

高等学校教材

# 鱼雷控制系统

计算机辅助分析

设计与仿真

主 编 徐德民

副主编 严卫生



西北工业大学出版社

● 责任编辑 王夏林  
● 封面设计 高许飞



## 鱼 雷 控 制 系 统

计算机辅助分析 设计与仿真

ISBN 7-5612-1206-2



9 787561 212066 >

ISBN 7-5612-1206-2/TJ-26(课)

定价: 14.00 元

PDG

# 鱼雷控制系统 计算机辅助分析 设计与仿真

主 编 徐德民

副主编 严卫生

西北工业大学出版社

2000年5月 西安

(陕)新登字 009 号

**【内容简介】** 本书是九五国家重点教材《鱼雷控制系统分析与设计》的配套教材。全书共分八章,内容有鱼雷控制系统计算机辅助设计概况、MATLAB 语言的使用与程序设计、利用 SIMULINK 进行建模和仿真的基本方法、MATLAB 控制系统工具箱简介、鱼雷控制系统的计算机辅助分析、鱼雷控制系统的计算机辅助经典设计方法、鱼雷控制系统的计算机辅助现代设计方法、鱼雷制导系统的计算机辅助建模与仿真。为方便读者学习使用,书中提供了大量的图表、数据、计算实例、设计与仿真源程序。

本书主要对象为高等学校鱼雷自动控制专业的学生,也可供鱼雷有关专业的师生以及从事鱼雷研究和设计的工程技术人员参考。

### 图书在版编目(CIP)数据

鱼雷控制系统计算机辅助分析 设计与仿真/徐德民主编. —西安:西北工业大学出版社, 2000. 3

ISBN 7-5612-1206-2

I. 鱼… II. 徐 III. ①鱼雷-控制系统-计算机辅助分析 ②鱼雷-控制系统-计算机辅助设计 ③鱼雷-控制系统-计算机仿真 IV. TJ630. 3-39

中国版本图书馆 CIP 数据核字(1999)第 57507 号

\*  
© 2000 西北工业大学出版社出版发行  
(邮编:710072 西安市友谊西路 127 号 电话:8493844)  
全国各地新华书店经销  
西安福利彩印厂印装  
\*

开本:787 毫米×1 092 毫米 1/16 印张:10.125 字数:243 千字  
2000 年 5 月第 1 版 2000 年 5 月第 1 次印刷  
印数:1—1 000 册 定价:14.00 元

购买本社出版的图书,如有缺页、错页的,本社发行部负责调换。

# 前 言



以往人们在自动控制系统设计过程中都使用笔算或借助于计算器来进行,由于运算量很大,对于大多数初学者来说,很少有人实际地计算出结果和描绘出响应的曲线。后来,人们使用计算机来求解控制的设计问题,不少人花费了很多的时间编制与调试程序,却很难得到满意的结果。MATLAB 软件的诞生为控制系统的分析、设计与仿真提供了强有力的工具。它使控制系统的分析、设计与仿真变得非常简单,也必将使传统的方法产生一种变革。由于该软件具有使用简便、矩阵运算功能强、控制理论与 CAD 应用程序集(或称工具箱)丰富等特点,因而已经成为国际控制领域内最流行的计算机辅助设计软件。该软件的推广应用必将提高鱼雷控制系统的设计水平。

本书根据教学改革的需要及多年的教学实践,结合九五国家重点教材《鱼雷控制系统分析与设计》的内容,介绍了鱼雷控制系统的计算机辅助分析、设计与仿真的一般使用方法。全书共分八章。绪论中介绍了鱼雷控制系统计算机辅助分析、设计与仿真的现状及其意义;第二章讲述了 MATLAB 语言的使用与程序设计;第三章介绍了利用 SIMULINK 进行动态建模和仿真的基本方法;第四章简要介绍了 MATLAB 控制系统工具箱的组成及其使用方法;第五章讲述了鱼雷控制系统的计算机辅助分析,包括稳定性分析、根轨迹分析、频域响应分析和时域响应分析;第六章讲述了鱼雷控制系统的计算机辅助经典设计方法,主要介绍了 PID、超前、滞后及并联反馈控制器的设计方法;第七章介绍了鱼雷控制系统的计算机辅助现代设计方法,主要内容包括极点配置方法、全维和降维观测器设计、线性二次型最优控制器设计等;第八章讲述了鱼雷制导系统的计算机辅助建模与仿真,包括导引系统的建模与仿真及整个制导回路的仿真。

本书是一本介绍 MATLAB 软件在鱼雷控制系统分析、设计与仿真方面应用的专业书,同时又可作为鱼雷自动控制专业学生的教材。我们希望此书的出版能够对从事鱼雷控制系统研究的专业技术人员和专业教师的工作,以及对研究生和大学生的学习给予一定的帮助。本书由徐德民和严卫生分别任主编、副主编,其中,第一章和第六章由徐德民编写;第二、三、四、五、七、八章由严卫生编写;参加编写的还有任章(第七章)、吴旭光(第八章)。在本书的编写和出版过程中得到许多同志的热情帮助和支持,特表谢忱。

由于编者水平所限,再加上时间紧迫,书中错误与不当之处在所难免,恳请读者批评指正。

编 著 者

1999 年 6 月

# 目 录



|                                |    |
|--------------------------------|----|
| 第一章 绪 论                        | 1  |
| 第二章 MATLAB 语言的使用与程序设计          | 3  |
| § 2-1 MATLAB 语言简介              | 3  |
| § 2-2 MATLAB 语言的安装             | 4  |
| § 2-3 MATLAB 语言的基本语句结构         | 5  |
| § 2-4 矩阵的基本运算                  | 8  |
| § 2-5 MATLAB 语言的编程方法           | 11 |
| § 2-6 MATLAB 语言工作空间及变量管理       | 15 |
| § 2-7 MATLAB 语言的输入与输出语句        | 17 |
| § 2-8 MATLAB 语言的绘图功能           | 18 |
| § 2-9 MATLAB 语言下 M 文件及 M 函数的编写 | 23 |
| 第三章 利用 SIMULINK 进行建模和仿真的基本方法   | 25 |
| § 3-1 SIMULINK 的窗口与菜单          | 25 |
| § 3-2 SIMULINK 的模型库简介          | 26 |
| § 3-3 SIMULINK 建模与仿真示例         | 29 |
| § 3-4 用 SIMULINK 进行建模的方法       | 35 |
| § 3-5 S 函数的编写与使用               | 44 |
| § 3-6 仿真运行参数的设置                | 46 |
| 第四章 MATLAB 语言控制系统工具箱简介         | 48 |
| § 4-1 控制系统的数学描述                | 48 |
| § 4-2 模型之间的转换与连接               | 50 |
| § 4-3 模型降阶与模型实现                | 56 |
| § 4-4 模型属性函数                   | 57 |
| § 4-5 时域响应                     | 59 |
| § 4-6 频域响应                     | 61 |
| § 4-7 线性系统的根轨迹分析               | 63 |

|                                      |            |
|--------------------------------------|------------|
| § 4-8 增益选择 .....                     | 64         |
| § 4-9 Riccati 和 Lyapunov 方程求解 .....  | 66         |
| <b>第五章 鱼雷控制系统的计算机辅助分析 .....</b>      | <b>68</b>  |
| § 5-1 鱼雷控制系统的稳定性分析 .....             | 68         |
| § 5-2 鱼雷控制系统的根轨迹分析 .....             | 69         |
| § 5-3 鱼雷控制系统的频域响应分析 .....            | 72         |
| § 5-4 鱼雷控制系统的时域响应分析 .....            | 78         |
| <b>第六章 鱼雷控制系统的计算机辅助经典设计方法 .....</b>  | <b>81</b>  |
| § 6-1 引言 .....                       | 81         |
| § 6-2 系统补偿 .....                     | 82         |
| § 6-3 PID 控制器 .....                  | 82         |
| § 6-4 超前补偿 .....                     | 90         |
| § 6-5 滞后补偿 .....                     | 96         |
| § 6-6 鱼雷侧向运动的控制器设计 .....             | 102        |
| § 6-7 鱼雷纵向运动的控制器设计 .....             | 105        |
| <b>第七章 鱼雷控制系统的计算机辅助现代设计方法 .....</b>  | <b>113</b> |
| § 7-1 概述 .....                       | 113        |
| § 7-2 用极点配置方法设计鱼雷控制系统 .....          | 113        |
| § 7-3 观测器设计 .....                    | 121        |
| § 7-4 降维观测器设计 .....                  | 127        |
| § 7-5 用线性二次型最优控制理论设计鱼雷控制系统 .....     | 134        |
| <b>第八章 鱼雷制导系统的计算机辅助建模与仿真 .....</b>   | <b>140</b> |
| § 8-1 鱼雷与目标的 SIMULINK 建模 .....       | 140        |
| § 8-2 变结构垂直命中导引系统的 SIMULINK 建模 ..... | 142        |
| § 8-3 变结构垂直命中导引系统的仿真 .....           | 149        |
| § 8-4 鱼雷偏航角速率控制系统的仿真 .....           | 152        |
| § 8-5 鱼雷整个制导回路的仿真 .....              | 154        |
| <b>参考文献 .....</b>                    | <b>155</b> |

# 第一章

## 绪 论

鱼雷是一种能在水中自动推进、自动控制和导引的航行器,是打击水面舰船和潜艇的进攻性武器。其中的控制系统是鱼雷的中枢,在制导精度和姿态控制中起着举足轻重的作用。随着计算机软硬件、图形及智能模拟等技术的进步,鱼雷控制系统的计算机辅助分析、设计与仿真(Computer Aided Analysis, Design & Simulation)已成为工程设计及科学研究不可缺少的组成部分。这些技术充分利用计算机的高速运算和处理能力,进行动态系统的建模、分析、设计和仿真,它不仅可以缩短产品的设计周期,减少设计人员的繁杂劳动,而且能够提高产品质量,降低成本。

自动控制作为一门学科出现在 20 世纪 40 年代。开始所采用的数学工具主要是传递函数和频率特性,而且设计主要针对单输入单输出系统。在 20 世纪 60 年代以前,控制系统的分析和设计主要依靠手工计算和借助于一些图表的帮助。因此,在经典控制理论中,人们引入了各种各样的图表和曲线,如 Bode 图、Nyquist 图、Nichols 图及 M 圆图等。由于高阶系统的根轨迹需要多次解出一元方程的根,而一元四次及更高次的方程无直接求根公式,于是许多教科书上都用了不少篇幅来介绍求根轨迹的近似方法。近 30 多年来,现代控制理论发展得特别快,出现了许多分析和设计控制系统的新理论和新方法。如果被研究的对象非常简单,那么还可以用手工来设计控制系统。如果被控对象比较复杂,那就不得不借助计算机来代替手工计算,于是出现了计算机辅助设计。计算机具有很强的计算和数据处理能力,工程技术人员具有丰富的经验及对系统的先验知识,两者的结合将能对控制系统的设计发挥最大的效能。近 20 多年来,国内外陆续建立了不少计算机辅助设计控制系统的软件包。

控制系统计算机辅助设计是一门控制系统理论、计算技术和工程设计相结合的应用科学,又是控制理论联系实际的桥梁。同时,计算机辅助设计控制系统反过来也促进了控制理论以及控制系统设计方法的发展。

本书主要针对鱼雷自动控制系统,研究以下三个方面的内容:

(1) 鱼雷控制系统的计算机辅助分析。对于鱼雷控制系统来说,它的性能好坏将直接影响鱼雷的战技指标,因此要保证鱼雷的性能就必须对鱼雷的控制系统进行分析。主要包括:控制系统的稳定性、能控性、能观性,时域指标(如超调量、调节时间、上升时间等),频域指标(如相位裕度、幅值裕度、频带宽度等)。

(2) 鱼雷控制系统的计算机辅助设计。对于不满足性能指标的鱼雷控制系统来说,就需要对其进行重新设计。主要的设计方法有经典方法、现代方法等。

对于经典设计方法来讲,单变量控制系统的设计方法主要包括利用代数的方法进行系统的频率特性的计算、根轨迹的计算、校正传递函数的计算等。这些方法没有计算机就可进行。然而如果有了计算机的帮助,运用一些比较简单的程序,就可使这些工作做得又快又精确。

对于现代设计方法来讲,利用状态空间设计控制系统主要有两种方法:一种是二次型最优控制设计方法;另一种是极点配置设计方法。

(3) 鱼雷控制系统的计算机仿真。在控制系统的分析和设计这两个阶段都要进行仿真研究。

采用数字计算机进行仿真可以是全数字仿真,即对象和控制器都用数学模型来代替,也可以是半实物仿真,例如,控制器和敏感元件用实际样机,而被控对象用数学模型来代替。由于实物的参与,要求数字计算机有足够高的运算速度,即具有实时计算能力。

以上三个方面的问题贯穿鱼雷控制系统设计的整个过程,缺一不可。为此,本书就针对上述三个步骤向读者逐一进行介绍。

# MATLAB 语言的使用与程序设计

## § 2-1 MATLAB 语言简介

早期的控制系统的设计主要借助于 FORTRAN 语言和 C 语言,虽然从一定程度上减轻了科技人员的工作,但是这些高级语言的编程,不仅需要对各种算法有深刻的了解,还须要熟练掌握所用语言的语法和编程技巧。例如对矩阵求逆这样一个运算,我们首先须要选择一个好的求逆算法,然后用 FORTRAN 语言或 C 语言编程来逐步实现此算法,且不说键入程序的枯燥和费事,等经过了艰苦烦琐的调试工作终于实现算法达到目的后,我们会发现,所编制的百余条甚至几百条语句仅仅是完成了一个矩阵求逆工作,我们不免为自己的工作效率大发感叹。

为了使广大的科技人员从繁重的工作中解脱出来,1980 年前后美国的 Cleve Moler 博士研制的 MATLAB 语言(简称 MATLAB)对后来的控制系统的理论及计算机辅助设计技术起到了巨大的推动作用。值得指出的是该语言原本并不是专门为控制理论领域的学者使用的,其最初目的是为 New Mexico 大学线性代数等课程提供一种方便可行的实验手段,因此取名为 MATtrix LABoratory,即缩写为 MATLAB 语言。它是集命令解释、科学计算于一身的一套交互式软件系统,经过在该大学进行了几年的使用之后,于 1984 年推出了该软件的正式版本。在 MATLAB 语言下,矩阵的各种运算变得异常容易,后来的版本又增添了丰富多采的图形图像处理及多媒体功能,使得 MATLAB 语言的应用范围越来越广泛,Moler 博士等一批数学家与软件专家组建了一个名为 MathWorks 的软件开发公司,专门扩展并改进 MATLAB 语言,并于 1994 年推出了 MATLAB 语言 4.2 版。

为了准确并直观地把一个控制系统的复杂模型输入给计算机,然后对之进行进一步的分析与仿真,1990 年 MathWorks 软件开发公司为 MATLAB 语言提供了新的控制系统模型图形输入与仿真工具,并定名为 SIMULAB,该工具很快在控制界得到了广泛的应用。但因其名字与著名的软件 SIMULA 类似,所以在 1992 年正式改名为 SIMULINK。SIMULINK 是一个用来对动态系统进行建模、仿真和分析的软件包,它支持连续、离散及两者混合的线性 and 非线性系统。SIMULINK 为用户提供了用方框图进行建模的图形接口,采用这种结构画模型就像我们平时用笔和纸来画一样容易。它与传统的采用微分方程和差分方程进行建模的软件包相

比,具有更直观、方便、灵活的优点。SIMULINK 包含有 Source(输入源)、Sinks(输出方式)、Discrete(离散环节)、Linear(线性环节)、Nonlinear(非线性环节)、Connections(连接与接口)6个模型库,每个模型库中又包含有相应的功能模块。用户也可以定制和创建自己的模块。

SIMULINK 的出现进一步完善了 MATLAB 语言的功能,使得控制系统的仿真变得轻松、方便、直观,并为控制系统的仿真及其在 CAD 中的应用打开了崭新的局面。

目前的 MATLAB 语言已经成为国际上最为流行的软件之一,它除了具有传统的交互式编程功能之外,还提供了丰富可靠的矩阵运算、图形绘制、数据处理、图像处理、方便的 Windows 编程等便利工具,出现了各种以 MATLAB 语言为基础的工具箱,可广泛地应用于自动控制、图像信号处理、生物医学、语音处理、信号分析、振动理论、时序分析与建模、化学统计、优化设计等领域,并显现出一般高级语言难于比拟的优势。MATLAB 语言提供的工具箱,主要包括:

- (1) 控制系统工具箱;
- (2) 系统辨识工具箱;
- (3) 鲁棒控制工具箱;
- (4) 多变量频域设计工具箱;
- (5)  $\mu$  分析与综合工具箱;
- (6) 神经网络工具箱;
- (7) 最优化工具箱;
- (8) 信号处理工具箱;
- (9) 模糊推理工具箱;
- (10) 小波分析工具箱;
- (11) 符号运算工具箱;
- (12) 非线性系统设计工具箱。

## § 2-2 MATLAB 语言的安装

早期的 MATLAB 语言是基于 DOS 操作系统,随着软件技术的飞速发展,特别是新的操作系统 WINDOWS 的广泛流行,MATLAB 语言发展成可在 WINDOWS 下运行的版本。目前国内最新的版本为 MATLAB 语言 5.3 版,下面的安装都是基于 MATLAB 语言 5.3 版。

对于已经购买 MATLAB 语言 5.3 版的 PC 用户,如要全部安装 MATLAB 语言,则至少需要 250 MB 以上的硬盘空间,16 MB 以上的内存,486 以上的 CPU,WINDOWS 3.1 以上的操作系统及一个光盘驱动器。

安装时,首先将装有 MATLAB 语言的光盘插入光盘驱动器,运行光盘中的 SETUP 程序,然后按照安装时的提示输入你的姓名、单位、所购买软件的系列号及安装路径后,安装程序将自动安装到用户的硬盘中。安装完毕后,安装程序会自动在用户的计算机桌面上安装 MATLAB 图标,如图 2-1 所示。



图 2-1 MATLAB 图标

用鼠标双击 MATLAB 语言在计算机桌面上的图标,则显示如图 2-2 所示的 MATLAB 语言命令窗口。在 MATLAB 语言命令窗口中,就可以输入有关的 MATLAB 语言命令。

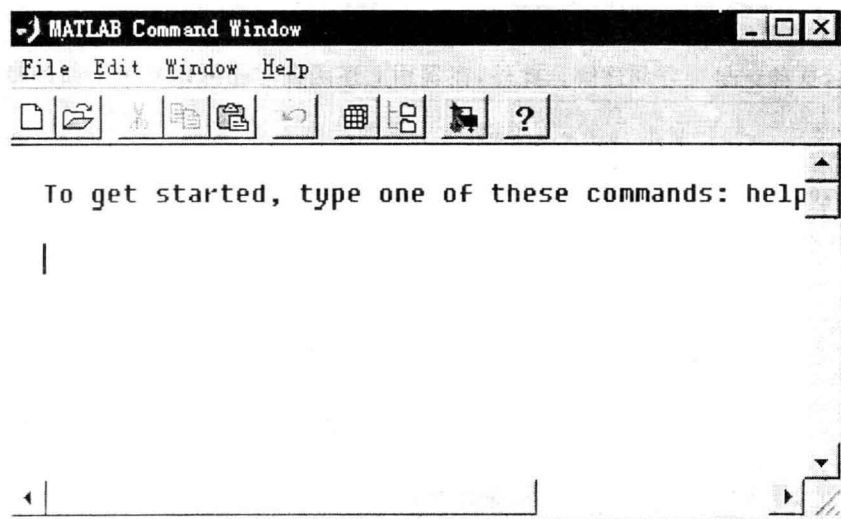


图 2-2 MATLAB 命令窗口

## § 2-3 MATLAB 语言的基本语句结构

MATLAB 语言可以被认为是一种解释性语言,用户可以在 MATLAB 语言的命令窗口中键入一个命令,也可以应用 MATLAB 语言编写一个应用程序,这样 MATLAB 语言软件对此命令或程序中的各条命令进行解释,然后在 MATLAB 语言环境中对它进行处理,最后返回运算结果。MATLAB 语言以复数矩阵为最基本的运算单元,既可以对它进行整体处理,也可以对它的某个或某些元素进行单独处理,所以操作起来比较方便。MATLAB 语言最基本的赋值语句结构为

变量名列表 = 表达式

其中,等号左边的变量名列表为 MATLAB 语言语句的返回值,等号右边的是表达式的定义,它可以是 MATLAB 语言允许的矩阵运算,也可以包含 MATLAB 语言下的函数调用。

(1) 等号右边的表达式可以由分号结束,也可以由逗号或换行号结束,但它们的含义是不同的。如果用分号结束,则左边的变量结果将不在屏幕上显示出来,否则将把左边返回矩阵的内容全部显示出来。

(2) 和 C 语言类似,MATLAB 语言能区分变量大小写。

(3) MATLAB 语言和 C 语言不同,在调用函数时 MATLAB 语言允许一次返回多个结果(也即多个矩阵),这时等号左边是由 [ ] 括起来的矩阵列表,例如:

$[m, p] = \text{bode}(n, d, w)$

中调用了控制系统工具箱中的 bode( ) 函数来求取传递函数分子为 n、分母为 d 在指定的频率段内的幅值响应 m 与相位响应 p,可见调用了这一函数之后,就可以在等号左边同时返回两

个矩阵  $m$  和  $p$ 。

(4) 从前面的例子可以看出, MATLAB 语言在函数调用时输入和输出变量分别在等号的两端列出, 这种记号很容易理解。此外, 在调用上述函数时还可以采用下面的格式。

$$[m,p]=bode(a,b,c,d,1,w)$$

其中,  $a, b, c, d$  为系统的状态方程描述, MATLAB 语言会自动地从输入参数的个数上判定给出的是传递函数还是状态方程模型, 从而进行正确的计算, 这类似于 C++ 语言中重载的概念。

在 MATLAB 语言下, 矩阵输入的方式是很直观的, 如矩阵  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  的输入方式为

$A = [1,2,3;4,5,6;7,8,9]$ , 其中同一行中的内容用逗号分割, 而采用分号来表示换行。按这种格式输入矩阵  $A$  后, 矩阵的内容将按下面的格式在屏幕上显示出来。

```
» A = [1,2,3;4,5,6;7,8,9]
      1  2  3
      A = 4  5  6
           7  8  9
```

如果在上面赋值式子的末尾加一个分号, 则矩阵的内容就不在屏幕上显示, 所以用户可以通过是否在语句末加分号的方式来决定运算的结果是否显示出来, 这样就可以使得不必要的中间结果部分不被显示。

在一般情况下, 用于同行中元素间分割的逗号可以用空格来代替。其实, MATLAB 语言的矩阵输入格式并不是很严格, 在语句中多加一些空格将不会影响整个赋值结果。此外, 前面的矩阵还可以等价地由下面两种方式来输入。

```
» A = [1 2 3;4 5 6           » A = [1 2 3;4,5,...
      7 8 9];                  6;7 8,9];
```

其中, 在前一种输入方法中, 第一行的末尾用一个回车键来表示换行, 它的作用和一个分号是一致的。如果用户采用后面一种输入方式, 则第一行末尾给出的三个点号(...)称为续行符号, 它表示下面的一行应该紧接在前一行上。当一个语句相当长, 难于由一行表示出来时, 往往需要采用这样的续行符号, 将一条语句分割成几行来表示。

向量的输入是矩阵输入的特殊情况, 例如下面两条命令:

```
» V1 = [1 2 3,4]           » V2 = [1;2;3;4]
```

可以简单地给出下面的行向量和列向量。

```
V1 = 1 2 3 4           V2 = 1
                        2
                        3
                        4
```

除了表示向量和矩阵之外, 利用 MATLAB 语言当然可以容易地表示标量。学会了矩阵的基本表示方法之后, 就可以容易地理解下面的输入方式和结果。

```
» A = [A; [1 3 5]]
```

```
A = 1  2  3
      4  5  6
      7  8  9
      1  3  5
```

可见,这种方法的作用是在原来  $A$  矩阵的下面再附加上一个行向量 $[1\ 3\ 5]$ 。如果矩阵各行的元素个数不同时,例如写成  $A = [A; [1\ 2]]$ ,这样 MATLAB 将给出下面的错误信息。

```
» A = [A; [1 2]]
??? All rows in the bracketed expression must have the same
number of columns.
```

提示该矩阵有误(行中元素个数不匹配),并给出一声鸣叫,从而提示用户重新输入矩阵。

MATLAB 语言定义了两个基本的复数常量  $i$  和  $j$ ,这些值为数学上的  $\sqrt{-1}$ 。如果它们不被重新赋值,则将保留这个定义。如果被重新赋值,则这两个量被视为变量,并将保留新的值。如果想把一个变量  $a$  赋成,  $\sqrt{-1}$  则采用 MATLAB 命令  $a = \text{sqrt}(-1)$  即可。

在 MATLAB 语言下  $4+3*i$  的赋值命令将得出  $4.0000+3.0000i$  的结果,而利用前面的  $A$  矩阵,在输入命令  $A+\text{ones}(\text{size}(A))*i$  后可以得出下面的结果。

```
ans =
1.0000 + 1.0000i  2.0000 + 1.0000i  3.0000 + 1.0000i
4.0000 + 1.0000i  5.0000 + 1.0000i  6.0000 + 1.0000i
7.0000 + 1.0000i  8.0000 + 1.0000i  9.0000 + 1.0000i
1.0000 + 1.0000i  3.0000 + 1.0000i  5.0000 + 1.0000i
```

其中,  $\text{ones}(\text{size}(A))$  命令将产生一个和  $A$  矩阵同样大小的、元素全为 1 的矩阵。

一般说来,在 MATLAB 语言环境下矩阵名可以为任意字符串,但是 MATLAB 语言保留了一些特殊的字符串,如判断 0 元素用的误差限  $\text{eps}$ ,其默认值为  $\text{eps} = 2.2204 \times 10^{-16}$ ,  $\text{pi}$  表示圆周率  $\pi$  的值,  $\text{Inf}$  表示无穷大  $\infty$ , MATLAB 语言允许的最大数据为  $1.7976931 \times 10^{308}$ ,一个数据大于此数则被认为是  $\text{Inf}$ 。  $\text{Inf}$  的另一种产生办法为计算  $1/0$  的值,这种运算在一般计算机语言中会被认为是“非法除 0”而将终止整个程序的执行,而 MATLAB 语言依照 IEEE 标准允许这种运算,不会因此产生异常中断,只给出“Warning: Divide by zero”这样的警告信息。 MATLAB 语言中保留的  $\text{NaN}$  常量,它是一个不定式(Not a Number),是由  $\text{Inf}/\text{Inf}$  或  $0/0$  这样的运算产生的。从而可见, MATLAB 语言有着比其它语言更高的容错性。

在变量赋值语句中,等号左边的矩阵名列表和等号一起可以省略,这时将把返回的矩阵名设置为  $\text{ans}$  变量。 $\text{ans}$  也是一个保留的 MATLAB 字符串,它表示上面一个式子的返回结果。例如如果输入  $[1\ 2, 3; 4, 5\ 6]$  将产生下面的输出。

```
» [1 2, 3; 4, 5 6]
ans = 1  2  3
      4  5  6
```

这时 MATLAB 语言会自动将该矩阵赋给  $\text{ans}$  变量,这样用户就可以对  $\text{ans}$  变量进行直接操作。即使在 MATLAB 语言中保留了若干个字符串,它们还可以重新再赋值,例如如果用户想将判 0 用的误差限扩大 10 倍,则可以采用  $\text{eps} = \text{eps} * 10$  命令来进行修正。重新启动 MATLAB 语言后,所有的保留常量将恢复其原有的默认值。

## § 2-4 矩阵的基本运算

如果一个矩阵  $A$  有  $n$  行、 $m$  列元素,则称  $A$  矩阵为  $n \times m$  矩阵,如果  $n=m$ ,则称矩阵  $A$  为方阵。MATLAB 语言定义了下面各种矩阵的基本运算。

(1) 矩阵的加减法运算:假设在 MATLAB 语言工作环境中有两个矩阵  $A$  和  $B$ ,则可以由下面的命令执行矩阵加减法: $C=A+B$  和  $C=A-B$ 。若  $A$  和  $B$  矩阵的维数相同,则可以执行矩阵的加减法,它会自动地使得  $A$  和  $B$  矩阵的相应元素相加减。如果矩阵  $A$  和  $B$  的维数不匹配,则 MATLAB 语言将自动地给出错误信息,提示用户两个矩阵的维数不匹配。

(2) 矩阵的转置:在数学公式中一般把一个矩阵的转置记作  $A^T$ ,假设  $A$  矩阵为一个  $n \times m$  的矩阵,则其转置矩阵  $B$  的元素定义为  $b_{ji}=a_{ij}$ ,  $i=1, \dots, n$ ;  $j=1, \dots, m$ 。若  $A$  矩阵含有复数元素,则对它进行转置时,其转置矩阵  $B$  的元素定义为  $b_{ji}=a_{ij}^*$ ,  $i=1, \dots, n$ ;  $j=1, \dots, m$  也即首先对各个元素进行转置,然后再逐项求取其共轭复数值。在 MATLAB 语言下,矩阵  $A$  的转置(包括复数矩阵)可以简单地由  $A'$  求出。例如下面的命令对原矩阵进行转置。

```
» A = [5+i, 2-i, 1; 6*i, 4, 9-i]; B = A'
```

```
B =
```

```
5.0000 - 1.0000i 0.0000 - 6.0000i
```

```
2.0000 + 1.0000i 4.0000
```

```
1.0000
```

```
9.0000 + 1.0000i
```

(3) 矩阵的翻转处理: MATLAB 语言还提供了一些矩阵的翻转处理的特殊命令,如  $B = \text{fliplr}(A)$  命令将矩阵  $A$  进行左右翻转再赋给  $B$ ,即  $b_{ij}=a_{i,n+1-j}$ ,而  $C = \text{flipud}(A)$  命令将  $A$  矩阵进行上下翻转并赋给矩阵  $C$ ,即  $C_{ij}=a_{j,n+1-i}$ 。  $D = \text{rot90}(A)$  将矩阵  $A$  旋转 90 后赋给  $D$ ,即  $d_{ij}=a_{j,n+1-i}$ ,下面将通过例子来演示各个函数的翻转效果。

**例 2-1** 假设  $A$  矩阵为  $A = [1, 2, 3; 4, 5, 6; 7, 8, 9]$ ,则调用这三个函数分别得出如下

```
» A = [1, 2, 3; 4, 5, 6; 7, 8, 9]
```

```
A = 1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
» B = fliplr(A)
```

```
B = 3 2 1
```

```
6 5 4
```

```
9 8 7
```

```
» C = flipud(A)
```

```
C = 7 8 9
```

```
4 5 6
```

```
1 2 3
```

```
» D = rot90(A)
```

```
D = 3 6 9
```

```
2 5 8
```

```
1 4 7
```

(4) 矩阵乘法:假设有两个矩阵  $A$  和  $B$ ,其中  $A$  矩阵的列数与  $B$  矩阵的行数相等,则称  $A$ 、 $B$  是可乘的,或称  $A$  和  $B$  矩阵维数相容。假设  $A$  为  $n \times m$  矩阵,  $B$  为  $m \times r$  矩阵,若  $C = AB$ ,则  $C$  为  $n \times r$  矩阵,其各个元素为

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}, \quad \text{其中 } i=1, 2, \dots, n \quad j=1, 2, \dots, r$$

例如下面给出的  $A$  和  $B$  矩阵及其乘积矩阵  $C$  为

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 5 \\ 7 & 8 \end{bmatrix}, \quad C = \begin{bmatrix} 19 & 21 \\ 43 & 47 \end{bmatrix}$$

在 MATLAB 语言下,矩阵  $A$  和  $B$  的乘积可以简单地由运算  $C=A * B$  求出

```

>> A = [1, 2; 3, 4]; B = [5, 5; 7, 8]; C = A * B
C =
    19    21
    43    47

```

在这里并不需要指定  $A$  和  $B$  矩阵的维数。如果  $A$  和  $B$  矩阵的维数相容,则可以准确无误地获得乘积矩阵  $C$ ,如果二者的维数不相容,则将给出错误信息,通知用户两个矩阵不可乘。

(5) 矩阵的乘方运算:一个方阵的乘方运算可以在数学上表达成  $A^x$ ,而其前提条件要求  $A$  矩阵为方阵。如果  $x$  为正整数,则乘方式子的结果可以将  $A$  矩阵自乘  $x$  次而得出。如果  $x$  为负整数,则可以将  $A$  矩阵自乘  $x$  次,然后对结果进行求逆运算就可以得出该乘方结果。如果  $x$  是一个分数,例如  $x = n/m$ ,其中  $n$  和  $m$  均为整数,则首先应该将  $A$  矩阵自乘  $n$  次,然后对结果再开  $m$  次方。

在 MATLAB 语言环境下,如果给定一个  $A$  矩阵,则其乘方矩阵和开方矩阵可以容易地由  $A^x$  求出,其中  $x$  为一个常数。仍考虑下面给出的  $A$  矩阵

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

该矩阵的乘方和 0.1 次幂可以由  $A^2$  和  $A^{0.1}$  两条命令求得

```

>> A^2
ans =
    30    36    42
    66    81    96
   102   126   150

>> A^0.1
ans =
  0.8465 + 0.2270i  0.3600 + 0.0579i -0.0967 - 0.1015i
  0.4016 + 0.0217i  0.4523 + 0.0132i  0.4433 - 0.0146i
 -0.0134 - 0.1740i  0.4849 - 0.0509i  1.0131 + 0.0819i

```

(6) MATLAB 的点运算:MATLAB 语言中定义了一种特殊的运算,即所谓的点运算。两个矩阵之间的点运算是该矩阵对应元素的直接运算,例如  $C=A . * B$  表示  $A$  和  $B$  矩阵的相应元素之间的直接乘法运算。然后将结果赋给  $C$ 。注意,点乘积运算要求  $A$  和  $B$  矩阵的维数相同。这种点乘积运算又称为 Hadamard 乘积。可以看出,这种运算和普通乘法运算是不同的,例如对两个简单矩阵  $A$  和  $B$  来说,为

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 1 \end{bmatrix}$$

普通乘法与点乘运算的结果分别为

$$A * B = \begin{bmatrix} 36 & 42 & 21 \\ 81 & 96 & 57 \\ 126 & 150 & 93 \end{bmatrix}, \quad A . * B = \begin{bmatrix} 2 & 6 & 12 \\ 20 & 30 & 42 \\ 56 & 72 & 9 \end{bmatrix}$$

矩阵点运算不光可以用于点乘积运算,还可以用于其它运算的场合。比如对前面给出的  $A$  矩阵作  $A^A$  运算,则将得出下面的结果:

```

»A^A
ans =
      1      4     27
    256    3125   46656
  823543 16777216 387420489

```

在这一例子中,每个矩阵元素都是原来元素的相应乘方。

(6) MATLAB 语言的除法运算:MATLAB 语言定义了除法运算,其意义相当于矩阵的求逆运算。更一般地,MATLAB 语言还定义了矩阵的左除和右除,在这里只给出一般的等效关系。

矩阵的左除:MATLAB 语言中用“\”运算符号表示两个矩阵的左除, $A \setminus B$  即由 gauss 消去法获得线性方程  $AX=B$  的解  $X$ ,也即  $X=A^{-1}B$ 。如果  $A$  矩阵不是个方阵,也可以求出  $A \setminus B$ ,这时将使用最小二乘解法来求取  $AX=B$  中的  $X$  矩阵,用这种方法得出的解可能与使用  $\text{pinv}()$  函数得出伪逆后再求出的结果不一致,所以使用时应该注意。

矩阵的右除:MATLAB 语言中定义了“/”符号,用于表示两个矩阵的右除。 $B/A$  为  $BA^{-1}$ 。

例 2-2 假如  $A$  和  $B$  由下式给出:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \\ 1 & 3 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

则由 MATLAB 语言可以容易地得出  $A/B$  和  $\text{pinv}(A') * B'$

|                                                                                                                            |                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <pre> »A/B ans =       0.0000      0.5000      -3.0000      3.5000     -12.0000    10.2500       1.0000      0.0000 </pre> | <pre> »pinv(A') * B' ans =       0.3559      0.3390      -0.0508      0.2373       0.0000      0.0000       0.8475      0.7119 </pre> |
|----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|

可见二者是不同的,所以在使用伪逆时应该注意。由于维数关系,前面给出的矩阵不能求取  $A \setminus B$ 。如果使用 MATLAB 语言,可以容易地求出  $A' \setminus B'$ 。

```

»A'\B'
ans = 0.0000  0.0000
      0.0000  0.2857
      0.0000  0.0000
      1.0000  0.8571

```

(7) 单个矩阵元素的赋值与运算:MATLAB 语言允许用户对每一个矩阵的单个元素进行赋值和操作,例如如果将前面矩阵  $A$  的第 2 行第 3 列的元素赋为 100,则可以通过下面语句来完成。

```

»A(2,3) = 100
A = 1  2  3
    4  5 100
    7  8  9

```

这时将只改变该元素的值,而不影响其它元素的值。如果给出的行数或列数大于原来矩阵的范围,则 MATLAB 语言将自动扩展原来的矩阵,并将扩展后未赋值的矩阵元素置为 0。