

工程结构电算与 OOP 编程

谭也平 著



中国建材工业出版社

图书在版编目(CIP)数据

工程结构电算与 OOP 编程/谭也平著. —北京:中国建材
工业出版社, 1999. 7

ISBN 7-80090-935-2

I. 工… II. 谭… III. 建筑结构—结构计算: 计算机辅助
计算—程序设计 IV. TU311.41

中国版本图书馆 CIP 数据核字(1999)第 34633 号

中国建材工业出版社出版

(北京市海淀区三里河路 11 号)

北京后沙峪印刷厂印刷

新华书店经销

1999 年 7 月第 1 版

1999 年 7 月第 1 次印刷

开本: 787×1092 毫米

1/16 印张: 19.875

字数: 534 千字

印数 1—2000 册

定价: 29.80 元

序 言

结构静力分析始终是工程设计和施工中的基本问题,电算是解决结构分析问题的基本手段。结构电算涉及一系列的知识领域,如线性代数、数值方法、结构力学、弹性力学以及计算机编程、软件集成等。本书汇集了这些领域中有用的知识,做了深入浅出的论述,并将分析中所需要的数学模型完整地表达出来,可以直接编制计算机分析程序。

面向对象的程序设计(OOP)是当今软件设计的主流方法,本书作者在这方面做了许多有益的尝试。最近二十年来,结构电算方面出版了不少优秀的教材和著作,但是采用 OOP 编程的还是第一次。使用 OOP 方法有结构化程序设计所不可比拟的优点,用它编程序易于开发和维护,用其编制的结构分析系统更容易集成到大型的 CAD/CAE 系统中,是一种值得推广的方法。

目前计算机应用正向网络计算迁移,信息和知识在国际互联网络上的传播在二十一世纪将成为知识传播的主要方式。本书作者打算将本书大部分的资源在网络上发布,并建立和维护一个结构分析的资源网站,便于读者下载所需的源程序,查阅相关的资料,这样是把工程软件成果真正服务于生产的一个很好的开端。

本文作者谭也平博士在同济大学攻读学位期间曾作为我的科研助手从事桥梁结构 CAD 和桥梁结构空间分析的研究工作,并参与了多项国家与省部级科研项目的工作,作出许多贡献,取得了很多成果,特别是对于软件编程的方法有其独到之处。在承担研究工作同时,也参与了一系列的实际的工程项目的设计与研究,积累了不少工程分析的经验。也平到深圳大学工作以后,继续从事结构力学、弹性力学、结构电算和有限元分析方面的教学与研究,同时参与和主持了城市立交等方面工程软件的编制工作。本书的出版是他在这些方面所作工作的一个总结。

本书是一种用面向对象的程序设计的新结构电算类的好教材,就本书的内容和深度而言,它适合于作为土木工程学科本科生和研究生的结构电算和有限元分析教科书或教学参考书,也可以供从事工程设计、施工、教学与研究的人员参考。

石洞

1999. 7. 21.

前 言

结构电算的核心内容是结构有限元分析,有限元分析是一种通用的数值解法,它的应用远远不只限于工程结构领域。但本书将内容着重在结构物的应力分析与结构力学问题的求解。有限元分析的列式以及有限元程序的编制方法大体上是相同的,所以我们可以从较直观的杆系结构入手。

本书尽可能将有限元的理论基础、计算机编程技术和有限元分析的实际应用有效地结合起来。有限元结构分析可以进行线性静力分析、动力分析、非线性与稳定问题分析等等。近年来,计算机在工程领域的应用取得了前所未有的发展,计算机应用的平台从文本界面向图形界面迁移,编程的手段从结构化编程向面向对象的编程手段迁移,结构分析的前后处理也从半自动向图形交互式的手段迁移,本书中采用的基本方法和手段都将尽可能地向这些目标靠拢。由于时间和篇幅的问题,本书仅涉及线性静力问题。

本书所需的基础知识并不多,读者只需掌握结构力学和材料力学的基础知识,对应变位移关系和应力应变关系有初步的概念即可,要求的数学基础知识也以大学低年级课程为限。至于弹性理论、连续体力学、板壳理论、能量方法和数值分析方面的知识只涉及其最基本的部分,而且只要必须,在本书中都详细解说。

本书所使用的编程语言为 C++,它是目前最流行的面向对象的编程语言,C++以 C 语言为基础,增加了对类和对象的支持,从而使它更适合于向量运算和工程计算。C 语言作为一门中级语言比 Fortran 等高级语言较难掌握,所以即使 C 语言是几乎所有专业程序员都使用的语言,C 语言也未成为大学工科的基本教学语言。但是凭作者的经验,我们有了一门高级语言的基础,再掌握 C 语言是容易的,所以将 C++作为基本的有限元编程语言是必须和恰当的。

早期使用结构化程序设计语言编制有限元程序,结构化程序设计语言的一个显著特征是程序和数据的分离,它可以通过模块化的技术把某些特定的过程和数据与程序的其它部分隔离开来,从而避免某些影响全局化使得程序维护的难度增大。使用子程序和局部变量是实现结构化编程的基本手段。但是,当程序的代码达到 25,000~100,000 行时,采用此方法设计的程序会难于理解和维护,使得开发组陷入困境。面向对象的程序设计方法是在结构化程序设计的基础上发展起来的一种先进的程序设计方法。对象是一个既包含数据(数据成员)又包含处理数据的过程(函数成员)的逻辑实体,同时,对象中的数据和过程支持存取控制,这样为对象提供了保护机制。面向对象的程序设计支持对象的继承、动态编连和多态性。继承的特性使得我们可以使用已有的对象,使得代码可以重用。多态性使得我们可以改变已有对象的某些特性,来满足我们特殊的需要。而动态编连的特点使得我们不必知道对象的细节就能使用对象,这一点在结构化编程中我们是做不到的。由于面向对象的编程方法有如此多的优点,所以我们可以编制超过 300,000 行代码的程序而又不会感到维护困难。

本书主要讨论结构静力分析问题,并着重于两点,即有限元的基本概念与编程的基本技巧和有限元的实际应用。本书中所使用的代码大部分是在 1994 年~1995 年作者开发

大型非线性分析有限元程序时编制的，在 Win95/NT 平台下开发和调试，原程序分析过一系列实际的工程问题。本书所提供的代码是一个具有针对性的精炼版本。本书可以提供给土木工程、工程力学和机械工程专业本科高年级学生和研究生作为有限元与结构电算课程的教材或教学参考书，也可以供工程技术人员参考。

本书中所附录的代码读者可以通过本人的电子邮件 tan_yeping_szu@163.net 或本书的主页 <http://fea.yeah.net> 免费获取。

张明峭、张秀丽对本书作了校正工作，深圳大学土木系 96 级学生为编制本书中的程序提供了有益的建议、并试用程序，在此表示感谢。

由于编写的时间仓促，文本和程序代码的错误在所难免，欢迎交流使用和批评指正。

作者

1999 年 3 月于深圳大学

目 录

第一章 矩阵与线性代数基础.....	1
§ 1-1 矩阵代数.....	1
§ 1-1-1 矩阵的定义.....	1
§ 1-1-2 矩阵的加减与数乘.....	3
§ 1-1-3 矩阵的转置.....	4
§ 1-1-4 特殊的矩阵.....	5
§ 1-1-5 矩阵相乘.....	7
§ 1-1-6 矩阵的行列式.....	8
§ 1-1-7 矩阵的逆.....	11
§ 1-1-8 矩阵的分块.....	13
§ 1-1-9 矩阵类的定义.....	14
§ 1-2 向量的概念与运算.....	16
§ 1-2-1 向量的定义及运算.....	16
§ 1-2-2 向量类的定义与使用.....	19
第二章 线性方程组的求解.....	22
§ 2-1 直接法求解线性方程组.....	22
§ 2-1-1 高斯消去法.....	22
§ 2-1-2 高斯—约当消去法.....	24
§ 2-2 迭代法求解线性方程组.....	27
§ 2-2-1 雅可比迭代法.....	28
§ 2-2-2 高斯—赛德尔迭代法.....	33
§ 2-2-3 逐次松弛迭代法.....	34
§ 2-3 矩阵求逆.....	36
§ 2-3-1 高斯消去法矩阵求逆.....	36
§ 2-3-2 矩阵分块求逆.....	40
§ 2-4 矩阵三角化分解.....	43
§ 2-4-1 杜里特尔分解.....	43
§ 2-4-2 LDLT 分解法.....	47
§ 2-4-3 三角矩阵求逆.....	53
§ 2-5 对称矩阵的一维存储与求解.....	54
§ 2-5-1 对称矩阵的一维存储.....	54
§ 2-5-2 稀疏矩阵的一维存储与求解.....	56
第三章 轴力杆系分析.....	64
§ 3-1 有限元的基本概念.....	65

§ 3-1-1 刚度和柔度	65
§ 3-1-2 刚度矩阵	66
§ 3-1-3 结构刚度矩阵	67
§ 3-2 铰接桁架的静力分析	68
§ 3-2-1 一维杆单元的刚度矩阵	68
§ 3-2-2 二维杆单元的刚度矩阵	69
§ 3-2-3 平面铰接桁架计算示例	72
§ 3-2-4 直接刚度法平面桁架分析程序	82
§ 3-2-5 三维桁架的单元刚度矩阵	97
§ 3-2-6 空间铰接桁架分析	99
§ 3-2-7 空间铰接桁架分析的程序设计	103
第四章 通用杆系结构静力分析	104
§ 4-1 平面连续梁单元的刚度矩阵	104
§ 4-2 平面框架单元的刚度矩阵	109
§ 4-3 单元的等效节点力	113
§ 4-4 平面框架分析	115
§ 4-5 半带宽、对角元地址与方程号	125
§ 4-6 通用杆件分析有限元程序设计	128
§ 4-6-1 端点力的释放	129
§ 4-6-2 有限元数据的表示	134
§ 4-6-3 程序数据结构设计	140
§ 4-6-4 通用平面杆系程序设计	147
§ 4-6-5 程序的实现细节	153
§ 4-7 算例	171
§ 4-8 空间杆件的分析	177
§ 4-8-1 杆件的扭转刚度矩阵	177
§ 4-8-2 弯曲刚度矩阵	178
§ 4-8-3 空间一般杆件的刚度矩阵	179
§ 4-8-4 坐标变换矩阵	180
第五章 线弹性体有限元	184
§ 5-1 弹性体势能与虚功原理	183
§ 5-2 有限元基本方程	184
§ 5-3 基于 Hermite 插值的单元	187
§ 5-3-1 梁单元	188
§ 5-3-2 桁单元	190
§ 5-3-3 常应变平面三角形单元	192
§ 5-4 基于 Lagrange 插值的单元	197
§ 5-4-1 拉格朗日插值函数与等参数单元的形函数	197

§ 5-4-2 等参数单元	204
§ 5-4-3 数值积分法	209
§ 5-5 坐标变换	212
§ 5-6 可变节点四边形平面单元	217
§ 5-7 可变节点三维体单元	219
§ 5-8 平板弯曲单元	221
§ 5-9 等参数桁元	224
§ 5-10 等参数梁元	227
§ 5-11 等参数板壳单元	232
第六章 有限元软件集成技术	238
§ 6-1 有限元与数据库的结合	238
§ 6-1-1 引言	238
§ 6-1-2 数据库接口的设计	242
§ 6-1-3 数据库设计	244
§ 6-2 基于图形环境的前处理	248
§ 6-2-1 引言	248
§ 6-2-2 基本数据结构	249
§ 6-2-3 人机界面设计的要求	252
§ 6-3 基于 AutoCAD 有限元前处理系统的实现	253
§ 6-3-1 基本思路	253
§ 6-3-2 前处理中的逻辑结构	253
§ 6-3-3 前处理程序的工作界面	256
附录 I 向量、矩阵运算和方程求解源代码	259
I-1 向量定义与运算	259
I-1-1 向量类定义	259
I-1-2 向量类的实现	260
I-1-3 矩阵类的定义	261
I-1-4 矩阵类的实现	263
I-1-5 方阵类的定义	266
I-1-6 下三角存储对称矩阵类的定义	266
I-2 方程求解	267
I-2-1 高斯消去法	267
I-2-2 高斯约当消去法	268
I-2-3 矩阵求逆	268
I-2-4 LR 分解	271
I-2-5 Cholesky 分解	272
I-2-6 雅可比迭代法	274
I-2-7 Gauss-Seidel 迭代法	275

I-2-8 逐次松弛法	276
I-3 一维条带矩阵	277
I-3-1 一维存储条带矩阵类的定义	277
I-3-2 一维存储条带矩阵类的实现	277
附录 II 轴力杆件分析的源代码	284
II-1 单元、节点与荷载类的定义	284
II-2 有限元数据的存储空间	285
II-3 单元类的实现	285
II-4 荷载类的实现	286
II-5 高斯消去法求解方程	286
II-6 节点类的实现	287
II-7 桁架有限元主程序	287
附录 III 通用杆系分析的源代码	290
III-1 全局数据类型定义	290
III-2 节点类定义	290
III-3 约束方程类的定义	291
III-4 约束方程类的实现	293
III-5 单元类的定义	293
III-6 荷载类定义	295
III-7 荷载类实现	295
III-8 单元类的实现	296
III-9 对角元类	302
III-A 带宽计算	303
III-B 叠加刚度矩阵	304
III-C 通用杆系分析主程序	305
参考文献	309

第一章 矩阵与线性代数基础

§1-1 矩阵代数

结构有限元分析是实现结构电算的基本手段，矩阵代数和电子计算机的应用是有限元实际应用的基础。因为只有矩阵才能以一种紧凑而清晰的方式把整个求解过程表示出来，并且便于在计算机中表示。本节扼要介绍在本书各个章节中用到的矩阵代数基础知识，特别是结构分析中位移法的矩阵表示。

从公式的推导角度来看，矩阵可以用一种简明的方式用少量的符号处理大量的数据，线性代数的大部分概念可以通过矩阵和向量的运算来表示，我们在本章简要介绍线性代数的基本概念，并且用 OOP 的概念实现他们在计算机中的表示和运算实现方法。

在结构电算中，我们采用有限元方法分析结构大量节点的位移和节点力，我们一旦知道了结构上位移与荷载之间的关系，我们就可以用线性方程组将其表述出来，这样用少量的符号和几行算式就可以描述整个求解过程。从实际应用的角度看，通用计算机非常适合于在计算机内部表示矩阵和向量。我们日常使用的 PC 机全部属于通用计算机的范畴，向量运算计算机尤其适于矩阵运算，某些大型的向量分析机可以把大气环流或地壳运动的数据在计算机内部表示出来，通过运算，可以进行中长期的天气预报分析和地震预报、石油勘测等。

我们可以通过使用商品化的矩阵和矩阵代数软件包做大部分的工作，这些软件包包括 LINPAC, EINPACK, MatLab, MathCAD 等等。但是，为了得到更有效的求解方式，在矩阵运算中仔细研究其运算细节是十分重要的，这需要我们很好的理解矩阵代数的定义和概念才能做到。我们有时还必须将矩阵运算集成到特定的分析软件中，所以我们必须讨论矩阵与向量运算的对象定义。

§1-1-1 矩阵的定义

矩阵的定义是从线性方程组的概念中引申而来的，例如，对于下面的方程：

$$\begin{aligned} 3x_1 + 5x_2 - 4x_3 &= 1 \\ 2x_1 + 4x_2 + 7x_3 - x_4 &= 5 \\ x_1 + 4x_2 + 7x_3 + x_4 &= 8 \\ 2x_2 + 7x_3 - 4x_4 &= 0 \end{aligned}$$

其中 x_1, x_2, x_3, x_4 为未知量。利用矩阵的方法表示，可以得到：

$$[A]\{x\} = \{b\}$$

或

$$Ax = b$$

其中

$$A \text{ 是线性方程组的系数矩阵, 并且: } A = \begin{bmatrix} 3 & 5 & -4 & 0 \\ 2 & 4 & 7 & -1 \\ 1 & 4 & 7 & 1 \\ 0 & 2 & 7 & -4 \end{bmatrix}$$

$$x \text{ 是未知量矩阵, 有: } x = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix}$$

$$b \text{ 是已知量矩阵, 有 } b = \begin{Bmatrix} 1 \\ 5 \\ 8 \\ 0 \end{Bmatrix}$$

一般地, 对于一组变量 x 与 b 之间存在线性关系

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \quad (1.1a)$$

也可以简写成:

$$[A]\{x\} = \{b\} \quad (1.1b)$$

或

$$Ax = b \quad (1.1c)$$

其中

$$A \equiv [A] \equiv \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$x \equiv \{x\} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

$$b \equiv \{b\} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix}$$

由此，我们可以定义：矩阵是有序数排成的数组，其一般形式是 $m \times n$ 个数排成 m 行 n 列，即

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad (1.2)$$

认为这个矩阵是 $m \times n$ 阶的。当矩阵只有一行($m=1$)或一列($n=1$)时，则 A 可以叫做向量，并且分别叫行向量和列向量。在本书中，一般用大写字母表示矩阵，用小写字母表示向量。

由此，我们可以把

$$\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}, \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}, [1 \ 2 \ 3]$$

全部叫做矩阵，其中第一个又叫列矩阵或列向量，第三个叫行矩阵或行向量。

在矩阵 A 中的第 i 行第 j 列的元素可以用 a_{ij} 来表示。利用矩阵的元素我们可以定义矩阵的相等运算：

两个矩阵当其阶数相等，并且每一个对应元素都相等时，则两个矩阵相等。

§1-1-2 矩阵的加减与数乘

两个矩阵 A 与 B 只有当其阶数相等，才可以相加，矩阵的加法是将其对应元素分别相加。如以 a_{ij} 和 b_{ij} 分别表示 A 与 B 的元素，则 $c_{ij} = a_{ij} + b_{ij}$ 表示 C 的一般元素，其中 $C=A+B$ 。例如，计算 $C=A+B$ ，其中

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{bmatrix}$$

$$B = \begin{bmatrix} 3 & 6 & 9 \\ 9 & 1 & 1 \end{bmatrix}$$

得到

$$C = A + B = \begin{bmatrix} 4 & 10 & 16 \\ 11 & 6 & 9 \end{bmatrix}$$

矩阵加减的次序是可交换的, 即 $A+B=B+A$ 。

矩阵 A 与数 α 相乘的结果就是矩阵元素与数的乘积, 如 $B = \alpha A$, 则有 $b_{ij} = \alpha a_{ij}$ 。同

样矩阵数乘的次序是可交换的。即 $\alpha A = A \alpha$ 。例如, 计算 $B = \alpha A$, 其中

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{bmatrix}$$

$$\alpha = 5$$

得

$$B = \begin{bmatrix} 5 & 20 & 35 \\ 10 & 25 & 40 \end{bmatrix}$$

矩阵的减法可以用矩阵的加法与数乘来定义, 如 $C = A - B$ 可以表示为 $C = A + (-1)B$ 。即对应的元素相减。

§1-1-3 矩阵的转置

矩阵转置的定义是按如下方式重新排列矩阵中的元素, 若:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

则有

$$B = A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} \quad (1.3)$$

即:

$$b_{ji} = a_{ij}$$

我们用符号上标 T 来表示矩阵的转置。

§1-1-4 特殊矩阵

若矩阵 A 的行数 m 与列数 n 相等, $m=n$, 则我们把 A 叫 n 阶方阵。若同时存在 $A^T=A$, 则我们把 A 叫对称矩阵, 对称矩阵一定是方阵。例如, 下面的矩阵是一个 3 阶对称方阵。

$$A = \begin{bmatrix} 1 & 3 & 1 \\ 3 & 0 & 3 \\ 1 & 3 & 5 \end{bmatrix}$$

单位矩阵是一种特殊的方阵, 它的特点是除对角元为 1 以外, 其余所有元素全为 0。如下所示:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (1.4)$$

单位矩阵也可以用 Kronecker δ 表示, 其定义为:

$$\delta_{kl} = \begin{cases} 1, & \text{当 } k=l \text{ 时} \\ 0, & \text{当 } k \neq l \text{ 时} \end{cases} \quad (1.5)$$

在实际的计算中, 单位矩阵往往是隐含的, 并不需要用内存空间来表示。

若一个矩阵的对角元之上的元素全为 0, 则此矩阵叫做一个下三角矩阵, 如下式所示:

$$A = \begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \quad (1.6)$$

若一个矩阵的对角元之下的元素全为 0, 则此矩阵叫做一个上三角矩阵, 如下式所示:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \quad (1.7)$$

在有限元系统中用到的结构刚度矩阵大多是对称并且条带分布的，即非 0 元素只分布在对角元的附近，如下所示：

$$\begin{bmatrix} \times & \times & \times & \times & & & & & & & \\ \times & \times & \times & \times & \times & \times & \times & & & & \\ \times & \times & \times & \times & \times & \times & \times & & & & \\ \times & \times & \times & \times & \times & \times & \times & & & & \\ & \times & \times & \times & \times & \times & \times & \times & \times & \times & \\ & \times & \times & \times & \times & \times & \times & \times & \times & \times & \\ & & \times & \times & \times & \times & \times & \times & \times & \times & \\ & & & \times & \times & \times & \times & \times & \times & \times & \\ & & & & \times & \times & \times & \times & \times & \times & \\ & & & & & \times & \times & \times & \times & \times & \\ & & & & & & \times & \times & \times & \times & \\ & & & & & & & \times & \times & \times & \times \end{bmatrix}$$

图 1-1 条带分布的矩阵

上图中，用 $\times$ 表示非 0 元素，用空白表示 0 元素。由于矩阵是对称的，可以用下面的式子表示条带分布的对称矩阵：

$$a_{ij} = 0, \quad j > i + m_A \quad (1.8)$$

其中

$$2m_A + 1$$

是矩阵的带宽。

还有一种特殊的矩阵为正交矩阵，如下式所示的矩阵是平面两维坐标的坐标变换矩阵：

$$T = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

它就是一个正交矩阵。正交矩阵的特点是矩阵的任意一行或一列元素的平方之和为 1，任意两行对应的元素乘积之和为 0，任意两列对应的元素乘积之和也为 0，所以下面的矩阵

$$T' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2}/2 & \sqrt{2}/4 & \sqrt{6}/4 \\ 0 & -\sqrt{2}/2 & \sqrt{2}/4 & \sqrt{6}/4 \\ 0 & 0 & -\sqrt{3}/2 & 1/2 \end{bmatrix}$$

也是一个正交矩阵。

§1-1-5 矩阵相乘

有两个矩阵 A 和 B，求其乘积 $C=AB$ 。

我们可以定义， $A(m \times n)$ 和 $B(n \times l)$ 的乘积为，当 A 的列数与 B 的行数相等时，其乘积 C 的行数与 A 的行数相等，其列数与 B 的列数相等。并且 $C(m \times l)$ 的任何一个元素为：

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \quad (1.9)$$

例如：

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 1 & 1 \end{bmatrix}$$

得到

$$C = AB = \begin{bmatrix} 2 & 1 \\ 5 & 2 \end{bmatrix}$$

由于我们可以将向量看作矩阵，所以上面的定义也适用于矩阵与向量的乘法。如向量与矩阵相乘，矩阵与向量相乘，向量与向量相乘等等。例如，矩阵与列向量相乘得到列向量：

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 2 \\ 1 \end{Bmatrix}$$

行向量与矩阵相乘得到行向量：

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 4 & 5 \end{bmatrix}$$

列向量与行向量相乘得到一个矩阵：

$$\begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{bmatrix}$$

行向量与列向量相乘，得到仅由一个元素的矩阵或标量：

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{Bmatrix} 1 \\ 0 \\ 2 \end{Bmatrix} = [7] \equiv 7$$

我们再来看下面的矩阵的乘积，看一看 AB 与 BA 有何不同，例如：

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

得到

$$AB = \begin{bmatrix} 1 & 4 \\ 3 & 10 \end{bmatrix}$$

$$BA = \begin{bmatrix} 7 & 10 \\ 3 & 4 \end{bmatrix}$$

所以，一般的说，矩阵的乘法是不可交换的。对于乘积的转置，有下面的关系存在：

$$(AB)^T = B^T A^T \quad (1.10)$$

此式可以用矩阵乘法的定义证明。

§1-1-6 矩阵的行列式

对于一个两维方阵 A ，它的行列式为：

$$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

并且

$$|A| = a_{11}a_{22} - a_{12}a_{21} \quad (1.11)$$

若矩阵是三维的，同样有：

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

且：