



基于 MATLAB/Simulink 的
系统仿真技术与应用

薛定宇 陈阳泉 著



清华大学出版社

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



基于 MATLAB/Simulink 的 系统仿真技术与应用

薛定宇 陈阳泉 著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书首先介绍了 MATLAB 语言的程序设计的基本内容,在此基础上系统介绍了系统仿真所必要的数值计算方法及 MATLAB 实现,并以 Simulink 为主要工具介绍了系统仿真方法与技巧,包括连续系统、离散系统、随机输入系统和复数系统的仿真,由浅入深介绍了模块封装技术、电力系统模块集、非线性系统设计模块集、S-函数编写与应用、Stateflow 有限状态机、虚拟现实工具箱等中高级使用方法,最后还介绍了半实物仿真技术与实时控制技术。

本书可作为一般读者学习和掌握 MATLAB/Simulink 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系统仿真类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系统仿真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, Simulink, Stateflow, xPC Target, Realtime Workshop, Power Systems Toolbox, Optimization Toolbox, Symbolic Toolbox, Virtual Reality Toolbox 为 The MathWorks 公司的注册商标

书 名: 基于 MATLAB/Simulink 的系统仿真技术与应用
作 者: 薛定宇 陈阳泉 著
出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)
<http://www.tup.tsinghua.edu.cn>
责任编辑: 欧振旭
印 刷 者: 北京昌平环球印刷厂
发 行 者: 新华书店总店北京发行所
开 本: 787×1092 1/16 印张: 28 字数: 645 千字
版 次: 2002 年 4 月第 1 版 2002 年 4 月第 1 次印刷
书 号: ISBN 7-302-05341-3/TP·3137
印 数: 0001~6000
定 价: 38.00 元

前 言

子曰：“工欲善其事，必先利其器”（《论语·卫灵公篇》）。如果有一种十分有效的工具能容易地解决在系统仿真及领域的教学与研究中遇到的问题，它可以使使用者从繁琐、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科学问题中，这样无疑会提高工作效率。本书介绍的 MATLAB[®]/Simulink[®] 语言正是这样的一种工具。在系统仿真、自动控制等领域，国外很多高校在教学与研究中都将 MATLAB/Simulink 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。

MATLAB 语言虽然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。

多年来，本人一直在试图以最实用的方式将 MATLAB 语言介绍给国内的读者，并在清华大学出版社出版了 3 部有关 MATLAB 语言及其应用方面的著作，受到了国内外广大中文读者的普遍欢迎。其中，1996 年出版的《控制系统计算机辅助设计——MATLAB 语言与应用》一书被公认为国内关于 MATLAB 语言方面书籍中出版最早、影响最广的著作。本书主要介绍目前最新的 MATLAB 6.1/Simulink 4.1 版本，并适当引入即将正式推出的 MATLAB 6.5/Simulink 5.0 版本的新内容。

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，积累了丰富的第一手经验；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。

本书的合作者陈阳泉博士在系统仿真和 MATLAB/Simulink 语言应用领域有很深的造诣和独到见解，在控制系统的理论研究和工业过程的半实物仿真与实时控制都颇有建树。

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。

应一些读者和朋友的建议，作者建立了“MATLAB 大观园”网站 (<http://matlab.myrice.com>) 和“MATLAB 语言与应用”交流论坛 (<http://matlab.netsh.net>)，一年多来在广大读者和 MATLAB 语言的使用者中有着良好的声誉和影响，解决了大量用户遇到的实际问题，是读者和使用者交流经验、解决问题或答疑的一方天地，被戏称为作者著作的“售后服务站”。

MATLAB 6.5/02

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，是他们将作者引入系统仿真和 MATLAB/Simulink 语言编程的乐园，并在这个领域开始了充满趣味的工作。

作者的一些同行和朋友也先后给予作者许多建议和支持，包括北方交通大学机电学院的朱衡君教授、中科院系统科学研究所的韩京清研究员、澳大利亚军事科学院大学学院的谢力博士、东北大学信息学院的王贞祥教授、东北大学材料冶金学院钟圣俊博士、哈尔滨工业大学土木工程学院王刚博士等，还有在互联网上交流的众多知名的和不知名的同行，在此表示深深的谢意。在本书部分内容的写作过程中，作者和博士生胡英、潘峰、龙德、司秉玉、吴炜等同学进行了较深入的讨论，在此一并表示感谢。

本书的出版得到了清华大学出版社蔡鸿程总编的关怀和帮助，还得到出版社的欧振旭编辑细心的加工，作者对他们的辛勤工作深表谢意。

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美国 The MathWorks 公司图书计划的支持，在此表示谢意，并特别感谢 Noami Fernandez 女士为作者作出的各种帮助。国外读者如果想购买 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

电话: +01-508-647-7000, 传真: +01-508-647-7101

E-mail: info@mathworks.com

网站: <http://www.mathworks.com>

国内读者可以直接和该公司在中国的惟一代理九州恒润科技有限公司联系：

北京市北三环中路 27 号商厦大厦 620 室，邮编 100029

电话: 010-62073595, 传真: 010-62073600

网站: <http://www.hirain.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。

多年来，我的妻子杨军在生活和事业上给予了我莫大的帮助与鼓励，没有她的鼓励和一如既往的支持，本书和前几部著作均不能顺利面世，谨以此书献给她和女儿薛杨。

薛定宇 (xue.dy@21cn.com)

2002 年 1 月 30 日于沈阳东北大学

目 录

第 1 章 系统仿真技术与应用	1
1.1 系统仿真技术概述	1
1.2 仿真软件的发展状况与应用	1
1.2.1 早期数学软件包的发展概况	2
1.2.2 仿真软件的发展概况	4
1.3 MATLAB 语言简介	6
1.3.1 MATLAB 语言发展简史	6
1.3.2 MATLAB 语言的特色	7
1.3.3 MATLAB/Simulink 在仿真中的应用演示	8
1.3.4 互联网上的 MATLAB 资源	13
1.4 本书的结构和代码	15
1.4.1 本书的结构	15
1.4.2 书中英文字体说明	16
1.5 习 题	16
第 2 章 MATLAB 语言程序设计基础	17
2.1 MATLAB 语言的基本使用环境	17
2.1.1 MATLAB 语言界面	17
2.1.2 MATLAB 的联机帮助与电子版手册	17
2.2 MATLAB 语言的数据结构	18
2.2.1 常量、变量与赋值语句	20
2.2.2 矩阵的 MATLAB 表示	21
2.2.3 多维数组的定义	23
2.2.4 数据结构体	25
2.2.5 单元结构	27
2.2.6 MATLAB 下的类与对象	28
2.3 MATLAB 下矩阵的运算	28
2.3.1 矩阵的代数运算	28
2.3.2 矩阵的逻辑运算	34
2.3.3 矩阵的比较关系	35
2.3.4 矩阵元素的数据变换	37
2.4 流程控制结构	38

2.4.1	循环结构	38
2.4.2	条件转移结构	42
2.4.3	开关结构	43
2.4.4	试探式语句结构	44
2.5	MATLAB 函数编写与技巧	45
2.5.1	MATLAB 语言的函数的基本结构	45
2.5.2	可变输入输出个数的处理	48
2.5.3	MATLAB 函数的跟踪调试	49
2.6	MATLAB 语言下图形绘制与技巧	50
2.6.1	基本二维图形绘制语句	51
2.6.2	带有其他选项的绘图函数	52
2.6.3	二维曲线的标注方法	53
2.6.4	在 MATLAB 图形上添加文字标注	54
2.6.5	MATLAB 的图形可视编辑工具	57
2.6.6	特殊图形绘制函数及举例	59
2.6.7	给定函数的曲线绘制	61
2.7	三维图形的绘制方法	62
2.7.1	三维曲线的绘制方法	62
2.7.2	三维曲面的绘制方法	64
2.7.3	局部图形的剪切处理	66
2.7.4	图像显示与处理	66
2.8	MATLAB 图形用户界面设计技术	68
2.8.1	图形界面设计工具 Guide	68
2.8.2	界面设计举例与技巧	74
2.9	提高 MATLAB 执行效率的技巧	82
2.9.1	测定程序执行时间和时间分配	82
2.9.2	加快 MATLAB 程序执行速度的建议	83
2.9.3	Mex 程序设计技术	86
2.10	习 题	93
第 3 章	MATLAB 语言在现代科学运算中的应用	96
3.1	解析解与数值解	96
3.2	数值线性代数问题及求解	98
3.2.1	特殊矩阵的 MATLAB 输入	98
3.2.2	矩阵的特征参数运算	100
3.2.3	矩阵的相似变换与分解	106
3.2.4	矩阵的特征值与特征向量	111
3.2.5	矩阵求逆与线性方程求解	114

3.2.6	矩阵的非线性运算	119
3.2.7	线性代数问题的解析求解	124
3.3	微积分问题的 MATLAB 求解	125
3.3.1	数值差分与微分运算	125
3.3.2	数值积分运算	130
3.3.3	多重定积分的数值求解	133
3.3.4	微积分问题的解析解运算	135
3.4	常微分方程的数值解法	138
3.4.1	常用常微分方程的数值解法	139
3.4.2	MATLAB 下的常微分方程求解函数	143
3.4.3	常微分方程举例	144
3.4.4	刚性方程的 MATLAB 求解	147
3.4.5	微分方程组的变换与技巧	151
3.4.6	微分代数方程的数值解法	155
3.4.7	二阶微分方程边值问题的数值解法	157
3.4.8	常微分方程的解析求解方法	163
3.5	非线性方程与最优化问题求解	165
3.5.1	非线性方程求解	165
3.5.2	无约束最优化问题求解	168
3.5.3	线性规划问题	170
3.5.4	二次型规划问题	171
3.5.5	一般非线性规划问题求解	172
3.6	数据插值与统计分析	175
3.6.1	一维数据的插值拟合	175
3.6.2	二维数据的插值拟合	177
3.6.3	最小二乘曲线拟合技术	178
3.6.4	数据简单排序	180
3.6.5	快速 Fourier 变换	181
3.6.6	数据分析与统计处理	182
3.7	习 题	188
第 4 章	MATLAB/Simulink 下数学模型建立与仿真	192
4.1	Simulink 模块库简介	192
4.1.1	信号源模块组	194
4.1.2	连续模块组	195
4.1.3	离散模块组	196
4.1.4	函数与表格模块组	197
4.1.5	数学运算模块组	198

4.1.6	非线性模块组	198
4.1.7	输出池模块组	199
4.1.8	信号与系统模块组	200
4.1.9	子系统模块组	200
4.1.10	其他模块组	201
4.2	Simulink 模型的建立	203
4.2.1	模型窗口建立	203
4.2.2	模块的连接与简单处理	203
4.2.3	模块的参数修正	206
4.2.4	Simulink 模块的联机帮助系统	208
4.2.5	Simulink 模型的输出与打印	208
4.2.6	启动仿真环境	209
4.3	Simulink 模型举例	214
4.4	线性系统的计算机仿真	224
4.4.1	线性系统的数学模型	224
4.4.2	线性连续系统的解析解	225
4.4.3	线性系统频域分析	226
4.4.4	Simulink 下的线性系统分析工具	227
4.5	随机输入下连续系统仿真	229
4.5.1	线性系统的仿真研究	231
4.5.2	在 Simulink 下的解决方法	235
4.5.3	仿真结果的统计分析	238
4.6	分形系统的仿真	242
4.6.1	分形树的仿真与绘图	242
4.6.2	Julia 图的仿真与绘制	245
4.6.3	Mandelbrot 图的仿真与绘制	247
4.7	习 题	249
第 5 章	Simulink 常用模块介绍与应用技巧	252
5.1	常用模块应用技巧	252
5.1.1	向量化模块举例	252
5.1.2	Simulink 模型的信号标识	256
5.1.3	线性系统模块	257
5.1.4	非线性环节与查表环节	259
5.1.5	微分代数方程的 Simulink 建模与求解	267
5.2	输出模块库	268
5.2.1	一般输出模块库	268
5.2.2	输出信号的表盘与量计显示	275

5.2.3 输出的数字信号处理	279
5.3 子系统与模块封装技术	281
5.3.1 子系统的处理	282
5.3.2 条件执行子系统	282
5.3.3 模块封装技术	285
5.3.4 组建自己的模块库	291
5.3.5 子系统应用举例——F14 战斗机仿真	293
5.4 电力系统模块集与电子线路仿真	296
5.4.1 电力系统模块集简介	296
5.4.2 电路的仿真	297
5.4.3 功率电子系统仿真	301
5.4.4 电机系统仿真	304
5.4.5 Spice 与 Simulink 的接口	309
5.5 非线性系统控制设计模块集	311
5.6 发动机模型——复杂系统建模实例	316
5.6.1 模型背景概述	316
5.6.2 发动机模型分析	317
5.6.3 开环系统的建模与仿真	321
5.6.4 闭环系统的建模与仿真	322
5.7 习 题	325
第 6 章 Simulink 仿真的高级技术	328
6.1 Simulink 模型的语句修改	328
6.1.1 Simulink 模型与文件的处理	328
6.1.2 模型属性与模块属性	329
6.1.3 用语句绘制方框图	333
6.2 系统仿真与线性化	338
6.2.1 仿真过程的命令化	338
6.2.2 非线性模型的线性化	339
6.2.3 纯时间延迟系统 Padé 近似	344
6.3 S-函数的编写及应用	350
6.3.1 用 MATLAB 语句编写 S-函数	350
6.3.2 S-函数设计举例——自抗扰控制器仿真	353
6.3.3 用 C 语句编写 S-函数	362
6.3.4 S-函数模块的进一步改进	366
6.4 Stateflow 原理与使用技巧	366
6.4.1 有限状态机简介	366
6.4.2 Stateflow 应用基础	367

6.4.3	Stateflow 的常用命令	372
6.4.4	Stateflow 应用举例	372
6.4.5	控制流程的 Simulink 仿真模块	376
6.5	MATLAB 虚拟现实工具箱及其应用	381
6.5.1	虚拟现实工具箱的安装与设置	382
6.5.2	VRML 语言程序设计入门与举例	383
6.5.3	在 MATLAB 下虚拟现实技术应用	386
6.5.4	Simulink 下虚拟现实技术应用	390
6.6	SimMechanics —— 机构系统模块集及应用	392
6.6.1	物理建模与机构系统仿真	392
6.6.2	SimMechanics 仿真简介	392
6.6.3	机构系统仿真举例	394
6.7	习 题	402
第 7 章	半实物仿真与快速原型设计技术	404
7.1	Simulink 仿真的实时工具 RTW	406
7.1.1	独立程序生成	406
7.1.2	实时仿真与目标计算机仿真	407
7.2	xPC 在仿真和快速原型设计中的应用	410
7.2.1	xPC 环境简介	410
7.2.2	建立基于 DOS 的可执行文件	411
7.2.3	基于 xPC 的半实物仿真技术	413
7.3	基于 dSPACE 的实时控制技术	417
7.3.1	dSPACE 硬件介绍	417
7.3.2	Control Desk 及虚拟仪器开发	420
7.4	习 题	422
附录	自编的 MATLAB/Simulink 程序索引	424
参考文献		429
索 引		432

第 1 章 系统仿真技术与应用

1.1 系统仿真技术概述

系统是由客观世界中实体与实体间的相互作用和相互依赖关系构成的具有某种特定功能的有机整体。系统的分类方法是多种多样的，习惯上依照其应用范围可以将系统分为工程系统和非工程系统。工程系统的含意是指由相互关联部件组成的一个整体，实现特定的目标，例如电机驱动自动控制系统是由执行部件、功率转换部件、检测部件所组成，用它来完成电机的转速、位置和其他参数控制的某个特定目标。

非工程系统涵盖的范围更加广泛，大至宇宙，小至微观世界都存在着相互关联、相互制约的关系，形成一个整体，实现某种目的，所以均可以认为是系统。

如果想定量地研究系统的行为，可以将其本身的特性及内部的相互关系抽象出来，构造出系统的模型。系统的模型分为物理模型和数学模型。由于计算机技术的迅速发展和广泛应用，数学模型的应用越来越普遍。

系统的数学模型是描述系统动态特性的数学表达式，用来表示系统运动过程中各个量的关系，是分析、设计系统的依据。从它所描述系统的运动性质和数学工具来分，又可以分为连续系统、离散时间系统、离散事件系统、混杂系统等。还可以细分为线性、非线性、定常、时变、集中参数、分布参数、确定性、随机等子类。

系统仿真是根据被研究的真实系统的数学模型研究系统性能的一门学科，现在尤指利用计算机去研究数学模型行为的方法。计算机仿真的基本内容包括系统、模型、算法、计算机程序设计与仿真结果显示、分析与验证等环节。

在系统仿真技术的诸多环节中，算法和计算机程序设计是很重要的一个环节，它直接决定原来问题是否能够正确地求解。基于国际上仿真领域最权威、最实用的计算机工具——MATLAB[®]语言介绍仿真问题的编程与求解方法将是本书最显著的特点。在第 1.2 节中将介绍数学软件、仿真软件的仿真概况和现状，第 1.3 节着重介绍 MATLAB/Simulink[®]语言的发展状况、语言特色和在该系统仿真领域的应用举例，可以初步领略 MATLAB 的强大功能，第 1.4 节中将介绍本书的结构和有关内容。

1.2 仿真软件的发展状况与应用

早期的计算机仿真技术大致经历了几个阶段：20 世纪 40 年代模拟计算机仿真；50 年代初数字仿真；60 年代早期仿真语言的出现等。80 年代出现的面向对象仿真技术为系统仿真方法注入了活力。我国早在 50 年代就开始研究仿真技术了，当时主要用于国防领域，以模拟计算机的仿真为主。70 年代初开始应用数字计算机进行仿真^[41]。随着数字

计算机的普及,近 20 年以来,国际、国内出现了许多专门用于计算机数字仿真的仿真语言与工具,如 CSMP, ACSL, SIMNON, MATLAB/Simulink, MatrixX/System Build, CSMP-C 等。

1.2.1 早期数学软件包的发展概况

数字计算机的出现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,出现了一些著名的数学软件包,如美国的基于特征值的软件包 EISPACK^[13, 44]和线性代数软件包 LINPACK^[8],英国牛津数值算法研究组 (Numerical Algorithm Group) 开发的 NAG 软件包^[38]及文献 [40] 中给出的程序集等,这些都是在国际上广泛流行的、有着较高声望的软件包。

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,这些子程序一般有较好的数值特性,比较可靠,为各国的研究者信赖。

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输入计算机，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，输入与调试程序也是很费事的，并无法保证所输入的程序 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 [33] 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^[5]，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分子程序来。

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，而需要在该机器上对源程序进行修改、编译后才可以执行。

尽管如此，数学软件包仍在继续发展，其发展方向是采用国际上最先进的数值算法，提供更高效率的、更稳定的、更快速、更可靠的数学软件包。例如在线性代数计算领

域, 全新的 LAPACK 已经成为当前最有影响的软件包, 但它们的目的是似乎已经不再为一般用户提供解决问题的方法, 而是为数学软件提供底层的支持。在新版的 MATLAB 已经抛弃了一直使用的 LINPACK 和 EISPACK, 采用 LAPACK 为其底层支持软件包。

1.2.2 仿真软件的发展概况

从前面提及的软件包的局限性看, 直接调用它们进行系统仿真将有较大的困难, 因为要掌握这些函数的接口是一件相当复杂的事, 准确调用它们将更难; 此外, 软件包函数调用直接得出的结果可信度也不是很高, 因为软件包的质量参差不齐。

抛弃成型的软件包另起炉灶自己编写程序也不是很现实的事, 毕竟在成型软件包中包含有很多同行专家的心血, 有时自己从头编写程序很难达到这样的效果, 所以必须采用经验证的信誉著称的高水平软件包或计算机语言来进行仿真研究。

仿真技术引起该领域各国学者、专家们的重视, 建立起国际的仿真委员会(Simulation Councils Inc, 简称为 SCi), 该公司于 1967 年通过了仿真语言规范。仿真语言 CSMP (Computer Simulation Modelling Language) 应该属于建立在该标准上的最早的专用仿真语言。中科院沈阳自动化研究所在 1988 年推出了该语言的推广版本——CSMP-C。

20 世纪 80 年代初期, 美国 Mitchell and Gauthier Associate 公司推出了依照该标准的著名仿真语言 ACSL (Advanced Continuous Simulation Language)^[34], 该语言出现后, 由于其功能较强大, 并有一些系统分析的功能, 很快就在仿真领域占据了主导地位。

ACSL 首先要求用户依照其语言规则建立起一个模型文件, 然后通过 ACSL 本身提供的命令对之进行仿真及辅助分析。ACSL 与 FORTRAN 语言的主要区别在于, ACSL 的语句更简练, 内容更丰富, ACSL 语言也可以直接调用由 FORTRAN 编写的子程序。ACSL 编程的结构比相应的 FORTRAN 语言更严格。程序的基本结构必须严格按照规定的格式来编写, 否则所得出的仿真结果可能出现意想不到的错误。ACSL 提供了几十个系统子模块 (macros), 其中包括很常用的线性和非线性子模块, 如传递函数模块 TRAN, 积分器模块 INTEG, 超前滞后环节 LEDLAG, 延迟模块 DELAY, 死区非线性模块 DEAD, 磁滞回环 BAKLSH, 限幅积分器 LIMINT 等, 用户可以利用这些子模块简单地编写出描述给定系统的仿真模型, 然后采用 ACSL 提供的功能来对系统进行仿真分析, 并绘制出结果的曲线表示。

编写完 ACSL 源程序后, 则可以采用 ACSL 的编译命令来编译并将此模型和 ACSL 库连接起来, 则可以形成一个可执行文件。这一过程完成之后 ACSL 将自动给出提示符 ACSL>, 在这个提示符下用户可以输入相应命令即可。

【例 1.1】著名的 Van der Pol 方程由下式给出:

$$\ddot{y} + \mu(y^2 - 1)\dot{y} + y = 0$$

若取 $\mu = 1$, 并取状态变量为 $y_1 = y, y_2 = \dot{y}$, 则 Van der Pol 方程可以写成

$$\dot{y}_1 = y_2, \dot{y}_2 = -y_1 - y_2^2$$

这时可以由 ACSL 所定义的语言写出此系统的模型如下:

```
PROGRAM VAN DER POL EQUATION
CINTERVAL CINT=0.01
CONSTANT Y1C=3.0, Y2C=2.5, TSTP=15.0
Y1=INTEG(Y1*(1-Y2**2)-Y2, Y1C)
Y2=INTEG(Y1, Y2C)
TERMT (T.GE.TSTP)
END
```

在此程序中把显示步长 CINT 设置为 0.01, 状态变量的初值由 Y1C 和 Y2C 表示, 而终止仿真时间 TSTP 设置为 15, 该程序中变量 T 为实际仿真时间。这样需要输入如下的语句:

```
ACSL> PREPAR T, Y1, Y2
ACSL> START
ACSL> PLOT Y1, Y2
```

来通知 ACSL 模型在仿真时需要保留 T, Y1, Y2 三个参数, 开始数字仿真, 并绘制出系统的相平面图 (Y1 和 Y2 直接的关系曲线)。还可以用下面的命令修改系统内部的参数:

```
ACSL> SET Y1C=-1, Y2C=-3
```

但必须先知道系统内部的变量名。

和 ACSL 大致同时的还有瑞典 Lund 工学院 Karl Åström 教授主持开发的 SIMNON, 英国 Salford 大学的 ESL 等, 这些语言的编程语句结构也是很类似的, 因为它们所依据的标准都是相同的。

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, 与之配套的 Simulink 仿真环境又为系统仿真技术提供了新的解决方案。MATLAB 语言发行后国际上又出现了很多仿照其思想的软件, 如稍后出现的美国的商品软件 Ctrl-C, Matrix-X, O-Matrix, 韩国汉城国立大学权旭铉教授主持开发的 CemTool, 以及现在仍作为免费软件的 Octave, SciLAB 等, 本书将以当前最新的 MATLAB 6.1/Simulink 4.1 版本为主要对象, 系统地介绍 MATLAB 语言的编程技术及其在系统仿真领域的应用。

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。

这些软件和语言还是很昂贵的, 所以有人更倾向于采用免费的, 但编程结构类似于

MATLAB 的计算机语言, 如 Octave 和 SciLAB, 这些软件的全部源程序也是公开的, 有较高的透明度, 但目前它们的功能已经无法与越来越强大的 MATLAB 语言相比。

子曰: “工欲善其事, 必先利其器”。系统仿真领域有很多自己的特性, 如果能选择一种能反映当今系统仿真领域最高水平, 也是最实用的软件或语言介绍仿真技术, 使得读者能直接采用该语言解决自己的问题, 将是很有意义的。实践证明, MATLAB 就是这样的仿真软件, 由于它本身卓越的功能, 已经使得它成为自动控制、航空航天、汽车设计等诸多领域仿真的首选语言。所以在本书中将介绍基于 MATLAB/Simulink 的系统仿真实理论与应用。

1.3 MATLAB 语言简介

1.3.1 MATLAB 语言发展简史

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^[8, 9, 10, 13, 19, 44]。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, 时任新墨西哥大学计算机系主任的 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), 这一软件利用了他研制的、在国际上颇有影响的 EISPACK^[44] (基于特征值计算的软件包) 和 LINPACK^[8] (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、科学计算于一身的一套交互式软件系统。

所谓交互式语言, 是指用户给出一条命令, 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极大的方便。在 MATLAB 下, 矩阵的运算变得异常的容易, 所以它一出现就广受欢迎, 这一系统逐渐发展、完善, 逐步走向成熟, 形成了今天的模样。

早期的 MATLAB 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。

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。最早的 PC 机版又称为 PC-MATLAB, 其工作站版本又称为 Pro MATLAB。1990 年推出的 MATLAB 3.5i 版是第一个可以运行于 Microsoft Windows 下的版本, 它可以在两个窗口上分别显示命令行计算结果和图形结果。稍后推出的 SimuLAB 环境首次引入了基于框图的仿真功能, 其模型输入的方式令人耳目一新, 该环境就是我们现在所知的 Simulink。