

(第2版)

基于MATLAB/Simulink的 系统仿真技术与应用

薛定宇 陈阳泉◎著

- 系统介绍MATLAB语言的编程技术及其在科学运算中的应用
- 从使用者的角度出发，解决在系统仿真领域经常遇到的问题
- 融合了笔者二十余年的实际编程、教学的经验和体会
- 结合第一手材料，体现了新的系统仿真方法和模块集

清华大学出版社

基于 MATLAB/Simulink 的系统 仿真技术与应用

(第 2 版)

薛定宇 陈阳泉 著

清华大学出版社

北京

内 容 简 介

本书首先介绍了 MATLAB 语言程序设计的基本内容,在此基础上系统地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,由浅入深地介绍了模块封装技术、复杂模型的线性化、多领域物理建模思想及工程系统仿真、非工程系统建模与仿真、Stateflow 有限状态机及离散事件系统建模与仿真等中高级使用方法,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,也可作为高等院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教学参考书,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。

MATLAB, Simulink, Stateflow, xPC Target, RealtimeWorkshop, Power Systems Toolbox, Optimization Toolbox, Symbolic Toolbox, Virtual Reality Toolbox 为 MathWorks 公司的注册商标

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

基于 MATLAB/Simulink 的系统仿真技术与应用/薛定宇,陈阳泉著. —2 版. —北京:清华大学出版社, 2011.2

ISBN 978-7-302-23880-5

I. ①基… II. ①薛… ②陈… III. ①算法语言-应用-自动控制系统-系统仿真 IV. ①TP273
②TP312

中国版本图书馆 CIP 数据核字(2010)第 183858 号

责任编辑:朱英彪

封面设计:张 岩

责任校对:王 云

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185×260 印 张:26.5 字 数:612 千字

版 次:2011 年 2 月第 2 版 印 次:2011 年 2 月第 12 次印刷

印 数:27501~32500

定 价:42.00 元

产品编号:038737-01

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越来越广，并且有着更广阔的应用前景和无穷无尽的潜能。

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；编写的几个通用程序在 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。

二十余年来，作者一直倡导在教学中引入 MATLAB，将其作为主要工具解决科学、工程仿真教学中的问题，1996 年，在清华大学出版社出版的《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 语言方面的书籍中出版最早、影响最广的著作，以 MATLAB 语言为主线的教学理念已经被广泛接受并成功地用于相关课程的教学，教材入选国家级精品教材，作者主讲的《控制系统仿真与 CAD》课程已列选为国家级精品课程。作者十余年来出版多部著作，并在美国出版社出版了两部英文著作，在 MATLAB 语言教学中的应用上有一定的造诣。

本书的合作者陈阳泉教授长期在美国 Utah 州立大学任教，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上都颇有建树。

本书第一版出版于 2002 年，出版以来，被很多学校选为教材和主要参考书，直到今年年初仍重印了一次，这在日新月异的计算机类书籍中是不多见的。多年来，随着 MATLAB/Simulink 版本的更新，出现了很多新内容，也出现了很多新的系统仿真方法和新的模块集，在一部教材中详细介绍所有的内容是不可能的，这使得本书新版的写作取舍困难，部分新内容最终成型于作者在上海宝山钢铁集团、美国 Utah 州立大学、东北大学和哈尔滨工业大学

等单位的讲座和报告(按报告顺序排序)。本书新版依旧从使用者的角度出发,并结合笔者二十余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,相信仍会受到广大读者的欢迎。

本书由东北大学信息学院徐心和教授主审。本书从酝酿到最终完稿整个写作过程始终得到了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 Atherton 教授也对本书的最终成型提供了很多的帮助,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教学与研究工作的。

一些同行和朋友也先后给予作者许多建议和支持,包括北京交通大学的朱衡君教授、中科院系统科学研究所的韩京清研究员、哈尔滨工业大学的张晓华教授等,还有在互联网上交流的众多知名的和不知名的同行,在此对他们表示深深的谢意。在本书部分内容的写作过程中,作者与同事潘峰博士、魏颖博士、陈大力博士、崔建江博士、高道祥博士、王良勇博士、方正博士等深入的交流为本书提供了许多新的观念和内容,在此一并表示感谢。

本书及第一版的出版得到了清华大学出版社蔡鸿程总编的关怀和帮助,还得到了欧振旭、朱英彪编辑细心的文字加工,作者对他们的辛勤工作深表谢意。

本书的出版还得到了美国 MathWorks 公司图书计划的支持,在此表示谢意。

由于作者水平有限,书中的缺点和错误在所难免,欢迎读者批评指正。作者电子邮箱为: xuedingyu@mail.neu.edu.cn 和 yqchen@ieee.org。

多年来,我的妻子杨军和女儿薛杨在生活和事业上给予了我莫大的帮助与鼓励,没有她们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,谨以此书献给她们。

薛定宇

2010年8月1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 仿真软件的发展概况	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 版本选择和建议	8
1.4 本书的结构和代码	8
1.4.1 本书的结构	8
1.4.2 代码下载和网上资源	9
1.4.3 书中英文字体说明	10
1.5 习 题	10
第2章 MATLAB 语言程序设计基础	11
2.1 MATLAB 语言的基本使用环境	11
2.1.1 MATLAB 语言环境操作界面	11
2.1.2 MATLAB 的联机帮助与电子版手册	11
2.2 MATLAB 语言的数据结构	12
2.2.1 常量与变量	13
2.2.2 赋值语句	13
2.2.3 矩阵的 MATLAB 表示	14
2.2.4 多维数组的定义	15
2.3 MATLAB 下矩阵的运算	16
2.3.1 矩阵的代数运算	16
2.3.2 矩阵的逻辑运算	19
2.3.3 矩阵的比较关系	20
2.3.4 矩阵元素的数据变换	20
2.4 流程控制结构	21
2.4.1 循环语句结构	21
2.4.2 条件转移语句结构	22

2.4.3	开关语句结构	23
2.4.4	试探式语句结构	24
2.5	MATLAB 函数编写与技巧	25
2.5.1	MATLAB 语言的函数的基本结构	25
2.5.2	可变输入、输出个数的处理	27
2.5.3	MATLAB 函数的跟踪调试	27
2.5.4	伪代码与代码保密处理	28
2.6	MATLAB 语言下图形的绘制与技巧	28
2.6.1	基本二维图形绘制语句	29
2.6.2	带有其他选项的绘图函数	30
2.6.3	二维曲线的标注方法	31
2.6.4	在 MATLAB 图形上添加文字标注	32
2.6.5	特殊图形绘制函数及举例	32
2.6.6	隐函数的曲线绘制	33
2.7	三维图形的绘制方法	34
2.7.1	三维曲线的绘制方法	34
2.7.2	三维曲面的绘制方法	35
2.7.3	局部图形的剪切处理	37
2.8	MATLAB 图形用户界面设计技术	38
2.8.1	图形界面设计工具 Guide	39
2.8.2	句柄图形学及句柄对象属性	40
2.8.3	菜单系统设计	44
2.8.4	界面设计举例与技巧	44
2.8.5	工具栏设计	49
2.9	提高 MATLAB 程序执行效率的技巧	51
2.9.1	测定程序执行时间和时间分配	51
2.9.2	加快 MATLAB 程序执行速度的建议	51
2.9.3	Mex 程序设计技术	53
2.10	习题	58
第3章	MATLAB 语言在现代科学运算中的应用	61
3.1	解析解与数值解	61
3.2	数值线性代数问题及求解	63
3.2.1	特殊矩阵的 MATLAB 输入	63
3.2.2	矩阵基本分析与运算	65
3.2.3	矩阵逆与广义逆运算	69
3.2.4	矩阵的相似变换与分解	70
3.2.5	矩阵的特征值与特征向量	74

3.2.6	代数方程求解	75
3.2.7	矩阵的非线性运算	79
3.3	微积分问题的 MATLAB 求解	81
3.3.1	微积分问题的解析解运算	81
3.3.2	数值差分与微分运算	82
3.3.3	数值积分运算	84
3.3.4	多重定积分的数值求解	85
3.4	常微分方程的数值解法	86
3.4.1	常微分方程的数值解法	86
3.4.2	MATLAB 下的常微分方程求解函数	87
3.4.3	微分方程组的变换和技巧	94
3.4.4	微分方程数值解正确性的验证	96
3.4.5	微分代数方程的数值解法	96
3.4.6	线性随机微分方程的数值解法	98
3.4.7	常微分方程的解析求解方法	101
3.5	非线性方程与最优化问题求解	101
3.5.1	非线性方程组求解	102
3.5.2	无约束最优化问题求解	105
3.5.3	线性规划问题	106
3.5.4	二次型规划问题	107
3.5.5	一般非线性规划问题求解	107
3.5.6	最优化问题的全局搜索解法	108
3.6	动态规划及其在路径规划中的应用	109
3.6.1	图的矩阵表示方法	109
3.6.2	有向图的路径寻优	109
3.6.3	无向图的路径最优搜索	112
3.7	数据插值与统计分析	112
3.7.1	一维数据的插值拟合	112
3.7.2	二维数据的插值拟合	113
3.7.3	最小二乘曲线拟合技术	116
3.7.4	数据简单排序	117
3.7.5	快速 Fourier 变换	117
3.7.6	数据分析与统计处理	118
3.8	习 题	122
第4章	Simulink 下数学模型的建立与仿真	129
4.1	Simulink 模块库简介	129
4.1.1	信号源模块组	131

4.1.2	连续模块组	132
4.1.3	离散模块组	133
4.1.4	查表模块组	134
4.1.5	用户自定义函数模块组	135
4.1.6	数学运算模块组	135
4.1.7	非线性模块组	136
4.1.8	输出池模块组	137
4.1.9	信号与系统模块组	137
4.1.10	子系统模块组	138
4.1.11	常用模块组	139
4.1.12	其他工具箱与模块集	139
4.2	Simulink 模型的建立	141
4.2.1	模型窗口建立	141
4.2.2	模块的连接与简单处理	141
4.2.3	模块的参数修改	144
4.3	模型的处理与仿真分析	146
4.3.1	模型建立及建模技巧	146
4.3.2	仿真模型的模块浏览	146
4.3.3	Simulink 模块的联机帮助系统	147
4.3.4	Simulink 模型的输出与打印	148
4.3.5	仿真环境的设置与启动	149
4.3.6	Simulink 模型的测试与调试工具	152
4.4	Simulink 模型举例	153
4.5	线性系统建模、仿真与分析	159
4.5.1	线性系统模型的输入	159
4.5.2	基于 Simulink 的线性系统分析界面	161
4.6	非线性随机系统的仿真方法	163
4.6.1	Simulink 下的随机信号仿真方法	163
4.6.2	仿真结果的统计分析	164
4.7	习题	167
第5章	Simulink 常用模块介绍与应用技巧	171
5.1	常用模块应用技巧	171
5.1.1	向量化模块举例	171
5.1.2	Simulink 模型的信号标识	174
5.1.3	线性多变量系统建模与仿真	175
5.1.4	Simulink 的代数环及消除方法	177
5.1.5	Simulink 的过零点检测与仿真	180

5.2	非线性环节与查表环节构建	182
5.2.1	单值非线性模块	182
5.2.2	多值非线性记忆模块	184
5.2.3	多维查表模块	187
5.2.4	静态非线性模块的代码实现	188
5.3	微分方程的 Simulink 框图求解	189
5.3.1	一般微分方程的 Simulink 建模	189
5.3.2	微分代数方程的 Simulink 建模与求解	190
5.3.3	延迟微分方程的 Simulink 求解	192
5.3.4	切换微分方程的 Simulink 求解	194
5.3.5	分数阶微分方程的 Simulink 求解	195
5.4	输出模块库	196
5.4.1	一般输出模块库	198
5.4.2	模型参数显示与模型浏览器	202
5.4.3	输出信号的表盘与量计显示	203
5.4.4	输出的数字信号处理	205
5.5	MATLAB/Simulink 仿真结果的三维动画显示	207
5.5.1	虚拟现实基础	208
5.5.2	VRealm 程序的配置与虚拟现实场景建模	209
5.5.3	在 MATLAB 下浏览虚拟现实场景	211
5.5.4	Simulink 下的三维动画场景驱动	211
5.6	子系统与模块封装技术	213
5.6.1	子系统的处理	214
5.6.2	条件执行子系统	214
5.6.3	模块封装技术	217
5.6.4	组建自己的模块库	223
5.6.5	子系统应用举例——F14 战斗机仿真	224
5.7	习 题	227
第6章	Simulink 仿真的高级技术	231
6.1	Simulink 模型的语句修改	231
6.1.1	Simulink 模型与文件的处理	231
6.1.2	Simulink 模型与模型文件	232
6.1.3	用语句绘制方框图	233
6.2	系统仿真与线性化	238
6.2.1	仿真过程的命令化	238
6.2.2	非线性模型的线性化	239
6.2.3	纯时间延迟环节的 Padé 近似	242

6.3	S-函数的编写及应用	245
6.3.1	用 MATLAB 语句编写 S-函数	245
6.3.2	S-函数设计与应用举例——自抗扰控制器仿真	248
6.3.3	二级 S-函数	254
6.3.4	用 C 语句编写 S-函数	257
6.3.5	S-函数模块的封装	261
6.4	仿真优化举例——控制系统最优控制器设计	261
6.4.1	伺服控制的最优性能指标选择	262
6.4.2	目标函数编写及最优控制器设计	263
6.4.3	全局最优化方法	266
6.5	习题	267
第7章	工程系统建模与仿真	271
7.1	物理系统建模仿真模块集 Simscape 简介	271
7.1.1	传统框图建模方法的局限性	271
7.1.2	Simscape 简介	273
7.1.3	Simscape 基础模块库简介	273
7.1.4	两类信号及其相互转换	276
7.1.5	Simscape 模块定义语言入门	278
7.1.6	复杂电路网络建模与仿真	279
7.2	电气系统模块集简介	280
7.3	电子线路及其仿真	286
7.3.1	SimElectronics 模块集简介	286
7.3.2	模拟电子线路仿真举例	288
7.3.3	数字电子线路仿真举例	291
7.3.4	电力电子系统仿真	294
7.3.5	将 Spice 模块嵌入 Simulink 仿真模型	295
7.4	电机拖动系统仿真	297
7.4.1	直流电机拖动系统仿真	297
7.4.2	交流电机拖动仿真	301
7.5	机械系统建模与仿真	305
7.5.1	简单力学系统的仿真	305
7.5.2	SimMechanics 简介	307
7.5.3	机构系统仿真举例	310
7.5.4	Simulink 与其他 CAD 软件的接口	315
7.6	习题	317
第8章	非工程系统的建模与仿真	319
8.1	药物动力学系统建模与仿真	319

8.1.1	药物动力学系统简介	319
8.1.2	药物动力学系统的舱室模型	320
8.1.3	基于生理的药物动力学建模及 Simulink 仿真	322
8.1.4	药效学建模	328
8.1.5	麻醉过程的非线性广义预测控制	329
8.2	影像与图像处理系统	330
8.2.1	图像与影像读取	330
8.2.2	图像与影像的显示与输出	331
8.2.3	图像处理基本模块介绍	333
8.2.4	图像与影像的处理入门	336
8.2.5	图像影像的实时处理	341
8.3	有限状态机仿真及 Stateflow 应用	342
8.3.1	有限状态机简介	343
8.3.2	Stateflow 应用基础	343
8.3.3	Stateflow 的常用命令	347
8.3.4	Stateflow 应用举例	347
8.3.5	用 Stateflow 描述流程	358
8.4	基于 SimEvents 的离散事件系统仿真方法	359
8.4.1	离散事件动态系统基本概念	359
8.4.2	SimEvents 模块集简介	359
8.4.3	排队服务模型建模与仿真	362
8.5	习 题	366
第9章	半实物仿真与实时控制	367
9.1	Simulink 仿真的实时工具 RTW	367
9.1.1	半实物仿真简介	367
9.1.2	独立程序的自动生成	369
9.1.3	实时仿真与目标计算机仿真	370
9.1.4	基于 xPC Target 的半实物仿真方法	374
9.2	dSPACE 简介与常用模块	378
9.2.1	dSPACE 简介	378
9.2.2	dSPACE 模块组	378
9.3	Quanser 简介与常用模块	380
9.3.1	Quanser 简介	380
9.3.2	Quanser 常用模块介绍	380
9.3.3	Quanser 旋转运动控制系列实验受控对象简介	381
9.4	半实物仿真与实时控制实例	382
9.4.1	受控对象的数学描述与仿真研究	382

9.4.2	Quanser 实时控制实验	384
9.4.3	dSPACE 实时控制实验	385
9.5	基于 NIAT 的低成本半实物仿真简介	387
9.5.1	NIAT 模块集常用模块简介	388
9.5.2	Pendubot 系统的数学模型、控制与仿真	388
9.5.3	Pendubot 控制系统的半实物仿真实验	392
9.6	习 题	393
	参考文献	395
	附录 A MATLAB 函数索引	399
	附录 B 关键词索引	405

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分为工程系统和非工程系统。工程系统是指由相互关联的部件组成一个整体,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、功率转换部件和检测部件所组成,用来完成电机的转速、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。根据数学模型所描述的系统的运动性质和数学工具来划分,又可以分为连续系统、离散时间系统、离散事件系统和混杂系统等;还可以细分为线性、非线性、定常、时变、集中参数、分布参数、确定性和随机等子类。

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,它直接决定原来的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,通过学习可以初步领略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 仿真软件的发展概况

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。20世纪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20世纪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。20世纪70年代初开始应用数字计算机进行仿真^[1]。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和CSMP-C等,随着MATLAB/Simulink等仿真工具的日益强大,前面列出的很多仿真语言已经退出了历史舞台,而MATLAB/Simulink已经成为仿真领域事实上的首选计算机语言和工具。

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^[2,3]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^[5]及参考文献[6]中给出的Numerical Recipes程序集等,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 W_R 、 W_I 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,然后依据EISPACK的格式做出定义和赋值,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,两者中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数百页的使用手册就十几本!

Numerical Recipes一书^[6]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,被各国的研究者所信赖。

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。例如,若想求解一个线性代数方程,用户需要首先编写一个主程序,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。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麻烦的,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,也容易出错。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,如参考文献 [7] 中的程序维数均定为 10 阶。所以,有时为使程序通用,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,而更大规模的问题仍然求解不了。在优秀的软件中往往需要动态地定维矩阵。

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^[8],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。

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 Linux 操作系统上直接运行,而需要在该操作系统上对源程序进行修改、重新编译后才可以执行。

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^[9],但它们的目的是似乎已经不再是为一般用户提

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 中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好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, 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)^[10]。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一个模型文件,然后通过 ACSL 本身提供的命令对其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,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 和限幅积分器 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,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,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。

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 $\dot{y}_1 = y_1(1 - y_2^2) - y_2, \dot{y}_2 = y_1$,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)
```