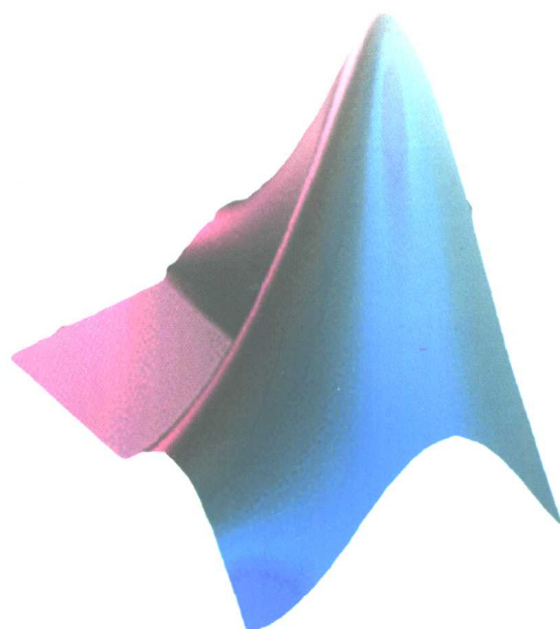
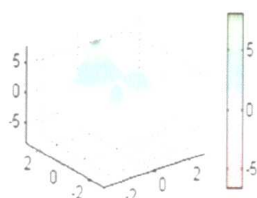




薛定宇 编著



科学运算语言 MATLAB 5.3

程序设计与应用



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



科学运算语言 MATLAB 5.3

程序设计与应用

薛定宇 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书系统地介绍了科学运算语言 MATLAB 的程序设计及应用,以最新的 MATLAB 5.3 版为基础,系统详尽地介绍了它的数据结构与程序设计基础、二维与三维绘图及图像处理、图形界面设计方法与实践以及 MATLAB 语言和其他程序设计语言的接口问题,并介绍了该语言在各个科学运算分支中的应用。

本书可作为一般读者学习和掌握 MATLAB 语言的教科书,也可作为高校理工科各类专业的本科生和研究生现代科学运算课程的教材和参考书,还可供科技工作者、教师作为学习和应用数值分析技术、解决实际问题的参考资料。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

MATLAB, Optimization Toolbox, Symbolic Toolbox 为 The MathWorks 公司的注册商标

MS06/0703

书 名: 科学运算语言 MATLAB 5.3 程序设计与应用

作 者: 薛定宇

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 世界知识印刷厂

发行者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 22 字数: 545 千字

版 次: 2000 年 10 月第 1 版 2000 年 10 月第 1 次印刷

书 号: ISBN 7-302-01226-1/TP·464

印 数: 0001 ~ 5000

定 价: 32.00 元

前 言

MATLAB 语言起源于 1980 年美国学者 Cleve Moler 教授在线性代数领域的早期工作,于 1984 年出现了 MATLAB 的第一个商业版本。在科学运算、自动控制及其他工程领域,国外很多高校在教学与研究中都将 MATLAB 语言作为首选的计算机工具。我国的科学工作者和教育工作者也逐渐认识到 MATLAB 语言的重要性,对 MATLAB 语言在理工科教学与科研中的地位也达成了共识。

虽然 MATLAB 语言是计算数学专家倡导并开发的,但其普及和发展离不开自动控制领域学者的贡献。甚至可以说, MATLAB 语言是自动控制领域学者和工程技术人员捧红的,因为在 MATLAB 语言的发展进程中,许多有代表性的成就和控制界的要求与贡献是分不开的。迄今为止,大多数工具箱也都是控制方面的。MATLAB 具有强大的数学运算能力、方便实用的绘图功能及语言的高度集成性,它在其他科学与工程领域的应用也是越来越广,并且有着更广阔的应用前景和无穷无尽的潜能。

子曰:“工欲善其事,必先利其器”。如果有一种十分有效的工具能解决在教学与研究中遇到的问题,那么 MATLAB 语言正是这样的一种工具。它可以将使用者从繁琐、无谓的底层编程中解放出来,把有限的宝贵时间更多地花在解决问题中,这样无疑会提高工作效率。

作者曾于 1996 年在清华大学出版社出版《控制系统计算机辅助设计——MATLAB 语言与应用》一书,受到了广大读者的欢迎与关注。该书已经成为很多大学的研究生和本科生用教材,使得该书(包括 1995 年的试印本)成为国内关于 MATLAB 语言方面书籍中出版最早、影响最广的著作。

作者从 1988 年开始系统地使用 MATLAB 语言进行程序设计与科学研究,使用过包括其免费版本和 1.0 版在内的各个版本,积累了丰富的第一手经验;正式使用 5.3 版也有了一年多的经验,用 MATLAB 语言编写的程序曾作为英国 Rapid Data 软件公司的商品在国际范围内发行,新近编写的几个通用程序在 The MathWorks 公司(MATLAB 语言的开发者)的网页上可以下载,得到了国际上很多用户的关注。

本书从使用者的角度出发,并结合作者十多年的实际编程经验和体会,系统地介绍 MATLAB 语言的编程技术及其在科学运算中的应用,书中融合了作者的许多编程经验和第一手材料,内容精心剪裁,相信仍旧会受到读者的欢迎。

书中含有丰富的实例,除其中较个别的例子,如第 1 章中的部分演示例子和多项式类的定义之外,都是作者自己选择的。

应一些读者和朋友的建议,作者建立了一个关于 MATLAB 编程与应用的免费论坛,该论坛挂靠在“网盛”网站下,供读者和使用者交流经验、解决问题或答疑之用。该论坛的网址为

<http://matlab.netsh.net>

本书由东北大学信息学院徐心和教授主审。本书从酝酿到整个写作过程始终得到徐老师的鼓励和支持。作者的导师,东北大学任兴权教授和英国 Sussex 大学的 Derek P. Atherton 教授也对本书的最终成形提供了很多的帮助。是他们将作者引入数学运算和 MATLAB 语言编程的乐园,并在这个领域开始了充满趣味的工作。

本书的出版过程得到了清华大学出版社蔡鸿程总编的关怀和帮助,还得到出版社的胡先福、田在儒编辑细心的加工,作者对他们的辛勤工作深表谢意。本书的写作和排版过程得到了沈阳天雄信息技术公司的支持,在此特致谢意。

作者的一些同事和朋友也先后给予作者许多建议和支持,并使作者获得了许多有益的信息与材料,这当中包括新加坡 Seagate 公司的陈旧泉博士、澳大利亚军事科学研究院大学学院的谢力博士、东北大学信息学院的肖文栋博士、郝培锋博士等,还有许多在互联网上交流的众多知名的和不知名的同行,在此一并表示深深的谢意。本书定稿排版得到了中科院数学与系统科学研究院应用数学研究所的王磊博士的极大帮助,特此致谢。

本书受“教育部高等学校骨干教师资助计划”资助,在此深表谢意。

本书的出版还得到了 MATLAB 软件的开发者——美国 The MathWorks 公司图书计划的支持,如果读者想购买 MATLAB 软件及系列产品,请和该公司直接联系:

The MathWorks, Inc., 3 Apple Hill Drive

Natick, MA, 01760-2098 USA

Tel: 508-647-7000, Fax: 508-647-7101

E-mail: info@mathworks.com

Web: <http://www.mathworks.com>

由于作者水平有限,书中的缺点错误在所难免,欢迎读者批评指教。

几年来,作者的妻子杨军在生活和事业上给予了作者莫大的帮助与鼓励,作者谨以此书献给她和女儿薛杨。

薛定宇

2000 年 5 月 7 日于沈阳东北大学

<http://xuedy.363.net/>

书中英文字体说明:

- 英文文体与公式中常量采用正体 Times-Roman 字体,如 MATLAB, e , x 轴
- 公式字体采用 Times-Roman 斜体字,如 x , A , 向量用黑体正体 $\mathbf{x}$
- 程序内容、函数、命令语句或文件名采用 Courier 正体,如 `eig()`
- 界面中的文字标识采用 Sans Serif 正体,如 File 菜单, OK 按钮

目 录

第 1 章 MATLAB 语言与现代科学运算概述	1
1.1 现代科学运算技术	1
1.1.1 数学问题的解析解法与数值解法	1
1.1.2 计算技术的发展	2
1.2 MATLAB 语言简介	4
1.2.1 MATLAB 语言的形成和发展	4
1.2.2 MATLAB 语言的特色	5
1.2.3 MATLAB 语言的工具箱	6
1.3 本书的结构	8
1.4 MATLAB 基本功能演示	9
1.5 习题	14
第 2 章 MATLAB 环境设置和编辑工具介绍	15
2.1 MATLAB 的图形界面	15
2.2 MATLAB 的基本命令	24
2.3 MATLAB 的源程序编辑器	26
2.3.1 源程序编辑器	26
2.3.2 MATLAB 的伪代码技术	27
2.4 MATLAB 的联机帮助系统	28
2.5 习题	31
第 3 章 MATLAB 语言程序设计基础	33
3.1 MATLAB 基本数据类型	33
3.1.1 变量、常量与赋值语句结构	33
3.1.2 矩阵的 MATLAB 表示	35
3.1.3 多维数组的定义	39
3.1.4 字符串变量及其处理	41
3.1.5 稀疏矩阵	45
3.2 MATLAB 语言的基本运算与输入输出	47
3.2.1 变量类型的测试	47
3.2.2 矩阵的代数运算	48
3.2.3 矩阵的逻辑运算	52
3.2.4 矩阵的比较关系	54

3.2.5	矩阵元素的数据变换	55
3.2.6	输入与输出语句	57
3.3	MATLAB 语言的程序流程语句	57
3.3.1	循环结构	57
3.3.2	条件转移结构	60
3.3.3	开关结构	62
3.3.4	试探式语句结构	63
3.4	MATLAB 语言的函数编写规范与技巧	65
3.4.1	MATLAB 语言的函数的基本结构	65
3.4.2	全局、局部变量、子函数与私有目录	68
3.4.3	MATLAB 函数的跟踪调试	68
3.4.4	MATLAB 函数输入和返回变量的其他信息	72
3.4.5	工具箱编写技巧	72
3.5	MATLAB 中新的数据结构	73
3.5.1	数据结构体	73
3.5.2	MATLAB 下类与对象的构造	75
3.5.3	单元结构	79
3.6	MATLAB 语言编程技巧	81
3.6.1	MATLAB 运行机制与工作空间管理	81
3.6.2	测定程序执行时间和时间分配	82
3.6.3	日期与时间函数	83
3.6.4	加快 MATLAB 程序执行速度的建议	84
3.6.5	文件格式与读写	86
3.7	习题	90
第 4 章	MATLAB 科学绘图	93
4.1	MATLAB 语言二维图形显示	94
4.1.1	基本二维图形绘制语句	94
4.1.2	带有其他选项的绘图函数	95
4.1.3	二维曲线的标注方法	97
4.1.4	MATLAB 绘图模式下鼠标的应用	97
4.2	MATLAB 句柄图形技术及应用	99
4.2.1	图形窗口的分割及坐标轴句柄设定	99
4.2.2	字符对象及其属性	104
4.2.3	曲线对象及其修改	105
4.2.4	在 MATLAB 图形上添加文字标注	106
4.3	MATLAB 5.3 版本的图形可视编辑工具	109
4.4	其他类型的 MATLAB 二维图形绘制方法	114

4.4.1	特殊图形绘制函数及举例	114
4.4.2	条形图与直方图绘制函数	116
4.4.3	极坐标曲线绘制函数	118
4.4.4	对数坐标绘制函数	118
4.5	MATLAB 语言的三维图形显示	120
4.5.1	三维曲线的绘制方法	120
4.5.2	标准三维曲面绘制	122
4.5.3	三维曲面的绘制方法	123
4.5.4	三维图形的特殊效果及处理	125
4.5.5	三维曲面的其他处理技术	133
4.6	视角变换与三视图	135
4.6.1	三维图形的视角与变换矩阵	136
4.6.2	三维图形的三视图	138
4.7	图像文件的读写与显示	140
4.8	MATLAB 图形输出技术与 PostScript 语言入门	143
4.8.1	图形输出菜单与应用	143
4.8.2	图形输出命令	144
4.8.3	PostScript 语言简介	145
4.9	习题	148
第 5 章	MATLAB 语言与现代科学运算	151
5.1	数值线性代数问题及求解	151
5.1.1	特殊矩阵的 MATLAB 输入	151
5.1.2	矩阵的特征参数运算	154
5.1.3	矩阵的相似变换与分解	159
5.1.4	矩阵的特征值与特征向量	165
5.1.5	矩阵求逆与线性方程求解	167
5.1.6	矩阵的非线性运算	172
5.2	数值微积分与微分方程求解	177
5.2.1	数值差分与微分运算	177
5.2.2	数值积分运算	182
5.2.3	多重定积分的数值求解	184
5.2.4	常微分方程数值解法	186
5.2.5	常微分方程的变换	190
5.3	非线性代数方程组与最优化问题求解	193
5.3.1	最优化工具箱简介	193
5.3.2	非线性方程组求解	194
5.3.3	无约束最优化问题求解	196

5.3.4	有约束最优化问题求解	197
5.3.5	线性规划与二次型规划	200
5.4	数据插值与统计分析	202
5.4.1	一维数据的插值拟合	202
5.4.2	二维数据的插值拟合	204
5.4.3	数据分析与统计处理	205
5.5	数学问题解析运算——符号运算工具箱及其应用	208
5.5.1	符号运算工具箱的若干辅助功能	209
5.5.2	线性代数问题的解析解法	209
5.5.3	极限与微积分运算	211
5.5.4	代数方程与微分方程解析解	213
5.5.5	积分变换	215
5.6	习题	216
第 6 章	MATLAB 图形用户界面设计技术	219
6.1	MATLAB 语言图形界面编程基础	220
6.1.1	MATLAB 图形界面中各对象的关系	220
6.1.2	窗口对象及属性设定	221
6.1.3	标准对话框及其调用	227
6.2	MATLAB 图形界面设计基本控件	230
6.2.1	MATLAB 支持的基本控件	231
6.2.2	控件的常用属性	232
6.2.3	GUI 程序设计技巧与举例	237
6.3	MATLAB 界面菜单系统设计	240
6.3.1	菜单系统的生成	240
6.4	可视界面开发工具 Guide 及应用	245
6.4.1	用户图形界面开发工具简介	245
6.4.2	程序设计举例	251
6.5	习题	260
第 7 章	MATLAB 和其他语言的接口	263
7.1	C 语言环境下提供的 MATLAB 变量格式及函数概述	264
7.2	Mex 技术——用 C 语言编写 MATLAB 可调用的函数	266
7.2.1	编译程序环境设置	266
7.2.2	Mex 文件的结构	267
7.2.3	Mex 文件的编写方法与编写步骤	271
7.2.4	其他类型输入输出变量应用	273
7.2.5	MAT 文件的读写方法	279

7.3	在 Mex 下调用 MATLAB 函数	282
7.4	生成独立于 MATLAB 的 C 代码	286
7.4.1	MATLAB 计算引擎	287
7.4.2	MATLAB 翻译程序与数学库	288
7.4.3	MATCOM——量佳的 MATLAB 翻译和编译程序	302
7.5	MATLAB 语言与 Windows 程序接口	302
7.5.1	动态数据交换 DDE 技术	302
7.5.2	ActiveX 技术与应用	307
7.6	MATLAB 笔记本及其应用	315
7.7	可视系统仿真环境——Simulink	317
7.7.1	系统模型绘制	318
7.7.2	Simulink 中的子模块和封装技术	322
7.8	习题	324
 附录A MATLAB 命令分类列表		327
A.1	MATLAB 通用命令	327
A.2	代数、逻辑运算与特殊符号	328
A.3	MATLAB 编程与调试语句	329
A.4	数值与数组	330
A.5	数据类型转换与处理	332
A.6	数值线性代数	333
A.7	数据分析与 Fourier 变换	334
A.8	MATLAB 图形绘制与界面设计	335
A.9	句柄图形学	337
A.10	用户图形界面设计技术	338
 参考文献		339

第 1 章 MATLAB 语言与现代科学运算概述

1.1 现代科学运算技术

1.1.1 数学问题的解析解法与数值解法

现代科学与工程的发展离不开数学。数学家们感兴趣的问题和其他科学家、工程技术人员所关注的问题是不同。数学家往往对数学问题的解析解，或称闭式解 (closed-form solution) 和解的存在性严格证明感兴趣，而工程技术人员一般对如何求出数学问题的解更关心，换句话说，能用某种方法获得问题的解则是工程技术人员更关心的问题。而获得这样解的最直接方法就是通过数值解法技术。在实际应用中，至少有两种情况需要数值解法：

● 解析解不存在时

解析解不存在的情况在数学上并不罕见，甚至可以说，这样的现象是常见的。例如定积分 $\int_a^b e^{-x^2/2} dx$ 在上下限均为无穷时就没有解析解，虽然数学家用其他的符号去定义这样的解，但解的值到底多大却不是一目了然的。所以，在这样的情况下，我们要想获得积分的值，就必须采用数值解技术。

再例如，圆周率 π 的值本身就没有解析解，中国古代的数学家、天文学家祖冲之早在公元 480 年就算定了该值在 3.1415926 和 3.1415927 之间，但直到现在仍有“数学家”在试图计算出更多的有效位，甚至 1995 年 10 月，有人算出 6442450938 位来，但可以说，即使是这样的值也不是解析解！实际上在一般科学与工程应用中，没有必要取那么多位的解，取 60 多亿位本身要占用多少计算机存储空间啊！在一般应用中取到祖冲之得到的近似值足矣，再精确的运算也至多取 20 多位就足够了。对计算机来说，取再多的位非但不会改进结果的精度，反而会加大计算机的负担，造成不必要的损失和浪费。其实，对问题的估算来说，使用公元前 250 年 (?) 阿基米德的 3.1418 也未尝不可，没有必要非去追求不存在的解析解不可。所以在这样的问题上，数值解法的优势就显示出来了。

● 解析解存在但不实用时

例如，考虑 n 元一次代数方程组的求解问题，由著名的 Cramer 法则，我们可以把该问题化简为 n 个 $n-1$ 元一次方程组的求解，而每个 $n-1$ 元方程组的解又可以简化为 $n-1$ 个 $n-2$ 元一次方程组进行求解。从理论上讲，总可以把多元一次的方程组简化成解析可解的形式，从而可以得出结论： n 元一次方程组的解析解是可以求出来的。

不幸的是， n 元一次方程组的求解需要进行 $(n-1)(n+1)! + n$ 次基本运算。如果想求解不是很大规模的 20 元方程问题，需要进行 9.7073×10^{20} 次基本运算，即使用速度极快的每秒 100 亿次巨型计算机去求解这样的问题，也需要解上 3000 年！而在某些科学工程计算中，又常常需要求解成百上千个变元的问题，指望用解析解的方法来求解也就成

了天方夜谭。

如果不去追求解析解,而想得到数值解,则可以通过计算数学的成果将原方程进行变换,比如用 Gauss 消元法等数值方法,这样 300 个变元的方程在一般计算机上 1 秒钟内就可以解决问题^①,而求解上千个变元的问题也用不了多长的时间。

数学问题的数值解法已经成功地应用于各个领域。例如,在力学领域,常用有限元法求解偏微分方程;在航空、航天与自动控制领域,经常用到数值线性代数与常微分方程的数值解法等解决实际问题;在工程与非工程系统的计算机仿真中,核心问题的求解也需要用到各种差分方程、常微分方程的数值解法;在高科技的数字信号处理领域,离散的快速 Fourier 变换 (FFT) 已经成为其不可或缺的工具。在科学工程研究中能掌握一个或多个实用的计算工具,无疑会为研究者提供解决实际问题的强有力手段。

1.1.2 计算技术的发展

数值计算技术的研究早在计算机出现之前就开始了,比如中国古代的数学家、天文学家祖冲之曾用多边形去逼近圆的方法求取圆周率,英国大科学家牛顿提出了微积分理论,并在非线性方程求解中引入导数,用迭代的方法求解方程,此外,前面提及的高斯求解多元代数方程的消去法等,这些都是早期计算数学发展的成功范例。

数字计算机出现以来,更给数值计算技术的研究注入了新的活力。在现代计算技术的早期发展中,出现了一些著名的软件包,如美国的基于特征值的软件包 EISPACK^[7, 20]和线性代数软件包 LINPACK^[4],英国牛津数值算法研究组 (Numerical Algorithm Group) 开发的 NAG 软件包^[18]及文献 [19] 中给出的程序集等。

美国的 EISPACK 和 LINPACK 都是基于矩阵特征值和奇异值解决线性代数问题的专用软件包。限于当时的计算机发展状况,这些软件包大都是由 Fortran 语言编写的源程序组成的。

例如若想求出 N 阶实矩阵 A 的全部特征值 (用 WR 、 WI 数组分别表示其实虚部) 和对应的特征向量矩阵 Z , 则 EISPACK 软件包给出的子程序建议调用路径为:

```
CALL BALANC(NM,N,A,IS1,IS2,FV1)
CALL ELMHES(NM,N,IS1,IS2,A,IV1)
CALL ELTRAN(NM,N,IS1,IS2,A,IV1,Z)
CALL HQR2(NM,N,IS1,IS2,A,WR,WI,Z,IERR)
IF (IERR.EQ.0) GOTO 99999
CALL BALBAK(NM,N,IS1,IS2,FV1,N,Z)
```

由上面的叙述可以看出,要求取矩阵的特征值和特征向量,首先要给一些数组和变

^① 比如利用 MATLAB 语言,先生成一个 300×300 的随机矩阵,然后对它进行求逆,这样,执行命令

```
>> A=rand(300,300); tic; B=inv(A); toc
elapsed_time =
    0.8200
```

在 Pentium III-550 上只需 0.82 秒,其中 tic 和 toc 为启动和停止秒表计时的 MATLAB 命令。本书中后面将用到的执行时间均是指在这类机器上的实际运行时间。

量依据 EISPACK 的格式作出定义和赋值, 并编写出主程序, 再经过编译和连接过程, 形成可执行文件, 最后才能得出所需的结果。

英国的 NAG 和美国学者的 Numerical Recipes 工具包则包括了各种各样数学问题的数值解法, 二者中 NAG 的功能尤其强大。NAG 的子程序都是以字母加数字编号的形式命名的, 非专业人员很难找到适合自己问题的子程序, 更不用说能保证以正确的格式去调用这些子程序了。这些程序包使用起来极其复杂, 谁也不能保证不发生错误, NAG 光数百页的使用手册就十几本!

Numerical Recipes 一书中给出的一系列算法语言源程序也是一个在国际上广泛应用的软件包。该书中的子程序有 C, Fortran 和 Pascal 等版本, 适合于科学研究者和工程技术人员直接应用。该书的程序包由 200 多个高效、实用的子程序构成, 这些子程序一般有很好的数值特性, 比较可靠, 为各国的研究者信赖。

在自动控制领域的经典软件包作品中有英国 Kingston Polytechnic 控制系统研究组开发的 SLICE (subroutine library in control engineering) 软件包等, 它也是一个具有特色的软件包。

具有 Fortran 和 C 等高级计算机语言知识的读者可能已经注意到, 如果用它们去进行程序设计, 尤其当涉及矩阵运算或画图时, 则编程会很麻烦。比如说, 若想求解一个线性代数方程, 用户得首先去编写一个主程序, 然后编写一个子程序去读入各个矩阵的元素, 之后再编写一个子程序, 求解相应的方程 (如使用 Gauss 消去法), 最后输出计算结果。如果选择的计算子程序不是很可靠, 则所得的计算结果往往可能会出现问题。如果没有标准的子程序可以调用, 则用户往往要将自己编好的子程序逐条地敲入计算机, 然后进行调试, 最后进行计算。这样一个简单的问题往往需要用户编写 100 条左右的源程序, 键入与调试程序也是很费事的, 并无法保证所敲入的程序 100% 可靠。求解线性方程组这样一个简单的功能需要 100 条源程序, 其他复杂的功能往往要求有更多条语句, 如采用双步 QR 法求取矩阵特征值的子程序则需要 500 多条源程序, 其中任何一条语句有毛病, 甚至调用不当 (如数组维数不匹配) 都可能导致错误结果的出现。

用软件包的形式编写程序有如下的缺点:

- **使用不方便** 对不是很熟悉所使用软件包的用户来说, 直接利用软件包编写程序是相当困难的, 也是容易出错的。如果其中一个子程序调用发生微小的错误则可能导致最终得出错误的结果。
- **调用过程繁琐** 首先需要编写主程序, 确定对软件包的调用过程, 再经过必要的编译和连接过程, 有时还要花大量的时间去调试程序以保证其正确性, 而不是想得出什么马上就可以得出的。
- **执行程序过多** 想求解一个特定的问题就需要编写一个专门的程序, 并形成可执行文件, 如果要求解的问题很多, 那么就需要在计算机硬盘上同时保留很多这样的可执行文件, 这样, 计算机磁盘空间的利用不是很经济, 管理起来也将十分困难。
- **不利于传递数据** 通过软件包调用方式会针对每个具体问题形成一个孤立的可执行文

件,因而在一个程序中产生的数据无法传入另一个程序,更无法使几个程序同时执行以解决所关心的问题。

- **维数指定困难** 在很多数学问题中最重要的变量是矩阵,如果要求解的问题维数较低,则形成的程序就不能用于求解高阶问题,例如文献 [14] 中的程序维数均定为 10 阶。所以有时为使得程序通用,往往将维数设置得很大,这样在解小规模问题时会出现空间的浪费,而更大规模问题仍然求解不了。在优秀的软件中往往需要动态地进行矩阵定维。

此外,这里介绍的大多数早期软件包都是由 Fortran 语言编写的,由于众所周知的原因,以前使用 Fortran 语言绘图并不是轻而易举的事情,它需要调用相应的软件包做进一步处理,在绘图方面比较实用和流行的软件包是 GINO-F^[3],但这种软件包只给出绘图的基本子程序,如果要绘制较满意的图形则需要用户自己用这些低级命令编写出合适的绘图子程序来。

除了上面指出的缺点以外,用 Fortran 和 C 等程序设计语言编程还有一个致命的弱点,那就是因为 C 语言本身的原因,致使在不同的机器平台上,扩展的 C 源程序代码是不兼容的,尤其在绘图及界面设计方面更是如此。例如在 PC 机的 Microsoft Windows 操作系统下编写的 C 语言程序不能立即在 SUN 工作站上直接运行,而需要在该机器上作出相应的修改才可以。

MATLAB[®] 语言的出现将数值计算技术与应用带入了一个新的阶段。MATLAB 语言发行后国际上又出现了很多仿照其思想的软件,如稍后出现的 Matrix-X 以及现在仍作为免费软件的 SciLAB 等。本书将以当前最新的 MATLAB 5.3 版本为主要对象,系统地介绍 MATLAB 语言的编程技术及其在科学运算领域的应用。

计算机代数系统是在本领域中又一个吸引人的主题,而解决数学问题解析计算又是 C 类语言直接应用的难点。于是国际上很多学者在研究、开发高质量的计算机代数系统。早期 IBM 公司开发的 mumath 和 Reduce 等软件为解决这样的问题提出了新的思路。后来出现的 Maple 和 Mathematica 逐渐占领了计算机代数系统的市场,成为比较成功的实用工具。

早期的 Mathematica 可以和 MATLAB 语言交互信息,比如通过一个称为 MathLink 的软件接口就可以很容易地完成这样的任务。为了解决计算机代数问题, MATLAB 语言的开发者——美国 The MathWorks 公司也研制开发了“符号运算工具箱”(Symbolic Toolbox[®]),该工具箱将 Maple 语言的内核作为 MATLAB 符号运算的引擎,使得二者能更好地结合起来。

1.2 MATLAB 语言简介

1.2.1 MATLAB 语言的形成和发展

MATLAB 语言的首创者 Cleve Moler 教授在数值分析,特别是在数值线性代数的领域中很有影响,他在 60 年代从师于计算数学界泰斗、斯坦福大学的 George Forsythe 教

授, 并参与编写了数值分析领域一些著名的著作[4, 5, 6, 7, 9, 20]。他曾在密西根大学、斯坦福大学和新墨西哥大学任数学与计算机科学教授。1980 年前后, 当时的新墨西哥大学计算机系主任 Moler 教授在讲授线性代数课程时, 发现了用其他高级语言编程极为不便, 便构思并开发了 MATLAB (MATrix LABoratory, 即矩阵实验室), 这一软件利用了当时流行的 EISPACK^[20] (基于特征值计算的软件包) 和 LINPACK^[4] (线性代数软件包) 两大软件包中可靠的子程序, 用 Fortran 语言编写了集命令翻译、科学计算于一身的交互式软件系统。其实 EISPACK 和 LINPACK 也是 Moler 教授参与开发的子程序库。

所谓交互式语言, 是指人们给出一条命令, 立即就可以得出该命令的结果。该语言无需像 C 和 Fortran 语言那样, 首先要求使用者去编写源程序, 然后对之进行编译、连接, 最终形成可执行文件。这无疑会给使用者带来了极大的方便。

早期的 MATLAB 是用 Fortran 语言编写的, 只能作矩阵运算; 绘图也只能用极其原始的方法, 即用星号描点的形式画图; 内部函数也只提供了几十个。但即使其当时的功能十分简单, 当它作为免费软件出现以来, 还是吸引了大批的使用者。

在 MATLAB 下, 矩阵的运算变得异常的容易, 这一系统逐渐发展、完善, 逐步走向成熟, 形成了今天的模样。Cleve Moler 和 John Little 等人成立了一个名叫 The MathWorks 的公司, Cleve Moler 一直任该公司的首席科学家。该公司于 1984 年推出了第一个 MATLAB 的商业版本。当时的 MATLAB 版本已经用 C 语言作了完全的改写, 其后又增添了丰富多彩的图形图像处理、多媒体功能、符号运算和它与其他流行软件的接口功能, 使得 MATLAB 的功能越来越强大。

The MathWorks 公司于 1992 年推出了具有划时代意义的 MATLAB 4.0 版本, 并于 1993 年推出了其微机版^[12], 可以配合 Microsoft Windows 一起使用, 使之应用范围越来越广。1994 年推出的 4.2 版本扩充了 4.0 版本的功能, 尤其在图形界面设计方面更提供了新的方法。

1997 年推出的 MATLAB 5.0 版允许了更多的数据结构, 如单元数据、数据结构体、多维矩阵、对象与类等, 使其成为一种更方便编程的语言。1999 年初推出的 MATLAB 5.3 版在很多方面又进一步改进了 MATLAB 语言的功能。

目前, MATLAB 已经成为国际上最流行的科学与工程计算的软件工具, 现在的 MATLAB 已经不仅仅是一个“矩阵实验室”了, 它已经成为了一种具有广泛应用前景的全新的计算机高级编程语言了, 有人称它为“第四代”计算机语言, 它在国内外高校和研究部门正扮演着重要的角色。MATLAB 语言的功能也越来越强大, 不断适应新的要求提出新的解决方法。可以预见, 在科学运算与科学绘图领域 MATLAB 语言将长期保持其独一无二的地位。

1.2.2 MATLAB 语言的特色

除了 MATLAB 语言的强大数值计算和图形功能外, 它还有其他语言难以比拟的功能, 此外, 它和其他语言的接口能够保证它可以和各种各样的强大计算机软件相结合, 发挥更大的作用。

MATLAB 目前可以在各种类型的常用计算机上运行。如果单纯地使用 MATLAB 语言进行编程而不采用其他外部语言, 则用 MATLAB 语言编写出来的程序不做丝毫地修改便可以直接移植到其他机型上使用, 所以说与其他语言不同, MATLAB 是和机器类型无关的。

即使是设计像图形绘制或图形界面设计这样往往依赖于机器类型的程序在 MATLAB 下也可以简单地将程序“拷贝”到其他机器下, 而不必担心它能否在新的机器环境下正确运行。事实上, 单纯地采用 MATLAB 语言就可以实现 C 或 Fortran 等编程语言能做到的绝大部分工作, 并且实现起来要比这些语言简洁方便得多。

依作者的观点, MATLAB 和其他高级语言之间的关系仿佛该高级语言和汇编语言的关系一样, 因为虽然高级语言的执行效率要低于汇编语言, 但其编程效率与可读性、可移植性要远远高于汇编语言。同样, MATLAB 比一般高级语言的执行效率要低, 而其编程效率与可读性、可移植性要远远高于其他高级语言, 所以在科学运算中较适合于从像 MATLAB 这样的专用高级语言入手, 这不但可以大大地提高编程的效率, 而且可以大大地提高编程的质量与可靠性。例如在实际应用中, 往往要求出一个矩阵的特征值, 而求取矩阵特征值的方法是多种多样的, 如 QR 双步方法、Jacobi 方法等, 所得出的结果也不尽相同。但不管矩阵的参数如何, 在 MATLAB 环境下就可以用一条指令立即求出系统的特征值来, 而不必去考虑是用什么算法以及如何实现等低级问题, 也不必深入了解相应算法的具体内容。这就像在 C 语言下不必去深究乘法是怎样实现的一样, 而只采用乘积的结果就可以了。其实对于专门从事科学运算研究的人员来说, 因为 MATLAB 语言可以轻易地再现 C 或 Fortran 语言几乎全部的功能, 所以即使用户不懂 C 或 Fortran 这样的程序设计语言也照样可以设计出功能强大、界面优美、稳定可靠的高质量程序来, 且开发周期会大大地缩短。

MATLAB 语言具有较高的运算精度。一般情况下, 在矩阵类运算中往往可达到 10^{-15} 级精度, 这当然符合一般科学与工程运算的要求。

如果矩阵的条件数很大, 则矩阵中一个参数的微小变化, 即使会使最终结果发生极大的变化, 这种现象在数学上称为坏条件问题。如果采用的方法不当, 则最后得出的结果可能是不正确的。采用 MATLAB 一般不会出现这类错误结果, 亦即 MATLAB 是可靠的、数值稳定的, 而采用 C 或其他高级语言编写出来的程序在求解这类问题时则可能得出错误结果。

1.2.3 MATLAB 语言的工具箱

每个 MATLAB 的工具箱实际上是一组实现某种功能的特定 MATLAB 函数。MATLAB 语言最初是由数值线性代数这个学科发展起来的, 所以说, MATLAB 本身就含有丰富的数值线性代数功能, 另外, 常微分方程数值求解也是该语言自身的特色。除了这两个计算数学分支外, MATLAB 还有丰富的数值计算工具箱, 主要的工具箱如下:

- Fuzzy Logic Toolbox —— 模糊逻辑工具箱
- Genetic Algorithm Toolbox —— 遗传算法工具箱

- NAG Foundation Toolbox —— 数值算法研究组工具箱
- Neural Network Toolbox —— 神经网络工具箱
- Optimization Toolbox —— 最优化工具箱
- Partial Differential Equation Toolbox —— 偏微分方程工具箱
- Signal Processing Toolbox —— 信号处理工具箱
- Spline Toolbox —— 样条插值工具箱
- Statistics Toolbox —— 统计学工具箱
- Symbolic Toolbox —— 符号运算工具箱
- Wavelet Toolbox —— 小波工具箱

除了上述的 MATLAB 商品工具箱外, 在 The MathWorks 公司的网页上还链接了很多免费的工具箱和函数, 例如矩阵微分方程程序、数值积分工具箱等, 这些工具箱或应用程序为用户解决自己的问题提供了极大的方便, 可以说, 用户可以用 MATLAB 语言解决任何数值计算问题, 而解决的大多数问题甚至无需用户自己去编写底层程序。

MATLAB 在很多工程领域有着广泛的应用现状与前景, 如在自动控制领域出现了各种各样分支的工具箱, 这些工具箱很多都成了该领域的标准。在非工程领域也有各种各样的 MATLAB 工具箱, 如它的 Financial 工具箱、数据库工具箱等。

由于 MATLAB 程序售价较昂贵, 国外很多部门还研制了仿照 MATLAB 的程序, 例如法国 INRIA 研制的 SciLAB, 该软件连同其源代码都可以在互联网上免费获得。该软件定义的语句格式很类似于 MATLAB, 其程序界面如图 1-1 所示。SciLAB 的主要应用目标是在自动控制领域, 在控制系统问题求解上有其自己的特色, 在基本数值分析与符号运算中也有很多其自己的工具箱, 如线性代数工具箱、最优化工具箱等, 这些工具箱也可以直接应用, 得出可靠的结果。

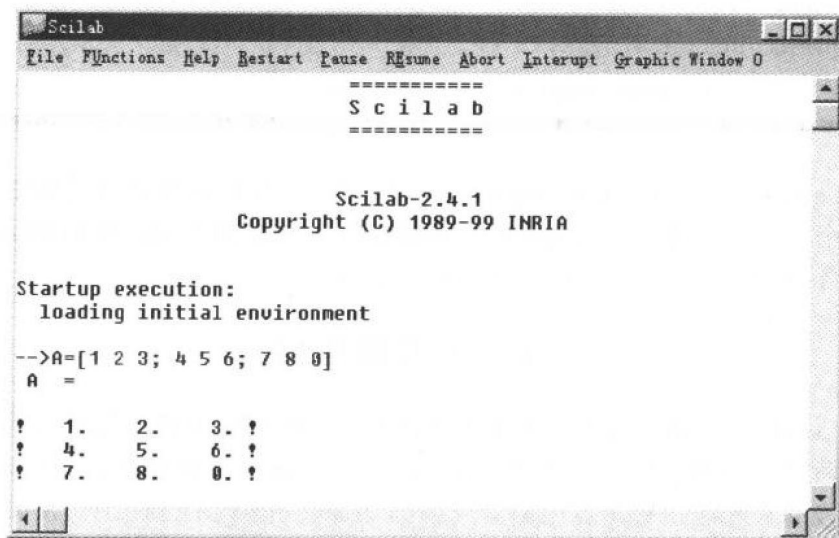


图 1-1 SciLAB 的程序界面