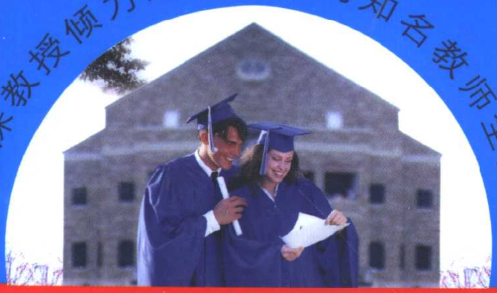


国外高校电子信息类优秀教材

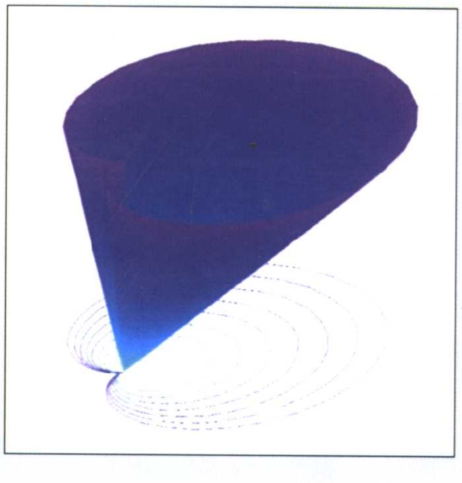
MATLAB 编程

（第二版）

MATLAB Programming for Engineers

(Second Edition)

(英文影印版)



Stephen J. Chapman 著

科学出版社
www.sciencep.com

国外高校电子信息类优秀教材(英文影印版)

MATLAB 编程

(第二版)

MATLAB Programming for Engineers

(Second Edition)

Stephen J. Chapman 著

科学出版社

北 京

内 容 简 介

本书为国外高校电子信息类优秀教材(英文影印版)之一。

本书详细讲述了如何用 MATLAB 进行程序设计,如何编写清楚、高效的程序。书中强调了自上而下的程序设计方法、函数的使用、MATLAB 内部工具的使用和数据结构,并指出了一些使用技巧和编程者常犯的错误。

本书可作为工科各专业本科生的教学辅导书,也可作为工程技术人员的参考书。

图字:01-2002-5256

图书在版编目(CIP)数据

MATLAB 编程:第 2 版/(美)查普曼(Chapman, S. J.)等著.—影印本.—北京:科学出版社,2003
(国外高校电子信息类优秀教材)
ISBN 7-03-011139-7

Ⅰ.M… Ⅱ.查… Ⅲ.计算机辅助计算-软件包,MATLAB-程序设计-高等学校-教材-英文 Ⅳ.TN391.75

中国版本图书馆 CIP 数据核字(2003)第 005593 号

责任编辑:巴建芬 李 宇/封面设计:黄华斌 陈 敬/责任印制:刘秀平

科学出版社 出版
北京东黄城根北街16号
邮政编码:100717
http://www.sciencep.com
双青印刷厂 印刷
科学出版社发行 各地新华书店经销

*

2003年3月第 一 版 开本:787×1092 1/16
2003年3月第一次印刷 印张:31 1/4
印数:1—3 000 字数:711 000

定价:41.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

This book is dedicated to my wife, Rosa, after 25 wonderful years together.

Preface

MATLAB (short for MATrix LABoratory) is a special-purpose computer program optimized to perform engineering and scientific calculations. It started life as a program designed to perform matrix mathematics, but over the years it has grown into a flexible computing system capable of solving essentially any technical problem.

The MATLAB program implements the MATLAB language, and provides a very extensive library of predefined functions to make technical programming tasks easier and more efficient. This extremely wide variety of functions makes it much easier to solve technical problems in MATLAB than in other languages such as Fortran or C. This book introduces the MATLAB language, and shows how to use it to solve typical technical problems.

This book teaches MATLAB as a technical programming language, showing students how to write clean, efficient, and well-documented programs. It makes no pretense at being a complete description of all of MATLAB's hundreds of functions. Instead, it teaches the student how to use MATLAB as a language, and how to locate any desired function with MATLAB's extensive online help facilities.

The first six chapters of the text are designed to serve as the text for an "Introduction to Programming / Problem Solving" course for freshman engineering students. This material should fit comfortably into a 9-week, 3-hour course. The remaining chapters cover advanced topics such as input/output and graphical user interfaces. These chapters may be covered in a longer course, or used as a reference by engineering students or practicing engineers who use MATLAB as a part of their coursework or employment.

Changes in the Second Edition

The second edition of this book is specifically devoted to MATLAB versions 6.0 and 6.1. While the basic MATLAB language has been largely constant since the release of version 5.0, the integrated tools, the windows, and the help subsystem have changed dramatically. In addition, a completely new paradigm for the design of MATLAB GUIs was introduced in version 6.0. Users working with versions of MATLAB before version 6.0 should be aware that the description of GUI development in Chapter 10 does not apply to them.

The book also covers the minor but important improvements in the language itself, such as the introduction of the `continue` statement.

The Advantages of MATLAB for Technical Programming

MATLAB has many advantages compared to conventional computer languages for technical problem solving. Among them are:

1. **Ease of Use**
   MATLAB is an interpreted language, like many versions of Basic. Like Basic, it is very easy to use. The program can be used as a scratch pad to evaluate expressions typed at the command line, or it can be used to execute large prewritten programs. Programs may be easily written and modified with the built-in integrated development environment, and debugged with the MATLAB debugger. Because the language is so easy to use, it is ideal for educational use, and for the rapid prototyping of new programs.

   Many program development tools are provided to make the program easy to use. They include an integrated editor/debugger, online documentation and manuals, a workspace browser, and extensive demos.

2. **Platform Independence**
   MATLAB is supported on many different computer systems, providing a large measure of platform independence. At the time of this writing, the language is supported on Windows 9x/NT/2000 and many different versions of UNIX. Programs written on any platform will run on all of the other platforms, and data files written on any platform may be read transparently on any other platform. As a result, programs written in MATLAB can migrate to new platforms when the needs of the user change.

3. **Predefined Functions**
   MATLAB comes complete with an extensive library of predefined functions that provide tested and prepackaged solutions to many basic technical tasks. For example, suppose that you are writing a program that must calculate the statistics associated with an input data set. In most languages, you would need to write your own subroutines or functions to implement calculations such as the arithmetic mean, standard deviation, median, etc. These and hundreds of other functions are built right into the MATLAB language, making your job much easier.

   In addition to the large library of functions built into the basic MATLAB language, there are many special-purpose toolboxes available to help solve complex problems in specific areas. For example, a user can buy standard toolboxes to solve problems in signal processing, control systems, communications, image processing, and neural networks, among many others.

4. **Device-Independent Plotting**
   Unlike other computer languages, MATLAB has many integral plotting and imaging commands. The plots and images can be displayed on any graphical output device supported by the computer on which MATLAB is running. This capability makes MATLAB an outstanding tool for visualizing technical data.

5. **Graphical User Interface**
   MATLAB includes tools that allow a programmer to interactively construct a graphical user interface (GUI) for his or her program. With this capability, the programmer can design sophisticated data analysis programs that can be operated by relatively inexperienced users.

6. **MATLAB Compiler**
   MATLAB's flexibility and platform independence is achieved by compiling MATLAB programs into a device-independent p-code, and then interpreting the p-code instructions at run-time. This approach is similar to that used by Microsoft's Visual Basic language. Unfortunately, the resulting programs can sometimes execute slowly because the MATLAB code is interpreted rather than compiled. We will point out features that tend to slow program execution when we encounter them.
   A separate MATLAB compiler is available. This compiler can compile a MATLAB program into a true executable that runs faster than the interpreted code. It is a great way to convert a prototype MATLAB program into an executable suitable for sale and distribution to users.

Features of This Book

Many features of this book are designed to emphasize the proper way to write reliable MATLAB programs. These features should serve a student well as he or she is first learning MATLAB, and should also be useful to the practitioner on the job. They include:

1. **Emphasis on Top-Down Design Methodology**
   The book introduces a top-down design methodology in Chapter 3, and then uses it consistently throughout the rest of the book. This methodology encourages a student to think about the proper design of a program *before* beginning to code. It emphasizes the importance of clearly defining the problem to be solved and the required inputs and outputs before any other work is begun. Once the problem is properly defined, it teaches the student to employ stepwise refinement to break the task down into successively smaller subtasks, and to implement the subtasks as separate

subroutines or functions. Finally, it teaches the importance of testing at all stages of the process, both unit testing of the component routines and exhaustive testing of the final product.

The formal design process taught by the book may be summarized as follows:

- Clearly state the problem that you are trying to solve.
- Define the inputs required by the program and the outputs to be produced by the program.
- Describe the algorithm that you intend to implement in the program. This step involves top-down design and stepwise decomposition, using pseudocode or flow charts.
- Turn the algorithm into MATLAB statements.
- Test the MATLAB program. This step includes unit testing of specific functions, and also exhaustive testing of the final program with many different data sets.

2. **Emphasis on Functions**
The book emphasizes the use of functions to logically decompose tasks into smaller subtasks. It teaches the advantages of functions for data hiding. It also emphasizes the importance of unit testing functions before they are combined into the final program. In addition, the book teaches about the common mistakes made with functions, and how to avoid them.

3. **Emphasis on MATLAB Tools**
The book teaches the proper use of MATLAB's built-in tools to make programming and debugging easier. The tools covered include the Launch Pad, Editor / Debugger, Workspace Browser, Help Browser, and GUI design tools.

4. **Good Programming Practice Boxes**
These boxes highlight good programming practices when they are introduced for the convenience of the student. In addition, the good programming practices introduced in a chapter are summarized at the end of the chapter. An example Good Programming Practice box is shown below.

Good Programming Practice

Always indent the body of an `if` construct by 2 or more spaces to improve the readability of the code.

5. **Programming Pitfalls Boxes**
These boxes highlight common errors so that they can be avoided. An example Programming Pitfalls box follows.

Programming Pitfalls

Make sure that your variable names are unique in the first 31 characters. Otherwise, MATLAB will not be able to tell the difference between them.

6. **Emphasis on Data Structures**
   Chapter 7 contains a detailed discussion of MATLAB data structures, including sparse arrays, cell arrays, and structure arrays. The proper use of these data structures is illustrated in the chapters on Handle Graphics and Graphical User Interfaces.

Pedagogical Features

The first six chapters of this book are specifically designed to be used in a freshman "Introduction to Programming / Problem Solving" course. It should be possible to cover this material comfortably in a 9-week, 3-hour course. If there is insufficient time to cover all of the material in a particular Engineering program, Chapter 6 may be deleted, and the remaining material will still teach the fundamentals of programming and using MATLAB to solve problems. This feature should appeal to harassed engineering educators trying to cram ever more material into a finite curriculum.

The remaining chapters cover advanced material that will be useful to the engineer and students as they progress in their careers. This material includes advanced input/output and the design of graphical user interfaces for programs.

The book includes several features designed to aid student comprehension. A total of 15 quizzes appear scattered throughout the chapters, with answers to all questions included in Appendix B. These quizzes can serve as a useful self-test of comprehension. In addition, there are approximately 140 end-of-chapter exercises. Answers to selected exercises are available at the book's Web site, and of course answers to all exercises are included in the Instructor's Manual. Good programming practices are highlighted in all chapters with special Good Programming Practice boxes, and common errors are highlighted in Programming Pitfalls boxes. End-of-chapter materials include Summaries of Good Programming Practice and Summaries of MATLAB Commands and Functions.

The book is accompanied by an Instructor's Manual containing the solutions to all end-of-chapter exercises. The source code for all examples in the book is available from the book's Web site, and the source code for all solutions in the Instructor's Manual is available separately to instructors.

A Final Note to the User

No matter how hard I try to proofread a document like this book, it is inevitable that some typographical errors will slip through and appear in print. If you should spot any such errors, please drop me a note via the publisher, and I will do my best to get them eliminated from subsequent printings and editions. Thank you very much for your help in this matter.

I will maintain a complete list of errata and corrections at the book's Web site, which is http://info.brookscole.com/chapman. Please check that site for any updates and/or corrections.

Acknowledgments

I would like to thank Bill Stenquist and the crew at Brooks/Cole for the support they have given me in getting this book to market. It has been gratifying to see the user response to the first edition, which was the result of our joint efforts.

In addition, I would like to thank my wife, Rosa, and our children, Avi, David, Rachel, Aaron, Sarah, Naomi, Shira, and Devorah, for being such delightful people, and the inspiration for my efforts.

Stephen J. Chapman

Contents