

新世纪电子信息课程系列规划教材

信号与系统综合实验教程

SHENHAO YU XITONG ZONGHE
SHIYAN JIAOCHENG

主 编 陆 毅 杨 艳

副主编 刘 强 李 鹏

刘晓杰



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

新世纪电子信息课程系列规划教材

信号与系统综合实验教程

主 编 陆 毅 杨 艳

副主编 刘 强 李 鹏 刘晓杰

东南大学出版社
• 南京 •

图书在版编目(CIP)数据

信号与系统综合实验教程/陆毅,杨艳主编. —南京:
东南大学出版社, 2010. 10

新世纪电子信息课程系列规划教材

ISBN 978-7-5641-2429-8

I. ①信… II. ①陆… ②杨… III. ①信号系统—实验—高等
学校—教材 IV. ①TN911.6-33

中国版本图书馆 CIP 数据核字(2010)第 176748 号

信号与系统综合实验教程

出版发行 东南大学出版社

出版人 江 汉

社 址 南京市四牌楼 2 号

邮 编 210096

经 销 全国新华书店

印 刷 南京新洲印刷有限公司

开 本 787 mm×1092 mm 1/16

印 张 7.25

字 数 187 千字

书 号 ISBN 978-7-5641-2429-8

印 次 2010 年 10 月第 1 次印刷

版 次 2010 年 10 月第 1 版

印 数 1—3000 册

定 价 18.00 元

(凡有印装质量问题,请与我社读者服务部联系。电话:025—83792328)

前 言

“信号与系统”课程是一门实用性较强、涉及面较广的专业基础课。该课程是将学生从电路分析的知识领域引入信号处理与传输领域的关键性课程,对后续专业课起到承上启下的作用。长期以来,“信号与系统”课程一直采用黑板式的单一教学方式,课程中大量信号分析结果缺乏可视化的直观表现,学生仅依靠做习题来巩固和理解教学内容,对课程中大量的应用性较强的内容不能实际动手设计、调试、分析,严重影响和制约了教学效果。近几年来,为适应 21 世纪科学技术的发展,加强基础性教育、增强适应性训练、注重人才综合素质的培养,已成为社会各界的共识。为了达到能让学生在“信号与系统”课程中学习到扎实的基础理论知识并与现代科学技术发展紧密结合的要求,实验性教学环节越来越受到教育工作者的重视。如何开设“信号与系统”实验课、开设哪些方面的内容,是十分重要的。本书从软件仿真实验到硬件实验平台进行了全方位的介绍,并给出了详细的讲解过程。

MATLAB 和 C 语言强大便利的计算编程功能,使越来越多的科技工作者将它作为编程语言。本书深入浅出地介绍了 MATLAB 和 C 语言在“信号与系统”课程中的实际应用。首先简明扼要地对工程软件进行概述,使读者轻松入门;然后深入介绍 MATLAB 在时域、频域和复频域中的实现过程;最后补充介绍 C 语言在信号与系统分析中的应用。另外,本书还提供了 MATLAB 常用命令大全,极大地方便了读者的阅读与参考。

“信号与系统”课程主要包含确定信号经过线性时不变系统所涉及的基本概念与基本分析方法。JH5004 型信号与系统实验箱紧密围绕当前《信号与系统》课程的核心内容,建设了一系列具有特色的实验项目,该硬件实验平台在基础性、实用性、全面性、简洁性及扩展性上都得到了很好的体现,便于教师对实验内容的组织和实施,丰富了实验教学手段。

本书共分 8 章。第 1 章简单介绍 MATLAB 的发展过程及特点;第 2 章针对性地介绍与本课程相关的 MATLAB 知识,主要包括程序设计环境、基本操作、绘图功能及 M 文件等;第 3 章和第 4 章详细介绍了 MATLAB 在连续信号与系统的时域分析和频域分析中的实现,这也是 MATLAB 在信号与系统中最基本的应用之一;第 5 章和第 6 章深入介绍了 MATLAB 在离散信号与系统的时域分析和频域分析中的实现;第 7 章运用 Turbo C 开发平台进行信号及其频谱分析的实现,让学生

掌握 C 语言在信号与系统中的基本应用;第 8 章详细介绍了 JH5004 型信号与系统实验箱各模块的使用方法及注意事项,进一步加强对学生动手能力的训练。

本书由江苏技术师范学院的陆毅和安徽蚌埠学院的杨艳担任主编,吉林北华大学的刘强、江西九江学院的李鹏以及江苏技术师范学院的刘晓杰担任副主编。第 1 章由李鹏编写,第 2 章由杨艳编写,第 3 章由陆毅和李鹏共同编写,第 4 章由陆毅和杨艳共同编写,第 5 章由吉林北华大学的刘强和宋艳霞执笔,第 6 章由吉林北华大学的刘强和邢砾云编写,第 7 章由刘晓杰、李鹏以及杨艳编写,第 8 章由上述老师共同编写。

本书在编写过程中得到了江苏技术师范学院、吉林北华大学、江西九江学院、安徽蚌埠学院和苏州科技学院(排名不分先后)的大力支持和帮助,并给予了許多宝贵的意见,编者在此表示衷心的感谢。

在本书编写过程中参阅了大量的论文与著作,并得到了许多同行的指导,在此表示衷心感谢。

由于编者水平有限,加之时间仓促,书中难免有疏漏和不妥之处,欢迎读者批评指正。

编 者
2010 年 6 月

目 录

1	MATLAB 基础	(1)
1.1	MATLAB 的起源和发展	(1)
1.2	MATLAB 的主要特点	(1)
1.3	MATLAB 的组成	(2)
2	MATLAB 程序设计	(4)
2.1	创建、保存和编辑 M 文件	(4)
2.2	MATLAB 控制语句	(5)
2.2.1	条件控制语句	(5)
2.2.2	循环控制语句	(7)
2.3	程序的调试和优化	(8)
2.3.1	程序的调试	(8)
2.3.2	程序的优化	(9)
3	MATLAB 对连续信号与系统的时域分析	(10)
3.1	常用连续信号的 MATLAB 实现	(10)
3.2	连续信号运算的 MATLAB 实现	(13)
3.2.1	连续信号的基本运算	(13)
3.2.2	连续信号的卷积计算	(17)
3.3	连续系统的响应	(19)
3.3.1	连续系统的冲激响应和阶跃响应	(19)
3.3.2	连续系统的零状态响应和全响应	(20)
3.4	连续信号的采样和恢复	(22)
3.4.1	连续信号的采样	(22)
3.4.2	连续信号的恢复	(23)
4	MATLAB 对连续信号与系统的频域分析和 s 域分析	(26)
4.1	连续信号的傅里叶变换	(26)
4.1.1	常用连续信号的傅里叶变换	(26)
4.1.2	傅里叶变换的性质	(27)

4.2	连续系统的频率响应	(29)
4.3	连续信号与系统的拉普拉斯变换和拉普拉斯反变换	(31)
4.3.1	拉普拉斯变换的 MATLAB 表示	(31)
4.3.2	MATLAB 实现拉普拉斯反变换	(31)
4.4	连续系统函数 $H(s)$ 的零极点分布和稳定性	(32)
4.5	连续系统状态方程的 MATLAB 求解	(34)
5	MATLAB 对离散信号与系统的时域分析	(36)
5.1	常用离散信号的 MATLAB 表示	(36)
5.2	离散信号运算的 MATLAB 实现	(40)
5.2.1	离散信号的基本运算	(40)
5.2.2	离散信号的卷积计算	(42)
5.3	离散系统的响应	(44)
5.3.1	离散系统的冲激响应和单位阶跃响应	(44)
5.3.2	离散系统的零状态响应和全响应	(45)
6	MATLAB 对离散信号与系统的频域分析和 z 域分析	(48)
6.1	离散系统的傅里叶变换	(48)
6.2	离散系统的 z 域分析	(51)
6.2.1	离散系统 z 正变换的 MATLAB 表示	(51)
6.2.2	MATLAB 实现离散系统的 z 反变换	(51)
6.3	离散系统的频率响应	(53)
6.4	离散系统函数的零极点分布与稳定性	(54)
6.5	离散系统状态方程的求解	(56)
7	运用 C 语言实现信号与系统的分析	(58)
7.1	单位阶跃响应上升时间的计算	(58)
7.2	单位阶跃响应过冲的计算	(60)
7.3	自相关函数的计算	(61)
7.4	sinc 函数积分的计算	(64)
7.5	罗斯稳定性准则	(66)
7.6	拉普拉斯变换留数的计算	(69)
7.7	巴特沃斯低通滤波器阶数的计算	(73)
7.8	切比雪夫低通滤波器阶数的计算	(76)
7.9	离散卷积的计算	(79)
7.10	状态变量方程的求解	(81)

8	JH5004 型信号与系统实验箱	(84)
8.1	JH5004 型信号与系统实验箱的特点	(84)
8.2	JH5004 型信号与系统实验箱电路的组成	(84)
8.3	JH5004 型信号与系统实验箱信号产生模块的使用方法	(85)
8.3.1	实验 1:常用信号的分类和观察	(85)
8.3.2	实验 2:信号的基本运算单元	(90)
8.3.3	实验 3:信号的合成	(93)
8.3.4	实验 4:线性时不变系统的测量	(94)
8.3.5	实验 5:零输入响应与零状态响应的分析	(96)
8.3.6	实验 6:信号的采样和恢复(PAM)	(98)
8.3.7	实验 7:一阶网络特性的测量	(100)
8.3.8	实验 8:二阶网络特性的测量	(101)
参考文献		(105)

1.1 MATLAB 的起源和发展

MATLAB 语言是由美国 Clever Moler 博士于 1980 年开发的,设计者的初衷是为解决《线性代数》课程的矩阵运算问题。MATLAB 的名称源自 Matrix Laboratory。

1983 年春,Clever Moler 到 Standford 大学讲学,他和 John Little 主持开发了各类数学分析的子模块,撰写用户指南和大部分的 M 文件,用 C 语言开发了第二代 MATLAB 专业版,也是 MATLAB 第一个商用版,同时赋予了它数值计算和数据图示化的功能。自第一版发行以来,已有众多的科技工作者加入到 MATLAB 的开发队伍中。1984 年,他们成立了 MathWorks 公司,发行了 MATLAB 第一版。MATLAB 的第一个商业化的版本是同年推出的 3.0 版。后来逐步将其发展成为一个集数值处理、图形处理、图像处理、符号计算、文字处理、数学建模、实时控制、动态仿真、信号处理为一体的数学应用软件。

20 世纪 90 年代初,MATLAB 在数值计算方面独占鳌头,Mathematica 和 Mathcad 分居符号计算软件的前两名。MATLAB 成为国际控制界公认的标准计算软件。

MathWorks 公司于 1992 年推出了 4.0 版;1993 年推出了 4.1 版;1994 年 4.2 版本扩充了 4.0 版的功能,在图形界面设计方面提供了新的方法;1995 年推出了 4.2 C 版;1997 年推出了 5.0 版,允许更多的数据结构,使其成为一种更方便编程的语言;1999 年推出了 5.3 版,进一步改进了 MATLAB 语言的功能;2000 年 10 月推出了全新的 6.0 正式版,其在核心数值算法、界面设计、外部接口、应用桌面等方面有了很大改进;2006 年 9 月推出了 MATLAB R2006b;2007 年 3 月发布 MATLAB R2007a。2008 年发布 MATLAB R2008Bf。

MATLAB 语言是当今国际上科学界最具影响力、最有活力的软件之一,已发展成一种高度集成的计算机语言。它提供了强大的科学运算、灵活的程序设计流程、高质量的图形可视化与界面设计、便捷的与其他程序和语言接口的功能。MATLAB 语言在各国高校与研究单位起着重要的作用。

1.2 MATLAB 的主要特点

1) 编程效率

MATLAB 是一种面向科学与工程计算的高级语言,允许用数学形式的语言编写程序。用 MATLAB 编写程序犹如在演算纸上排列出公式与求解问题。因此,MATLAB 语言也可通俗地称为演算纸式科学算法语言,由于其编写简单,所以编程效率高。

2) 用户使用

MATLAB 语言是一种解释执行的语言,使用灵活、方便,其调试程序手段丰富,调试速度快,需要学习的时间少。人们用任何一种语言编写程序和调试程序一般都要经过四个步骤:编

辑、编译、连接以及执行和调试。MATLAB 语言与其他语言相比,较好地解决了上述问题,把这四个步骤融为一体,能在同一幅画面上进行灵活操作,快速排除输入程序中的书写错误、语法错误以至语意错误,从而加快用户编写、修改和调试程序的速度。

3) 扩充能力

高版本的 MATLAB 语言具有丰富的库函数,在进行复杂的数学运算时可以直接调用,而且 MATLAB 的库函数与用户文件在形成上一样,所以用户文件也可作为 MATLAB 的库函数来调用。因此,用户可以根据自己的需要方便地建立和扩充新的库函数,以便提高 MATLAB 使用效率和扩充其功能。另外,它能方便地调用有关的其他语言的子程序。

4) 语言功能

MATLAB 语言中最基本、最重要的成分是函数,一个函数是由函数名、输入变量($d, e, f, \dots$)、输出变量($a, b, c, \dots$)组成,同一函数名 F ,不同数目的输入变量(包括无输入变量)及不同数目的输出变量,代表不同的含义。这不仅使 MATLAB 的库函数功能更丰富,而且大大减少了需要的磁盘空间,使得 MATLAB 编写的 M 文件简单、短小且高效。

5) 矩阵和数组运算

MATLAB 语言规定了矩阵的算术运算符、关系运算符、逻辑运算符、条件运算符及赋值运算符,而且这些运算符大部分可以毫无改变地照搬到数组间的运算,有些如算术运算符只要增加“.”就可用于数组间的运算,另外,它不需定义数组的维数,并给出矩阵函数、特殊矩阵专门的库函数,使之在求解诸如信号处理、建模、系统识别、控制、优化等领域的问题时,显得大为简洁、高效、方便,这是其他高级语言所不能比拟的。在此基础上,高版本的 MATLAB 已逐步扩展到科学及工程计算的其他领域。

6) 绘图功能

MATLAB 的绘图十分方便,它有一系列绘图函数,例如线性坐标、对数坐标,半对数坐标及极坐标,均只需调用不同的绘图函数,在图上标出图题、X-Y 轴标注,格绘制也只需调用相应的命令,简单易行。另外,在调用绘图函数时,调整自变量可绘出不同颜色的点、线、复线或多重线。这种为科学研究着想的设计是通用的编程语言所不及的。

7) 容错功能

MATLAB 的容错能力很强,而且可靠。

8) 兼容与接口功能

MATLAB 具有应用灵活的兼容与接口功能。

9) 联机检索功能

MATLAB 具有信息量丰富的联机检索功能。

1.3 MATLAB 的组成

MATLAB 系统是由 MATLAB 开发环境、MATLAB 数学函数库、MATLAB 语言、MATLAB 图形处理系统、MATLAB 应用程序接口、MATLAB 的专用领域箱、MATLAB Compiler、MATLAB Simulink、Stateflow 和 Real-time Workshop 等部分组成。

1) MATLAB 开发环境

MATLAB 开发环境是一套方便用户使用 MATLAB 函数和文件的工具集,其中许多工具

是图形化用户接口。它是一个集成化的工作空间,可以让用户输入、输出数据,并提供了 M 文件的集成和调试环境。它包括 MATLAB 桌面、命令窗口、M 文件编辑调试器、MATLAB 工作空间和在线帮助文档。

2) MATLAB 数学函数库

MATLAB 数学函数库包括大量的计算算法,从基本算法(如加法等)到复杂算法(如矩阵求逆、快速傅里叶变换等)。

3) MATLAB 语言

MATLAB 语言是一个高级的基于矩阵、数组的语言,具有程序流控制、函数、数据结构、输入输出和面向对象编程等特色。

4) MATLAB 图形处理系统

图形处理系统使得 MATLAB 能方便地图形化显示向量和矩阵,而且能对图形添加标注和打印。它包括二维、三维图形函数、图像处理和动画显示等函数。

5) MATLAB 应用程序接口

MATLAB 应用程序接口是一个使 MATLAB 语言能与 C 等其他高级编程语言进行交互的函数库,该函数库的函数通过调用动态链接库实现与 MATLAB 文件的数据交换。

6) MATLAB 的专用领域工具箱

许多学科在 MATLAB 中都有专用工具箱,现已有 30 多个工具箱,但 MATLAB 语言的扩展开发还远远没有结束,各学科的相互促进,将使得 MATLAB 更加强大。MATLAB 的常用工具箱有:①MATLAB 主工具箱;②Simulink 仿真工具箱;③信号处理工具箱;④通信工具箱;⑤神经网络工具箱;⑥符号数学工具箱;⑦控制系统工具箱;⑧图像处理工具箱;⑨系统辨识工具箱;⑩金融工具箱。

7) MATLAB Compiler

MATLAB Compiler(编译器)提供了 MATLAB 语言编写的 M 文件自动转换成 C 或者 C++ 格式文件的能力,支持用户进行独立应用开发。利用 MATLAB Compiler,用户可以快速开发出功能强大的独立应用。

8) MATLAB Simulink

Simulink 是一个对系统进行建模、仿真和分析的软件包。它既可以仿真线性系统,又可以仿真非线性系统,使 MATLAB 功能得到进一步提高,比如可以实现可视化建模,实现与 M 文件的数据共享,将理论与工程实际有机结合在一起。

9) Stateflow

与 Simulink 的模型框结合,描述复杂事件驱动系统的逻辑行为,驱动系统在不同模块之间进行切换。

10) Real-time Workshop

Real-time Workshop 与 Stateflow 直接从 Simulink 模型与 Stateflow 框图中生成高效的可移植 C 代码或 Ada 代码。

2.1 创建、保存和编辑 M 文件

MATLAB 的语言结构一般可以归结为：

MATLAB 语言结构=窗口命令+M 文件

MATLAB 为用户提供了专用的 M 文件编辑器,便于完成 M 文件的创建、保存和编辑工作。

1) 创建 M 文件

用 M 文件编辑器创建新 M 文件可以有两种方式：

(1) 启动 MATLAB,选中命令窗口菜单栏【File】下【New】菜单选项的【M-File】命令,便可打开 MATLAB 的 M 文件编辑器窗口,如图 2.1 所示。

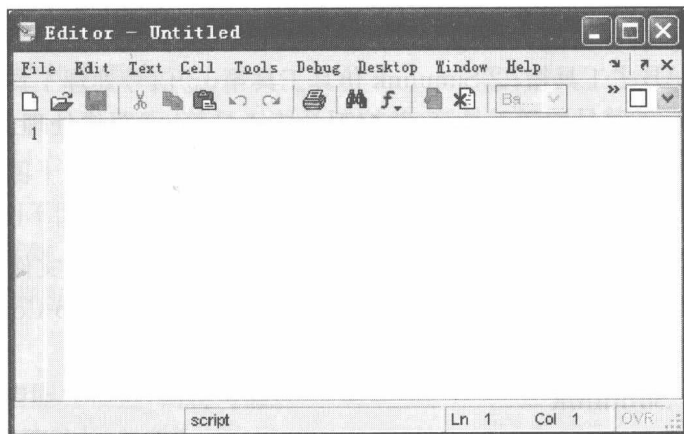


图 2.1 M 文件编辑器窗口

(2) 点击 MATLAB 命令窗口工具栏的“New M-File”图标,即可打开如图 2.1 所示的 M 文件编辑器。

2) 保存 M 文件

M 文件编辑完成后,可将 M 文件保存。保存 M 文件可以有两种方式：

(1) 点击 M 文件编辑器窗口下菜单栏【File】菜单的【Save】命令,弹出如图 2.2 所示的对话框。

(2) 点击 M 文件编辑器窗口工具栏的【Save】图标,弹出如图 2.2 所示的对话框。

对话框中,系统默认的文件保存目录为“work”,可以通过“文件保存”对话框对文件保存位置进行重新设定,系统默认的文件名“untitled”,对文件重新命名。



图 2.2 “M 文件保存”对话框

3) 编辑 M 文件

对已保存的文件进行修改和编辑可以有以下两种方式：

(1) 点击 MATLAB 命令窗口工具栏的“Open File”图标，也可以点击在命令窗口菜单栏【File】菜单下的【Open】，系统便会启动 M 文件编辑器并打开需要的 M 文件。

(2) 假设要进行编辑的 .m 文件名为 xinhaojuanji.m。在 MATLAB 命令窗口中输入命令：

```
>>edit xinhaojuanji 或者 edit xinhaojuanji.m
```

或者输入命令：

```
>>open xinhaojuanji 或者 open xinhaojuanji.m
```

系统即可打开 MATLAB 编辑器，并打开这个名为 xinhaojuanji.m 的文件开始编辑。若在当前路径中未找到此文件，则 MATLAB 会提示是否创建为此文件名的新的 .m 文件，然后打开 MATLAB 编辑器供编辑文件，并在保存时自动保存名为 xinhaojuanji.m。

2.2 MATLAB 控制语句

2.2.1 条件控制语句

1) if 语句

if 语句一般有两种格式：

(1) if 逻辑表达式

指令 1

else

指令 2

⋮

end

此语句先计算逻辑表达式的值，如果逻辑表达式的值为 true，则执行 if 后面的指令，如果逻辑

辑表达式的值为 false,则执行指令 2,以此类推下去。例如:

```

if a>2
    b=zeros(1,5)
else
    b=ones(1,5)
end    //若 a>2,则向量所有的元素值为 1,否则全为 0
(2) if 逻辑表达式 1
指令 1
elseif 逻辑表达式 2
指令 2
:
else 指令
end

```

若逻辑表达式 1 的值为 false,则跳过指令 1,检查逻辑表达式 2 的值,若其值为 true,则执行指令 2,否则跳过指令 2,检查逻辑表达式 3 的值,以此类推,当逻辑表达式 n 的值为 true 时,执行指令 n,否则跳过指令 n,直接执行 else 后面的指令。例如:

```

if i==j
    a(i,j)=0    //若 i=j,矩阵 a[i,j]为 0
elseif abs(i-j)==1
    a(i,j)=-1    //若 i 与 j 的差的绝对值为 1,矩阵 a[i,j]为-1
else
    a(i,j)=2    //否则矩阵 a[i,j]为 2
end

```

2) Switch 语句

Switch 语句可以根据一个变量或表达式的值执行特定的语句。Switch 语句的常用格式为:

```

switch 表达式 0
case 表达式 1 的值
指令 1
case 表达式 2 的值
指令 2
:
otherwise
指令
end

```

首先计算 Switch 后面表达式 1 的值,然后检查 case 子句的值与表达式 1 的值是否相等,若相等,则执行指令 1,否则继续检查第 2 个 case 子句,以此类推,若所有 case 子句的值都不等于表达式 1 的值,则执行 otherwise 子句后面的指令。例如:

```

Switch a(i).Marks
case 100

```

```

a(i).Rank='满分'    //得分为 100,等级为“满分”
case A
a(i).Rank='优秀'    //得分为 90 与 99 之间,等级为“优秀”
case B
a(i).Rank='良好'    //得分为 80 与 89 之间,等级为“良好”
case C
a(i).Rank='中等'    //得分为 70 与 79 之间,等级为“中等”
case D
a(i).Rank='及格'    //得分为 60 与 69 之间,等级为“及格”
otherwise
a(i).Rank='不及格'    //低于 60 分,等级为“不及格”
end

```

3) try-catch 语句

try-catch 提供了一种错误捕获机制。try-catch 语句的常用格式为:

```

try
命令 1
catch
命令 2~
end

```

命令 1 中的所有命令都要执行。若命令 1 中没有 MATLAB 错误出现,则在执行完命令 1 之后,程序就直接执行 end 语句,但是,若在执行命令 1 的过程中出现了 MATLAB 错误,程序就立即转到 catch 语句,然后执行命令 2。例如:

```

A=[1,1;2,2]
B=[1,1]
try
    A+A,A+B,B+B    //执行 A+B 时发生错误
catch
    disp('error ')
end
disp(lasterr)    //显示最后的错误信息

```

2.2.2 循环控制语句

1) for 语句

for 语句是循环语句,用于事先知道循环需要几次的情况下。for 语句的常用格式如下:

```

for 计数器=初始值:增量:终止值
    指令 1,指令 2,...,指令 n
end

```

在用 for 语句实现多次循环时,for 和 end 必须成对出现。增量没有指明时,默认值为 1,增量为正时,计数的值为初始值加一次增量,直至计数的值大于终止值;增量为负时,计数的值为初始值减一次增量,直至计数的值小于终止值。例如:

```
for i=1:1:10
    a(i)=2*i
end
```

则运行结果为:

```
a=2 4 6 8 10 12 14 16 18 20
```

很明显,上述程序实现的功能是实现 1 到 10 的整数的 2 倍。

2) While 语句

While 语句也是循环语句,但与 for 循环语句不同,它的循环次数是一个不定数,其用途更广阔。While 语句的常用格式为:

```
While 表达式
指令行
end
```

当表达式为 true 时,重复执行 While 和 end 的指令。若有 break 语句,执行到它时,就退出循环。例如:

```
s=1
n=1
while n<=100
s=s*n
n=n+1
end
```

上述程序实现的功能是 100!

2.3 程序的调试和优化

2.3.1 程序的调试

在 MATLAB 中可能存在两种类型的错误,即语法错误和运行错误,所以在 MATLAB 的程序编辑器中提供了相应的程序调试功能。对于识别错误的基本技巧,一般来说,语法错误比较容易识别,MATLAB 会提示相应位置的错误信息,以便于检查和定位。对于运行错误,比较难以识别,因为发生运行错误时,系统会自动终止对 M 文件的调用,这样就关闭了函数的工作区间,无法找到需要的数据信息。可以通过一些 MATLAB 提供的函数来辅助检查错误,表 2.1 为常用的错误识别函数;也可以通过对程序断点的设置来检查错误,表 2.2 为常用断点操作函数。

表 2.1 常用的错误识别函数

函数名	功能描述
echo	在函数运行时显示代码
disp	显示特定的值或信息
Sprintf&fprintf	显示不同格式和类型的数据
Whos	列出工作区间的所有变量
Size	显示矩阵的维数
keyboard	中断程序运行,允许用户从键盘进行交互操作
return	回复 keyboard 命令后函数的运行
Warning	显示特定的警告信息
Error	显示特定的错误信息
Lasterr	返回最后一次的错误
Lasterror	返回最后一次的错误及其相关信息
lastwarn	返回最后一次的警告

表 2.2 常用断点操作函数

函数名	功能描述
Dbstop	用于在 M 文件中设置断点
dbstatus	显示断点信息
Dbtype	显示 M 文件文本
Dbstep	该函数用于从断点处继续执行 M 文件
Dbstack	显示 M 文件执行时调用的堆栈
Dbup/dbdown	实现工作区间的切换
dbquit	结束调试状态

2.3.2 程序的优化

1) 程序好坏的判断

在 MATLAB 语言中,使用 profile 函数以及计时函数 tic 和 toc 来分析程序中各个部分的耗时情况,从而帮助用户找出程序中需要改进的地方。其中,函数 profile 在计算相对耗时以及查找文件执行过程中查看问题时最为有效,而函数 tic 和 toc 在计算绝对耗时更为有效。

2) 程序优化的基本技巧

循环运算是 MATLAB 中的最大弱点,在程序设计中,应当尽量避免使用循环运算。因为 MATLAB 是一门矩阵语言,它只是为向量和矩阵运算设计的。可以通过将 M 文件向量化来优化 M 文件。所谓向量化,就是使用向量和矩阵运算来代替 for 循环和 while 循环。为了给程序优化,还可以设置数据的预定义,即对于可能出现变量维数不断扩大的问题,应当预先估计变量可能出现的最大维数,进行预先定义。表 2.3 是使用 zeros 和 cell 函数进行预定义的示例。

表 2.3 zeros 和 cell 函数进行预定义

矩阵类型	函 数	示 例
数值型	zeros	y=zeros(1,50)
单元型	cell	B=cell(2,3);B{1,3}=1;3