国际著名数学图书——影印版

Spectral Methods in MATLAB

MATLAB中的谱方法

Lloyd N. Trefethen 著

TSINGHUA UNIVERSITY PRESS

siam

Spectral Methods in MATLAB

MATLAB中的谱方法

Lloyd N. Trefethen 著

TSINGHUA UNIVERSITY PRESS
北京

siam

Lloyd N. Trefethen
Spectral Methods in MATLAB
ISBN: 0-89871-465-6

Copyright © 2000 by SIAM.

Original American edition published by SIAM: Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania. All Rights Reserved.

本书原版由SIAM出版。版权所有，盗印必究。

Tsinghua University Press is authorized by SIAM to publish and distribute exclusively this English language reprint edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本英文影印版由SIAM授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内（不包括中国香港、澳门特别行政区及中国台湾地区）销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可，不得以任何方式复制或发行本书的任何部分。

北京市版权局著作权合同登记号　图字：01-2008-0791

版权所有，翻印必究。举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

MATLAB中的谱方法 = Spectral Methods in MATLAB：英文/(美)特弗森(Trefethen, L.N.)著. -- 影印本.-- 北京：清华大学出版社，2011.2
(国际著名数学图书)
ISBN 978-7-302-24504-9

Ⅰ. ①M…　Ⅱ. ①特…　Ⅲ. ①谱分析(数学)-计算机辅助计算-软件包，MATLAB-英文
Ⅳ. ①O177.7-39

中国版本图书馆CIP数据核字(2011)第006667号

责任编辑：陈朝晖
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 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　销：全国新华书店
开　本：175×245　**印　张**：11.75
版　次：2011 年 2 月第 1 版　　**印　次**：2011 年 2 月第1 次印刷
印　数：1～3000
定　价：26.00 元

产品编号：027799-01

Preface

The aim of this book is to teach you the essentials of spectral collocation methods with the aid of 40 short MATLAB® programs, or "M-files."* The programs are available online at `http://www.comlab.ox.ac.uk/oucl/work/nick.trefethen`, and you will run them and modify them to solve all kinds of ordinary and partial differential equations (ODEs and PDEs) connected with problems in fluid mechanics, quantum mechanics, vibrations, linear and nonlinear waves, complex analysis, and other fields. Concerning prerequisites, it is assumed that the words just written have meaning for you, that you have some knowledge of numerical methods, and that you already know MATLAB.

If you like computing and numerical mathematics, you will enjoy working through this book, whether alone or in the classroom—and if you learn a few new tricks of MATLAB along the way, that's OK too!

Spectral methods are one of the "big three" technologies for the numerical solution of PDEs, which came into their own roughly in successive decades:

1950s: finite difference methods
1960s: finite element methods
1970s: spectral methods

Naturally, the origins of each technology can be traced further back. For spectral methods, some of the ideas are as old as interpolation and expan-

*MATLAB is a registered trademark of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA, tel. 508-647-7000, fax 508-647-7001, `info@mathworks.com`, `http://www.mathworks.com`.

sion, and more specifically algorithmic developments arrived with Lanczos as early as 1938 [Lan38, Lan56] and with Clenshaw, Elliott, Fox, and others in the 1960s [FoPa68]. Then, in the 1970s, a transformation of the field was initiated by work by Orszag and others on problems in fluid dynamics and meteorology, and spectral methods became famous. Three landmarks of the early modern spectral methods literature were the short book by Gottlieb and Orszag [GoOr77], the survey by Gottlieb, Hussaini, and Orszag [GHO84], and the monograph by Canuto, Hussaini, Quarteroni, and Zang [CHQZ88]. Other books have been contributed since then by Mercier [Mer89], Boyd [Boy00] (first edition in 1989), Funaro [Fun92], Bernardi and Maday [BeMa92], Fornberg [For96], and Karniadakis and Sherwin [KaSh99].

If one wants to solve an ODE or PDE to high accuracy on a simple domain, and if the data defining the problem are smooth, then spectral methods are usually the best tool. They can often achieve ten digits of accuracy where a finite difference or finite element method would get two or three. At lower accuracies, they demand less computer memory than the alternatives.

This short textbook presents some of the fundamental ideas and techniques of spectral methods. It is aimed at anyone who has finished a numerical analysis course and is familiar with the basics of applied ODEs and PDEs. You will see that a remarkable range of problems can be solved to high precision by a few lines of MATLAB in a few seconds of computer time. Play with the programs; make them your own! The exercises at the end of each chapter will help get you started.

I would like to highlight three mathematical topics presented here that, while known to experts, are not usually found in textbooks. The first, in Chapter 4, is the connection between the smoothness of a function and the rate of decay of its Fourier transform, which determines the size of the aliasing errors introduced by discretization; these connections explain how the accuracy of spectral methods depends on the smoothness of the functions being approximated. The second, in Chapter 5, is the analogy between roots of polynomials and electric point charges in the plane, which leads to an explanation in terms of potential theory of why grids for nonperiodic spectral methods need to be clustered at boundaries. The third, in Chapter 8, is the three-way link between Chebyshev series on $[-1,1]$, trigonometric series on $[-\pi,\pi]$, and Laurent series on the unit circle, which forms the basis of the technique of computing Chebyshev spectral derivatives via the fast Fourier transform. All three of these topics are beautiful mathematical subjects in their own right, well worth learning for any applied mathematician.

If you are determined to move immediately to applications without paying too much attention to the underlying mathematics, you may wish to turn directly to Chapter 6. Most of the applications appear in Chapters 7–14.

Inevitably, this book covers only a part of the subject of spectral methods. It emphasizes collocation ("pseudospectral") methods on periodic and on

此为试读，需要完整PDF请访问：www.ertongbook.com

Chebyshev grids, saying next to nothing about the equally important Galerkin methods and Legendre grids and polynomials. The theoretical analysis is very limited, and simple tools for simple geometries are emphasized rather than the "industrial strength" methods of spectral elements and *hp* finite elements. Some indications of omitted topics and other points of view are given in the Afterword.

A new era in scientific computing has been ushered in by the development of MATLAB. One can now present advanced numerical algorithms and solutions of nontrivial problems in complete detail with great brevity, covering more applied mathematics in a few pages than would have been imaginable a few years ago. By sacrificing sometimes (not always!) a certain factor in machine efficiency compared with lower level languages such as Fortran or C, one obtains with MATLAB a remarkable human efficiency—an ability to modify a program and try something new, then something new again, with unprecedented ease. This short book is offered as an encouragement to students, scientists, and engineers to become skilled at this new kind of computing.

Acknowledgments

I must begin by acknowledging two special colleagues who have taught me a great deal about spectral methods over the years. These are André Weideman, of the University of Stellenbosch, coauthor of the "MATLAB Differentiation Matrix Suite" [WeRe00], and Bengt Fornberg, of the University of Colorado, author of *A Practical Guide to Pseudospectral Methods* [For96]. These good friends share my enthusiasm for simplicity—and my enjoyment of the occasional detail that refuses to be simplified, no matter how hard you try. In this book, among many other contributions, Weideman significantly improved the crucial program `cheb`.

I must also thank Cleve Moler, the inventor of MATLAB, a friend and mentor since my graduate school days. Perhaps I had better admit here at the outset that there is a brass plaque on my office wall, given to me in 1998 by The MathWorks, Inc., which reads: *FIRST ORDER FOR MATLAB, February 7, 1985, Ordered by Professor Nick Trefethen, Massachusetts Institute of Technology*. I was there in the classroom at Stanford when Moler taught the numerical eigensystems course CS238b in the winter of 1979 based around this new experimental interface to EISPACK and LINPACK he had cooked up. I am a card-carrying MATLAB fan.

Toby Driscoll, author of the SC Toolbox for Schwarz–Christoffel conformal mapping in MATLAB [Dri96], has taught me many MATLAB tricks, and he helped to improve the codes in this book. He also provided key suggestions for the nonlinear waves calculations of Chapter 10. The other person whose suggestions improved the codes most significantly is Pascal Gahinet of The MathWorks, Inc., whose eye for MATLAB style is something special. David Carlisle of NAG, Ltd., one of the authors of LaTeX 2_ε, showed me how to make blank lines in MATLAB programs come out a little bit shortened when included as verbatim input, saving precious centimeters for display of figures. Walter Gautschi and Sotiris Notaris informed me about matters related to Clenshaw–Curtis quadrature, and Jean-Paul Berrut and Richard Baltensperger taught me about rounding errors in spectral differentiation matrices.

A number of other colleagues commented upon drafts of the book and improved it. I am especially grateful to John Boyd, Frédéric Dias, Des Higham, Nick Higham, Álvaro Meseguer, Paul Milewski, Damian Packer, and Satish Reddy.

In a category by himself goes Mark Embree, who has read this book more carefully than anyone else but me, by far. Embree suggested many improvements in the text, and beyond that, he worked many of the exercises, catching errors and contributing new exercises of his own. I am lucky to have found Embree at a stage of his career when he still has so much time to give to others.

The Numerical Analysis Group here at Oxford provides a stimulating environment to support a project like this. I want particularly to thank my three close colleagues Mike Giles, Endre Süli, and Andy Wathen, whose friendship has made me glad I came to Oxford; Shirley Dickson, who cheerfully made multiple copies of drafts of the text half a dozen times on short notice; and our outstanding group secretary and administrator, Shirley Day, who will forgive me, I hope, for all the mornings I spent working on this book when I should have been doing other things.

This book started out as a joint production with Andrew Spratley, a D. Phil. student, based on a masters-level course I taught in 1998 and 1999. I want to thank Spratley for writing the first draft of many of these pages and for major contributions to the book's layout and figures. Without his impetus, the book would not have been written.

Once we knew it would be written, there was no doubt who the publisher should be. It was a pleasure to publish my previous book [TrBa97] with SIAM, an organization that manages to combine the highest professional standards with a personal touch. And there was no doubt who the copy editor should be: again the remarkable Beth Gallagher, whose eagle eye and good sense have improved the presentation from beginning to end.

Finally, special thanks for their encouragement must go to my two favorite younger mathematicians, Emma (8) and Jacob (6) Trefethen, who know how I love differential equations, MATLAB, and writing. I'm the sort of writer who polishes successive drafts endlessly, and the children are used to seeing me sit down and cover a chapter with marks in red pen. Jacob likes to tease me and ask, "Did you find more mistakes in your book, Daddy?"

A Note on the MATLAB Programs

The MATLAB programs in this book are terse. I have tried to make each one compact enough to fit on a single page, and most often, on half a page. Of course, there is a message in this style, which is the message of this book: you can do an astonishing amount of serious computing in a few inches of computer code! And there is another message, too. The best discipline for making sure you understand something is to simplify it, simplify it relentlessly.

Without a doubt, readability is sometimes impaired by this obsession with compactness. For example, I have often combined two or three short MATLAB commands on a single program line. You may prefer a looser style, and that is fine. What's best for a printed book is not necessarily what's best for one's personal work.

Another idiosyncrasy of the programming style in this book is that the structure is flat: with the exception of the function `cheb`, defined in Chapter 6 and used repeatedly thereafter, I make almost no use of functions. (Three further functions, `chebfft`, `clencurt`, and `gauss`, are introduced in Chapters 8 and 12, but each is used just locally.) This style has the virtue of emphasizing how much can be achieved compactly, but as a general rule, MATLAB programmers should make regular use of functions.

Quite a bit might have been written to explain the details of each program, for there are tricks throughout this book that will be unfamiliar to some readers. To keep the discussion focused on spectral methods, I made a deliberate decision not to mention these MATLAB details except in a very few cases. This means that as you work with the book, you will have to study the programs, not just read them. What is this "`pol2cart`" command in Program 28.

(p. 120)? What's going on with the index variable "`b`" in Program 36 (p. 142)? You will only understand the answers to questions like these after you have spent time with the codes and adapted them to solve your own problems. I think this is part of the fun of using this book, and I hope you agree.

The programs listed in these pages were included as M-files directly into the LaTeX source file, so all should run correctly as shown. The outputs displayed are exactly those produced by running the programs on my machine. There was a decision involved here. Did we really want to clutter the text with endless formatting and Handle Graphics commands such as **fontsize**, **markersize**, **subplot**, and **pbaspect**, which have nothing to do with the mathematics? In the end I decided that yes, we did. I want you to be able to download these programs and get beautiful results immediately. Equally important, experience has shown me that the formatting and graphics details of MATLAB are areas of this language where many users are particularly grateful for some help.

My personal MATLAB setup is nonstandard in one way: I have a file `startup.m` that contains the lines

```
set(0,'defaultaxesfontsize',12,'defaultaxeslinewidth',.7,...
     'defaultlinelinewidth',.8,'defaultpatchlinewidth',.7).
```

This makes text appear by default slightly larger than it otherwise would, and lines slightly thicker. The latter is important in preparing attractive output for a publisher's high-resolution printer.

The programs in this book were prepared using MATLAB versions 5.3 and 6.0. As later versions are released in upcoming years, unfortunately, it is possible that some difficulties with the programs will appear. Updated codes with appropriate modifications will be made available online as necessary.

To learn MATLAB from scratch, or for an outstanding reference, I recommend SIAM's new *MATLAB Guide*, by Higham and Higham [HiHi00].

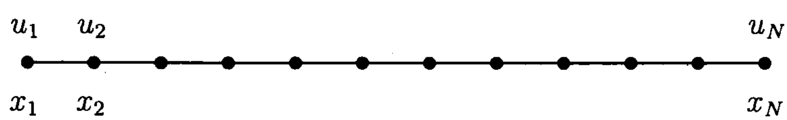
Contents

此为试读，需要完整PDF请访问：www.ertongbook.com

1. Differentiation Matrices

Our starting point is a basic question. Given a set of grid points $\{x_j\}$ and corresponding function values $\{u(x_j)\}$, how can we use this data to approximate the derivative of u? Probably the method that immediately springs to mind is some kind of finite difference formula. It is through finite differences that we shall motivate spectral methods.

To be specific, consider a uniform grid $\{x_1, \ldots, x_N\}$, with $x_{j+1} - x_j = h$ for each j, and a set of corresponding data values $\{u_1, \ldots, u_N\}$:



Let w_j denote the approximation to $u'(x_j)$, the derivative of u at x_j. The standard second-order finite difference approximation is

$$w_j = \frac{u_{j+1} - u_{j-1}}{2h}, \tag{1.1}$$

which can be derived by considering the Taylor expansions of $u(x_{j+1})$ and $u(x_{j-1})$. For simplicity, let us assume that the problem is periodic and take $u_0 = u_N$ and $u_1 = u_{N+1}$. Then we can represent the discrete differentiation process as a matrix-vector multiplication,

$$\begin{pmatrix} w_1 \\ \\ \vdots \\ \\ w_N \end{pmatrix} = h^{-1} \begin{pmatrix} 0 & \frac{1}{2} & & & & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \ddots & & & \\ & & \ddots & & & \\ & & & \ddots & 0 & \frac{1}{2} \\ \frac{1}{2} & & & & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ \\ \vdots \\ \\ u_N \end{pmatrix}. \tag{1.2}$$

(Omitted entries here and in other sparse matrices in this book are zero.) Observe that this matrix is *Toeplitz*, having constant entries along diagonals; i.e., a_{ij} depends only on $i - j$. It is also *circulant*, meaning that a_{ij} depends only on $(i - j) \pmod N$. The diagonals "wrap around" the matrix.

An alternative way to derive (1.1) and (1.2) is by the following process of local interpolation and differentiation:

For $j = 1, 2, \ldots, N$*:*

- *Let* p_j *be the unique polynomial of degree* ≤ 2 *with* $p_j(x_{j-1}) = u_{j-1}$, $p_j(x_j) = u_j$, *and* $p_j(x_{j+1}) = u_{j+1}$.
- *Set* $w_j = p_j'(x_j)$.

It is easily seen that, for fixed j, the interpolant p_j is given by

$$p_j(x) = u_{j-1}a_{-1}(x) + u_j a_0(x) + u_{j+1}a_1(x),$$

where $a_{-1}(x) = (x - x_j)(x - x_{j+1})/2h^2$, $a_0(x) = -(x - x_{j-1})(x - x_{j+1})/h^2$, and $a_1(x) = (x - x_{j-1})(x - x_j)/2h^2$. Differentiating and evaluating at $x = x_j$ then gives (1.1).

This derivation by local interpolation makes it clear how we can generalize to higher orders. Here is the fourth-order analogue:

For $j = 1, 2, \ldots, N$*:*

- *Let* p_j *be the unique polynomial of degree* ≤ 4 *with* $p_j(x_{j\pm2}) = u_{j\pm2}$, $p_j(x_{j\pm1}) = u_{j\pm1}$, *and* $p_j(x_j) = u_j$.
- *Set* $w_j = p_j'(x_j)$.

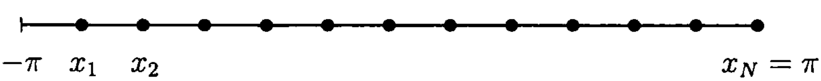
Again assuming periodicity of the data, it can be shown that this prescription amounts to the matrix-vector product

$$\begin{pmatrix} w_1 \\ \\ \\ \vdots \\ \\ \\ w_N \end{pmatrix} = h^{-1} \begin{pmatrix} & & \ddots & & & \frac{1}{12} & -\frac{2}{3} \\ & & \ddots & -\frac{1}{12} & & & \frac{1}{12} \\ & & \ddots & \frac{2}{3} & \ddots & & \\ & & \ddots & 0 & \ddots & & \\ & & \ddots & -\frac{2}{3} & \ddots & & \\ -\frac{1}{12} & & & \frac{1}{12} & \ddots & & \\ \frac{2}{3} & -\frac{1}{12} & & & \ddots & & \end{pmatrix} \begin{pmatrix} u_1 \\ \\ \\ \vdots \\ \\ \\ u_N \end{pmatrix}. \tag{1.3}$$

This time we have a pentadiagonal instead of tridiagonal circulant matrix.

The matrices of (1.2) and (1.3) are examples of *differentiation matrices.* They have order of accuracy 2 and 4, respectively. That is, for data u_j obtained by sampling a sufficiently smooth function u, the corresponding discrete approximations to $u'(x_j)$ will converge at the rates $O(h^2)$ and $O(h^4)$ as $h \to 0$, respectively. One can verify this by considering Taylor series.

Our first MATLAB program, Program 1, illustrates the behavior of (1.3). We take $u(x) = e^{\sin(x)}$ to give periodic data on the domain $[-\pi, \pi]$:



The program compares the finite difference approximation w_j with the exact derivative, $e^{\sin(x_j)}\cos(x_j)$, for various values of N. Because it makes use of MATLAB sparse matrices, this code runs in a fraction of a second on a workstation, even though it manipulates matrices of dimensions as large as 4096 [GMS92]. The results are presented in Output 1, which plots the maximum error on the grid against N. The fourth-order accuracy is apparent. This is our first and last example that does not illustrate a spectral method!

We have looked at second- and fourth-order finite differences, and it is clear that consideration of sixth-, eighth-, and higher order schemes will lead to circulant matrices of increasing bandwidth. The idea behind spectral methods is to take this process to the limit, at least in principle, and work with a differentiation formula of infinite order and infinite bandwidth—i.e., a dense matrix [For75]. In the next chapter we shall show that in this limit, for an

Program 1

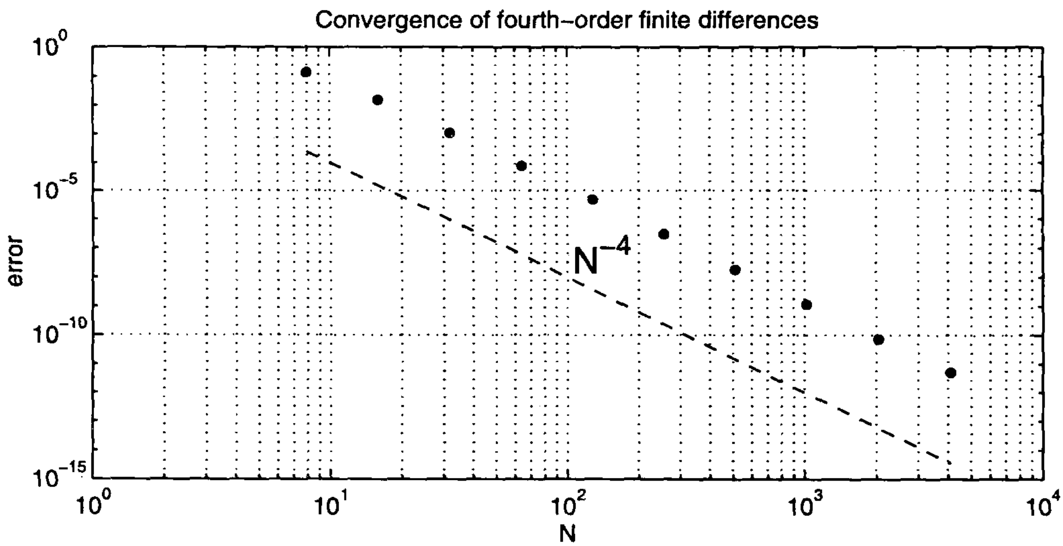
```
% p1.m - convergence of fourth-order finite differences

% For various N, set up grid in [-pi,pi] and function u(x):
  Nvec = 2.^(3:12);
  clf, subplot('position',[.1 .4 .8 .5])
  for N = Nvec
    h = 2*pi/N; x = -pi + (1:N)'*h;
    u = exp(sin(x)); uprime = cos(x).*u;

    % Construct sparse fourth-order differentiation matrix:
    e = ones(N,1);
    D =   sparse(1:N,[2:N 1],2*e/3,N,N)...
        - sparse(1:N,[3:N 1 2],e/12,N,N);
    D = (D-D')/h;

    % Plot max(abs(D*u-uprime)):
    error = norm(D*u-uprime,inf);
    loglog(N,error,'.','markersize',15), hold on
  end
  grid on, xlabel N, ylabel error
  title('Convergence of fourth-order finite differences')
  semilogy(Nvec,Nvec.^(-4),'--')
  text(105,5e-8,'N^{-4}','fontsize',18)
```

Output 1



Output 1: *Fourth-order convergence of the finite difference differentiation process (1.3). The use of sparse matrices permits high values of N.*

infinite equispaced grid, one obtains the following infinite matrix:

$$D = h^{-1} \begin{pmatrix} & \vdots & \\ \ddots & \frac{1}{3} & \\ \ddots & -\frac{1}{2} & \\ \ddots & 1 & \\ & 0 & \\ & -1 & \ddots \\ & \frac{1}{2} & \ddots \\ & -\frac{1}{3} & \ddots \\ & \vdots & \end{pmatrix} . \tag{1.4}$$

This is a skew-symmetric ($D^T = -D$) doubly infinite Toeplitz matrix, also known as a *Laurent operator* [Hal74, Wid65]. All its entries are nonzero except those on the main diagonal.

Of course, in practice one does not work with an infinite matrix. For a finite grid, here is the design principle for spectral collocation methods:

- *Let p be a single function (independent of j) such that $p(x_j) = u_j$ for all j.*
- *Set $w_j = p'(x_j)$.*

We are free to choose p to fit the problem at hand. For a periodic domain, the natural choice is a trigonometric polynomial on an equispaced grid, and the resulting "Fourier" methods will be our concern through Chapter 4 and intermittently in later chapters. For nonperiodic domains, algebraic polynomials on irregular grids are the right choice, and we will describe the "Chebyshev" methods of this type beginning in Chapters 5 and 6.

For finite N, taking N even for simplicity, here is the $N \times N$ dense matrix we will derive in Chapter 3 for a periodic, regular grid:

$$D_N = \begin{pmatrix} & \vdots & \\ \ddots & \frac{1}{2}\cot\frac{3h}{2} & \\ \ddots & -\frac{1}{2}\cot\frac{2h}{2} & \\ \ddots & \frac{1}{2}\cot\frac{1h}{2} & \\ & 0 & \\ & -\frac{1}{2}\cot\frac{1h}{2} & \ddots \\ & \frac{1}{2}\cot\frac{2h}{2} & \ddots \\ & -\frac{1}{2}\cot\frac{3h}{2} & \ddots \\ & \vdots & \end{pmatrix} . \tag{1.5}$$

Program 2

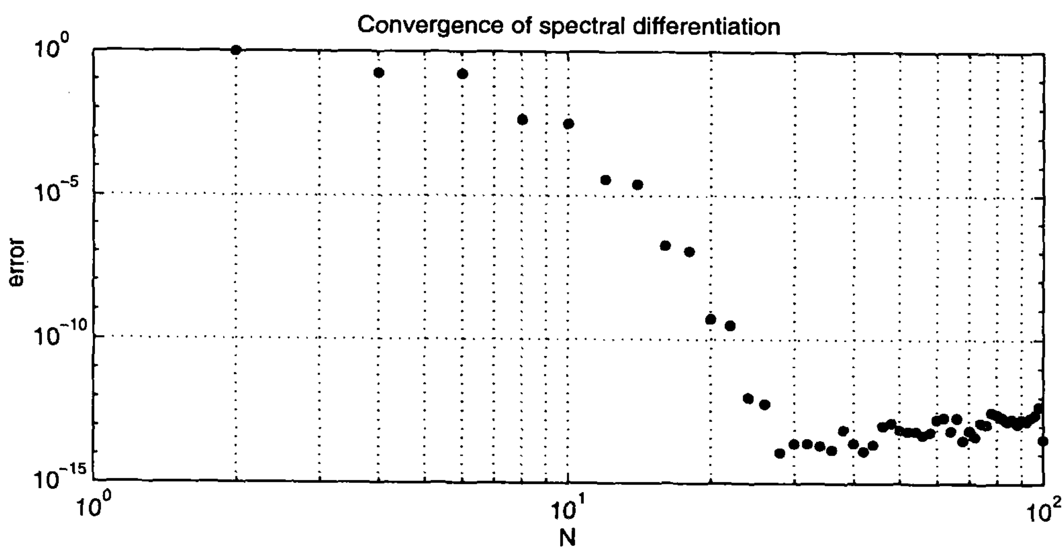
```
% p2.m - convergence of periodic spectral method (compare p1.m)

% For various N (even), set up grid as before:
  clf, subplot('position',[.1 .4 .8 .5])
  for N = 2:2:100;
    h = 2*pi/N;
    x = -pi + (1:N)'*h;
    u = exp(sin(x)); uprime = cos(x).*u;

    % Construct spectral differentiation matrix:
    column = [0 .5*(-1).^(1:N-1).*cot((1:N-1)*h/2)];
    D = toeplitz(column,column([1 N:-1:2]));

    % Plot max(abs(D*u-uprime)):
    error = norm(D*u-uprime,inf);
    loglog(N,error,'.','markersize',15), hold on
  end
  grid on, xlabel N, ylabel error
  title('Convergence of spectral differentiation')
```

Output 2



Output 2: *"Spectral accuracy" of the spectral method* (1.5), *until the rounding errors take over around* 10^{-14}. *Now the matrices are dense, but the values of* N *are much smaller than in Program* 1.

此为试读，需要完整PDF请访问：www.ertongbook.com