成本最低: $\min\ 0.2x_1+0.7x_2+0.4x_3+0.3x_4+0.5x_5$.

操练二　面包产量配比

田园食品公司生产的面包很出名. 他们生产两种面包:一种是叫“唐师”的白面包,另一种是叫“宋赐”的大黑面包. 每个唐师面包的利润是 0.05 元,宋赐面包是 0.08 元. 两种面包的月生产成本是固定的 4 000 元,不管生产多少面包.

该公司的面包生产厂分为两个部门:分别是烤制部和调配部.

烤制部有 10 座大烤炉，每座烤炉的容量是每天出 140 台，每台可容纳 10 个唐师面包或 5 个更大的宋赐面包. 可以在一台上同时放两种面包，只需注意宋赐面包所占的空间是唐师面包的两倍.

调配部每天可以调配最多 8 000 个唐师面包和 5 000 个宋赐面包. 有两个自动调配器分别用于两种面包的调配而不至于发生冲突.

田园公司决定找出这两种面包产品的最佳产量配比，即确定两种面包的日产量，使得在公司面包厂的现有生产条件下利润最高.

决策变量：

TS：唐师面包日产量（个/日），

SC：宋赐面包日产量（个/日）.

约束条件：

$1/10TS + 1/5SC \leqslant 10 \times 140 = 1\ 400$（每天占烤炉所烤出的台数），

$TS \leqslant 8\ 000$，

$SC \leqslant 5\ 000$.

目标函数：

Max $Profit = 0.05TS + 0.08SC - 4\ 000/30$.

更多的相关信息资源

1 Phillips D T, Ravindran A and Solberg J J. OPERATIONS RESEARCH: Principles and Practice. John Wiley & Sons, Inc., 1976

2 Cooper L, Bhat N and LeBancc L J. Introduction to Operation Research Models. New York: W. B. Saunders Company. 1980

3 傅鹂，龚劬，刘琼荪，何中市. 数学实验. 北京：科学出版社，2000

4 http://www - unix. mcs. anl. gov/otc/Guide/faq/linear - programming - faq. html

第 10 章

世界本复杂，如何做得最好

——非线性规划

后来，人们仍然无法回避模糊性. 还是更早地就去处理了不确定性. 最终，还得十分勇敢地进入了非线性的无限天地. 有痛苦也有惊喜. 非线性的世界是那样地复杂混沌、难测难控. 但是人们终于充分意识到，线性其实根本不是完美，它也让世界单调乏味没有悬念；而正是非线性，使得世界如此多姿多彩，富有悬念！你看看一年三百六十五天的风雨阴晴，你看看人间千姿百态的喜怒哀乐.

——作者

§10.1 公交公司的调控策略

我们的各大城市正在经历大堵车的痛苦阶段（但愿只是一个阶段），重现着发达国家曾经（或仍然）“拥有”的奇观. 比过去多得多，宽得多的路看起来却更像巨大的停车场，而真正的停车场却显得那么少那么小. 发展高效率的公交系统，才是解决问题的根本出路.

于是你创办了红狐狸公交公司（假设是这样），公司发展异常迅猛，规模已经十分巨大.

营业额最大化问题 红狐狸公交公司的年预算为 2 亿元，公司希望在预算内实现最大营业额. 市场调查分析表明，现阶段客源是绝对充足的，在这种情况下，营业额与公司所有营运车辆能够行驶的总里程成正比. 经过统计分析，总里程是车辆总数、员工总数、燃油总量的函数，有如下经验公式：

$$M = 15.7B^{0.06}W^{0.32}F^{0.56},$$

其中 M 是总里程（单位：10^3km），B 是车辆总数，W 是员工总数，F 是燃油总量（单位：m^3）. 因此公司可以通过调控车辆总数、员工总量、燃油量来实现最大营业额.

目前，公司有巴士 700 辆，员工 2 200 名. 要增加车辆的数量，就要购买新车辆，当前的价格是每辆 21 万元；要减少车辆的数量，就要卖掉若干车辆，当前卖

掉一辆车可收回 7.5 万元. 每辆车的年维护费用是 8 000 元. 员工数量的增减就是雇佣和解雇,分别都要产生一定的费用,为了雇佣一个新员工要花费 8 000 元(包括新员工培训费用),而解雇一个员工要花费 6 000 元(包括一次性补偿). 每个员工的年薪是 3 万元. 公司必须为每辆车配备至少 3 名员工. 燃油价格是 3 000 元/立方米. 每辆车每年最多只能消耗 50 m^3 燃油.

红狐狸公交公司希望知道,在上述条件下,公司如何能够实现年营业额最大化?

---

1) 客观世界中,多种因素联合产生对结果的影响有两种基本情形,一种是"加",一种是"乘":

"加": $Y=\alpha+\omega_1X_1+\omega_2X_2+\omega_3X_3+\cdots$,

"乘": $Y=\alpha X_1^{\omega_1}X_2^{\omega_2}X_3^{\omega_3}\cdots$,

其中 X_i 是影响 Y 的某个因素,而 ω_i 是该因素影响力的方向和大小(大小可称为权重).

提示

2) "乘"的关系取对数后形式上可以转化为"加"的关系,但是客观世界中关系本身属于哪种是无法改变的. 现实中,似乎有不少的人在某些问题的算法上显得糊里糊涂,根源可能就在于对"加"和"乘"的"应用题"从来没有真正吃透过.

3) 总里程的经验公式 $M=15.7B^{0.06}W^{0.32}F^{0.56}$ 属于"乘"的关系.

4) 当然还有许多复杂的关系,比如"加"和"乘"的结合,各种数学映射及其复合,那就太多了.

---

§10.2 营业额最大化

红狐狸的问题怎么解决? 相信你多半已经读过上一章关于"线性规划"的内容,因此可能还没有忘记首先要建立最优化问题数学模型,而最优化问题的模型有三个要素,即"决策变量"、"约束条件"和"目标函数". 下面我们就红狐狸公交公司的营业额最大化问题进行三要素分析.

请参见"营业额最大化问题"中的描述和有关数据.

决策变量

BP:要新购买的巴士数.

BS:要出售的巴士数.

WH:要新雇佣的员工数.

WF:要解雇的员工数.

FL:每辆车年用燃油量(单元:m^3).

约束条件

1)年预算约束(单位:万元):

车辆费用:$21BP-7.5BS+0.8(700+BP-BS)$.

员工费用:$3(2\ 200+WH-WF)+0.8WH+0.6WF$.

燃油费用:$0.3(700+BP-BS)FL$.

于是预算约束为:

$$21BP-7.5BS+0.8(700+BP-BS)+3(2\ 200+WH-WF)+0.8WH+0.6WF+0.3(700+BP-BS)FL\leqslant 20\ 000.$$

2)最少员工数量:

$$3(700+BP-BS)\leqslant 2\ 200+WH-WF.$$

3)每辆车每年最大燃油消耗量(单元:m^3):

$$FL\leqslant 50.$$

4)非负约束:

$$BP,BS,WH,WF,FL\geqslant 0.$$

目标函数

车辆总数:$700+BP-BS$.

员工总数:$2\ 200+WH-WF$.

燃油总量:$(700+BP-BS)FL$.

所以目标函数为:使得 $M=15.7(700+BP-BS)^{0.06}(2\ 200+WH-WF)^{0.32}[(700+BP-BS)FL]^{0.56}$最大化.

为了更简明一点,令 $BP=x_1,BS=x_2,WH=x_3,WF=x_4,FL=x_5$,综合起来,就得到下面的数学模型:

求 x_1,x_2,x_3,x_4,x_5,使 $15.7(700+x_1-x_2)^{0.06}(2\ 200+x_3-x_4)^{0.32}[(700+x_1-x_2)x_5]^{0.56}$达到最大,并满足:

$$21x_1-7.5x_2+0.8(700+x_1-x_2)+3(2\ 200+x_3-x_4)+0.8x_3+0.6x_4+0.3(700+x_1-x_2)x_5\leqslant 20\ 000,$$

$$3(700+x_1-x_2)\leqslant 2\ 200+x_3-x_4,$$

$$0\leqslant x_1,$$

$$0\leqslant x_2,$$

$$0\leqslant x_3,$$

$0 \leqslant x_4$,

$0 \leqslant x_5 \leqslant 50$.

现在,红狐狸公交公司的营业额最大化问题变成了决策变量 x_1, x_2, x_3, x_4, x_5 各等于多少?

§10.3 非线性规划——在复杂的世界里做得最好

红狐狸公交公司的营业额最大化问题是一个非线性规划问题,上一节中,我们已经得到了一个非线性规划的数学模型.

和线性规划惟一不同的就是,非线性规划模型中,目标函数或约束条件中至少含有一个非线性项. 例如红狐狸公交公司的营业额最大化模型中,目标函数是非线性的,约束条件中的年预算约束是非线性的.

1) 非线性因素使得非线性规划问题的求解比线性规划困难得多.

2) 非线性规划一般是指非线性约束优化(constrained optimization),但只要目标函数是非线性的,也可以讨论无约束优化问题,这一点与线性规划不同.

无约束的线性规划有没有实际意义?

下面给出非线性规划模型的规范写法.

1. 非线性规划模型 - 无约束优化

无约束优化(unconstrained optimization):

$$\min_{x \in \mathbf{R}^n} f(\boldsymbol{x}),$$

其中 $\boldsymbol{x}$ 是 n 维决策变量(向量),$f(\boldsymbol{x})$ 是目标函数,min 即 minimization,表示"极小化"(如果是"极大化",用 max 表示,即 maximization). 无约束优化的目标函数必须是非线性函数,否则无实际意义. 例如

$$\min_{x_1, x_2}(2x_1^2 + 3x_2^2)$$

是可以的,而

$$\min_{x_1,x_2}(2x_1+3x_2)$$

就是没有意义的，因为没有任何约束条件，包括对决策变量的非负约束，求最小化或最大化只可能使决策变量取正或负无穷大.

1）在实际中，真正无约束的情况是很少的，比如决策变量的取值实际上一般不可能是无穷大（正或负无穷大）.

2）有些约束对于优化问题有可能是不起作用的，例如，对目标函数 $f(x)=x^2$ 最小化，$x\leqslant 5$ 之类的约束就不起作用.

2. 非线性规划模型 - 约束优化问题

约束优化（constrained optimization）：

$$\min_{x\in \mathbf{R}^n} f(\boldsymbol{x}).$$

$$\text{s.t } h_i(\boldsymbol{x})=0\ (i=1,2,\cdots,m),$$

$$g_i(\boldsymbol{x})\leqslant 0\ (i=1,2,\cdots,p),$$

$$\boldsymbol{L}\leqslant \boldsymbol{x}\leqslant \boldsymbol{U}.$$

其中 $\boldsymbol{x}$ 是 n 维决策变量（向量），$f(\boldsymbol{x})$ 是目标函数，$h_i(\boldsymbol{x})\ (i=1,2,\cdots,m)$ 是等式约束函数，$g_i(\boldsymbol{x})\ (i=1,2,\cdots,p)$ 是不等式约束函数，$\boldsymbol{L}$ 是变量下界向量（分量可为负无穷），$\boldsymbol{U}$ 为变量上界向量（分量可为正无穷）. s. t. 是 subject to 的缩写，表示“使…满足”.

这种写法是一种很实用的形式. 约束优化的一个例子如下：

$$\min_{x_1,x_2}(2x_1^2+3x_2^2).$$

$$\text{s.t. } x_1^2+x_2^2\geqslant 1,$$

$$x_1\geqslant 0,$$

$$0\leqslant x_2\leqslant 2.$$

试用作图的方法把上述例子的解求出来.

一般可用 $\boldsymbol{x}^*$，$f^*=f(\boldsymbol{x}^*)$ 分别表示优化问题的解（或称最优解）和最优值（或最优目标函数值）.

根据上面的表示方法，红狐狸公交公司的营业额最大化问题，稍加整理后可以写得更加简明规范：

$$\min f(\boldsymbol{x}) = -15.7(700 + x_1 - x_2)^{0.06}\ (2\,200 + x_3 - x_4)^{0.32}\ [(700 + x_1 - x_2)x_5]^{0.56}.$$

s. t.

$$21.8x_1 - 8.3x_2 + 3.8x_3 - 2.4x_4 + 210x_5 + 0.3(x_1 - x_2)x_5 - 12\,840 \leqslant 0,$$

$$3x_1 - 3x_2 - x_3 + x_4 - 100 \leqslant 0,$$

$$0 \leqslant x_1,$$

$$0 \leqslant x_2,$$

$$0 \leqslant x_3,$$

$$0 \leqslant x_4,$$

$$0 \leqslant x_5 \leqslant 50.$$

现在的问题是，如何求解这个模型，为红狐狸公交公司的营业额最大化问题给出一个答案. 与线性规划一样，问题的求解需要相应的方法和算法. 尽管非线性规划也有相当丰富的求解方法，但远不如线性规划那样在实践中具有高效的性能，这是由非线性问题本身的复杂性决定的.

针对前面介绍的无约束优化问题和约束优化问题，非线性规划方法/算法也有相应的两类，即无约束优化问题算法和约束优化问题算法.

无约束优化问题算法的选择不仅要考虑计算的速度，还要考虑计算准备的方便性，而这两者常常是矛盾的. 通常，使用梯度的算法计算速度更快，例如 DFP 和 BFGS 都是很出色的算法，但要用到梯度，准备工作就更麻烦；另一方面，不使用梯度的算法，准备工作比较容易做，但计算速度多半就要慢得多. 有关算法的细节可以参考本章后面更多相关信息资源中的 1.

约束优化问题要比无约束优化问题更难，因此约束优化问题的一类有效而实用的方法是将其化为(一系列)无约束优化问题. 例如序列无约束极小化技术(SUMT)、可变容差法、逼近精确罚函数法(AEP)等. 其中逼近精确罚函数法(AEP)跟 SUMT 一样简洁，理论分析和数值试验都显示其可靠性高、效率高，可在更多相关信息资源中的 2 和 3 中找到详细的介绍.

1) 如果你并不是要从事非线性规划本身的研究，而只是应用，那么你对其算法只需略知一二就行了，多半是利用现有的软件工具对问题进行求解.

2) 如果你确实需要自己编写非线性规划求解程序，那就值得去读一本专门讲解非线性规划的书了.

§10.4 用非线性规划软件求解最大营业额问题

通常我们尽量利用现成的软件而不是动辄自己开发软件，那样做要么成功渺茫，要么大可不必. 求解非线性规划问题也是同样，我们首先考虑使用现成的软件，比如还用 MATLAB.

下面向你推荐两个最有用的分别求解无约束优化和约束优化的 MATLAB 函数(MATLAB 6.1).

1. 无约束优化

`[x,fmin] = FMINUNC(@ fun,x0)`

此函数求解下列无约束优化问题：

$\min f(\boldsymbol{x})$ （n 维决策变量）

其中：

`fun`：M－文件“`fum.m`”中定义了函数 `f = fun(x)`，`f = f(x)` 就是目标函数.

`x0`：输入参数 `x` 是决策变量的初始点(最优点的一个估计，可以随便给，当然越接近最优点求解越快).

`x`：输出的最优点.

`fmin`：输出的最优值.

2. 约束优化

`[x,fmin] = FMINCON(@ funobj,x0,A,b,Aeq,beq,xL,xU,@ funcon)`

此函数求解下列约束优化问题：

$$\min f(\boldsymbol{x}) \quad (n \text{ 维决策变量}).$$

s. t.

$$\boldsymbol{Ax} \leqslant \boldsymbol{b} \quad (\text{若干个线性不等式约束}),$$

$$\boldsymbol{Aeq} = \boldsymbol{beq} \quad (\text{若干个线性等式约束}),$$

$$g(\boldsymbol{x}) \leqslant 0 \quad (\text{若干个非线性不等式约束}),$$

$$h(\boldsymbol{x}) = 0 \quad (\text{若干个非线性等式约束}),$$

$$\boldsymbol{xL} \leqslant \boldsymbol{x} \leqslant \boldsymbol{xU} \quad (n \text{ 对上下界约束})$$

其中：

$\boldsymbol{A},\boldsymbol{b},\boldsymbol{Aeq},\boldsymbol{beq},\boldsymbol{xL},\boldsymbol{xU}$：与问题中的参数是完全对应的，不存在的东西可令其为空矩阵，比如不存在线性等式约束，则 `Aeq = [];beq = [];`

`funobj`：M－文件“`funobj.m`”中定义了函数 `f = funobj(x)`，`f = f(x)` 是目标函数.

x0:决策变量的初始点(最优点的一个估计,可以随便给,当然估计越准求解越快).

funcon:M-文件"funcon.m"中定义了对应的非线性约束函数[g,h]=funcon(x),g=g(x)为不等式约束,h=h(x)为等式约束,同样,没有的东西用空矩阵表示,比如没有非线性等式约束,则可写h=[].

x:决策变量的最优点.

fmin:目标函数最优值.

提示

1) MATLAB函数的表达形式,比如$[Y_1,Y_2]=f(a,b,x_1,x_2)$,可明白地表达函数参数的三种情况(输入、输出、输入/输出)以及多个返回值,大多数计算机语言似都不及.

2) 求解无约束优化问题应当首先要编一个M-文件fun.m(名称自己定),代表问题的目标函数;求解约束优化问题要首先编两个M-文件,分别是目标函数和非线性约束条件.

现在我们可以来求解红狐狸公司的问题了!为了清楚方便,把§10.3中得到的最后模型先拷贝过来:

$$\min f(\boldsymbol{x}) = -15.7(700+x_1-x_2)^{0.06}(2\,200+x_3-x_4)^{0.32}[(700+x_1-x_2)x_5]^{0.56}.$$

s.t.

$$21.8x_1-8.3x_2+3.8x_3-2.4x_4+210x_5+0.3(x_1-x_2)x_5-12\,840\leqslant 0,$$

$$3x_1-3x_2-x_3+x_4-100\leqslant 0,$$

$$0\leqslant x_1,$$

$$0\leqslant x_2,$$

$$0\leqslant x_3,$$

$$0\leqslant x_4,$$

$$0\leqslant x_5\leqslant 50.$$

1) 程序

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Busoptim.m

min -15.7*((700+x1-x2)^0.06)*((2 200+x3-x4)^0.32)*(((700+
x1-x2)*x5)^0.56)
```

```
s.t. 21.8 * x1 - 8.3 * x2 + 3.8 * x3 - 2.4 * x4 + 210 * x5 + 0.3 * (x1 - x2) *
x5 - 12840 <= 0
             3 * x1 - 3 * x2 - x3 + x4  <= 100
             0 <= x1
             0 <= x2
             0 <= x3
             0 <= x4
             0 <= x5 <= 50

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    function f = BusFunobj(x)
    f = -15.7 * ((700 + x(1) - x(2)) ^0.06) * ((2200 + x(3) - x(4)) ^0.32) *
    (((700 + x(1) - x(2)) * x(5)) ^0.56);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    function [g,h] = BusFuncon(x)

    g = 21.8 * x(1) - 8.3 * x(2) + 3.8 * x(3) - 2.4 * x(4) + 210 * x(5) + 0.3 *
    (x(1) - x(2)) * x(5) - 12840;
    h = [];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    Busoptim.m
    echo off;
    close all hidden;
    fclose('all');
    clear;
    clc;
    format short;

    x0 = [0;0;0;0;25];
    A = [3  -3  -1 1 0];
    b = [100];
    Aeq = [];
    beq = [];
```

```
xL = [0;0;0;0;0];
xU = [inf;inf;inf;inf;50];

[x,fmin] = FMINCON(@ BusFunobj,x0,A,b,Aeq,beq,xL,xU,@ BusFuncon);

Kilometers = - fmin

BusesPurchased = x(1)
BusesSold = x(2)
WorkersHired = x(3)
WorkersFired = x(4)
Fuel = x(5)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2）MATLAB 输出结果

```
Kilometers =
  1.0355e+005

BusesPurchased =
   -1.0661e-029

BusesSold =
  6.7049e-026

WorkersHired =
  615.7895

WorkersFired =
   -5.6645e-027

Fuel =
       50
```

3）实际问题的解答

红狐狸公交公司本年度不应购买新车，也不要卖掉车辆，不必解雇员工．要做的只是，增加新员工 616 名，每辆车配给燃油 50 m^3，总里程就将达到最大 103 550 10^3km，也就是营业额最大．

车辆数和员工数实际上都是不可能带有小数的，而我们在建立模型和求解模型时，显然没有加上这种限制，你怎么看待这一点？

在线性规划一章中，我们介绍了整数线性规划，同样，在非线性规划中，如果加上整数约束，则成为整数非线性规划（integer nonlinear programming），问题就难上加难！但是同样，在建立实际问题的模型时，有时已经存在相当多的近似与模糊，在这种情况下，就不必要对自己要求过高.

§10.5 山有多少峰，哪里是最高峰

线性的世界是平坦的. 线性规划中，约束条件是不可缺少的，否则，一个完全平面的无限世界如果存在最优，那就会在无尽头的地方——实际上是不可能达到的. 线性规划中的约束条件，就像是在平原上用平整的墙围成了区域，在有限的区域里总有最优.

非线性的世界就截然不同了. 在非线性规划中，存在没有约束条件的无约束优化问题. 非线性规划好像是在崇山峻岭中寻找山峰，山峰可能有很多很多，人们当然希望到达最高峰. 不幸的是，传统的非线性规划方法只能保证达到一个山峰，而不能保证达到最高峰. 寻求最高峰的问题，成为全局最优化（global optimization），众多山峰中的某一个山峰称为局部最优（local optima）.

全局优化的问题是一个非常基本的问题. 非常基本的意思就是很多问题最终都可以归结为全局优化的问题. 因此人们一直非常重视全局优化的研究，但是没有根本的突破. 近代的一些研究把眼光转向了奥妙无穷的自然界，物理世界和生物世界里其实就存在全局优化的问题. 于是有了模拟退火算法（simulated annealing algorithm）、遗传算法（genetic algorithm）为代表的“仿自然”算法，为全局优化领域带来了无限生机.

山路比平地难行. 但是，如果没有山高水低，世界一定是索然无味. 人们已经相信，世界的多姿多彩，来源于非线性.

§10.6 操　　练

操练一　桃李花园小区

有故事说，在一大片空地上，不知道路应该修在哪里. 有贤人指点，空地上种

满草，过一段时间，看哪些地方被踩出了道，就知道路该修在哪里了. 这是一种办法，但那种奢侈的条件，我们会碰见多少？即便一张白纸上好画最美的图画，如果缺少科学和艺术的规划，那种遗憾我们还要见多少？下面是一个园区规划中的一个小例子.

桃李花园是新建的一个生活住宅区，共有 20 栋住宅楼. 桃李花园内的所有道路都是东西或南北走向. 开发商拟在该区修建一个服务中心，地址选在离所有楼房的总路程最小的地方. 为了保证建筑物之间有足够的空间，服务中心的位置与其他楼房位置之间的距离不能少于 30 m(已经考虑了所有建筑的占地面积). 请你确定服务中心的位置.

参数：

$(a_i,b_i)(i=1\sim20)$：第 i 栋住宅楼的坐标.

决策变量：

(x,y)：服务中心的坐标.

约束条件：

$(x-a_i)^2+(y-b_i)^2\leqslant30^2=900\ (i=1\sim20)$.

目标函数：

$\min\ \sum_i|x-a_i|+|y-b_i|$.

操练二　翡翠温泉度假村

最近在翡翠山麓发现了温泉资源，西乐旅游开发公司获得开发权，决定在此修建“翡翠温泉度假村”. 该处主要的、相对集中的泉眼有 12 个. 现在的问题是，公司想首先划定一个大致是圆形的区域，用围墙围起来，围墙内包围了所有那 12 个泉眼，并且圆形区域越小越好. 请你解决这个问题.

参数：

(a_i,b_i) $(i=1\sim12)$：第 i 个泉眼的坐标.

决策变量：

(x,y)：围墙中心的坐标.

z：围墙的半径.

约束条件：

$(x-a_i)^2+(y-b_i)^2\leqslant z^2(i=1\sim12)$，

$0\leqslant z$.

目标函数：

$\min\ z$.

更多的相关信息资源

1 胡毓达主编. 非线性规划. 北京:高等教育出版社,1990

2 傅鹂. 两类逼近精确罚函数算法及其数值试验. 高校计算数学学报,1998 (2):Vol. 20. No. 2. pp154 - 162

3 傅鹂,龚劬,刘琼荪,何中市. 数学实验. 北京:科学出版社,2000

4 http://www.fi.uib.no/ ~ antonych/glob.html

第 11 章

如何表示二元关系

——图的模型及矩阵表示

改变问题的描述方式，往往是创造性的启发式解决问题的手段. 一种描述方式就好比我们站在一个位置和角度观察目标，有的东西被遮住，但若换一个位置和角度，原来隐藏着的东西就可能被发现. 采用一种新的描述方式，可能会产生新思想. 图论中的图提供了一种直观、清晰地表达已知信息的方式. 它有时就像解小学数学应用题中的线段图一样，能使我们用语言描述时未显示的或不易观察到的特征、关系，直观形象地呈现在我们面前，帮助我们分析和思考，激发我们的灵感.

——作者

§11.1 如何排课使占用的时间段数最少

学校要为一年级的研究生开设六门基础数学课：统计(S)、数值分析(N)、图论(G)、矩阵论(M)、随机过程(R)和数理方程(P). 按培养计划，注册的学生必须选修其中的一门以上，你作为教务管理人员，要设法安排一个课表，使每个学生所选的课程，在时间上不会发生冲突. 由于他们都要必修的外语和自然辩证法课程占用了许多时间，可供排课的时段不多，但可供使用的教室足够多，因此，有些课可能需要安排在同样的时段. 上述六门数学课中任何两门，只要不被同一个学生选择，可以安排在同样的时间段.

假设六门数学课的选课名单如表 11.1，哪些课程可以安排在同样的时段上？将六门课排在四个不同时段可行吗？如果只提供三个不同的时段，是否存在这样的安排，使每个学生都能上到他所选的课程. 要作出这样的安排的最少时段数是多少？

相对于我们要解决的问题来说，表 11.1 提供的信息太多，如何只把我们需要的信息单独地提取并清晰地表示出来使之更利于问题的解决？实际上，我们只需要知道哪些课会发生冲突，即哪些课被同一个同学选择，这样的课安排在不

同的时段即可. 如何描述课与课之间的相互冲突关系呢? 用六个小圆圈(称为顶点)代表六门课,当且仅当两门课同时被一个以上学生选择而不能安排在同样的时段时,在相应的两顶点间连一条线,例如,S 和 N 同时被黄大度同学选修,所以在 S 与 N 之间要连一条线,同样 S 与 G 同时被欧阳金同学选修,所以在 S 与 G 之间也要连一条线(见图 11.1(a)). 依次查看所有课程时,最终作出的图如图 11.1(b)所示.

表 11.1

S	N	G	M	R	P
李春兰	陈奇峰	化范文	张星	赵小民	许茂
郑文国	刘云	李出荣	夏雯	息志强	陈俊
姚南	刘元元	张惠	邵桂芳	陈修建	周清武
陈奇峰	黄大度	赵云	王学权	邹鑫	樊雪峰
王润惠	董舟	曹林军	单富民	刘元兵	刘伟
邹文燕	邹鑫	胡志强	董舟	杨成宝	甄军
万华	赵云	张敏	杨欣	邱吉洲	姜永东
李祖军	王凯	陈修建	吴军		
黄大度	李白彤	欧阳金	查小辉		
史武军	甄军	李晓	王坚		
刘昆	李欣	李白彤	程静波		
欧阳金	陈俊	万华	邹文燕		
郭志伟	于洪	曾光伟	卫迎新		

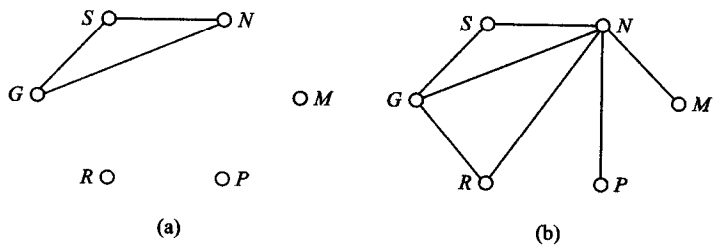


图 11.1

现在课与课之间的所有冲突关系已从表 11.1 中提取并直观形象地表示出来,比表 11.1 更加简洁、清晰和明了. 试一试,根据图 11.1(b)来解决前述问题,是否更容易.

表 11.2

时段	课　　程
1	
2	
3	
4	

哪些课程可以安排在同样的时段上，将六门课在四个不同时段的安排填入表 11.2. 我们可在作好的图上来操作，用 1,2,3,4 这四个数来标记该图的顶点，使标记相同的顶点不相邻，则标记相同的顶点对应的课程可安排在同样的时段上，因此标记为 1,2,3,4 的顶点对应的课程可分别安排在四个不同的时段. 比如在图 11.1(b)中，顶点 N 标记 1，顶点 S、P 标记 2，顶点 G、M 标记 3，顶点 R 标记 4. 由此表示课程 N 安排在第 1 时段上，课程 S,P 安排在第 2 时段上，课程 G、M 可以安排在第 3 时段上，课程 N 安排在第 4 时段上. 进一步地思考，如果只提供三个不同的时段，是否存在这样的安排，使每个学生都能上到他所选的课程.

用为顶点标记的方法，求出使学生上课不发生冲突且要求所花的时段数尽可能少的课程时间安排.

§11.2　一种直观形象的表示工具——图

像图 11.1(b)那样的由一些小圆圈(称为顶点 vertex)和一些联结它们的线(称为边 edge)所构成的图称为无向图(undirected graph)，一般用大写字母 G,H 表示. 图中顶点的位置，边的曲直长短都无关紧要，重要的是顶点与顶点之间的连接关系. 图 11.1(b)可以只需列出它的顶点 S,N,G,M,R,P 和边的端点对

$$e_1=(S,N),e_2=(S,G),e_3=(N,G),e_4=(N,M),e_5=(N,R),e_6=(N,P),e_7=(G,R)$$

便可确定. 通常用 V,E 分别表示图的顶点集，边集，图 G 可用 (V,E) 表示. 在图 11.1(b)中，

$$V=\{S,N,G,M,R,P\},$$

$$E=\{(S,N),(S,G),(N,G),(N,M),(N,R),(N,P),(G,R)\}.$$

任何元素集及其元素对之间的关系都可用图来描述. 例如，参加某次聚会的

人之间的相识关系，以人为顶点，当且仅当两顶点所代表的两人相识时，两顶点间连一边，所得之图便形象地表达出这群人中哪些人相识. 又如设 V 为一个城市中街道的交汇点的集合，E 代表所有联结交汇点的街道段，则 $G=(V,E)$ 便是图. 顶点的位置不必是实际的物理位置，边的长度也不必与真实的物理长度成比例. 还有铁路网中的火车站，道路交叉点，道路尽头看作顶点，铁路看作边，铁路网就是图. 其他的什么公路网、灌溉网、管道网、电话线网、计算机通讯网、输电线网统统都是图.

一个州的立法机关由许多委员会组成. 某些资深的立法委员身兼数职，因而各委员会的委员互相交叠. 说明怎样用图来描述这种委员会委员的互相交叠. 如果委员会 A 有委员(编号)1,3,5,6；委员会 B 有委员 2,4,8,10；委员会 C 有委员 1,7,9；委员会 D 有委员 2,5,8；委员会 E 有委员 2,4,10；委员会 F 有委员 11,12,13. 试绘出描述委员会委员互相交叠的图.

关于图中顶点与边的几个术语：

1）若边 e 的端点为 u,v，则称 e 与顶点 u,v **相关联**.

2）若顶点 u,v 之间有边相连，则称 u 与 v **相邻**.

3）若边 e_1,e_2 与同一顶点相关联，则称**相邻**.

4）端点相同的两边称为**重边**. 两端点为同一个点的一条边称为**环**.

例如，在图 11.2 中 e_1 与 v_1 相关联；v_1 与 v_2 相邻，e_1 与 e_2 相邻，e_4 与 e_5 是重边，e_6 是环.

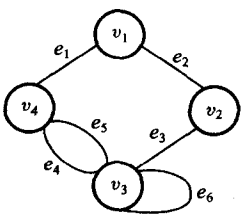


图 11.2

两种特殊图

简单图　无重边，无环的图.

完全图　任两顶点之间皆有边相连的简单图，记为 K_n，n 为图的顶点数.

若每条边都有方向，则称为**有向图**(directed graph). 你能给出一个有向图描述的实际例子吗？

若给每条边赋予一个或多个实数，这样的图称为**网络**. 这些数字可以代表距离、费用、可靠性或其他的相关参数.

在无向图中，与顶点 v 相关联的边的数目称为 v 的**度数**，记为 $d(v)$；一般用 $v(G)$ 和 $\varepsilon(G)$ 分别表示图 G 的顶点数和边数. 在有向图中，从顶点 v 引出的边的数目称为 v 的**出度**，记为 $d^+(v)$；指向顶点 v 的边的数目称为 v 的**入度**，记为 $d^-(v)$.

在一个足球赛季中，某足球联合会内每两队间比赛一次. 假定每次比赛结束后总有一方获胜(即无平局). 如何用一有向图来表示该赛季所有比赛的结局(即每次哪队获胜). 设 A,B,C,D 为该足球联合会各队，A 胜 B,D；B 胜 C,D；C 胜 A；D 胜 C，试绘出相应的图. 根据所绘之图，确定比赛名次.

§11.3 图的矩阵表示方法

任何事物群体及其元素之间的二元关系均可用图来描述，非常直观，形象，能使一些抽象的关系跃然纸上，助你思考，激发灵感. 另一方面，若想借助计算机来解决有关图的问题，又需将形转化为数，矩阵或数组就是一种表示图的很好工具，可作为图在计算机里的存储结构.

11.3.1 邻接矩阵

简单无向图的邻接矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$，其中

$$a_{ij}=\begin{cases}1, \text{当 } v_i \text{与 } v_j \text{相邻},\\ 0, \text{当 } v_i \text{与 } v_j \text{不相邻}.\end{cases}$$

图 11.3 中的图对应的邻接矩阵为

$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\end{array}\begin{array}{c}\begin{array}{ccccccc}1&2&3&4&5&6&7\end{array}\\ \begin{bmatrix}0&1&0&1&0&0&0\\1&0&1&0&1&1&0\\0&1&0&1&0&1&0\\1&0&1&0&0&1&1\\0&1&0&0&0&1&0\\0&1&1&1&1&0&1\\0&0&0&1&0&1&0\end{bmatrix}\end{array}.$$

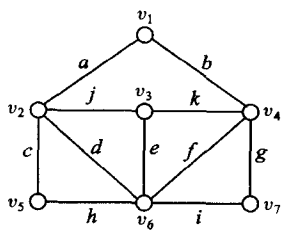


图 11.3

无向图的邻接矩阵是对称的，每行的元素之和为对应顶点的次数，每列的元素之和也为对应顶点的次数. 由上述邻接矩阵能作出原图吗？是否任意一个0－1矩阵都能惟一确定一个图？

对有向图，其邻接矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$ 中的元素 a_{ij} 取为 v_i 指向 v_j 的有向边的数目，图 11.4 中的有向图的邻接矩阵为

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}.$$

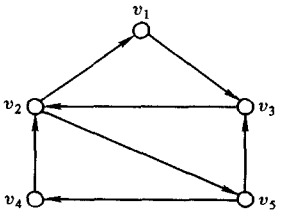


图 11.4

加权有向图的带权邻接矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$，其中 a_{ij} 取为 v_i 指向 v_j 的有向边上的权，当无边时取为 ∞，对角线上的元素为 0.

加权无向图的邻接矩阵可类似定义，是对称阵.

写出 MATLAB 环境下建立带权邻接矩阵的命令 M 文件，并由此产生图 11.5 的带权邻接矩阵.

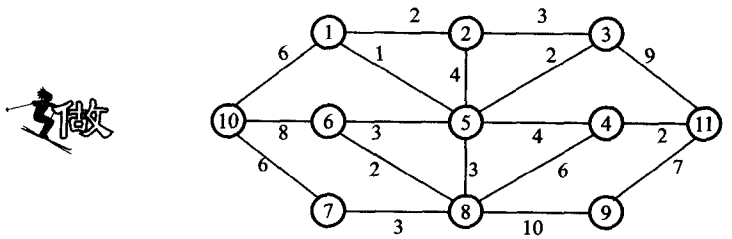


图 11.5

11.3.2 关联矩阵

无向图的关联矩阵 $\boldsymbol{M}=(m_{ij})_{n\times m}$，其中

$$m_{ij}=\begin{cases}1, & \text{若 } v_i \text{ 与 } e_j \text{ 相关联},\\ 0, & \text{若 } v_i \text{ 与 } e_j \text{ 不相关联}.\end{cases}$$

图 11.3 中无向图的关联矩阵为

$$
\begin{array}{c}
\\
1\\
2\\
3\\
4\\
5\\
6\\
7
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} a & b & c & d & e & f & g & h & i & j & k \end{array}\\
\left[\begin{array}{cccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0
\end{array}\right].
\end{array}
$$

注意关联矩阵每行的元素之和为对应顶点的次数，每列的元素之和为 2.

有向图的关联矩阵 $M=(m_{ij})_{n\times m}$，其中 m_{ij} 的取值为 1，-1，0，分别对应于 v_i 是 e_j 的起点，v_i 是 e_j 的终点和 v_i 不是 e_j 的端点三种情形.

11.3.3 边权矩阵

对于有向图，可定义一个 $2\times m$ 的矩阵 $\boldsymbol{E}$，第一，二行分别存放边的起点和终点. 若第 i 条边 e_i 的起点和终点分别为 v_j，v_k，则 $\boldsymbol{E}(1,i)=j$，$\boldsymbol{E}(2,i)=k$.

对于无向图，同样也可定义一个 $2\times m$ 的矩阵 $\boldsymbol{E}$，第一，二行分别存放边的两个端点，这两个端点中随便哪个放在第 1 行都可以. 若第 i 条边 e_i 的端点为 v_j，v_k，则 $\boldsymbol{E}(1,i)=j$，$\boldsymbol{E}(2,i)=k$ 或 $\boldsymbol{E}(1,i)=k$，$\boldsymbol{E}(2,i)=j$.

例如，图 11.3 中的无向图对应的边矩阵为

$$
\boldsymbol{E}=\begin{array}{c}
\begin{array}{cccccccccc} a & b & c & d & e & f & g & h & i & j & k \end{array}\\
\left[\begin{array}{cccccccccc}
1 & 1 & 2 & 2 & 3 & 4 & 4 & 5 & 6 & 2 & 3\\
2 & 4 & 5 & 6 & 6 & 6 & 7 & 6 & 7 & 3 & 4
\end{array}\right].
\end{array}
$$

对加权图，只需增加一行来存放各条边上的权，这样的矩阵称为**边权矩阵**.

写出 MATLAB 环境下由加权图的边权矩阵表示转化为带权邻接矩阵表示的 M 文件函数.

§11.4 操　　练

操练一　传输网络的可达度

可达度对于交通网络或通信网络来说非常重要，它是人员、货物或信息流动能力的直接表征，一个发达而有效的传输系统可达度较高. 可达度定义为一个点

到达其他点的能力的一种度量. 你认为,如何具体地规定一个点甚至一个图的可达度比较合理呢?

假设有 5 个站点,我们以图 11.6 所示的方式把它们联结起来,可按下述方法来比较它们的可达性.

1) 求出图 11.6 中四个图的邻接矩阵 $\boldsymbol{A}$ 的幂矩阵 $\boldsymbol{A}^2,\boldsymbol{A}^3,\boldsymbol{A}^4,\boldsymbol{A}$ 的 i 行 j 列元素正好是图中顶点 i 到顶点 j 的长度为 1 的路径数,$\boldsymbol{A}^k(k=2,3,4)$ 的 i 行 j 列元素呢?

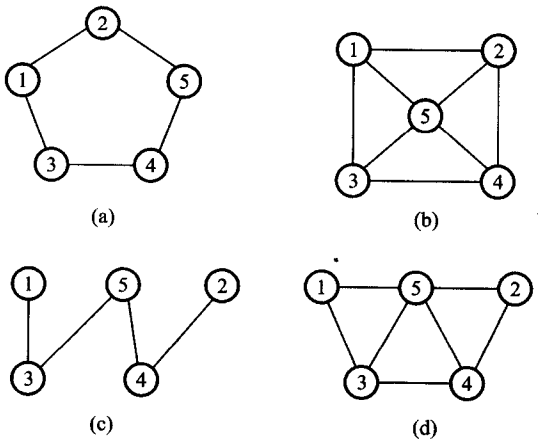


图 11.6

2) 图的直径 d 为相距最远的两顶点之间的距离,总可达矩阵为

$$\boldsymbol{T}=\sum_{k=1}^{d}\boldsymbol{A}^k,$$

计算图 11.6 的总可达矩阵,其元素的意义是什么?

3) 如何求出图 11.6 中四个图的距离矩阵 $\boldsymbol{D},\boldsymbol{D}=(d_{ij})_{n\times n}$,$d_{ij}$ 为顶点 i 到顶点 j 的最短路径的长度. 每个顶点的易到达性指标为它到其余各点的距离总和的倒数,试求出每个图的最易到达顶点,即易到达性指标最大的顶点.

操练二　传输网络的可达度

某城市计划新修建一条环线,市政部门想知道它对现有交通网络的影响以及对城市经济活动的影响. 城市的交通网络如图 11.7 所示,每个点的人口分别为:a 点有人口 5 000,b 点有人口 15 000,c 点有人口 25 500,d 点有人口 11 100,e 点有人口 9 500,f 点有人口 7 000,g 点有人口 4 500,h 点有人口 17 500,i 点有人口 18 300,j 点有人口 8 700,k 点有人口 3 000. 假设聘请你作为顾问来评估该城市交通网可达性的变化情况.

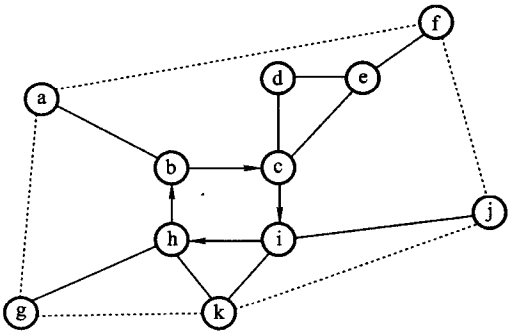


图 11.7

你的任务是：

1）计算现有交通网络（不包括点线）的可达性矩阵，给出最易到达的地点；

2）如果增加这条环线，计算该交通网络（包括点线）的可达性矩阵，并给出最易到达的地点；

3）在可达性方面有什么改进，哪些地点的可达性增长最大.

已知商业活动较大程度地依赖于可达性级别，在该城市范围的商业活动会如何？换言之，如果环线建好，城市的商业活动地点可能会有哪些？

操练三　田径赛的时间安排

假设某校的田径选拔赛共设六个项目的比赛，即跳高、跳远、标枪、铅球、100 m和 200 m 短跑，规定每个选手至多参加三个项目的比赛. 现有七名选手报名，选手所选项目如表 11.3 所示. 现在要求设计一个比赛日程安排表，使得在尽可能短的时间内完成比赛.

表 11.3　参赛选手比赛项目表

姓　名	项目 1	项目 2	项目 3
赵　宁	跳　高	跳　远	铅　球
钱　虎	跳　远	100 m	
孙　正	跳　高	200 m	
李　江	200 m	标　枪	铅　球
杨　众	跳　远	铅　球	跳　高
刘　平	铅　球	跳　高	200 m
王　跃	标　枪	跳　远	100 m

更多的相关信息资源

1 傅鹂,龚劬,刘琼荪,何中市. 数学实验. 北京:科学出版社,2000

2 Kenneth H. Rosen. Discrete Mathematics and Its Applications. 北京:机械工业出版社,2002

3 http:// www. math. fau. edu/ locke/graphthe. htm

第 12 章

如何连接通信站使费用最少

——最小生成树

数学家之擅长证明应缘于对证明过程的大量研读和反复实践. 同样,可以通过涉猎引人入胜、特色各异的算法,尝试设计各种问题的解决方法,培养算法设计的成熟性和机敏性.

——作者

§12.1 美国 AT&T 的网络设计算法攻关

1956 年,美国 AT&T 公司需要计算对几家商业客户的索价. 这几家客户想用数据通信线把一些站点与一个称为 Private Wire Service 的专用网络联结起来,如何求出使通信线花费最少的连接方式?

实际上若不允许通信线在非站点处连接,求使通信线花费最少的连接方式可以归结为求加权连通图的最小生成树的问题. 为了找出这样的树,工作人员作过各种可能的模型. 他们把一张大比例的美国地图铺在地板上,开始寻找联结所有站点的网络. 为了找出连线总长度最小的网络,要求对任意一对站点始终不出现两种联结方式. 以这种方式,比较聪明的人确实能得到很好的结果,但这种能用手工(并且跪着)操作的方式完成的问题是很有限的.

那时,Kruskal 算法刚刚发表,它的第一步是要整理所有站点对之间的距离表. 当有 n 个站点时,共有 $\binom{n}{2}=\frac{n(n-1)}{2}$ 个站点对. 对于站点数较少的问题,Kruskal 算法很不错,但 AT&T 需要解决的是有 500 个站点的网络连接问题. 没有一个人会用手工整理这样一个网络的 $500\times499/2=124\ 750$ 条边,而那时的计算机也不具有处理这样大规模数据集的能力. 在这种情况下,人们需要另一种算法.

1957 年,领导着贝尔实验室数学研究室的 Prim,得到了他的算法. Prim 算法优于 Kruskal 算法之处是 Prim 算法一次处理的数据不超过 n,因此 Prim 算法所需的存储器要比 Kruskal 算法小.

美国 AT&T 公司借助于计算机成功地应用这些算法为各种客户找到了连接若干站点的最经济连接方式.

§12.2 最小生成树——最经济的连接方式

美国 AT&T 公司需要为客户找到将一组站点连接起来的最经济的连接方式. 若不允许通信线在非站点处连接,任意两个站点可经过若干中介站点,取得联系.

为简便,以一个小型的问题为例. 假设各站点间能够铺设通信线路进行连接的情况如图 12.1 所示,顶点为站点,边为连接两站点之间的通信线,边的权为其预算费用.

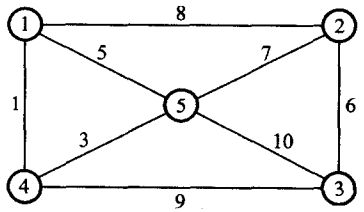


图 12.1

目的是要找到图 12.1 的一个连接所有顶点的具有最小总权数的连通子图. 先介绍几个术语.

连通图(connected graph) 其中任意两点之间都有路径的图. 如图 12.1 就是一个连通图.

圈(cycle) 当一条路径的起点和终点是同一顶点时,称这条路径为圈. 如图 12.1 中的路线 1—2—5—4—1 就是一个圈.

事实上,在最经济的网络中,不应该有任何圈,否则,去掉圈上的一条边,这圈上任意两个顶点仍能取得联系,正如在橡皮筋圈上剪一刀后,仍然是一个整段.

树(tree) 没有圈的连通图称为树. 如图 12.2.

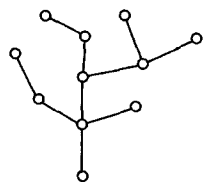


图 12.2 一个树图

树具有许多非常好的性质，这些性质包括：

A. 树中任意两点间有惟一路径.

B. 树的边数恰好等于顶点数减 1.

C. 在树中任意去掉一条边，将会不连通.

D. 树中任意两个不相邻顶点间添一边后，就恰好含一个圈.

E. 要将 n 个点连接起来至少需要 $n-1$ 条边.

图 G 的**子图**(subgraph)　由 G 的一些边和一些顶点组成，它是 G 的一个部分图，且必须满足：当一条边在子图时，这条边的两个端点也要在子图中.

生成树或支撑树(spanning tree)　若 G 的子图 T 是树，且其顶点集等于 G 的顶点集，则称 T 是 G 的生成树；如图 12.3 中的两个图均是图 12.1 的生成树.

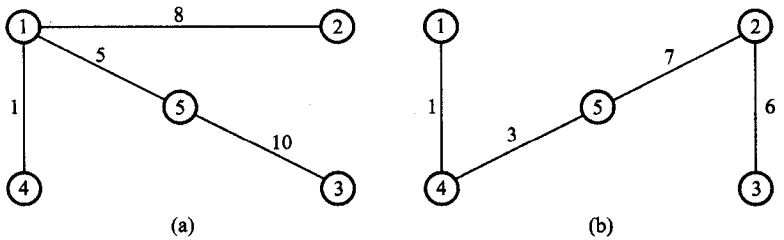


图 12.3

显然，每个生成树对应的线路费用都低于原图对应的线路费用，且生成树中没有多余边，随意去掉其中的一条边都会破坏其连通性.

因此，确定应在哪些站点之间铺设通信线路，就可看作是在相应的加权图中构造最小费用的生成树的问题.

一般地，定义生成树的权为其上所有边权之和.

最小生成树(minimum - weight spanning tree)　在一个加权连通图 G 中，权最小的那棵生成树称为 G 的最小生成树.

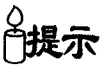
最大生成树(maximum - weight spanning tree)　在一个加权连通图 G 中，权最大的那棵生成树称为 G 的最大生成树.

一个简单连通图只要不是树，其生成树就不惟一，甚至非常多. 如：10 个顶点的完全图，其不同的生成树就有一亿棵. 一般地，n 个顶点的完全图，其不同的生成树个数为 n^{n-2}.

要求出最小生成树，一般不能用穷举法. 30 个顶点的完全图就有 30^{28} 个生成树，30^{28} 有 42 位，即使应用最现代的计算机，在我们有生之年也是无法穷举的，穷举法是无效算法，坏算法. 必须寻求其他的有效算法，这将在下一节介绍.

§12.3 最小生成树算法

在这一节，我们将介绍求最小生成树的两个算法：Prim 算法和 Kruskal 算法，它们都蕴涵了贪婪法的思想.



贪婪法是一种可被用于各种各样问题的处理，它把解看成是由若干个部件构成，每一步求出解的一个部件，但这不是从整体或长远的角度去考虑的，只是局部或当前的最好选择. 求出的一个个部件组合而作为最终的解.

贪婪法只是一种试探法，计算上简便、有效，可提供正确解的一个近似. 但一般情况下，不能保证输出的解是正确的. 其正确性需要证明，这往往比较困难.

12.3.1 Kruskal 算法

假设给定了一个加权连通图 G，G 的边集合为 E，顶点个数为 n. Kruskal 于 1956 年证明了，按下述贪婪法总可得到 G 的一棵最小生成树 T.

Kruskal 算法的粗略描述

为直观，假设 T 中的边和顶点均涂成红色，其余边为白色. 一开始 G 中的边均为白色.

1）将所有顶点涂成红色；

2）在白色边中，挑选一条权最小的边，使其与红色边不形成圈，将该白色边涂红；

3）重复 2）直到有 $n-1$ 条红色边，这 $n-1$ 条红色边便构成最小生成树 T 的边集合.

用上述方法在图上手工操作，求出图 12.1 的最小生成树.

手工操作时，第 2 步中，判断是否形成圈一眼就可看出，但计算机实现就不

是那么直接.

注意到在算法执行过程中,红色顶点和红色边会形成一个或者一个以上的连通分图,它们都是 G 的子树. 一条边与红色边形成圈当且仅当这条边的两个端点属于同一个子树. 因此判定一条边是否与红色边形成圈,只需判断这条边的两端点是否属于同一个子树.

上述判断可这样实现:给每个子树一个不同的编号,对每一个顶点引入一个标记 t,表示这个顶点所在的子树编号. 当加入一条红色边,就会使该边两端点所在的两个子树连接起来,成为一个子树,从而两个子树中的顶点标记要改变成一样. 综上,可将 Kruskal 算法细化使其更容易用计算机实现.

变量说明:

c:生成树的费用;

T:生成树的边集合;

j:迭代次数;

k:记录已经被选入生成树的边数.

Kruskal 算法

输入加权连通图 G 的边权矩阵$[b(i,j)]_{m\times 3}$,顶点数 n.

1) 整理边权矩阵

将$[b(i,j)]_{m\times 3}$按第三行由小到大的次序重新排列,得到新的边权矩阵$[B(i,j)]_{m\times 3}$;

2) 初始化

j←0,T←ϕ,c←0,k←0;对所有 i,t(i)←i.

3) 更新 T,c,t(i)

j←j+1,若 t(B(1,j))=t(B(2,j)),则转 4);否则,若 t(B(1,j))≠t(B(2,j)),则 T←TU(B(1,j),B(2,j)),c←c+B(3,j),k←k+1,对所有 i,若 t(i)=max{t(B(1,j)),t(B(2,j))},则 t(i)←min{t(B(1,j)),t(B(2,j))},

4) 若 k=n-1 或 j=n,则终止,输出 T,c;否则返回 3).

1) Kruskal 算法最终输出的必定是最小生成树,是最优解.

2) Kruskal 算法的时间复杂度为 $O(m\log_2 m)$.

例 12.1 一个编程例子.

借助 MATLAB 软件,用 Kruskal 算法求图 12.1 所示的加权图的最小生成树.

加权图的存储结构采用边权矩阵$[b(i,j)]_{m\times 3}$,编制的 MATLAB 程序

Kruskal.m 如下：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %  Kruskal's algorithm
  b=[1 1 1 2 2 3 3 4;2 4 5 3 5 4 5 5;8 1 5 6 7 9 10 3];
  [B,i]=sortrows(b',3);B=B';%按边权由小到大重新排列矩阵 b 的列
  m=size(b,2);n=5;
  t=1:n;k=0;T=[];c=0;
  for i=1:m
      if t(B(1,i))~=t(B(2,i))%判断第 i 条边是否与树中的边形成圈
          k=k+1;T(k,1:2)=B(1:2,i),c=c+B(3,i)
          tmin=min(t(B(1,i)),t(B(2,i)));
          tmax=max(t(B(1,i)),t(B(2,i)));
          for j=1:n
              if t(j)==tmax
                  t(j)=tmin;
              end
          end
      end
      if k==n-1
          break;
      end
  end
  T,c
```

程序运行结果：

```
  T =
      1    4
      4    5
      2    3
      2    5
  c =
      17
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

因此，图 12.1 的最小生成树的边集合为{(1,4),(4,5),(2,3),(2,5)}，费用为 17.

12.3.2 Prim 算法

Prim 算法也是一种贪婪法，其基本思路是：任选一个顶点 v_1，将其涂红，其

余顶点为白点;在一个端点为红色,另一个端点为白色的边中,找一条权最小的边涂红,把该边的白端点也涂成红色;如此,每次将一条边和一个顶点涂成红色,直到所有顶点都成红色为止. 最终的红色边和顶点便是最小生成树,上面的描述就是最小生成树的逐步生长过程.

做 用上述方法在图上手工操作,求出图 12.1 的最小生成树.

显然,Prim 算法的关键是如何找出连接红点与白点的具有最小权的边. 若把 G 看成完全图,当前有 k 个红点,则有 $k(n-k)$ 条连接红、白点的白边. 从如此多的白边中选取最短边显然费时. 可构造一个较小的候选边集,只要保证最短边在里面即可. 事实上,对每个白点,从该点到各红点的边中,必存在一条最短的白边,只要将所有 $n-k$ 个白点所关联的最短白边作为候选集,就必定能保证所有 $k(n-k)$ 条连接红、白点的白边中最短的白边属于该候选集.

Prim 算法

输入加权图的带权邻接矩阵 $[a(i,j)]_{n\times n}$.

1) 建立初始候选边表,$T\leftarrow\phi$;

2) 从候选边中选取最短边 (u,v),$T\leftarrow T\cup(u,v)$;

3) 调整候选边集;

4) 重复 2),3)直到 T 含有 n - 1 条边.

例 12.2 一个编程例子.

借助 MATLAB 软件,用 Prim 算法求图 12.1 所示的加权图的最小生成树.

加权图的存储结构采用带权邻接矩阵 $[a(i,j)]_{n\times n}$.

MATLAB 程序 Prim.m:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Prim's algorithm
  a =[0     8     inf  1      5;
      8     0       6  inf    7;
      inf   6       0  9     10;
      1     inf     9  0      3;
      5     7      10  3      0];
  T = [];c = 0;v = 1;n = 5;sb = 2:n;% 1 是第一个红点,sb 是白点集
  for j = 2:n         % 构造初始候选边集
      b(1,j - 1) = 1;
      b(2,j - 1) = j;
      b(3,j - 1) = a(1,j);
```

```
    end
    while size(T,2) < n - 1
        [min,i] = min(b(3,:));  % 在候选边集中找最短边
        T(:,size(T,2) + 1) = b(:,i);
        c = c + b(3,i);
        v = b(2,i);                    % v 是新红点
        temp = find(sb == b(2,i));
        sb(temp) = [];b(:,i) = [];
        for j = 1:length(sb)  % 调整候选边集
            d = a(v,b(2,j));
            if d < b(3,j)
                b(1,j) = v;b(3,j) = d;
            end
        end
    end
    T,c
```

程序运行结果:

```
    T =
         1    4    5    2
         4    5    2    3
         1    3    7    6
    c =
         17
```

%%

因此,图 12.1 的最小生成树的边集合为{(1,4),(4,5),(5,2),(2,3)},费用为 17.

注意

1) Prim 算法最终输出的必定是最小生成树,是精确解.

2) Prim 算法的时间复杂度为 $O(n^2)$.

---

§12.4 用最小生成树解决通信网络的优化设计问题

12.4.1 问题

美森公司要为一个客户设计一个有 9 个通信站点的局部网络,使其造价最

低. 这 9 个站的直角坐标为

$a(0,15)$, $b(5,20)$, $c(16,24)$, $d(20,20)$, $e(33,25)$, $f(23,11)$, $g(35,7)$, $h(25,0)$, $i(10,3)$.

两通信站之间的线路费用正比于两点间的直角折线距离,即

$$d = |x_1 - x_2| + |y_1 - y_2|.$$

两点间的线路费用正比于线路的长度. 在不允许通信线在非站点处连接的条件下,如何布线使连接通信站的线网费用最低.

12.4.2 模型的建立和求解

以 9 个站点为顶点作一个完全图 K_9,每条边的边权取为对应两点之间的直角折线距离,得到一个加权图 G. 由于不允许通信线在非站点处连接,因此,G 的最小生成树便是造价最低的连接 9 个站点的线网.

1. 求出任意两点间的直角折线距离矩阵 D

编写M文件 distance. m 如下:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x = [0,5,16,20,33,23,35,25,10];
    y = [15,20,24,20,25,11,7,0,3];
    for i = 1:9
        for j = 1:9
            D(i,j) = abs(x(i) - x(j)) + abs(y(i) - y(j));
        end
    end
    D
    运行该程序得到
    D =
         0  10  25  25  43  27  43  40  22
        10   0  15  15  33  27  43  40  22
        25  15   0   8  18  20  36  33  27
        25  15   8   0  18  12  28  25  27
        43  33  18  18   0  24  20  33  45
        27  27  20  12  24   0  16  13  21
        43  43  36  28  20  16   0  17  29
        40  40  33  25  33  13  17   0  18
        22  22  27  27  45  21  29  18   0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2. 求最小生成树

编写M文件 example. m 如下：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = [ ];c =0;v =1;n =9;sb =2:n;
for j =2:n
        b(1,j -1) =1;
        b(2,j -1) =j;
        b(3,j -1) =D(1,j);
end
while size(T,2) <n -1
        [min,i] =min(b(3,:));
        T(:,size(T,2) +1) =b(:,i);
        c =c +b(3,i);
        v =b(2,i);
        temp =find(sb ==b(2,i));
        sb(temp) =[ ];b(:,i) =[ ];
        for j =1:length(sb)
            d =D(v,b(2,j));
            if d <b(3,j)
                b(1,j) =v;b(3,j) =d;
            end
        end
end
        T,c
        运行该程序,得到
        T =
             1    2   3    4    6    6    3    8
             2    3   4    6    8    7    5    9
            10   15   8   12   13   16   18   18
        c =
           110
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

因此,最小生成树的边集合为{ab, bc, cd, df, fh, fg, ce, hi},线路总长度为110. 作出最小生成树如图 12.4 所示.

在不允许通信线在非站点处连接的条件下,沿着站点 a 到 b,a 到 b,b 到 c,c 到 d,d 到 f,f 到 h,f 到 g,c 到 e,h 到 i 布线,可使连接通信站的线网总长度最短,结果为 110,从而费用最低.

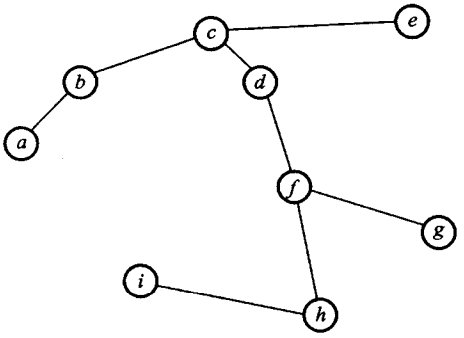


图 12.4

§12.5 怎样使线网费用进一步降低

12.5.1 问题分析及模型

给定 n 个通信站点，用通信线把这些站点联结起来，允许通信线在非站点处连接，如何连接，可使连接通信站的线网费用最低？这个问题不同于前面的最小生成树问题，在那里，不允许通信线在非站点处连接，这一限制使问题变得简单. 在这里取消了这个限制，而允许通信线在站点以外的点（即“虚设站”或 Steiner 点）连接，这样可以使线网费用进一步降低，但问题要复杂得多.

例如，有三个通信站，直角坐标分别为 $a(0,0)$，$b(4,3)$，$c(6,0)$. 两点间的距离为直角折线距离，以这三个站为顶点，距离为边权的加权完全图，见图 12.5.

若不允许通信线在非站点处连接，则图 12.5 的最小生成树即代表最小费用的线网，其长度为 11. 但若允许通信线在非站点处连接，即可引入“虚设站”，则“虚设站”的个数和位置将是解决问题的关键. 若在 $(4,0)$ 处设置一个“虚设站” d，则 a,b,c,d 四个点的完全图的最小生成树是图 12.6，其长度为 9. 小于不加“虚设站”时的长度 11.

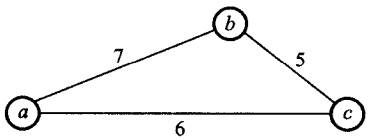


图 12.5

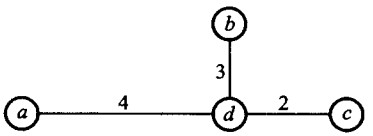


图 12.6

通过加入若干“虚设站”后，构造出由原站点和虚设站生成的最小 Steiner 树，若虚设站设置得恰当，就可降低由原站点生成的最小生成树所需的费用. 用这种方法可降低费用多达 13.4%. 因此，要求费用最少的连接方式，转化为构造最低费用的 Steiner 树，即最小 Steiner 树的问题.

1）Steiner 树允许线路在通信站点以外连接，这种连接点即为虚设站.

2）为构造一个有 n 个站的网络，最小 Steiner 树最多只需 $n-2$ 个虚设站，这些虚设站称为 Steiner 点. Steiner 点位于给定通信站点的 x 坐标线，y 坐标线形成的格点上.

12.5.2 最小 Steiner 树的求解思路

“虚设站”（即 Steiner 点）的个数和位置是解决问题的关键，根据提示“Steiner 点位于给定通信站点的 x 坐标线，y 坐标线形成的格点上”，可知最多有 $n^2-n=n(n-1)$ 个 Steiner 点的可能位置，这些位置就是 Steiner 点的候选点. 当 $n=9$ 时，有 $n(n-1)=72$ 个 Steiner 点的可能位置.

V_0：给定的 n 个通信站点的集合；

V_p：Steiner 点的候选点集合，设其点数为 p，$V_p \cap V_0 = \varnothing$；

以 $V=V_p \cup V_0$ 为顶点集作一个加权完全图 K_{p+n}，其中的边 (u,v) 的权取为点 u 与 v 之间的直角折线距离. 我们的问题就是：求加权完全图 K_{p+n} 中包含 V_0（也允许包含 V 中的其他点）的权最小的子树. 此即求加权完全图 K_{p+n} 中，V_0 的最小 Steiner 树问题.

求 V_0 的最小 Steiner 树可分解为两个问题：

1）求 Steiner 点；2）求最小生成树.

根据提示“最小 Steiner 树最多只需 $n-2$ 个虚设站（Steiner 点）”，

V_s：表示 V_p 中任意 s 个点的集合.

对满足 $0 \leqslant s \leqslant n-2$ 的整数 s 和点集 $V_s \subseteq V_p$，以 $V=V_s \cup V_0$ 为顶点集的加权完全图 K_{s+n} 的最小生成树记为 T_{V_s}，所有 T_{V_s} 中权最小者记为 T^*，T^* 即为所要求的最小 Steiner 树.

1）所有的 T_{V_s} 共有多少？当 $n=9$ 时，依次用求最小生成树的算法求出一个个的 T_{V_s} 是否可行？如何解决该问题？比如，穷举法，贪婪法.

2）当 $n=9$ 时，有 $n(n-1)=72$ 个 Steiner 点的可能位置，72 可否再减少？

12.5.3 最小 Steiner 树的求解算法设计

求最小 Steiner 树问题是 NP 难题，点数较小的问题可用穷举法，但若规模较大，应寻求近似算法.

1. 穷举法

由于费用最少的 Steiner 树 T^* 上最多只需引入 $n-2$ 个虚设点，因此可从 $m \leqslant n(n-1)$ 个可能的 Steiner 点位置中任取 s 个点，$s=0,1,2,\cdots,n-2$，连同给定的 n 个点一起，用 Kruskal 算法，求由这 $n+s$ 个点确定的赋权完全图（图中边权取为两点间的直角折线距离）的最小生成树 T_s. 由于从 m 个点中任取 s 个点的取法有 $\binom{m}{s}$，因此共有 $\binom{m}{0}+\binom{m}{1}+\cdots+\binom{m}{s}$ 个可能的 Steiner 点集. 每次迭代，对每个可能的 Steiner 点集（s 个顶点）连同给定的 n 个点一起确定的赋权完全图，用 Kruskal 算法求该完全图的最小生成树需用多项式时间，共需进行 $\binom{m}{0}+\binom{m}{1}+\cdots+\binom{m}{s}$ 次迭代. 若 m 不大，此法可行，否则若 m 大，此法将无效.

提示

n 个通信站点所在的最小长方形区域的四个拐角区域不可能有 Steiner 点，即：设 $V_0=\{v_i(x_i,y_i)\mid i=1,2,\cdots,n\}$，

对每个 y_k，$k=1,2,\cdots,n$，记

$$x_{00}(k)=\min_{(x_i,y_i)\in V_0,y_i<y_k}\{x_i\},\quad x_{01}(k)=\min_{(x_i,y_i)\in V_0,y_i>y_k}\{x_i\},$$

$$x_{10}(k)=\max_{(x_i,y_i)\in V_0,y_i<y_k}\{x_i\},\quad x_{11}(k)=\max_{(x_i,y_i)\in V_0,y_i>y_k}\{x_i\}.$$

想

则在下述四类区域中不含 Steiner 点

$$D_1=\{(x,y)\mid x<x_{00}(k),y<y_k\},D_2=\{(x,y)\mid x<x_{01}(k),y>y_k\},$$

$$D_3=\{(x,y)\mid x>x_{10}(k),y<y_k\},D_4=\{(x,y)\mid x>x_{11}(k),y>y_k\}.$$

如图 12.7，星号点是给定的 9 个通信站点. 共有 $n(n-1)=72$ 个 Steiner 点的可能位置，它们位于过 9 个点的水平线与垂直线的交点上，且由于区域 D_1，D_2，D_3 和 D_4 内不含 Steiner 点，72 个可能的 Steiner 点位置可减少到 31 个（图12.7

中小圆圈所示的 31 个位置). $m=31$, 迭代次数减少到 $\binom{31}{0}+\binom{31}{1}+\cdots+\binom{31}{7}=3\ 572\ 224$ 次. 假设每次迭代只需用 1/60 s 的时间, 3 572 224 次迭代需要大约 17 h.

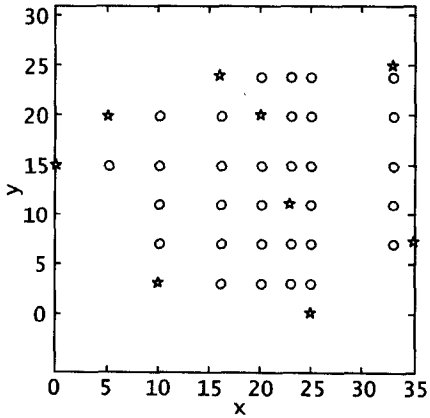


图 12.7　9 个给定点和 31 个可能的 Steiner 点

2. 贪婪试探算法

图 12.5 的最小生成树如图 12.8(a), 顶点按直角坐标之定位放置, 图 12.8(b)是把边画成直角折线, 该图称为其三个点的最小直角折线支撑树. 若将图 12.8(b)图中重边的端点 d 作为虚设站加入, 则 4 个点的最小生成树的权减少了 2.

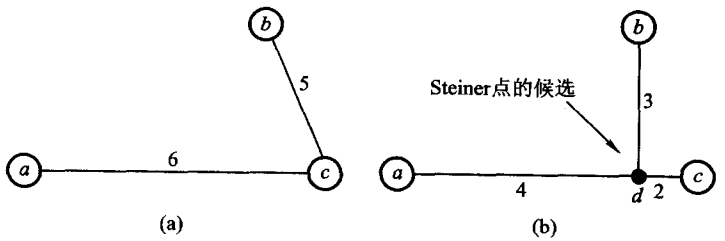


图 12.8

因此, 一般情况下, 可把最小直角折线支撑树中重边的端点作为 Steiner 点的候选点. 结合贪婪法的思想, 可构造出如下的试探法, 能否得到正确解, 需要证明.

1）输入给定的 n 个通信站点的坐标；

2）计算最小直角折线支撑树；

3）找重边，则重边的端点便是 Steiner 点的候选点；

4）分别计算出每个候选点作为 Steiner 点加入后所减少的费用，该费用称为此点的价值；

5）把最大价值的候选点也作为一个给定点，重复 2）到 5）直到没有正价值的候选点.

编写上面介绍的贪婪试探算法的 MATLAB 程序，求给定的 9 个通信站的最小 Steiner 树.

3. 改进型试探算法

如果每次迭代，都按照随意的顺序加入“虚设站”，并使得到的最小生成树费用有所减少，直到已加入 $n-2$ 个“虚设站”，或加入任何一个剩余的可能的 Steiner 点都不能使费用减少为止. 按步骤描述如下：

1）求给定的 n 个点的最小生成树，记录其费用；

2）取一个可能的 Steiner 点加入，求最小生成树，若该树的费用小于当前的费用，则记录此树并更新费用；

3）重复 2）直到已有 $n-2$ 个 Steiner 点，或任何剩余的 Steiner 点加入都不能减少费用.

编写上面介绍的改进型试探算法的 MATLAB 程序，求给定的 9 个通信站的最小 Steiner 树.

贪婪试探算法和改进型试探算法都是近似算法，对一般的问题未必能得到最优解. 对本问题给定的 9 个通信站的情况，求出的解是否为最小 Steiner 树，需要证明，否则，应分析解的近似程度.

4. 模拟退火法

这是一种通用的随机搜索法，是解决 NPH 问题比较有效的方法.

1）给定点集连同一些虚设点一起构成点集 Z，求 Z 的最小支撑树，其费用记为 C，置 $k=0$；

2）产生新的点集 S.

从以下几种方式中随机选择一种：

- 加入一个新的虚设点
- 去掉一个存在的虚设点
- 移动一个现有的虚设点到一个随机的允许位置

3）确定新点集 S 的最小支撑树，其费用记为 C_1，

若 $C_1 \leqslant C$，则更新 C 为 C_1，更新当前点集 Z 为 S，当 $k = M$ 时停止，否则 $k = k + 1$，转 2）；

若 $C_1 > C$，则仅以一定的概率（可取为 $\exp\{-(C_1 - C)/T(k)\}$，其中 T 为一控制参数，称为温度，随 k 的增大而减小，比如取 $T(k) = T(0)/k$，称为冷却方案）接受 S 作为当前点集 Z，转 2）.

用模拟退火法来求图 12.7 所给出的 9 个点的最小 Steiner 树，在 25 MHz 型 386 计算机上运行，大约 1.5 min 便得到了最优解. 对应于随机数产生器的不同种子的不同运行，可给出全部五个不同的最优解. 在上百次运行中，模拟退火法都总是收敛到五个最优解中的一个. 计算时间方面的优越性表明当该问题的规模较大，穷举法不可行时，模拟退火法的价值.

5. 修正的 Prim 启发式算法

受到求最小生成树的 Prim 算法的启发，根据 Prim 算法的思想，构造出下面的求最小直角折线 Steiner 树的方法.

先引入一些记号：

Z：给定的通信站点集合；

G：给定通信站点的 x 坐标线，y 坐标线形成的格点全体构成的集合；

T：当前 Steiner 树的顶点集；

$S = G - Z$.

算法步骤

1）选取 Z 中距离最近的两点 $z_i = (x_i, y_i)$，$z_j = (x_j, y_j)$；

2）这两点的 x 坐标或 y 坐标相同，则将两点连接起来，并把该路径上所有在 G 中的点以及 z_i，z_j 加入 T，否则

a. 构造过 (x_i, y_j) 的连接 z_i，z_j 的直角折线路径 path_1，将 path_1 上所有属于 G 的点以及 z_i，z_j 加入 T；

b. 在 $Z - T$ 中找到与当前树距离最近的顶点 z，其距离记为 dist1，然后删掉树中的 path_1；

c. 构造过 (x_i, y_j) 的连接 z_i，z_j 的直角折线路径 path_1，将 path_1 上所有属于 G 的点以及 z_i，z_j 加入 T；

d. 在 $Z - T$ 中找到与当前树距离最近的顶点 z，其距离记为 dist2，然后删掉

树中的 $path_2$；

e. 若 dist1 < dist2，则加入 $path_1$，

若 dist2 < dist1，则加入 $path_2$，

若 dist1 = dist2，则对下一个最近点重复 2) a 到 2) e，直到 dist1 ≠ dist2，或穷尽了 Z 中所有顶点（此时任意选择）；

3) 取 $z_i \in Z \cap (G-T)$，$z_j \in T$，使 z_i，z_j 尽可能近；

4) 重复 2)，3) 直到 Z 中的顶点均在 T 中.

用手工操作的方式，用修正的 Prim 启发式算法求给定的 9 个通信站的最小 Steiner 树，体会该算法的思想.

最小 Steiner 树问题是 NP 难题. 前面介绍的方法，除穷举法之外，每个算法都是有效算法，但不一定能得到最优解，一般需要对解的近似程度进行分析. 对算法的好坏，可随机给出一些通信站，用这些方法来求解，对算法给出的解进行比较，看哪个算法效果最佳.

12.5.4 结果

用前述方法可得到给定的 9 个通信站的最小 Steiner 树，共有 5 个，费用为 94. 图 12.9 给出了这 5 个最小 Steiner 树，其中每个 Steiner 树都含 4 个或 5 个 Steiner 点. 因此可按图 12.9 中任何一个 Steiner 树来设计给定 9 个站的费用最少的局部网络. 例如，在 5 个位置 $j(16,20)$，$k(25,20)$，$l(25,11)$，$m(25,7)$，$n(25,3)$ 处各建立一个虚拟站点，与原来的 9 个站点共 14 个站点的最小生成树的边集合为 $\{ab,bj,jc,jd,dk,kl,lf,lm,mn,nh,mg,ke,ni\}$，沿着站点 a 到 b；b 到 j，j 到 c，j 到 d，d 到 k，k 到 l，l 到 f，l 到 m，m 到 n，n 到 h，m 到 g，k 到 e，n 到 i 布线，可使线网最短，长度为 94（见图 12.9(d)）. 或在 4 个位置 $j(16,20)$，$o(23,20)$，$m(25,7)$，$n(25,3)$ 处各建立一个虚拟站点，与原来的 9 个站点共 13 个站点的最小生成树的边集合为 $\{ab,bj,jc,jd,do,of,fm,mn,nh,mg,oe,ni\}$，沿着站点 a 到 b，b 到 j，j 到 c，j 到 d，d 到 o，o 到 f，f 到 m，m 到 n，n 到 h，m 到 g，o 到 e，n 到 i 布线，可使线网最短，长度为 94（见图 12.9(e)）.

与计算机网络设计相类似的问题还有许多，如：有线电视网、电线网、石油传输管网以及其他的管道网、铁路、公路的建设等等.

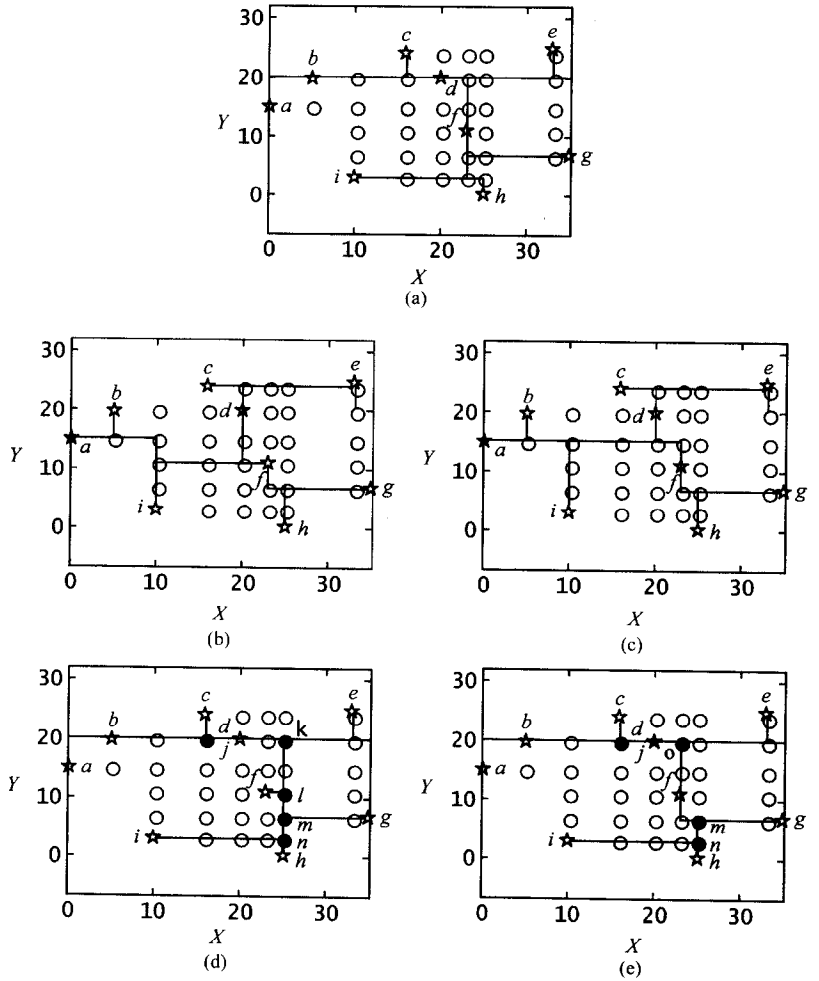


图 12.9

§12.6 操　　练

操练　如何设计海底管道网

某石油公司在墨西哥海湾拥有几个石油钻井平台，每个平台开采出的石油需要运往路易斯安娜的炼油厂. 要在平台与路易斯安娜海岸之间建造一个管道网，使石油通过管道传输. 这管道网该如何设计，才能使建造费用最低.

图 12.10 中,顶点代表钻井平台及炼油厂,边表示其两端点之间可以铺设管道,边上的权代表该段管道的建设费用.

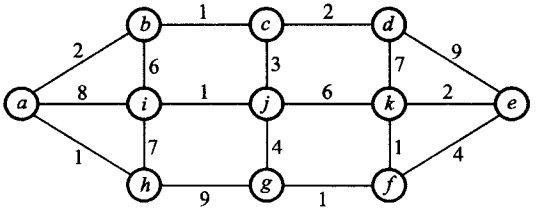


图 12.10

更多的相关信息资源

1 http://www.cs.brown.edu/publications/jgaa/

2 T. McGrath, M. Menzies and C. Smith. Iterative and construtive models for minimal rectilinear Steiner trees. The UMAP Journal, Vol. 12, No. 3, 1991. pp. 265 – 278

3 P. J. Melody, H. L. Moore and M. Wood. The construction of a minimal – cost Steiner tree. The UMAP Journal, Vol. 12, No. 3, 1991. pp. 19 – 36

4 王学辉等. MATLAB6.1 最新应用详解. 北京:中国水利水电出版社,2002

第 13 章

如何实现汽车自主导航

——最短路径

面对一个问题，起初找到一种解决方法，人们便会欣喜. 但渐渐地可能会发现这种方法对于大规模的问题简直无能为力，人们便开始找寻有效的算法，并不断地追求更加快速的算法. 然而大千世界还有大量问题是属于非常困难的问题类，即 NP 难题，对于这类问题科学家们倾向于认为它们不存在有效算法，要想寻求其精确的有效算法多半会徒劳无功. 当然，在图论领域既有许多 NP 难题，也有不少已设计出有效算法的问题，如最小生成树问题和最短路径问题.

——作者

§13.1 卫星定位汽车自动导航系统

当你在一个陌生的道路环境中驾车，面对错综复杂的交通网，多么渴望有人能告诉你所在的位置、指引到达目的地应走的正确路线，告诉你最近的餐馆在哪里，最近的加油站在哪里，最近的取款机在哪里等等. 汽车自动导航系统就能为你提供这些服务.

目前大多数汽车导航产品包括这样几种类型，如手机一般大小的手持式和手表式导航仪，大多以经纬度坐标的形式表示当前的位置，非专业人员阅读有一定难度. 另一种掌上电脑式导航仪，体积小巧、灵便，但显示地图范围很小，装载的地图有限. 国际上较为流行的汽车导航产品是车载式的. 许多进口及国产中高档轿车中原本都装有这一设备，它是固定在车上的，通过一张光盘，在屏幕上以一张巨大的电子地图供你使用. 比如你是一个驾车旅行者，从北京出发，不管是北上还是南下，导航仪的地图都能自动切换，可以是蒙古，也可以是广东，你可以不熟悉任何路段，导航仪为你随时指示所在位置，协助驾车者在陌生的道路环境中，通过电子地图和语音指南，准确地掌握前往目的地的路线. 实现了真正的自主导航.

车载自动导航系统是依赖 GPS 系统与车载 GPS 接收机监测车辆当前位置，

并将数据跟用户自定义的目的地比较、参照电子地图计算行驶路线，并实时将信息提供给驾车者.

GPS 系统的全称是 Global Positioning System，即全球定位系统. 它是美国政府在冷战期间继阿波罗登月计划、航天飞机计划之后的第三项重点空间计划. 从 1973 年开始至 1994 年的 20 年间，耗费巨资达 120 亿美元，共发射了 24 颗定位卫星，建立了完整的环绕全球的网络. 该系统是一套能够实时、全天候、为全球范围内的陆地、海上、空中的各类用户目标提供连续实时的三维定位、三维速度及精确时间信息的系统. 美国出于经济利益及各方面的考虑，已经确立 GPS 系统军民共用、世界共享的政策，GPS 技术已成为一种稳定、先进的资源. 随着不久后中国、欧洲的伽利略卫星系统发射完成并投入应用，卫星导航将步入一个竞争更加激烈的、基于不同卫星平台的导航时代.

随着全球定位系统（简称 GPS）、地理信息系统（Geographical Information System，简称 GIS）与遥感技术（Remote Sensing，简称 RS）的发展与结合，促进了现代空间数据快速获取的集成技术、计算机技术与通信技术的发展与结合，人们不断从更广泛的领域中涉足导航系统和导航电子地图的研究，特别是在美、日等一些发达国家，已逐步将多维智能交通系统变成现实，形成了一个全球交通事业深远和跨时代的革命，汽车自动导航系统的使用已非常普及. 我国从 20 世纪 90 年代开始这方面的研究工作，已研制出具有我国自主知识产权和结合我国实际的汽车自动导航系统，并已投放市场.

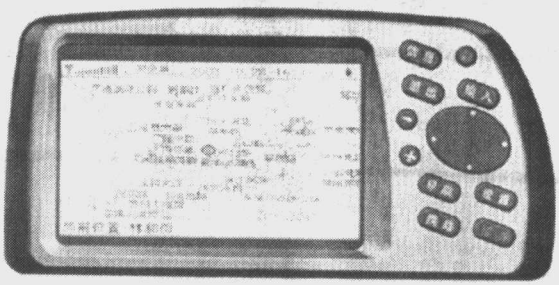


图 13.1　汽车导航仪显示屏

汽车自动导航系统（如图 13.1）的工作过程包括：定义目标数据信息和显示电子地图.

如何定义目标数据信息呢？驾车者可以在系统显示的电子地图上直接选取目标地点，或将目的地名称输入到系统中. 根据输入设备的不同，有不同的地名输入方法，依靠语音、键盘或触摸屏可实现几乎全部的操纵功能.

如何显示电子地图呢？存储在光盘或内置存储器（如硬盘）中的电子地图，

是汽车导航系统中至关重要的一部分. 电子地图的信息库包括地理、道路和交通管制等信息,并与地点对应存储了相关的经纬度信息. 当主机从 GPS 接收机得到经过筛选计算确定的当前点经纬度数值,然后通过与电子地图数据的对照,确定车辆当前所在的地点. 一般来说,导航系统会将定义目的地时车辆的当前位置默认为出发点,当目的地确定后,导航系统根据电子地图上存储的地图信息,自动计算出一条最佳的推荐路线. 有些系统中,用户还可以指定途中希望经过的地点,或者定义一定的路线选择逻辑(如不允许经过高速公路、按照行驶路线最短的原则等). 推荐的路线将以醒目的方式显示在屏幕地图中,同时屏幕上即时显示出车辆的当前位置,以作为参照. 如果行驶过程中车辆偏离了推荐的路线,系统会自动更改原有路线并以车辆当前点为出发点重新计算最佳路线,并将修正后的路线作为新的推荐路线.

§13.2 汽车导航系统如何为你选择最佳路线

假设你正在北京旅行,驾车要从天安门广场东侧去往目的地天坛公园北门(如图 13.2 所示),汽车导航仪会自动确定车辆所在的位置,以此作为起点,你只需告诉它目的地,它便可显示出涵盖起点和终点的道路网络图.

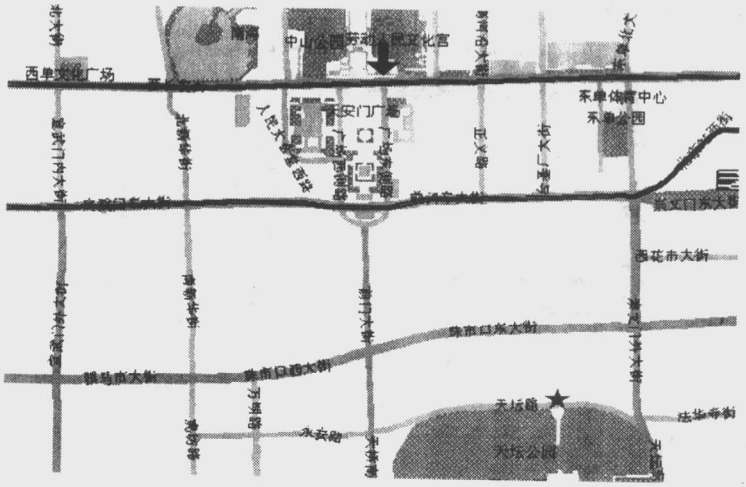


图 13.2 道路网(箭头所指为车辆当前所在位置,五角星处为目的地)

系统如何为你计算距离最短的路线呢? 在地理信息的基础上,以道路网中的道路交叉点和路线的起点与终点为顶点,两交叉点之间的道路为边作线图,道

路长度(可以由系统存放的地理信息得到)为边权,图 13.2 中的道路网对应的加权图如图 13.3 所示. 其中 5 号顶点和 25 号顶点分别为起点和终点,原问题就转化为求图 13.3 中的加权图从顶点 5 到顶点 25 的最短路径. 这可以采用图论中求最短路径的算法得到.

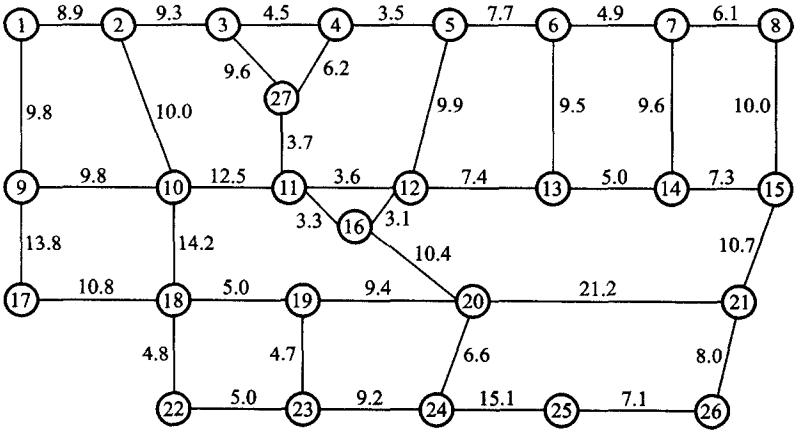


图 13.3　对应于图 13.2 中道路网的加权图

§13.3　最短路径问题和算法的类型

给定一个加权图 G,每条边上都有一个数字代表边的长度. 在实际问题中这个长度可以代表费用、时间、可靠度或其他性能指标.

普通长度(ordinary path length)　路径长度定义为该路径所包含的全体边的长度之和. 对图中任意给定的两点 u,v,在它们之间可能存在多条路径.

普通型最短路径问题(ordinary shortest-path problem)　求从 u 到 v 的路径中普通长度最短的路径,该路径称为从 u 到 v 的**最短路径**(shortest-path).

按路径长度的不同定义可将最短路径问题分为两大类:普通路径长度和一般路径长度. 后者是指路径权被定义为其上边权的其他函数,如路径的权为其包含的所有边权之积,边权的最大值或其他更复杂的函数. 再如,在交通网络中,在道路的交叉口转弯时,可能会增加一个"转弯罚数"(turn penalty). 详细的分类见表 13.1.

我们先介绍求解具有普通路径长度的最短路径问题的算法.

表 13.1　最短路径问题的分类

一、普通路径长度
- A. 无约束
  - 1）最短路径
    - a. 两个指定顶点间的最短路径
    - b. 一个指定顶点到其余各顶点的最短路径
    - c. 任意两顶点间的最短路径
  - 2）第二，第三，…，第 k 短路径
- B. 带约束
  - 1）包含一些指定顶点的最短路径
  - 2）包含一些指定边的最短路径

二、一般路径长度
- A. 带转弯罚数
- B. 路径权为其上边权的其他函数形式，如边权之积

§13.4　最短路径算法

图论问题的求解与数值问题的求解（如方程式求根，插值计算，数值积分和函数逼近等）有很大的不同，前者是“非数值性问题”，涉及的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式来描述. 解决此类问题的关键已不再是分析数学和计算方法，而是能设计出合适的数据结构. 所谓数据结构是指数据（信息的载体，能够被计算机识别、存储和加工处理）之间的相互关系. 在这一章，我们将从粗略描述开始，逐步将精细化的算法设计过程展示出来，并给出经调试通过的 MATLAB 程序. 在这些程序中，你将会注意到，主要是进行判断和比较，而不是进行算术运算.

寻求从一固定起点 v_0 到其余各点的最短路径的最有效算法之一是 Dijkstra 算法，它是一种迭代算法. 为叙述方便，我们把从起点 v_0 到顶点 v 的最短路径简称为 v 的最短路径.

要求　加权图中无负权.

出发点　最短路径上的任何子段仍是最短路径，距 v_0 远的顶点的最短路径必经过距 v_0 近的顶点. 因此可按与 v_0 的距离由近及远地逐个求出各顶点的最短路径和长度.

算法思路　设置一个集合 S,存放已求出其最短路径长度的顶点.

1) $S \leftarrow \{v_0\}$;

2) 求出 $\overline{S} = V - S$ 中与 v_0 距离最近的顶点 u,将 u 加入到 S 中;

3) 重复 2) 直到 $\overline{S} = \varnothing$,

其中 $\varnothing$ 表示空集合.

例 13.1　一个简单例子

求图 13.4 中从顶点 1 到顶点 6 的最短路径及其长度.

以 1 号顶点为起点,首先,$S = \{1\}$,与 1 号顶点距离最近的顶点为 4 号顶点,将其加入到 S 中,$S = \{1,4\}$,1 到 4 的最短路径已求出,见图 13.5 中的粗线所示,并在顶点 4 旁边标上该路径的长度,S 中的顶点在图中为实心点. 在 $\overline{S}$ 中与顶点 1 距离最近的顶点一定是 S 的邻点 2,5,3 或 6,经过比较得到,顶点 1 到顶点 2 的距离最近为 6,其最短路径在图 13.6 中用粗线表示,将顶点 2 加入到 S 中,$S = \{1,4,2\}$.

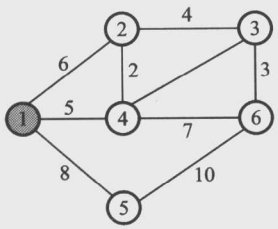


图 13.4

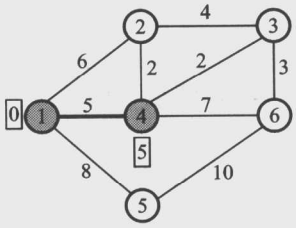


图 13.5

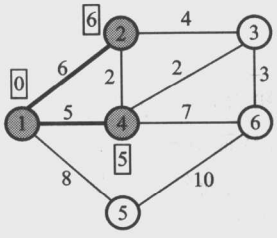


图 13.6

此时,在 $\overline{S}$ 中与顶点 1 最近的顶点为 3,经顶点 4 到达 3,距离为 5 + 2 = 7,已求出的最短路径在图 13.7 中用粗线表示,顶点 1 到各顶点的最短距离标在顶点的旁边,此时 $S = \{1,4,2,3\}$. 与前面类似,下一个离顶点 1 最近的顶点为顶点 5,接着便是顶点 6,其最短路径和距离如图 13.8 所示,该图的粗线是一棵树,树上任意两点间有惟一路径,这些路径均为最短路径. 该树称为最短路径树.

在这个简单例子里,第 2 步,找出 $\overline{S}$ 中离顶点 1 最近的顶点用手工操作较容易,计算机如何才能实现呢?

为直观,想像把集合 S 中的顶点涂成红色,$\overline{S}$ 中的顶点为白色. 如何在白点集 $\overline{S}$ 中找出最短路径长度最小的顶点 u,加入到红点集 S 呢?

对于图中每个顶点 v,引入一个标记 $l(v)$ 来记录从 v_0 到 v 的,且中间只经过红点,不经过白点的路径中的最短路径长度(当从 v_0 出发,经过红点集 S 中的顶点不能到达 v 时,$l(v)$ 取 ∞).

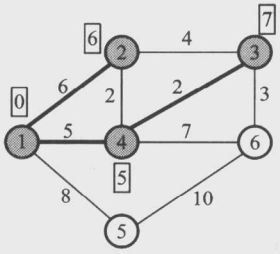


图 13.7

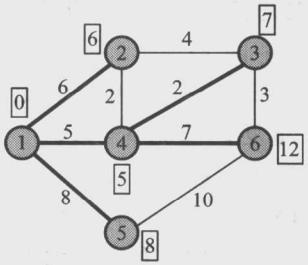


图 13.8

1）当 $v\in S$ 时，$l(v)$ 是 v_0 到 v 的最短路径长度；

2）当 $v\in\overline{S}$ 时，$l(v)$ 不小于 v_0 到 v 的最短路径长度；

3）若 $l(u)=\min\limits_{v\in\overline{S}}\{l(v)\}$，则 u 是 $\overline{S}$ 中距离 v_0 最近的顶点，且 $l(v)$ 是 u 的最短路径长度.

上述结论成立吗？为什么？

最初 $l(v_0)=0$，$\forall v\neq v_0$，$l(v)=\infty$，标记最小的顶点为 $u=v_0$，将其涂红加入到 S 中，从而 $S=\{v_0\}$.

当新红点 u 加入 S 后，S 改变，红点的标记不会改变，白点 v 的标记将怎样变化呢？

从起点 v_0 出发，中间只经红点到 v 的最短路径只可能是如下两种之一：

1）v 的前一个点为老红点；

2）v 的前一个点为新红点.

第一种情形 v 的最短路径长度为 $l(v)$，第二种情形 v 的最短路径长度为 $l(u)+w(u,v)$（其中 $w(u,v)$ 为边 (u,v) 的权）. 因此 $l(v)$ 应更改为

$$\min\{l(v),l(u)+w(u,v)\}.$$

上述算法只求出了最短路径的长度，如要求出最短路径，还需记下路径. 为此，对每个顶点 v，引入一个父亲点 $f(v)$，记录在 v 的只经过红点的最短路径上，v 的前一个顶点. 与 $l(v)$ 一样，$f(v)$ 将随着 S 的变化而不断更新. $f(v)$ 最终的取值就可以确定从起点 v_0 出发到其余各点的最短路径. 怎么确定？为什么？

因此，算法可进一步细化为 Dijkstra 算法：

输入加权图的带权邻接矩阵 $\boldsymbol{w}=[w(v_i,v_j)]_{n\times n}$，所求路径的起点为 v_0，

1）初始化

令 $u=v_0, S=\{v_0\}, l(v_0)=0, \forall v\neq v_0, l(v)=\infty$；

2）更新 $l(v), f(v)$

对所有不在 S 中的顶点 v，若 $l(v)>l(u)+w(u,v)$，则更新 $l(v), f(v)$，即 $l(v)\leftarrow l(u)+w(u,v), f(v)\leftarrow u$；

3）寻找不在 S 中的顶点 u，使 $l(u)$ 为最小，把 u 加入到 S 中；

4）重复 2），3）直到所有顶点都在 S 中为止.

$l(v)$ 最终的值便是 v_0 到 v 的最短路径的长度，最短路径可由最终的 $f(v)$ 求出. 在 v_0 到 v 的最短路径上，v 的前一个顶点为 $f(v)$，$f(v)$ 的前一个顶点为 $f[f(v)]$，这过程继续直到追踪到 v_0 为止，这样便可得到 v_0 到 v 的最短路径.

例 13.2 一个计算例子

用 Dijkstra 算法求图 13.9 从 1 号顶点到 5 号顶点的最短路径.

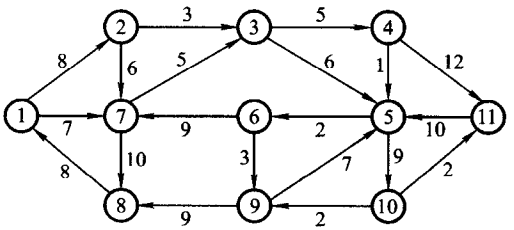


图 13.9

我们将各次迭代中，每个顶点标记 $l(v)$、各顶点的父亲点 $f(v)$ 的值以及新加入到 S 中的当前顶点 u 在表 13.2 中给出. S 中顶点的标记 $l(v)$ 和父亲点 $f(v)$ 用方框“□”框着，“□”内的数值在其后的迭代中不会改变. 到第五次迭代，5 号顶点已在 S 中，因此，$l(5)=17$ 便是其最短路径的长度. 该最短路径可由父子关系追踪而得. 由 $f(5)=3$ 知，5 的前一个点为 3；由 $f(3)=2$ 知，3 的前一个点为 2；由 $f(2)=1$ 知，2 的前一个点为 1. 因此得到 5—3—2—1，倒过来就是 1 到 5 的最短路径.

表 13.2

迭代	$l(v)$ $(f(v))$											u
	1	2	3	4	5	6	7	8	9	10	11	
0	[0]	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1
1	[0]	8(1)	∞	∞	∞	∞	[7(1)]	∞	∞	∞	∞	7
2	[0]	[8(1)]	12(7)	∞	∞	∞	[7(1)]	17(7)	∞	∞	∞	2
3	[0]	[8(1)]	[11(2)]	∞	∞	∞	[7(1)]	17(7)	∞	∞	∞	3
4	[0]	[8(1)]	[11(2)]	[16(3)]	17(3)	∞	[7(1)]	17(7)	∞	∞	∞	4
5	[0]	[8(1)]	[11(2)]	[16(3)]	[17(3)]	∞	[7(1)]	17(7)	∞	∞	28(4)	5
	……											

注意

1）若最终某顶点 v 的标记为 ∞，且 v 在 S 中，则表明从 v_0 到 v 没有路径，即从 v_0 出发不能到达 v.

2）Dijkstra 算法的运算时间是 $O(n^2)$ 级的. 该法既适合于有向图又适合于无向图，只是要求所有权都必须非负.

想一想

1）如果想要求出两点间的全体最短路径应怎么办？当最短路径被占用或出现故障时，就必须改用次短，再次短的路径，如何求出第 k 短的最短路径？

2）如果带有负权，如何求最短路径？

§13.5 Dijkstra 算法的 MATLAB 程序

我们首先利用 Dijkstra 算法编写一个 MATLAB 函数 M 文件 Dijkstra.m 求加权图（有向图、无向图均可）两顶点间最短路径，要求图中没有负权. 取输入变量为带权邻接矩阵 weight、起点 start 和终点 terminal，输出变量为起点到终点的最短路径 path 及其距离 distance.

用 Dijkstra 算法求从起点 start 和终点 terminal 的最短路径的 M 文件 Dijkstra.m 为

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    function [distance,path] = Dijkstra(weight,start,terminal)
    % ------为顶点标记 l,父亲点 f,当前顶点 u 和集合 S 赋初值------
    n = size(weight,1);l(start) = 0;f(start) = start;
    for i = 1:n
        if i ~ = start
            l(i) = inf;
    end,end
    S(1) = start;u = start;
    while length(S) < n
    % ------对所有不在 S 中的顶点 v,当 l(v) > l(u) + w(u,v)时,更新 l(v),f(v)------
        for i = 1:n
          ins = 0;
          for j = 1:length(S)
            if i == S(j)
                ins = 1;
```

```
        end,end
        if ins==0
          v=i;
          if l(v)>(l(u)+weight(u,v))
            l(v)=(l(u)+weight(u,v));f(v)=u;
    end,end,end
% ---把不在S中的标号最小顶点v1加到S中,并把v1设置为当前顶点u---
    v1=0;
    k=inf;
    for i=1:n
            ins=0;
            for j=1:length(S)
              if i==S(j)
                  ins=1;
            end,end
            if ins==0
              v=i;
              if k>l(v)
                    k=l(v);  v1=v;
    end,end,end
    S(length(S)+1)=v1;
    u=v1;
end % ---------------------"while length(S)<n"的end
% -------为start到terminal的距离distance和最短路径path赋值-----
distance=l(terminal);path(1)=terminal;
i=1;
while path(i)~=start
        path(i+1)=f(path(i));
        i=i+1;
end
path(i)=start;path=path(length(path):-1:1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

例 13.3 调用函数 M 文件 Dijkstra 解答例 13.2.

图 13.9 的边权矩阵为

$$
\boldsymbol{E}=\begin{bmatrix} 2 & 3 & 1 & 2 & 7 & 3 & 4 & 4 & 1 & 6 & 5 & 11 & 8 & 7 & 6 & 9 & 5 & 10 & 9 & 10 \\ 3 & 4 & 2 & 7 & 3 & 5 & 5 & 11 & 7 & 7 & 6 & 5 & 1 & 8 & 9 & 5 & 10 & 11 & 8 & 9 \\ 3 & 5 & 8 & 6 & 5 & 6 & 1 & 12 & 7 & 9 & 2 & 10 & 8 & 10 & 3 & 7 & 9 & 2 & 9 & 2 \end{bmatrix}
$$

编写下面的 M 文件 shortpath. m 求顶点 1 到顶点 5 的最短路径

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    edge =[2  3  1  2  7  3  4  4  1  6  5  11  8  7  6  9  5  10  9  10;...
           3  4  2  7  3  5  5  11  7  7  6  5  1  8  9  5  10  11  8  9;...
           3  5  8  6  5  6  1  12  7  9  2  10  8  10  3  7  9  2  9  2];
    n =11;weight = inf*ones(n,n);
    for i =1:n
       weight(i,i) =0;
    end
    for i =1:size(edge,2)
       weight(edge(1,i),edge(2,i)) =edge(3,i);
    end
    [dis,path] =Dijkstra(weight,1,5)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

运行 shortpath 的输出为

```
dis =
     17
path =
      1      2      3      5
```

因此顶点 1 到顶点 5 的距离为 17,最短路径为 1—2—3—5,与例 13.2 中手算的结果一致.

§ 13.6 从天安门到天坛的最短行车路线

我们回到 § 13.2 讲述的实例. 在北京市交通地图信息的基础上,我们已将求从天安门到天坛的最短行车路线的问题转化为了求图 13.3 中加权图从顶点 5 到顶点 25 的最短路径问题.

图 13.3 中无向图的边权矩阵为 edge = $[\boldsymbol{E}_1,\boldsymbol{E}_2]$,其中

$$
\boldsymbol{E}_1=\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 1 & 2 & 3 & 27 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 27 & 11 & 27 & 12 & 13 & 14 & 15 & 10 & 11 & 12 & 13 \\ 8.9 & 9.3 & 4.5 & 3.5 & 7.7 & 4.9 & 6.1 & 9.8 & 10 & 9.6 & 3.7 & 6.2 & 9.9 & 9.5 & 9.6 & 10 & 9.8 & 12.5 & 3.6 & 7.4 \end{bmatrix}
$$

$$E_2=\begin{bmatrix} 13 & 14 & 9 & 10 & 11 & 12 & 16 & 15 & 17 & 18 & 19 & 20 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 \\ 14 & 15 & 17 & 18 & 16 & 16 & 20 & 21 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 26 & 23 & 24 & 25 & 26 \\ 5 & 7.3 & 13.8 & 14.2 & 3.3 & 3.1 & 10.4 & 10.7 & 10.8 & 5 & 9.4 & 21.2 & 4.8 & 4.7 & 6.6 & 8 & 5 & 9.2 & 15.1 & 7.1 \end{bmatrix}$$

编写下面的 M 文件 tian.m 求顶点 5 到顶点 25 的最短路径，由于该图是无向图，因此其带权邻接矩阵为对称阵，设置的方法与例 13.3 中有向图时不同，注意区别.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
edge=[1 2 3 4 5 6 7 1 2 3 27 4 5 6 7 8 9 10 11 12 13 14 9 10 11 12 16 15 17 18…
  19 20 18 19 20 21 22 23 24 25;2 3 4 5 6 7 8 9 10 27 11 27 12 13 14 15 10 11…
  12 13 14 15 17 18 16 16 20 21 18 19 20 21 22 23 24 26 23 24 25 26;8.9…
  9.3 4.5 3.5 7.7 4.9 6.1 9.8 10 9.6 3.7 6.2 9.9 9.5 9.6 10 9.8 12.5 3.6 7.4 5 7.3…
  13.8 14.2 3.3 3.1 10.4 10.7 10.8 5 9.4 21.2 4.8 4.7 6.6 8 5 9.2 15.1 7.1];
n=26;weight=inf*ones(n,n);
for i=1:n
    weight(i,i)=0;
end
for i=1:size(edge,2)
    weight(edge(1,i),edge(2,i))=edge(3,i);
    weight(edge(2,i),edge(1,i))=edge(3,i);
end
[dis,path]=Dijkstra(weight,5,25)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

运行程序 tian,输出结果为

```
dis =
    45.1000
path =
      5     12     16     20     24     25
```

因此顶点 5 到顶点 25 的距离为 45.1，最短路径为 5—12—16—20—24—25. 对应行车路线如图 13.10 所示，该图会在汽车导航显示屏显示. 画面可在一定比例范围放大或缩小. 首先沿着广场东侧路南行，到达路口沿弧线转向前门大街，南行直到天坛路口，再往西行 1.5 km 左右便可到达目的地天坛公园北门.

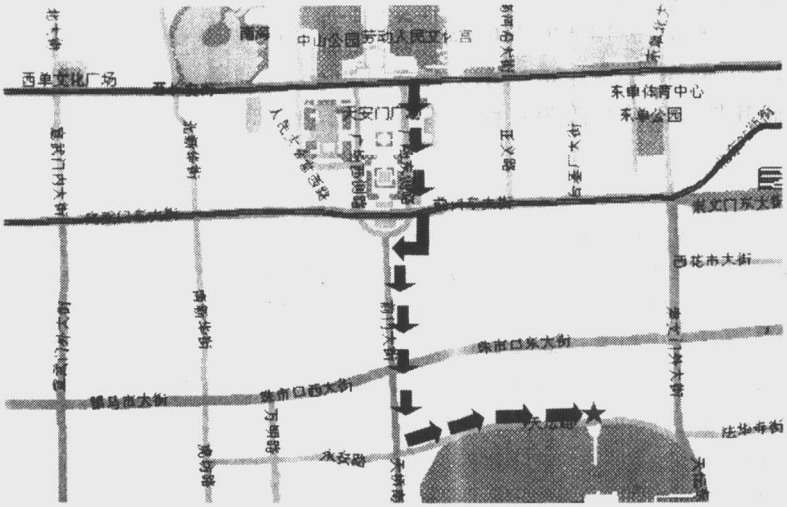


图 13.10

§13.7 如何快速求任意两顶点之间的最短路径

显然此问题可由重复 Dijkstra 算法来解决，每次取定一个顶点作起点，但这需要大量重复计算，效率不高. Floyd 另辟蹊径，提出了比这更好的算法，可一次性地求出任意两点间的最短路径和距离，其思想方法很有创意，与 Dijkstra 算法截然不同.

Floyd 算法的基本思路：从图的带权邻接矩阵 $\boldsymbol{A}=[a(i,j)]_{n\times n}$ 开始，递归地进行 n 次更新，即由矩阵 $\boldsymbol{D}^{(0)}=\boldsymbol{A}$，按一个公式，构造出矩阵 $\boldsymbol{D}^{(1)}$；又用同样的公式由 $\boldsymbol{D}^{(1)}$ 构造出矩阵 $\boldsymbol{D}^{(2)}$；……；最后又用同样的公式由 $\boldsymbol{D}^{(n-1)}$ 构造出矩阵 $\boldsymbol{D}^{(n)}$. 矩阵 $\boldsymbol{D}^{(n)}$ 的 i 行 j 列元素便是 i 号顶点到 j 号顶点的最短路径长度，称 $\boldsymbol{D}^{(n)}$ 为图的距离矩阵，同时还可引入一个后继点矩阵 path 来记录两点间的最短路径.

递推公式为

$\boldsymbol{D}^{(0)}=\boldsymbol{A}$，

$\boldsymbol{D}^{(1)}=[d_{ij}^{(1)}]_{n\times n}$，其中 $d_{ij}^{(1)}=\min\{d_{ij}^{(0)},d_{i1}^{(0)}+d_{1j}^{(0)}\}$，

$\boldsymbol{D}^{(2)} = [d_{ij}^{(2)}]_{n\times n}$，其中 $d_{ij}^{(2)} = \min\{d_{ij}^{(1)}, d_{i2}^{(1)} + d_{2j}^{(1)}\}$，

……

$\boldsymbol{D}^{(n)} = [d_{ij}^{(n)}]_{n\times n}$，其中 $d_{ij}^{(n)} = \min\{d_{ij}^{(n-1)}, d_{i,n-1}^{(n-1)} + d_{n-1,j}^{(n-1)}\}$.

提示

$d_{ij}^{(1)}$：中间只允许经过 1 号顶点，从 i 到 j 的路径中，最短路径的长度，

$d_{ij}^{(2)}$：中间只允许经过 1,2 号顶点，从 i 到 j 的路径中，最短路径的长度，

……

$d_{ij}^{(k)}$：中间只允许经过 $1,2,\cdots,k$ 号顶点，从 i 到 j 的路径中，最短路径的长度，

……

$d_{ij}^{(n)}$：中间允许经过 $1,2,\cdots,n$ 号顶点（即任何顶点），从 i 到 j 的路径中，最短路径的长度，此即为 i 到 j 的最短路径长度.

上述矩阵序列 $\{\boldsymbol{D}^{(k)}\}$ 可递归地产生，利用循环迭代便可简便求出. 算法的详细步骤如下：

Floyd 算法步骤：

$d(i,j)$：$d_{ij}^{(k)}$；

path(i,j)：对应于 $d_{ij}^{(k)}$ 的路径上 i 的后继点，最终的取值为 i 到 j 的最短路径上 i 的后继点.

输入带权邻接矩阵 $\boldsymbol{A} = [a(i,j)]_{n\times n}$

1）赋初值

对所有 i,j，$d(i,j) = a(i,j)$；当 $a(i,j) = \infty$ 时，path$(i,j) = 0$，否则 path$(i,j) = j$；$k = 1$.

2）更新 $d(i,j)$，path(i,j)

对所有 i,j，若 $d(i,k) + d(k,j) \geqslant d(i,j)$，则转 3）；否则 $d(i,j) = d(i,k) + d(k,j)$，path(i,j) = path(i,k)，$k = k + 1$，继续执行 3）.

3）重复 2）直到 $k = n + 1$.

例 13.4 一个编程例子

借助 MATLAB 软件，用 Floyd 算法求图 13.11 所示的加权有向图中任意两点间的最短路径及距离.

加权有向图的存储结构采用带权邻接矩阵$[a(i,j)]$.

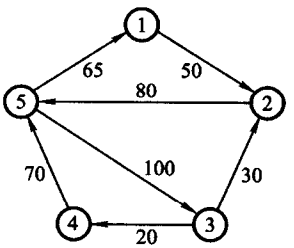


图 13.11

MATLAB 程序：

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %  Floyd's  Algorithm
  function[D,path] = floyd1(a)
  n = size(a,1);
  %设置 D 和 path 的初值
  D = a;path = zeros(n,n);
  for i = 1:n
      for j = 1:n
          if D(i,j) ~ = inf
              path(i,j) = j;      %j 是 i 的后继点
          end
      end
  end
  %做 n 次迭代,每次迭代均更新 D(i,j)和 path(i,j)
  for k = 1:n
      for i = 1:n
          for j = 1:n
              if D(i,k) + D(k,j) < D(i,j)
                D(i,j) = D(i,k) + D(k,j);     %修改长度
                path(i,j) = path(i,k);        %修改路径
              end
          end
      end
  end
```

在 MATLAB 命令窗键入：

```
  a = [0 50 inf inf inf; inf 0 inf inf 80; inf 30 0 20 inf;
inf inf inf 0 70; 65 inf 100 inf 0];
  [D,path] = floyd1(a)
```

运行结果：

```
  D =
       0    50   230   250   130
     145     0   180   200    80
     155    30     0    20    90
     135   185   170     0    70
      65   115   100   120     0
  path =
```

```
     1     2     2     2     2
     5     2     5     5     5
     4     2     3     4     4
     5     5     5     4     5
     1     1     3     3     5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

因此,由最短距离矩阵 **D** 和最短路径矩阵 path,容易得出任意两点之间的最短路径及其长度. 如,顶点 1 到顶点 3 的最短路径长度:**D**(1,3) = 230,最短路径:1→2→5→3. 这是因为,path(1,3) = 2,意味着顶点 1 的后继点为 2,又 path(2,3) = 5,从而顶点 2 的后继点为 5,同理,因 path(5,3) = 3,从而顶点 5 的后继点为 3,故 1→2→5→3 便是顶点 1 到顶点 3 的最短路径.

在该程序运行过程中,每次循环所得到的 **D** 和 path 的变化情况在表 13.3 中给出.

表 13.3　Floyd 算法求解例 13.4 的动态执行情况

```
k=0   D =                                  path =
        0     50    Inf   Inf   Inf           1   2   0   0   0
       Inf    0     Inf   Inf   80            0   2   0   0   5
       Inf    30    0     20    Inf           0   2   3   4   0
       Inf    Inf   Inf   0     70            0   0   0   4   5
        65    Inf   100   Inf   0             1   0   3   0   5
k=1   D =                                  path =
        0     50    Inf   Inf   Inf           1   2   0   0   0
       Inf    0     Inf   Inf   80            0   2   0   0   5
       Inf    30    0     20    Inf           0   2   3   4   0
       Inf    Inf   Inf   0     70            0   0   0   4   5
        65    115   100   Inf   0             1   1   3   0   5
k=2   D =                                  path =
        0     50    Inf   Inf   130           1   2   0   0   2
       Inf    0     Inf   Inf   80            0   2   0   0   5
       Inf    30    0     20    110           0   2   3   4   2
       Inf    Inf   Inf   0     70            0   0   0   4   5
        65    115   100   Inf   0             1   1   3   0   5
```

续表

k = 3	D =					path =				
	0	50	Inf	Inf	130	1	2	0	0	2
	Inf	0	Inf	Inf	80	0	2	0	0	5
	Inf	30	0	20	110	0	2	3	4	2
	Inf	Inf	Inf	0	70	0	0	0	4	5
	65	115	100	120	0	1	1	3	3	5
k = 4	D =					path =				
	0	50	Inf	Inf	130	1	2	0	0	2
	Inf	0	Inf	Inf	80	0	2	0	0	5
	Inf	30	0	20	90	0	2	3	4	4
	Inf	Inf	Inf	0	70	0	0	0	4	5
	65	115	100	120	0	1	1	3	3	5
k = 5	D =					path =				
	0	50	230	250	130	1	2	2	2	2
	145	0	180	200	80	5	2	5	5	5
	155	30	0	20	90	4	2	3	4	4
	135	185	170	0	70	5	5	5	4	5
	65	115	100	120	0	1	1	3	3	5

1）Floyd 算法中记录最短路径的矩阵 Path，与 Dijkstra 算法中记录最短路径的向量 f 有何区别和联系？如何由矩阵 Path 得到任意两点间的最短路径？为什么？在 m 文件 floyd1. m 中加入一些语句，使其直接输出全部顶点间的最短路径及其长度，而不只是 D 和 path.

2）两算法的基本思路完全不同，是图论中颇具代表性的两类算法，它们各自有什么优点？受此启发，构思不同于 Floyd 算法的求任意两点间最短路径方法.

§13.8 操　　练

操练　新建公路的线路设计及其合理性论证

某城市拟在两相距约 100 km 的两城镇之间修建一条公路，图 13.12 给出了该地域的地图. 政府想对该公路的建设做一个可行性研究，确定诸如建设费用、经济效益和环境影响这些运输计划要素. 该课题交给交通运输计划处，要求提供

一条合理的线路选择，并给出理由. 基本的工程成本为 30 万元/公里，每修建一座桥梁，工程成本将增加 200 万元. 由于地质结构的不同，修路的费用也不同，为简便，将该地区划分为 14 行 25 列的 5 km × 5 km 的小方形，共 350 个，每经过一个崎岖小方形时，建路费用将增加 100 万到 500 万.

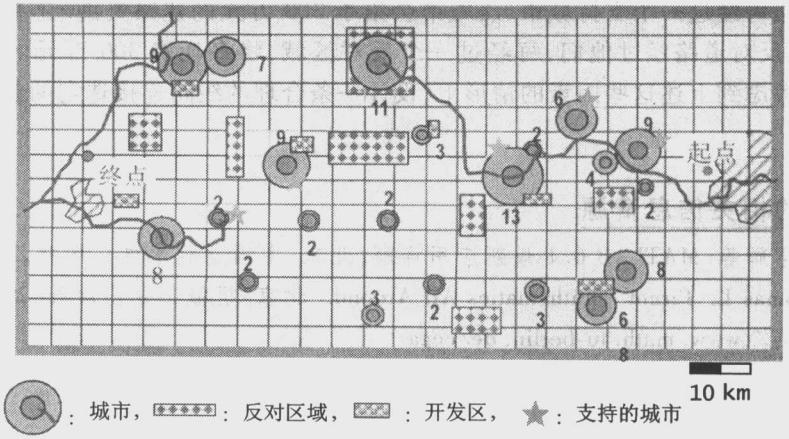


图 13.12　修建公路地区的地图，曲线代表河流，
城市旁边的数字代表该城市的人数（单位：万人）

各小方形的崎岖程度分为 0，1，2，3 级共 4 个级别，增加的费用分别为 0，100 万，300 万和 500 万. 该地区各小方形的崎岖程度为：

$$\begin{bmatrix}
3&3&2&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
3&3&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
2&2&3&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0\\
1&1&0&0&1&2&1&0&1&0&0&0&0&0&1&1&3&2&1&0&0&0&0&0&0&0\\
1&0&0&0&3&3&3&0&0&1&1&1&0&0&1&3&3&1&0&0&0&0&0&0&0&0\\
0&0&0&2&3&3&1&0&0&0&2&2&1&2&2&3&1&0&0&0&0&0&0&0&0&0\\
0&0&0&3&3&3&2&1&0&3&3&3&3&3&2&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&3&3&2&2&2&3&3&3&3&2&1&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&2&1&0&0&0&0&1&2&2&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&3&3&3&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

线路选择的基本要求是：不能穿过城市中心（地图中的黑色圆）且不能分岔. 你的任务是：

1）设计一条线路连接已选定的起点和终点，使费用尽可能低，总长度尽量短.

2）考虑到政府确定了6个开发区，为加速其发展，修建的公路每经过一个开发区，财政部将拨给500万的额外款项. 另外，有5个城市对修建公路表示支持，公路每经过一个支持城市，该城市会给予300万元的建路资助. 还有7个区域强烈反对道路经过他们，每经过一个反对区域，将多耗费300万元的占地补偿. 在考虑到上述这些因素的情形下，设计一条合理的线路连接选定的起点和终点.

更多的相关信息资源

1 王学辉等. MATLAB 6.1 最新应用详解. 北京：中国水利水电出版社，2002

2 Thomas L. Pirnot. Mathematics All Around. 北京：机械工业出版社，2003

3 http://www.math.tu-berlin.de/coga/

附录　MATLAB 软件简介

§A.1　概　　述

MATLAB 软件是适合多学科、多种工作平台的功能强大、界面友好、且开放性很强的大型优秀应用软件. 可以实现数值分析、优化技术、数理统计、偏微分方程数值解、自动控制、数字信号处理、图像处理、时间序列分析、动态系统仿真等各领域的计算和绘图功能,它将各种算法与处理以函数的形式分类成库,使用时按相应格式直接调用就可以解决问题. 借助于该软件可使科学研究和解决各种具体问题的效率大大提高.

MATLAB 语言是以数组为基本数据单位,包括控制流程语句、函数、数据结构、输入输出及面向对象等特点的高级语言,具有以下主要特点:

1) 运算符和库函数极其丰富,语言简洁,编程效率高. MATLAB 除了提供和 C 语言一样的运算符外,还提供广泛的矩阵和向量运算符. 利用其运算符和库函数可使其程序相当简短,两三行语句就可实现几十行甚至几百行 C 语言或 FORTRAN 语言编写的程序功能.

2) 既具有结构化的控制语句(如 for 循环、while 循环、break 语句、if 语句和 switch 语句),又有面向对象的编程特性.

3) 图形功能强大. 它既包括对二维和三维数据可视化、图像处理、动画制作等高层次的绘图命令,也包括可以完全修改图形局部及编制完整图形界面的、低层次的绘图命令.

4) 功能强大的工具箱. 工具箱可分为两类:功能性工具箱和学科性工具箱. 功能性工具箱主要用来扩充其符号计算功能、图示建模仿真功能、文字处理功能以及与硬件实时交互的功能. 而学科性工具箱是专业性比较强的,如优化工具箱、统计工具箱、控制工具箱、小波工具箱、图像处理工具箱、通信工具箱等.

5) 易于扩充. 除内部函数外,所有 MATLAB 的核心文件和工具箱文件都是可读可改的源文件,用户可修改源文件和加入自己的文件,它们可以与库函数一

样被调用.

§ A.2 MATLAB 环境

MATLAB 既是一种语言,又是一个编程环境.这一节将集中介绍 MATLAB 提供的编程环境.作为一个编程环境,MATLAB 提供了很多方便用户管理变量、输入输出数据以及生成和管理 M 文件的工具.所谓 M 文件,就是用 MATLAB 语言编写的、可在 MATLAB 中运行的程序.

A.2.1 MATLAB 的运行方式

MATLAB 提供两种运行方式,命令行运行方式和 M 文件运行方式.

命令行运行方式 在命令窗口输入命令行来实现计算或绘图功能.MATLAB 命令行的一般形式为:变量 = 表达式 或 表达式.例如,要求矩阵 A 的逆矩阵,其中

$$A=\begin{bmatrix}2 & 7\\ 3 & 5\end{bmatrix}.$$

在命令窗口输入下面的命令行

```
A=[2,7;3,5];
B=inv(A)
```

将在命令行的下面显示:

```
B=
   -0.4545    0.6364
    0.2727   -0.1818
```

若在表达式后面跟分号";",运行后不会显示表达式的计算结果,这对有大量输出数据的程序特别有用,因为写屏将花费大量系统资源来进行十进制和二进制之间的转换,用分号关掉不必要的输出将会使程序运行速度成倍甚至成百倍的提高.在表达式后面跟逗号",",或不跟任何符号,运行后会显示该表达式的计算结果.

M 文件运行方式 在 MATLAB 窗口中单击 New M - File 快捷按钮,打开 M 文件编辑窗口,在此输入 MATLAB 程序文件,可以进行调试或运行.运行后的结

果会显示在命令窗口.

A.2.2 MATLAB 6.x 中的窗口

MATLAB 中常用的窗口有命令窗口、M 文件窗口、起始面板、工作空间窗口、命令历史窗口、当前路径窗口和图形窗口等.

1. 命令窗口

用户与 MATLAB 进行交互的主要场所. 可以在此键入各种 MATLAB 命令进行各种操作,键入数学表达式进行计算. 例如,当键入变量赋值命令:x = 4.5 并回车,将显示:

```
x =
    4.5
```

再键入:

```
y = sin(x * pi)
```

并回车,将显示:

```
y =
    1
```

接着键入:

```
x = 6;
z = 8;
2 * x + y - 3 * z
```

输出:

```
ans =
     -11
```

1) 当不指定输出变量时,MATLAB 将计算值赋给缺省变量 ans (answer 的缩写).

2) 在 MATLAB 里,有很多控制键和方向键可用于命令行的编辑.

例如,当漏敲命令 ho = (1 + sqrt(5))/2 的字符"r"时,将会给出错误信息:

```
Undefined function or variable 'sqt'
```

这时你不用重新键入整行命令,而只需按"↑"键,就会再显示刚才键入的命令行,在相应的位置键入"r",接着按回车即可正常运行.

反复使用"↑"键,可以回调以前键入的所有命令行. 表 A.1 给出了 MATLAB 的控制键及其作用.

表 A.1　命令窗口的控制键功能

键	相应快捷键	功能
↑	ctrl-p	重调前一行
↓	ctrl-n	重调下一行
←	ctrl-b	向左移一个字符
→	ctrl-f	向右移一个字符
ctrl→	ctrl-r	向右移一个字
ctrl←	ctrl-l	向左移一个字
Home 键	ctrl-a	移动到行首
End 键	ctrl-e	移动到行尾
Esc 键	ctrl-u	清除一行
Del	ctrl-d	删除光标处字符
Backspace		删除光标左边字符
	ctrl-k	删除至行尾

若一个表达式在一行写不下，可换行，但必须在行尾加上三个英文句号.

例如，可键入：

```
s =1 -1/2 +1/3 +1/4 +sin(3 * x +y) -cos(x)...
-1/8 +1/10 +1/20
```

运算符 =、+、- 前后的空格不影响计算结果.

2. MATLAB 的程序编辑窗口

可在该窗口编辑 M 文件，这里也有菜单栏和工具栏，使编辑和调试程序非常方便. 单击“run”选项，可以运行 M 文件.

3. 起始面板

该窗口中显示 MATLAB 总包和已安装的工具箱的帮助、演示、GUI 工具和产品主页等 4 个方面的内容. 如想查看这 4 个方面的内容，则双击对应的图标即可.

4. 工作空间窗口

该窗口列出了自启动 MATLAB 软件以来所建立的所有变量的信息，包括变量名、变量型号、变量数据字节大小和变量类型. 这些变量的值和信息存储在内存区域，该区域称为工作空间.

每打开一次 MATLAB，MATLAB 会自动建立一个工作空间，运行 MATLAB 的程序或命令时，产生的所有变量被加入到工作空间. 除非用特殊的命令删除某变量，否则该变量在关闭 MATLAB 之前一直保存在工作空间. 工作空间在 MATLAB 运行期间一直存在，关闭 MATLAB 后，工作空间自动消除.

可以随时查看工作空间中变量的值和删除变量，工作空间中的所有变量可以保存到一个文件中，便于以后使用.

clear	清除工作空间中的所有变量
clear 变量名	清除指定的工作空间变量
save 文件名	将当前工作空间的变量储存在一个 MAT - 文件中
load 文件名	调出一个 MAT - 文件

quit ↵或单击右上角的“×”按钮，退出工作区

也可以在工作空间窗口中选择某个数值型变量以后，单击  图标，将打开数组编辑器，显示该变量的值，并可以对其进行修改. 选中变量后，单击  图标，将删除该变量. 还可单击保存按钮，将当前工作空间的变量储存在一个 MAT - 文件中.

5. 命令历史窗口

该窗口列出了命令窗口中所有执行过的命令，可以通过双击其中某个命令行来执行该命令，也可通过拖拉或复制操作将命令行复制到命令窗口再执行.

6. 图形窗口

在 File 菜单的 New 次级菜单中选择 Figure 选项或执行其他绘图命令，将打开图形窗口，利用图形窗口菜单和工具栏中的选项，可以对图形进行线型、颜色、标记、三维视图、光照和坐标轴等内容的设置.

7. 当前目录窗口

该窗口中显示当前目录下所有文件的文件名、文件类型和最后修改时间. 选中某个文件后，利用 File 菜单下的 Open 选项可打开该文件，利用 Edit 菜单下的选项可剪切、复制和删除该文件.

A.2.3 MATLAB 的帮助系统

MATLAB 6.x 里有以下几种方法获得帮助：帮助命令、联机帮助、演示帮助或直接链接到 Math Works 公司(对于已联网的用户).

1. 帮助命令

帮助命令是查询函数语法的最基本方法，查询信息直接显示在命令窗口.

```
help  函数名↵
```

可寻求关于某函数的帮助

例如，键入：

```
help  sqrt↵
```

显示：

```
SQRT  square root.…
```

帮助文本中的函数名 SQRT 是大写的，以突出函数名，但在使用函数时，应用小写 sqrt.

MATLAB 按照函数的不同用途分别将其存放于不同的子目录下.

help 子目录标题　　可显示某一类的所有函数或命令.

例如，键入：

```
help graph2d
```

显示：

```
Two dimensional graphs.
Elementary X-Y graphs.
plot          -Linear plot.
loglog        -Log-log scale plot.
semilogx      -Semi-log scale plot.
semilogy      -Semi-log scale plot.
…
```

命令 help 将显示帮助的所有子目录标题.

```
lookfor 关键词
```

它是通过搜索所有 MATLAB help 子目录标题与 MATLAB 搜索路径中 M 文件的第一行，返回包含所指定关键词的那些项. 最重要的是关键词不一定为命令.

例如，键入：

```
lookfor complex
```

显示：

```
CONJ          Complex conjugate.
IMAG          Complex imaginary part.
REAL          Complex real part.…
demo          可浏览例子和演示
help demos 将给出所有的演示题目.
```

2. 联机帮助

在 MATLAB 界面中单击工具栏里的问号按钮或单击 Help 菜单中的 MATLAB Help 选项，或在命令窗口键入 helpwin 命令，可以打开联机帮助窗口. 在界面左边的目录栏中单击项目名称或图标，将在右侧的窗口中显示对应的帮助信息.

3. 演示帮助

单击 Help 菜单中的 Demos 选项，可以打开演示窗口，在左边的窗口中选

择总包或工具箱名称，然后进一步选择希望观看的内容项目.选中后，将在右侧上面的窗口中显示对应项目的演示说明，单击“run”按钮，进行演示.

§A.3 数值运算

A.3.1 变量

对于变量，MATLAB 不需要任何类型的说明或维数语句，当输入一个新变量名时 MATLAB 自动建立变量并为其分配内存空间. 如果变量已经存在，MATLAB 将用新的内容取代该变量原来的内容. 要想显示变量的内容，只需键入该变量名即可.

变量名命名规则

1）以字母开头，后面可跟字母，数字或下短线；

2）大小写字母有区别；

3）不超过 31 个字符.

例如，xie23_1，a，A 和 Arui32r 这四个都是变量.

特殊变量名

ans	用于结果的缺省变量名
pi	圆周率
eps	计算机的最小数，当和 1 相加就产生比 1 大的数
inf	无穷大，如 1/0
NaN	不定量，如 0/0
i 或 j	$i=j=-1$ 的开方
realmin	最小可用正实数
realmax	最大可用正实数

一般在 MATLAB 中数据的存储与计算都是以双精度进行的，当然，用户可以改变其在命令窗口的显示格式. 注意这不会改变其计算和存储的数据精度. 控制数据显示格式的命令是 format，其调用格式如下：

```
format short   默认值,5 位定点表示
format long   15 位定点表示
format compact   变量之间没有空行
format loose   变量之间有空行
……
```

A.3.2 数组的创建与运算

MATLAB 中最基本的数值运算对象是数组或矩阵. 标量可看作是 1×1 型的矩阵,向量可看作是 $1\times n$ 或 $n\times1$ 的矩阵. 一维数组是向量,二维数组便是矩阵,还有三维甚至更高维的数组. 标量运算是数学的基础,然而,当需要对多个数执行同样的运算时,采用数组或矩阵运算将非常简捷和方便.

1. 创建矩阵

1) 直接定义

键入:

```
A=[1 2 3;4 5 6]
```

输出:

```
A=
   1   2   3
   4   5   6
```

这里 A 为一个 2 行 3 列的数组或矩阵. 空格或逗号用于分隔某一行的元素,分号表示开始新的一行.

键入:

```
A(2,3)=0   %将第 2 行,第 3 列的元素置为 0.
```

输出:

```
A=
   1   2   3
   4   5   0
```

2) 向量的简单构造

前面我们通过键入矩阵或数组中每个元素来输入一个矩阵或数组,当数组中的元素有成百上千时,怎么办呢? 对于向量有两种简单的输入格式. 例如,

```
x=0:0.1:1                %  从 0 到 1,增量为 0.1 的等间隔数
x=linspace(0,pi,11)      %  11 个从 0 到 pi 的等间隔数
```

在 MATLAB 中这两种创建向量的方式是最常见的.

上述向量创建形式所得到的向量其元素之间是线性分隔的特殊情况,当需要对数分隔的向量时,MATLAB 提供了函数 `logspace`.

格式:`x=logspace(first,last,n)`

创建从 10 的 `first` 次方开始,到 10 的 `last` 次方结束,有 `n` 个元素的对数分隔行向量 `x`.

有时所需的向量不具有易于描述的线性或对数分隔关系,这时使用向量编址和表达式结合的功能可避免每次一个地输入数组元素. 例如,

键入：

```
a=1:5;b=1:2:9;c=[b a]
```

输出：

```
c=
  1  3  5  7  9  1  2  3  4  5
```

创建的向量 c，由 b 中元素和 a 中元素构成.又如，

键入：

```
d=[a(1:2:5)  1  0  1]
```

输出：

```
d=
  1  3  5  1  0  1
```

上述所创建的向量都是行向量，如何创建列向量呢？可使用转置算子(')把行向量变成列向量.如

键入：

```
a=1:4;          %  表示从 1 到 4，增量为 1 的行向量
b=a'            %  表示向量的转置
```

输出：

```
b=
     1
     2
     3
     4
```

有两种转置的符号：

① 当数组是复数时，(')产生的是复数共轭转置；

② (')只对数组转置，但不进行共轭.

3) 向量的操作

键入：

```
x=[0,0.1*pi,0.2*pi,0.3*pi,0.4*pi,0.5*pi,0.6*pi,0.7*pi,0.8*
pi,0.9*pi,pi]
y=sin(x)
```

输出：

```
y=
Columns1 through 7
0   0.3090   0.5878   0.8090   0.9511   1.0000   0.9511
Columns 8 through 11
0.8090   0.5878   0.3090   0.0000
```

在 MATLAB 中,数组元素用下标访问,如 y(2)是 y 的第 2 个元素. 例如,

键入:

```
y(3)    % 表示 y 的第 3 个元素
```

输出:

```
ans =
        0.5878
```

为了同时访问一块元素,MATLAB 用冒号来表示.

键入:

```
x(1:5)
```

输出;

```
ans =
     0    0.3142    0.6283    0.9425    1.2566
```

键入:

```
y(3:-1:1)
```

输出:

```
ans =
     0.5878      0.3090      0
```

3:-1:1 表示从 3 开始减 1 计数,到 1 为止. 又如

键入:

```
x(2:2:7)
```

输出:

```
ans =
     0.3142  0.9425  1.5708
```

2:2:7 表示从 2 开始加 2 计数,到 7 为止. 再如

键入:

```
y([8  2  9  1])
```

输出:

```
ans =
     0.8090  0.3090  0.5878  0
```

这里是按照数组[8 2 9 1]提供的次序来提取 y 数组中的元素.

4) 矩阵的剪裁与拼接

从一个矩阵中取出若干行(列)构成新矩阵称为**剪裁**,":"是非常重要的剪裁工具. 例如,

键入:

```
A=[1 2 3;4 5 6;7 8 9];
A(3,:)  % A 的第 3 行
```

输出：

```
ans =
     7   8   9
```

键入：

```
A(:,1)  %A 的第 1 列
```

输出：

```
ans =
     1
     4
     7
```

键入：

```
B = A(2:3,:)  %A 的第 2,3 行
```

输出：

```
B =
     4   5   6
     7   8   9
```

键入：

```
C = A(1:2,[1  3])  %A 的第 1,2 行,第 1,3 列
```

输出：

```
C =
     1   3
     4   6
```

还有 A(1:2:3,3:-1:1)，想想将输出什么？

将几个矩阵接在一起称为**拼接**，左右拼接行数要相同，上下拼接列数要相同. 例如，

键入：

```
D = [C,zeros(2,1)]
```

输出：

```
D =
     1   3   0
     4   6   0
```

键入：

```
E = [D;eye(2),ones(2,1)]
```

输出：

```
E =
     1   3   0
     4   6   0
```

```
1  0  1
0  1  1
```

A(:)　逐列提取 A 中的所有元素作为一个列向量.

A(i)　把 A 看作列向量 A(:),提取其中第 i 个元素.

A(r,c)　提取 A 中,由索引向量 r 定义的行,和由索引向量 c 定义的列所构成的 A 的子数组.

A(r,:)　提取 A 中,由索引向量 r 定义的行,和全部列所构成的 A 的子数组.

A(:,c)　提取 A 中,由全部行,和由索引向量 c 定义的列所构成的 A 的子数组.

2. 数组的运算

1) 标量 - 数组运算

标量与数组的加、减、乘、除和点乘方(.^)是对数组的每个元素进行运算,得到同样大小的数组.例如,

键入:

```
a = 1:5;  3 * a - 5
```

输出:

```
ans = -2  1  4  7  10
```

键入:

```
a.^2
```

输出:

```
ans = 1  4  9  16  25
```

2) 数组 - 数组运算

当两个数组具有相同大小时,加、减、点乘(.*)、点除(./)、和点乘方运算(.^)是按元素对元素方式进行的.例如,

键入:

```
g = [1  2  3;5  6  7;8  9  10];h = [1  1  1;2  2  2;3  3  3];
g.*h
```

输出:

```
ans =
     1     2     3
    10    12    14
    24    27    30
```

提示

设 a = [a1 a2 a3], b = [b1 b2 b3], c = 标量

```
标量加法  a + c = [a1 + c  a2 + c  a3 + c];
标量乘法  a * c = [a1 * c  a2 * c  a3 * c];
数组加法  a + b = [a1 + b1  a2 + b2  a3 + b3];
数组乘法  a .* b = [a1 * b1  a2 * b2  a3 * b3];
数组右除  a ./ b = [a1 / b1  a2 / b2  a3 / b3];
数组左除  a .\ b = [b1 / a1  b2 / a2  b3 / a3];
数组求幂  a .^ c = [a1^c a2^c a3^c]; c .^ a = [c^a1  c^a2  c^a3];
          a .^ b = [a1^b1  a2^b2  a3^b3];
```

3. 矩阵的运算

MATLAB 提供了下列矩阵运算：

+ 加法；- 减法；' 转置运算；* 乘法；^ 乘幂；\ 左除；/ 右除

除除法外其他运算都与线性代数中定义的一样. 这里只介绍一下除法运算.

设 $\boldsymbol{A}$ 是可逆矩阵，

1) $\boldsymbol{AX}=\boldsymbol{B}$ 的解是 $\boldsymbol{A}$ 左除 $\boldsymbol{B}$，即 $\boldsymbol{X}=\boldsymbol{A}\backslash\boldsymbol{B}$，意为用 $\boldsymbol{A}^{-1}$ 左乘以 $\boldsymbol{B}$.

2) $\boldsymbol{XA}=\boldsymbol{B}$ 的解是 $\boldsymbol{A}$ 右除 $\boldsymbol{B}$，即 $\boldsymbol{X}=\boldsymbol{B}/\boldsymbol{A}$，意为用 $\boldsymbol{A}^{-1}$ 右乘以 $\boldsymbol{B}$.

注意

1) 当 A 为方阵时，p > 1 整数，A^p 表示 A 自乘 p 次；

2) 当 A 和 P 均为矩阵时，不能计算 A^P.

A.3.3 函数

1. 常用的数学函数

单变量数学函数的自变量可以是数组，此时，输出的是各元素的函数值构成的同规格数组. 例如，

键入：

```
a = [1  2  3;4  5  6];sin(a)
```

输出：

```
ans =
    0.8415    0.9093    0.1411
   -0.7568   -0.9589   -0.2794
```

MATLAB 中的常用数学函数有：

三角函数　正弦 sin(x)，双曲正弦 sinh(x)，反正弦 asin(x)，反双曲正弦 asinh(x)，cos(x)，tan(x)，cot(x)，sec(x)，csc(x)等.

指数函数　exp(x),自然对数 log(x),常用对数 log10(x),以 2 为底的对数 log2(x),平方根 sqrt(x)等.

整值函数　朝零方向取整 fix(x),朝 $-\infty$ 方向取整 floor(x),朝 $+\infty$ 方向取整 ceil(x),四舍五入到最接近的整数 round(x),符号函数 sign(x)等.

其他数学函数:绝对值或复数的幅值 abs(x)等.

2. 数组特征及矩阵操作函数

size(A)　返回一个二元素向量,第一个元素为 A 的行数,第二个元素为 A 的列数

size(A,1)　返回 A 的行数

size(A,2)　返回 A 的列数

length(A)　返回 max(size(A))

flipud(A)　矩阵作上下翻转

fliplr(A)　矩阵作左右翻转

diag(A)　提取 A 的对角元素返回列向量

diag(v)　以向量 v 作对角元素创建对角矩阵

最大值 max,最小值 min,求和 sum,求平均值 mean,按升序排列 sort 等函数,只有当它们作用于向量时才有意义,它们也可作用于矩阵,此时产生一个行向量,行向量的每个元素是函数作用于矩阵相应列向量的结果.例如,

键入:

```
a=[-4.5 9 7 -2.8 3.5 9.5 5.4 7.3];
min(a),[m,im]=min(a),[M,iM]=max(a),[ra,ir]=sort(a)
```

输出:

```
ans =
    -4.5000
m =
   -4.5000
im =
     1
M =
    9.5000
iM =
     6
ra =
   -4.500   -2.800   3.5000   5.4000   7.0000   7.3000
   9.000    9.5000
ir =
```

```
1    4    5    7    3    8    2    6
```

3. 矩阵函数

d=eig(A),[v,d]=eig(A)	特征值与特征向量
det(A)	行列式计算
inv(A)	矩阵的逆
poly(A)	特征多项式
rank(A)	矩阵的秩
zeros(m,n)	m 行 n 列的零矩阵
ones(m,n)	m 行 n 列的全 1 矩阵
eye(n)	n 阶单位矩阵
rand(m,n)	m 行 n 列的均匀分布随机数矩阵
randn(m,n)	m 行 n 列的正态分布随机数矩阵

利用帮助了解向量函数 max, min, sum, mean, sort, length.矩阵函数 rand,size 的功能和用法.

§A.4 图形功能

A.4.1 二维图形

1. 基本的绘图命令

plot 命令打开一个称为图形窗口的窗口,将坐标轴缩扩以适应数据,绘制数据.如果已经存在一个图形窗口,则 plot 命令会清除当前图形窗口的图形,绘制新的图形.

1) plot(y) 当 y 为向量时,是以 y 的分量为纵坐标,以元素序号为横坐标,用直线依次连接数据点,绘制曲线.若 y 为实矩阵,则按列绘制每列对应的曲线,图中曲线数等于矩阵的列数.

2) plot(x,y) 若 y 和 x 为同维向量,则以 x 为横坐标,y 为纵坐标绘制连线图.若 x 是向量,y 是行数或列数与 x 长度相等的矩阵,则绘制多条不同色彩的连线图,x 被作为这些曲线的共同横坐标.若 x 和 y 为同型矩阵,则以 x,y 对应列元素为横纵坐标分别绘制曲线,曲线条数等于矩阵的列数.

3) plot(x1,y1,x2,y2,…) 在此格式中,每对 x,y 必须符合 plot(x,y)中的要求,不同对之间没有影响,命令将对每一对 x,y 绘制曲线.

在以上三种格式中的 x,y 都可以是表达式.

例 A.1 作出 $y = \sin x$ 在$[0,2\pi]$上的图形.

键入:

```
x = linspace(0,2 * pi,30);
y = sin(x);
plot(x,y);
```

结果如图 A.1.

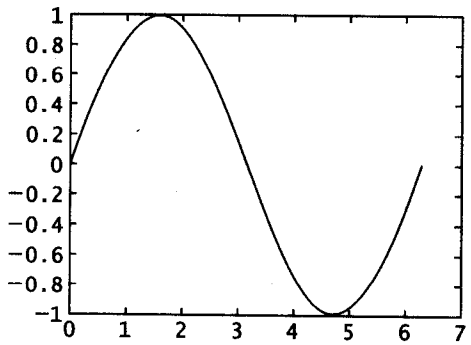


图 A.1

例 A.2 在同一个坐标下作出两条曲线:$y = \sin x$ 和 $y = \cos x$ 在$[0,2\pi]$上的图形.

键入:

```
x = 0:2 * pi/30:2 * pi;y = [sin(x);cos(x)];plot(x,y);
```

或键入:

```
x = 0:2 * pi/30:2 * pi;y1 = sin(x);y2 = cos(x);
plot(x,y1,x,y2);
```

都可画出图 A.2.

多条曲线的另一种画法是利用 hold 命令.在已画好的图形上,若设置 hold on,MATLAB 将把新的 plot 命令产生的图形画在原来的图形上.而命令 hold off 将结束这种状态.例如,

```
x = linspace(0,2 * pi,30);
y = sin(x);plot(x,y);
```

先画好图 A.1,然后用

```
hold on,z = cos(x);plot(x,z);hold off
```

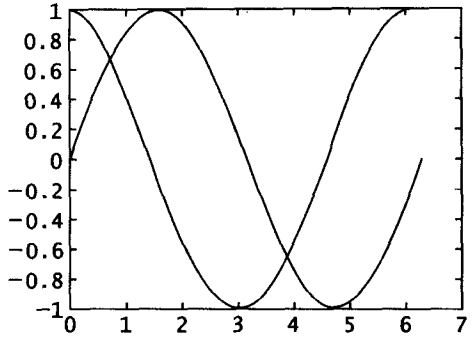


图 A.2

增加 cos(x)的图形,也可得到图 A.2.

2. 基本的绘图控制

在调用 plot 时可以指定颜色、线型和数据点图标,基本的调用格式为

```
plot(x,y,' color-linestyle-marker ')
```

其中,color-linestyle-marker 为一个字符串,由颜色、线型和数据点图标组成.例如,

```
plot(x,y,' y:o ')
```

该例的字符串"y:o"中,第一个字符"y"表示曲线颜色为黄色;第二个字符":"表示曲线为点线;"o"表示曲线上每个数据点处用小圆圈标出.当只指定数据点图标时,数据点将不连成线,而只画出一个个孤立的数据点.字符串参数的取值如下

颜色 y(黄);r(红);g(绿);b(蓝);w(白);k(黑);m(紫);c(青).

线型 -(实线);:(点线);-.(虚点线);--(虚线).

数据点图标 .(小黑点);+(加号);*(星号);o(小圆圈);pentagram(五角星).

在调用 plot 时可以指定线条的粗细,以及数据点图标的大小,调用格式为

```
plot(x,y,' color-linestyle-marker ',' linewidth ',width,' markersize ',size)
```

其中,width 和 size 均为非负数,即曲线的宽度和图标的大小.例如,

```
x = linspace(0,2 * pi,30);
y = sin(x);
plot(x,y,' linewidth ',2);
```

作出的曲线如图 A.3 所示.

又如:

```
x = 1:8;y = cos(x);
```

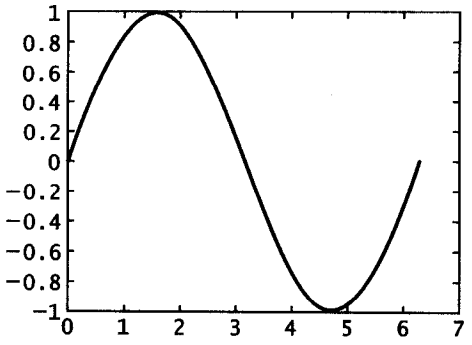


图 A.3

```
plot(x,y,'.','markersize',3)
```

作出的图形如图 A.4 所示.

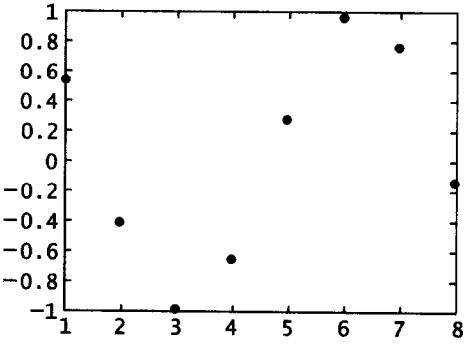


图 A.4

坐标系的控制:不特别指定时,MATLAB 自动指定图形的横纵坐标比例和显示的范围,如果你不满意,可用 axis 命令来控制,常用的有

axis([xmin xmax ymin ymax]) [ ]中分别给出 x 轴和 y 轴的最小、最大值

axis equal x 轴和 y 轴的单位长度相同

axis square 图框呈方形

axis off 取消坐标轴

3. 图形标注

MATLAB 提供了标注图形的命令,常用的有

xlabel、ylabel 和 zlabel 分别用于对 x、y、z 轴加标注

title　用于给整个图形加标题

text 和 gtext　用于在图形中特定的位置加字符串，前者字符串的位置在命令中指定，后者用鼠标指定

grid　在图形上加网格

例 A.3　在同一坐标系下画出 sin x 和 cos y 的函数图形，并适当标注.

键入：

```
x = linspace(0,2 * pi,30);y = [sin(x);cos(x)];
plot(x,y);grid;xlabel(' x ');ylabel(' y ');title(' Sine and Cosine Curves ');
text(3 * pi/4,sin(3 * pi/4),' \leftarrowsinx ');
text(3 * pi/2,cos(3 * pi/2),' cosx \rightarrow ',' HorizontalAlignment ',' right ');
```

输出结果为图 A.5.

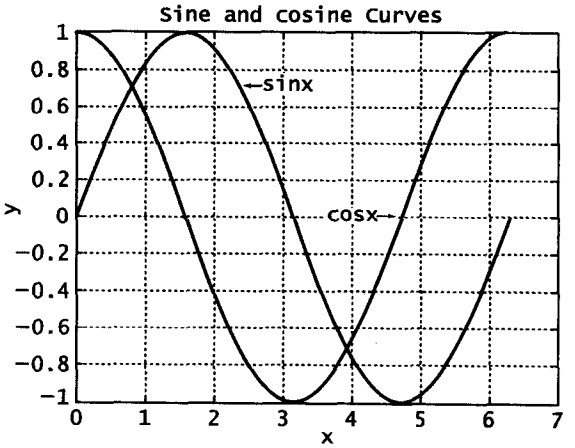


图 A.5

若使用命令 gtext('sinx')代替命令 text，则在图形窗口会出现十字线，其交点是字符串的位置，移动鼠标可移动该交点，鼠标点击一下就可将字符串固定在那里.

4. 多幅图形

Subplot(m,n,p)　可在同一个图形窗口，画出多幅不同坐标系的图形

该命令把一个画面分为 m × n 个图形区域，p 代表当前的区域号，在每个区域中分别画一个图. 子图沿第一行从左至右编号，接着排第二行，以次类推. 用法如下例.

键入：

```
x = linspace(0,2 * pi,30);y = sin(x);z = cos(x);
u = 2 * sin(x) . * cos(x);v = sin(x) . /cos(x);
subplot(2,2,1),plot(x,y),title( ' sin(x) ' )
subplot(2,2,2),plot(x,z),title( ' cos(x) ' )
subplot(2,2,3),plot(x,u),title( ' 2sin(x)cos(x) ' )
subplot(2,2,4),plot(x,v),title( ' sin(x)/cos(x) ' )
```

输出图形见图 A.6.

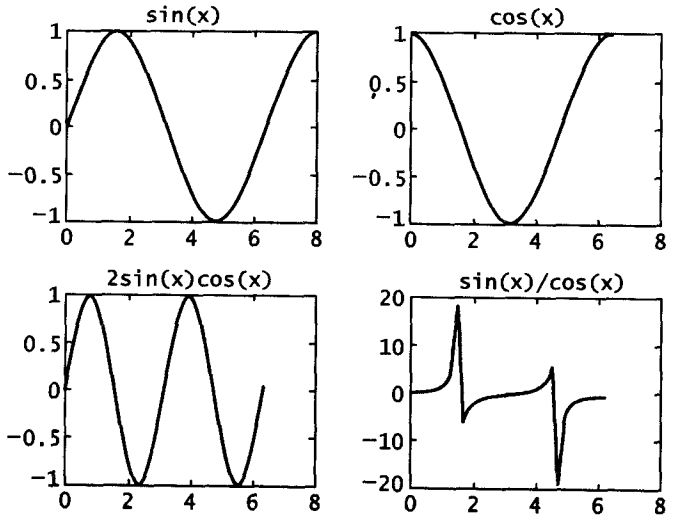


图 A.6

A.4.2 三维图形

1. 空间曲线

例 A.4 作螺旋线 $x = \sin t, y = \cos t, z = t.$

键入：

```
t = 0:pi/50:10 * pi;plot3(sin(t),cos(t),t);
```

输出图形见图 A.7.

2. 带网格的曲面

命令：[X,Y] = meshgrid(x,y);mesh(X,Y,Z)和 surf(X,Y,Z)

例 A.5 作曲面 $z = f(x,y)$ 的图形，$z = \dfrac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}$，$-7.5 \leqslant x \leqslant 7.5$，$-7.5 \leqslant y \leqslant 7.5$.

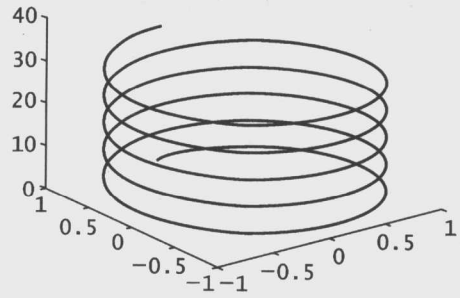


图 A.7

键入：

```
x = -7.5:0.5:7.5;
y = x;
[X,Y] = meshgrid(x,y);
R = sqrt(X.^2 + Y.^2) + eps;
Z = sin(R)./R;
mesh(X,Y,Z);
```

输出图形见图 A.8.

可将上述的画网格图的 mesh 命令改为 surf(X,Y,Z);则输出的曲面图，效果有所不同.

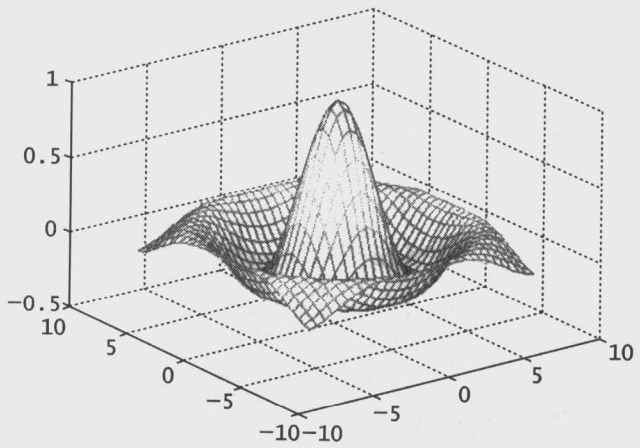


图 A.8

3. 等高线

MATLAB 还提供了画二维和三维等高线图的函数 contour 和 contour3.

例 A.6 作出由 MATLAB 的函数 peaks 产生的二元函数的曲面及其等值线图.

键入:

```
[X,Y,Z] = peaks(30);
surf(X,Y,Z);
figure(2);                    % 打开另一个图形窗口
contour(X,Y,Z,16);
figure(3);
contour3(X,Y,Z,16);
```

输出的三个图形见图 A.9 到图 A.11.

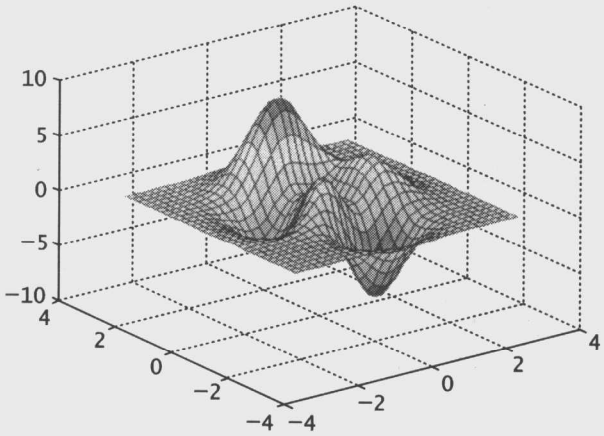


图 A.9 函数 peaks 的曲面图

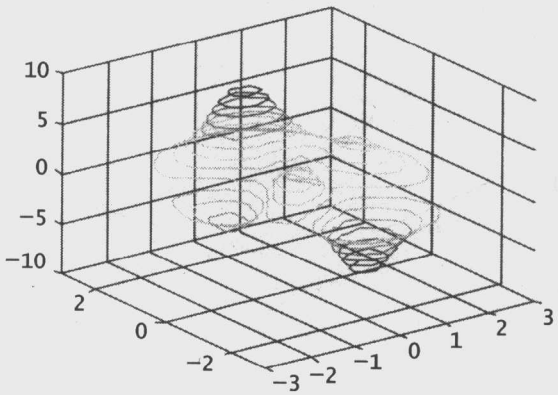


图 A.10 函数 peaks 的等值线图

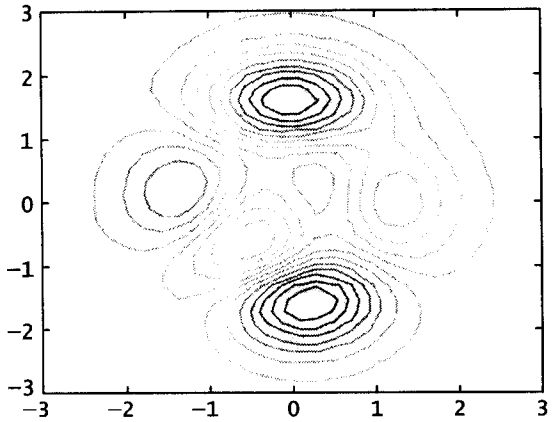


图 A.11　函数 peaks 的三维等值线图

§A.5　符号运算

前面介绍的数值运算、变量均要事先赋值，才能出现在表达式中参与运算. 但经常需对含有字符的矩阵和函数进行处理和运算，如要求函数的微分、积分，求符号矩阵的逆矩阵等等，该怎么办呢？这就要借助于 MATLAB 软件的符号运算功能. 符号运算包括符号变量、表达式、符号矩阵的创建，符号代数运算、线性代数，因式分解、展开和简化，符号代数方程求解，符号微积分，符号微分方程，符号作图等功能.

A.5.1　字符串变量、符号变量和符号表达式的建立

1. 用单引号来设定字符串

在 MATLAB 中，所有的字符串都要用单引号来引入. 例如，

键入：

```
name = ' Zhang Hua '
```

会显示：

```
name =
       Zhang Hua
```

如此,便建立了一个字符串变量 name.

键入:

```
f = ' cos(x) '
```

显示:

```
f =
   cos(x)
```

键入:

```
g = ' 3 * x^2 +2 * x +6 =0 '
```

显示:

```
g =
   3 * x^2 +2 * x +6 =0
```

2. 用函数 syms 来定义符号变量

调用格式为: syms var1 var2 …

如, syms x y u v t

就把变量 x,y,u,v,t 定义成为符号变量,后面就可以在没有赋值的情况下,出现在表达式里.

3. 用函数 sym 来建立符号表达式

调用格式为:变量 = sym('表达式')

如,键入:

```
y = sym( ' 2 +cos(x) ' )
```

将显示:

```
y =
   2 +cos(x)
```

这就建立了一个符号表达式.

键入:

```
X = sym( '[a b c;d e f;b f d]' )
```

将显示:

```
X =
   [a  b  c]
   [d  e  f]
   [b  f  d]
```

建立了一个符号矩阵 X.

4. 用函数 syms 来建立符号表达式

如:

```
syms y u;
   p = exp( -y/u)
   q = y^2 +u^3 +u * y
```

这样就建立了两个符号表达式,分别存放在变量 p 和 q 中.

A.5.2 符号和数值之间的转换

可以将数值符号转化为符号变量,并用该符号变量进行运算,这对多次重复使用一个数值表达式进行精确运算非常有用.

1. 用 sym 将数值表达式转换为符号表达式

调用格式为:变量 = sym('数值表达式')

如,键入:

```
a = sym(' 1 +2 * sqrt(3) ')
```

显示:

```
a =
   1 +2 * sqrt(3)
```

如,键入:

```
X = [8.2  1  6.3;3  5  7;4  9  1  2];A = sym(X)
```

将显示:

```
A =
   [41/5,  1,  63/10]
   [3,  5,  7]
   [4,  91/10,  2]
```

这样就得到了一个数值符号矩阵 A.

2. 用 numeric 将符号表达式转换为数值表达式

如,键入:

```
a = sym(' 1 +2 * sqrt(3) ');
numeric(a)
```

将显示:

```
ans =
     4.4641
```

3. 用 eval 计算符号表达式的值

调用格式:eval(表达式),其中表达式可以是符号表达式或字符串,也可以是有效的 MATLAB 命令或语句,其作用是执行该表达式,如键入:

```
f = sym(' 2 +x^2 ');
x = [1,2;3,4];
y = eval(f)
```

输出:

```
y =
```

```
     9    12
    17    24
```

若再键入：

```
y = eval( ' 2 + x^2 ' )
```

则输出和上面一样.

4. 符号的可变精度运算

因为数值运算的精度受每次操作所保留的位数限制，所以数值的任何运算都会引入误差，而符号表达式的运算由于不涉及数值运算，因此它的运算结果是精确的. MATLAB 数值运算的缺省精度为 32 位，还可以由函数 digits 与 vpa 来调整 .digits 返回当前的精度位数，digits(d)，把当前的精度位数设置为 d 位 .vpa(表达式，d)，使用 d 位精度返回该表达式的值 .例如

键入：

```
w = vpa( ' (1 + sqrt(5))/2 ' ,40)
```

输出：

```
w =
   1.618033988749894848204586834365638117721
```

A.5.3 符号表达式的基本代数运算

符号表达式的加、减、乘、除及幂运算等的基本的代数运算，与矩阵的数值运算几乎完全一样，用“ + ”、“ - ”、“ * ”、“/”和“^”符号进行运算. 但字符串不能用这些符号进行运算. 例如，键入

```
syms x
f = cos(x);g = sin(2 * x);
f/g + f * g
```

输出：

```
ans =
        cos(x)/sin(2x) + cos(x) * sin(2x)
```

又比如，键入：

```
syms x
f = 2 * x^2 + 3 * x - 5;g = x^2 + x - 7;
h = f + g,k = f * g
```

则输出：

```
h =
   3 * x^2 + 4 * x - 12
k =
   (2 * x^2 + 3 * x - 5) * (x^2 + x - 7)
```

这个例子中的 f 和 g 均为符号表达式.但若 f 和 g 是字符串时,会出错.例如,键入

```
f = ' 2*x^2+3*x-5 ';g = ' x^2+x-7 ';
h = f+g
```

显示:

```
??? Error using⇒ +
Array dimensions must match for binary array op.
```

显示了错误信息.对于字符串所表达的有效表达式,可用如下的运算符来进行运算.当然,这些运算符也适合于符号表达式.

```
symadd(a,b)      a 与 b 相加
symsub(a,b)      a 与 b 相减
symmul(a,b)      a 与 b 相乘
symdiv(a,b)      a 除以 b
sympow(a,b)      a 的 b 次幂
symop(  )        综合运算
```

例如,键入:

```
f = ' 4*x+6*y+3 ';g = ' 2*x^2+5*x+6 ';
h = symmul(f,g)
```

输出:

```
h =
   (4*x+6*y+3)*(2*x^2+5*x+6)
```

又如,键入:

```
f = ' cos(x) ';g = ' sin(2*x) ';
symop(f,' /',g,' + ',f,' * ',g)
```

输出:

```
ans =
     cos(x)/sin(2*x)+cos(x)*sin(2*x)
```

这些运算符对于符号表达式也适合.例如,键入:

```
syms  x
f =2*x^2+3*x-5;g =x^2+x-7;
h = symadd(f,g)
```

输出:

```
h =
   3*x^2+4*x-12
```

A.5.4 符号微积分

MATLAB 的符号数学工具箱为我们提供了快速、简便地计算微积分的工具,

包括符号极限、符号微分和符号积分等. 其调用格式如下：

limit(f,x,a)	返回符号表达式 f 当 $x\to a$ 时的极限
limit(f,x,a,' left ')	返回符号表达式 f 当 $x\to a$ 时的左极限
limit(f,x,a,' right ')	返回符号表达式 f 当 $x\to a$ 时的右极限
diff(f)	对缺省变量求微分
diff(f,v)	对指定变量 v 求微分
diff(f,v,n)	对指定变量 v 求 n 阶微分
int(f)	对 f 表达式的缺省变量求积分
int(f,v)	对 f 表达式的 v 变量求积分
int(f,v,a,b)	对 f 表达式的 v 变量在 (a,b) 区间求定积分

A.5.5 符号函数作图

MATLAB 的符号工具箱也为我们作出符号函数的图形提供了方便. 可以通过函数“ezplot”或“fplot”来实现. 其调用格式如下：

ezplot(f)	在默认区间 $[-2\pi,2\pi]$ 绘制 $y=f(x)$ 的函数图
ezplot(f,[a,b])	在区间 $[a,b]$ 上绘制 $y=f(x)$ 的函数图
ezplot(x,y,[tmin,tmax])	绘制由参数方程 $x=x(t),y=y(t)$，$t_{\min}\leqslant t\leqslant t_{\max}$ 表示的曲线
ezpolar(f,[a,b])	绘制由极坐标方程 $r=f(\theta),a\leqslant\theta\leqslant b$ 表示的曲线

ezplot3,ezmesh,ezsurf,ezcontour 类似.

§ A.6 程序设计——M 文件的编写

MATLAB 软件不仅具有强大的数值计算、符号计算、绘图功能，而且还可以像 C 语言等高级语言一样，进行程序设计，其程序文件以 m 为文件扩展名，称为 M 文件. MATLAB 提供了一个内置的具有编辑和调试功能的 M 文件编辑器，编辑器窗口也有菜单栏和工具栏，可以方便地进行程序设计.

程序结构一般分为顺序结构、循环结构和分支结构 3 种. MATLAB 也提供了这 3 种程序结构，可以构造功能强大的程序.

A.6.1 M 文件简介

当你要解决一个具体的问题，要求执行的命令数比较多，或要改变变量的值后，重新执行一系列的命令时，在 MATLAB 命令窗口键入命令，逐行执行，就非常麻烦. 此时可进入程序编辑窗口编写 MATLAB 程序，即 M 文件.

M 文件有两类，即脚本式 M 文件和函数式 M 文件，脚本式 M 文件就是命令行的简单罗列，它与批处理文件很相似. 而函数式 M 文件的第一句是以 function 语句作为引导的.

两类 M 文件的相同之处在于它们都是有 m 扩展名的文本文件.

1. 脚本式 M 文件

在通信方面，脚本式 M 文件中的命令可以访问 MATLAB 工作空间中的所有变量，脚本式 M 文件运行结束，所产生的变量保留在工作空间，都是全局变量，直到关闭 MATLAB 或用命令删除. 下面是一个命令文件的例子.

程序：

```
%  文件名 example.m
x = 4;y = 6;z = 2;
items = x + y + z
cost = x * 25 + y * 22 + z * 99
average_cost = cost/items
```

当这个文件在程序编辑窗口输入并以名为 example. m 的 M 文件存盘后，只需简单地在 MATLAB 命令编辑窗口键入 example 即可运行，并显示同命令窗口输入命令一样的结果.

1）在 M 文件中对程序的注释是以符号“%”开始直到该行结束的部分，程序执行时会自动忽略.

2）M 文件的文件名必须以字母开头（不能以数字等开头），后面可跟字母、数字或下划线（不能跟减号，小数点等）.

上例运行结果如下：

```
example
items =
       12
cost =
      430
average_cost =
```

```
35.8333
```

用户可以重复打开 example. m 文件,改变 x,y,z 的值,保存文件并让 MATLAB 重新执行文件中的命令. 若你把 example. m 文件放在自己的工作目录下,那么在运行 example. m 之前,应该先使该目录处于 MATLAB 的搜索路径范围. 可以选择"File"菜单下的"Set Path"项,打开路径浏览器把该目录永久保存在 MATLAB 搜索路径中. 也可把自己的工作目录设置为当前目录.

2. 函数式 M 文件及其调用

MATLAB 本身的许多库函数就是函数式 M 文件,我们还可以根据需要在 MATLAB 编辑窗口建立自己的函数式 M 文件,它们能够像库函数一样方便地调用,从而可扩展 MATLAB 的功能. 如果对于一类特殊的问题,建立起许多函数式 M 文件,就能形成工具箱.

函数 M 文件的第一行有特殊的要求,其形式必须为:

```
function[输出变量列表] = 函数名(输入变量列表)
函数体语句;
```

输出变量相当于函数的因变量,而输入变量相当于函数的自变量. 例如,一个只有两行的函数式 M 文件:

```
function f = fun(x)
f = 100 * (x(2) - x(1)^2)^2 - (1 - x(1))^2;
```

一旦该函数式 M 文件建立,在 MATLAB 的命令窗口或别的 M 文件里,就可用下列命令调用:

```
x = [2,3];f = fun(x)
```

结果为:

```
f =
   99
```

该函数的自变量为 x,是一个长为 2 的向量,因变量为 f.

函数 M 文件的文件名最好与其函数名相同,否则,调用时容易出错. 若函数名与文件名不相同时,调用时要用文件名.

又如:

```
function[F,G] = fun2(x)
F = 2 * x(1)^2 + 2 * x(2)^2 - 2 * x(1) * x(2) - 4 * x(1) - 6 * x(2);
G = [x(1) + 5 * x(2) - 5   2 * x(1)^2 - x(2); - x(1)   - x(2)];
```

可用下列命令调用:

```
x1 = [4,5];[F1,G1] = fun2(x1)
```

结果为

```
F1 =
     -4
G1 =
     24   27
     -4   -5
```

1）当函数无输出参数时，输出参数项空缺或者用空的中括号表示.如：

```
function printresults(x)     或
                function [ ] =printresults(x)
```

2）函数M文件只能访问输入变量，不能访问MATLAB工作空间中的其他变量，它的所有中间变量除特别申明外，均为局部变量，它们在自己专有的工作空间工作.只有输入、输出变量才保留在MATLAB工作空间.

A.6.2 运算符

MATLAB的运算符可分为三类：算术运算符、关系运算符和逻辑运算符.其中算术运算符的优先级最高，其次是关系运算符，再其次是逻辑运算符.算术运算符在前面已经介绍，这里只介绍关系运算符和逻辑运算符.

1. 关系运算符

关系运算符对于程序的流程控制非常有用.MATLAB共有六个关系运算符，它们是

“<”小于；“<=”小于等于；“>”大于；“>=”大于等于；

“==”等于；“~=”不等于.

关系运算符可以比较同型矩阵，此时将生成一个0-1矩阵，当相应元素经关系运算为真时，对应位置上生成1，否则为0.关系运算符也可以比较标量和矩阵，此时是标量与矩阵的每个元素分别比较，生成一个0-1矩阵.

2. 逻辑运算符

MATLAB共有三个逻辑运算符：与(&)、或(|)、非(~).

对于数值矩阵，当元素为0时，逻辑上为假；当元素为非0时，逻辑上为真.同关系运算一样，逻辑运算符两端的运算数可以是同型矩阵，对两矩阵的相应元素分别运算，结果为一个0-1矩阵.当逻辑表达式的值为真时，赋值1，否则为0.同样，另一个矩阵也可以是标量.

与(&)运算　两运算数都为真时，结果为真，其他情况下(一真一假或两个都假)结果为假.

或(|)运算　两个运算数都为假时,结果为假,其他情况下(一真一假或两个都真)为真.

非(~)运算　只有一个运算数,当该运算数为真时,结果为假,否则,结果为真.

A.6.3　循环结构

在实际计算中,经常会碰到许多有规律的重复计算,此时就要对某些语句进行重复执行.一组被重复执行的语句称为循环体,每个循环语句都要有循环条件,以判断循环是否继续.MATLAB 的循环语句有 for 循环和 while 循环两种.

1. for **循环**

for 循环允许一组命令以固定的和预定的次数重复.for 循环的一般形式为

```
for x = expression(表达式)
    statements;(执行语句)
end
```

其中表达式的值为数组,在 for 与 end 之间的执行语句是按该数组中的每一列执行一次,即在每次循环中,数组的列向量一个一个地被赋给变量 x,由执行语句执行.该循环体的执行过程如图 A.12 所示.

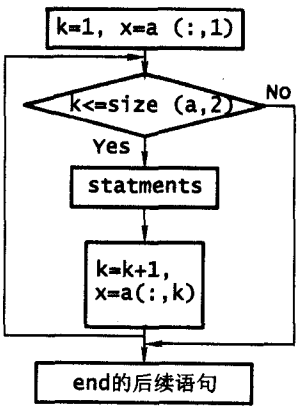


图 A.12

例如,程序:

```
for k = 1:4
      x(k) = 1/k;
end
```

```
format rat  %设置输出格式为有理数
 x
```

将输出：

```
x =
   1   1/2   1/3   1/4
```

1）for 语句可以嵌套使用，即循环体内又包含另一个完整的循环结构，内嵌的循环中还可以嵌套循环，形成多层循环.

2）当有一个等效的数组方法来解给定的问题时，应避免用 for 循环.例如，上例可被重写为

```
n = 1:4;    x = 1./n;  format rat;  x
```

这种方法执行更快，要求较小的输入.

3）不能在 for 循环体内重新对循环变量 k 赋值来终止循环的执行.

2. while **循环**

与 for 循环固定的次数执行一组命令相反，while 循环一般用于事先不能确定循环次数的情况.while 循环的一般形式为

```
while 关系表达式
    statements;(执行语句)
end
```

只要关系表达式的值为 1(真)，就执行 while 与 end 之间的语句体，直到表达式的值为 0(假)时终止该循环.通常，表达式的值为标量，但对数组值也同样有效，此时，数组的所有元素都为真，才执行 while 与 end 之间的语句体.while 循环的执行过程如图 A.13 所示.

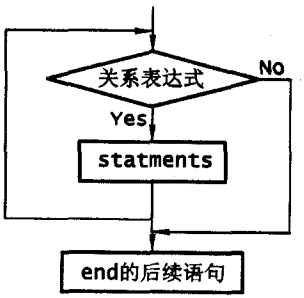


图 A.13

程序：

```
n = 0;EPS = 1;
while(1 + EPS) > 1
    EPS = EPS/2;n = n + 1;
end
n,EPS = EPS * 2
```

运行结果：

```
n =
        53
```

```
EPS =
    2.2204e-016
```

这个例子给出了计算 MATLAB 的特殊变量 eps 的一种方法，eps 是一个可加到 1，在计算机的有限精度下，而使结果大于 1 的最小数值. 这里我们用大写 EPS，以便与 eps 相区别. EPS 从 1 开始，不断被 2 除，直到 EPS + 1 不大于 1. 因为 MATLAB 用 16 位数来表示数据，因此，当 EPS 接近 10^{-16} 时，它会认为 EPS + 1 不大于 1，于是 while 循环结束.

A.6.4 分支结构

在程序设计中，经常要根据一定的条件来执行不同的语句. 当某些条件满足时，只执行其中的某个语句或某些语句. 此时，就可用分支结构来实现. MATLAB 的分支结构有 if - else - end 结构和 switch - case - end 结构两种.

1. if - else - end 结构

if - else - end 结构的最简单形式为

```
if 关系表达式
    语句体
end
```

如果关系表达式的值为真，则执行 if 与 end 之间的语句体，否则，执行 end 的后续命令.

if 结构的另一种形式

```
if 关系表达式
    语句体 1
else
    语句体 2
end
```

如果关系表达式的值为真，则执行语句体 1，然后跳出该选择结构，执行 end 的后续语句；如果关系表达式的值为假，则执行语句体 2，之后，执行 end 的后续语句.

当有三个或更多的选择时，可采用 if 结构的下列形式

```
if 关系表达式 1
    语句体 1
elseif 关系表达式 2
    语句体 2
...
elseif 关系表达式 n
    语句体 n
```

```
else
    语句体 n+1
end
```

如果关系表达式 j(j=1,2,…,n)为真,则执行语句体 j,然后执行 end 的后续语句.否则,当 if 和 elseif 后的所有关系表达式的值都为假时,执行语句体 n+1,然后执行 end 的后续语句.例如,可用以下程序得到符号函数.

```
function y = fuhao(x)
if  x<0
    y = -1;
elseif x = =0
    y=0;
else
    y=1;
end
```

可用 if 和 break 语句来跳出 for 循环和 while 循环.例如,程序:

```
EPS=1;
for n=1:1000
    EPS=EPS/2;
    if(1+EPS)<=1;
      EPS=EPS*2;break
end,end
n,EPS
```

其运行结果同前面一样,n=53,EPS=2.2204e-016.

在此例中,当执行 break 语句时,MATLAB 跳到循环外的下一个语句.如果一个 break 语句出现在一个嵌套的 for 循环或 while 循环里,那么只跳出 break 所在的那个循环,不跳出整个嵌套结构.

2. switch-case-end 结构

switch 语句根据表达式的值来执行相应的语句,一般形式为

```
switch  表达式(标量或字符串)
case  值1,
    语句体1
case  {值2.1,值2.2,…}
    语句体2
…
otherwise,
    语句体n
```

```
end
```

当表达式的值为值 1 时,执行语句体 1,然后执行 end 的后续语句;当表达式的值为{值 2.1,值 2.2,…}中之一时,执行语句体 2,然后执行 end 的后续语句;……;若表达式的值不为任何关键字"case"所列的值时,则执行语句体 n,接着执行 end 的后续语句.注意:只执行一个语句体,然后就执行 end 的后续语句.例如,假设 NAME 是一个字符串变量,下列程序将在 NAME 取值为各种不同字符串的情形下,显示相应的信息.

```
switch lower(NAME)
case{' zhanghua ',' lijiang '}
  disp(' He comes from China.')
case  ' peter '
  disp(' He comes from United States.')
case  ' monika '
  disp(' She comes from Germany ')
otherwise
  disp(' He or she comes from other countries.')
end
```

表 A.2

里程/km	s < 250	250 < = s < 500	500 < = s < 1000
折扣	0	2%	5%
里程/km	1 000 < = s < 2 000	2 000 < = s < 3 000	3 000 < = s
折扣	8%	10%	15%

例 A.7 运输公司计算运费的方式是,距离(s)越远,每公里运费越低.标准见表 A.2,编写一个求折扣的函数式 M 文件.

我们先用 if - else - end 结构来编写运费折扣的函数式 M 文件.

程序:

```
function g = zhekou1(s)
if s < 250
    g = 0;
elseif s < 500
    g = 0.02;
elseif s < 1000
    g = 0.05;
elseif s < 2000
    g = 0.08;
elseif s < 3000
```

```
    g=0.1;
else
    g=0.15;
end
```

再用 switch-case-end 结构来实现同样的功能.

```
function g=zhekou2(s)
switch fix(s/250)
case{0}
    g=0;
case{1}
    g=0.02;
case{2,3}
    g=0.05;
case{4,5,6,7}
    g=0.08;
case{8,9,10,11}
    g=0.1;
otherwise
    g=0.15;
end
```

§A.7 操　　练

操练一　数组操作及运算练习

1) 设有分块矩阵 $\boldsymbol{A}=\begin{bmatrix}\boldsymbol{E}_{3\times3} & \boldsymbol{R}_{3\times2}\\ \boldsymbol{O}_{2\times3} & \boldsymbol{S}_{2\times2}\end{bmatrix}$,其中 $\boldsymbol{E},\boldsymbol{R},\boldsymbol{O},\boldsymbol{S}$ 分别为单位阵、随机阵、零阵和对角阵,试通过数值计算验证 $\boldsymbol{A}^2=\begin{bmatrix}\boldsymbol{E} & \boldsymbol{R}+\boldsymbol{RS}\\ \boldsymbol{O} & \boldsymbol{S}^2\end{bmatrix}$.

2) 某零售店有 9 种商品的单件进价(单位:元)、售价(单位:元)及一周的销量如表 A.3,问哪种商品的利润最大,哪种商品的利润最小. 按收入由小到大,列出所有商品及其收入. 求这一周该 10 种商品的总收入和总利润.

表 A.3

货号	1	2	3	4	5	6	7	8	9
单件进价/元	7.15	8.25	3.20	10.30	6.68	12.03	16.85	17.51	9.30
单件售价/元	11.10	15.00	6.00	16.25	9.90	18.25	20.80	24.15	15.50
销量	568	1 205	753	580	395	2 104	1 538	810	694

操练二　作图练习

1）用两种方法在同一个坐标下作出 $y_1=x^2, y_2=x^3, y_3=x^4, y_4=x^5$ 这四条曲线的图形，并要求用两种方法在图上加各种标注.

2）用 subplot 分别在不同的坐标系下作出下列四条曲线，并为每幅图形加上标题.

① 概率曲线 $y=e^{-x^2}$；

② 四叶玫瑰线 $\rho=\sin 2\theta$；

③ 叶形线 $\begin{cases} x=\dfrac{3t}{1+t^3}, \\ y=\dfrac{3t^2}{1+t^3}; \end{cases}$

④ 曳物线 $x=\ln\dfrac{1\pm\sqrt{1-y^2}}{y}\mp\sqrt{1-y^2}$.

3）作出下列曲面的 3 维图形，

① $z=\sin(\pi\sqrt{x^2+y^2})$；

② 环面：$\begin{cases} x=(1+\cos u)\cos v, \\ y=(1+\cos u)\sin v, \\ z=\sin u, \end{cases}$ $\begin{matrix} u\in(0,2\pi) \\ v\in(0,2\pi) \end{matrix}$.

操练三　符号运算与编写 M－文件

1）求函数的极限、导数或积分：

① $\lim(x+3^x)^{\frac{1}{x}}$，当 $x\to\infty$ 时；　② $\lim\dfrac{e^x\sin x-x(x+1)}{x^3}$，$x\to 0$；

③ $f(x)=\dfrac{x^2+2x-1}{e^{-x}\sin x+1}$，求 $f'(x)$；　④ 已知 $f(x)=\dfrac{x^2}{1-x^2}$，求 $f^{(n)}(0)$；

⑤已知 $\arctan\dfrac{y}{x}=\ln\sqrt{x^2+y^2}$，求 $\dfrac{dx}{dy}$；

⑥ $z=x\arctan y$，求 $\dfrac{\partial z}{\partial x},\dfrac{\partial z}{\partial y}$，画函数图；　⑦ $\int\dfrac{e^{2x}}{e^x+2}dx$.

2）建立一个命令 M 文件：求所有的“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字的立方和等于该数本身. 例如，153 是一个水仙花数，因为

$153 = 1^3 + 5^3 + 3^3$.

3）编写函数 M 文件 Kaif. m：用迭代法求 $x = \sqrt{a}$ 的值. 求平方根的迭代公式为

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right),$$

迭代的终止条件为前后两次求出的 x 的差的绝对值小于 10^{-5}.

更多的相关信息资源

1 （美）D. Hanselman, B. Littlefield 著，张航 黄攀译. 精通 Matlab 6. 北京：清华大学出版社，2002

2 王学辉等. MATLAB 6.1 最新应用详解. 北京：中国水利水电出版社，2002

3 苏金，阮沈勇编著. MATLAB6.1 实用指南. 北京：电子工业出版社，2002

4 http://www. matlab-world. com/