


The Symmetric Eigenvalue Problem



Beresford N. Parlett

C • L • A • S • S • I • C • S

In Applied Mathematics

20

NOTATION

Item	Description and Example
Definition	\equiv
Indices (positive integers)	i, j, k, l, m, n , and sometimes p
Displays (equations, theorems, etc.)	(j.k) k th display in Chapter j
Scalars (real or complex numbers)	Lowercase Greek α, β, ξ_1 , or lowercase italic a_{ij}, x_i
Vectors (column)	Lowercase sans-serif roman a, x, q_1 ; null vector \mathbf{o}
Matrices	Uppercase sans-serif roman A, B, Γ
Symmetric matrices	Symmetric letters $A, H, M, T, U, V, W, X, Y$
Diagonal matrices	Greek letters $\Lambda, \Theta, \Phi, \Delta$
Special matrices	Null matrix O , identity matrix I , shifted matrix $A - \sigma$ (omitting I)
Conjugate transpose	u^*, B^*
Vector spaces and subspaces	Script letters \mathcal{R}, \mathcal{S} , or span (b_1, \dots, b_m) , or x^\perp
Dimension	All vectors are n -dimensional and all matrices are n by n unless the contrary is stated
Eigenvalues	$\lambda_1, \lambda_2, \dots$ $\lambda_j[M]$ represents the j th eigenvalue of M $\lambda_{-n} \leq \dots \leq \lambda_{-2} \leq \lambda_{-1}$ (useful for big matrices)
Determinants	$\det B$, or $\det[B]$
Characteristic polynomials	$\chi(\tau) \equiv \chi_A(\tau) \equiv \det[\tau - A]$, χ is monic
Angles	$\angle(f, g)$ represents the angle between f and g
Orthogonality	$x \perp y, x \perp S$
Norms	$\ x\ \equiv \sqrt{x^*x}$ $\ B\ \equiv \max \ Bu\ /\ u\ $ over all $u \neq \mathbf{o}$ $\ B\ _F \equiv \sqrt{\sum_{i=1}^n [\sum_{j=1}^m b_{ij} ^2]}$ (F for Frobenius)
Direct sum	$B \oplus C = \text{diag}(B, C)$
More difficult material	* (star)



The Symmetric Eigenvalue Problem



SIAM's Classics in Applied Mathematics series consists of books that were previously allowed to go out of print. These books are republished by SIAM as a professional service because they continue to be important resources for mathematical scientists.

Editor-in-Chief

Robert E. O'Malley, Jr., *University of Washington*

Editorial Board

Richard A. Brualdi, *University of Wisconsin-Madison*

Herbert B. Keller, *California Institute of Technology*

Andrzej Z. Manitius, *George Mason University*

Ingram Olkin, *Stanford University*

Stanley Richardson, *University of Edinburgh*

Ferdinand Verhulst, *Mathematisch Instituut, University of Utrecht*

Classics in Applied Mathematics

C. C. Lin and L. A. Segel, *Mathematics Applied to Deterministic Problems in the Natural Sciences*

Johan G. F. Belinfante and Bernard Kolman, *A Survey of Lie Groups and Lie Algebras with Applications and Computational Methods*

James M. Ortega, *Numerical Analysis: A Second Course*

Anthony V. Fiacco and Garth P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*

F. H. Clarke, *Optimization and Nonsmooth Analysis*

George F. Carrier and Carl E. Pearson, *Ordinary Differential Equations*

Leo Breiman, *Probability*

R. Bellman and G. M. Wing, *An Introduction to Invariant Imbedding*

Abraham Berman and Robert J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*

Olvi L. Mangasarian, *Nonlinear Programming*

*Carl Friedrich Gauss, *Theory of the Combination of Observations Least Subject to Errors: Part One, Part Two, Supplement*. Translated by G. W. Stewart

Richard Bellman, *Introduction to Matrix Analysis*

U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*

K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*

Charles L. Lawson and Richard J. Hanson, *Solving Least Squares Problems*

J. E. Dennis, Jr. and Robert B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*

Richard E. Barlow and Frank Proschan, *Mathematical Theory of Reliability*

*First time in print.

Classics in Applied Mathematics (continued)

Cornelius Lanczos, *Linear Differential Operators*

Richard Bellman, *Introduction to Matrix Analysis, Second Edition*

Beresford N. Parlett, *The Symmetric Eigenvalue Problem*

Richard Haberman, *Mathematical Models: Mechanical Vibrations, Population Dynamics, and Traffic Flow*

Peter W. M. John, *Statistical Design and Analysis of Experiments*

Tamer Başar and Geert Jan Olsder, *Dynamic Noncooperative Game Theory, Second Edition*

Emanuel Parzen, *Stochastic Processes*

Petar Kokotović, Hassan K. Khalil, and John O'Reilly, *Singular Perturbation Methods in Control: Analysis and Design*

Jean Dickinson Gibbons, Ingram Olkin, and Milton Sobel, *Selecting and Ordering Populations: A New Statistical Methodology*

James A. Murdock, *Perturbations: Theory and Methods*

Ivar Ekeland and Roger Temam, *Convex Analysis and Variational Problems*

Ivar Stakgold, *Boundary Value Problems of Mathematical Physics, Volumes I and II*

J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*

David Kinderlehrer and Guido Stampacchia, *An Introduction to Variational Inequalities and Their Applications*

F. Natterer, *The Mathematics of Computerized Tomography*

Avinash C. Kak and Malcolm Slaney, *Principles of Computerized Tomographic Imaging*

R. Wong, *Asymptotic Approximations of Integrals*

O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*

David R. Brillinger, *Time Series: Data Analysis and Theory*

Joel N. Franklin, *Methods of Mathematical Economics: Linear and Nonlinear Programming, Fixed-Point Theorems*

Philip Hartman, *Ordinary Differential Equations, Second Edition*

Michael D. Intriligator, *Mathematical Optimization and Economic Theory*

Philippe G. Ciarlet, *The Finite Element Method for Elliptic Problems*

Jane K. Cullum and Ralph A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. I: Theory*

M. Vidyasagar, *Nonlinear Systems Analysis, Second Edition*

Robert Mattheij and Jaap Molenaar, *Ordinary Differential Equations in Theory and Practice*

Shanti S. Gupta and S. Panchapakesan, *Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations*

This page intentionally left blank



The Symmetric Eigenvalue Problem



Beresford N. Parlett

University of California
Berkeley, California

siam.

Society for Industrial and Applied Mathematics
Philadelphia

Copyright ©1998 by the Society for Industrial and Applied Mathematics.

This SIAM edition is an unabridged, corrected republication of the work first published by Prentice-Hall, Englewood Cliffs, NJ, 1980.

10 9 8 7 6 5 4 3 2 1

All rights reserved. Printed in the United States of America. No part of this book may be reproduced, stored, or transmitted in any manner without the written permission of the publisher. For information, write to the Society for Industrial and Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688.

Library of Congress Cataloging-in-Publication Data

Parlett, Beresford N.

The symmetric eigenvalue problem / Beresford N. Parlett.

p. cm. -- (Classics in applied mathematics ; 20)

"This SIAM edition is an unabridged, corrected republication of the work first published by Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1980"--T.p. verso.

Includes bibliographical references and index.

ISBN 0-89871-402-8 (pbk.)

1. Symmetric matrices. 2. Eigenvalues. I. Title. II. Series.

QA188.P37 1997

512.9'434--dc21

97-40623

To my parents,
Terèse and Norman

The greatest gift is the power to estimate correctly the value of things.
François de la Rochefoucauld (1613–1680)
Maxim number 244

This page intentionally left blank

Contents

Preface to the First Edition	xvii
Preface to the Classics Edition	xxi
Introduction	xxiii
CHAPTER 1. Basic Facts	
About Self-Adjoint Matrices	1
1.1. Introduction	1
1.2. Euclidean Space	1
1.3. Eigenpairs	5
1.4. Self-Adjoint Matrices	7
1.4.1. Invariant Subspaces	10
1.4.2. Hermitian Matrices	10
1.5. Quadratic Forms	11
1.6. Matrix Norms	15
1.7. The Generalized Eigenvalue Problem	18
CHAPTER 2. Tasks, Obstacles, and Aids	
2.1. What Is Small? What Is Large?	21
2.2. Tasks	22
2.3. Conflicting Requirements	24
2.3.1. Reliability	24
2.3.2. Accuracy	25
2.3.3. Swift Execution	26
2.3.4. Storage Requirements	26

*The more difficult material is indicated with a star both in the table of contents and in the text.

2.3.5. Short Program	27
2.4. Finite Precision Arithmetic	27
2.5. Cancellation	30
2.6. Inner Product Analysis	34
2.6.1. Numerical Example	34
2.6.2. The General Case	35
2.6.3. The Algorithm	35
2.6.4. Notation	35
2.6.5. Analysis	36
2.6.6. Matrix–Vector Products	37
2.6.7. Higher Precision Accumulation of Inner Products	37
2.7. Can Small Eigenvalues Be Found with Low Relative Error?	38
2.8. Software	39
2.8.1. Basic Linear Algebra Subprograms (BLAS)	40
2.8.2. Systems and Libraries	41
2.9. Alternative Computer Architecture	41
2.9.1. Vector Computers	41
2.9.2. Parallel Computers	42
CHAPTER 3. Counting Eigenvalues	43
3.1. Triangular Factorization	43
3.1.1. Remarks	44
3.1.2. Cost	45
3.2. Error Analysis of Triangular Factorization	48
3.3. Slicing the Spectrum	50
3.3.1. The Tridiagonal Case	52
3.3.2. Accuracy of the Slice	52
3.4. Relation to Sturm Sequences	54
3.5. Bisection and Secant Methods	56
3.5.1. Bisection	56
3.5.2. The Secant Method	56
3.6. Hidden Eigenvalues	58
3.7. The Characteristic Polynomial	60
CHAPTER 4. Simple Vector Iterations	61
4.1. Eigenvectors of Rank-One Matrices	61
4.2. Direct and Inverse Iteration	62
4.2.1. Convergence	63
4.2.2. Inverse Iteration (INVIT)	65
4.2.3. Shifts of Origin	66

4.2.4.	Cost	67
4.3.	Advantages of an Ill-Conditioned System	68
4.4.	Convergence and Orthogonality	72
4.5.	Simple Error Bounds	73
4.6.	The Rayleigh Quotient Iteration	75
4.7.	Local Convergence	76
4.8.	Monotonic Residuals	79
*4.9.	Global Convergence	80
CHAPTER 5. Deflation		87
5.1.	Deflation by Subtraction	87
5.2.	Deflation by Restriction	90
5.3.	Deflation by Similarity Transformation	91
CHAPTER 6. Useful Orthogonal Matrices		93
(Tools of the Trade)		
6.1.	Orthogonals Are Important	93
6.2.	Permutations	94
6.3.	Reflections and Direct Rotations	96
6.3.1.	Computing with Reflections	96
6.3.2.	The Direct Rotation	97
6.4.	Plane Rotations	99
6.4.1.	Jacobi Rotation	100
6.4.2.	Givens Rotation	100
6.4.3.	Operation Count	101
6.4.4.	Compact Storage	101
6.5.	Error Propagation in a Sequence of Orthogonal Congruences .	102
6.6.	Backward Error Analysis	105
6.7.	The QR Factorization and Gram–Schmidt	106
6.7.1.	Modified Gram–Schmidt (MGS)	107
6.7.2.	Householder’s Method	107
*6.8.	Fast Scaled Rotations	108
6.8.1.	Discarding Multiplications	109
6.8.2.	Avoiding Square Roots	109
6.8.3.	Error Analysis	111
6.8.4.	Accumulating the Product	112
*6.9.	Orthogonalization in the Face of Roundoff	113

CHAPTER 7. Tridiagonal Form	119
7.1. Introduction	119
7.2. Reduction Parameters	120
7.2.1. Remarks	122
7.3. Minimizing Characteristics	123
7.4. Explicit Reduction of a Full Matrix	125
7.4.1. Exploiting Symmetry	127
7.4.2. Plane Rotations	128
7.5. Reduction of a Banded Matrix	129
7.5.1. The Method	130
7.5.2. Operation Count	131
7.5.3. Storage	131
7.5.4. Reference	131
7.6. Irrelevant Instability	132
7.7. Eigenvalues Are Simple	134
7.8. Orthogonal Polynomials	134
7.9. Eigenvectors of T	137
7.10. Sturm Sequences	141
7.11. When to Neglect an Off-Diagonal Element	144
7.11.1. The Mathematical Problem	144
7.11.2. Algorithmic Problem 1	145
7.11.3. Algorithmic Problem 2	145
7.11.4. The Last Off-Diagonal Element	146
7.12. Inverse Eigenvalue Problems	146
CHAPTER 8. The QL and QR Algorithms	151
8.1. Introduction	151
8.2. The QL Transformation	152
8.2.1. The QL Algorithm	152
8.3. Preservation of Bandwidth	153
8.4. Relation between QL and QR	155
8.5. QL, the Power Method, and Inverse Iteration	156
8.6. Convergence of the Basic QL Algorithm	158
8.7. The Rayleigh Quotient Shift	159
8.8. The Off-Diagonal Elements	161
8.9. Residual Bounds Using Wilkinson's Shift	163
8.10. Tridiagonal QL Always Converges	165
8.11. Asymptotic Convergence Rates	168
8.12. Tridiagonal QL with Explicit Shift	171
8.13. Chasing the Bulge	173

8.13.1.	Comparison of Explicit and Implicit Shifts	175
8.14.	Shifts for all Seasons	176
8.14.1.	No Shifts	177
8.14.2.	Rayleigh Quotient Shift	177
8.14.3.	Newton's Shift	177
8.14.4.	Saad's Shifts	177
8.14.5.	The Ultimate Shifts	178
*8.15.	Casting Out Square Roots	178
8.15.1.	Derivation of the Algorithm	179
8.15.2.	Ortega-Kaiser Version	181
8.15.3.	Reinsch's Algorithm	182
8.15.4.	Slicing the Spectrum	182
8.15.5.	Stability	182
8.16.	QL for Banded Matrices	185
8.16.1.	Origin Shift	186
8.16.2.	Operation Count	187
8.16.3.	Storage	187
CHAPTER 9. Jacobi Methods		189
9.1.	Rotation in the Plane	189
9.2.	Jacobi Rotations	190
9.2.1.	Rutishauser's Modifications	192
9.3.	Convergence	193
9.4.	Strategies	195
9.4.1.	Classical Jacobi	195
9.4.2.	The Cyclic Jacobi Methods	195
9.4.3.	Threshold Methods	196
9.5.	Ultimate Quadratic Convergence	197
9.5.1.	Multiple Eigenvalues	198
9.6.	Assessment of Jacobi Methods	199
CHAPTER 10. Eigenvalue Bounds		201
10.1.	Cauchy's Interlace Theorem	202
10.2.	Minmax and Maxmin Characterization	206
10.3.	The Monotonicity Theorems	207
10.4.	The Residual Interlace Theorem	211
*10.5.	Lehmann's Optimal Intervals	216
*10.6.	The Use of Bounds on the Missing Submatrix	221
*10.7.	The Use of Gaps in A's Spectrum	225

CHAPTER 11. Approximations from a Subspace	229
11.1. Subspaces and Their Representation	229
11.2. Invariant Subspaces	232
11.3. The Rayleigh–Ritz Procedure	234
11.4. Optimality	234
11.5. Residual Bounds on Clustered Ritz Values	239
11.6. No Residual Bounds on Ritz Vectors	242
11.7. Gaps in the Spectrum	244
11.7.1. Gap Theorems for Subspaces	247
11.8. Condensing the Residual	249
*11.9. A Priori Bounds for Interior Ritz Approximations	250
*11.10. Nonorthogonal Bases	253
CHAPTER 12. Krylov Subspaces	261
12.1. Introduction	261
12.2. Basic Properties	263
12.2.1. A Theoretical Limitation	263
12.2.2. Invariance Properties	264
12.3. Representation by Polynomials	265
*12.4. The Error Bounds of Kaniel and Saad	269
*12.5. Better Bounds	275
12.6. Comparison with the Power Method	279
12.7. Partial Reduction to Tridiagonal Form	282
CHAPTER 13. Lanczos Algorithms	287
13.1. Krylov + Rayleigh–Ritz = Lanczos	287
13.1.1. Simple Lanczos	288
13.2. Assessing Accuracy	290
13.3. The Effects of Finite Precision Arithmetic	293
13.4. Paige’s Theorem	295
13.5. An Alternative Formula for β_j	299
13.6. Convergence \implies Loss of Orthogonality	300
13.7. Maintaining Orthogonality	303
*13.8. Selective Orthogonalization (SO)	305
13.8.1. LanSO Flowchart (Lanczos Algorithm with SO)	308
*13.9. Analysis of SO	310
13.9.1. Orthonormalizing the Good Ritz Vectors	311
13.9.2. The Effect of Purging on Angles	311
13.9.3. The Governing Formula	313
13.10. Band (or Block) Lanczos	316

13.10.1. Block Lanczos	317
13.10.2. Band Lanczos	318
13.11. Partial Reorthogonalization (PRO)	319
13.12. Block Versus Simple Lanczos	320
CHAPTER 14. Subspace Iteration	323
14.1. Introduction	323
14.2. Implementations	324
14.3. Improvements	329
14.3.1. Chebyshev Acceleration	329
14.3.2. Randomization	330
14.3.3. Termination Criteria	331
*14.4. Convergence	331
14.5. Sectioning	336
CHAPTER 15. The General Linear Eigenvalue Problem	339
15.1. Introduction	339
15.2. Symmetry Is Not Enough	340
15.3. Simultaneous Diagonalization of Two Quadratic Forms	343
15.4. Explicit Reduction to Standard Form	346
15.4.1. Remarks	347
15.4.2. Reduction of Banded Pencils	348
*15.5. The Fix-Heiberger Reduction	349
15.6. The QZ Algorithm	353
15.7. Jacobi Generalized	353
15.8. Implicit Reduction to Standard Form	354
15.9. Simple Vector Iterations	356
15.9.1. Other Iterative Techniques	359
15.10. RR Approximations	360
15.11. Lanczos Algorithms	362
15.11.1. Selective Lanczos Algorithm for (A, M)	363
15.11.2. Remarks	363
15.11.3. How to Use a Lanczos Program	364
15.12. Subspace Iteration	364
15.12.1. Comments on Table 15.1	365
15.13. Practical Considerations	367
Appendix A. Rank-One and Elementary Matrices	369
Appendix B. Chebyshev Polynomials	371

Annotated Bibliography	375
References	379
Index	393

Preface to the First Edition

Whenas in silks my Julia goes
Then, then methinks how sweetly flows
The liquefaction of her clothes.
Next when I cast mine eyes and see
That brave vibration, each way free,
Oh, how that glittering taketh me.

Robert Herrick (1591-1674)

The fact of harmony between Heaven and Earth
and Man does not come
from physical union, from a direct action,
it comes from a tuning on the same note producing
vibrations in unison.

Tong Tshung-chu (second century B.C.)

$$\Delta\psi + \frac{8\pi^2m}{\hbar^2}(E - V)\psi = 0.$$

E. Schrödinger (1925)

Vibrations are everywhere, as the quotations suggest, and so too are the eigenvalues (or frequencies) associated with them. The concert-goer unconsciously analyzes the quivering of her eardrum, the spectroscopist identifies the constituents of a gas by looking at eigenvalues, and in California the State Building Department requires that the natural frequencies of many buildings

should lie outside the earthquake band. Indeed, as mathematical models invade more and more disciplines we can anticipate a demand for eigenvalue calculations in an ever richer variety of contexts. The reader who is not sure what an eigenvalue is should turn to Chapter 1 or some other work.

The interesting differences between various eigenvalue problems disappear when the formulation is made sufficiently abstract and the practical problems reduce to the single task of computing the eigenvalues of a square matrix with real or complex entries. Nevertheless there is one distinction worth maintaining: some matrices have real eigenvalues and others do not. The former usually come from so-called self-adjoint problems which are much nicer than the others. Apart from a handful of numerical analysts there appear to be few people interested in computations for both self-adjoint and non-self-adjoint problems. There are advantages in treating the two cases separately, and this book is devoted to eigenvalue computations for real symmetric matrices. This may be the easier case but expectations are correspondingly higher.

For the newcomer to eigenvalue calculations I would like to give a brief summary of the situation. (A more detailed discussion of the contents of the book follows in the introduction.) Matrices are either small or large, as described in Chapter 2. For the small ones, good programs are now available in most scientific centers for virtually all the requests that users are likely to make. Furthermore, the understanding of the methods by the programs is essentially complete. By dint of being worked and reworked the theory has become simple to the point of elegance. The story is told in Chapters 6 through 9.

Attention has now turned to large matrices. The tasks are harder; some good methods have been developed but the subject is far from tidy. There is yet no general consensus on the right techniques for each task and there are financial incentives, in addition to the intellectual ones, for making more progress. In 1978 I was told that the computation of 30 eigenvalue/eigenvector pairs of a certain matrix of order 12,000 required \$12,000 of computer time (excluding the cost of program development). The last five chapters present the tools which any large scale eigenvalue prospector should have at hand.

Any author would like to be both brief and intelligible; of these two virtues, the latter depends the more strongly on the background and fortitude of the reader. The level of difficulty of this book goes up and down according to the topic but please, gentle reader, do not expect to go through proofs as you would read a novel. The transition from one line to the next sometimes requires the proper marshaling of facts presented earlier and only by engaging in that irksome activity can the material be made your own. The exercises at the end of each section are to reinforce this exhortation.

I hope there is something of interest in each chapter, even for the expert. Classical topics have been reworked and a fair proportion of the results, though not always new, are either little known or else unavailable. One strong incentive for writing this book was the realization that my friend and colleague W. Kahan would never narrow down his interests sufficiently to publish his own very useful insights.

The most difficult part of the enterprise was trying to impose some sort of order on the mass of worthwhile information. After many rearrangements the final list of contents seemed to emerge of its own accord. The book is intended to be a place of reference for the most important material as of 1978. As such it aims to be a sequel to Chapter 5 of J. H. Wilkinson's 1965 masterpiece *The Algebraic Eigenvalue Problem*.

Selections from the material before Chapter 10 are in use as part of a senior level course at Berkeley and some of the later material was covered in a graduate seminar. There is a noticeable increase in difficulty after Chapter 9. Despite the mathematical setting of the discussions they are actually intended to enlighten anyone with a need to compute eigenvalues. Nothing would please me more than to learn that some consumers of methods, who make no claim to be numerical analysts, have read parts of the book with profit, or even a little pleasure.

At this point I would like to acknowledge some of my debts. My thesis advisor, the late George Forsythe, made numerical analysis seem both useful and interesting to all his students. Jim Wilkinson led many numerical analysts out of the wasteland of unenlightening analyses and showed us how to think right about matrix computations. Vel Kahan has been a patient and helpful guide for my study of symmetric matrices and many other topics as well. The persevering reader will learn the magnitude of his contribution, but those who know him will appreciate that with any closer cooperation this book would never have seen the light of day. Naturally I have learned a lot from regular contacts with Gene Golub, Cleve Moler, Chris Paige, and Pete (G. W.) Stewart, and I have been helped by my present and former students. The manuscript was read by Joel Franklin, Gene Isaacson, Cleve Moler, and Gil Strang. I am grateful for the blemishes they caught and the encouragement they gave.

Next I wish to thank Dick Lau and the late Leila Bram, of the Office of Naval Research, whose generous support enabled me to write the book. Ruth (bionic fingers) Suzuki once again lived up to her exalted reputation for technical typing.

Beresford N. Parlett

This page intentionally left blank

Preface to the Classics Edition

It is trite but true to say that research on the symmetric eigenvalue problem has flourished since the first edition of this book appeared in 1980. I had dreamed of including the significant new material in an expanded second edition, but my own research obsessions diverted me from reading, digesting, and then regurgitating all that work. So it was with some relief that I accepted Gene Golub's suggestion that I let SIAM republish the original book in its Classics in Applied Mathematics series.

Naturally I could not resist the temptation to remove blemishes in the original, to sharpen some results, and to include some slicker proofs. In the first of these activities I was greatly helped by three people who read all, or nearly all, of the first edition with close attention. They are Thomas Ericsson, Zhaojun Bai, and Richard O. Hill, Jr.

In the first edition I tried to organize all the information that I deemed to be necessary intellectual equipment for eigenvalue hunters. Although that knowledge is not dated, it may not be adequate for the second millennium.

As to the progress that has been made since 1980 I have limited myself to a few pointers at the end of some chapters. This is the treatment meted out to parallel algorithms, divide and conquer techniques, the implementation of Lanczos algorithms, and the revival of qd-like algorithms for tridiagonals. Perhaps each of these topics warrants a monograph to itself?

I like to think that Chapter 8 still provides the simplest and fullest treatment of the QL and QR algorithms for symmetric tridiagonals, and little has been added. In Chapter 10 the proofs of the classic results of Cauchy, Courant–Fischer, and Weyl are new. They are all consequences of the elementary fact that in n dimensions, $n < \infty$, any two subspaces the sum of whose dimensions exceeds n must have a nontrivial intersection! Why were those theorems always made to look so complicated?

Recently R. O. Hill, Jr. and I managed to remove an unverifiable hypoth-

esis from Kahan's residual interlace theorem (Theorem 10.4.1) and, naturally, that new proof had to be included. In Chapter 11 the factor $\sqrt{2}$ has been replaced by the best possible value 1 in Kahan's bound on clustered eigenvalues using residual norms derived from nonorthogonal bases (Theorem 11.10.1). Chapter 13, on Lanczos algorithms, now ends with a few remarks to put the modifications called selective orthogonalization (SO) and partial reorthogonalization (PRO) in their proper setting as ways to maintain semiorthogonality among computed Lanczos vectors.

Those who are familiar with the first edition will find that the new edition is essentially a spruced up version of the old one.

Beresford N. Parlett
Berkeley, California

Introduction

Principal notations are indicated inside the front cover.

At many places in the book, reference is made to more or less well known facts from matrix theory. For completeness these results had to be present, but for brevity I have omitted proofs and elementary definitions. These omissions may help the reader to see the subject as a whole, an outcome that does not automatically follow a course in linear algebra. Not all the facts in Chapter 1 are elementary.

Chapter 2 is what the reader should know about the computer's influence on the eigenvalue problem. To my mind it is the need to harmonize conflicting requirements that makes the concoction of algorithms a fascinating task. At the other extreme my heart always sinks when the subject of roundoff error is mentioned. The sometimes bizarre effects of finite precision arithmetic should be at the heart of work in matrix computations. Yet formal treatment of roundoff, though it seems to be necessary, rarely enlightens me. Fortunately there is Wilkinson's excellent little book, *Rounding Errors in Algebraic Processes*, which supplies a thorough and readable treatment of the fundamentals. Thus I felt free to concentrate on those topics which I consider essential, such as the much maligned phenomenon of cancellation. In addition I point to some of the excellent programs that are now available.

The book gets underway in Chapter 3 which centers on the remarkable fact that the number of negative pivots in the standard Gaussian elimination process applied to $A - \xi I$ equals the number of A 's eigenvalues less than ξ .

The power method and inverse iteration are very simple processes, but the ideas behind them are quite far reaching. It is customary to analyze the methods by means of eigenvector expansions, but a little plane trigonometry yields results which are sharper and simpler than the usual formulations. In the same vein I have tried to find the most elegant proofs of the well-known error bounds which also seem to belong to Chapter 4.

The Rayleigh quotient iteration is perhaps the best-known version of inverse iteration with variable shifts. Its rapid convergence, when it does converge to an eigenvalue/eigenvector pair, is well known. Kahan's discovery that the iteration does in fact converge from almost all starting vectors rounds out the theory very nicely but is little known. The proof (section 4.9) is relatively difficult and should be omitted from an undergraduate course.

The important point about "deflating" a known eigenpair from a matrix is that the safety and desirability of the operation depend on how it is done. It seemed useful to put the various techniques side by side for easy comparison.

Chapter 6 prepares the way for discussing explicit similarity transformations, but in fact orthogonal matrices play an important role in many other matrix calculations.

It does not seem to be appreciated very widely that any orthogonal matrix can be written as the product of reflector matrices. Thus the class of reflections is rich enough for all occasions and yet each member is characterized by a single vector which serves to describe its mirror. Pride of place must go to these tools despite the fact that they were generally ignored until Householder advocated their use in 1958.

Plane rotations made a comeback into favor in the late 1960s because, when proper scaling is used, they are no less efficient than reflections and are particularly suited for use with sparse matrices. In addition to the material mentioned above, Chapter 6 presents Wilkinson's simple demonstration of the very small effect that roundoff has on a sequence of explicit orthogonal similarity transformations. Last, the QR factorization is introduced as simply a matrix description of the Gram-Schmidt orthonormalizing process. The actual way that the factors Q and R should be computed is a different question, and the answer depends on the circumstances. The effect of roundoff on orthogonalization is treated in the final section.

Tridiagonal matrices have been singled out for special consideration ever since 1954 when Wallace Givens suggested reducing small full matrices to this form as an intermediate stage in computing the eigenvalues of the original matrix. Chapter 7 begins with the useful theorem which says to what extent the tridiagonal form is determined by the original matrix. Next comes a neglected but valuable characterization of the off-diagonal elements in the tridiagonal form. After this preparation the techniques of reduction are developed together with the analysis, due to Wilkinson, that an apparent instability in the reduction is not to be feared.

For later use I found it necessary to include one section which presents the interesting relationships governing elements in the eigenvectors of tridiagonal matrices, another section on the difficult question of when an off-diagonal ele-

ment is small enough to be neglected, and a brief account of how to recover the tridiagonal matrix from its eigenvalues together with some extra information. These later sections are not standard material.

The champion technique for diagonalizing small tridiagonal matrices is the QR algorithm (the latest version is called the QL algorithm, just to add to the confusion). It is rather satisfying to the numerical analyst that the best method has turned out not to be some obvious technique like the power method, or the characteristic polynomial, or some form of Newton's method, but a sequence of sophisticated similarity transformations. Understanding of the algorithm has increased gradually, and we now have an elegant explanation of why it always works. The theory is presented first, followed by the equally interesting schemes for implementing the process. For the sake of efficiency, the QL algorithm is usually applied to matrices of narrow bandwidth.

Over a hundred years older than QR is the Jacobi method for diagonalizing a symmetric matrix. The idea behind the method is simple, and Chapter 9 covers the salient features. I must confess that I cannot see the niche where Jacobi methods will fit in the computing scene of the future. These doubts are made explicit in the final section.

With Chapter 10 the focus of the book shifts to large problems, although the chapter itself is pure matrix theory. It brings together results, old and new, which estimate eigenvalues of a given matrix in terms of eigenvalues of submatrices or low-rank modifications. Until now these results have been scattered throughout the literature. Here they are brought together and are given in complete detail. This makes the chapter highly nutritious but rather indigestible at first encounter. (It came from notes used by Kahan in various lectures.)

Chapter 11 is a rather thorough description of a useful tool called the Rayleigh–Ritz procedure, but it is confined to matrices instead of differential operators. Section 11.4 should make clear in what senses the Ritz approximations are optimal and in what senses they are not. I often got confused on this point before I came to write the book.

The next chapter introduces a special sequence of subspaces, called Krylov spaces for no sufficient reason, and applies the Rayleigh–Ritz procedure to them. I believe that the attractive approximation theory in this chapter will be new to most readers. It establishes the great potential of the Lanczos algorithm for finding eigenvalues.

Chapter 13 is concerned with the widespread fear that roundoff will prevent us from reaping the harvest promised by Chapter 12. In his pioneering 1971 thesis, Paige showed that the fear is not warranted. In giving a proof of the main theorem I have tried to strip away the minutiae which encrusted

Paige's penetrating analysis of the simple Lanczos algorithm. Next comes selective orthogonalization, a modification guided by Paige's theory, which subdues quite economically the much feared loss of orthogonality among the Lanczos vectors. The method was announced in 1978 and is still ripe for further development. The chapter ends with a discussion of block Lanczos schemes which are likely to be needed for coping with the biggest matrices.

Chapter 14 treats the highly polished and effective methods based on block inverse iteration. These were refined by physicists, engineers, and numerical analysts from the 1960s onward while the Lanczos algorithm lay under a cloud. I doubt that they will maintain their superiority.

The last chapter turns to the generalized problem and begins with some basic properties which are not universally appreciated. Indeed, I have seen books which get them wrong. The next topic, a key issue, is whether the given problem should be reduced to the standard form either explicitly or implicitly or not at all. The rest of the chapter, though long, is simply concerned with how the ideas and methods in the rest of the book can be extended to the general problem.

Both appendices have proved useful to me; the first describes elementary matrices, and the second lists the most useful properties of Chebyshev polynomials.

From time to time there occur applications in which special conditions permit the use of economical techniques which are, in general, unstable or inapplicable. The power to recognize these situations is one of the benefits of understanding a whole corpus of numerical methods.

For large matrices A the choice of method depends strongly on how many of the following operations are feasible:

1. multiply any conformable vector x by A ,
2. split A into $H + V$ where V is small and the triangular factors of H can be computed readily,
3. triangular factorization of A ,
4. triangular factorization of $A - \sigma I$ for any values of σ .

The more that can be done with A , the more powerful the methods which can be invoked to compute its eigenvalues. Early drafts of this book presented the methods according to these operations and, although that arrangement has been abandoned, this scheme provides a useful structure for thinking about possible algorithms.

Sections 15.9.1 and 15.13 extend this discussion to the generalized eigenvalue problem.

Basic Facts About Self-Adjoint Matrices

1.1. Introduction

This chapter offers a quick tour of those parts of linear algebra which are relevant to a discussion of numerical methods for approximating eigenvalues and eigenvectors of symmetric matrices. The notational conventions used throughout the book are introduced along the way, but they are also collected inside the cover for those readers who want to skip ahead. The list of facts may be of use to readers who wish to gauge their mastery of linear algebra by actually proving the assertions made on the tour. Those who would like a more leisurely trip should consult the annotated bibliography for a suitable text.

Before beginning, it is worth recalling that notation assists us by hiding information. Any particular notation is successful insofar as it displays only what is necessary. Matrix notation suppresses reference to individual elements of vectors and matrices as much as possible. The result is confusion before this language is mastered and satisfaction thereafter.

1.2. Euclidean Space

This book is not concerned with the abstract notion of a vector space as any set of objects which is closed under an operation that satisfies certain axioms; instead the discussion goes straight to the coordinate representation and takes \mathbb{R}^n as the vector space of all n -dimensional column vectors with real components.

However, linear algebra embraces complex vectors as easily as it does real vectors, and for this chapter, and this chapter only, each number (or scalar¹) is complex unless the contrary is stated. Throughout the book scalars are usually

¹A quantity having magnitude but no direction (*Concise Oxford Dictionary*).

denoted by lowercase Greek letters (α, β) and vectors by lowercase sans-serif letters (x, q_1). The space of all complex n -dimensional vectors is denoted by \mathbb{C}^n .

To aid the imagination, a vector x in \mathbb{R}^2 can be thought of as a line segment, or arrow, directed from the chosen origin o to a point in the plane with coordinates $x = [\xi_1 \ \xi_2]$. (See Figure 1.1.)

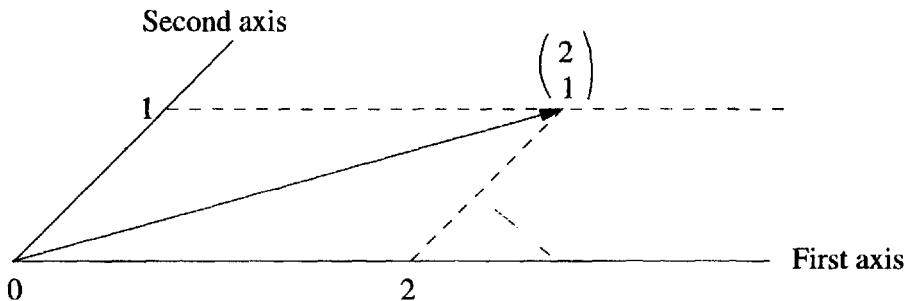


FIG. 1.1. *Axes.*

The proper setting for problems involving real symmetric matrices is not just \mathbb{R}^n but n -dimensional Euclidean space, called \mathcal{E}^n , which is \mathbb{R}^n embellished with some extra structure. What distinguishes \mathcal{E}^n from \mathbb{R}^n or \mathbb{C}^n is the idea that any pair of vectors x and y make a certain *angle*. More precisely, the extra structure enjoyed by \mathcal{E}^n is an *inner product function* which assigns to each pair x (with components ξ_1, \dots, ξ_n) and y (with components η_1, \dots, η_n) a number written as (x, y) and defined by

$$(x, y) \equiv \sum_{i=1}^n \bar{\eta}_i \xi_i. \quad (1.1)$$

Throughout the book \equiv denotes a definition and $\bar{\alpha}$ denotes the complex conjugate of α . Other more complicated definitions of the inner product function are appropriate in certain applications, such as the analysis of the stability of buildings, and are discussed later. However (1.1) is the simplest and distinguishes Euclidean space from other inner product spaces. In some contexts, when the components of a vector refer to dissimilar quantities such as pressure, velocity, and density, there may be no appropriate inner product and \mathbb{R}^n is the proper setting.

Apart from the next section the whole book is focused on real Euclidean space \mathcal{E}^n . Strictly speaking \mathcal{E}^2 is not the set of points in the plane (or arrow tips) because it makes no sense to speak of the angle between two points.

Pictorially \mathcal{E}^2 is the set of all arrows emanating from some chosen origin o . The axes, to which all the other vectors are referred, are taken *perpendicular* (or *orthogonal*) to each other. The powerful notions of orthogonality, angle, and length are defined in terms of (1.1).

The Euclidean *length*, or *norm*, of x is defined by

$$\|x\| \equiv \sqrt{(x, x)}. \quad (1.2)$$

There are many other norms which can be imposed on \mathbb{C}^n or \mathbb{R}^n , but (1.2) belongs to \mathcal{E}^n . The famous *Cauchy–Schwarz inequality*, namely,

$$|(x, y)| \leq \|x\| \cdot \|y\| \text{ for all } x, y \text{ in } \mathcal{E}^n, \quad (1.3)$$

justifies the definition of the angle given in (1.4) below. The angle in radians between x and y , written $\angle(x, y)$, is the real number θ satisfying $0 \leq \theta \leq \pi$ and

$$\cos \theta = \frac{\operatorname{Re}(x, y)}{\|x\| \cdot \|y\|}. \quad (1.4)$$

Usually it is not just x which is of interest but the line determined by x , namely, the collection of all scalar multiples of x , including $-x$. We call this $\operatorname{Span}(x)$. The (acute) angle ϕ between the line on x and the line on y is the real number ϕ satisfying $0 \leq \phi \leq \pi/2$ and

$$\cos \phi = \frac{|(x, y)|}{\|x\| \cdot \|y\|}. \quad (1.5)$$

Figure 1.2 illustrates the difference between θ and ϕ .

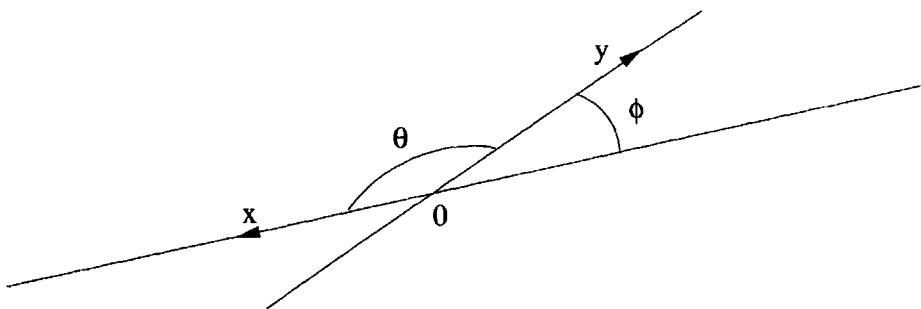


FIG. 1.2. Angles between rays and lines in real \mathcal{E}^2 .

In particular, two vectors x and y are orthogonal if $(x, y) = 0$. A vector x is normalized, or a unit vector, if $\|x\| = 1$.

THE SYMMETRIC EIGENVALUE PROBLEM

Matrices, always denoted by capital letters, are usually engaged in transforming vectors. If F is m by n and w lies in \mathbb{C}^n then the product Fw lies in \mathbb{C}^m . It is useful to imagine F multiplying all the vectors in \mathbb{C}^n , and in this way it transforms \mathbb{C}^n into \mathbb{C}^m . The transformation is *linear*, i.e., $F(\alpha x + \beta y) = \alpha Fx + \beta Fy$, and, in fact, any linear transformation of \mathbb{C}^n into \mathbb{C}^m can be represented by some m -by- n matrix F . This important property of matrix multiplication should have been called *additivity* rather than *linearity*, but it is too late to change now.

When F is square and invertible it can be regarded as simply effecting a change of basis in \mathbb{C}^n instead of a transformation of vectors in \mathbb{C}^n . Thus Fw is either the image of w under F or, equally well, the new coordinates for the same old vector that used to have coordinates w . This flexibility is helpful to those who are familiar with it but confusing to the beginner. The notation y^* for the row vector $(\bar{y}_1, \dots, \bar{y}_n)$ is widespread, yet fundamentally the conjugate transpose operation $*$ is yet another consequence of the Euclidean inner product structure. Let F denote an m -by- n matrix. Its *adjoint* matrix F^* is n by m and is defined abstractly by the property that, for all x in \mathcal{E}^n and u in \mathcal{E}^m , F^*u is the only vector satisfying

$$(x, F^*u) = (Fx, u). \quad (1.6)$$

Note that (\cdot, \cdot) denotes the dot product in \mathcal{E}^n on the left of (1.6) and in \mathcal{E}^m on the right side. It can then be shown that F^* is the familiar *conjugate transpose* of F (Exercise 1.2.3). For example, if $i^2 = -1$, α and β are real, then

$$\begin{bmatrix} i & \alpha + i\beta \\ 0 & 1 \end{bmatrix}^* = \begin{bmatrix} -i & 0 \\ \alpha - i\beta & 1 \end{bmatrix}.$$

Whenever the product FG is defined then

$$(FG)^* = G^*F^*. \quad (1.7)$$

In particular $(Fw)^* = w^*F^*$. Even when F is real the same symbol F^* , rather than F^H or F^T or F^t , can and will be used for the *transpose* of F . Furthermore, when F is square and invertible then

$$(F^*)^{-1} = (F^{-1})^*, \quad (1.8)$$

and each will be written simply as F^{-*} .

An m -by- n matrix P , with $m \geq n$, is *orthonormal* if its columns are orthonormal, that is, if

$$P^*P = I_n \quad (= I \text{ for brevity}). \quad (1.9)$$

Square orthonormal matrices are called *unitary* (when complex) and *orthogonal* (when real). It is tempting to replace these two less-than-apt adjectives by the single natural one, *orthonormal*. The great importance of these matrices is that inner product formula (1.1) is preserved under their action, i.e.,

$$(Px, Py) = y^* P^* Px = y^* x = (x, y). \quad (1.10)$$

When an orthogonal change of basis occurs in \mathcal{E}^n then the coordinates of all vectors are multiplied by some orthonormal P but, mercifully, the values of norms and angles stay the same. In fact, the set of orthogonal transformations, together with simple translations, are the familiar rigid body motions of Euclidean geometry.

Exercises on Section 1.2

- 1.2.1. Derive the Cauchy–Schwarz inequality by noting that $(x + e^{i\phi}\mu y)^*(x + e^{i\phi}\mu y) = \|x + e^{i\phi}\mu y\|^2 \geq 0$ and making the right choice for ϕ and μ .
- 1.2.2. Given complex x and y is it possible to choose a real θ such that $\exp(i\theta)x$ and y are orthogonal? Consider (1.2).
- 1.2.3. By using coordinate vectors e_i for x and y show that (1.6) implies $(F^*)_{ij} = (\bar{F})_{ji}$.
- 1.2.4. Derive (1.7) from (1.6).
- 1.2.5. Prove that (1.10) holds if and only if (1.9) holds.

1.3. Eigenpairs

The notions of eigenvalue and eigenvector do not depend on length, angle, or inner product, and so we forsake \mathcal{E}^n for this section only in favor of \mathbb{C}^n . Of central importance in the study of any n -by- n matrix B are those special vectors in \mathbb{C}^n whose spans are not changed when multiplied by B . Any such vector z must satisfy

$$Bz = z\lambda = \lambda z, \quad z \neq 0 \quad (1.11)$$

for some scalar λ , called an *eigenvalue*² of B . Each nonzero multiple of z is an eigenvector,³ and λ and z belong to (or are associated with) each other. By convention o is never an eigenvector.

FACT 1.1. Let $C = FBF^{-1}$. If $\{\lambda, z\}$ is an eigenpair of B then $\{\lambda, Fz\}$ is an eigenpair of C .

The mapping $B \rightarrow FBF^{-1}$ is a *similarity transformation* (more briefly, a similarity) of B . The algebraic view is that similarity is an equivalence relation on the set of n -by- n matrices which preserves eigenvalues and changes eigenvectors in a simple way. The geometric interpretation of Fact 1.1 (and the bane of new students) is that B and C each represent the same (abstract) linear transformation on \mathbb{C}^n while F defines a change of basis in \mathbb{C}^n (old coordinates w , new coordinates Fw). This relationship is shown in Figure 1.3; there are two ways of going from the top left to the top right corner.

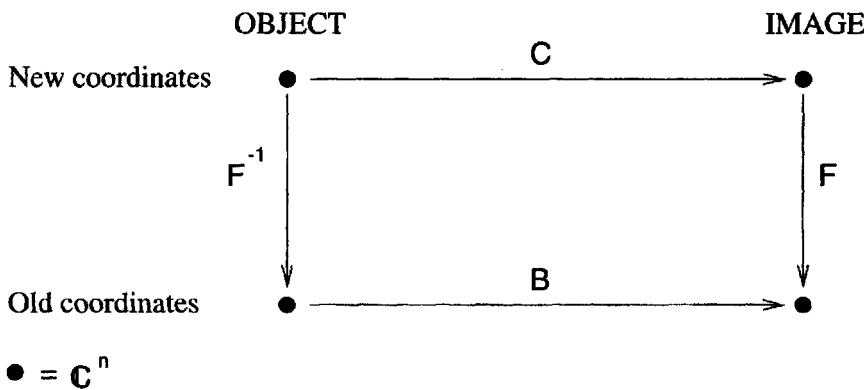


FIG. 1.3. *The similarity transformation.*

The *characteristic polynomial*, χ , is defined by

$$\chi(\zeta) \equiv \chi_B(\zeta) \equiv \det[\zeta I - B]. \quad (1.12)$$

By the theory of linear equations (1.11) has a nonzero solution z if and only if

²In German the word *eigen* means characteristic or special.

³If we adopted the usual symbol for an eigenvector, namely x , the matrix of eigenvectors $X \equiv (x_1, \dots, x_n)$ would then be denoted by a symmetric letter thus violating a useful convention which is introduced in the next section.

$\chi(\zeta) = 0$. So B can have at most n eigenvalues. The set of these values, in the complex plane, constitutes B 's *spectrum*. Note that χ is a *monic* polynomial (its leading coefficient is 1) because we avoided the usual definition $\det[B - \zeta I]$.

1.4. Self-Adjoint Matrices

We now go back to n -by- n matrices premultiplying vectors in \mathcal{E}^n and focus on those matrices which satisfy

$$M^* = M. \quad (1.13)$$

Note that (1.13) is not preserved under arbitrary change of basis in C^n or R^n . In more general contexts than \mathcal{E}^n , linear operators which satisfy (1.13) are called *self-adjoint*, and then there is no need to make a distinction between the real and complex case. Unfortunately in n dimensions such matrices are called *Hermitian*, when complex and *symmetric*, when real. In this chapter we will persevere with the adjective self-adjoint; afterward we shall throw generality to the winds and concentrate on real symmetric matrices. Unless the contrary is stated, letters which are *symmetric about a vertical axis*, namely, A , H , \dots , Y , represent self-adjoint matrices.

FACT 1.2. All eigenvalues of self-adjoint matrices are real.

As a result of Fact 1.2, we may label eigenvalues in increasing order

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n, \quad (1.14)$$

and $\lambda_j[M]$ denotes the j th (smallest) eigenvalue of M . Any normalized eigenvector belonging to λ_i is denoted by z_i ; $Az_i = z_i\lambda_i$, $i = 1, \dots, n$. For large values of n it is sometimes convenient to label the largest eigenvalue without reference to dimension. Hence, we also order the eigenvalues by

$$\lambda_{-n} \leq \dots \leq \lambda_{-2} \leq \lambda_{-1}. \quad (1.15)$$

Thus, λ_{-1} is always the largest eigenvalue (algebraically).

FACT 1.3. If $\lambda_k \neq \lambda_j$ then $(z_j, z_k) = z_k^* z_j = 0$.

A few words must be said about multiple eigenvalues. It is tempting to interpret Fact 1.3 as saying that distinct eigenvectors of self-adjoint matrices

are orthogonal. Consideration of the identity matrix I shows that the proper formulation is that eigenvectors of A “may always be *chosen* to be pairwise orthogonal.” Multiple eigenvalues furnish a wide choice of associated eigenvectors.

If λ is an eigenvalue of A , i.e., if $\lambda = \lambda_j[A]$, then \mathcal{N}_λ , the null space of $A - \lambda I$, i.e., the set of all x such that $(A - \lambda I)x = 0$, is sometimes called the eigenspace belonging to λ . The only vector in \mathcal{N}_λ that is not an eigenvector is 0 . The multiplicity of λ is the dimension of \mathcal{N}_λ . (For nonsymmetric matrices the notion of eigenvalue multiplicity has two meanings.) Eigenspaces are the simplest invariant subspaces (discussed at the end of this section), and one consequence of the next fact is that all invariant subspaces are spanned by eigenvectors.

FACT 1.4 (the spectral theorem). Any A is similar to a diagonal matrix Λ via an orthonormal similarity. In symbols,

$$A = Z\Lambda Z^* = \sum_{i=1}^n \lambda_i z_i z_i^*,$$

$$I = ZZ^* = \sum_{i=1}^n z_i z_i^*.$$

$Z = (z_1, \dots, z_n)$ is a matrix of orthonormalized eigenvectors of A .

Definition 1.4.1. A matrix E is a *projector* if $E^2 = E$. It is an *orthogonal projector* (in contrast to an *oblique projector*) if $Ey = 0$ for all y orthogonal to E 's range (also called *span* E). The condition for this is simply

$$E^* = E. \quad (1.16)$$

For any square matrix B the *spectral projector* (sometimes called *idempotent*) E_λ for an eigenvalue λ satisfies

$$BE_\lambda = E_\lambda B = \lambda E_\lambda. \quad (1.17)$$

To achieve a *unique* decomposition in the presence of multiple eigenvalues one uses the eigenspaces \mathcal{N}_λ or, equivalently, the spectral projectors H_λ defined

by

$$H_\lambda x = \begin{cases} x & \text{if } x \in N_\lambda, \\ 0 & \text{if } x \in N_\mu, \mu \neq \lambda. \end{cases} \quad (1.18)$$

Now the spectral theorem can be written unambiguously as

$$A = \sum \lambda_j H_j, \quad I = \sum H_j \quad (1.19)$$

where the sums are over A's spectrum. Because A is self-adjoint its spectral projectors are also orthogonal projectors which is why we used a symmetric letter H in (1.18) and (1.19).

Example 1.4.1.

$$\begin{aligned} A &= \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}, & \alpha_1 = 0, \\ z_1 &= \frac{1}{2} \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, & H_1 = z_1 z_1^* = \frac{1}{4} \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, \\ z_4 &= \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, & H_4 = z_4 z_4^* = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \end{aligned}$$

z_2, z_3 are not unique. One suitable pair is

$$\frac{1}{2}(1, -1, 1, -1)^*, \quad \frac{1}{2}(1, 1, -1, -1)^*;$$

another pair is

$$(1/\sqrt{2})(1, 0, 0, -1)^*, \quad (1/\sqrt{2})(0, 1, -1, 0)^*.$$

However,

$$H_2 = z_2 z_2^* + z_3 z_3^* = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

is unique and

$$A = 0 \cdot H_1 + 2 \cdot H_2 + 4 \cdot H_4, \quad I = H_1 + H_2 + H_4.$$

1.4.1. Invariant Subspaces

An important consequence of Fact 1.4 is that any subspace \mathcal{S} of \mathcal{E}^n which is invariant under A , i.e., $A\mathcal{S} \subseteq \mathcal{S}$, is just the span of appropriate eigenvectors. Associated with each invariant \mathcal{S} is the linear operator $A|_{\mathcal{S}}$, the *restriction* of A to \mathcal{S} , whose action is the same as A but whose domain is \mathcal{S} . Strictly speaking any function is defined by its action together with its domain, and so any change in domain produces a new operator which must receive its own name. It can be verified that $A|_{\mathcal{S}}$ is self-adjoint and its eigenvalues and eigenvectors are the appropriate subset of those of A .

1.4.2. Hermitian Matrices

We close this section on a practical note.

FACT 1.5. Hermitian matrices H may be replaced by real symmetric matrices \hat{H} of twice the order for the purpose of computing eigensystems.
See Exercise 1.4.6.

Fact 1.5 is useful in two situations; not only when programs for complex Hermitian matrices are unavailable but also when they are available and yet the complex arithmetic operations are implemented inefficiently. When complex arithmetic is implemented well those codes which treat the original Hermitian H should be twice as fast as the programs for real symmetric matrices working on \hat{H} .

Exercises on Section 1.4

- 1.4.1. Find a 2-by-2 complex symmetric matrix (not Hermitian) with complex eigenvalues.
- 1.4.2. Prove Fact 1.2 by considering $z_i^* A z_i$.
- 1.4.3. Prove Fact 1.3 by considering $z_i^* A z_j$.
- 1.4.4. Verify Fact 1.4 for $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Compute $E_1 = z_1 z_1^*$ and $E_2 = z_2 z_2^*$.
- 1.4.5. Show that $\text{rank}(E_j) = \dim(\mathcal{N}_j)$. Note that $\text{rank}(E_j) = \text{trace}(E_j)$. Use the fact that $E_j^2 = E_j$ and $\text{trace}(BC) = \text{trace}(CB)$. $\text{Trace}(F) \equiv \sum_{j=1}^n f_{jj}$.

- 1.4.6. Let $H = M + iS$ be Hermitian ($i^2 = -1$) and let $H(u + iv) = (u + iv)\lambda$, λ real. Verify that $\hat{H} = \begin{bmatrix} M & -S \\ S & M \end{bmatrix}$ is symmetric and has eigenvectors $\begin{bmatrix} u \\ v \end{bmatrix}$ and $\begin{bmatrix} -v \\ u \end{bmatrix}$ belonging to λ .
- 1.4.7. Use Fact 1.3 to show that a spectral projector is orthogonal when A is self-adjoint.
- 1.4.8. Show that $H_\lambda = \sum z_i z_i^*$ where $\{z_i\}$ is any orthonormal basis of \mathcal{N}_λ .
- 1.4.9. Show that if S is a subspace invariant under A then any eigenvalue of $A|_S$ is an eigenvalue of A . Let $\{z_1, \dots, z_n\}$ be an eigenvector basis for \mathbb{C}^n . Is it always true that some subset of $\{z_1, \dots, z_n\}$ is a basis for S ?

1.5. Quadratic Forms

Self-adjoint matrices arise naturally in the study of pure quadratic forms (or functions), not mixed with lower degree terms, typically,

$$\psi(x) \equiv x^* A x = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \bar{x}_i x_j. \quad (1.20)$$

These forms frequently represent some form of energy of a system, be it atom or skyscraper. Linear invertible changes of variable, say $x \rightarrow y = F^{-1}x$, force a change in the form, i.e.,

$$\psi(x) = \hat{\psi}(y) \equiv y^* \hat{A} y \text{ for all } x \quad (1.21)$$

if and only if

$$\hat{A} = F^* A F. \quad (1.22)$$

The mapping $A \rightarrow F^* A F$ is a *congruence transformation* of A ; we say \hat{A} is *congruent* to A . These transformations preserve self-adjointness but do not, in general, preserve eigenvalues. Nevertheless congruencies do in some sense preserve the signs(\pm) of eigenvalues. That is the gist of the next fact.

FACT 1.6 (Sylvester's inertia theorem). Each A is congruent to a matrix $\text{diag}(I_\pi, -I_\nu, O_\zeta)$, where the number triple (π, ν, ζ) depends only on A and is called A 's *inertia*. Moreover π, ν, ζ are the number of positive, negative, and zero eigenvalues of A .

In addition,

$$\begin{aligned} \pi + \nu + \zeta &= n, \\ \pi + \nu &= \text{rank}(A), \\ \pi - \nu &= \text{signature}(A). \end{aligned} \quad (1.23)$$

Fact 1.6 shows that two self-adjoint matrices are congruent if and only if they have the same inertia. See Exercise 1.5.9.

It is overkill to derive the inertia theorem from the spectral theorem.

Among quadratic forms are those which are strictly positive for all nonzero vectors: $\psi(x) > 0$ if $x \neq 0$. Such forms, and the self-adjoint matrices associated with them, are positive definite. (The word definite serves to distinguish these matrices from those which are merely positive, that is, matrices B satisfying $b_{ij} \geq 0$ for all i, j .) If $\psi(x) \geq 0$ for all $x \neq 0$ then it, and its matrix, are *positive semidefinite*. If $\psi(x)$ takes on both positive and negative values it is *indefinite*.

FACT 1.7. The following statements are equivalent:

1. A is positive definite.
2. A 's inertia is $(n, 0, 0)$.
3. A 's eigenvalues are positive.
4. A has a unique Cholesky factor C which is upper triangular with positive diagonal and satisfies $A = C^*C$ (see Chapter 3 on triangular factorization).

In general it requires less work to check statement 4 than any of the other three conditions.

The pure quadratic form ψ is homogeneous of degree 2: $\psi(\alpha x) = \alpha^2 \psi(x)$. Consequently there is no loss of information in restricting attention to its values on the unit sphere in \mathcal{E}^n . The new function is Rayleigh's quotient ρ and is usually defined by

$$\rho(u) \equiv \rho(u; A) \equiv \frac{u^* A u}{u^* u}, \quad u \neq 0. \quad (1.24)$$

FACT 1.8. The Rayleigh quotient enjoys the following basic properties:

Homogeneity: $\rho(\alpha u) = \rho(u)$, $\alpha \neq 0$ (degree 0).

Boundedness: $\rho(u)$ ranges over the interval $[\lambda_1, \lambda_{-1}]$ as u ranges over all nonzero n -vectors.

Stationarity: $\rho(u)$ is stationary (i.e., the gradient of ρ is o^*) at and only at the eigenvectors of A .

Rayleigh's quotient plays a big role in the computation of eigenvalues and eigenvectors and is discussed in more detail in Chapter 4. One basic property is worth mentioning now. For any $u \neq o$ define the special residual vector $r(u)$ by

$$r(u) = [A - \rho(u)]u. \quad (1.25)$$

Figure 1.4 illustrates the fact that

$$Au = \rho(u)u + r(u) \quad (1.26)$$

is an orthogonal decomposition of Au .

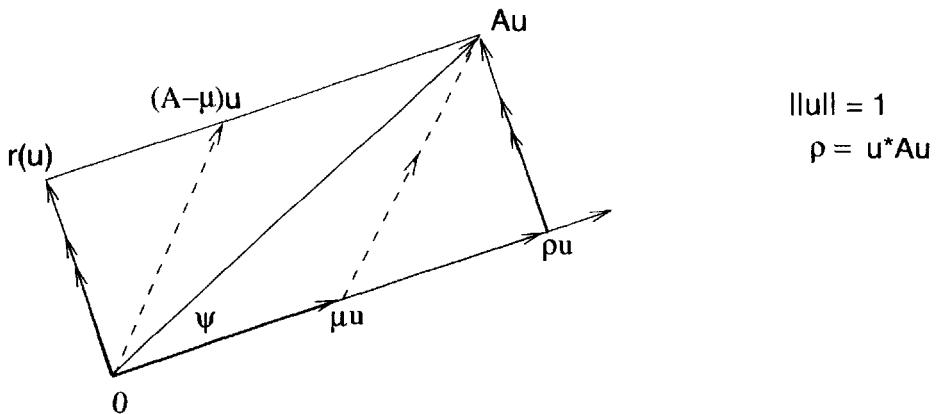


FIG. 1.4. *Geometric meaning of the residual.*

In other words, see Fact 1.9.

FACT 1.9 (the minimal residual property). For each u in \mathcal{E}^n ,

$$\|[A - \rho(u)]u\| \leq \|[A - \mu]u\| \quad \text{for all } \mu \text{ in } \mathbb{C}.$$

Exercises on Section 1.5

- 1.5.1. Express x^*Ax as a sum of squares

$$A = \begin{bmatrix} 10 & 1 & -14 \\ 1 & 17 & -4 \\ -14 & -4 & 20 \end{bmatrix}.$$

- 1.5.2. Find the inertia of

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}.$$

- 1.5.3. Show that the matrix B^*B is positive semidefinite for any m -by- n matrix B . When is it positive definite?

- 1.5.4. Show that the eigenvalues of a positive definite matrix are positive. Is the converse true?

- 1.5.5. Show that the triangular factorization

$$A = L\Delta L^*$$

is unique when it exists. Here L is lower triangular with 1's on the diagonal and Δ is diagonal.

- 1.5.6. Establish Fact 1.8. For the third part try $u = z + \epsilon w$ with $w^*z = 0$, $\|w\| = 1$, $Az = z\lambda$, and let $\epsilon \rightarrow 0$.

- 1.5.7. For the matrix A given in the example of section 1.4 and $u = (1, 1, 1, 0)^*/\sqrt{3}$, evaluate $\rho(u)$ and compute $\|r(u)\|$.

- 1.5.8. Show that if $Az = z\lambda$ then $\rho(z; A) = \lambda$.

- 1.5.9. Prove Sylvester's inertia theorem in two stages. First show that A can be reduced to diagonal form by congruence transformations $A \rightarrow FAF^t$. One way is to use triangular factorization with a sensible strategy for interchanging rows and the corresponding columns. Second assume that A is congruent to two diagonal matrices D_1 and D_2 . Suppose that

$\pi(D_1) < \pi(D_2)$ and derive a contradiction. A useful tool here is that any system of linear homogeneous equations $Cy = 0$ with fewer equations than unknowns must have a nontrivial solution.

1.6. Matrix Norms

There is a special matrix norm associated with the Euclidean vector norm, namely,

$$\|B\| \equiv \max_{u \neq 0} \frac{\|Bu\|}{\|u\|} = \sqrt{(\lambda_i[B^*B])}. \quad (1.27)$$

This is called the *spectral norm* or the *bound norm*. It is the smallest norm which satisfies the useful inequality

$$\|Bu\| \leq \text{norm}(B)\|u\| \quad \text{for all } u \in \mathcal{E}^n. \quad (1.28)$$

Unfortunately it is expensive to compute. Another norm which satisfies (1.28) and is a simple function of the matrix elements is the *Frobenius* or *Schur* or *Hilbert–Schmidt* norm

$$\|B\|_F \equiv \sqrt{\text{trace}(B^*B)} = \left[\sum_i^n \sum_j^n |b_{ij}|^2 \right]^{\frac{1}{2}}. \quad (1.29)$$

For any m -by- n matrix B , with $m \geq n$,

$$\|B\| \leq \|B\|_F \leq \sqrt{\text{rank}(B)} \|B\| \leq \sqrt{n} \|B\|. \quad (1.30)$$

When $B = A = A^*$ then

$$\|A\| = \max \{|\lambda_1|, |\lambda_n|\} \equiv \text{spectral radius of } A, \quad (1.31)$$

$$\|A\|_F = \left[\sum_{i=1}^n \lambda_i^2 \right]^{\frac{1}{2}}. \quad (1.32)$$

The next fact shows that a sequence of unitary transformations cannot change either norm.

FACT 1.10 (unitary invariance of the norms).

$$\|JBG\| = \|B\|, \quad \|JBG\|_F = \|B\|_F$$

for all m -by- n B if and only if J and G are orthonormal, i.e.,

$$J^*J = JJ^* = I_m, \quad G^*G = GG^* = I_n.$$

Each norm finds a natural place in the following useful inequalities.

FACT 1.11 (eigenvalues are perfectly conditioned).

$$\max_j |\lambda_j[\mathbf{A}] - \lambda_j[\mathbf{M}]| \leq \|\mathbf{A} - \mathbf{M}\|,$$

$$\sum_{j=1}^n (\lambda_j[\mathbf{A}] - \lambda_j[\mathbf{M}])^2 \leq \|\mathbf{A} - \mathbf{M}\|_F^2.$$

The latter inequality is called the *Wielandt–Hoffman inequality*. An elementary but long proof can be found in [Wilkinson 1965, Chapter 3].

The first inequality is proved in Chapter 10. It is often interpreted as saying that each eigenvalue of a self-adjoint matrix is *perfectly conditioned*; that is, the (absolute) change in an eigenvalue is not more than the (absolute) change in the matrix. In other words, the problem of determining eigenvalues of self-adjoint matrices is always well posed; the solution is well determined by the data. This is not the case for some nonsymmetric matrices.

It is also valuable to know for which matrices \mathbf{A} some eigenvalues, particularly those close to 0, are determined well, in a *relative* sense, by \mathbf{A} . This means that uncertainty in the sixth decimal, say, of the entries of \mathbf{A} causes uncertainty in the sixth decimal of an eigenvalue. See [R.-C. Li, 1994, Parts I and II] and [Ch. K. Li and Mathias, 1998, to appear].

For eigenvectors the situation is more delicate. Let $\mathbf{A}\mathbf{z} = \mathbf{z}\alpha$, $\mathbf{M}\mathbf{s} = \mathbf{s}\mu$. If μ is separated by a gap γ from \mathbf{A} 's eigenvalues other than α , then

FACT 1.12. $|\sin \angle(\mathbf{z}, \mathbf{s})| \leq \|\mathbf{A} - \mathbf{M}\|/\gamma.$

This fact, and some extensions of it, are established in Chapter 11.

What is remarkable about these bounds is that they are not asymptotic; there is no requirement that the “perturbation” $\mathbf{M} - \mathbf{A}$ be small. On the other hand, without a gap eigenvectors can be very sensitive functions of the data. Suppose that \mathbf{A} 's elements are functions of a parameter t . If $\mathbf{A}(t_0)$ has a multiple eigenvalue, then there is no guarantee that the normalized eigenvectors vary continuously in a neighborhood of t_0 as is revealed in the following example

constructed by Givens:

matrix: $\mathbf{A}(t) = \begin{bmatrix} 1 + t \cos(2/t) & t \sin(2/t) \\ t \sin(2/t) & 1 - t \cos(2/t) \end{bmatrix},$

spectrum: $\{1 + t, 1 - t\},$

eigenvectors: $\begin{bmatrix} \cos(1/t) \\ \sin(1/t) \end{bmatrix}, \begin{bmatrix} \sin(1/t) \\ -\cos(1/t) \end{bmatrix}.$

As $t \rightarrow 0$ the normalized eigenvectors become dense in the unit disc in the plane and when $t = 0$, $\mathbf{A}(0) = \mathbf{I}$, the eigenvectors do fill out the unit disc (see Exercise 1.6.8).

The discontinuity at $t = 0$ in the formula for the eigenvectors must not be construed as signaling a pathological situation. It merely signals that two distinct eigenlines have joined together to become an eigenplane in which no single pair of vectors is distinguished from any other pair.

These remarks suggest that the proper object to compute when a few eigenvalues are tightly clustered is *any* orthonormal basis for the invariant subspace belonging to the whole cluster rather than separate eigenvectors for each eigenvalue. But where do you draw the line between merely close and tightly clustered eigenvalues?

Exercises on Section 1.6

1.6.1. Prove that $\|\mathbf{A}\| = \|\mathbf{A}\|_F$ if and only if $\mathbf{A} = \pm \mathbf{u}\mathbf{u}^*$ for some \mathbf{u} .

1.6.2. Prove (1.31) using Fact 1.4.

1.6.3. Show that $\|\mathbf{B}\|_F$ is the Euclidean norm of \mathbf{B} in the n^2 -dimensional space of n -by- n matrices.

1.6.4. Prove (1.27) using Fact 1.4.

1.6.5. Prove Fact 1.10.

- 1.6.6. Prove that for each j there is a k such that $|\lambda_j[\mathbf{A}] - \lambda_k[\mathbf{M}]| \leq \|\mathbf{A} - \mathbf{M}\|$ by writing $\mathbf{A} - \mathbf{M} = \mathbf{H}$, $\mathbf{M} = \mathbf{P}\Lambda\mathbf{P}^*$ and using the fact that determinant $(\mathbf{M} + \mathbf{H} - \alpha)$ vanishes when $\alpha = \lambda_j[\mathbf{A}]$. Fact 1.6 must also be used. This result is a special case of the Bauer–Fike theorem: given $\mathbf{B} = \mathbf{G}^{-1}\Lambda\mathbf{G}$, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, then to each λ_j there is an eigenvalue μ of $\mathbf{B} + \mathbf{C}$ such that $|\mu - \lambda_j| \leq \|\mathbf{G}\|\|\mathbf{G}^{-1}\|\|\mathbf{C}\|$. The extra fact in the self-adjoint case is that the correspondence between μ and λ_j is one to one.
- 1.6.7. Verify Fact 1.12 when $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, $\mathbf{M} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, and $\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
- 1.6.8. Compute one eigenvector of the matrix $\mathbf{A}(t)$ (from the given formula) when $t = 10^{-3}$ and $t = 10^{-4}$. Through what angle does the eigenvector change?
- 1.6.9. Define $\|\mathbf{B}\|_k^2 \equiv \sum_{i=1}^k \lambda_{-i}[\mathbf{B}^*\mathbf{B}]$. Show that this is a unitarily invariant matrix norm.

1.7. The Generalized Eigenvalue Problem

In many branches of science the problems involve two quadratic forms. In mechanics, for example, $\mathbf{u}(t)$ may represent a state of some system, $\dot{\mathbf{u}}$ its time derivative, $\dot{\mathbf{u}}^*\mathbf{M}\dot{\mathbf{u}}$ its kinetic energy, and $\mathbf{u}^*\mathbf{A}\mathbf{u}$ its potential energy. Physical principles dictate that, in the absence of external forces, the actual state \mathbf{u} will minimize the ratio of these energies. Consequently a key function in problems with two quadratic forms, \mathbf{A} and \mathbf{M} , is the *Rayleigh quotient*, defined for all $\mathbf{u} \neq \mathbf{o}$ by

$$\rho(\mathbf{u}) = \rho(\mathbf{u}; \mathbf{A}, \mathbf{M}) = \frac{\mathbf{u}^*\mathbf{A}\mathbf{u}}{\mathbf{u}^*\mathbf{M}\mathbf{u}} \quad (\mathbf{M} \text{ positive definite}). \quad (1.33)$$

It turns out that ρ is stationary at $\mathbf{z} \neq \mathbf{o}$ if and only if

$$(\mathbf{A} - \lambda\mathbf{M})\mathbf{z} = \mathbf{o} \quad (1.34)$$

for some scalar λ . We say that (λ, \mathbf{z}) is an eigenpair of the *pair*, or *pencil* (\mathbf{A}, \mathbf{M}) . Fact 1.8 continues to hold for $\rho(\mathbf{u}; \mathbf{A}, \mathbf{M})$ provided that either \mathbf{A} or \mathbf{M} , or some $\alpha\mathbf{A} + \mu\mathbf{M}$, is positive definite. We shall assume that \mathbf{M} is positive definite.

The proper setting for this problem is not \mathcal{E}^n but the inner product space \mathcal{M}^n consisting of \mathbb{C}^n (or \mathbb{R}^n) furnished with the inner product

$$(\mathbf{x}, \mathbf{y}) = \mathbf{y}^*\mathbf{M}^{-1}\mathbf{x} \quad \text{or} \quad (\mathbf{x}, \mathbf{y}) = \mathbf{y}^*\mathbf{M}\mathbf{x}. \quad (1.35)$$

The fundamental notions of length and angle defined by (1.2), (1.4), and (1.5) take on new values when the inner product given in (1.35) replaces the

familiar Euclidean version, but the spectral theorem and most of section 1.4 extend to pencils (A, M) with positive definite M .

Abstractly, this problem is indistinguishable from the standard eigenvalue problem (one inner product is as good as another), but, in practice, the presence of M complicates the task and increases the cost. Chapter 15 explores the problem in greater detail.

This page intentionally left blank

Tasks, Obstacles, and Aids

2.1. What Is Small? What Is Large?

In 1954 came the invention of the programming language FORTRAN and the ensuing ability of the programmer to access any element of a matrix A as easily as any other by simply writing $A(I, J)$. Each computer system can accommodate square matrices up to a certain order as conventional two-dimensional arrays. These are the storable or stored matrices. Naturally the precise upper limit on the order of such matrices varies from system to system, but the meaning of the adjective *small* is always that *all matrix elements are equally and rapidly accessible*. On most scientific computers a full 50-by-50 matrix is small. In addition a matrix is called *dense* if it is not worth trying to exploit the presence of any zero entries. Algorithms for such matrices act as if there were no zero entries.

Now that satisfactory methods are available for most, but not all, eigenvalue tasks for small dense symmetric matrices, attention has turned to harder problems. We say that a matrix is large if only part of it can be held at one time in high-speed storage. However, the order of the matrix is too crude a measure of storage demand. A 1000-by-1000 symmetric matrix with a half-bandwidth of 20 (i.e., $a_{ij} = 0$ if $|i - j| > 20$) is small for many computing systems, whereas a 400-by-400 symmetric matrix with a random pattern of zero elements might be large. The two phrases *conventionally storable* and *not conventionally storable* would be more precise, but we prefer the simpler words *small* and *large*.

It happens that most large matrices are also sparse (most of their elements are zero), but, as indicated above, what matters is how easily the pattern of zero elements can be exploited. A property such as narrow bandwidth⁴ is too

⁴The half-bandwidth of A is defined as $\max |i - j|$ over all i, j such that $a_{ij} \neq 0$.

valuable to be lightly sacrificed.

A key factor in handling large sparse matrices is the way in which their nonzero elements are to be represented. For example, one obvious possibility is to create a list of the nonzero values and attach to each value, a_{ij} , five integers,

$$m, i, j, k, l,$$

where m is the position of this element in the list, k points to the position of the next nonzero element in row i , and l points to the next nonzero element in column j . There are better schemes, but we shall not pursue this matter any further here although it is an important aspect of the efficiency with which sparse matrices can be handled. A good introduction to this subject is given in [Duff, Erisman, and Reid, 1987]. A sparse matrix from the Harwell–Boeing collection (see [Duff, Grimes, and Lewis, 1992]) is shown in Figure 2.1.

Another category consists of *structured* matrices. Here the n^2 entries depend on only n or $2n$ parameters. Examples are Toeplitz matrices (entry (i, j) depends on $i - j$), Hankel matrices (entry (i, j) depends on $i + j$), and Cauchy matrices (entry (i, j) is $1/(c_i - d_j)$). The trick here is to try to stay within the class as the matrices are transformed. These techniques are beyond the scope of this book and are advancing rapidly at the turn of the century.

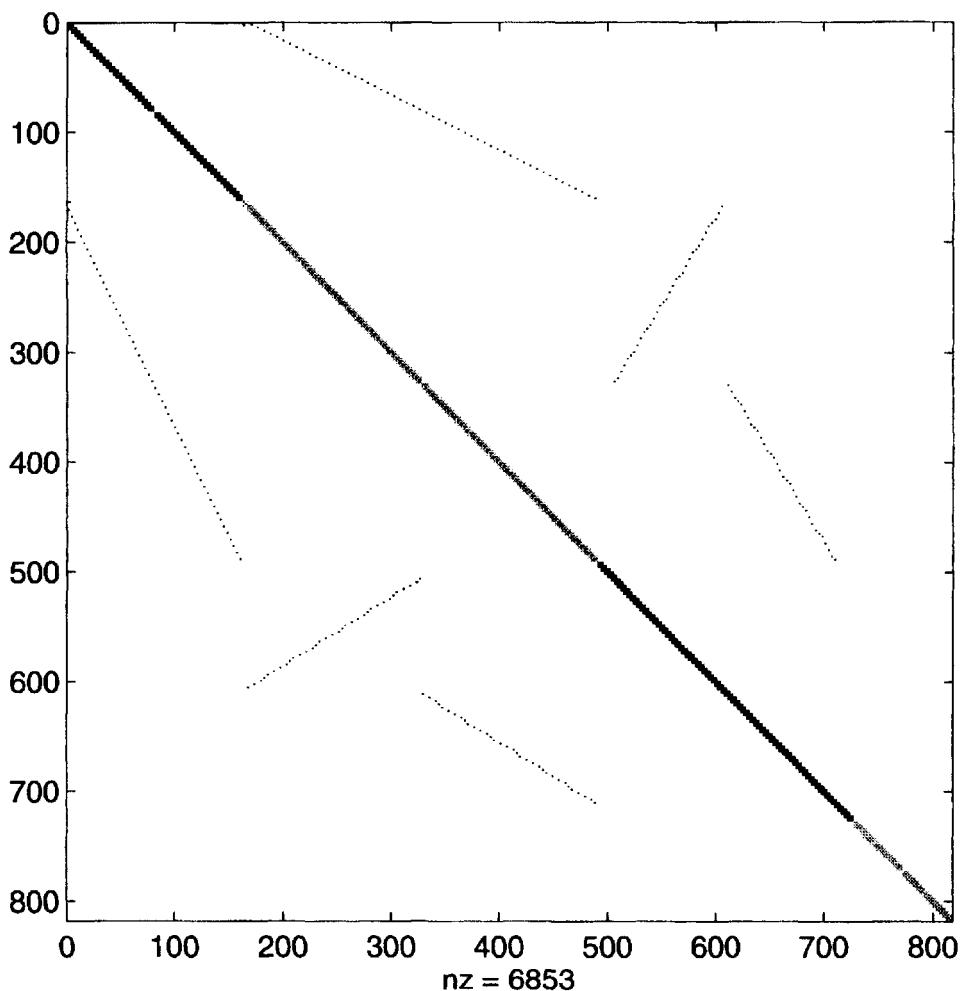
2.2. Tasks

Mathematically this book is concerned with a single problem—the computation of eigenvalues and eigenvectors of symmetric matrices. However, as soon as we get down to details and confront questions of efficiency the underlying unity gives way to a mosaic of different tasks. To a certain extent the best method for one task turns out to be inadequate for another, and this fact leads to an unpleasant profusion of programs.

Before listing some typical requests in order to give the flavor of the subject we must emphasize a feature of today's tasks which sets them apart from those of 1950.

The results of many, but not all, matrix calculations are of no intrinsic interest. They are merely intermediate quantities needed in a larger computation.

1. Find the smallest eigenvalue, $\lambda_1[\mathbf{F}^*\mathbf{F}]$, where $\mathbf{F} = (\mathbf{f}_1, \dots, \mathbf{f}_k)$, $\|\mathbf{f}_i\| = 1$,

FIG. 2.1. *Portrait of a sparse matrix.*

$i = 1, \dots, k$, correct to two significant figures. The number $\sqrt{\lambda_1}$ is a simple measure of linear independence among the f_i .

2. Find all the eigenvalues of a small matrix A correct to working precision but no eigenvectors. Such calculations occur, for example, in the course of computing a few eigenvalues of large matrices. In some analyses of electrical networks all the λ_i 's themselves are wanted [Cullum and Willoughby, 1985].

3. Find the p smallest (leftmost) eigenvalues of large \mathbf{A} and their matching eigenvectors correct to four significant decimals. Such tasks ($p = 30, n = 6000$) occur in nuclear shell model calculations.
4. Find the p smallest (leftmost) eigenvalues, λ_i , and their eigenvectors, \mathbf{z}_i , for the pencil (\mathbf{A}, \mathbf{M}) , i.e., satisfying $(\mathbf{A} - \lambda_i \mathbf{M})\mathbf{z}_i = \mathbf{0}$, $i = 1, \dots, p$. In the analysis of bridges, buildings, and vehicles suffering small displacements from equilibrium each \mathbf{z}_i represents the shape of a normal mode vibration and $\sqrt{\lambda_i}$ represents the corresponding frequency of that mode. Moreover, if the vector $\mathbf{u}(t)$ represents the displacement of the structure at time t then $\mathbf{u}^* \mathbf{A} \mathbf{u}$ gives the potential (or strain) energy while $\dot{\mathbf{u}}^* \mathbf{M} \dot{\mathbf{u}}$ gives the kinetic energy of the system. Here $\dot{\mathbf{u}}$ is the derivative of \mathbf{u} . In practice \mathbf{A} and \mathbf{M} are themselves approximations, and the higher eigenpairs have no physical meaning.
5. Find all eigenvalues in a given interval and their matching eigenvectors. Such tasks occur in modelling the observable tides in the Atlantic Ocean [Cline, Golub, and Platzman, 1976] and also in testing whether structures are likely to withstand local earthquakes.
6. To discover significant variables in statistical analysis it is useful to find a few of the largest eigenpairs of a correlation matrix $\mathbf{F}\mathbf{F}^*/(m - 1)$, where \mathbf{F} is n by m and $1 < m \ll n$.

2.3. Conflicting Requirements

One of the attractions of the study of numerical methods is to see how well various methods satisfy the following mutually incompatible demands: reliability, appropriate accuracy, swift execution, low storage, and short programs. For any particular computing task one or more of these requirements may be in abeyance. The dramatic developments in computer technology keep upsetting the balance between these five goals and, no doubt, will continue to do so.

2.3.1. Reliability

As computing tasks get harder there is a rise in the possibility that a method will sometimes fail. Failure is no disgrace; indeed, it is the only reasonable response to an illegitimate request: e.g., for the real square root of a negative number or the Cholesky factorization of an indefinite matrix. The unpardonable sin is for a method to lie, to deliver results which appear to be reasonable but which are utterly wrong.

A numerical method, or a program, can be regarded as a function mapping input to output. As such the user is concerned with its domain, i.e., the set of matrices for which the method works. Sometimes this domain is unknown, and then failure can be passed off as the discovery that a particular set of data (a certain matrix, perhaps) is not in the domain of the method. Reliability also concerns the way in which failure is signaled to the rest of the computation: cryptic message, abort the job, or graceful termination.

Of course, the user wants to have methods with broad and cleanly described domains. Whether the user can be satisfied depends on the task, on the power of the method, and on the completeness of its analysis.

2.3.2. Accuracy

It seems reasonable, at first glance, to seek methods which are capable of giving results to any desired precision. We might hope that results with low accuracy would cost less than those with high accuracy. In practice, however, numerical methods do not work that way.

First, such flexibility may not be possible. The eigenvalues and eigenvectors of a matrix are strongly interconnected. It is quite plausible that acceptance of a two-decimal approximation to λ_{\max} may impair, or even prevent, the calculation of a two-decimal approximation to λ_{\min} , and so on.

Second, the number representation in most computers is quite rigid; arithmetic is performed in floating point mode with a fixed precision which is, in most cases, equivalent to 7, 14, or 16 decimals. Operations with twice the working precision are occasionally available.

The eigenvalues of symmetric \mathbf{A} are sturdy functions of \mathbf{A} 's elements and so are eigenvectors which belong to isolated eigenvalues. For small \mathbf{A} the major source of error is roundoff, and the methods which will be described usually give the exact results for a matrix $\mathbf{A} + \mathbf{H}$ for some unknown \mathbf{H} . In all cases we expect that

$$\|\mathbf{H}\| < n\epsilon\|\mathbf{A}\|,$$

where ϵ is the relative precision of the arithmetic. This is a strong, simple statement but cannot be backed up by proof. To be rigorous one must replace n by a fussy, uninformative function of n which depends on the details of the particular method. This function reflects the difficulty of the analysis quite as much as it reflects the real possibilities for error growth.

In any case, by Fact 1.11, the error in a computed eigenvalue is bounded by $\|\mathbf{H}\|$, and for most purposes this is far more accuracy than is needed. Two points must be made about this apparently excessive accuracy. The marginal savings reaped by accepting less accurate results are negligible, and, second, the

cost of small matrix computations has decreased steadily. All the eigenvalues of a 50-by-50 matrix can be computed for less than a 1970 dollar.

The situation is rather different for very large matrices ($n > 10,000$). Roundoff error does not dominate the accuracy of the results, and the iterative methods in current use sometimes converge quite slowly. There is a high premium on those methods which are capable of delivering the desired accuracy and no more! Here again it often happens that if 20 eigenvalues are wanted, then the first 15 will be correct to working precision by the iterative step at which the twentieth is accurate to four significant decimals.

2.3.3. Swift Execution

This is the traditional criterion for comparing two methods each of which is reliable. For historical reasons the standard measure of execution time used to be the number of multiplications required. In complex computing systems, with multiprocessing or time sharing and optimizing compilers, the multiplication count had become much less relevant by 1975 than it was in 1960. However, it still has some value, and gross differences in operation count (n^4 versus n^3 operations) will be reflected, in subdued form, in execution costs.

As computer technology advanced, the time required to perform one multiplication dropped (from 10^{-3} seconds in 1955 to 10^{-6} seconds in 1970 to 10^{-9} in 1988) and so the duration of small matrix computations seems to have fallen below the threshold at which it is worth attention. Although this is the case for large scientific computers the picture has changed significantly as more and more small computations are performed on miniprocessors, desktop computers, and even on hand-held calculators. By the end of the 1980s the communication costs (fetching and storing the data) were as important as the arithmetic costs. Moreover the ability to have several or even many arithmetic units at work simultaneously has made judgments very difficult.

2.3.4. Storage Requirements

As storage devices have developed, this criterion has suffered violent changes in status. Initially, around 1950, memory was of paramount importance. With the advent of large (32,000 cell!) ferrite core storage devices it seemed to vanish (circa 1960) and then, with the rise of large matrix problems and small computers, it reappeared in full strength by 1980. In addition the hierarchy in storage continues to acquire more levels: registers, caches, main memory, and secondary storage. The goal is to keep all the arithmetic units busy.

For small matrices the issue is the number of n -by- n arrays which are

needed and whether the original A must be saved. For large matrices the issue is the number of n -vectors which are needed in the fast storage device.

There are computer systems with massive primary stores and also systems in which no part of the data is kept permanently in the fast store.

2.3.5. Short Program

This property of an algorithm is, in one way, subsumed under the storage needs. Nevertheless, it appears separately in two contexts. As reliable program libraries containing more and more programs began to appear in the 1970s, the proliferation problem forced itself on the attention of the makers of these libraries. What a relief if one program could be made to carry out all eigenvalue tasks for small symmetric matrices even if it were not optimal for any of them! This thorny question is far from resolved and falls in the domain of mathematical software rather than mathematical analysis.

The other environment where program length is paramount is the hand-held calculator whose programs must fit into the 100 or 200 steps allotted to them. The demand for small matrix manipulation in desktop computers is already of commercial significance.

On the other hand, in many scientific computing laboratories program length is of no consequence at all.

2.4. Finite Precision Arithmetic

Consider the representation of $10^{10}\pi = 10^{11} \times 0.314159265\dots$ in a number system which permits only five decimal digits in the fractional part of its numbers. Many digital computers simply drop the extra digits to produce the chopped version

$$(10^{10}\pi)_{\text{chopped}} = 0.31415 \times 10^{11} \quad \begin{array}{l} \text{absolute error} \\ 0.93 \times 10^6 \end{array} \quad \begin{array}{l} \text{relative error} \\ 0.29 \times 10^{-4} \end{array}.$$

A more accurate, but more expensive approximation, is the rounded version

$$(10^{10}\pi)_{\text{rounded}} = 0.31416 \times 10^{11} \quad \begin{array}{l} \text{absolute error} \\ -0.73 \times 10^5 \end{array} \quad \begin{array}{l} \text{relative error} \\ -0.23 \times 10^{-5} \end{array}.$$

Uniqueness in the representation of these computer numbers is obtained by normalizing them: the fractional part is less than one and has a nonzero leading digit.

What can be said in general about the size of the inevitable roundoff errors in such a system? A moment's reflection shows that an upper bound on the absolute error must involve the absolute value of the number, whereas an upper bound on the relative error for normalized numbers does not. Consequently it is usually simpler to discuss relative error. For example, on the five-digit system used above the relative error in chopping is always less than 10^{-4} (in absolute value). This bound is halved if results are rounded. (The fact that most computers work with numbers represented in bases 2 or 16 rather than 10 is irrelevant to the points being made here.)

Consider next the basic arithmetic operations $+, -, \times, /$ which act on normalized numbers held in the machine. To be specific we may suppose that the numbers are written in floating point (i.e., scientific) decimal notation with unlimited exponents but only five fractional digits. Even if the exact result of an addition or multiplication were obtained by the arithmetic unit, the computer must round or chop when storing it. Thus the best that can be hoped for is that the relative error in each arithmetic operation is bounded by 10^{-4} . This number, 10^{-4} , or its analogue for each computer, is called the *machine epsilon* or the relative precision of the computer and denoted by *macheeps*.

The system which we are discussing is called *floating point arithmetic* because the proper exponents of the results are produced automatically by the arithmetic unit. (In the 1950s some computers left this chore to the programmer and worked in fixed point mode.) A useful notational device is that the stored result of any calculation C is written $fl(C)$. This is to be read as "the floating point result of C ." The *unit roundoff* $\epsilon = macheeps/2$ when results are rounded to the nearest representable number instead of being chopped to the permissible length.

$$\epsilon = \text{minimum } \xi \text{ such that } fl(1 + \xi) > 1.$$

Floating point binary numbers are uniformly spaced when the exponent is fixed, but this spacing doubles each time the exponent increases by one.

In order to avoid distracting details we shall use the following standard model.

Each basic arithmetic operation is done exactly in the arithmetic unit. The only error occurs when the result is stored.

The actual behavior of most digital computers does not conform to the model, but their departures from it are not significant for this book.

It is worth mentioning that on many computers the word length in the registers is greater than the word length in the cache or main memory. Thus results can change depending on whether or not some intermediate quantity is retained in a register or put back to the cache.

The model leads to some very simple relations on which error analyses are based. Let α and β be any normalized floating point numbers and let \square stand for any of $+, -, \times, /$; then

$$fl(\alpha \square \beta) = (\alpha \square \beta)(1 + \rho) \quad (\beta \neq 0 \text{ for } /), \quad (2.1)$$

where ρ , the relative error, is a complicated function of α, β , and \square and the precision of the arithmetic. Nevertheless, ρ satisfies

$$|\rho| < \epsilon, \quad (2.2)$$

where ϵ is independent of α, β , and \square and is the unit roundoff of the machine. The thrust of error analysis is to majorize ρ by ϵ and thus obtain simple but pessimistic error bounds which are as independent of the data as possible.

It should be pointed out that absolute and relative errors are not rivals. In more complicated calculations the absolute errors in the parts are needed for the relative error of the whole. The fundamental relation (2.1) can be interpreted in various ways, all valid and each useful on the appropriate occasion. For example, close to α and β are some real numbers $\bar{\alpha} \equiv \alpha(1 + \rho)$ and $\bar{\beta} \equiv \beta(1 + \rho)$, not usually computer numbers, which are related to α and β as follows:

$$\begin{aligned} fl(\alpha \pm \beta) &= \bar{\alpha} \pm \bar{\beta}, \\ fl(\alpha\beta) &= \begin{cases} \bar{\alpha}\bar{\beta} \text{ and} \\ \alpha\bar{\beta} \text{ and} \\ [\alpha(1 + \rho)^{1/2}] [\beta(1 + \rho)^{1/2}], \end{cases} \\ fl(\alpha/\beta) &= \begin{cases} \bar{\alpha}/\beta \text{ and} \\ \alpha/\bar{\beta}. \end{cases} \end{aligned} \quad (2.3)$$

In each case the computed result is regarded as the exact operation with slightly perturbed data. This interpretation can be very useful and goes under the name of a *backward* analysis. It aims to put roundoff error on the same footing

as uncertainty in the data. Wilkinson has used this approach to give intelligible yet rigorous error analyses of the majority of methods used for small matrix calculations. It is necessary to look at some earlier attempts to analyze the effects of roundoff in order to appreciate the great simplification he brought to the subject.

The more conventional approach which simply bounds the error in a final or intermediate result is called *forward* analysis. The two approaches are not rivals, and success usually comes through an adroit use of both techniques.

A thorough and elementary presentation of error analysis is given in [Wilkinson, 1964], and so we hope that the reader will be content with an informal discussion of a few important issues.

2.5. Cancellation

Genuine subtraction produces the difference of two numbers with the same sign, say plus. There is a widespread misconception that subtraction of numbers which are very close is an inherently inaccurate process in which the relative error is greatly enhanced. This is known as *catastrophic cancellation*.⁵

As with most misconceptions there is a grain of truth in the saying, but the simple fact is that the error in the subtraction of normalized floating point numbers with the same exponent is zero. Thus

$$fl(0.31416 \times 10^{11} - 0.31415 \times 10^{11}) = 0.10000 \times 10^7.$$

What about close numbers with adjacent exponents? According to our model the exact result will be formed and will require at most five digits (probably fewer) for its representation. Thus, e.g.,

$$fl(0.10012 \times 10^{-8} - 0.99987 \times 10^{-9}) = 0.13300 \times 10^{-11},$$

and again there is no error at all.

Arithmetic units which conform to the IEEE standard for Floating Point Binary Arithmetic, P754, enjoy the following property: if x and y are normalized floating point numbers satisfying $\frac{1}{2} \leq \frac{x}{y} \leq 2$, then

$$fl(x - y) = x - y.$$

At this point we must make a brief digression to mention that the most significant feature which prevents a number of computers (including the Control Data machines) from adhering to our simple model of arithmetic is the

⁵The reader will find no discussion of such catastrophes in the elegant theories of R. Thom. (See Zeeman 1976.)

lack of a *guard digit*. A guard digit is an extra digit on the low-order end of the arithmetic register whose purpose is to catch the low-order digit which otherwise would be pushed out of existence when the radix points are aligned. Without the guard digit the final 7 in Example 2.5.1 is lost, and the relative error is *enormous*.

Example 2.5.1.

$$\begin{array}{rcl} \alpha & = & 0.10012 \times 10^{-8} \\ \beta & = & 0.09998[7] \times 10^{-8} \\ \hline 0.00014 & \times 10^{-8} & = 0.14 \times 10^{-11} \text{ (normalized)} \end{array}$$

Absolute error: -0.70×10^{-13} ,

Relative error: $-0.53 \times 10^{-1} \approx -500\epsilon !!$

For machines without a guard digit, we have the anomalous result that the relative error in the subtraction of very close numbers is usually 0 but can be as great as 9 ($\approx 10^5\epsilon!$) when the exponents happen to differ. Thus the bound in (2.2) fails completely for such machines in these special cases. Nevertheless, it is still the case that

$$fl(\alpha - \beta) = \bar{\alpha} - \beta \quad (|\alpha| \geq |\beta|),$$

where

$$\bar{\alpha} = \alpha(1 + \rho), \quad |\rho| < \epsilon.$$

Thus $\bar{\alpha} = 0.100127 \times 10^{-8}$ in the example above.

Let us return to our model in which the subtraction of close numbers is always exact. The numbers α and β will often, but not always, be the result of previous calculations and thus have an inherent *uncertainty*. It is the relative *uncertainty* in the difference which is large when cancellation occurs, not the relative error in subtraction. This distinction is *not* academic; sometimes the close numbers are exact and their difference has no uncertainty and no error. The phenomenon of cancellation is not limited to computer arithmetic and can be expressed formally as follows. Let $0 < \beta < \alpha$, $\beta \approx \alpha$, and let η_α and η_β be the relative uncertainty in α and β , respectively; then

$$\begin{aligned} \alpha(1 + \eta_\alpha) - \beta(1 + \eta_\beta) &= \alpha - \beta + (\alpha\eta_\alpha - \beta\eta_\beta) \\ &= (\alpha - \beta)(1 + \rho), \end{aligned}$$

where

$$\rho = (\alpha\eta_\alpha - \beta\eta_\beta)/(\alpha - \beta)$$

and

$$|\rho| < \left(\frac{\alpha}{\alpha - \beta} \right) (|\eta_\alpha| + |\eta_\beta|).$$

In exact arithmetic the magnification factor ($\alpha/(\alpha - \beta)$) can be arbitrarily large, but in our decimal system it is bounded by $10/\epsilon$.

There is more to be said. If α and β were formed by rounding previous calculations then information, in the form of low-order digits, was discarded in previous storage operations. It is this lost information we mourn when cancellation occurs. Those discarded digits seemed negligible at the time. The subtraction is not to blame; it merely signals the loss.⁶ The following examples illustrate these remarks.

Example 2.5.2. In our simple model of arithmetic the relative error in $fl(\alpha + \beta)$ is bounded by ϵ , the unit roundoff. However, the relative error in $fl(\alpha + \beta + \gamma)$ can be as large as 1. Take $\alpha = 1$, $\beta = 10^{17}$, and $\gamma = -10^{17}$. The first addition $fl(\alpha + \beta)$ produces β (α is annihilated) and the second $fl(\beta + \gamma)$ produces complete cancellation. This example also shows how the associative law of addition fails. The calculation $fl(\alpha + fl(\beta + \gamma))$ produces the correct value in this case.

Example 2.5.3. (See section 6.9 for the remedy.)

Find the unit vector x orthogonal to y in the plane $\text{span}(z, y)$. The formula is $(I - yy^*/y^*y)z$.

$$y = \begin{bmatrix} 0.16087 \times 10^0 \\ -0.11852 \times 10^0 \\ 0.98216 \times 10^{-1} \end{bmatrix}, \quad z = \begin{bmatrix} -0.50069 \times 10^{-1} \\ 0.36889 \times 10^{-1} \\ -0.30569 \times 10^{-1} \end{bmatrix},$$

$$\theta = \frac{z^*y}{y^*y} = -0.31123 \text{ (in five-decimal chopped arithmetic),}$$

$$w = \theta y = \begin{bmatrix} -0.50067[5701] \times 10^{-1} \\ 0.36886[9796] \times 10^{-1} \\ -0.30567[7657] \times 10^{-1} \end{bmatrix},$$

⁶The ancient Spartans, so I am told, used to execute messengers who had the misfortune to bring bad news.

$$\begin{aligned}\tilde{x} = z - w &= \begin{bmatrix} -0.50069 \times 10^{-1} \\ 0.36889 \times 10^{-1} \\ -0.30569 \times 10^{-1} \end{bmatrix} - \begin{bmatrix} -0.50067 \times 10^{-1} \\ 0.36886 \times 10^{-1} \\ -0.30567 \times 10^{-1} \end{bmatrix} \\ &= \begin{bmatrix} -2.0000 \times 10^{-6} \\ 3.0000 \times 10^{-6} \\ -2.0000 \times 10^{-6} \end{bmatrix}.\end{aligned}$$

Now compare the computed and exact results:

$$x = \tilde{x}/\|\tilde{x}\| \quad \text{Exact Solution}$$

$$\begin{bmatrix} -0.48507 \\ 0.72760 \\ -0.48507 \end{bmatrix} \quad \begin{bmatrix} 0.62307 \\ 0.77789 \\ -0.81837 \times 10^{-1} \end{bmatrix}$$

$$\cos \phi = \frac{x^*y}{\|x\|\|y\|} = -0.95177; \quad \phi = 2.8298 \text{ rad} \approx 162^\circ!$$

This ridiculous output is easily avoided as section 6.9 reveals.

We conclude this section with three pieces of numerical wisdom.

- When an algorithm does turn out to be unreliable it is not because millions of tiny roundoff errors gradually build up enough to contaminate the results; rather, it is because the rounding of a few numbers (perhaps only one) discards crucial information. Error analysis aims to detect such sensitive places.
- Severe cancellation is not always a bad thing. It depends on the role of the difference in the rest of the computation. Complete cancellation occurs at the root of an equation.
- Subtractions which may provoke severe cancellation can sometimes be transformed algebraically into products or quotients, e.g.,

$$\begin{aligned}|\alpha| - \sqrt{\alpha^2 - \beta^2} &= \beta^2 / (|\alpha| + \sqrt{\alpha^2 - \beta^2}), \\ \alpha^2 - \beta^2 &= (\alpha - \beta)(\alpha + \beta).\end{aligned}$$

2.6. Inner Product Analysis

2.6.1. Numerical Example

Once in a lifetime a user of computer arithmetic should examine the details of a backward error analysis. Here is a detailed account of the computation of y^*z for the vectors of Example 2.5.3 given above. When these details are mastered, the general pattern becomes clear. In this section y_i denotes the i th element of the vector y , and an overbar is added each time a quantity is involved in an arithmetic operation.

The product $y_1 z_1$ is $-0.805460003 \times 10^{-2}$ and the stored result $(-0.80546 \times 10^{-2})$ can be written $\bar{y}_1 z_1$, where $\bar{y}_1 = y_1(0.80546/0.805460003) = 0.16086999992$. Similarly the product $y_2 z_2$ is $-0.437208428 \times 10^{-2}$, and the stored result $(-0.43720 \times 10^{-2})$ can be written $\bar{y}_2 z_2$, where $\bar{y}_2 = y_2(0.43720/0.437208428) = -0.1185177153$. The sum $s_2 = \bar{y}_1 z_1 + \bar{y}_2 z_2$ is $-0.1242660 \times 10^{-1}$, and the stored result $(-0.12426 \times 10^{-1})$ can be written $\bar{s}_2 = \bar{y}_1 z_1 + \bar{y}_2 z_2$ where

$$\begin{aligned}\bar{y}_1 &= 0.1608622315 &= \bar{y}_1(0.12426/0.1242660), \\ \bar{y}_2 &= -0.1185119926 &= \bar{y}_2(0.12426/0.1242660).\end{aligned}$$

The final product $y_3 z_3$ is $-0.3002364904 \times 10^{-2}$, and the stored result $(-0.30023 \times 10^{-2})$ can be written $\bar{y}_3 z_3$ where $\bar{y}_3 = y_3(0.30023/0.3002364904) = 0.9821387679 \times 10^{-1}$. The final sum $s_3 = \bar{s}_2 + \bar{y}_3 z_3$ is $-0.1542830 \times 10^{-1}$, and the stored inner product $(-0.15428 \times 10^{-1})$ can be written \tilde{y}^*z where

$$\tilde{y}_1 = 0.1608591034 = \bar{y}_1 \phi, \quad \text{whereas } y_1 = 0.16087,$$

$$\tilde{y}_2 = -0.1185096878 = \bar{y}_2 \phi, \quad \text{whereas } y_2 = -0.11852,$$

$$\begin{aligned}\tilde{y}_3 &= 0.9821196701 \times 10^{-1} = \bar{y}_3 \phi, \quad \text{whereas } y_3 = 0.98216 \times 10^{-1}, \\ \phi &= 0.15428/0.1542830 \approx 1 - 1.944 \times 10^{-5}.\end{aligned}$$

The true inner product y^*z is $-0.154294921 \times 10^{-1}$. In this case there is no cancellation in the summation, and the relative error in the computed value is less than $\frac{1}{2}10^{-4}$. The backward error analysis has exhibited a vector \tilde{y} , close to y , such that

$$fl(y^*z) = \tilde{y}^*z.$$

A different, less systematic, rearrangement of the errors in the additions could produce another vector \hat{y} , even closer to y than \tilde{y} , such that for it too

$$fl(y^*z) = \hat{y}^*z.$$

2.6.2. The General Case

We now turn to a formal backward error analysis of $fl(x^*y)$. Computer addition is not associative and so the order in which the summation is organized affects the output. The results are a little bit simpler when $\sum x_i y_i$ is calculated with i running from n down to 1. To be definite we shall obtain a vector \tilde{x} such that

$$fl(x^*y) = \tilde{x}^*y,$$

but, of course, the error could be shared between x and y .

Formal error analyses are rather dull, but if you have never studied one we urge you to work through this one line by line.

2.6.3. The Algorithm

$s_n = 0$; then for $j = n, n-1, \dots, 2, 1$,

$$p_j = fl(x_j y_j), \quad (2.4)$$

$$s_{j-1} = fl(p_j + s_j). \quad (2.5)$$

The only properties required of the arithmetic system are

$$fl(\alpha\beta) = (1 - \mu)\alpha\beta \quad \text{where } |\mu| < \epsilon, \quad (2.6)$$

$$fl(\alpha + \beta) = (1 - \rho)\alpha + (1 - \sigma)\beta \quad \text{where } |\rho| < \epsilon, |\sigma| < \epsilon. \quad (2.7)$$

In our simple model $\rho = \sigma$, but no great advantage accrues from that property in this application. Moreover, most computers (with or without a guard digit) satisfy (2.7).

2.6.4. Notation

$x_j^{(k)}$ denotes the result of multiplying x_j by k factors of the form $(1 - \tau)$ where $|\tau| < \epsilon$. Thus

$$|x_j^{(k)} - x_j|/|x_j| < (1 + \epsilon)^k - 1 \approx k\epsilon \quad (\text{if } k\epsilon < 0.01). \quad (2.8)$$

2.6.5. Analysis

Hold tight and watch the superscripts.

$$\begin{aligned}
 p_n &= (1 - \mu_n)x_n y_n \\
 &\equiv x_n^{(1)} y_n, \quad \text{by (2.4) and (2.6),} \\
 s_{n-1} &= p_n, \quad \text{since } s_n = 0, \\
 p_{n-1} &= (1 - \mu_{n-1})x_{n-1} y_{n-1} \equiv x_{n-1}^{(1)} y_{n-1}, \quad \text{defining } x_{n-1}^{(1)}, \\
 s_{n-2} &= (1 - \rho_{n-1})x_{n-1}^{(1)} y_{n-1} + (1 - \sigma_{n-1})x_n^{(1)} y_n, \quad \text{by (2.5) and (2.8),} \\
 &\equiv x_{n-1}^{(2)} y_{n-1} + x_n^{(2)} y_n, \quad \text{defining } x_{n-1}^{(2)} \text{ and } x_n^{(2)}, \\
 p_{n-2} &= (1 - \mu_{n-2})x_{n-2} y_{n-2} = x_{n-2}^{(1)} y_{n-2}, \\
 s_{n-3} &= (1 - \rho_{n-2})x_{n-2}^{(1)} y_{n-2} + (1 - \sigma_{n-2})s_{n-2}, \quad \text{by (2.5) and (2.7),} \\
 &\equiv x_{n-2}^{(2)} y_{n-2} + x_{n-1}^{(3)} y_{n-1} + x_n^{(3)} y_n, \quad \text{defining the } x_i^{(k)}, \\
 p_{n-3} &= (1 - \mu_{n-3})x_{n-3} y_{n-3} \equiv x_{n-3}^{(1)} y_{n-3}, \quad \text{defining } x_{n-3}^{(1)}, \\
 s_{n-4} &= (1 - \rho_{n-3})x_{n-3}^{(1)} y_{n-3} + (1 - \sigma_{n-3})s_{n-3}, \\
 &\equiv x_{n-3}^{(2)} y_{n-3} + x_{n-2}^{(3)} y_{n-2} + x_{n-1}^{(4)} y_{n-1} + x_n^{(4)} y_n, \\
 &\dots \\
 s_{j-1} &= x_j^{(2)} y_j + x_{j+1}^{(3)} y_{j+1} + \dots + x_n^{(n-j+1)} y_n \quad (\text{proof by induction}) \\
 &\dots \\
 p_1 &= (1 - \mu_1)x_1 y_1 \equiv x_1^{(1)} y_1, \\
 s_0 &= (1 - \rho_1)x_1^{(1)} y_1 + (1 - \sigma_1)s_1, \\
 &\equiv x_1^{(2)} y_1 + x_2^{(3)} y_2 + \dots + x_{n-1}^{(n)} y_{n-1} + x_n^{(n)} y_n, \\
 &\equiv \tilde{x}^* y, \quad \text{defining } \tilde{x}.
 \end{aligned}$$

Moreover,

$$|\tilde{x}_i - x_i|/|x_i| < (1 + \epsilon)^{i+1} - 1 \approx (i + 1)\epsilon \quad (\text{if } n\epsilon < 0.01).$$

This completes the backward analysis, but some further comments are in order.

1. No mention has been made of the error in s_0 . In general no bound can be placed on the relative error because three or more additions are involved. For the absolute error,

$$|s_0 - x^* y| = |(\tilde{x} - x)^* y| < [(1 + \epsilon)^n - 1] \|x\| \|y\| \approx n\epsilon \|x\| \|y\|.$$

The factor n reflects the generality of the result rather than the behavior of the error.

2. If there is uncertainty in the elements of x (or y) of at least n units in the last place, then this uncertainty dominates the effects of roundoff and we say that the computed value is as good as the data warrants.
3. \tilde{x} is close to x not just in norm but element by element.
4. When the elements of x and y are known to decrease in absolute value ($|x_i| > |x_{i+1}|$), it is worthwhile to sum from n down to 1.

2.6.6. Matrix–Vector Products

It is a corollary of the inner product analysis that

$$fl(Ay) = \tilde{A}y,$$

where $\tilde{a}_{ij} = a_{ij}$ if $a_{ij} \neq 0$ and otherwise

$$|\tilde{a}_{ij} - a_{ij}|/|a_{ij}| < (1 + \epsilon)^n - 1.$$

If the indices are taken in decreasing order, then

$$|\tilde{a}_{ij} - a_{ij}|/|a_{ij}| < (1 + \epsilon)^{j+1} - 1.$$

If the elements of A are uncertain by n units in the last place and $n\epsilon < 0.01$, then the computed product is as good as the data warrants.

2.6.7. Higher Precision Accumulation of Inner Products

It is possible to form the products $p_i = fl(x_i y_i)$ exactly so that $\mu_i = 0$ in the foregoing analysis. In the 1960s on the IBM 360 and 370 machines the cost to obtain the long product of two short “words” (numbers with approximately six decimal digits) was no greater than to obtain the short (truncated) product. On CDC machines two separate operations were required to get the exact product of two single-precision (14 decimal) numbers, and so the cost was doubled.

The extra accuracy is worthless unless these products are summed in extra precision. This can be done and the result is often, but not always, the exact inner product. Nevertheless, the result must be stored and so one single precision rounding error will be made. Note that this calculation requires no extra storage; the double length work is done in the arithmetic registers.

What are the rewards of this extra accuracy?

1. Often, but not always, $fl(x^*y) = x^*y(1 - \rho)$, $|\rho| < \epsilon$, independent of n .
2. Always $fl(x^*y) = \tilde{x}^*y$ with $|\tilde{x}_i - x_i| < \epsilon|x_i|$, provided $n\epsilon < 0.01$.
3. Always $|fl(x^*y) - x^*y| < n\epsilon^2\|x\|\|y\| < \epsilon\|x\|\|y\|$, provided $n\epsilon < 0.01$.

In other words the factor n is removed from the error bounds and this helps construction of simple, even elegant, error analyses. However, the absolute error itself is not reduced by a factor n because the single precision result is already more accurate than our worst-case analysis indicates. The upshot is that double precision accumulation of inner products was usually practiced only when the extra cost is small, e.g., less than 5%.

In most of the mini and micro computers of the 1980s the registers in the arithmetic unit are longer than cells in the main store. Some of the benefits described above occur automatically but, on the bad side, unintentional storage of intermediate quantities will change the result of a calculation. See the remarks preceding formula (2.1).

IBM produces a system called ACRITH which computes dot products exactly (before storage) and uses this as a basis for as many computations as possible. See [Bleher et al., 1987].

2.7. Can Small Eigenvalues Be Found with Low Relative Error?

The answer to the question is that we have no right to expect such accuracy in general, but if both the matrix representation is right and the numerical method is right then all eigenvalues can be obtained with low relative error. Such accuracy is well worth having.

All good numerical methods (for the spectrum of A) produce numbers which are exact eigenvalues of a matrix $A + W$ close to the original matrix A . This means that $\|W\|$ is small (like roundoff) compared with $\|A\|$. Fact 1.11 in Chapter 1 ensures that the absolute error in each eigenvalue is bounded by $\|W\|$. This result is the best that can be hoped for, in general, as the case $W = \epsilon I$ reveals. This is bad news for the accurate computation of the smallest eigenvalue (when it is tiny compared with $\|A\|$). Consider

$$A_1 \approx \begin{bmatrix} 1 + 2\gamma & 1 \\ 1 & 1 - 2\gamma \end{bmatrix}$$

when $\gamma^2 \approx \epsilon$, the unit roundoff. Add $3\gamma^2 I$ to A_1 and $\lambda_1[A]$ changes from $-2\gamma^2$ to γ^2 . We say that λ_1 is very poorly determined by A_1 , and no numerical method can determine λ_1 accurately without using extra precision. Small

relative changes to any element of A_1 provoke large relative changes in λ_1 . So much for the general case.

Now consider

$$A_2 \approx \begin{bmatrix} 2 & \gamma^2 \\ \gamma^2 & -2\gamma^2 \end{bmatrix}$$

which has approximately the same spectrum as A_1 . Note that if $3\gamma^2 I$ is added to A_2 a huge relative change is made in the (2, 2) element, precisely the same as the relative change in $\lambda_1[A_2]$. What is special about A_2 is that small relative perturbations of the elements lead to small relative changes in both the eigenvalues. In fact, A_2 is an extreme example of a graded matrix. A less bizarre specimen is

$$A_3 = \begin{bmatrix} 1 & 10 & 10 & 0 & 0 \\ 10 & 10 & 10^2 & 10^2 & 0 \\ 10 & 10^2 & 10^2 & 10^3 & 10^3 \\ 0 & 10^2 & 10^3 & 10^3 & 10^4 \\ 0 & 0 & 10^3 & 10^4 & 10^4 \end{bmatrix}.$$

Such a matrix can arise when a generalized eigenvalue problem is reduced to standard form $(A - \lambda I)z = 0$.

It is legitimate to ask of a method that it find small eigenvalues accurately whenever this is possible. The Jacobi algorithm described in Chapter 9 can do so.

A whole new line of work has opened up with the realization that when a tridiagonal matrix T may be written as $T = L\Omega L^*$ with L bidiagonal and Ω diagonal with entries ± 1 , then the small eigenvalues of T are usually determined to high relative accuracy by L , not T .

For recent work on this problem see Chapter 7 at the end of Notes and References.

2.8. Software

There are now available a few high-quality packages of programs which compute eigenvalues and/or eigenvectors of matrices of various types including real symmetric matrices, either full or tridiagonal ($a_{ij} = 0$, $|i - j| > 1$). These programs can be invoked in the same way as the elementary functions such as SIN and EXP. The number of input parameters has inevitably increased with the complexity of the task, but the user needs very little knowledge of numerical methods, mainly because of the excellent documentation.

These programs are based on [Wilkinson and Reinsch, 1971],⁷ which is a collection of procedures written in the language Algol 60 together with pertinent comments on method and the all-important computational details. This collection was edited by J. H. Wilkinson and C. Reinsch, the acknowledged leaders in the field of matrix computations at the time, but it represents a remarkably cooperative effort by many others working on these problems. What is more, all of the programs had been published previously and almost all had been modified and improved in the light of usage. (That fact is quite puzzling; it is rare indeed for mathematicians to publish regular improvements on their proofs—the aim is to get them right the first time!) The difficulties in turning a good method into a good program are not easy to identify, and everybody involved was surprised at the length of time required to reach the proper level of reliability and efficiency.

2.8.1. Basic Linear Algebra Subprograms (BLAS)

In 1973 Hanson, Krogh, and Lawson proposed the use of standard names and calling conventions for routines that computed inner products, the adding of a multiple of one vector to another, or other standard operations with vectors. The idea caught on and was adopted in the software package LINPACK for performing various tasks connected with solving linear equations in 1979; see [Lawson, Hanson, Kincaid, and Krogh, 1979]. By now the names of these standard operations have become an informal enhancement to FORTRAN that hides most inner loops and provides a clearer description of algorithms. A good reference is Dodson and Lewis (1985). Efficient implementations of the BLAS,⁸ as they came to be called, were provided for each major type of computer. This permitted FORTRAN programs to be both portable and efficient, a very desirable outcome.

The next step was to have common names and calling conventions for standard matrix–vector operations such as solving triangular systems or adding Ax to y . The impetus for these so-called Level 2 BLAS was the efficient use of vector computers.

Thus the differences between rival vector computer architectures could be accommodated in the implementation of these routines, and the user would not have to change his higher level algorithm to suit the target machine.

However, vector and parallel computers demand different representations of matrices for efficient computation. The major choice is between a matrix as

⁷Hereafter called the Handbook.

⁸A common way of saying that one is not feeling very energetic is to say “I am feeling blah” or “I have got the blahs.” I would say that the BLAS cure the blahs.

a sequence of columns or as a block matrix whose entries are (sub) matrices. Such fundamental differences between computer architectures has spurred the creation of Level 3 BLAS that perform standard matrix-matrix operations. See [Dongarra et al., 1990].

If the vectors and matrices are of order n , then the original BLAS perform $O(n)$ scalar arithmetic operations, the Level 2 BLAS are $O(n^2)$, and the Level 3 are $O(n^3)$ procedures.

2.8.2. Systems and Libraries

MatlabTM⁹ designed by Cleve Moler and sold by The MathWorks, Inc., has been greeted with delight by scientists and engineers and also by those who work on the development of numerical methods for use by the public. The only data type in MatlabTM is matrix. MatlabTM provides an environment in which sophisticated calculations may be performed as naturally and painlessly as possible.

EISPACK [Garbow, Boyle, Dongarra, and Moler, 1977] is a collection written in FORTRAN 66, and it was the successor to the *Handbook for Automatic Computation*, Vol. II (1971), edited by Wilkinson and Reinsch (see annotated bibliography).

LAPACK [Anderson et al., 1995] succeeds EISPACK. It is written in FORTRAN 77 and was first released in 1992. It covers far more than eigenvalue tasks and is still developing, with new releases published from time to time. One of its purposes is to exploit BLAS 3 subroutines.

Another useful, but large, collection is NETLIB, which is available through <http://www.netlib.org>.

2.9. Alternative Computer Architecture

2.9.1. Vector Computers

Scientific computation seems to be dominated by operations on vectors, in particular, the formation of linear combinations $\alpha x + y$ and the evaluation of inner products y^*x . As computations grow in ambition the vectors grow in length (i.e., dimension). Vector computers have special hardware (pipelined registers) that greatly facilitates these operations; for example, one element of the output vector can be produced at each so-called minor cycle of the computer (at 10^8 per second). The price for this desirable feature is that

⁹Matlab is a registered trademark of The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01700, USA, (508) 647-7001.

there is a severe start-up overhead cost for each vector operation. This start-up penalty cancels the rapidity of the subsequent execution for small vectors ($n < 100$). Nevertheless, for large sparse problems ($n > 100$) the gains in speed are impressive, and methods should be examined for the ease with which they can be “vectorized.”

2.9.2. Parallel Computers

Computer systems which can make simultaneous use of several, or many, arithmetic units or processors are called parallel computers. They vary in the degree of independence of the processors and the structure of the memory.

By the late 1980s several parallel computers were on the market: Intel's Hypercube (with 32, 64, or 128 processors), ALLIANT, even the CONNECTION MACHINE. The life expectancy of these machines is short.

Parallel computation is turning into a discipline in its own right. In principle it might be necessary to employ totally new methods for eigenvalue problems, and the current favorites that were designed for serial computers (one arithmetic operation at a time) may be put to rest. Will the contents of this book become obsolete by 2000?

A little reflection suggests that the question is misplaced. This book is concerned mainly with the mathematical properties of computable approximations to invariant subspaces and eigenvalues, their accuracy, and sensitivity to roundoff errors. These properties are to a great extent independent of the way the approximations are computed. For example, the undoubted impact of new computer architecture may be felt in the implementation of the BLAS (see section 2.8.2) and is less evident in the higher level algorithm that employs the BLAS. Parallel computation has focused attention on the central importance of data movement to efficient execution. Most of the information in this book is independent of the way the matrix is stored.

Until I have more experience with eigenvalue extraction from large matrices on parallel machines, it seems best to keep silent on this topic.

Counting Eigenvalues

3.1. Triangular Factorization

It is easy to solve triangular systems of equations because the unknowns can be calculated one by one provided only that the equations are taken in the proper order. For this reason it is an important fact that most square matrices can be written as a product of triangular matrices as shown in Figure 3.1.

$$\begin{bmatrix} 4 & 12 & 16 \\ 8 & 26 & 28 \\ -4 & -6 & -27 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 4 & & \\ & 2 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 4 \\ & 1 & -2 \\ & & 1 \end{bmatrix}$$

B L D U

Zero elements have been suppressed for clarity.

The leading principal submatrices of B are

$$B_1 = [4], \quad B_2 = \begin{bmatrix} 4 & 12 \\ 8 & 26 \end{bmatrix}, \quad B_3 = B.$$

FIG. 3.1. *Triangular factorization.*

For theoretical work it is convenient to separate out D and U from the product (DU) .

The standard Gauss elimination technique for solving systems of linear equations is best thought of as a process for computing the triangular factorization of the coefficient matrix. The multipliers which are needed during the elimination are the elements of the L -factor, and the result is DU (Exercise 3.1.1). By means of this factorization the direct solution of a full system

$Bx = b$ is reduced to the solution of two triangular systems.

Algorithm for the solution of $Bx = b$.

1. $B = LDU$ (triangular factorization),
2. solve $Lc = b$ for c (forward substitution),
3. solve $(DU)x = c$ for x (back substitution).

That is all there is to it for small matrices.

There do exist matrices B for which step 1 fails. Such matrices are not necessarily strange or pathological. The simplest of them is

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

This example suggests that after rearranging A 's rows triangular factorization might be possible. Indeed whenever A is nonsingular such a rearrangement can be found, but it will destroy symmetry (Exercises 3.1.3 and 3.1.4).

In practice the best way to determine whether triangular factorization is possible is to try it and see whether or not the process breaks down. For theoretical work it is useful to know conditions under which factorization is guaranteed, and it is also convenient to normalize L and U by putting 1's on their diagonals.

The theorem behind triangular factorization employs the leading principal submatrices of B . These are denoted by B_j and are exhibited in Figure 3.1. Since D is not a symmetric letter, we will replace it by Δ from now on. Unfortunately U is a symmetric letter, but it will soon be replaced by U^* .

Theorem 3.1.1 (LDU theorem). *If $\det B_j \neq 0$ for $j = 1, 2, \dots, n - 1$, then unique normalized triangular factors L, Δ, U exist such that $B = L\Delta U$. Conversely, if $\det B_j = 0$ for some $j < n$ then the factorization may not exist, but even if it does, the factors are not unique.*

3.1.1. Remarks

1. $\det B = 0$ is permitted. Singular matrices of rank $n - 1$ may or may not permit triangular factorization.

2. If $A^* = A$ and $A = L\Delta U$ then $U = L^*$.
3. If A is positive definite then $A = L\Delta L^*$ and Δ is also positive definite. The factorization may be written

$$A = (L\Delta^{\frac{1}{2}})(L\Delta^{\frac{1}{2}})^* \equiv C^*C,$$

where $C \equiv \Delta^{\frac{1}{2}}L^*$ is called the Cholesky factor of A . Regrettably C is sometimes called the square root of A , but that term should be reserved solely for the unique positive definite symmetric solution X of $X^2 = A$.

4. If A is banded and $A = L\Delta L^*$ then L inherits A 's band structure; that is, if $a_{ij} = 0$ when $|i - j| > m$ then $l_{ij} = 0$ when $i - j > m$.

Example:

$$A = \begin{bmatrix} 4 & 4 & 0 \\ 4 & 6 & 2 \\ 0 & 2 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & & \\ 1 & 1 & \\ 0 & 1 & 1 \end{bmatrix}.$$

The number m is the half-bandwidth of A .

5. The block form of triangular factorization is often useful. There is much economy in the apt use of partitioned matrices; i.e., matrices whose elements are matrices. For example if B_{11} is invertible then

$$\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} I & O \\ B_{21}B_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} B_{11} & O \\ O & \hat{B}_{22} \end{bmatrix} \begin{bmatrix} I & B_{11}^{-1}B_{12} \\ O & I \end{bmatrix},$$

where

$$\hat{B}_{22} = B_{22} - B_{21}B_{11}^{-1}B_{12}.$$

Sometimes \hat{B}_{22} is called the Gauss transformation of B_{22} , sometimes the Schur complement of B_{11} .

3.1.2. Cost

Triangular factorization is such a useful tool that it warrants further consideration. Is it expensive? Is it reliable? The answers come in this and the next section.

Storage. For full general matrices an n -by- n array can be used to hold L , Δ , and U . In the absence of interchanges the bandwidth of a matrix is preserved. Within the band, “fill-in” may occur. For A 's with half-bandwidth m a conventional n -by- $(m+1)$ array is often used for L and Δ . For large sparse matrices with a peculiar pattern of nonzero elements a special data structure may be needed.

TABLE 3.1
Arithmetic operations.

	General, n -by- n , dense	Symmetric, n -by- n , half-bandwidth m
<i>Divs</i>	$\frac{1}{2}n(n - 1)$	$m(n - m) + \frac{1}{2}m(m - 1)$
<i>Mults and adds</i>	$\frac{1}{6}n(n - 1)(2n - 1)$	$\frac{1}{2}m(m + 1)[(n - m) + \frac{1}{3}(m + 2)]$
$n \rightarrow \infty$	$\frac{1}{3}n^3$	$\frac{1}{2}m(m + 1)n$

Table 3.1 exhibits the number of arithmetic operations required in two important cases. The reader who has never done such a count is urged to verify the numbers in the table (by writing out the algorithm and inspecting the inner loop). The dramatic change from an $O(n^3)$ process (the general case) to an $O(n)$ process (symmetric matrices with narrow band) shows that the sparsity structure plays a vital role in the efficiency of algorithms. When A is tridiagonal the computation of $A^{-1}u$ costs little more than the computation of Au !

Storage costs show the same sort of advantage for matrices of narrow bandwidth. The problem of fill-in has received a great deal of attention and has become a special field known as *sparse matrix technology*. The “fill” in any matrix transformation is the set of those elements initially zero, which become nonzero in the course of the transformation. Two good references for sparse matrix computations are [George and Liu, 1980] and [Duff, Erisman, and Reid, 1987].

Exercises on Section 3.1

- 3.1.1. Carry out the standard Gauss elimination process on the matrix in Figure 3.1 and verify that L does hold the multipliers and the resulting triangular matrix is ΔU .
- 3.1.2. By equating the (i, j) elements on each side of $B = L\Delta U$ by columns obtain the Crout algorithm which generates the elements of L , Δ , and U directly without computing any intermediate reduced matrices $B^{(j)}$, $j = 1, \dots, n - 1$.
- 3.1.3. Show that there are nonsingular matrices A such that no symmetric permutation of A , i.e., PAP^* , permits triangular factorization.

- 3.1.4. Exhibit one nondiagonal 3-by-3 matrix of rank two which permits triangular factorization and another which does not. Hint: work backward.
- 3.1.5. Show that if $A = L\Delta U$ then $U = L^*$. Recall that $A^* = A$.
- 3.1.6. Show that $A = L\Delta L^*$ for all positive definite A . One approach is to use a 2-by-2 block matrix and use induction. What about the semidefinite case when A has rank $n - 1$?
- 3.1.7. Compute the Cholesky factorization of

$$\begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{bmatrix}.$$

Use a calculator if possible.

- 3.1.8. Prove that if $a_{ij} = 0$ when $|i - j| > m$ then $l_{ij} = 0$ when $i - j > m$.
- 3.1.9. Let $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ where B_{11} and B_{22} are n by n . Suppose that B_{11} is nonsingular. Find a formula for $\det B$ as a product of two determinants of order n . Find another formula for the case when B_{11} is singular but B_{22} is not.
- 3.1.10. Verify the multiplication count for triangular factorization of a full matrix. What is the count for a symmetric matrix assuming that full advantage is taken of symmetry?
- 3.1.11. Verify the operation count for a symmetric matrix of half-bandwidth m .
- 3.1.12. Show the fill-in during triangular factorization of a positive definite matrix with the indicated nonzero elements.

$$\left[\begin{array}{ccc|cc} x & x & & x & \\ x & x & x & & x \\ x & x & x & & x \\ x & x & & & x \\ \hline x & & & x & x \\ & x & & x & x \\ & & x & & x \\ & & & x & x \end{array} \right].$$

- 3.1.13. Use induction and remark 5 of section 3.1 to prove the first assertion of the LDU theorem. Exhibit 2-by-2 matrices which have more than one triangular factorization.

3.2. Error Analysis of Triangular Factorization

The triangular factorization of some well-conditioned matrices can be ruined by the roundoff errors in two or three places in the calculation.

The decomposition is so useful that it is worthwhile to understand how it can fail. Fortunately a complete error analysis is not needed to reveal when and how disaster can strike. It suffices to look carefully at the first step in the reduction process. Let \mathbf{A} be written in partitioned form

$$\mathbf{A} = \begin{bmatrix} \alpha & \mathbf{c}^* \\ \mathbf{c} & \mathbf{M} \end{bmatrix}, \quad \mathbf{M} \text{ is } (n-1) \text{ by } (n-1). \quad (3.1)$$

Using the 2-by-2 block factorization shown in remark 5 of section 3.1, we get

$$\mathbf{A} = \begin{bmatrix} 1 & \mathbf{o}^* \\ \mathbf{b} & 1 \end{bmatrix} \begin{bmatrix} \alpha & \mathbf{o}^* \\ \mathbf{o} & \mathbf{A}^{(1)} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{b}^* \\ \mathbf{o} & 1 \end{bmatrix}, \quad \mathbf{b} = \mathbf{c}/\alpha, \quad (3.2)$$

where

$$\mathbf{A}^{(1)} = \mathbf{M} - \mathbf{b}\mathbf{a}\mathbf{b}^* = \mathbf{M} - \mathbf{c}\mathbf{c}^*/\alpha, \quad (3.3)$$

or, descending to the element level

$$a_{ij}^{(1)} = a_{i+1,j+1} - b_i \alpha b_j, \quad i, j = 1, \dots, n-1. \quad (3.4)$$

$\mathbf{A}^{(1)}$ is called the *first reduced matrix* in the process.

At the next step, the same reduction is performed on $\mathbf{A}^{(1)}$. To see the role of roundoff, let \mathbf{b} and $\mathbf{A}^{(1)}$ now denote the quantities actually stored in the computer and suppose that, by some stroke of luck, each $a_{ij}^{(1)}$ is computed with a minimal relative error (see the model of arithmetic in Chapter 2 for more details). Thus,

$$\begin{aligned} a_{ij}^{(1)} &= fl(a_{i+1,j+1} - b_i \alpha b_j) \\ &= (a_{i+1,j+1} - b_i \alpha b_j)(1 - \rho_{ij}), \quad |\rho| < \epsilon. \end{aligned} \quad (3.5)$$

Here ϵ is the unit roundoff. It is convenient to rewrite $1 - \rho_{ij}$ as $1/(1 + \eta_{ij})$. So $\eta_{ij} = \rho_{ij}/(1 - \rho_{ij}) \approx \rho_{ij}$. The key observation is that (3.5) can now be rewritten as

$$a_{ij}^{(1)} + b_i \alpha b_j = a_{i+1,j+1} - \eta_{ij} a_{ij}^{(1)}, \quad i, j = 1, \dots, n-1, \quad i \geq j. \quad (3.6)$$

In matrix notation (3.6) says

$$\begin{bmatrix} 1 & \mathbf{o}^* \\ \mathbf{b} & 1 \end{bmatrix} \begin{bmatrix} \alpha & \mathbf{o}^* \\ \mathbf{o} & \mathbf{A}^{(1)} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{b}^* \\ \mathbf{o} & 1 \end{bmatrix} \equiv \mathbf{A} - \mathbf{H}^{(1)}, \quad (3.7)$$

$$h_{i+1,j+1}^{(1)} = \eta_{ij} a_{ij}^{(1)}, \quad i, j = 1, \dots, n-1. \quad (3.8)$$

Roundoff error made the algorithm factor $\mathbf{A} - \mathbf{H}^{(1)}$ instead of \mathbf{A} . That is not satisfactory if $\mathbf{A}^{(1)}$ is huge compared with \mathbf{A} , even under the favorable assumption that each arithmetic step is done with minimal error. It is not hard to prove the converse without making any favorable assumptions, namely if $\|\mathbf{A}^{(1)}\| \approx \|\mathbf{A}\|$ then $\|\mathbf{H}^{(1)}\|$ is small, like roundoff, compared with $\|\mathbf{A}\|$. When all rounding errors are included, the expression for $h_{ij}^{(1)}$ includes a multiple of $b_\mu b_\nu a_{11}$ for $\mu \leq i, \nu \leq j$. But this simply makes $\|\mathbf{H}^{(1)}\|$ tiny compared with $\max(\|\mathbf{A}\|, \|\mathbf{A}^{(1)}\|)$. See Exercise 3.2.2.

At the next step, the computed quantities $\mathbf{b}^{(2)}$ and $\mathbf{A}^{(2)}$ turn out to be part of the decomposition not of $\mathbf{A}^{(1)}$ but of $\mathbf{A}^{(1)} - \hat{\mathbf{H}}^{(2)}$. And so on. At the end the factors \mathbf{L} and Δ are seen to satisfy

$$\mathbf{L} \Delta \mathbf{L}^* = \mathbf{A} - \mathbf{H}^{(1)} - \mathbf{H}^{(2)} - \dots - \mathbf{H}^{(n-1)} \equiv \mathbf{A} - \mathbf{H}, \quad (3.9)$$

where $\mathbf{H}^{(k)} \equiv \mathbf{O}_k \oplus \hat{\mathbf{H}}^{(k)}$ is tiny, like roundoff, compared with the larger of $\mathbf{A}^{(k)}$ and $\mathbf{A}^{(k-1)}$. If any of the $\|\mathbf{A}^{(k-1)}\|$ are huge compared with $\|\mathbf{A}\|$, then the algorithm has simply factored the wrong matrix.¹⁰

Note that small pivots, $\delta_k = a_{kk}^{(k)}$, may or may not provoke a large $\mathbf{A}^{(k+1)}$, and, likewise, large multipliers in $\mathbf{b}^{(k)}$ may or may not provoke large $\mathbf{A}^{(k+1)}$. These traditional scapegoats, whenever factorization fails, are simply not the relevant quantities. It is their outer products $(\mathbf{b}^{(k)} \mathbf{b}^{(k)*}) a_{kk}^{(k)}$ which matter.

Triangular factorization, without any pivoting, can indeed fail and a failure is always indicated by element growth, i.e., $\|\mathbf{A}^{(k)}\| \gg \|\mathbf{A}\|$ for some $k < n$. It is easy to monitor this growth, and so there is no need for failure to go undetected. The next section pursues this topic.

For positive definite matrices, $\|\mathbf{A}^{(k)}\| < \|\mathbf{A}^{(k-1)}\|$ and so triangular factorization is very stable (Exercise 3.2.5).

The use of pivoting to achieve a stable factorization is not central to our purposes, and the reader is referred to [Forsythe and Moler, 1967] for this information. One-sided pivoting spoils symmetry, and symmetric pivoting (using matching row and column interchanges) usually spoils the band structure of sparse matrices.

Exercises on Section 3.2

- 3.2.1. Follow the error analysis numerically on the given matrix using four-

¹⁰Refer to the notation list on the inside cover for a description of the symbol \oplus .

decimal floating point arithmetic (chopped). Exhibit L , Δ , and H .

$$A = \begin{bmatrix} 10^{-3} & 10 \\ 10 & 14 \end{bmatrix}.$$

- 3.2.2. If a calculator is available factorize the given matrices using four-decimal arithmetic. Then form the product $L\Delta L^*$ exactly and compare with the original matrix.

$$H = \begin{bmatrix} 1.000 & 0.5000 & 0.3333 & 0.2500 \\ 0.5000 & 0.3333 & 0.2500 & 0.2000 \\ 0.3333 & 0.2500 & 0.2000 & 0.1666 \\ 0.2500 & 0.2000 & 0.1666 & 0.1428 \end{bmatrix},$$

$$W = \begin{bmatrix} 5 & 1 & & \\ 1 & 3 & 1 & \\ & 1 & 1 & 1 \\ & 1 & -1 & 1 \\ & & 1 & -3 & 1 \\ & & & 1 & -5 \end{bmatrix}.$$

Next compute L and Δ in full precision and compare them with the four-decimal versions.

- 3.2.3. Prove that if B is nonsingular there always is a permutation matrix P , possibly many, such that PB permits triangular factorization. Hint: first show that a row interchange can always be made at each step in the factorization to ensure that it does not break down. Second (this is harder) show why it is permissible to pretend that all the interchanges have been done in advance before factorization begins.
- 3.2.4. By Sylvester's inertia theorem (Fact 1.6) show that $A^{(k)}$ is positive definite if $A^{(k-1)}$ is positive definite.
- 3.2.5. Let A be positive definite. Show that the k th pivot $\delta_k = a_{kk}^{(k)} \leq a_{kk}^{(k-1)}$ for all k . Use the result of Exercise 3.2.4 to deduce that $\|A^{(k)}\| \leq \|A^{(k-1)}\|$.

3.3. Slicing the Spectrum

There is an elegant way to determine the number of A 's eigenvalues that are less than any given real number σ . Since the result is an integer, the technique appears to offer deliverance from the machinations of roundoff error, and the extent to which this hope is justified will be examined below. The technique

has been used by theoretical physicists for many years, certainly since the early 1950s.

The method is a corollary of Sylvester's inertia theorem (Fact 1.6) which states the invariance of $\nu(W)$, the number of negative eigenvalues of W , under congruence transformations. Of great use is the fact that the technique is directly applicable to the general eigenvalue problem $A - \lambda M$.

Theorem 3.3.1. Suppose that $A - \sigma M$ permits triangular factorization $A - \sigma M = L_\sigma \Delta_\sigma L_\sigma^*$ where Δ_σ is diagonal, and suppose that the pair (A, M) has a full set of real eigenvectors. Then

$$\nu(A - \sigma I) = \nu(A - \sigma M) = \nu(\Delta_\sigma),$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ and λ_i is an eigenvalue of the pair (A, M) .

Proof. Since L_σ is unit lower triangular it is invertible and so $A - \sigma M$ is congruent to Δ_σ . By Theorem 15.3.2 (the simultaneous reduction of two quadratic forms), there is an invertible matrix F such that

$$F^*(A - \sigma M)F = \Lambda - \sigma I,$$

whence $A - \sigma M$ is congruent to $\Lambda - \sigma I$. The result follows from Sylvester's inertia theorem applied to the congruent diagonal matrices $\Lambda - \sigma I$ and Δ_σ . \square

On one hand $\nu(\Delta_\sigma)$ is simply the number of negative elements on Δ_σ 's diagonal. On the other hand $\nu(\Lambda - \sigma I)$ is the number of eigenvalues of the pencil (A, M) which are less than σ .

The computation of Δ_σ and $\nu(\Delta_\sigma)$ reveals how σ slices the spectrum into two pieces. We call $\nu(\Delta_\sigma)$, or $\pi(\Delta_\sigma)$ if you prefer to dwell on the positive, *the spectrum slicer* and show it at work in Example 3.3.1.

Example 3.3.1 (slicing the spectrum).

$$A = \begin{bmatrix} 1 & 2 & 1 & 3 \\ 2 & 3 & -2 & 0 \\ 1 & -2 & -1 & -7 \\ 3 & 0 & -7 & 0 \end{bmatrix}, \quad M = I,$$

$$(A - 0) = \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 1 & 4 & 1 & \\ 3 & 6 & 1 & 1 \end{bmatrix} \cdot \text{diag} \begin{bmatrix} 1 \\ -1 \\ 14 \\ 13 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 & 3 \\ & 1 & 4 & 6 \\ & & 1 & 1 \\ & & & 1 \end{bmatrix},$$

$$(A - 2) = \begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ -1 & 0 & 1 & \\ -3 & 1.2 & 2 & 1 \end{bmatrix} \cdot \text{diag} \begin{bmatrix} -1 \\ 5 \\ -2 \\ 7.8 \end{bmatrix} \cdot \begin{bmatrix} 1 & -2 & -1 & -3 \\ & 1 & 0 & 1.2 \\ & & 1 & 2 \\ & & & 1 \end{bmatrix}.$$

Count: $\nu(\Delta_0) = 1, \nu(\Delta_2) = 2$.

Conclusion: $\lambda_1[A] < 0 < \lambda_2[A] < 2 < \lambda_3[A] \leq \lambda_4[A]$.

3.3.1. The Tridiagonal Case

In this important application it turns out that the elements of L and Δ need not be stored. Only one cell of workspace is needed. (See Exercise 3.3.5.)

For future reference we give the procedure in detail. Let $\alpha_1, \dots, \alpha_n$ be the diagonal elements, let β_1, \dots, β_n be the off-diagonal elements, let δ be the extra cell, and let σ be the sample point (or origin shift). The purpose is to compute $\nu = \nu[T - \sigma]$ where T is the tridiagonal.

Initialize: $\nu \leftarrow 0; \delta \leftarrow \alpha_1 - \sigma$; if $\delta < 0$ then $\nu \leftarrow 1$;

Loop: for $k = 1, \dots, n - 1$ repeat

$$\begin{cases} \text{if } \delta = 0 \text{ then } \delta \leftarrow \epsilon(\beta_k + \epsilon), \\ \delta \leftarrow (\alpha_{k+1} - \sigma) - \beta_k(\beta_k/\delta), \\ \text{if } \delta < 0 \text{ then } \nu \leftarrow \nu + 1. \end{cases} \quad (3.10)$$

Actually I prefer the following: If $\delta = 0$ then change σ slightly, and start again.

The expression $\beta_k(\beta_k/\delta)$ avoids possible overflows that can occur in evaluating β_k^2 before the division.

3.3.2. Accuracy of the Slice

For any symmetric A consider $A - \sigma$ as σ varies over the real numbers. The LDU theorem shows that the factorization fails to exist if and only if one or more of the leading principal submatrices, $A_k - \sigma$, is singular. By Cauchy's interlace theorem (section 10.1) the eigenvalues $\lambda_i^{(k)}$ of A_k interlace those of A_{k+1} . Note that $A_n = A$. Consequently there are $\sum_{k=1}^{n-1} k = n(n-1)/2$ values of σ , not

necessarily distinct, for which $\mathbf{A} - \sigma$ does not permit an $L\Delta L^*$ factorization. For σ in tiny intervals around each of these values the factorization is unreliable in the face of roundoff. To see why this is so let \mathbf{A} be partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{V} & \mathbf{C}^* \\ \mathbf{C} & \mathbf{M} \end{bmatrix} \quad (3.11)$$

and suppose that σ agrees with some $\lambda_j[\mathbf{V}]$ to p significant decimals. Then one of the reduced matrices encountered in the factorization process is (see Exercise 3.3.2)

$$\hat{\mathbf{M}} = \mathbf{M} - \sigma - \mathbf{C}(\mathbf{V} - \sigma)^{-1}\mathbf{C}^* \quad (3.12)$$

and, because σ is so close to an eigenvalue of \mathbf{V} (Exercise 3.3.3),

$$\begin{aligned} \|(\mathbf{V} - \sigma)^{-1}\| &> 10^p / |\lambda_j[\mathbf{V}]| \\ &> 10^p / \|\mathbf{A}\|. \end{aligned} \quad (3.13)$$

It is certainly possible to have $\|\mathbf{C}\| \approx \|\mathbf{A}\|$ and, without favorable cancellation,

$$\|\mathbf{C}(\mathbf{V} - \sigma)^{-1}\mathbf{C}^*\| \approx 10^p \|\mathbf{A}\|. \quad (3.14)$$

The memorable fact is that there are only $n(n-1)/2$ danger spots for σ (namely the eigenvalues of the principal submatrices $\mathbf{A}_j, j = 1, \dots, n-1$). The probability of failure in the triangular factorization of $\mathbf{A} - \sigma$ is low. Failure of the decomposition can be taken as a signal to choose a different value of σ . However, when an eigenvalue of \mathbf{A} coincides with an eigenvalue of a leading principal submatrix such a change in σ is not acceptable. Here is a real weakness in spectrum slicing. One remedy is to invoke block triangular factorization. One reason that this blemish has not received more attention is that in the most common situation (tridiagonal matrices) triangular factorization of a singular matrix can only fail if the corresponding eigenvector has a zero entry.

The computed factors \mathbf{L}_σ and Δ_σ satisfy

$$\mathbf{L}_\sigma \Delta_\sigma \mathbf{L}_\sigma^* = (\mathbf{A} - \sigma) - \mathbf{H}_\sigma \quad (\text{from section 3.2}) \quad (3.15)$$

and so

spectrum slicing is impervious to the roundoff errors in triangular factorization provided that

$$\nu(\mathbf{A} - \mathbf{H}_\sigma - \sigma) = \nu(\mathbf{A} - \sigma). \quad (3.16)$$

When no element growth occurs then H_σ is tiny relative to A and the slice is certainly as good as the precision of the calculation warrants. However, the two ν values in (3.16) may still be equal even when H_σ is not tiny. Thus inaccurate triangular factorization does not necessarily ruin the accuracy of the slice.

Conversely, let us note the implications of a wrong count. By the Weyl monotonicity theorem (section 10.3 or Fact 1.11)

$$|\lambda_j[A - H_\sigma] - \lambda_j[A]| < \|H_\sigma\|, \quad j = 1, \dots, n. \quad (3.17)$$

So if $\nu(A - H_\sigma - \sigma) \neq \nu(A - \sigma)$ then

$$\|H_\sigma\| > \min_j |\lambda_j[A] - \sigma|. \quad (3.18)$$

The precision with which eigenvalues can be located by slicing is least satisfactory when, for some i and j , $\lambda_i[A]$ and $\lambda_j[V]$ agree too closely. V is shown in (3.11).

Exercises on Section 3.3

- 3.3.1. Strictly speaking $\nu(A)$ denotes the number of negative elements in any diagonal matrix congruent to A . By using the spectral factorization of A show that $\nu(A)$ is the number of negative eigenvalues of A .
- 3.3.2. Establish (3.12) by invoking the uniqueness of triangular factorization.
- 3.3.3. Establish (3.13).
- 3.3.4. Let T be tridiagonal with k th row $(\dots 0, \beta_{k-1}, \alpha_k, \beta_k, 0 \dots)$. Write out the algorithm for factoring $T - \sigma = L \Delta L^*$, and then show how the elements of L can be eliminated to yield the algorithm given in this section.
- 3.3.5. Suppose that A and W are both tridiagonal except that $a_{1n} = a_{n1} \neq 0$, $w_{1n} = w_{n1} \neq 0$. Let DA and DW be n -vectors holding the diagonal elements; let EA and EW be n -vectors holding the other nonzero elements. Write an algorithm which computes $\nu(A - \sigma W)$ without using any more arrays, just six or seven extra simple variables.

3.4. Relation to Sturm Sequences

Let $\chi_j(\tau)$ denote the characteristic polynomial of the leading principal j -by- j submatrix of A . Thus, $\chi_1(\tau) = \tau - a_{11}$. Let $\chi_0(\tau) = 1$. By the Cauchy interlace theorem (section 10.1) the sequence $\{\chi_0, \chi_1, \dots, \chi_n\}$ is a *Sturm sequence*

of polynomials; that is, the zeros of consecutive χ 's interlace each other. A rather careful argument¹¹ then shows that the number of sign *agreements* in consecutive terms of the numerical sequence $\{\chi_j(\sigma), j = 0, 1, \dots, n\}$ equals the number of zeros of χ_n which are less than σ .¹²

For general A there is no cheap way to compute $\chi_j(\sigma)$ from the $\chi_i(\sigma), i < j$, but when A is tridiagonal there is a well-known three-term recurrence

$$\chi_{j+1}(\sigma) = (\sigma - a_{j+1,j+1})\chi_j(\sigma) - a_{j+1,j}^2\chi_{j-1}(\sigma), \quad (3.19)$$

which is also discussed in Chapter 7. This recurrence was used to count the number of eigenvalues less than σ in the trend-setting report by [Givens, 1954].

If $A - \sigma = L\Delta L^*$ (triangular factorization) then (Exercise 3.4.1)

$$\begin{aligned} (-1)^j \chi_j(\sigma) &= \delta_1 \cdots \delta_j = \det \Delta_j, & j = 1, \dots, n, \\ \delta_j &= -\chi_j(\sigma)/\chi_{j-1}(\sigma). \end{aligned} \quad (3.20)$$

Thus, sign agreements in the sequence $\{\chi_j(\sigma)\}$ correspond to negative values in the sequence $\{\delta_j\}$. However, the rational functions δ_j are far more sedate than the polynomials χ_j , and their use is to be preferred in finite precision computations. Problems with overflow and underflow recede when triangular factorization is used in place of (3.19).

The continual emphasis on Sturm sequences (i.e., (3.19)) delayed the application by numerical analysts of spectrum slicing to banded matrices and the general eigenvalue problem. Indeed some authors use the term Sturm sequence technique for both (3.10) and (3.19). It is true that the sequence of rational functions $\{\delta_0(\sigma), \dots, \delta_n(\sigma)\}$ does have interlacing zeros and is sometimes called a Sturm sequence. The attraction of (3.10) is that there are no overflow troubles and the attraction of (3.19) is that there are no divisions.

Exercises on Section 3.4

- 3.4.1. Derive (3.20) and then obtain from (3.19) a three-term recurrence for the δ_j which requires 1 division and no multiplications at each step.
- 3.4.2. For the example in the preceding section compute the Sturm sequence $\{\chi_j(\sigma)\}$ and contrast with $\{\delta_j\}$.

¹¹This argument is given in [Wilkinson, 1965, p. 300].

¹²A lot of research went into the vexed question of the right sign to be attributed to one, or even two, zero values of $\chi_j(\sigma)$. This led to such esoterica as *Gundelfinger's rule* which can be found in [Browne, 1930].

3.5. Bisection and Secant Methods

3.5.1. Bisection

If a half open interval $[\alpha, \beta]$ is known to contain at least one eigenvalue then the slicing techniques in section 3.3 can be used to find whether $[\alpha, (\alpha + \beta)/2]$ also contains one. The process can be repeated again and again to locate an eigenvalue to an accuracy limited only by the errors in the triangular factorization.

This idea has been implemented with great care in the program BISECT (p. 249 of the Handbook) but for tridiagonal matrices only.

Cost. One triangular factorization per slice. The operation counts are given in section 3.1.

Usage. Bisection is efficient when low accuracy is required (say three correct decimals) and also for refining approximate eigenvalues to get maximal accuracy. It is also good for finding a few eigenvalues of matrices with narrow bandwidth, particularly when all the eigenvalues in a given interval are required. Bisection can often deliver results correct to full working precision when asked to do so.

The technique is not recommended when more than $n/4$ eigenvalues are wanted or when the matrix is rather full.

3.5.2. The Secant Method

When an interval $[\alpha, \beta]$ has been found to contain a single eigenvalue λ then bisection is a rather primitive and slow way to pin down λ to high accuracy (say 10 correct decimals). The classical problem of finding the zero of a polynomial known to lie in a given interval has enjoyed a lot of attention for many years and some first-class algorithms have been developed.

For a mere $(n - 1)$ extra multiplications the factorization program can evaluate the polynomial itself because

$$(-1)^n \chi(\sigma) = \delta_1 \cdots \delta_n, \text{ (watch out for over/underflow),}$$

and so any variation of the secant method can be implemented. The order of convergence is 1.618, as against 1 for bisection; so secant methods should be faster than bisection for isolated eigenvalues. For tridiagonal matrices $\chi(\sigma)$ can be calculated directly from (3.19) but for fatter matrices triangular factorization is the natural approach, and the extra cost of forming $\delta_1 \cdots \delta_n$ is small. An important alternative to χ is discussed in section 3.6, namely the rational function $\delta_n(\sigma)$.

The formula for one step of the secant iteration may be written

$$\xi_{j+1} = \xi_j - \omega \chi(\xi_j)/[\xi_j, \xi_{j-1}]\chi, \quad (3.21)$$

where

$$[\alpha, \beta]\chi \equiv (\chi(\alpha) - \chi(\beta))/(\alpha - \beta) \quad (3.22)$$

is the first-order divided difference of χ . The value $\omega = 1$ gives the standard formula; $\omega = 2$ gives the more practical double secant iteration. See Exercise 3.5.4 for more details.

In good programs the secant formula is not used blindly; it provides one among other candidates for the next approximation. By making the program more complicated and checking for various difficult cases the performance of zero finders has become robust. Clearly, if the secant method's approximation lies outside the smallest interval known to contain λ , then it should not be used. What should take its place? That is an interesting question. It is tempting to return to bisection, but there are better solutions than that. The interested reader is referred to [Brent, 1973] and to [Bus and Dekker, 1975] for a full discussion of this topic.

Warning. Polynomial zero finders do not have at their disposal anything comparable to the spectrum slicer $\nu(\Delta_\sigma)$ and so they have to be very careful not to jump over two zeros (leaving no sign change). Eigenvalue codes should not be so cautious. See [Bathé and Wilson, 1976, Chapter 11] for the application of these ideas in computations with large matrices.

With tridiagonal matrices it is possible to obtain recurrences for evaluation of $\chi'(\sigma)$ and even higher derivatives. Thus the whole gamut of zero finders could be considered for finding λ . However, the recurrences do not generalize nicely to matrices with greater bandwidth and this avenue will not be explored.

Finally we remind ourselves that secant iterations, Newton iterations, and even sophisticated hybrid techniques like Zeroin can be very slow initially. Usually bisection is more powerful than these higher-order iterations until three-decimal accuracy has been obtained. Although it cannot always be estimated, the following criterion is useful:

while $(|\xi - \lambda_1| > |\lambda_2 - \lambda_1|/n)$ use bisection.

Exercises on Section 3.5

- 3.5.1. Assume that the secant iteration converges to λ . Find an expression which relates the error in one step to the product of the previous two errors. Then deduce that the order of convergence is $(1 + \sqrt{5})/2$.

- 3.5.2. Write a simple program which combines bisection with secant acceleration and test it on the examples given in this chapter. Compare your results with those from simple bisection and include the relative costs of one step in the two techniques.
- 3.5.3. Project. Take the algorithm BISECT in the Handbook (p. 249) and replace the Sturm sequence technique by triangular factorization of a banded matrix.
- 3.5.4. Let $\lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$ be zeros of χ . Let $\mu_1 \leq \mu_2 \leq \dots \leq \mu_{n-1}$ be zeros of χ' , the derivative of χ . If $\xi_{j-1} < \xi_j < \lambda_1$ show that for equation (3.21)
- if $\omega = 1$ then $\xi_{j+1} < \lambda_1$,
 - if $\omega = 2$ then $\xi_{j+1} < \mu_1$ (unpublished result of Kahan—difficult exercise).
- In neither case can ξ_{j+1} jump over *two* zeros of χ .

3.6. Hidden Eigenvalues

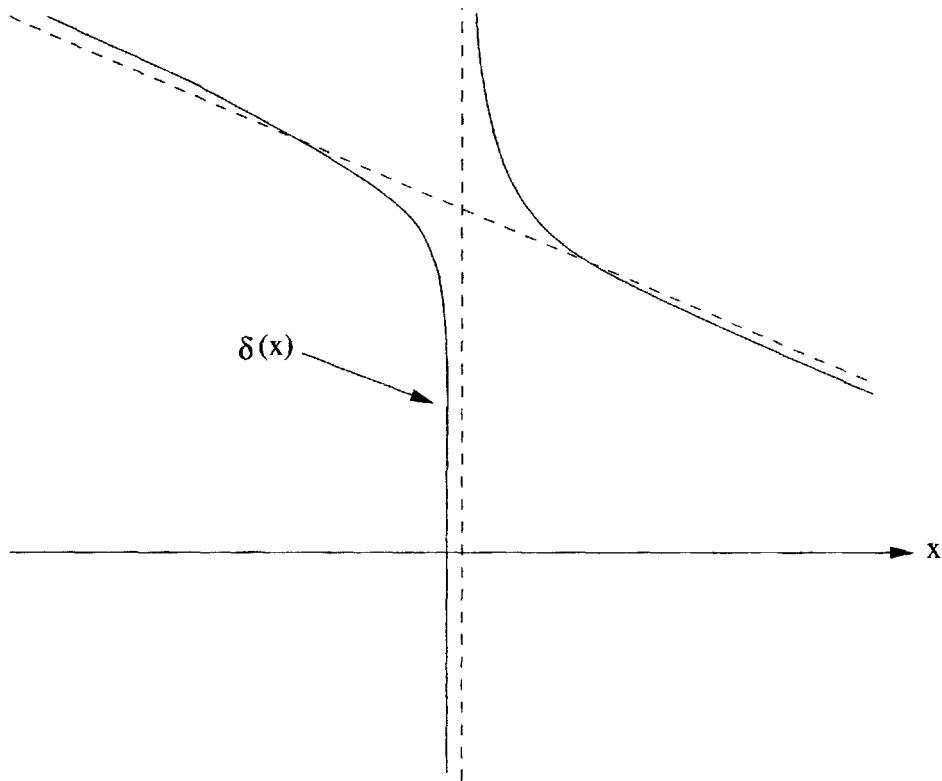
By (3.20) the final pivot $\delta_n(\sigma)$ in the factorization of $A - \sigma$ is equal to $-\chi_n(\sigma)/\chi_{n-1}(\sigma)$ and so has the same zeros as χ_n . Consequently the secant method, or any of its variations, can be applied to the rational function $\delta_n(\sigma)$ thereby avoiding the $(n-1)$ multiplications required to evaluate $\chi_n(\sigma)$ from the known pivots $\delta_i(\sigma)$, $i = 1, \dots, n$.

An interesting phenomenon can easily mar a straightforward implementation of this replacement of χ by δ_n . It can happen that λ is a well-isolated simple zero of χ_n and is also extremely close to a zero of $\chi_{n-1}, \chi_{n-2}, \chi_{n-3}$, etc. In other words $|\chi'_n(\lambda)|$ is large while $|\chi_{n-1}(\lambda)|$ is very small. The effect is that for almost all values of σ the pole of δ_n very near λ cancels out, or conceals, the zero at λ and persuades a good number of root finders that there is no treasure to be found near λ . Figure 3.2 gives a picture of a hidden zero.

Wilkinson constructed a beautiful example: the tridiagonal matrix W_{21}^- defined by

$$\begin{aligned} w_{ii} &= 11 - i && \text{for } i = 1, \dots, 21, \\ w_{i,i+1} &= w_{i+1,i} = 1 && \text{for } i = 1, \dots, 20. \end{aligned}$$

The largest eigenvalue λ_{21} ($= 10.746\dots$) agrees with the largest eigenvalue of the leading submatrix of order 20 in its first *fifteen* decimal digits. The graph of $\delta_{21}(\sigma)$ is close to $-(\sigma + 10)$ on all of the interval $[10, 11]$ except for a

FIG. 3.2. *A hidden zero.*

subinterval of width less than 10^{-13} at λ_{21} ! On many computer systems the critical subinterval is not detectable.

It must be emphasized that W_{21}^- is not a pathological matrix. If the algorithm in section 3.3 is run in reverse (for $k = n, \dots, 2$) then λ_{21} is not hidden from the new $\delta(\sigma)$ and is easily found. The point is that some rational functions, in practice, conceal their zeros from prying eyes.

Exercise on Section 3.6

- 3.6.1. Find the smallest value of n such that either $\lambda_1[W_n^-]$ or $\lambda_n[W_n^-]$ cannot be detected by evaluating δ_n on your computer system. Print out the zeros of $\chi_n, \chi_{n-1}, \chi_{n-2}$, etc.

3.7. The Characteristic Polynomial

If the coefficients of χ_A are given or can be obtained easily, then the matrix problem reduces to the classical task of computing some or all zeros of a polynomial and the reader is referred to [Brent, 1973], [Traub, 1964], or [Wilkinson, 1964] which treat that problem. However, if the matrix A is given then there is NO incentive to compute the coefficients of χ_A because the zeros of a polynomial are extraordinarily sensitive functions of these coefficients. If $n = 300$ (medium-order matrix) it is not clear how many hundred decimal digits in the coefficients are needed to determine all the zeros to two or three decimal places. We will not give further consideration to the coefficients of χ_A .

On the other hand $\chi_A(\sigma)$ can be evaluated satisfactorily by triangular factorization as described in section 3.5.

A fundamental constraint on all algorithms for computing eigenvalues may be mentioned here. Over a hundred years ago Galois proved that there cannot be a finite procedure, utilizing only basic arithmetic operations and the extraction of roots, that will yield a zero of an arbitrary fifth- (or higher) degree polynomial. However, the coefficients of χ_A can be computed from the elements of A by various finite schemes some of which are described in [Faddeev and Faddeeva, 1963]. It follows that, in the context of exact arithmetic, there can be no finite computer algorithm which will produce the eigenvalues of any given matrix if its order n is permitted to exceed four. Consequently, all eigenvalue programs contain somewhere an iterative component. A method is sometimes called *direct* if it employs a finite number of similarity transformations (perhaps none at all), but this distinction does not seem to be useful.

Simple Vector Iterations

The power method is no longer a serious technique for computing eigenvectors. Nevertheless, it warrants study because it is intimately related to current algorithms, and so a good grasp of it is helpful in understanding more complicated methods. In particular inverse iteration is a useful technique although it is not used now in quite the way it was originally envisaged. This chapter first looks at the methods theoretically, in the context of exact arithmetic, and then practically, in the context of limited precision.

The most important variation on inverse iteration is the Rayleigh quotient iteration (RQI, for short). We shall see that it converges for almost all starting approximations, however bad, and that ultimately—usually after two or three steps—the number of correct digits triples with each iteration.

4.1. Eigenvectors of Rank-One Matrices

There is a class of full symmetric matrices whose eigenvalues are easy to find, namely those of rank one.

Example 4.1.1 (a full matrix of rank one).

$$A = \begin{bmatrix} 8.41 & -6.09 & 3.19 \\ -6.09 & 4.41 & -2.31 \\ 3.19 & -2.31 & 1.21 \end{bmatrix} = \begin{bmatrix} 2.9 \\ -2.1 \\ 1.1 \end{bmatrix} \begin{bmatrix} 2.9 & -2.1 & 1.1 \end{bmatrix}.$$

Suppose that, unknown to the user, $A = vv^*$. Then the first step is to take any $x \neq o$ and compute

$$y \equiv Ax [= v(v^*x)]. \quad (4.1)$$

If $y = o$ then x is an eigenvector belonging to the eigenvalue 0. Otherwise y is an eigenvector belonging to eigenvalue (v^*v) because, using (4.1),

$$Ay = Av(v^*x) = v(v^*v)(v^*x) = y(v^*v). \quad (4.2)$$

The computer, human or not, does not know this yet. The second step is

1. compute $\mathbf{z} = \mathbf{A}\mathbf{y}$,
2. compute $\min(z_i/y_i)$ and $\max(z_i/y_i)$ over all i for which y_i is not 0 (nor tiny). All the ratios will be $\mathbf{v}^*\mathbf{v}$.

Of course, if \mathbf{A} is known to be of rank one then any nonzero column of \mathbf{A} will be a dominant eigenvector. Without prior knowledge only *two* matrix–vector products are needed to discover it. Any vector orthogonal to \mathbf{v} is an eigenvector with 0 eigenvalue (i.e., a null vector).

One might hope that one more matrix–vector product would yield the dominant eigenvector of rank-two matrices. Unfortunately that is not the case as Example 4.1.2 shows.

Example 4.1.2.

$$\mathbf{A} = \begin{bmatrix} -4 & 10 & 8 \\ 10 & -7 & -2 \\ 8 & -2 & 2 \end{bmatrix}, \quad \text{Rank}(\mathbf{A}) = 2.$$

Successive iterates, normalized to have the biggest element equal to one, are shown in Table 4.1.

TABLE 4.1
The power method.

<i>Step</i>	0	1	2	3	4
<i>Iterate</i>	1.0	1.0	0.1429	1.0	?
	1.0	0.0714	0.9286	0.5	?
	1.0	0.5714	1.0	0.1	?

4.2. Direct and Inverse Iteration

The special case of rank-one matrices is relevant to the general case because, for large k , a normalized multiple of \mathbf{A}^k is close to a rank-one matrix (Exercise 4.2.1). Thus it is only necessary to find $\mathbf{y} = \mathbf{A}^k\mathbf{x}$ and then check the

accuracy by comparing y and Ay . Fortunately it is not necessary to compute A^k explicitly since

$$A^5x = A(A(A(A(Ax)))).$$

The power method (PM). Pick a unit vector x_0 then

$$\text{for } k = 1, 2, 3, \dots \left\{ \begin{array}{l} 1. \text{ form } y_k = Ax_{k-1}, \\ 2. \text{ normalize, } x_k = y_k / \|y_k\|, \\ 3. \text{ test for convergence of } x_k. \end{array} \right.$$

Table 4.1 shows a few steps of the process.

4.2.1. Convergence

Recall our standard notation:

$$Az_i = z_i \lambda_i, \quad \|z_i\| = 1, \quad \text{and}$$

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n. \tag{4.3}$$

Note that $\|A\|$ is either $-\lambda_1$ or λ_n . We shall assume that it is the latter, just to be definite, and we shall ignore roundoff effects here.

The analysis is essentially two dimensional. At step k there is a unique plane containing x_k and the wanted eigenvector z_n . Let u_k be the unit vector in that plane which is orthogonal to z_n as shown in Figure 4.1. As the algorithm proceeds, the plane flaps on its one fixed axis z_n like an unlatched window. With the geometry established we can write

$$x_k = z_n \cos \theta_k + u_k \sin \theta_k \tag{4.4}$$

and $\theta_k \equiv \angle(x_k, z_k)$ is the error angle.

In some ways θ_k is a more natural measure of error than the usual $\|x_k - z_n\|$ ($= 2 \sin(\theta_k/2)$) which so often brings on unnecessary normalizations. We hope that the power sequence $\{x_0, x_1, x_2, \dots\}$ converges to z_n .

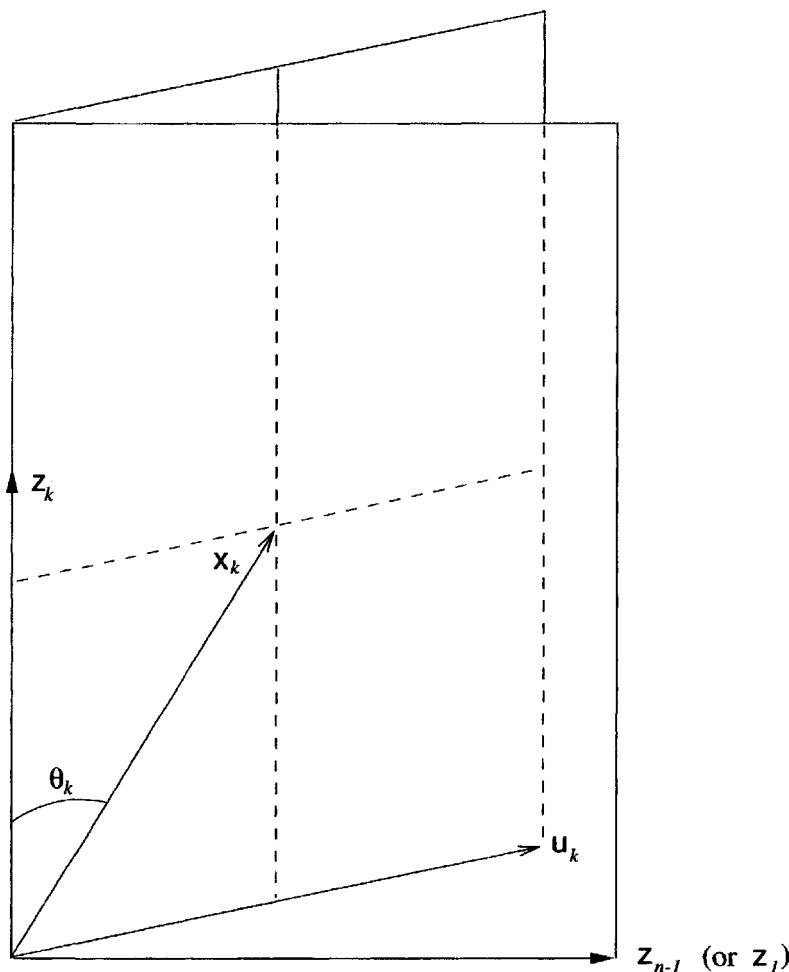


FIG. 4.1. Representation of \mathbf{x}_k .

Theorem 4.2.1. If λ_n is the unique dominant eigenvalue of \mathbf{A} and if $\mathbf{z}_n^* \mathbf{x}_0 \neq 0$ then, as $k \rightarrow \infty$, $\mathbf{x}_k \rightarrow \mathbf{z}_n$ linearly with convergence factor

$$\max\{\lambda_{n-1}/\lambda_n, |\lambda_1|/\lambda_n\}.$$

When convergence is linear, interest focuses on the *convergence factor* defined, in our problem, as $\lim_{k \rightarrow \infty} \theta_{k+1}/\theta_k$.

Proof. Consider the $(k+1)$ th step wherein $\mathbf{A}\mathbf{x}_k$ is formed. Premultiply (4.4) by \mathbf{A} and use (4.3) to find

$$\mathbf{A}\mathbf{x}_k = \mathbf{z}_n \lambda_n \cos \theta_k + \left(\frac{\mathbf{A}\mathbf{u}_k}{\|\mathbf{A}\mathbf{u}_k\|} \right) \|\mathbf{A}\mathbf{u}_k\| \sin \theta_k. \quad (4.5)$$

The key observation (see Exercise 4.2.4) is that $\mathbf{A}\mathbf{u}_k$, like \mathbf{u}_k , is orthogonal to \mathbf{z}_n . Since \mathbf{x}_{k+1} is a multiple of $\mathbf{A}\mathbf{x}_k$, (4.5) provides an orthogonal decomposition of \mathbf{x}_{k+1} . Compare (4.5) with (4.4) to obtain

$$\mathbf{u}_{k+1} = \mathbf{A}\mathbf{u}_k / \|\mathbf{A}\mathbf{u}_k\| \quad (4.6)$$

and

$$\tan \theta_{k+1} = \|\mathbf{A}\mathbf{u}_k\| \sin \theta_k / \lambda_n \cos \theta_k. \quad (4.7)$$

For all k , \mathbf{u}_k is confined to the invariant subspace \mathbf{z}_n^\perp . Hence we can invoke \mathbf{A}^\perp , the restriction of \mathbf{A} to \mathbf{z}_n^\perp (see section 1.4.1) to obtain a bound

$$\|\mathbf{A}\mathbf{u}_k\| = \|\mathbf{A}^\perp \mathbf{u}_k\| \leq \|\mathbf{A}^\perp\| = \max\{-\lambda_1, \lambda_{n-1}\}. \quad (4.8)$$

From (4.7) and (4.8) comes the decisive inequality

$$\frac{\tan \theta_{k+1}}{\tan \theta_k} = \frac{\|\mathbf{A}\mathbf{u}_k\|}{\lambda_n} \leq \frac{\|\mathbf{A}^\perp\|}{\|\mathbf{A}\|} \equiv \rho < 1. \quad (4.9)$$

By hypothesis $|\tan \theta_0| < \infty$ and so, as $k \rightarrow \infty$,

$$|\theta_k| \leq |\tan \theta_k| \leq \rho^k |\tan \theta_0| \rightarrow 0. \quad (4.10)$$

Rapid convergence can come in two ways: small ρ and/or small θ_0 .

It is left as Exercise 4.2.6 to show that almost always $\|\mathbf{A}\mathbf{u}_k\| \rightarrow \|\mathbf{A}^\perp\|$ as $k \rightarrow \infty$. Thus the convergence factor usually achieves its upper bound ρ . \square

4.2.2. Inverse Iteration (INVIT)

This is the power method invoking \mathbf{A}^{-1} . There is no need to invert \mathbf{A} ; instead replace $\mathbf{y}_k = \mathbf{A}\mathbf{x}_{k-1}$ in the power method by

$$1' : \text{ solve } \mathbf{A}\mathbf{y}_k = \mathbf{x}_{k-1} \text{ for } \mathbf{y}_k.$$

TABLE 4.2
INVIT iterates.

k	0	1	2	3	...	∞
x_k	1.0	0.700	0.559	0.554	...	0.554
	1.0	1.00	1.00	1.00	...	1.00
	1.0	-0.720	-0.931	-0.943	...	-0.944
$\frac{\sin \theta_{k+1}}{\sin \theta_k}$	0.1758	0.0439	0.0433			$0.0449 \approx \lambda_2/\lambda_3 \equiv \nu_0$

The convergence properties follow from Theorem 4.2.1.

Corollary 4.2.1. If λ_1 is the eigenvalue closest to 0, if $z_1^*x_0 \neq 0$ then, as $k \rightarrow \infty$ in INVIT, $x_k \rightarrow z_1$ linearly with convergence factor at worst $\nu = |\lambda_1/\lambda_2|$.

Example 4.2.1 (INVIT).

$$A \equiv \begin{bmatrix} -4 & 10 & 8 \\ 10 & -7 & -2 \\ 8 & -2 & 3 \end{bmatrix}, \quad \begin{aligned} \lambda_1 &\approx -17.895, \\ \lambda_2 &\approx 0.425, \\ \lambda_3 &\approx 9.470. \end{aligned}$$

Successive iterates, normalized to have the biggest element equal to one, are shown in Table 4.2. Compare the results with those in Table 4.1.

4.2.3. Shifts of Origin

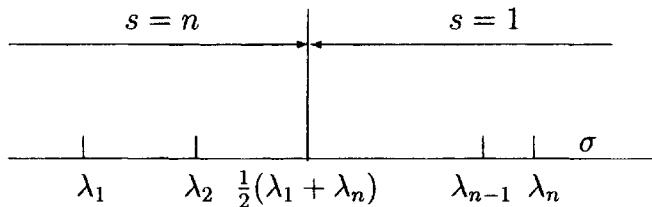
The power method and inverse iteration can easily be used with $A - \sigma I$ in place of A . The new convergence factors are as follows (Exercise 4.2.3).

Power method: $\rho_\sigma = \max_{j \neq s} |\lambda_j - \sigma| / |\lambda_s - \sigma|$
 where $|\lambda_s - \sigma| = \max_m |\lambda_m - \sigma|$.

Inverse iteration: $\nu_\sigma = |\lambda_t - \sigma| / \min_{j \neq t} |\lambda_j - \sigma|$
 where $|\lambda_t - \sigma| = \min_m |\lambda_m - \sigma|$.

Figure 4.2 reveals something special about the indices implicitly defined above.

Power method
finds λ_s



Inverse iteration
finds λ_t

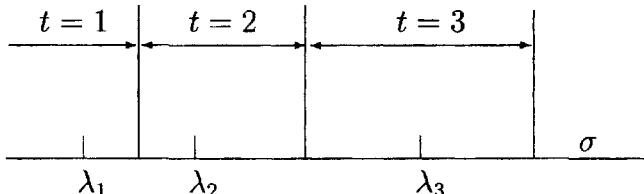


FIG. 4.2. The effect of σ on the limit.

As σ varies over all real values, s can take on only the value 1 or the value n . Thus the power method can only converge to z_1 or z_n and the choice of σ which minimizes ρ_σ is $(\lambda_2 + \lambda_n)/2$ in the former case and $(\lambda_1 + \lambda_{n-1})/2$ in the latter; neither gives a significant improvement over ρ_0 . In contrast $\nu_\sigma \rightarrow 0$ as $\sigma \rightarrow \lambda_j$ and so INVIT converges rapidly when σ is chosen well.

4.2.4. Cost

For a small full A each step of the power method requires n^2 ops to form Ax , n ops for $\|Ax\|^2$, and n divisions for y . Surprisingly each step of INVIT costs the same (approximately) provided that a triangular factorization $A - \sigma = L\Delta L^*$ is available. See section 3.1 for more information on $L\Delta L^*$. The new iterate y in $1'$ is computed in two stages, $Lw = x$ and $\Delta L^*y = w$, each of which requires $n^2/2$ ops. The heavy initial investment in the triangular factorization, namely

$n^3/3$ ops, can be amortized over all the steps taken. The catch in this argument is that INVIT is frequently run for only one step! That occurs when σ is a computed eigenvalue correct to nearly full precision.

The case of banded matrices is left as an exercise (Exercise 4.2.5).

For large matrices it is hard to make a general statement about cost. For large banded matrices factorization is often feasible despite the fill-in within the band. For matrices which cannot be factored the equation $Ay_k = x_{k-1}$ is sometimes itself solved iteratively and so the cost cannot be bounded a priori.

Exercises on Section 4.2

- 4.2.1. Consider $A^k/\|A^k\|$ and give a bound for the norm of the difference between it and a close rank-one matrix when $0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1} \leq \lambda_n$.
- 4.2.2. Show that $x_{k+1} = A^k x_1 / \|A^k x_1\|$.
- 4.2.3. Using Theorem 4.2.1 and its corollary establish the given equations for the convergence factors of the power method and inverse iteration with shifts.
- 4.2.4. With reference to (4.3), (4.4), and (4.5) show that if u_k is orthogonal to z_n then so is Au_k .
- 4.2.5. Compare the costs of one step of PM and INVIT when A has a half-bandwidth m ; $a_{ij} = 0$ when $|i - j| > m$.
- 4.2.6. Show that the sequence $\{u_k\}$ in the proof of Theorem 4.2.1 is produced by the power method using A^\perp . What condition ensures that $\|Au_k\| \rightarrow \|A^\perp\|$ as $k \rightarrow \infty$?
- 4.2.7. Suppose that $\lambda_1 > 0$. What value must λ_n/λ_1 take in order that $\min_\sigma \rho_\sigma = \frac{1}{2}\rho_0$ when $\rho_0 (= \lambda_{n-1}/\lambda_n)$ is close to 1? Conclude that shifts do not help the PM.

4.3. Advantages of an Ill-Conditioned System

Inverse iteration is often, but not always, used with a shift σ which is very close to some eigenvalue λ_j . The EISPACK programs INVIT and TINVIT use as σ a computed eigenvalue and it will often be correct to working accuracy. In these circumstances $A - \sigma$ may have a condition number for inversion as large as 10^{15} . Now one of the basic results in matrix computations is that roundoff errors can give rise to completely erroneous “solutions” to very ill conditioned systems of equations. So it seems that what is gained in the theoretical convergence rate by a good shift is lost in practice through a few roundoff errors. Indeed some textbooks have cautioned users not to take σ too close to any eigenvalue.

Fortunately these fears are groundless and furnish a nice example of confusing ends with means. In this section we explain why the INVIT techniques work so well.

Let x denote the unit starting vector and y the computed result of one step of inverse iteration with shift σ . Because of roundoff, described in section 3.2 or [Forsythe and Moler, 1967], y satisfies

$$(A - \sigma - H)y = x + f \quad (4.11)$$

where, with a good linear equations solver, $\|H\|$ is tiny compared with $\|A - \sigma\|$ and $\|f\|$ is tiny compared with 1 ($= \|x\|$). We can assume that $\|y\|$ is huge (e.g., 10^{10}) compared with 1.

For the sake of analysis we make use of the vector $g \equiv f + Hy$ so that we can rewrite (4.11) as

$$(A - \sigma)y = x + g. \quad (4.12)$$

Thus the “true” solution is $t \equiv (A - \sigma)^{-1}x$ and the error $e \equiv (A - \sigma)^{-1}g$. Of course g involves y and so this is an unnatural approach for *proving* something about y . However, a very weak assumption about g suffices to show why roundoff is not to be feared.

Figure 4.3 illustrates the situation.

Let σ be very close to λ_j and let $\psi = \angle(g, z_j)$. Then g can be decomposed orthogonally as

$$g = (z_j \cos \psi + u \sin \psi)\|g\|, \quad (4.13)$$

where $u^*z_j = 0$. It turns out that $\|g\|$ is irrelevant; all we need to assume about g ’s direction is some modest bound. In terms of Figure 4.3, suppose that

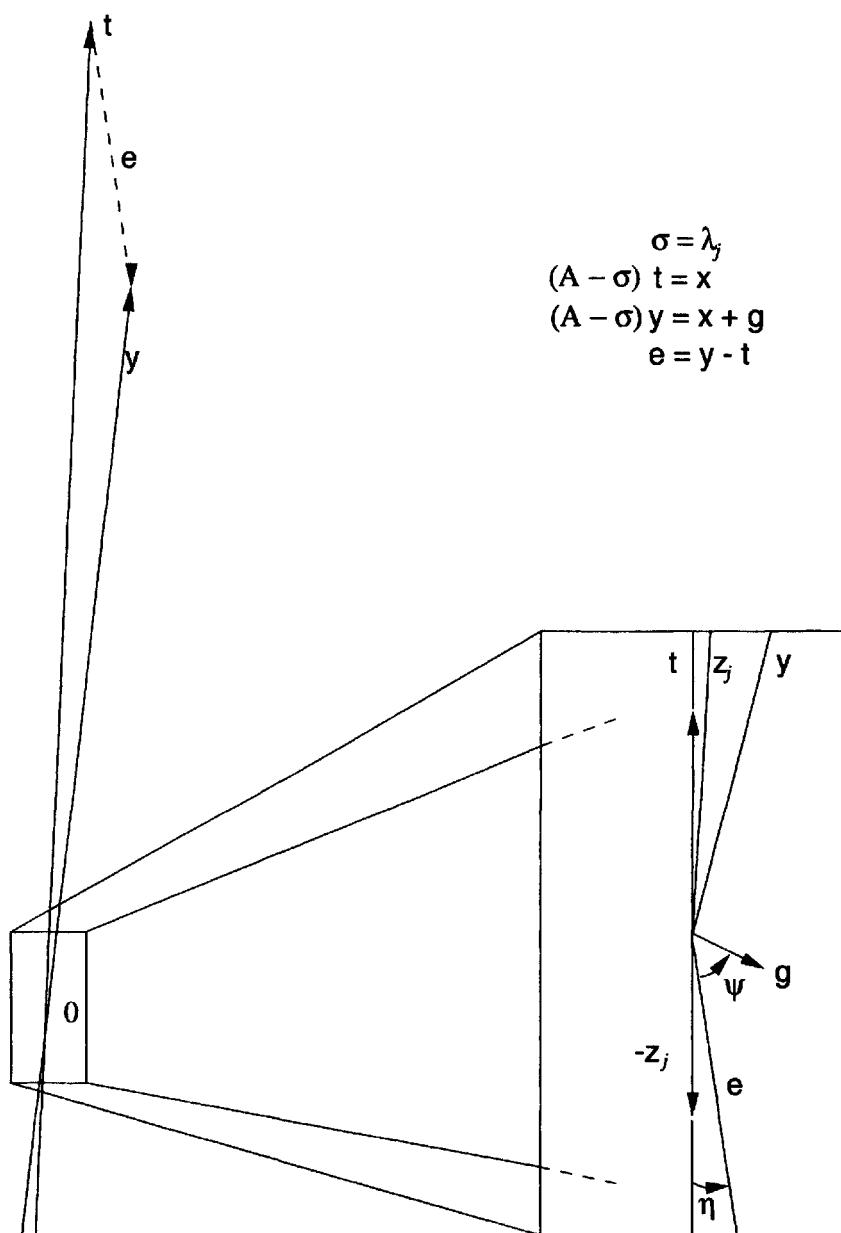
$$|\tan \psi| \leq 100. \quad (4.14)$$

From (4.13) we obtain an orthogonal decomposition of the dreaded error

$$e = (A - \sigma)^{-1}g = [z_j \cos \psi / (\lambda_j - \sigma) + \hat{u}\|(\lambda_j - \sigma)^{-1}u\| \sin \psi]\|g\|, \quad (4.15)$$

where \hat{u} is the normalized version of $(A - \sigma)^{-1}u$ and is, like u , orthogonal to z_j (see Exercise 4.3.1). If η denotes the all-important error angle between e and z_j then (4.15) says

$$\begin{aligned} \tan \eta &= \|(\lambda_j - \sigma)^{-1}u\|(\lambda_j - \sigma) \tan \psi \\ &\leq \frac{|\lambda_j - \sigma|}{\min_{k \neq j} |\lambda_k - \sigma|} \tan \psi, \quad \text{since } u \perp z_j. \end{aligned} \quad (4.16)$$

FIG. 4.3. *The error of inverse iteration.*

Our interest is precisely in the case when the factor multiplying $\tan \psi$ in (4.16) is very small (e.g., 10^{-10}). What (4.14) and (4.16) show is that

the error e , which may be almost as large as the exact solution $(A - \sigma)^{-1}x$, is almost entirely in the direction of z_j .

This result is alarming if we had hoped for an accurate solution of $(A - \sigma)y = x$ (the means) but is a delight in the search for z_j (the end).

When σ is correct to working precision then one step of inverse iteration is normally sufficient for convergence as Example 4.3.1 shows.

Example 4.3.1 (inverse iteration with a good shift).

Same A and x as in Example 4.2.1.

$\sigma = 9.463$. The condition of $(A - \sigma) = |\lambda_1 - \sigma|/|\lambda_3 - \sigma| = 10^5$.

y	t	-e
3301.3	3598.1	296.8
1567.6	1708.6	141.0
3598.4	3921.8	323.4

$\frac{y}{\ y\ }$	$\frac{t}{\ t\ }$	$\frac{-e}{\ e\ }$	Eigenvector
0.91744	0.91746	0.91775	0.91745
0.43564	0.43567	0.43500	0.43562
1.0000	1.0000	1.0000	1.0000

Exercises on Section 4.3

4.3.1. Show that if z is an eigenvector of A and if $u \perp z$ then $(A - \sigma)^{-1}u \perp z$.

4.3.2 (Calculator). Take the triangular matrix

$$B = \begin{bmatrix} 123.4 & 0.2273 & 0.1428 \\ 0 & 31.41 & -0.8571 \\ 0 & 0 & 2.718 \end{bmatrix}, \quad \sigma = 123.3.$$

Take one step of inverse iteration with $x = (1, \frac{1}{6}, \frac{1}{36})^*$. First do it in four-digit arithmetic (retain only the leading four digits of *each* intermediate quantity). Next do it to full precision. Take the difference of the two y 's as an estimate of e before normalizing them.

A diagonal matrix is too special, and a symmetric matrix requires more work than does the nonsymmetric B .

4.4. Convergence and Orthogonality

The k th step of inverse iteration computes y_k satisfying, in exact arithmetic,

$$(A - \sigma)y_k = x_{k-1}, \quad \|x_{k-1}\| = 1. \quad (4.17)$$

The *residual vector* of the approximate eigenpair (σ, y_k) is defined by

$$r_k \equiv (A - \sigma)y_k/\|y_k\| = x_{k-1}/\|y_k\|, \quad (4.18)$$

and, by (4.17), $\|r_k\| = 1/\|y_k\|$. Theorem 4.5.1 shows that σ differs from an exact eigenvalue of A by no more than $\|r_k\|$. Consequently, the computed number $\|y_k\|$ is used as a measure of convergence.

The main, perhaps the only, weakness of inverse iteration is that the computed eigenvectors for two close eigenvalues may be acceptable (because their residuals are small) and yet not be mutually orthogonal. *This sounds like a contradiction* because the exact eigenvectors must be orthogonal. However a small residual vector r_k only guarantees accuracy for isolated eigenvalues as the gap theorems in Chapter 11 reveal.

The current procedure in such a case is to take all the computed eigenvectors belonging to a cluster of eigenvalues (who shall decide on the clusters?) and compute the Rayleigh–Ritz approximations from their span as described in Chapter 11. After this facility has been incorporated into a program, however, the beautiful simplicity of inverse iteration has vanished and rival techniques, such as the QR algorithm of Chapter 8, become increasingly attractive, at least for small matrices.

A simpler remedy for clusters is to orthogonalize each approximate eigenvector, as soon as it is computed, against any eigenvectors already in the cluster. This technique and other devices are used in the subroutine TINVIT in EISPACK for tridiagonal A . Details are given under the name Tristurm in the last contribution to the Handbook (II/18).

In LAPACK the successor to TINVIT is called xstein (x indicates the word length). It is generally more accurate than TINVIT although slower but can still fail. See [Dhillon, 1998] for a careful analysis of xstein.

Sophisticated methods to compute mutually orthogonal eigenvectors for clustered eigenvalues (agreeing to at least three decimals) *without using orthogonalization* are under development in the late 1990s.

4.5. Simple Error Bounds

Let x be an approximate eigenvector and let $y = Ax$. The simplest approximation to an eigenvalue which can be derived from x and y is $\sigma = y_i/x_i$, where x_i is a maximal element of x . A better but more expensive approximation will be given below. First we ask how good are σ and x ?

Theorem 4.5.1. *For any scalar σ and any nonzero vector x there is an eigenvalue λ of A satisfying*

$$|\lambda - \sigma| \leq \|Ax - x\sigma\|/\|x\|.$$

Proof 1. If $\sigma = \lambda$, the result is immediate. If $\sigma \neq \lambda$ then $A - \sigma$ is invertible. So $x = (A - \sigma)^{-1}(A - \sigma)x$ and

$$\begin{aligned} 0 \neq \|x\| &\leq \|(A - \sigma)^{-1}\| \cdot \|(A - \sigma)x\| \\ &= (1/\min_i |\lambda_i[A] - \sigma|) \|(A - \sigma)x\|. \quad \square \end{aligned}$$

The proof given above rests on the not-so-obvious fact that for real symmetric matrices the spectral norm coincides with the spectral radius. For nonnormal matrices this coincidence fails as does Theorem 4.5.1. Next we give a second proof that emphasizes the Rayleigh quotient $\rho(x)$.

Proof 2.

$$\begin{aligned} (\|Ax - x\sigma\|/\|x\|)^2 &= \rho(x; (A - \sigma)^2) \\ &= \text{a weighted average of } \{(\lambda_i[A] - \sigma)^2\}_{i=1}^n \\ &\geq \min_i (\lambda_i[A] - \sigma)^2. \quad \square \end{aligned}$$

In most applications the vector $y (= Ax)$ and the number $\|x\|$ will be on hand, in which case the error bound can be computed for the modest price of

$2n$ ops. For another n ops the Rayleigh quotient $\rho = \rho(x) \equiv x^*(Ax)/\|x\|^2$ can be computed. Fact 1.9 in Chapter 1 shows that ρ is the scalar which minimizes, over all σ , the bound in Theorem 4.5.1. Consequently, it is quite reasonable to define the residual of x by $r = r(x) \equiv Ax - x\rho$. Then, as shown in section 1.5, $r^*x = 0$, and this yields an alternative and instructive derivation of the error bound in Theorem 4.5.1. The first step is Theorem 4.5.2.

Theorem 4.5.2. *Let x be any nonzero vector with Rayleigh quotient ρ and residual vector r . Then (ρ, x) is an eigenpair for a matrix $A - M$ and*

$$\|M\| \equiv \mu \equiv \|r\|/\|x\| \geq 0.$$

Proof. The magic formula for M is $(xr^* + rx^*)/\|x\|^2$. Verification that the nonzero eigenvalues of M are $\pm\mu$ and that M fulfills the claim of the theorem is left as an exercise. \square

This result accords with the spirit of backward error analysis (Chapter 2). If μ is less than the uncertainty in A then the question of error becomes moot; the pair (ρ, x) is as good an eigenpair as A warrants. In these circumstances a new question arises: How sensitive are the eigenvalues and eigenvectors to changes to, or uncertainties in, A ?

Fact 1.11 in Chapter 1 states that the eigenvalues of symmetric matrices are robust, that they change by no more than the change in the elements of A . Applying this to Theorem 4.5.2 above yields $|\lambda_i[A] - \rho| \leq \|A - (A - M)\| = \|M\| = \mu$ for some value of i . This is the same result as Theorem 4.5.1. More refined bounds are given in Chapter 11.

There is no analogous error bound for x , essentially because eigenvectors are not uniquely defined for multiple eigenvalues as Example 4.5.1 shows.

Example 4.5.1.

$$A = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 + \epsilon & 0 \\ 0 & 1 - \epsilon \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

With $x = e_1$ then $\rho(x) = 1$, and $r(x) = \epsilon e_2$. So $\|r\| = \mu = \epsilon$ and yet the error angle for x is $\pi/4$ or 45° ! Of course the close matrix $A - M$ of Theorem 4.5.2 is I which certainly has x as an eigenvector.

The experienced reader may object to this example as a cheat! Remember that eigenvectors do not change under translation of \mathbf{A} ($\mathbf{A} \rightarrow \mathbf{A} - \sigma$) nor under scaling ($\mathbf{A} \rightarrow \alpha\mathbf{A}$). Consequently \mathbf{A} has the same eigenvectors as $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, provided that ϵ is not negligible and the residual of \mathbf{e}_1 for this matrix has norm 1. So clearly (?) \mathbf{e}_1 is a poor approximate eigenvector.

When more information is available good bounds can be placed on the error in computed eigenvectors. Some of these results are given in Chapter 11.

Exercise on Section 4.5

- 4.5.1. Using the \mathbf{x} and \mathbf{r} in Theorem 4.5.2 and $\mu = \|\mathbf{r}\|/\|\mathbf{x}\|$ exhibit the eigenvectors of \mathbf{M} which belong to eigenvalues $\pm\mu$. Verify that (ρ, \mathbf{x}) is an eigenpair for $\mathbf{A} - \mathbf{M}$.

4.6. The Rayleigh Quotient Iteration

A natural extension of inverse iteration is to vary the shift at each step. The error bounds of the previous section suggest that the best shift which can be derived from the current eigenvector approximation \mathbf{x} is the *Rayleigh quotient* of \mathbf{x} , namely, $\rho(\mathbf{x}) = \mathbf{x}^* \mathbf{A} \mathbf{x} / \|\mathbf{x}\|^2$.

The Rayleigh quotient iteration. Pick a unit vector \mathbf{x}_0 ; then, for $k = 0, 1, 2, \dots$, repeat the following:

- RQI {
1. Compute $\rho_k = \rho(\mathbf{x}_k)$.
 2. If $\mathbf{A} - \rho_k$ is singular then solve $(\mathbf{A} - \rho_k)\mathbf{x}_{k+1} = \mathbf{o}$ for unit vector \mathbf{x}_{k+1} and stop. Otherwise, solve equation $(\mathbf{A} - \rho_k)\mathbf{y}_{k+1} = \mathbf{x}_k$ for \mathbf{y}_{k+1} .
 3. Normalize, i.e., $\mathbf{x}_{k+1} = \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\|$.
 4. If $\|\mathbf{y}_{k+1}\|$ is big enough then stop.

Definition 4.6.1. *The Rayleigh sequence generated by \mathbf{x}_0 is $\{\mathbf{x}_k, k = 0, 1, \dots\}$. It is not essential to use the Euclidean norm in 3 and 4.*

In the course of computing the fundamental mode of a vibrating system Lord Rayleigh, in the 1870s, improved an approximate eigenvector \mathbf{x}_1 by solving $[\mathbf{A} - \rho(\mathbf{x}_1)]\mathbf{y}_1 = \mathbf{e}_1$ for \mathbf{y}_1 . This technique is less powerful than RQI and, as far as we know, Lord Rayleigh never studied RQI. The name simply points to the shifts that are used to accelerate inverse iteration.

Cost. A new system of equations must be solved at each step. The increase in cost over inverse iteration with a fixed shift is large when \mathbf{A} is full but modest when \mathbf{A} is tridiagonal. See Exercise 4.6.2.

The rewards for the extra cost are considerable because the Rayleigh sequence converges very rapidly as Example 4.6.1 shows.

Example 4.6.1.

Same A and x as in Example 4.2.1.

k	0	1	2	∞
	1.000	0.8686	0.9176	0.9175
x_k	1.000	0.3644	0.4358	0.4356
	1.000	1.000	1.000	1.0000
ρ_k	8.000	9.444	9.468	9.468

Invariance properties. Analysis of the behavior of the sequence $\{\rho_k, x_k\}$ is more difficult than for inverse iteration because RQI is a *nonstationary* iteration; i.e., the function which maps x_k to x_{k+1} changes at every step. Consequently the behavior of RQI cannot be described easily in terms of $A - \rho_0$. For example, $\{\rho_k\}$ need not converge to the eigenvalue closest to ρ_0 as $k \rightarrow \infty$.

The following invariance properties of RQI are worth noting. Let A and x_0 produce the sequence $\{\rho_k, x_k : k = 0, 1, 2, \dots\}$.

Scaling. The matrix αA , $\alpha \neq 0$, and x_0 produce $\{\alpha \rho_k, x_k\}$.

Translation. The matrix $A - \alpha$ and x_0 produce $\{\rho_k - \alpha, x_k\}$.

Orthogonal similarity. The matrix $Q A Q^*$, where $Q^* = Q^{-1}$, and Qx_0 produce $\{\rho_k, Qx_k\}$.

Exercises on Section 4.6

- 4.6.1. Take one step of RQI using $A = \text{diag}(4, 2, 1)$ and $x_0^* = (0.1, 1.0, 0.1)$.
- 4.6.2. Do an operation count for one step of RQI in two separate cases:
(a) A full, n by n ; (b) A with half-bandwidth m .
- 4.6.3. Verify the invariance properties of RQI.

4.7. Local Convergence

When the Rayleigh sequence $\{x_k\}$ does converge to an eigenvector z the behavior is best described in terms of $\phi_k = \angle(x_k, z)$, the error angle. It turns out that as $k \rightarrow \infty$, $\phi_k \rightarrow 0$ to third order which ensures that the number of correct

digits in x_k triples at each step for k large enough—and this often means $k > 2$ (see Example 4.6.1).

The current iterate x_k can be written in terms of ϕ_k as

$$x_k = z \cos \phi_k + u_k \sin \phi_k, \quad (4.19)$$

where $u_k^* z = 0$ and $\|u_k\| = 1 = \|z\|$.

In part the phenomenal convergence rate can be attributed to the stationarity of the Rayleigh quotient ρ at eigenvectors; in particular,

$$\lambda - \rho(x_k) = [\lambda - \rho(u_k)] \sin^2 \phi_k, \quad (4.20)$$

where $Az = z\lambda$. Verification of (4.20) is left as Exercise 4.7.2.

We can now state the result formally. The analysis is similar to that for the power method and, if necessary, the reader should refer to section 4.2.

Theorem 4.7.1. *Assume that the Rayleigh sequence $\{x_k\}$ converges to an eigenvector. As $k \rightarrow \infty$ the error angles ϕ_k satisfy $\lim |\phi_{k+1}/\phi_k^3| \leq 1$. Equality holds almost always.*

Proof. We ignore the pleasant but unlikely possibility that the iteration terminates at finite k with ρ_k an eigenvalue and x_{k+1} its eigenvector.

First apply $(A - \rho_k)^{-1}$ to (4.19) and obtain

$$y_{k+1} = z \cos \phi_k / (\lambda - \rho_k) + u_{k+1} \sin \phi_k \| (A - \rho_k)^{-1} u_k \|, \quad (4.21)$$

where $u_{k+1} = (A - \rho_k)^{-1} u_k / \| (A - \rho_k)^{-1} u_k \|$ and is orthogonal to z . Since x_{k+1} is a multiple of y_{k+1} (see step 3 of RQI), (4.21) says

$$\begin{aligned} \tan \phi_{k+1} &= \sin \phi_k \| (A - \rho_k)^{-1} u_k \| / \cos \phi_k (\lambda - \rho_k)^{-1} \\ &= (\lambda - \rho_k) \| (A - \rho_k)^{-1} u_k \| \tan \phi_k \\ &= [\lambda - \rho(u_k)] \| (A - \rho_k)^{-1} u_k \| \tan \phi_k \sin^2 \phi_k. \end{aligned} \quad (4.22)$$

The line above made use of (4.20). The cleanest way to bound the last norm on the right of (4.22) is to invoke the restriction of $(A - \rho_k)$ to the invariant subspace z^\perp . Thus

$$\begin{aligned} \| (A - \rho_k)^{-1} u_k \| &= \| [(A - \rho_k)^{-1}]^\perp u_k \|, \text{ since } u_k \in z^\perp \\ &\leq \| [(A - \rho_k)^{-1}]^\perp \|, \text{ since } \|u_k\| = 1 \\ &= 1 / \min_{\lambda_i \neq \lambda} |\lambda_i - \rho_k|, \text{ using section 1.4.1.} \end{aligned} \quad (4.23)$$

The multiplicity of λ itself is irrelevant.

The hypothesis that $\{x_k\} \rightarrow z$ is used to bound the right-hand side in (4.23). Let the *gap* γ be defined by $\gamma \equiv \min |\lambda_i - \lambda|$ over all $\lambda_i \neq \lambda$. Since $\phi_k \rightarrow 0$, (4.20) shows that $\rho_k \equiv \rho(x_k) \rightarrow \lambda$ as $k \rightarrow \infty$. Thus

$$|\lambda_i - \rho_k| \geq \gamma/2 \quad \text{for large enough } k. \quad (4.24)$$

Cubic convergence of ϕ_k to 0 follows from (4.20), (4.22), (4.23), and (4.24), but further consideration of u_k reveals, surprisingly, that the gap γ does not affect the asymptotic convergence rate itself; a small value of γ merely delays the onset of the asymptotic regime. Further consideration of (4.21) reveals that the sequence $\{u_k\}$ is produced by inverse iteration with a variable shift ρ_k which is converging to λ . For large enough k the transformation from u_k to u_{k+1} is arbitrarily close to a step of inverse iteration in the subspace z^\perp with fixed shift λ .

Case 1. $\{u_k\}$ converges. The limit vector \hat{z} must be an eigenvector of A in z^\perp . Its eigenvalue $\hat{\lambda}$ will almost always be the one closest to λ but that is not germane. As $k \rightarrow \infty$ the key terms in (4.22) satisfy

$$\begin{aligned} [\lambda - \rho(u_k)] \|(\mathbf{A} - \rho_k)^{-1} u_k\| &\rightarrow \\ \pm \|(\lambda - \hat{\lambda})\hat{z}/(\hat{\lambda} - \lambda)\| &= \pm 1. \end{aligned} \quad (4.25)$$

On substituting (4.25) into (4.22) the limit in Theorem 4.7.1 is seen to be 1.

Case 2. $\{u_k\}$ does not converge. It is left to Exercises 4.7.3 and 4.7.4 to show that there are two eigenvalues of A equidistant from λ and the accumulation points of $\{u_k\}$ are two vectors in the plane spanned by the matching eigenvectors, say $\alpha z_p \pm \beta z_q$ where $\alpha^2 + \beta^2 = 1$, $\alpha \neq 0$, $\beta \neq 0$. It follows that, as $k \rightarrow \infty$, $\rho(u_k)$ converges and $|\phi_{k+1}/\phi_k^3| \rightarrow |\alpha^2 - \beta^2| < 1$. \square

Example 4.7.1 (cubic convergence).

Same data as in Example 4.6.1.

k	0	1	2
ρ_k	8.000	9.444	9.468
ϕ_k	0.3073	0.4954×10^{-1}	0.1204×10^{-3}
$\frac{\phi_{k+1}}{\phi_k^3}$	1.708	0.9903	—

Exercises on Section 4.7

- 4.7.1. Prove that $\|(\mathbf{A} - \sigma)^{-1}\mathbf{u}\| \leq 1/\gamma$ if \mathbf{u} is a unit vector and γ is the gap between σ and the eigenvalues of \mathbf{A} .
- 4.7.2. Verify (4.20).
- 4.7.3. Let $\{\mathbf{u}_k\}$ be generated by $(\mathbf{A} - \sigma)\mathbf{u}_{k+1} = \mathbf{u}_k\tau_k$ where τ_k ensures that $\|\mathbf{u}_{k+1}\| = 1$. Assume that $\{\mathbf{u}_k\}$ does not converge as $k \rightarrow \infty$ and show, by an eigenvector expansion or otherwise, that the accumulation points of $\{\mathbf{u}_k\}$ are of the form $\alpha\mathbf{z}_p + \beta\mathbf{z}_q$ where $\lambda_p = \sigma + \delta$, $\lambda_q = \sigma - \delta$. Deduce that $\rho(\mathbf{u}_k) \rightarrow \sigma + (\alpha^2 - \beta^2)\delta$ and $\|(\mathbf{A} - \sigma)^{-1}\mathbf{u}_k\| \rightarrow 1/\delta$.

4.8. Monotonic Residuals

The best computable measure of the accuracy of (ρ_k, \mathbf{x}_k) as an eigenpair for \mathbf{A} is the residual vector $\mathbf{r}_k \equiv (\mathbf{A} - \rho_k)\mathbf{x}_k$. The key fact, not appreciated until 1965, is that however poor the starting vector \mathbf{x}_0 may be the residuals always decrease in norm.

Theorem 4.8.1. *For the RQI, for all k , $\|\mathbf{r}_{k+1}\| \leq \|\mathbf{r}_k\|$. Equality holds if and only if $\rho_{k+1} = \rho_k$ and \mathbf{x}_k is an eigenvector of $(\mathbf{A} - \rho_k)^2$.*

In general an eigenvector of \mathbf{M}^2 need not be an eigenvector of \mathbf{M} . Thus \mathbf{x}_k in Theorem 4.8.1 need not be, indeed cannot be, an eigenvector of \mathbf{A} .

Proof.

$$\begin{aligned} \|\mathbf{r}_{k+1}\| &\equiv \|(\mathbf{A} - \rho_{k+1})\mathbf{x}_{k+1}\|, \quad \text{by definition,} \\ &\leq \|(\mathbf{A} - \rho_k)\mathbf{x}_{k+1}\|, \quad \text{by Fact 1.9,} \\ &= |\mathbf{x}_k^*(\mathbf{A} - \rho_k)\mathbf{x}_{k+1}|, \quad \text{since } \mathbf{x}_k \text{ is a multiple of } (\mathbf{A} - \rho_k)\mathbf{x}_{k+1}, \\ &\leq \|(\mathbf{A} - \rho_k)^*\mathbf{x}_k\| \cdot \|\mathbf{x}_{k+1}\|, \quad \text{by Cauchy-Schwarz inequality,} \\ &= \|\mathbf{r}_k\|, \quad \text{since } \mathbf{A} - \rho_k \text{ is symmetric and } \|\mathbf{x}_j\| = 1 \text{ for all } j. \end{aligned}$$

Equality holds in the first \leq only if $\rho_{k+1} = \rho_k$ and in the second instance only if \mathbf{r}_k is a multiple of \mathbf{x}_{k+1} , i.e., only if $(\mathbf{A} - \rho_k)\mathbf{x}_k = (\mathbf{A} - \rho_k)^{-1}\mathbf{x}_k\nu_k$ for some ν_k . \square

Example 4.8.1. This example comprises Tables 4.3 and 4.4.

TABLE 4.3
Normal case.

$$\mathbf{A} = \text{diag}(3, 2, 1)$$

k	0	1	2	3	4
x_k	0.3333	0.1374	0.0322	0.0019	0.2×10^{-5}
	0.6667	0.5494	0.3242	0.0411	0.7×10^{-4}
	0.6667	-0.8242	0.9455	-0.9992	1.0
ρ_k	1.6667	1.340	1.107	1.002	1.0
$\ r_k\ $	0.6667	0.5119	0.3104	0.0413	0.7×10^{-4}

TABLE 4.4
Stagnant case.

k	0	1	2
x_k	0	0.	0
	0.707	0.707	0.707
	0.707	-0.707	0.707
ρ_k	1.5	1.5	1.5
$\ r_k\ $	1	1.	1

*4.9. Global Convergence

Inverse iteration was introduced as a way of *improving* an approximate eigenvector. With the impact of automatic computation by digital computers it was natural to ask whether the method could *find* eigenvectors starting from scratch. Our analysis of inverse iteration with a fixed shift shows that convergence will occur from all starting vectors which are not orthogonal to *the* target eigenvector. However, some qualification is necessary because if $(\mathbf{A} - \sigma)$ has a \pm pair of smallest eigenvalues then $\{x_k\}$ does not know to which of the rival eigenvectors it should converge, although it tends to the subspace spanned by them without any hesitation.

When variable shifts are used convergence can be accelerated but there is the nasty possibility that, when initial approximations are poor, the shifts might lead to endless cycling. Consequently it is important to know that for the RQI (defined in section 4.6) the probability of such a cycle is zero. The proof here is a minor variation on Kahan's original argument.

Theorem 4.9.1. Let $\{x_k\}$ be the Rayleigh sequence generated by any unit vector x_0 . As $k \rightarrow \infty$,

- (1) $\{\rho_k\}$ converges, and either
- (2) $(\rho_k, x_k) \rightarrow (\lambda, z)$, cubically, where $Az = z\lambda$, or
- (3) $x_{2k} \rightarrow x_+, x_{2k+1} \rightarrow x_-$, linearly, where x_+ and x_- are the bisectors of a pair of eigenvectors whose eigenvalues have mean $\rho = \lim_k \rho_k$.

The situation in (3) is unstable under perturbations of x_k .

Proof. By the monotonicity of the residual norms, which was proved in the previous section,

$$\|r_k\| = \|(A - \rho_k)x_k\| \rightarrow \tau \geq 0 \text{ as } k \rightarrow \infty. \quad (4.26)$$

Since the sequence $\{x_k\}$ is confined to the unit sphere, a compact subset of \mathcal{E}^n , $\{x_k\}$ must have one or more accumulation points (vectors which are grazed infinitely often by the sequence). It remains to characterize these points and thereby count them. Note that ρ_k is also confined to a compact subset of \mathbb{R} , namely, $[-\|A\|, \|A\|]$.

Case 1: $\tau = 0$ (the usual case). Any accumulation point $(\bar{\rho}, z)$ of $\{\rho_k, x_k\}$ is, by definition, the limit of a subsequence $\{(\rho_j, z_j) : j \in \mathcal{J}\}$ for some index set \mathcal{J} . Let $j \rightarrow \infty$ in \mathcal{J} to see that, because $\rho(\cdot)$ is a continuous function on the unit sphere,

$$\rho(z) = \lim_{\mathcal{J}} \rho(x_j) = \lim_{\mathcal{J}} \rho_j = \bar{\rho} \quad (4.27)$$

and

$$\|(A - \bar{\rho})z\| = \lim_{\mathcal{J}} \|r_j\| = \tau = 0. \quad (4.28)$$

Thus $(\bar{\rho}, z)$ must be an eigenpair of A . By the local convergence theorem in section 4.7, as soon as both $|\rho_j - \bar{\rho}|$ and $\|x_j - z\|$ are small enough then, as $k \rightarrow \infty$ through all subsequent integer values, whether in \mathcal{J} or not, $|\rho_k - \bar{\rho}| \rightarrow 0$ and $\|x_k - z\| \rightarrow 0$ very rapidly. Consequently $\lim_{k \rightarrow \infty} x_k$ exists and is the eigenvector z .

There appears to be no simple description of how z depends on x_0 . See Exercise 4.9.1.

In order to analyze the harder case $\tau > 0$ we write the defining equation for RQI in the form

$$(A - \rho_k)x_{k+1} = x_k \tau_k, \quad \tau_k = 1/\|(A - \rho_k)^{-1}x_k\|. \quad (4.29)$$

Let $\theta_k = \angle(r_k, x_{k+1})$. Premultiply (4.29) by x_k^* to see that

$$0 < \tau_k = r_k^* x_{k+1} = \|r_k\| \cos \theta_k, \quad (4.30)$$

and so θ_k is acute. Moreover $\|r_{k+1}\|^2 = \|r_k\|^2 \cos^2 \theta_k - (\rho_k - \rho_{k+1})^2$.

Case 2: $\tau > 0$. The characteristic feature of this case is that $\|r_{k+1}\|/\|r_k\| \rightarrow \tau/\tau = 1$ as $k \rightarrow \infty$ and so the two equality conditions of the monotonic residual theorem must hold in the limit, namely, as $k \rightarrow \infty$,

$$|\rho_{k+1} - \rho_k| \rightarrow 0 \quad (4.31)$$

(which does not of itself imply that $\{\rho_k\}$ converges) and, since θ_k is acute,

$$\|r_k - (x_{k+1}\|r_k\|)\| \rightarrow 0, \quad (4.32)$$

i.e., $\theta_k \rightarrow 0$. Now (4.29), (4.30), and (4.32) yield an important result; as $k \rightarrow \infty$,

$$\begin{aligned} & \|[(A - \rho_k)^2 - \|r_k\|^2 \cos \theta_k]x_k\| \\ &= \|(A - \rho_k)(r_k - x_{k+1}\|r_k\|)\| \\ &\leq \|A - \rho_k\| \cdot \|r_k - x_{k+1}\|r_k\|\| \rightarrow 0. \end{aligned} \quad (4.33)$$

So any accumulation point $\bar{\rho}$ of the bounded sequence $\{\rho_k\}$ must satisfy

$$\det[(A - \bar{\rho})^2 - \tau^2] = 0, \quad (4.34)$$

whether or not $\{x_k\}$ converges. In other words, $\bar{\rho} = \lambda_i \pm \tau$ for one or more eigenvalues λ_i of A . This gives only a finite number of possible accumulation points and with this limitation (4.31) does imply that $\{\rho_k\}$ actually converges,¹³ i.e., $\rho_k \rightarrow \bar{\rho}$ as $k \rightarrow \infty$.

Turning to $\{x_k\}$ we see that any accumulation point x must be an eigenvector of $(A - \bar{\rho})^2$ without being an eigenvector of A (Case 1). This can

¹³Ultimately the sequence $\{\rho_k\}$ cannot jump a gap between two accumulation points because of (4.31).

only happen (Exercise 4.9.2) when orthogonal eigenvectors belonging to distinct eigenvalues $\bar{\rho} + \tau$ and $\bar{\rho} - \tau$ become a basis for a whole eigenspace of $(A - \bar{\rho})^2$. (When $\rho + \tau$ and $\rho - \tau$ are simple eigenvalues of A then their eigenvectors span an eigenplane of $(A - \rho)^2$. For example, if $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ then $A^2v = v$ for all v .)

The sequence $\{x_k\}$ cannot converge since, by definition of ρ_k , $x_k \perp r_k$ and, by (4.32), $r_k/\|r_k\| \rightarrow x_{k+1}$, as $k \rightarrow \infty$. This suggests that we consider $\{x_{2k}\}$ and $\{x_{2k-1}\}$ separately. Let x_+ be any accumulation point of $\{x_{2k}\}$. Then, as $2j \rightarrow \infty$ through the appropriate subsequence, the relations (4.29), (4.30), and (4.32) show that

$$x_{2j-1} = (A - \rho_{2j-1})x_{2j}/\tau_{2j-1} \rightarrow (A - \bar{\rho})x_+/\tau \equiv x_-.$$
 (4.35)

Multiply (4.35) by $(A - \bar{\rho})$ and use (4.34) to get

$$(A - \bar{\rho})x_- = (A - \bar{\rho})^2x_+/\tau = \tau x_+.$$
 (4.36)

Finally (4.35) and (4.36) combine to yield

$$(A - \bar{\rho})(x_+ \pm x_-) = \pm \tau(x_+ \pm x_-).$$
 (4.37)

The last equation shows that x_+ and x_- are indeed bisectors of eigenvectors of A belonging to $\bar{\rho} + \tau$ and $\bar{\rho} - \tau$. If these eigenvalues are simple then x_+ and x_- are unique (to within \pm) and so must actually be the limits of their respective sequences $\{x_{2k}\}$ and $\{x_{2k-1}\}$. Even if $\bar{\rho} + \tau$ and $\bar{\rho} - \tau$ are multiple eigenvalues there is still a unique normalized eigenvector that may be associated with each eigenvalue and together these eigenvectors serve to fix x_+ and x_- unambiguously. These eigenvectors are (Exercise 4.9.3) the projections of x_0 onto the eigenspaces of $\bar{\rho} + \tau$ and $\bar{\rho} - \tau$.

It remains to show the linearity of the convergence of $\{x_{2k}\}$ to x_+ and its instability under perturbations.

In the limit $\{x_{2k}\}$ behaves like inverse iteration with the matrix $(A - \bar{\rho})^2$. If τ^2 is its smallest eigenvalue then convergence will be linear and the reduction factor will depend on the next smallest eigenvalue of $(A - \bar{\rho})^2$. If there are eigenvalues smaller than τ^2 then any components of x_{2k} in the corresponding directions will have to vanish in the limit to permit convergence to x_+ . The details are left to Exercise 4.9.4.

To verify the instability it is simplest to check that if $v = x_+ + \epsilon x_-$ then (Exercise 4.9.5)

$$\begin{aligned} \|(A - \rho(v))v\|^2 &= \tau^2 - 3\epsilon^2\tau^2 + O(\epsilon^3), \quad \|v\|^2 = 1 + \epsilon^2, \\ &< \tau^2 \text{ for small enough } \epsilon. \end{aligned}$$

Thus x_+ is a saddle point of the residual norm and any perturbations of x_k that upset the balance between x_+ and x_- will drive the norm below τ and into Case 1. Perturbations orthogonal to both x_+ and x_- do not upset the regime of Case 2 (Exercise 4.9.6) and keep the residual norm above τ . \square

Rayleigh quotient shifts can be used with inverse iteration right from the start without fear of preventing convergence, but we cannot say to which eigenvector $\{x_k\}$ will converge.

Exercises for Section 4.9

- 4.9.1. Show by a 3-by-3 example that RQI can converge to an eigenvalue which is not the closest to ρ_0 and to an eigenvector which is not closest to x_0 .
- 4.9.2. Show that any eigenvector of A is an eigenvector of A^2 . Using an eigenvector expansion, or otherwise, show that the converse fails only when distinct eigenvalues of A are mapped into the same eigenvalue of A^2 .
- 4.9.3. By exercising the choice in the selection of eigenvectors for multiple eigenvalues show that there is no loss of generality in assuming that all eigenvalues are simple for the analysis of simple vector iterations.
- 4.9.4. See what conditions must be obtained so that $x_{2k} \rightarrow x_+$, $\rho_k \rightarrow \bar{\rho}$, $\|r_k\| \rightarrow \tau$ even when $(A - \bar{\rho})^2$ has eigenvalues smaller than τ^2 . Is it essential that x_0 have no components in the corresponding eigenvectors?
- 4.9.5. Recall that $(A - \bar{\rho})x_{\pm} = \pm\tau x_{\pm}$. Compute $\rho(x_+ + \epsilon x_-)$ and then

$$\| [A - \rho(v)]v \|^2$$

with $v = x_+ + \epsilon x_-$ retaining all terms through ϵ^4 . Evaluate the gradient and the Hessian of $\|(A - \rho)v\|^2$ at $v = x_+$.

- 4.9.6. Show that perturbations ϵu of x_+ increase the residual norm if $u \perp x_+$.
- 4.9.7. Let $\{u_k\}$ be generated by $(A - \sigma_k)u_{k+1} = u_k \tau_k$ where τ_k ensures that $\|u_{k+1}\| = 1$. Assume that (a) $\sigma_k \rightarrow \sigma$ and (b) $\{u_k\}$ does not converge. Show that the regime described in Exercise 4.7.3 must hold in this case too.

Notes and References

[Householder, 1964] and [Wilkinson, 1965] give valuable references to the development of the power method and inverse iteration during the 1950s and even earlier. Wilkinson was instrumental in exposing the myth that the proximity

of $A - \sigma$ to a singular matrix, when σ is a computed eigenvalue, will spoil the powerful convergence of inverse iteration.

The simple error bounds in section 4.5 have been derived by many people but are still not as well known as they should be.

Ostrowski [1958 and 1959] devoted two difficult papers to the RQI for symmetric matrices and gave a rigorous proof of asymptotic cubic convergence. See also [Crandall, 1951] and [Temple, 1952]. The key observation that the norms of the residual vectors are monotone decreasing is due to Kahan, as is the related global convergence theorem [Parlett and Kahan, 1969]. The proof of the monotone decrease in the residuals comes from [Parlett, 1974].

One reason for studying the global behavior of RQI is that it is essentially the same as that of the QL (or QR) algorithm of Chapter 8 when the shift is chosen to be the first (or last) diagonal entry.

Years of experience with inverse iteration on tridiagonal matrices has shown the importance of finding a good starting vector instead of relying on one with randomly chosen entries.

This page intentionally left blank

Deflation

When eigenvectors, or eigenvalues, are computed one by one it is necessary to prevent the algorithm from computing over again the quantities which it has already produced. In other words it is essential to get rid of each eigenvector immediately after it has been found. The established word for this banishment is *deflation*. Various ways of doing it are described below.

In each case the vector to be banished is a unit vector \hat{z} which makes a small angle η with some unit eigenvector z , so it can be written

$$\hat{z} = z \cos \eta + w \sin \eta, \quad w^* z = 0, \quad \|w\| = 1. \quad (5.1)$$

5.1. Deflation by Subtraction

The spectral theorem (Fact 1.4 in Chapter 1) expresses A as $\sum_{i=1}^n \lambda_i(z_i z_i^*)$. If λ_n and z_n were known then it would be tempting to work with the new n -by- n matrix \bar{A} defined by

$$\bar{A} = A - \lambda_n z_n z_n^* = \sum_{i=1}^{n-1} \lambda_i(z_i z_i^*), \quad (5.2)$$

which has traded z_n 's old eigenvalue λ_n for a new one $\bar{\lambda}_n = 0$. If $|\lambda_n| > |\lambda_{n-1}| > |\lambda_{n-2}| > \dots$ then, for example, the power method applied to \bar{A} will converge to z_{n-1} and the deflation process may be repeated again to yield $\bar{\bar{A}}$ whose largest eigenvalue is λ_{n-2} , and so on. That is the formal theory, but what happens in practice? Consider deflation of A by the vector \hat{z} given in (5.1) above. First observe that

$$\begin{aligned} \hat{z} \hat{z}^* &= z z^* \cos^2 \eta + \frac{1}{2} \sin 2\eta (z w^* + w z^*) + w w^* \sin^2 \eta \\ &= z z^* + W, \quad \text{defining } W, \end{aligned} \quad (5.3)$$

and $\|W\| = \sin \eta$ (Exercise 5.1.1). Given only \hat{z} the best approximation to z 's eigenvalue λ is the Rayleigh quotient (section 1.5),

$$\mu \equiv \rho(\hat{z}) = \lambda - [\lambda - \rho(W)] \sin^2 \eta. \quad (5.4)$$

Even if the subtraction of $\mu \hat{z} \hat{z}^*$ were performed exactly the result would be $\hat{A} \equiv A - \mu \hat{z} \hat{z}^*$ instead of \bar{A} . It is left as Exercise 5.1.2 to show that as $\eta \rightarrow 0$,

$$\|\bar{A} - \hat{A}\| = \mu \eta + O(\eta^2). \quad (5.5)$$

By Fact 1.11 some eigenvalues may be damaged by as much as $|\mu \eta|$. If μ approximates the smallest eigenvalue and η is tiny, then (5.5) is very satisfactory.

If μ approximates the dominant eigenvalue λ_n and/or η is not tiny (say $\eta = 10^{-4}$ rather than $\eta = 10^{-13}$), then (5.5) shows that \hat{A} is not very close to \bar{A} , and we may fear that by using \hat{A} we have already lost accuracy in the small eigenvalues. This fear is unnecessary as the following analysis suggests.

Let (μ, \hat{z}) approximate (λ_n, z_n) and let us see how close (λ_1, z_1) is to an eigenpair of $\hat{A} = A - \mu \hat{z} \hat{z}^* + H$ where H accounts for the roundoff error incurred in the subtractions $a_{ij} - \mu \xi_i \xi_j$. Assume that $\|H\| \leq 2\epsilon \|A\|$. The residual vector is

$$\hat{A}z_1 - z_1 \lambda_1 = (Az_1 - z_1 \lambda_1) - \mu \hat{z} (\hat{z}^* z_1) + Hz_1 \quad (5.6)$$

and

$$\hat{z}^* z_1 = (z_n \cos \eta + w_n \sin \eta)^* z_1 = (w_n^* z_1) \sin \eta. \quad (5.7)$$

By Theorem 4.5.1 there is an eigenvalue $\hat{\lambda}_1$ of \hat{A} satisfying

$$|\hat{\lambda}_1 - \lambda_1| \leq \|\hat{A}z_1 - z_1 \lambda_1\| \leq \mu |w_n^* z_1| \sin \eta + 2\epsilon \|A\|. \quad (5.8)$$

The new feature, missing in (5.5), is $w_n^* z_1$. Section 4.2 shows that $w_n \approx z_{n-1}$ and so $|w_n^* z_1| \approx \epsilon$. Thus it is the second term in (5.8) which dominates the error.

The change in distant eigenvalues and eigenvectors caused by deflation of an approximate eigenpair is the same as the change induced by perturbing the elements of A in the last place held.

Example 5.1.1 bears this out.

Example 5.1.1 (deflation in six-decimal arithmetic).

$A = 6\text{-by-}6$ Hilbert matrix

$$a_{ij} = 1/(i + j - 1)$$

$$\{\lambda_i[A]\}$$

$$\hat{A} = A - \lambda_6 z_6 z_6^*$$

$$\{\lambda_i[\hat{A}]\}$$

$$\begin{aligned} 0.11193 \times 10^{-6} \\ 0.12568 \times 10^{-4} \\ 0.61576 \times 10^{-3} \\ 0.16322 \times 10^{-1} \\ 0.24236 \times 10^0 \\ 0.16189 \times 10^1 \end{aligned}$$

$$\begin{aligned} 0.11832 \times 10^{-6} \\ 0.12569 \times 10^{-4} \\ 0.61576 \times 10^{-3} \\ 0.16322 \times 10^{-1} \\ 0.24236 \times 10^0 \\ 0.24544 \times 10^{-7} \end{aligned}$$

Eigenvector	z	\hat{z}	$\angle(z, \hat{z})$
$\lambda, \hat{\lambda}$	0.11193×10^{-6}	0.11833×10^{-6}	
	-0.1246×10^{-2}	-0.8704×10^{-1}	
	-0.3553×10^{-1}	-0.8728×10^{-1}	
$(Az = \lambda z)$	0.2406×10^0	0.2010×10^0	6.6°
	-0.6254×10^0	-0.6510×10^0	
$(\hat{A}\hat{z} = \hat{\lambda}\hat{z})$	0.6899×10^0	0.6602×10^0	(0.12 radian)
	-0.2717×10^0	-0.2912×10^0	
$\lambda, \hat{\lambda}$	0.12568×10^{-4}	0.12569×10^{-4}	$\angle(z, \hat{z})$
	0.1114×10^{-1}	0.1238×10^{-1}	
	-0.1797×10^0	-0.1790×10^0	
$(Az = \lambda z)$	0.6042×10^0	0.6043×10^0	0.63°
	-0.4437×10^0	-0.4434×10^0	
$(\hat{A}\hat{z} = \hat{\lambda}\hat{z})$	-0.4414×10^0	-0.4409×10^0	(0.011 radian)
	0.4591×10^0	0.4593×10^0	

The differences in the other eigenvectors were not significant.

Exercises on Section 5.1

5.1.1. Show that the matrix W of (5.3) satisfies $\|W\| = \sin \eta$.

5.1.2. By using Exercise 5.1.1 and (5.4) show that (5.5) holds as $\eta \rightarrow 0$.

5.1.3 (Computer project). Perturb elements in A of Example 5.1.1, recompute λ_1 and z_1 , and compare with Example 5.1.1.

5.2. Deflation by Restriction

If $Az = z\lambda$ and z is known then it is natural to continue working with A^\perp , the restriction of A to the invariant subspace z^\perp . See section 1.4 for basic information on z^\perp . A^\perp has all of A 's eigenpairs except for (λ, z) . To use the power method or inverse iteration with A^\perp it is only necessary, in exact arithmetic, to choose a starting vector orthogonal to z . Of course the product $(A^\perp)x$ is at least as expensive as Ax however many eigenvectors have been removed, an inevitable consequence of using the original A .

In practice we have \hat{z} instead of z , and roundoff will ensure that each computed product Ax will have a tiny but nonzero component of those eigenvectors already found. If ignored these components will grow until eventually they become dominant again. In fact it is a toss up whether the current sequence of vectors will converge to a new eigenvector before the dominant eigenvector pulls the sequence toward itself. This is the numerical analyst's version of the race between the hare and the tortoise.

In order to avoid these games it is wise to orthogonalize the current vector x against the computed eigenvectors from time to time. The current eigenvalue estimate can be used to determine the frequency with which the known vectors should be suppressed. See Exercise 5.2.2.

By these means the power method, or inverse iteration, can be made to converge to each eigenvector in turn. The accuracy is limited only by the accuracy with which the previous eigenvectors have been computed and the care taken to keep down their components in subsequent calculation. The price paid for this nice feature is a small increase in cost. To suppress the component of \hat{z} in x it is necessary to compute \hat{z}^*x and then replace x by $\hat{x} \equiv x - \hat{z}(\hat{z}^*x) = (I - \hat{z}\hat{z}^*)x$. This requires $2n$ ops. If all the eigenvectors were to be found by the power method and this mode of deflation, then the cost of the starting vectors for the later eigenvectors would become quite heavy. However, the point is that this technique is capable of giving accurate results and is attractive for large sparse matrices.

Exercises on Section 5.2

- 5.2.1. If the power method or inverse iteration produces the sequence $\{x_k\}$ then, for any eigenvector z , $x_1^*z = 0$ implies $x_k^*z = 0$ for $k > 1$. Show this and then examine the effect of explicitly orthogonalizing x_k against \hat{z} as given in (5.1). Assume $x_k^*z = O(\epsilon)$.
- 5.2.2. Let (λ_n, z_n) be a computed eigenpair. Let (μ, x) be the current estimates of a different eigenpair. Show that the increase in the component of z_n at the next step of the power method is approximately $|\lambda_n/\mu|$. If this

component is to be kept less than $\sqrt{\epsilon}$ find a formula for the frequency with which z_n should be purged from the current approximation.

5.3. Deflation by Similarity Transformation

The domain of the operator A^\perp mentioned in section 5.2 has dimension $n - 1$ and so it is not unreasonable to consider finding an $(n - 1)$ -by- $(n - 1)$ matrix which represents A^\perp rather than continuing to work with A . Formally such a matrix is easy to find. Take any orthogonal matrix P whose first column is the eigenvector z . Thus $P^* = P^{-1}$ and $Pe_1 = z$. The desired representation is the matrix $A^{(1)}$ shown in (5.9). Consider the orthogonal similarity transformation of A induced by P ,

$$\begin{aligned} P^*AP &= P^*(z\lambda, \dots) \\ &= \begin{bmatrix} \lambda & 0^* \\ 0 & A^{(1)} \end{bmatrix}, \quad \text{since } P^*z = e_1. \end{aligned} \quad (5.9)$$

If $A^{(1)}u = u\alpha$ then $P\binom{0}{u}$ is an eigenvector of A with eigenvalue α . Thus computation may proceed with $A^{(1)}$ provided that P is preserved in some way. In Chapters 6, 7, and 8 it will be shown how to pick a simple P , carry out the transformation, and preserve P . It must be admitted that at first sight this appears to be a complicated and costly technique. However, if all eigenvalues are wanted then the advantage of reducing the order of the matrix at each deflation is a definite attraction.

A subtle and important feature of the method can be brought out without going into great detail. In practice only the vector \hat{z} given in section 5.1 will be on hand, not the eigenvector z . Thus we must consider the orthogonal matrix \hat{P} with $\hat{P}e_1 = \hat{z}$ and the associated similarity transformation

$$\hat{P}^*\hat{A}\hat{P} = \begin{bmatrix} \mu & c^* \\ c & \hat{A}^{(1)} \end{bmatrix}. \quad (5.10)$$

It can be shown (Exercise 5.3.1) that if $\eta = \angle(z, \hat{z})$ is small enough then $\|c\| \leq \|A - \mu\|\eta + O(\eta^2)$. The point is that explicit computation of $\hat{P}^*\hat{A}\hat{P}$ may sometimes reveal that c is not small enough to be neglected. That is valuable information because, instead of deflating automatically by simply ignoring c , the algorithm can seek further orthogonal similarities which will reduce the (2, 1) and (1, 2) blocks as described in section 7.4. When this has been done (5.3) is replaced by

$$Q^*\hat{P}^*\hat{A}\hat{P}Q = \begin{bmatrix} \hat{\mu} & \hat{d}^* \\ \hat{d} & \hat{A}^{(1)} \end{bmatrix}, \quad (5.11)$$

where d is negligible and $\hat{\mu}$ is a probably undetectable improvement in μ . The advantage is that $\hat{A}^{(1)}$ is known to be an adequate representation of A^\perp and the computation can proceed safely. This is exactly what occurs in the tridiagonal QL algorithm of Chapter 8. Similarity transformations are continued until the matrix *declares itself reduced* (to working accuracy) and then the deflation occurs naturally.

Exercise on Section 5.3

- 5.3.1. Show that $\hat{P}^*(A\hat{z} - \hat{z}\mu) = \begin{pmatrix} 0 \\ c \end{pmatrix}$. Simplify $A\hat{z} - \hat{z}\mu$ using (5.1) and then conclude that $\|c\| = \|(\mathbf{A} - \mu)\mathbf{w}\| \sin \eta + O(\eta^2)$. Take $\mu = \rho(\hat{z}) = \hat{z}A\hat{z}$.

Notes and References

Deflation by subtraction is often attributed to [Hotelling, 1943], although the idea is quite natural. Several early papers of Wilkinson analyzed deflation, and the first and third methods are discussed thoroughly in [Wilkinson, 1965] and also in introductory texts such as [Fox, 1964]. Mention should be made of the influential paper by [Householder, 1961]. Deflation by similarities is the most attractive method for matrices in compact form. Typical examples are matrices with small bandwidth and arrowhead matrices that are nonzero only on or near the main diagonal as well as the last few rows and columns.

Useful Orthogonal Matrices (Tools of the Trade)

6.1. Orthogonals Are Important

Table 6.1 indicates the importance of orthogonal matrices.

TABLE 6.1
Preserving nice properties.

<i>Property</i>	<i>Transformations which preserve the property</i>	<i>Symbol</i>
Eigenvalues	Similarities	$A \rightarrow FAF^{-1}$
Symmetry	Congruences	$A \rightarrow FAF^*$ F invertible

Both properties are so valuable that we restrict attention to those transformations which preserve them. This entails

$$F^* = F^{-1}.$$

In the real case this forces F to be orthogonal, and the transformations induced by such F are called *orthogonal congruences* or, using one more syllable, *orthogonal similarities*.

Definition 6.1.1. *A real matrix F is orthogonal if*

$$F^*F = FF^* = I.$$

An orthogonal F is proper if $\det F = +1$. The definition of orthogonality can be interpreted as saying that the columns of F are mutually orthonormal and so are the rows.

The next step is to find classes of simple orthogonal matrices that are rich enough so that any orthogonal matrix can be obtained by forming products of the simple ones.

6.2. Permutations

A permutation of an ordered set of n objects is a rearrangement of those objects into another order; in other words a permutation is a one-to-one transformation of the set onto itself. One notation for a permutation π is

$$(\pi_1, \pi_2, \dots, \pi_n), \quad (6.1)$$

where π_j is the new position of the object initially in position j .

Example 6.2.1.

$$\begin{aligned} n &= 3, & \pi &= (3, 1, 2), & O &= (O_1, O_2, O_3), \\ \pi O &= (O_2, O_3, O_1). \end{aligned}$$

If an ordered set O of n objects is permuted first by $\pi^{(1)}$ and then by $\pi^{(2)}$, the result is another permutation of O called the composition or product of $\pi^{(1)}$ and $\pi^{(2)}$ and written like multiplication as

$$\pi^{(2)}\pi^{(1)}O. \quad (6.2)$$

The permutations form a group under composition; in particular each permutation has an inverse.

Example 6.2.2.

$$\pi = (3, 1, 2), \quad \pi^{-1} = (2, 3, 1).$$

If the objects are the columns of a matrix B , then a permutation of them can be effected by postmultiplying B by a special matrix P called a *permutation matrix*. P is obtained from I by permuting its *columns* by π . Thus, if $\pi = (3, 1, 2)$ then

$$\pi(b_1, b_2, b_3) = (b_2, b_3, b_1) = BP, \quad P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (6.3)$$

Note that P is also obtained by rearranging the rows of I according to π^{-1} .

The n -by- n permutation matrices form a group under matrix multiplication. Each such matrix can be represented compactly by the corresponding permutation π .

Most permutation matrices which occur in eigenvalue computations are products of very simple matrices called *interchanges* or *swaps*, which swap a pair of rows or columns and leave all the others unchanged. It is a fact that all permutations can be written as a product of interchanges, usually in several ways.

Example 6.2.3.

$$(3, 1, 2) = (1, 3, 2)(2, 1, 3) \text{ (multiplication is from right to left).}$$

A more compact way of representing a permutation is as a product of disjoint cycles. However, this representation does not seem to be used in matrix computations, presumably because the conversion to and from interchanges to cycles does not seem to be warranted.

Every interchange is its own inverse, and so a sequence of swaps simultaneously represents both a permutation and its inverse. The inverse is obtained by performing the swaps in reverse order.

Although permutations involve no floating point arithmetic operations they are useful tools. In the solution of large sparse sets of linear equations the ordering of the equations has a decisive influence on the cost of Gauss elimination.

For more information on the implementation of permutations in a computer see [Knuth, 1969, vol. I, section 1.3.3].

Exercises on Section 6.2

6.2.1. If P is the permutation matrix corresponding to π prove that P^* corresponds to π^{-1} .

6.2.2. Often in matrix applications a sequence of interchanges has the special form $(1, \nu_1), (2, \nu_2), \dots, (n - 1, \nu_{n-1})$ where $\nu_j \geq j$. This sequence can be represented compactly by the array $(\nu_1, \nu_2, \dots, \nu_{n-1}) = n$.

- (a) Write an algorithm to effect the inverse of the permutation represented by n .
- (b) Write an algorithm to convert from the representation n to a product of disjoint cycles.

6.3. Reflections and Direct Rotations

A basic result in real Euclidean geometry is that any rigid motion which leaves the origin fixed (i.e., any orthogonal transformation) can be represented as a *product of reflections*. The mirror for a reflection in \mathcal{E}^n is an $(n - 1)$ space or hyperplane, which is most easily characterized by the direction orthogonal (or *normal*) to it.

Definition 6.3.1. *The hyperplane normal to u is $\{x : u^*x = 0\}$.*

To each u there corresponds a unique reflection which reverses u and leaves invariant any vector orthogonal to u .

Definition 6.3.2. *The matrix $H(u)$ which effects the reflection $x \rightarrow H(u)x$ is the reflector which reverses u ;*

$$H(u)v = \begin{cases} -v & \text{if } v = \alpha u, \\ v & \text{if } u^*v = 0. \end{cases}$$

Note that $H(\alpha u) = H(u)$ for any $\alpha \neq 0$. It is easily verified that the matrix representation of $H(u)$ is

$$H(u) = I - \gamma uu^*, \quad \gamma = \gamma(u) = 2/u^*u.$$

The basic properties of $H(u)$ are covered in Exercises 6.3.1 and 6.3.2. Note that $H(u)$ has everything: it is elementary, symmetric, orthonormal, and its own inverse!

By a sequence of reflections we can build up any orthogonal transformation. No other tool is necessary (Exercise 6.3.3). There is however one small blemish; I is not a reflector (but $H^2 = I$). If a sequence of orthogonal matrices converges to I then the reflector factors of each matrix will not converge to I . In section 6.3.2 we present a proper orthogonal matrix (determinant +1) which is analogous to $H(u)$.

Standard task: Given b find c such that $H(c)b = e_1\mu$.

Solution: $\mu = \mp\|b\|$, $c = b \pm e_1\mu$ (see Exercise 6.3.4).

6.3.1. Computing with Reflections

Let $b = (\beta_1, \dots, \beta_n)^*$, $c = (\gamma_1, \dots, \gamma_n)^*$. The only arithmetic step in computing c is the calculation $\gamma_1 = \beta_1 \pm \mu$.

Case 1. $\mu = -\|b\|\text{sign}(\beta_1)$. The quantity $\gamma_1 = (|\beta_1| + \|b\|)\text{sign}(\beta_1)$ involves addition of positive numbers and will have a low relative error always.

Case 2. $\mu = \|\mathbf{b}\| \operatorname{sign}(\beta_1)$. The formula $\gamma = (|\beta_1| - \|\mathbf{b}\|) \operatorname{sign}(\beta_1)$ involves genuine subtraction which will result in a high relative error whenever $|\beta_1| \doteq \|\mathbf{b}\|$. Example 6.3.1 shows the dire effects of using the obvious formula in this case.

It is sometimes said that Case 2 is unstable and should not be used. This is not right. It is the formula $(|\beta_1| - \|\mathbf{b}\|)$ that is dangerous. Below we give a formula which always gives low relative error because the subtraction is done analytically.

$$\begin{aligned} |\beta_1| - \|\mathbf{b}\| &= (\beta_1^2 - \|\mathbf{b}\|^2)/(|\beta_1| + \|\mathbf{b}\|) \\ &= -\sigma/(|\beta_1| + \|\mathbf{b}\|), \end{aligned}$$

where $\sigma = \beta_2^2 + \cdots + \beta_n^2$. The extra cost is one division and one addition.

Recent work (see Notes and References at the end of the chapter) shows that when \mathbf{b} is close to \mathbf{e}_1 then Case 2 preserves the information in \mathbf{b} better than does Case 1 [Dubrulle, 1996].

Example 6.3.1 (reflections in Case 2).

Consider four-decimal arithmetic ($1 + 10^{-4} \rightarrow 1$).

$$\begin{aligned} \mathbf{b} &= \begin{bmatrix} 1 \\ 10^{-2} \end{bmatrix}, \quad \|\mathbf{b}\| = 1, \quad \mathbf{c} = \begin{bmatrix} 0 \\ 10^{-2} \end{bmatrix}, \\ \mathbf{H}(\mathbf{c}) &= \mathbf{I} - \gamma \mathbf{c} \mathbf{c}^* = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \\ \mathbf{H}(\mathbf{c})\mathbf{b} &= \begin{bmatrix} 1 \\ 10^{-2} \end{bmatrix} \neq \mathbf{e}_1!! \end{aligned}$$

Use of the proper formula yields $\mathbf{c} = \begin{bmatrix} 10^{-4}/2 \\ 10^{-2} \end{bmatrix}$, which is correct to working precision.

In Case 1 \mathbf{b} is reflected in the external bisector of \mathbf{b} and \mathbf{e}_1 ; in Case 2 it is the internal bisector. For real vectors Case 1 (which is the popular one) yields the vector $-\|\mathbf{b}\|\mathbf{e}_1$. This is a mild nuisance because the QR factorization (section 6.7) is defined with a positive diagonal for \mathbf{R} and Case 1 makes it negative. Also it is more natural to reflect in the internal bisector when \mathbf{b} is close to \mathbf{e}_1 .

6.3.2. The Direct Rotation

There are many orthogonal matrices that map one subspace of \mathcal{E}^n into another and among them there is an optimal choice, optimal because it is the one that

is closest to the identity matrix I . Either the Frobenius norm or the spectral norm may be used to measure the difference. The optimal orthogonal matrix is called the *direct rotation*.

This valuable idea was introduced and developed in [Davis, 1958] in the setting of Hilbert space, although mention of the idea can be found in other books on functional analysis such as [Kato, 1966]. When Davis joined forces with Kahan the explicit form of the direct rotation in \mathcal{E}^n was revealed; see [Davis and Kahan, 1969]. In our one-dimensional application the direct rotation offers the extra attraction of simplicity, and simplicity leads to efficiency in implementation. Nevertheless, for no apparent reason, the direct rotation is hardly ever used. It must be admitted that in the one-dimensional case the direct rotation differs from the identity I by a rank-two matrix and thus is not an elementary matrix.

Task: Map a two-component vector $b = \begin{pmatrix} \beta \\ c \end{pmatrix}$ onto $e_1 \|b\| \text{sign}(\beta)$.

Solution:

$$K = \begin{pmatrix} \text{abs}(\beta) & c^* \text{sign}(\beta) \\ -c \text{sign}(\beta) & W \end{pmatrix} / \|b\| \quad (\text{sign}(0) = 1),$$

where $W/\|b\|$ is a symmetric elementary matrix

$$W = \|b\| I - \nu cc^*,$$

$$\nu = 1/(\|b\| + \text{abs}(\beta)) \quad (\text{Exercise 6.3.6}).$$

Note that $W/\|b\|$ is not a reflector matrix and $\det[K] > 0$. The verification of the properties of K is left as an exercise. There is no need to remember the precise expression for ν because it is determined by the requirement that K should be orthogonal.

We now indicate why K gives the direct rotation. To this end we write $\cos \theta = \text{abs}(\beta)/\|b\|$ and $s = c/\|c\|$. Further let $\dot{s} = \begin{pmatrix} 0 \\ s \end{pmatrix}$ and $P = (e_1, \dot{s})$. Thus $b \text{sign}(\beta) = e_1 \cos \theta + \dot{s} \sin \theta$ belongs to range P and with the given basis in range P the required rotation is simply the matrix Θ given by

$$\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$$

as it should be. On the complement, $\text{range}(P)^\perp$, the direct rotation is simply I . Thus

$$K = P\Theta P^* + (I - PP^*).$$

However, column 1 of P is just e_1 , and so the form of K is clear except for $W/\|b\|$ which must be

$$\begin{aligned} & s \cos \theta s^* + (I_{2:n} - ss^*) \\ &= I_{2:n} - s(1 - \cos \theta)s^* \\ &= I_{2:n} - s \sin \theta \nu (s \sin \theta)^*/\|b\| \\ &= I_{2:n} - c \nu c^*/\|b\| \end{aligned}$$

since $s \sin \theta = c \operatorname{sign}(\beta)$.

Exercises on Section 6.3

- 6.3.1. Show that $H(u)^* = H(u)$, $H^2(u) = I$, and $H(u)$ is elementary. Recall that a matrix is elementary if it differs from I by a rank-one matrix.
- 6.3.2. Find the eigenvalues and eigenvectors of $H(u)$. Show that $\det[H] = -1$.
- 6.3.3. Express $\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$ as a product of two reflectors. Draw a picture to show the mirrors.
- 6.3.4. Derive the formula $u = v - e_1 \mu$ which ensures $H(u)v = e_1 \mu$. Find the limiting form of both reflectors as $v \rightarrow e_1$. Take $v = e_1 + \epsilon s$ with $\epsilon \rightarrow 0$.
- 6.3.5. How should u be chosen so that $H(u)v = w$ when $w^*w = v^*v$?
- 6.3.6. Let s be an $(n-1)$ -vector and let $\gamma^2 + \|s\|^2 = 1$. How must ν be chosen so that the partitioned matrix

$$K = \begin{bmatrix} \gamma & -s^* \\ s & I - \nu ss^* \end{bmatrix}$$

is orthogonal?

- 6.3.7. Find an efficient implementation of the product KMK^* by partitioning M in the same way as K . This is the core of the reduction of a dense A to tridiagonal form using direct rotations. How does the operation count compare with the use of reflectors?
- 6.3.8. Find a nice expression for the direct rotation that maps x onto a vector y of the same norm. Use an n -by-2 matrix P like the one introduced in section 6.3.2.

6.4. Plane Rotations

The matrix representation of the transformation which rotates \mathcal{E}^2 through an angle θ (counterclockwise) is

$$R(\theta) \equiv \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad R^{-1}(\theta) = R(-\theta).$$

There are two different tasks for R:

1. Find θ so that $R(\theta)b = e_1\mu$ given $b = (\beta_1, \beta_2)^*$. The solution is given by $\tan \theta = -\beta_2/\beta_1$ (Exercise 6.4.1).
2. Find θ so that $R(\theta)AR(-\theta)$ is diagonal. The two solutions, $0 \leq \theta < \pi$, are found from $\cot 2\theta = (a_{22} - a_{11})/2a_{12}$ (Exercise 6.4.2).

In n -space we use a *plane rotation* $R(i, j, \theta)$ which rotates the (i, j) plane, that is, the plane spanned by e_i and e_j , through an angle θ and leaves the orthogonal complement of this plane invariant. So $R(i, j, \theta)$ is the identity matrix except that $r_{ii} = r_{jj} = \cos \theta$, $-r_{ij} = r_{ji} = \sin \theta$ ($i < j$). Here is a class of proper orthogonal matrices which includes I. Each $R(i, j, \theta)$ is an elementary matrix (Exercise 6.4.3).

The analogue of task 1 above is to transform a given b into a multiple of e_1 by a *sequence* of plane rotations. Either of the following choices will work:

$$R(1, n, \theta_n) \cdots R(1, 3, \theta_3)R(1, 2, \theta_2)b = e_1\mu,$$

$$R(1, 2, \phi_2) \cdots R(1, n-1, \phi_{n-1})R(1, n, \phi_n)b = e_1\nu,$$

where

$$|\mu| = |\nu| = \|b\|.$$

Other sequences of planes are possible. In general each sequence produces a different orthogonal transformation of b into $\pm e_1\|b\|$. These solutions are rivals to the two reflections which accomplished the same task in section 6.3.

Task 2 has two variants that are given below. Figure 6.1 helps to distinguish them.

6.4.1. Jacobi Rotation

Find θ such that the (i, j) and (j, i) elements of $R(i, j, \theta)AR(i, j, -\theta)$ are zero. These rotations are discussed more fully in Chapter 9.

6.4.2. Givens Rotation

Find ϕ such that the (k, l) and (l, k) elements of $R(i, j, \phi)AR(i, j, -\phi)$ are zero. Since rows i and j are the only ones to change we must have one of the pair i, j equal to one of the pair k, l .

Jacobi actually used his rotations in 1846, but the reflections and Givens rotations were not introduced until the 1950s. In many ways Givens rotations

$$\begin{array}{c}
 R(3, 5, \theta)^* \\
 \left[\begin{array}{ccccc} 1 & & & & \\ & 1 & & & \\ & & c & +s & \\ & & 1 & & \\ & -s & c & & \\ & & & 1 & \end{array} \right] \quad A \quad \left[\begin{array}{ccccc} * & * & a'_{13} & * & a'_{15} & * \\ * & * & a'_{23} & * & a'_{25} & * \\ , & , & a''_{31} & a''_{32} & a''_{33} & a''_{34} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ * & * & a''_{43} & * & a''_{45} & * \\ a''_{51} & a''_{52} & a''_{53} & a''_{54} & a''_{55} & a''_{56} \\ * & * & a''_{63} & * & a''_{65} & * \end{array} \right] \quad R(3, 5, \theta) \\
 \left[\begin{array}{ccccc} 1 & & & & \\ & 1 & & & \\ & & c & -s & \\ & & 1 & & \\ & +s & c & & \\ & & & 1 & \end{array} \right]
 \end{array}$$

Key:

*	Unchanged
a'	Changed once
a''	Changed twice
Blank	is zero

FIG. 6.1. Effect of plane rotation of A .

are more useful than Jacobi rotations. The initial attraction of the Jacobi rotation arises from the fact that, of all choices for θ , with given i, j , it produces the biggest decrease in the sum of the squares of the off-diagonal elements.

6.4.3. Operation Count

If $B' = R(i, j, \theta)B$ then

$$\begin{aligned}
 b'_{ik} &= b_{ik} \cos \theta - b_{jk} \sin \theta &= \cos \theta(b_{ik} - b_{jk} \tan \theta), \\
 b'_{jk} &= b_{ik} \sin \theta + b_{jk} \cos \theta &= \cos \theta(b_{ik} \tan \theta + b_{jk})
 \end{aligned}$$

for all k . This operation requires $4n$ multiplications. The count is identical for the formation of RAR^{-1} provided that one takes advantage of symmetry. A square root is required for the calculation of $\cos \theta$.

For scaled versions of $R(i, j, \theta)$ see section 6.8.

6.4.4. Compact Storage

When a long sequence of plane rotations must be recorded it is often advantageous to accept a little extra computation in order to encode each rotation with a single number rather than the pair $\cos \theta, \sin \theta$. Clearly θ itself is wasteful and $t = \tan \theta$ is the natural choice. Formally c and s are recovered from

$$c = 1/\sqrt{1+t^2}, \quad s = c \cdot t \quad (-\pi/2 < \theta \leq \pi/2).$$

This scheme will fail when θ is $\pi/2$ or very close to it and a slightly more complicated process is called for. [Stewart, 1976b] advocates saving the number ρ defined by

$$\rho = \begin{cases} 1 & \text{if } \sin \theta = 1, \\ \sin \theta & \text{if } |\sin \theta| < \cos \theta, \\ \sec \theta \cdot \operatorname{sign}(\sin \theta) & \text{if } |\sin \theta| \geq \cos \theta. \end{cases}$$

Exercises on Section 6.4

- 6.4.1. Show that $R(\theta)v = \pm e_1\|v\|$ if $\tan \theta = -v_2/v_1$.
- 6.4.2. Show that $e_1^* R(\theta) A R(-\theta) e_2 = 0$ if $\cot 2\theta = (a_{22} - a_{11})/2a_{12}$. What is the relation between the two possible values of θ ?
- 6.4.3. Show that $R(i, j, \theta)R(i, j, \pi/2)$ is an elementary matrix. Hint: Think of $\theta/2$.
- 6.4.4. Find the Givens rotation in the $(2, 3)$ plane which annihilates elements $(1, 3)$ and $(3, 1)$.
- 6.4.5. Write out the product $R(2, 3, \theta)R(2, 4, \phi)$ when $n = 5$.
- 6.4.6. Find a neat, stable algorithm to recover $\sin \theta$ and $\cos \theta$ from Stewart's encoding ρ given above.

6.5. Error Propagation in a Sequence of Orthogonal Congruences

This section is of fundamental importance to the understanding of the stability of many popular eigenvalue computations based on transforming the given matrix to simpler form. The typical situation is that A_0 is given and A_k is computed, in principle, as the last term in the sequence

$$A_j = B_j^* A_{j-1} B_j, \quad j = 1, \dots, k,$$

where each B_j is orthogonal and depends on A_{j-1} .

This is not what happens in practice. Let us consider a typical step. From the current matrix A_{j-1} we compute B_j but it will not be exactly orthogonal. Let

$$B_j = G_j + \hat{F}_j, \tag{6.4}$$

where G_j is an unknown orthogonal matrix very close to B_j . We will discuss the size of the error \hat{F}_j later. Now we try to compute $B_j^* A_{j-1} B_j$ but fail. Instead the resulting matrix A_j will satisfy

$$A_j = B_j^* A_{j-1} B_j + \bar{W}_j, \tag{6.5}$$

where \bar{W}_j is the local error matrix which results from roundoff error in the similarity transformation. Sometimes it is useful to write $A_j = fl(B_j^* A_{j-1} B_j)$ to indicate the intention; *fl* is an acronym for “floating point result of.” Because of the nice properties of orthogonal matrices we replace B_j by G_j to find

$$\begin{aligned} A_j &= (G_j + \hat{F}_j)^* A_{j-1} (G_j + \hat{F}_j) + \bar{W}_j \\ &= G_j^* A_{j-1} G_j + W_j, \end{aligned} \quad (6.6)$$

where

$$W_j \equiv \bar{W}_j + G_j^* A_{j-1} \hat{F}_j + \hat{F}_j^* A_{j-1} G_j + \hat{F}_j^* A_{j-1} \hat{F}_j. \quad (6.7)$$

Here W_j is a name for a messy, but hopefully tiny, local error matrix. Now apply (6.6) for $j = k, k-1, \dots, 2, 1$ in turn to find

$$\begin{aligned} A_k &= G_k^* A_{k-1} G_k + W_k \\ &= G_k^* (G_{k-1}^* A_{k-2} G_{k-1} + W_{k-1}) G_k + W_k \\ &= \dots \\ &= J_0^* A_0 J_0 + \sum_{j=1}^k J_j^* W_j J_j, \end{aligned} \quad (6.8)$$

where

$$J_j = G_{j+1} \cdots G_k, \quad J_k = I. \quad (6.9)$$

The final step is to rewrite (6.8) in a simple form

$$A_k = J_0^* (A_0 + M_0) J_0, \quad (6.10)$$

where

$$M_0 = J_0 \left(\sum_{j=1}^k J_j^* W_j J_j \right) J_0^*. \quad (6.11)$$

All we have done is to relate A_k to A_0 in terms of matrices which are orthogonal. Equation (6.10) says that the *computed* A_k is exactly orthogonally congruent to $A_0 + M_0$.

Definition 6.5.1. M_0 is called the equivalent perturbation matrix induced by the computation.

When A_k is finally written over A_{k-1} the best that can be done is to find the eigenvalues of $A_0 + M_0$; those of A_0 are beyond recall. Because the G_j , and therefore the J_j , are orthogonal we have

$$\|M_0\| \leq \sum_{j=1}^k \|W_j\|. \quad (6.12)$$

Moreover, using (6.7) and the orthogonality of G , we get

$$\|W_j\| \leq \|\bar{W}_j\| + \|A_{j-1}\|(2\|\hat{F}_j\| + \|\hat{F}_j\|^2). \quad (6.13)$$

Only at this point do the details of the process need to be examined. The reflectors $H(w)$ and plane rotations R_{ij} fail to be orthogonal only because, for the computed quantities γ , s , and c , it will turn out that $2 - \gamma(w^*w) \neq 0$ and $s^2 + c^2 - 1 \neq 0$ and (when the arithmetic unit is good)

$$\|\hat{F}_j\| \leq \epsilon = \text{the roundoff unit}. \quad (6.14)$$

We say that H and R_{ij} are orthogonal to working accuracy. Note that any error in w or θ ($s = \sin \theta$) is irrelevant to (6.14).

Some effort is required to bound \bar{W}_j because it depends strongly on the form of B_j . Note that if A_j were simply the result of rounding the exact product $B_j^* A_{j-1} B_j$ we would have

$$\begin{aligned} \|\bar{W}_j\| &\leq \epsilon \|B_j^* A_{j-1} B_j\| \leq \epsilon \|B_j\|^2 \|A_{j-1}\| \\ &\leq \epsilon(1 + \epsilon)^2 \|A_{j-1}\| \text{ using (6.4) and (6.14)}. \end{aligned}$$

In practice the error is a little bigger. Suppose that for each j ,

$$\|\bar{W}_j\| \leq n\epsilon \|A_{j-1}\| \quad (6.15)$$

and

$$\|A_j\| \leq (1 + \epsilon)^3 \|A_{j-1}\|. \quad (6.16)$$

Putting (6.15) and (6.16) into (6.12) yields

$$\|M_0\| \leq (k+1)n\epsilon \|A_0\|. \quad (6.17)$$

Even if the errors did achieve these crude bounds the result is considered satisfactory because of the usual values of k , n , and ϵ . For example, if $k = 99$, $n = 100$, and $\epsilon = 10^{-14}$ then $\|M_0\|/\|A_0\| < 10^{-10}$. If computation with $\epsilon = 10^{-7}$ were compulsory then it would pay to show when the n in (6.15) can be replaced by a constant like 5.

With large sparse matrices ($n > 400$) it is rare to employ a sequence of orthogonal similarity transformations because they tend to make the matrix full and then storage costs escalate. Chapter 13 presents an indirect way to reduce a large sparse matrix to tridiagonal form.

6.6. Backward Error Analysis

Equation (6.10) expresses an exact relationship between the input A_0 and the output A_k ; it is neither approximate nor asymptotic. Yet the reader might be forgiven for thinking that little has been said and the results are trivial. To a certain extent it is this elementary character which underlies their importance. Early attempts to analyze the effect of roundoff error made the task seem both difficult and boring. The new simplicity follows from a point of view developed mainly by Wilkinson in the 1950s and is referred to as *backward error analysis*. This “backward” step was a great advance.

1. No mention has been made of the error! No symbol has been designated for the true “ A_k ” which would have been produced with exact arithmetic.
2. Use of symbols such as \oplus to represent the computer’s addition operation have been abolished. (The author had to struggle with these pseudo-operations when he first studied numerical methods.) Because $=$, $+$, etc. have their ordinary meaning all the nice properties such as associativity and commutativity can be invoked without special mention.
3. Equation (6.10) casts the roundoff error *back* as though it were a perturbation of the data. The effect of all these errors is the same *as if* an initial change M_0 were made in A_0 and then all calculations were done exactly with the orthogonal matrices G_j . If, by chance, M_0 is smaller, element for element, than the uncertainty in A_0 then roundoff error is inconsequential.
4. The (forward) error, $\|A_k - "A_k"\|$, could be large even when M_0 is small relative to A_0 . This may or may not matter, but it can only happen when the function taking A_0 into “ A_k ” is intrinsically sensitive to small changes in A_0 . Intuitively we think of this function as having a large derivative and we say that the calculation is *ill conditioned*. In other words the backward approach distinguishes clearly how the actual (forward) error depends on the algorithm, namely, M_0 , from how it depends on the task itself, namely, its innate sensitivity. It is easy to make the mistake of thinking that big (forward) error means bad method.
5. For eigenvalue calculations the computation of A_k may be an intermediate goal; for example, A_k might be a tridiagonal matrix. In this case M_0 is far *more* relevant than the actual error because the eigenvalues of A_k cannot differ from those of A_0 by more than $\|M_0\|$ however large the error in A_k may be.

6. Forward error analysis (bound the error) is not a defeated rival to backward error analysis. Both are necessary. In particular, users are primarily interested in estimating the accuracy of their output. The combination of a backward analysis (when it can be done) and perturbation theory often gives better bounds than a straightforward attempt to see how roundoff enlarges the intermediate error at each step.

6.7. The QR Factorization and Gram–Schmidt

Any nonnull rectangular m -by- n matrix B can be written as $B = QR$ with m -by- r Q satisfying $Q^*Q = I_r$, and r -by- n R upper triangular with nonnegative diagonal elements; $r = \text{rank}(B)$. Both Q and R are unique when B has full rank.

See Exercise 6.7.3 for the proof in the general case.

Example 6.7.1.

$$\begin{aligned} r = 2 & : \begin{bmatrix} 2 & 1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2/\sqrt{6} & 1/\sqrt{21} \\ -1/\sqrt{6} & 4/\sqrt{21} \\ 1/\sqrt{6} & 2/\sqrt{21} \end{bmatrix} \begin{bmatrix} \sqrt{18} & \sqrt{2} \\ 0 & \sqrt{7} \end{bmatrix} \frac{1}{\sqrt{3}}, \\ r = 1 & : \begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & -6 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix} \begin{bmatrix} 1 & -2 & 3 \end{bmatrix} \sqrt{5}. \end{aligned}$$

The QR factorization is the matrix formulation of the Gram–Schmidt process for orthonormalizing the columns of B in the order b_1, b_2, \dots, b_n . In other words the set $\{q_1, q_2, \dots, q_j\}$ is one orthonormal basis of the subspace spanned by $\{b_1, \dots, b_j\}$ for each $j = 1, 2, \dots, n$; in symbols, $\text{span}(Q_j) = \text{span}(B_j)$. Here we assume that $r = n$.

When B has full rank, i.e., when $r = n$, then R is the (upper) Cholesky factor of B^*B since

$$R^*R = R^*Q^*QR = B^*B.$$

In the full rank case another orthonormal matrix with the same column space as B is $B(B^*B)^{-\frac{1}{2}}$. See Exercise 6.7.6. This is the basis of choice for theoretical analysis.

In finite precision arithmetic it is unwise to form \mathbf{Q} and \mathbf{R} by a blind imitation of the formal Gram–Schmidt process, namely, when $r = n$,

$$\left. \begin{aligned} \tilde{\mathbf{q}}_i &= \mathbf{b}_i - \sum_{k=1}^{i-1} \mathbf{q}_k (\mathbf{q}_k^* \mathbf{b}_i) \\ \mathbf{q}_i &= \tilde{\mathbf{q}}_i / \|\tilde{\mathbf{q}}_i\| \end{aligned} \right\} \quad i = 1, \dots, n.$$

Here are some alternatives.

6.7.1. Modified Gram–Schmidt (MGS)

As soon as \mathbf{q}_j is formed deflate it from all remaining \mathbf{b} 's. Let $\mathbf{b}_i^{(1)} = \mathbf{b}_i, i = 1, \dots, n$, and then, for $j = 1, \dots, n$, form

$$\begin{aligned} \mathbf{q}_j &= \mathbf{b}_j^{(j)} / \|\mathbf{b}_j^{(j)}\|, \\ \mathbf{b}_i^{(j+1)} &= \mathbf{b}_i^{(j)} - \mathbf{q}_j (\mathbf{q}_j^* \mathbf{b}_i^{(j)}), \quad i = j + 1, \dots, n. \end{aligned}$$

This is a rearrangement of the standard Gram–Schmidt process. It is preferable when there is strong cancelation in the subtractions.

6.7.2. Householder's Method

Premultiply \mathbf{B} by a sequence of reflectors \mathbf{H}_i to reduce it to upper-triangular form. At the first step form

$$\mathbf{H}_1 \mathbf{B} = \left[\begin{array}{c|c} r_{11} & \mathbf{r}^* \\ \mathbf{o} & \mathbf{B}^{(2)} \end{array} \right], \quad \mathbf{H}_1 = \mathbf{H}(\mathbf{w}_1), \quad \mathbf{w}_1 = \mathbf{b}_1 \pm \mathbf{e}_1 \|\mathbf{b}_1\|.$$

Then work on $\mathbf{B}^{(2)}$ and continue until

$$\mathbf{H}_n \cdots \mathbf{H}_1 \mathbf{B} = \left[\begin{array}{c} \mathbf{R} \\ \mathbf{O} \end{array} \right],$$

and then

$$\mathbf{Q} = \mathbf{H}_1 \cdots \mathbf{H}_n \mathbf{E}_n,$$

where \mathbf{E}_n denotes the first n columns of \mathbf{I}_m ($n \leq m$).

Another technique, suitable for large m , is given in section 6.8.

The arrival of BLAS 3 (see section 2.8.1) made it desirable to represent the product of a sequence of reflectors $\mathbf{H}_b \mathbf{H}_{b-1} \cdots \mathbf{H}_1$ in a compact form rich in matrix–matrix multiplications. It turns out that such a product can be written as $\mathbf{I} - \mathbf{P}\mathbf{N}\mathbf{P}^*$ where $\mathbf{P}^*\mathbf{P} = \mathbf{I}_b$ and \mathbf{N} is triangular. See [Bischof and van Loan, 1987].

Exercises on Section 6.7

- 6.7.1. How are the coefficients which occur in MGS related to the elements of R ?
- 6.7.2. Show that B^*B is positive definite when $r = n \leq m$.
- 6.7.3. Establish the existence of the QR factorization in the rank deficient case by modifying the Gram-Schmidt process to cope with zero vectors.
- 6.7.4. Find the QR factorization of the three matrices

$$(abc), \quad (bca), \quad (cab)$$

where

$$a = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}.$$

- 6.7.5. Do an operation count for the formation of Q when the vectors w_i which determine the reflections H_i are given.
- 6.7.6. If X is positive definite then $X^{\frac{1}{2}}$ is the unique symmetric positive definite matrix whose square is X . Verify that when the rank of m -by- n B is n , then $B(B^*B)^{\frac{1}{2}}$ exists and is orthonormal.

*6.8. Fast Scaled Rotations

Consider premultiplication of any B by a plane rotation $R(k, j, \theta)$. Only rows k and j of B are changed so that we can simplify our discussion by writing $\xi_i = b_{ki}$, $\eta_i = b_{ji}$, $\gamma = \cos \theta$, $\sigma = \sin \theta$. The task is to compute, for $i = 1, 2, 3, \dots, n$,

$$\begin{aligned} \xi'_i &= \gamma \xi_i - \sigma \eta_i, \\ \eta'_i &= \sigma \xi_i + \gamma \eta_i. \end{aligned} \tag{6.18}$$

A minor drawback of the conventional Givens transformation is the square root needed to compute γ and σ . The major drawback is the four products needed in (6.18) for each i . The goal is to get rid of half these multiplications and the trick is to hold vectors in factored form.

If we restrict attention to just one rotation then no improvement can be made. However in practice many plane rotations are done in sequence as when B is reduced to upper-triangular form as discussed in section 6.7.

To introduce the idea we rewrite (6.18) as a two-phase operation. Let $\tau = \tan \theta$, $\gamma = \cos \theta$. First compute

$$\hat{\xi}_i \equiv \xi_i - \tau \eta_i, \quad \hat{\eta}_i \equiv \tau \xi_i + \eta_i, \quad i = 1, \dots, n, \quad (6.19)$$

and then note that

$$\xi'_i = \gamma \hat{\xi}_i, \quad \eta'_i = \gamma \hat{\eta}_i. \quad (6.20)$$

The payoff comes from *postponing* the execution of (6.20). The price is that the new matrix B' is held in factored form $\Delta \hat{B}$ where the diagonal matrix Δ must hold the multiplier γ in positions k and j . Our exposition below is based on [Hammarling, 1974].

6.8.1. Discarding Multiplications

With the preceding remarks as motivation we begin again. Suppose that B is given in factored form as ΔC with Δ diagonal. Our task is to compute $R(k, j, \theta)B$ in the form $\Delta' C'$ where Δ' may be chosen at our convenience. Thus, looking at rows k and j , we see that

$$\begin{bmatrix} \mu' & 0 \\ 0 & \nu' \end{bmatrix} \begin{bmatrix} \xi'_1 & \xi'_2 & \cdots & \xi'_n \\ \eta'_1 & \eta'_2 & \cdots & \eta'_n \end{bmatrix} = \begin{bmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{bmatrix} \begin{bmatrix} \mu & 0 \\ 0 & \nu \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \\ \eta_1 & \eta_2 & \cdots & \eta_n \end{bmatrix} \quad (6.21)$$

and for each i

$$\begin{aligned} \mu' \xi'_i &= \gamma \mu \xi_i - \sigma \nu \eta_i, \\ \nu' \eta'_i &= \sigma \mu \xi_i + \gamma \nu \eta_i. \end{aligned} \quad (6.22)$$

There are several choices for μ' and ν' which permit ξ'_i and η'_i to be computed with only two multiplications:

$$\mu' = \gamma \mu, \quad \nu' = \gamma \nu; \quad (6.23)$$

$$\mu' = \sigma \nu, \quad \nu' = \sigma \mu; \quad (6.24)$$

$$\mu' = \gamma \mu, \quad \nu' \text{ forces } \gamma \nu / \nu' = \sigma \nu / \mu'; \quad (6.25)$$

$$\mu' = \sigma \nu, \quad \nu' \text{ forces } \gamma \mu / \mu' = \sigma \mu / \nu'. \quad (6.26)$$

6.8.2. Avoiding Square Roots

In order to avoid taking a square root we must utilize the fact that a Givens rotation is specifically designed to annihilate a matrix element, typically η'_1 . In this case (6.22) with $i = 1$ gives $\tau = \tan \theta$ as

$$\tau = \sigma / \gamma = -\nu \eta_1 / \mu \xi_1, \quad (6.27)$$

and, from basic trigonometric identities

$$\gamma^2 = 1/(1 + \tau^2), \quad \sigma^2 = \gamma^2 \tau^2. \quad (6.28)$$

To be specific we choose μ' and ν' by (6.23) above. Then, from (6.22)

$$\begin{aligned} \xi'_i &= \xi_i - \left(\frac{\sigma\nu}{\gamma\mu} \right) \eta_i, \\ \eta'_i &= \left(\frac{\sigma\mu}{\gamma\nu} \right) \xi_i + \eta_i. \end{aligned} \quad (6.29)$$

The key observation is that σ/γ is a multiple of ν/μ . By (6.27)

$$-\frac{\sigma\nu}{\gamma\mu} = \frac{\nu^2 \eta_1}{\mu^2 \xi_1}, \quad -\frac{\sigma\mu}{\gamma\nu} = \frac{\eta_1}{\xi_1}. \quad (6.30)$$

Equation (6.30) shows that μ and ν are not needed and that μ^2 and ν^2 suffice! To prepare for the next Givens rotation we only need

$$(\mu')^2 = \gamma^2 \mu^2, \quad (\nu')^2 = \gamma^2 \nu^2, \quad (6.31)$$

and there is no need to compute γ , σ , or τ . To summarize, note the following statement:

The fast Givens transformation updates Δ^2 and C so that ΔC becomes the rotated matrix.

Using (6.23) the actual algorithm gives

$$\begin{aligned} \alpha &\leftarrow \eta_1/\xi_1, \quad \beta \leftarrow (\nu^2)\alpha/(\mu^2), \quad \omega \leftarrow 1/(1 + \alpha\beta), \\ (\mu^2) &\leftarrow (\mu^2)\omega, \quad (\nu^2) \leftarrow (\nu^2)\omega, \quad \xi_1 \leftarrow \xi_1 + \beta\eta_1, \end{aligned}$$

and, for $i = 2, \dots, n$,

$$\lambda \leftarrow \xi_i, \quad \xi_i \leftarrow \xi_i + \beta\eta_i, \quad \eta_i \leftarrow \eta_i - \alpha\lambda. \quad (6.32)$$

Example 6.8.1 (fast Givens).

$$\begin{bmatrix} \Delta^2 \\ \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} C \\ \begin{bmatrix} 5 & 2 & 4 & 6 \\ 2 & 3 & 7 & 5 \end{bmatrix} \end{bmatrix}$$

$$\alpha \leftarrow \eta_1/\xi_1 = 2/5 = 0.4, \quad \beta \leftarrow (\nu^2)\alpha/(\mu^2) = 4(0.4)/1 = 1.6,$$

$$\omega \leftarrow 1/(1 + \alpha\beta) = 1/(1 + (0.4)(1.6)) = 0.61,$$

$$\mu'^2 \leftarrow (\mu^2)\omega = 1(0.61) = 0.61, \quad \nu'^2 \leftarrow (\nu^2)\omega = 4(0.61) = 2.44,$$

ξ'_i and η'_i are calculated by using (6.32).

$$\begin{array}{c} \Delta'^2 \\ \left[\begin{array}{cc} 0.61 & 0 \\ 0 & 2.44 \end{array} \right] \end{array} \quad \begin{array}{c} C' \\ \left[\begin{array}{cccc} 8.2 & 6.8 & 15.2 & 14. \\ 0 & 2.2 & 5.4 & 2.6 \end{array} \right] \end{array}$$

The elements of Δ can never increase but there is a possibility of underflow. To forestall such a calamity it is best not to settle on any one formula in (6.23)–(6.26) but rather to let the program choose, say, (6.23) when $\nu^2|\eta_1| \leq \mu^2|\xi_1|$ and (6.24) otherwise, to ensure that $(\mu')^2 \geq (\mu^2)/2$.

6.8.3. Error Analysis

Algorithms such as (6.32) are robust in the face of roundoff. This is to be expected because (6.32) is just a scaled version of the standard Givens rotation, and products are always computed with tiny relative error (barring underflow or overflow).

The situation is simple enough to warrant presentation. In order to suppress distracting details let ϵ denote a tiny number (e.g., $|\epsilon| = 10^{-14}$) which *may be different at every appearance*. Thus $(1 + \epsilon)^2$ is shorthand for $(1 + \epsilon_1)(1 + \epsilon_2)$. We denote computed quantities by overbars, $\bar{\alpha}$, $\bar{\beta}$, etc., and make the realistic assumption that in executing (6.32),

$$\begin{aligned} \bar{\alpha} &= \alpha(1 + \epsilon), & \bar{\beta} &= \beta(1 + \epsilon), \\ \bar{\omega} &= \omega(1 + \epsilon)^2 = (1 + \epsilon)^2/(1 + \tau^2) = [(1 + \epsilon)\gamma]^2. \end{aligned} \quad (6.33)$$

We worked hard to remove γ and σ from the computation in (6.32) but for analysis we want to bring them back. To do this we define $\bar{\mu}'$, which is never seen, as the exact square root of the new value of the variable μ^2 and assume realistically that

$$\bar{\mu}' = (1 + \epsilon)\mu\gamma, \quad \bar{\nu}' = (1 + \epsilon)\nu\gamma. \quad (6.34)$$

Now we are ready to consider the main computation in (6.32), where

$$\bar{\xi}'_i = [\xi_i + \bar{\beta}\eta_i(1 + \epsilon)](1 + \epsilon),$$

because of the multiply and the add operations. Next multiply by $\bar{\mu}'$ and use (6.34), (6.33), and (6.30) to find

$$\begin{aligned} \bar{\mu}'\bar{\xi}'_i &= \gamma\mu\xi_i(1 + \epsilon)^2 + \gamma\mu(\sigma\nu/\gamma\mu)\eta_i(1 + \epsilon)^3, \\ \bar{\nu}'\bar{\eta}'_i &= -(\sigma/\gamma)\gamma\nu\xi_i(1 + \epsilon)^3 + \gamma\nu\eta_i(1 + \epsilon)^2. \end{aligned} \quad (6.35)$$

Replace $(1 + \epsilon)^k$ by $1 + k\epsilon$ (remember ϵ 's mercurial nature) and recall (6.22) to obtain the desired relation

$$\begin{bmatrix} \bar{\mu}'\bar{\xi}'_i \\ \bar{\nu}'\bar{\eta}'_i \end{bmatrix} = \begin{bmatrix} \mu'\xi'_i \\ \nu'\eta'_i \end{bmatrix} + \begin{bmatrix} 2\epsilon\gamma & 3\epsilon\sigma \\ -3\epsilon\sigma & 2\epsilon\gamma \end{bmatrix} \begin{bmatrix} \mu\xi_i \\ \nu\eta_i \end{bmatrix}. \quad (6.36)$$

This equation is of exactly the same form as the equation for the errors incurred in a standard Givens rotation. The new formulas are as stable as the old ones—and underflow is avoided by allowing flexibility in the choice of formulas in (6.23).

6.8.4. Accumulating the Product

In some applications the product of all the plane rotations is needed. Normally this product is accumulated gradually; at each step the current product is premultiplied by the current plane rotation. In this situation the values of γ and σ are needed, and it looks as though the square root must be taken after all! Of course it is still desirable to keep the two-multiplication formulas for updating Δ^2 and C .

Once again it turns out that the square root is not necessary. By computing a slightly unorthodox variant of the QR decomposition, a scaled version of Q can be updated at each step without taking a square root. Let us illustrate how this is done with the (6.23) choice of scaling. Denote by Q the current orthogonal matrix. The nontrivial part of the Givens rotation has been written as

$$R = \gamma G = \begin{bmatrix} \gamma & 0 \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} 1 & -\tau \\ \tau & 1 \end{bmatrix}.$$

Suppose Q is scaled so that the two rows to be modified are

$$Q = \Psi S = \begin{bmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{bmatrix} \begin{bmatrix} \dots & \dots & \cdot \\ \dots & \dots & \cdot \end{bmatrix}.$$

To preserve the two-multiplication update we must permute G and Ψ . Thus

$$\begin{aligned} RQ &= \gamma G \Psi S \\ &= \gamma (\Psi \Psi^{-1}) G \Psi S \\ &= (\gamma \Psi) (\bar{G} S), \end{aligned}$$

where

$$\bar{G} \equiv \Psi^{-1} G \Psi = \begin{bmatrix} 1 & -\tau\rho \\ \tau/\rho & 1 \end{bmatrix}, \quad \rho = \psi_2/\psi_1,$$

has the same desirable form as G . For arbitrary positive ψ_1, ψ_2 this permutation of G and Ψ can be done and the savings in multiplications in the formation of $\bar{G}S$ can be achieved. However, the quantities γ, τ, μ , and ν are not directly available from the algorithm in (6.32) and the formation of $\gamma\Psi$ seems to require the square root of ω . Fortunately the choice $\psi_1 = \mu, \psi_2 = \nu$ (i.e., $\Psi = \Delta$) removes this small blemish because, from (6.27), (6.31), and (6.32),

$$\begin{aligned}\tau\rho &= -(\nu\eta_1/\mu\xi_1)(\nu/\mu) = -\beta = \text{known}, \\ \tau/\rho &= -(\nu\eta_1/\mu\xi_1)(\mu/\nu) = -\alpha = \text{known}, \\ \gamma\psi_1 &= \gamma\mu = \mu', \quad \gamma\psi_2 = \gamma\nu = \nu'.\end{aligned}$$

The result is that exactly the same scale factors are used for B as for the transforming matrix Q . In symbols,

$$\begin{aligned}B &= \Delta C \rightarrow B' = RB = \Delta'C', \\ Q &= \Delta S \rightarrow Q' = RQ = \Delta'S',\end{aligned}$$

where the work is done in forming

$$C' = \bar{G}C, \quad S' = \bar{G}S, \quad (\Delta')^2 = \omega\Delta^2,$$

and

$$\bar{G} = \begin{bmatrix} 1 & \beta \\ -\alpha & 1 \end{bmatrix}.$$

This was an example of elegance in applied mathematics.

*6.9. Orthogonalization in the Face of Roundoff

The simplest orthogonalization calculation, to which all others can be reduced, is this: given vectors $y \neq 0$ and z , produce z 's component p orthogonal to y . The formula is

$$p = z - y(y^*z/y^*y) = [I - y(y^*y)^{-1}y^*]z. \quad \text{See Figure 6.2.}$$

Roundoff complicates matters in two ways. First, we can only compute an approximation λ to y^*z/y^*y . Second, we can only compute an approximation x to $p = z - \lambda y$. In particular, when z is almost parallel to y the latter approximation will largely cancel, leaving a handful of rounding errors in place of p as shown in Example 6.9.1.

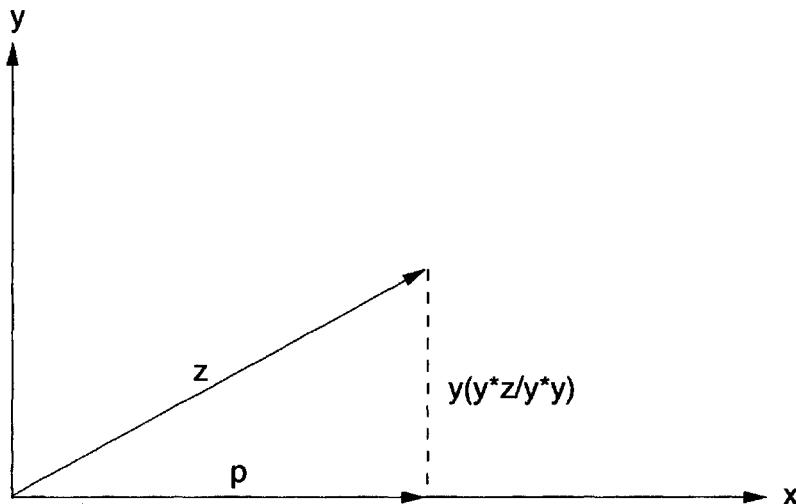


FIG. 6.2. Orthogonality in the face of roundoff.

Example 6.9.1. The vertical rule separates the figures which remain from those which will be discarded.

$$\text{Given } \mathbf{y} = \begin{bmatrix} 0.3179 \\ 0.0253 \\ 0.0082 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} 0.3170 \\ 0.0258 \\ 0.0085 \end{bmatrix},$$

$$\text{Compute } \lambda = \frac{\mathbf{y}^* \mathbf{z}}{\mathbf{y}^* \mathbf{y}} = 0.9973|3707,$$

$$\lambda \mathbf{y} = \left[\begin{array}{c|c} 0.3170 & 5346 \\ 0.0252 & 3263 \\ 0.0081 & 7816 \end{array} \right],$$

$$\mathbf{p} \equiv \mathbf{z} - (\lambda \mathbf{y})_8 = \left[\begin{array}{c} -0.00005346 \\ 0.00056737 \\ 0.00032184 \end{array} \right], \quad \frac{\mathbf{y}^* \mathbf{p}}{\|\mathbf{y}\| \|\mathbf{p}\|} = -6.634 \times 10^{-6},$$

$$\mathbf{x} \equiv \mathbf{z} - (\lambda \mathbf{y})_4 = \left[\begin{array}{c} 0.0000 \\ 0.0006 \\ 0.0004 \end{array} \right], \quad \frac{\mathbf{y}^* \mathbf{x}}{\|\mathbf{y}\| \|\mathbf{x}\|} = 0.0802.$$

Most applications of orthogonalization require that \mathbf{p} 's approximation \mathbf{x} have two properties:

I. x is orthogonal to y to working accuracy.

II. z is very nearly a linear combination of x and y .

Then x is very nearly the component, orthogonal to y , of a vector very near z .

How close would we expect to come to satisfying these conditions if we naively compute

$$\lambda = y^* z / \|y\|^2 \text{ with roundoff, } x = z - \lambda y \text{ with roundoff?}$$

Then we shall have, as it turns out,

$$|\lambda - y^* z / \|y\|^2| \leq \delta_1 \|z\| / \|y\| \quad \text{and} \quad \|x - (z - \lambda y)\| \leq \delta_2 \|z\|$$

for some tiny positive δ_1 and δ_2 dependent on the precision carried and other computational details. The latter inequality satisfies property II above. But since

$$e \equiv x - p = x - (z - \lambda y) - (\lambda - y^* z / \|y\|^2)y,$$

then

$$\|e\| \leq (\delta_1 + \delta_2) \|z\|,$$

and we find

$$|y^* x| = |y^* e| \leq \|y\| \cdot \|e\| \leq (\delta_1 + \delta_2) \|y\| \cdot \|x\| \cdot (\|z\| / \|x\|)$$

so property I may fail if $\|z\| / \|x\|$ is huge, as will happen whenever z and y are nearly parallel.

Nonetheless it is possible to compute an x which satisfies both I and II, despite roundoff, though at a price. The following ("twice is enough") algorithm and analysis are due to W. Kahan.

Suppose that a simple subprogram *orthog* is available which, given $y \neq 0$ and z , computes an approximation x' to $p \equiv z - y(y^* z / \|y\|^2)$. Let the error $e' (\equiv x' - p)$ satisfy $\|e'\| \leq \epsilon \|z\|$ for some tiny positive ϵ independent of y and z . Let κ be any fixed value in the range $[1/(0.83 - \epsilon), 0.83/\epsilon]$.

Algorithm. First call *orthog*(y, z, x') to get x' .

Case 1. If $\|x'\| \geq \|z\|/\kappa$ accept $x = x'$ and $e = e'$.

Otherwise call *orthog*(y, x', x'') to get x'' with error $e'' \equiv x'' - (x' - yy^* x' / \|y\|^2)$ satisfying $\|e''\| \leq \epsilon \|x'\|$ and proceed to Case 2.

Case 2. If $\|x''\| \geq \|x'\|/\kappa$ accept $x = x''$, $e = x'' - p$.

Case 3. If $\|x''\| < \|x'\|/\kappa$ accept $x = 0$, $e = -p$.

Note that when κ is small, like 1.25, the bounds are very good, but Case 1

will be rarer and the algorithm will be more expensive than when $\kappa = 100$. In some applications it is preferable to take for x an arbitrary nonzero vector orthogonal to y instead of o .

LEMMA. The vector x computed by the algorithm ensures that $\|e\| \leq (1 + 1/\kappa)\epsilon\|z\|$ and $|y^*x| \leq \kappa\epsilon\|y\|\|x\|$.

Proof. We examine the three cases which can occur in the algorithm.

Case 1.

$$\|e\| = \|e'\| \leq \epsilon\|z\| \leq \left(1 + \frac{1}{\kappa}\right) \epsilon\|z\|,$$

$$|y^*x| = |y^*x'| = |y^*e'| \leq \|y\| \cdot \|e'\| \leq \epsilon\|y\| \cdot \|z\| \leq \kappa\epsilon\|y\| \cdot \|x\|.$$

Case 2.

$$\begin{aligned} |y^*x| &= |y^*x''| = |y^*e''| \leq \|y\| \cdot \|e''\| \leq \epsilon\|y\| \cdot \|x'\| \\ &\leq \kappa\epsilon\|y\| \cdot \|x\|. \end{aligned}$$

Substitute for x'' and then x' and use y^*p to find

$$\begin{aligned} \|e\| &= \|x'' - p\| = \|p + (1 - yy^*/\|y\|^2)e' + e'' - p\| \\ &\leq \|(1 - yy^*/\|y\|^2)e'\| + \|e''\| \\ &\leq \|e'\| + \|e''\| \leq \epsilon\|z\| + \epsilon\|x'\| \leq \epsilon \left(1 + \frac{1}{\kappa}\right) \|z\|. \end{aligned}$$

Case 3. Since $x = o$, $|y^*x| = 0 \leq \kappa\epsilon\|y\| \cdot \|x\|$. Next we must infer that $e = -p$ satisfies $\|p\| \leq (1 + \frac{1}{\kappa})\epsilon\|z\|$. Actually we shall infer more, namely, $\|p\| < [1 - (\epsilon + \kappa^{-1})^2]^{-\frac{1}{2}}\|e'\|$, from which the desired result will follow because $\|e'\| \leq \epsilon\|z\|$ and $[1 - (\epsilon + \kappa^{-1})^2]^{-\frac{1}{2}} \leq 1 + \kappa^{-1}$, the last inequality being the constraint that inspired and is implied by the lemma's simpler bounds on κ . See Exercise 6.9.2.

Write $e' = a + b$ where $a = (1 - yy^*/\|y\|^2)e'$ and $b = yy^*e'/\|y\|^2$. Then, with the aid of Exercise 6.9.1 and Case 1,

$$\begin{aligned} \kappa^{-1}\|x'\| &> \|x''\| = \|p + a + e''\| \\ &\geq \|p + a\| - \|e''\| \geq \|p + a\| - \epsilon\|x'\|, \end{aligned}$$

so that

$$(\epsilon + \kappa^{-1})\|x'\| = (\epsilon + \kappa^{-1})\|\mathbf{p} + \mathbf{a} + \mathbf{b}\| > \|\mathbf{p} + \mathbf{a}\|.$$

Since $\mathbf{b}^*\mathbf{p} = \mathbf{b}^*\mathbf{a} = 0$, squaring yields

$$(\epsilon + \kappa^{-1})^2(\|\mathbf{p} + \mathbf{a}\|^2 + \|\mathbf{b}\|^2) > \|\mathbf{p} + \mathbf{a}\|^2,$$

whence

$$\begin{aligned}\|\mathbf{p}\| &\leq \|\mathbf{p} + \mathbf{a}\| + \|\mathbf{a}\| < \|\mathbf{a}\| + \|\mathbf{b}\|(\epsilon + \kappa^{-1})/\sqrt{1 - (\epsilon + \kappa^{-1})^2} \\ &< \sqrt{1 + (\epsilon + \kappa^{-1})^2/[1 - (\epsilon + \kappa^{-1})^2]} \sqrt{\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2} \\ &= \|\mathbf{e}'\|/\sqrt{1 - (\epsilon + \kappa^{-1})^2}\end{aligned}$$

as claimed. \square

Example 6.9.2 shows the algorithm in action.

Example 6.9.2. $\kappa = 10$. $\epsilon = 10^{-5}$ (from Example 2.5.3)

$$\begin{bmatrix} \mathbf{y} \\ 0.16087 \\ -0.11852 \\ 0.98216 \times 10^{-1} \end{bmatrix} \quad \begin{bmatrix} \mathbf{z} \\ -0.50069 \times 10^{-1} \\ 0.36889 \times 10^{-1} \\ -0.30569 \times 10^{-1} \end{bmatrix} \quad \begin{bmatrix} \mathbf{x}' (= \mathbf{z} - \lambda\mathbf{y}) \\ -0.20000 \times 10^{-5} \\ 0.30000 \times 10^{-5} \\ -0.20000 \times 10^{-5} \end{bmatrix}$$

$$\|\mathbf{y}\| = 0.22264 \quad \|\mathbf{z}\| = 0.69297 \times 10^{-1} \quad \|\mathbf{x}'\| = 0.41231 \times 10^{-5}$$

$$\begin{bmatrix} \mathbf{x}'' (= \mathbf{x}' - \lambda'\mathbf{y}) \\ 0.8353 \times 10^{-6} \\ 0.9110 \times 10^{-6} \\ -0.2689 \times 10^{-6} \end{bmatrix} \quad \begin{bmatrix} \mathbf{p} \\ 0.48705 \times 10^{-6} \\ 0.60812 \times 10^{-6} \\ -0.63987 \times 10^{-8} \end{bmatrix} \quad \begin{bmatrix} \mathbf{e} (= \mathbf{x}'' - \mathbf{p}) \\ 0.34825 \times 10^{-6} \\ 0.30288 \times 10^{-6} \\ -0.20431 \times 10^{-6} \end{bmatrix}$$

$$\|\mathbf{x}''\| = 0.12648 \times 10^{-5} \geq 0.41231 \times 10^{-6} = \|\mathbf{x}'\|/\kappa,$$

$$\|\mathbf{e}\| = 0.50498 \times 10^{-6} \leq 0.76227 \times 10^{-6} = (1 + 1/\kappa)\epsilon\|\mathbf{z}\|,$$

$$|(\mathbf{x}'')^*\mathbf{y}| = 0.72914 \times 10^{-11} \leq 0.28160 \times 10^{-10} = \kappa\epsilon\|\mathbf{x}''\|\|\mathbf{y}\|.$$

Exercises on section 6.9

6.9.1. Show that $\mathbf{x}'' = \mathbf{e}'' + \mathbf{a} + \mathbf{p}$.

- 6.9.2. Solve $1 = (1 + \mu)^2(1 - \mu^2)$ for $\mu > 0$. A calculator may help. For what values of μ is $[1 - (\epsilon + \mu)^2](1 + \mu)^2 \geq 1$?

Notes and References

Plane rotations were used in [Jacobi, 1846] and became well-known tools after their rediscovery in [Bargmann, Montgomery, and von Neumann, 1946]. It is odd that reflectors (or symmetries as they are called by group theorists), which are the only elementary matrices that can generate all orthogonal matrices, were not introduced as tools for computation until 1958 in [Householder, 1958]. Their rapid acceptance owes much to [Wilkinson, 1960]. The stable formulas for reflections in internal bisectors were given in [Parlett, 1971] but were ignored in the United States.

[Dubrulle, 1996] shows that in the delicate situation when the input and output vectors are close then reflectors in the internal bisector are preferable. In his formulation Dubrulle shows that these reflectors carry the important information better than the popular ones.

The possibility of avoiding half the multiplications in a sequence of plane rotations by adroit use of scaling was presented in [Gentleman, 1973]. Our presentation follows [Hammarling, 1974]. Very similar ideas are used in [Gill, Murray, and Saunders, 1975] where the Gram–Schmidt matrix Q in $B = QR$ is itself held in scaled form. Mention should also be made of recent work on updating Q when B changes a little; in particular, see [Gill, Golub, Murray, and Saunders, 1974]. Section 6.9 is based on unpublished notes by Kahan, but the fact that “twice is enough” when orthogonalizing in the presence of roundoff is not as well known as it should be. The results in [Daniel et al., 1976] can be regarded as extensions of this analysis to the case of orthogonalizing against several nearly orthonormal vectors at the same time.

The error analysis in sections 6.5 and 6.6 is based on [Wilkinson, 1965, Chapter 3].

Tridiagonal Form

7.1. Introduction

This chapter is concerned with the reduction of an arbitrary symmetric matrix A to a similar tridiagonal matrix T and also with the special properties which T enjoys.

Definition 7.1.1. T is tridiagonal if $T_{ij} = 0$ whenever $|i - j| > 1$.

In order to simplify notation it is useful to write $t_{ii} = \alpha_i$, $t_{i,i+1} = \beta_i$ (sometimes it is more convenient to set $t_{i,i+1} = \beta_{i+1}$), and

$$T_{\mu:\nu} = \begin{bmatrix} \alpha_\mu & \beta_\mu & & & \\ \beta_\mu & \alpha_{\mu+1} & \beta_{\mu+1} & & \\ & \beta_{\mu+1} & & & \\ & & \alpha_{\nu-1} & \beta_{\nu-1} & \\ & & \beta_{\nu-1} & \alpha_\nu & \end{bmatrix}. \quad (7.1)$$

Definition 7.1.2. $T (= T_{1:n})$ is unreduced if $\beta_i \neq 0$, $i = 1, \dots, n - 1$.

If, for some k , $\beta_k = 0$ then T is a direct sum of two tridiagonal matrices, say, T_1 of order k and T_2 of order $n - k$. The eigenvalues and eigenvectors of T can be recovered easily from those of T_1 and T_2 (Exercise 7.1.1), and in this way the computation may be reduced to finding the eigenvalues and eigenvectors of smaller tridiagonal submatrices which are unreduced. So there is no loss of generality in confining attention to the unreduced case.

The key facts are as follows:

1. The eigenvalues and eigenvectors of T can be found with significantly fewer arithmetic operations than are required for a full A .

2. Every A can be reduced to a similar T by a finite number of elementary orthogonal similarity transformations. In principle an infinite number of such transformations are needed to diagonalize a matrix.
3. If T is unreduced its eigenvalues are distinct (though they may be very close) and its eigenvectors enjoy some useful special properties. See sections 7.7 and 7.9.

The tridiagonal form is not always the way to go; the smaller the bandwidth of a matrix the larger is the cost of reduction to tridiagonal form *relative* to the total calculation of eigenvalues and/or eigenvectors by other means.

Tridiagonal matrices sometimes occur as primary data. For example, they are associated with families of orthogonal polynomials and with special functions, like the Bessel functions J_m , which satisfy three-term recurrence relations. For example, the zeros of J_m are given by $2/\sqrt{\mu_k}$, $k = 1, 2, \dots$, where the μ_k are the eigenvalues of $T_{1:\infty}$ and

$$\begin{aligned}\alpha_i &= 2/(m + 2i - 1)(m + 2i + 1), \\ \beta_i^2 &= 1/(m + 2i)(m + 2i + 1)^2(m + 2i + 2).\end{aligned}$$

Exercise on Section 7.1

7.1.1. Let $T = \text{diag}(T_1, T_2)$ and let

$$T_i = S_i \Theta_i S_i^*$$

be the spectral decomposition of T_i , $i = 1, 2$. What is the spectral decomposition of T ? Suppose θ is an eigenvalue of T_1 and of T_2 . Describe the eigenvectors of T belonging to θ .

7.2. Reduction Parameters

There are several ways to reduce an arbitrary A to tridiagonal form T , but before discussing any of them in detail it is good to know to what extent the resulting T depends on the method. The answer is given in the theorem below, but before presenting it some further normalization is necessary.

Lemma 7.2.1. Let T be unreduced and let T_+ be the matrix obtained by replacing each β_i by $|\beta_i|$, $i = 1, \dots, n - 1$. Then

$$T_+ = \Delta T \Delta = \Delta T \Delta^{-1}, \quad (7.2)$$

where $\Delta = \text{diag}(\delta_1, \dots, \delta_n)$ and $\delta_i = \pm 1$.

The proof is left as Exercise 7.2.1. The lemma shows that there is no loss of generality in taking $\beta_i > 0$, $i = 1, \dots, n - 1$, although the eigenvectors suffer sign changes.

Theorem 7.2.1. Let $Q^* A Q = T_+$ with Q orthogonal. Then T_+ and $Q \equiv (q_1, q_2, \dots, q_n)$ are uniquely determined by A and q_1 or by A and q_n .

Proof. Since $Q^* = Q^{-1}$ the hypothesis can be rewritten as

$$QT_+ = AQ. \quad (7.3)$$

Now equate the j th column on each side of (7.3), use the fact that the j th column of T_+ has only three nonzero elements, and rearrange terms to find an important relation

$$q_{i+1}\beta_j = Aq_j - q_j\alpha_j - q_{j-1}\beta_{j-1} \equiv r_j. \quad (7.4)$$

This holds for $j = 1, \dots, n$ if we define $\beta_0 = \beta_n = 0$. Thus $r_n = 0$ and q_0, q_{n+1} are undefined. Next use the orthogonality of Q in the form $q_i^* q_k = \delta_{ik}$ (Kronecker's δ symbol) to obtain

1. $0 = q_j^*(q_{j+1}\beta_j) = q_j^* A q_j - 1 \cdot \alpha_j - 0 \cdot \beta_{j-1}$,
2. $\beta_j = \|q_{j+1}\beta_j\| = \|r_j\|$,
3. $q_{j+1} = r_j/\beta_j$, since $\beta_j > 0$ by hypothesis.

Thus β_{j-1}, q_{j-1} , and q_j determine, in turn, α_j, r_j, β_j , and q_{j+1} for $j = 1, 2, \dots, n$. Since $\beta_0 = 0$, q_1 alone determines α_1, r_1, β_1 , and q_2 , and

so, by finite induction, \mathbf{q}_1 determines uniquely all the elements of T_+ and Q .

By rewriting (7.4) in the form

$$\mathbf{q}_{j-1}\beta_{j-1} = \mathbf{A}\mathbf{q}_j - \mathbf{q}_j\alpha_j - \mathbf{q}_{j+1}\beta_j \equiv \tilde{\mathbf{r}}_j$$

it can be shown that β_j , \mathbf{q}_{j+1} , and \mathbf{q}_j determine α_j , $\tilde{\mathbf{r}}_j$, β_{j-1} . Since $\beta_n = 0$, \mathbf{q}_n also determines T_+ and Q uniquely. \square

7.2.1. Remarks

1. We have *not* shown that for a given \mathbf{A} each unit vector \mathbf{q}_1 determines a unique T_+ and Q . The uniqueness breaks down whenever some $r_j = 0$. Then $\beta_j = \|\mathbf{r}_j\| = 0$ and T is reduced. This rare case of breakdown is not a curse but a blessing. Breakdown can occur only if \mathbf{q}_1 is orthogonal to at least one eigenvector of \mathbf{A} .

If $\beta_j = 0$ then \mathbf{q}_{j+1} can be taken as any unit vector orthogonal to the preceding \mathbf{q} 's and the process can then be continued. Only uniqueness has been lost.

2. In the proof β_{j-1} was defined by $\beta_{j-1} = \|\mathbf{r}_{j-1}\|$. However, if (7.4) is premultiplied by \mathbf{q}_{j-1}^* it is clear that

$$\beta_{j-1} = \mathbf{q}_{j-1}^* \mathbf{A} \mathbf{q}_j. \quad (7.5)$$

These two expressions for β_{j-1} are equivalent in exact arithmetic but turn out to be very different when used by digital computers. It turns out (in Chapter 13) that (7.5) is to be avoided, but this result is far from obvious and indeed (7.5) has often been recommended.

3. There is no suggestion that T and Q should necessarily be computed by the procedure used in the proof which happens to be the *Lanczos algorithm*. See Chapter 13. The choice of method depends on, among other things, whether all the eigenvalues of \mathbf{A} are wanted or only a few.

Exercises on Section 7.2

- 7.2.1. Prove Lemma 7.2.1. Given an eigenvector \mathbf{s} of T_+ and Δ give an algorithm for overwriting \mathbf{s} with the corresponding eigenvector of T .

7.2.2. Let

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Reduce \mathbf{A} to tridiagonal form by following the proof of Theorem 7.2.1.
Try two starting vectors \mathbf{e}_1 and \mathbf{e}_3 .

7.2.3. Reduce $\text{diag}(1, 10, 100)$ to tridiagonal form starting with $(1, 1, 1)^*/\sqrt{3}$.

7.3. Minimizing Characteristics

Let $\mathbf{T} = \mathbf{T}_+ = \mathbf{Q}^* \mathbf{A} \mathbf{Q}$. Since \mathbf{T} and \mathbf{Q} are determined by $\mathbf{q}_1 (= \mathbf{Q}\mathbf{e}_1)$ it is plausible that the formulas for at least some of the elements of \mathbf{T} and \mathbf{Q} are nice and simple. This section presents some such formulas, and their derivation is little more than a systematic exploitation of the tridiagonal form. Consider, for instance, a 4-by-4 example:

$$\begin{aligned} \mathbf{T}\mathbf{e}_1 &= \begin{bmatrix} \alpha_1 \\ \beta_1 \\ 0 \\ 0 \end{bmatrix}, & \mathbf{T}^2\mathbf{e}_1 &= \begin{bmatrix} \alpha_1^2 + \beta_1^2 \\ \beta_1(\alpha_1 + \alpha_2) \\ \beta_1\beta_2 \\ 0 \end{bmatrix}, \\ \mathbf{T}^3\mathbf{e}_1 &= \begin{bmatrix} \alpha_1^3 + (2\alpha_2 + \alpha_3)\beta_1^2 \\ \beta_1(\alpha_1^2 + \alpha_1\alpha_2 + \alpha_2^2 + \beta_1^2 + \beta_2^2) \\ \beta_1\beta_2(\alpha_1 + \alpha_2 + \alpha_3) \\ \beta_1\beta_2\beta_3 \end{bmatrix}. \end{aligned}$$

It turns out that the character of the first columns of the powers of \mathbf{T} gives a bizarre but useful characterization of the β 's and \mathbf{q} 's in terms of certain polynomials.

Recall from section 7.1 that the leading principal j -by- j submatrix of \mathbf{T} may be written $\mathbf{T}_{1:j}$. Its characteristic polynomial is χ_j , so

$$\chi_j(\xi) \equiv \det[\xi - \mathbf{T}_{1:j}], \quad j = 1, \dots, n.$$

For convenience let $\chi_0(\xi) = 1$ and let \mathcal{MP}_k denote the set of monic polynomials of degree k .

Theorem 7.3.1. *Let tridiagonal T have positive subdiagonal elements as given in (7.1). Then, for $j = 1, \dots, n - 1$,*

$$\begin{aligned}\beta_1\beta_2\cdots\beta_j &= \|\chi_j(T)\mathbf{e}_1\| = \min \|\psi(T)\mathbf{e}_1\| \text{ over all } \psi \in \mathcal{MP}_j, \\ \mathbf{e}_{j+1} &= \chi_j(T)\mathbf{e}_1 / (\beta_1\cdots\beta_j).\end{aligned}$$

Proof. Observe that the last nonzero element in $T^j\mathbf{e}_1$ is $(\beta_1\cdots\beta_j)$ and is in row $j + 1$ (Exercise 7.3.1). It follows that the same is true for $\phi(T)\mathbf{e}_1$ for any ϕ in \mathcal{MP}_j because T^j is the only term which contributes a nonzero value in row $j + 1$. Hence

$$\beta_1\beta_2\cdots\beta_j \leq \|\phi(T)\mathbf{e}_1\|,$$

with equality only if $\phi(T)\mathbf{e}_1 = \mathbf{e}_{j+1}(\beta_1\cdots\beta_j)$.

A closer look at the first j elements of $\phi(T)\mathbf{e}_1$, or rather of the $T^k\mathbf{e}_1$, $k < j$, shows that they involve $T_{1:j}$ only. More precisely, let $E_j = (\mathbf{e}_1, \dots, \mathbf{e}_j)$; then (Exercise 7.3.2), for any $\phi \in \mathcal{MP}_j$,

$$E_j^* \phi(T) \mathbf{e}_1 = \phi(T_{1:j}) \mathbf{e}_1. \quad (7.6)$$

Note that \mathbf{e}_1 on the left is in \mathcal{E}^n while \mathbf{e}_1 on the right is in \mathcal{E}^j . By the Cayley–Hamilton theorem $\chi_j(T_{1:j}) = \mathbf{0}$ and so $E_j^* \chi_j(T) \mathbf{e}_1 = \mathbf{0}$. By the second sentence in this proof $\chi_j(T)\mathbf{e}_1$ must be $\mathbf{e}_{j+1}(\beta_1\cdots\beta_j)$ and thus both assertions are established. \square

Of course, with T in front of us there is little incentive to characterize the β 's or the \mathbf{e}_i at all. However, we can substitute Q^*AQ for T to get expressions involving A and \mathbf{q}_1 . See Exercise 7.3.3.

Corollary 7.3.1. *Let $T = T_+ = Q^*AQ$, with $Q = (\mathbf{q}_1, \dots, \mathbf{q}_n)$ orthogonal. Then, for $j = 1, \dots, n - 1$,*

$$\beta_1\cdots\beta_j = \|\chi_j(A)\mathbf{q}_1\| = \min \|\psi(A)\mathbf{q}_1\| \text{ over all } \psi \in \mathcal{MP}_j, \quad (7.7)$$

$$\mathbf{q}_{j+1} = \chi_j(A)\mathbf{q}_1 / (\beta_1\cdots\beta_j). \quad (7.8)$$

There are situations in which A and q_1 are known but T and Q are not. Nor is χ_j known, but it is interesting that the unknown β 's must have the minimal property (7.7). When A is fixed then χ_j is said to be the minimal polynomial for q_1 in \mathcal{MP}_j . The $\{\chi_j\}$ are sometimes called the Lanczos polynomials for q_1 .

Exercises on Section 7.3

- 7.3.1. By induction, or otherwise, show that $e_k^* T^j e_1 = 0$ if $k > j + 1$, or $e_k^* T^j e_1 = \beta_1 \cdots \beta_j$ if $k = j + 1$.
- 7.3.2. By induction, or otherwise, show that for $k \leq j$,

$$E_j^* T^k e_1 = T_{1:j}^k e_1.$$

Then conclude that $E_j^* \phi(T) e_1 = \phi(T_{1:j}) e_1$ for all ϕ in \mathcal{MP}_j .

- 7.3.3. Establish Corollary 7.3.1 from Theorem 7.3.1.
- 7.3.4. Find an expression for α_j in terms of q_1 and polynomials in A .
- 7.3.5. Prove Corollary 7.3.1 directly by equating the j th column on each side of $AQ = QT$ and then using the fact that the χ_j satisfy a certain three-term recurrence relation to show that q_{j+1} must be a multiple of $\chi_j(A)q_1$.
- 7.3.6. Let

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 8 \end{bmatrix} \quad \text{and} \quad q_1 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} / \sqrt{2}.$$

Find an upper bound on β_1 and $\beta_1\beta_2$. Use the method of proof of Theorem 7.2.1 to compute, in order, $\alpha_1, \beta_1, q_2, \alpha_2$, and β_2 . Exhibit χ_1 and χ_2 .

7.4. Explicit Reduction of a Full Matrix

Any A can be reduced to tridiagonal form T by a sequence of simple orthogonal similarity transformations which introduce zero elements column by column. The first step is typical and can be described most simply by partitioning A as shown below.

$$A = A_1 = \begin{bmatrix} \alpha_1 & c_1^* \\ c_1 & M_1 \end{bmatrix}, \quad \alpha_1 = a_{11}.$$

Now consider any orthogonal similarity transformation of A_1 which leaves the $(1, 1)$ element unaltered.

$$\begin{aligned}\hat{A}_1 &= \begin{bmatrix} 1 & 0^* \\ 0 & P_1^* \end{bmatrix} \begin{bmatrix} \alpha_1 & c_1^* \\ c_1 & M_1 \end{bmatrix} \begin{bmatrix} 1 & 0^* \\ 0 & P_1 \end{bmatrix} \\ &= \begin{bmatrix} \alpha_1 & c_1^* P_1 \\ P_1^* c_1 & P_1^* M_1 P_1 \end{bmatrix}.\end{aligned}$$

Any orthogonal matrix P_1 such that $P_1^* c_1 = e_1 \beta_1$ will cause \hat{A}_1 to be tridiagonal in its first column. Since P_1 is orthogonal

$$|\beta_1| = \|e_1 \beta_1\| = \|P_1^* c_1\| = \|c_1\|.$$

Two practical choices for P_1 are described below. When P_1 has been chosen then $P_1^* M_1 P_1$, which we call A_2 , is calculated explicitly and, since this step is the heart of the algorithm, its cost is critical for the efficiency of the method. At first glance it appears that two matrix multiplications are involved and this normally requires $2(n - 1)^3$ operations. However, we can do much better than that.

At the second step the same technique is applied to

$$P_1^* M_1 P_1 \equiv A_2 = \begin{bmatrix} \alpha_2 & c_2^* \\ c_2 & M_2 \end{bmatrix}, \quad A_2 \text{ is } (n - 1) \times (n - 1).$$

An orthogonal matrix P_2 is chosen so that $P_2^* c_2 = e_1 \beta_2$, and then $A_3 = P_2^* M_2 P_2$ must be computed. The crucial observation is that *the similarity transformation at the second step does not destroy the zero elements introduced at the first step*. This can be seen by forming the product of the three matrices shown below:

$$\begin{bmatrix} 1 & 0 & 0^* \\ 0 & 1 & 0^* \\ 0 & 0 & P_2^* \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & 0^* \\ \beta_1 & \alpha_2 & c_2^* \\ 0 & c_2 & M_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0^* \\ 0 & 1 & 0^* \\ 0 & 0 & P_2 \end{bmatrix}.$$

The process continues, the submatrix which is not yet in tridiagonal form shrinks, and finally after $(n - 2)$ steps the tridiagonal T is obtained.

There is one more point to notice:

$$\begin{aligned}T &= \begin{bmatrix} I_{n-2} & 0^* \\ 0 & P_{n-2}^* \end{bmatrix} \cdots \begin{bmatrix} 1 & 0^* \\ 0 & P_1^* \end{bmatrix} A \begin{bmatrix} 1 & 0^* \\ 0 & P_1 \end{bmatrix} \\ &\quad \times \begin{bmatrix} I_2 & 0^* \\ 0 & P_2 \end{bmatrix} \cdots \begin{bmatrix} I_{n-2} & 0^* \\ 0 & P_{n-2} \end{bmatrix} \\ &= Q^* A Q, \text{ defining } Q\end{aligned}$$

and

$$Qe_1 = \begin{bmatrix} 1 & 0^* \\ 0 & P_1 \end{bmatrix} \cdots \begin{bmatrix} I_{n-2} & 0^* \\ 0 & P_{n-2} \end{bmatrix} e_1 = e_1.$$

By the reduction uniqueness Theorem 7.2.1 the *product* of the $n - 2$ orthogonal matrices involving the P_i is uniquely determined despite the varied possibilities for each P_i , $i = 1, \dots, n - 3$.

7.4.1. Exploiting Symmetry

When A is full and can be stored in the high-speed memory the preferred choice for P to satisfy $Pc = e_1\beta$ is the reflector matrix (see section 6.3):

$$P = H(W) = I - \gamma WW^*, \quad \gamma = 2/W^*W,$$

where

$$w = c + \beta e_1, \quad \beta = \|c\| \text{sign}(e_1^* c).$$

Two valuable assets arise from this choice. In the first place, P is not needed as an explicit two-dimensional array; it is determined by w and w differs from c only in its first element. So no extra n -by- n arrays are required. Second, the similarity transformation $H(w)MH(w)$ can be carried out with great efficiency, as explained below. It would be ridiculous to fill out a j -by- j array H and then execute two matrix multiplications at a cost of $2j^3$ scalar multiplications. Instead we use the fact that H is elementary and write

$$\begin{aligned} HMH &= (I - \gamma WW^*)M(I - \gamma WW^*) \\ &= M - \gamma w(w^*M) - \gamma(Mw)w^* + \gamma^2 w(w^*Mw)w^* \\ &= M - wp^* - pw^* + (\gamma w^* p)ww^*, \end{aligned} \tag{7.9}$$

where

$$p = \gamma Mw.$$

This use of p is good and reduces the operation count to $3j^2$ multiplications. Yet there is an extra trick (due to Wilkinson) which is a nice example of the *art* of writing programs to implement numerical methods. The quantity $\gamma w^* p = \gamma^2 w^* Mw$ is real (even when M is complex Hermitian), and so the fourth term in (7.9) may be shared between the second and third terms as follows. Compute

$$q = p - w(\gamma w^* p)/2,$$

and then

$$HMH = M - wq^* - qw^*.$$

The order of computation and operation count are shown below:

Quantity	w	γ	p	w^*p	q	$M - wq^* - qw^*$	Total
Multiplications	0	j	$j(j + 1)$	j	j	j^2	$2j(j + 2)$

At each step of the reduction to tridiagonal form the order j shrinks by one. The total number of multiplications is $\sum_{j=2}^{n-1} 2j(j + 2) = \frac{2}{3}n^3 + n^2 + O(n)$ while the most naive implementation would require $n^2(n + 1)^2/2$ ops. The number of square roots is $(n - 2)$.

In 1997 I realized that the direct rotation (see section 6.3.2) would have been as efficient as a reflection for implementing the reduction of A to T.

7.4.2. Plane Rotations

Another way to reduce c to $\pm e_1 \|c\|$ is by a sequence of $(j - 1)$ plane rotations. These elementary orthogonal matrices are described in section 6.4. Each rotation affects two rows (and two columns) of the matrix and creates one new zero in the vector which started as c. (This method and the whole idea of explicitly reducing A to T was introduced by Givens in 1954.)

The elements can be annihilated in the natural order $2, 3, \dots, j$ or the reverse. Moreover there are two popular ways of annihilating element k : either rotate in the $(2, k)$ plane or in the $(k - 1, k)$ plane.

Each of the $(j - 1)$ rotations requires $4j$ multiplications and 1 square root. This yields a total multiplication count

$$\sum 4j(j - 1) = \frac{4}{3}n^3 + O(n)$$

and $n(n - 1)/2$ square roots.

When plane rotations are used in the scaled, or fast, form described in section 6.8, then the number of multiplications is halved and this technique becomes competitive with the reflections of section 7.4.1. Moreover in the treatment of sparse matrices, in particular those of banded form, it is easy to skip unnecessary rotations. On the other hand on computers that favor the formation of the products of dense matrices block reflections are preferred. Thus w becomes a tall thin matrix and $\gamma = 2(w^*w)^{-1}$.

Exercises on Section 7.4

$$M = \begin{bmatrix} 0 & & & \text{sym} \\ 1 & -1 & & \\ 2 & 4 & 2 & \\ 4 & -2 & 0 & 4 \end{bmatrix}.$$

- 7.4.1. Carry out the first step of the reduction of M to tridiagonal form by using the direct rotation K .
 - 7.4.2. Reduce the first column of M above to tridiagonal form using plane rotations in the $(2,3)$ and $(2,4)$ planes.
 - 7.4.3. Use the fast Givens transformations described in section 6.8 to reduce M to factored tridiagonal form $\Delta \tilde{T} \Delta$ where Δ is diagonal.

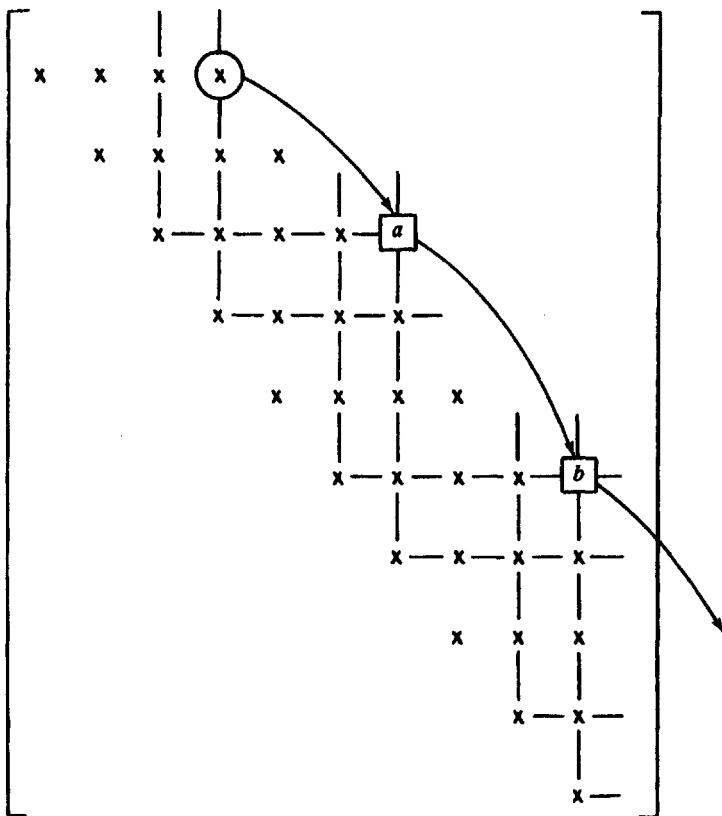
7.5. Reduction of a Banded Matrix

$$A = \begin{bmatrix} & & \\ & \textcircled{1} & \\ & & \textcircled{1} \\ \textcircled{1} & & & \end{bmatrix}$$

A matrix of order 500 and bandwidth 51 requires 13,000 words of storage and can be processed in fast memory *provided that* the bandwidth is not increased during the algorithm. By annihilating pairs of elements a_{ij} and a_{ji} in an ingenious order, banded matrices can be reduced to tridiagonal form without enlarging the bandwidth in the process.

Recall that by use of a suitable plane rotation, any off-diagonal element of A in column j can be annihilated by forming R^*AR with $R(i, j, \theta)$. When the doomed element is a_{ij} this is often called a *Jacobi rotation*; otherwise it is called a *Givens rotation*. The choice of the angle θ is different in the two cases. In the algorithm of this section $i = j - 1$ and the doomed element is a_{lj} with $l < j - 1$.

The idea of the pattern of elements to be annihilated is revealed in Figure 7.1 for the case $N = 10, m = 3$.

FIG. 7.1. *Reduction of bandwidth.*

7.5.1. The Method

The first row is reduced to tridiagonal form, then the second, and so on. The elements in each row are annihilated one by one, from the outside in. But it takes several rotations to eliminate a single element *and* to preserve bandwidth. The treatment of the first row is typical.

1. Rotate in plane $(m, m + 1)$ to annihilate element $(1, m + 1)$. This creates a nonzero element (a) at $(m, 2m + 1)$ *outside the band*.
2. Rotate in plane $(2m, 2m + 1)$ to annihilate element $(m, 2m + 1)$. This creates a nonzero element (b) at $(2m, 3m + 1)$ *outside the band*.
3. Rotate in plane $(3m, 3m + 1)$ to annihilate element $(2m, 3m + 1)$. In this case ($n = 10$); the bulge has been chased off the bottom of the matrix.

In general more of these rotations (to a total of $[(n - 1)/m]$) would be required to restore the original bandwidth.

The next element, within the band, to be annihilated is $(1, m)$, and the resulting bulge has to be chased off the end of the matrix and so on. After the $(1, 3)$ element is treated the second row can be processed and so on.

7.5.2. Operation Count

The total number of rotations, $N_{\text{rotations}}$, satisfies

$$N_{\text{rotations}} \leq n^2(m - 2)/(2m).$$

Each rotation requires one square root and, usually, $8m + 13$ ops. Thus

$$N_{\text{ops}} \leq n^2(m - 2)(4 + 13/2m).$$

Operation counts show that the standard reduction is faster unless $m < n/6$. However, it is storage considerations which usually dictate the use of banded reduction.

7.5.3. Storage

It is customary to use a rectangular array, n by $(m + 1)$, whose columns hold the successive diagonals of the banded matrix; the main diagonal is in the first column.

7.5.4. Reference

An explicit ALGOL program called *bandrd* is given in [Contribution II/8 by H. R. Schwarz, in Wilkinson and Reinsch, 1971]. This method is in EISPACK.

The fast Givens rotations can be used to halve the work.

Exercises on Section 7.5

Let

$$\mathbf{M} = \begin{bmatrix} 6 & & & & \\ -4 & 6 & & & \text{sym} \\ 1 & -4 & 6 & & \\ 0 & 1 & -4 & 6 & \\ 0 & 0 & 1 & -4 & 6 \end{bmatrix}.$$

- 7.5.1. Using the algorithms of the previous section, reduce \mathbf{M} to tridiagonal form by means of four plane rotations.

- 7.5.2. Reduce M to tridiagonal form by means of fast Givens rotations to obtain a factored result $\Delta \hat{T} \Delta$ where Δ is diagonal. A hand-held calculator will be useful. You should compute Δ^2 and \hat{T} . See section 6.8.

7.6. Irrelevant Instability

We are not going to present detailed analyses of the influence of roundoff errors on the preceding algorithms because the general analysis given in section 6.5 covers both of them. In each case the computed tridiagonal matrix will be exactly similar to a perturbation $A + W$ of the original matrix A , and $\|W\|/\|A\|$ is a modest multiple of the unit roundoff. Even if the pessimistic bounds on the ratio were actually attained, that would still be quite satisfactory for most calculations.

Instead we present one example (Example 7.6.1) of an interesting phenomenon which appears, at first glance, to contradict the preceding paragraph. Examples of this type have generated a lot of unnecessary worry. In Example 7.6.1 the same reduction program was run on a nice 24-by-24 banded matrix on two different machines and produced quite different results; check α_6 and α_7 for the example in Table 7.1. Is there a bug in the program, has one (or both) of the computers malfunctioned, or is the algorithm unstable? Surely reproducibility of results is essential to scientific work?

Indeed the two T 's are different, but they are each similar to A to working precision and that is what counts in eigenvalue calculations. The *particular sequence* of similar matrices derived from this A by the algorithm is extremely sensitive to the tiny perturbations produced by roundoff. We may say that the explicit reduction of A to T by orthogonal transformations is forward unstable but backward stable. This phenomenon makes it difficult to judge algorithms. The temptation to compare computed output with the correct (exact arithmetic) results is always lurking in our subconscious minds.

The eigenvalues and eigenvectors of A were computed to as much accuracy as was warranted on each computer.

Care was taken so that each machine worked on the same matrix.

Example 7.6.1. Sensitivity of the tridiagonal form to roundoff errors in the reduction

$$A = \begin{bmatrix} M & C_2^* & \circ & \circ \\ C_2 & M & C_3^* & \circ \\ \circ & C_3 & M & C_4^* \\ \circ & \circ & C_4 & M \end{bmatrix}, \quad M = \begin{bmatrix} H & B^* \\ B & H \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -1 & -1/2 \\ 1 & 0 & -1/3 \\ 1/2 & 1/3 & 0 \end{bmatrix},$$

$$C_i = (\epsilon_6 e_i^* - \epsilon_1 e_6^*)/10^{2i}.$$

TABLE 7.1
Irrelevant instability.

$$t_{i,i} = \alpha_i; \quad t_{i,i+1} = t_{i+1,i} = \beta_i.$$

i	β_i		α_i	
	CDC 6400	PDP 11	CDC 6400	PDP 11
1	0.0000	-0.0080	0.4428	0.4428
2	0.0009	-0.0011	-0.8777	-0.8777
3	0.0141	0.1441	0.4429	0.4444
4	0.6169	1.5997	11.4038	11.2215
5	-0.0115	-0.0002	-0.8466	-0.6659
6	-0.0642	-4.3642	0.4431	9.2954
7	0.3520	-0.2957	11.4244	2.5251
8	-0.0007	-0.0002	-0.8676	-0.8205
9	0.5058	-1.3585	11.4115	11.2643
10	0.0114	0.0531	0.4661	0.6113
11	0.0004	-0.0006	-0.8776	-0.8756
12	0.0000	-1.6807	11.4348	11.1716
13	-0.0088	-0.0535	0.4429	0.7040
14	-0.0319	0.0317	-0.8775	-0.8755
15	-0.3579	-0.0118	11.4231	11.4348
16	0.0435	-0.0731	0.5431	-0.8737
17	-0.2052	-1.1474	-0.8728	0.5599
18	0.0001	-0.0000	11.4314	11.3138
19	-0.2107	-0.9605	0.4088	-0.7741
20	1.3226	2.2464	-0.6994	10.8806
21	-0.0000	-0.0000	11.2907	0.8936
22	-0.8784	-0.8784	-0.0453	-0.0453
23	-5.8618	-5.8618	5.0453	5.0453
24	-	-	6.0000	6.0000

(Mantissa: CDC-48 bits; PDP-24 bits.)

Values are given to four decimal places.

$$\max_i |\lambda_i(\mathbf{A})_{\text{CDC}} - \lambda_i(\mathbf{A})_{\text{PDP}}| \leq 9. \times 10^{-6} \leq \frac{1}{5} n \epsilon_{\text{PDP}} \|\mathbf{A}\|$$

7.7. Eigenvalues Are Simple

There is little that is special about the eigenvalues of T with the exception of Lemma 7.7.1.

Lemma 7.7.1. *The eigenvalues of an unreduced T are distinct.*

Proof. For all ξ the minor of the $(1, n)$ element of $T - \xi$ is $\beta_1\beta_2 \cdots \beta_{n-1} \neq 0$. Consequently, $\text{rank } [T - \xi] \geq n - 1$ and the dimension of the null space of $T - \xi$ is either 0 (when ξ is not an eigenvalue) or 1 (when ξ is an eigenvalue). This means that there is only one linearly independent eigenvector to each eigenvalue. Since T has a full set of orthogonal eigenvectors, the multiplicity of each eigenvalue must therefore be 1. \square

This result is useful in theoretical work, but it can mislead us into the false assumption that if two eigenvalues are very close then some β_i must be small. A well-known counterexample is W_{21}^+ (discussed in [Wilkinson, 1965]):

$$\alpha's: 10, 9, 8, \dots, 1, 0, 1, \dots, 10; \quad \beta_i = 1 \text{ all } i.$$

$$\lambda_{21}[W^+] = 10.74619\ 41829\ 0339\dots$$

$$\lambda_{20}[W^+] = 10.74619\ 41829\ 0332\dots$$

7.8. Orthogonal Polynomials

A big advance was made in linear algebra with the realization that the length of a vector is *not* an a priori, intrinsic property of that vector. In fact there are infinitely many legitimate inner product functions and each gives rise to its own version of length and angle. The numerical measure of the angle between two vectors is a matter of convention, not reality. It is the same with functions; there are many inner products.

Important in applied mathematics are real functions ϕ, ψ, \dots of one real variable and, in particular, the set P_n of polynomials of degree not exceeding n . We shall not consider the general integral inner products

$$(\phi, \psi) \equiv \int_a^b \omega(x)\phi(x)\psi(x)dx$$

but go straight to the discrete case

$$(\phi, \psi) \equiv \sum_{i=1}^n \omega_i \phi(\xi_i) \psi(\xi_i). \quad (7.10)$$

To each set of n distinct real numbers $\{\xi_1, \dots, \xi_n\}$ and positive weights $\{\omega_1, \dots, \omega_n : \omega_i > 0\}$ there corresponds one, and only one, inner product function on \mathcal{P}_{n-1} as defined by (7.10). The corresponding norm and angle are given in Chapter 1.

Polynomials are rather special functions, and for each inner product there is a *unique* family of monic orthogonal polynomials $\{\phi_0, \phi_1, \dots, \phi_{n-1}\}$; that is, ϕ_j has degree j , leading coefficient 1, and $(\phi_j, \phi_k) = 0$ for $j \neq k$. This family is the distinguished basis of the inner product (or Hilbert) space \mathcal{P}_{n-1} enriched with the given inner product.

Tridiagonal matrices come into the picture because there is a remarkable three-term recurrence relation among the ϕ 's; for $j = 1, 2, \dots, n$ and $\beta_0 = 0$,

$$\phi_j(\xi) = (\xi - \alpha_j)\phi_{j-1}(\xi) - \beta_{j-1}^2 \phi_{j-2}(\xi). \quad (7.11)$$

Once such a relationship has been guessed, it is straightforward to verify what the α 's and β 's must be (see Exercise 7.8.1). These numbers may be put into an unreduced symmetric tridiagonal matrix T in the obvious way, as in (7.1), and then for $j = 1, \dots, n$, Exercise 7.8.2 reveals that

$$\phi_j(\xi) = \chi_j(\xi) \equiv \det[\xi - T_{1:j}]. \quad (7.12)$$

Thus the ξ 's and the ω 's determine a unique T . The question we pose now is how to determine the ξ 's and ω 's from a given T . In other words, which inner products make the χ_j , $j = 0, 1, \dots, n$, mutually orthogonal?

Theorem 7.8.1. *Let $T = S\Theta S^*$ be the spectral decomposition of an unreduced T . Then the associated inner product of the form (7.10) is given by*

$$\xi_i = \theta_i, \quad \omega_i = \gamma s_{1i}^2, \quad i = 1, \dots, n, \quad (7.13)$$

for any positive γ ; $\sum_1^n \omega_i = \gamma$.

Proof. The essential observation is that S^* , not S , “reduces” Θ to tridiagonal form T and so, by Corollary 7.3.1,

$$S^* e_{j+1}(\beta_j \cdots \beta_1) = \chi_j(\Theta) S^* e_1.$$

The orthogonality of columns $(j+1)$ and $(k+1)$ of S^* gives

$$\begin{aligned} 0 &= \sum_{i=1}^n (\chi_j(\theta_i) s_{1i})(\chi_k(\theta_i) s_{1i}) \\ &= (\chi_j, \chi_k) \text{ if } \omega_i = s_{1i}^2 \text{ and } \xi_i = \theta_i \text{ in (7.10).} \end{aligned}$$

What other values for ω_i will work? The necessary condition $(\chi_0, \chi_k) = 0$ requires that

$$\sum_{i=1}^n \omega_i \chi_k(\theta_i) = 0, \quad k = 1, 2, \dots, n-1, \quad (7.14)$$

whereas, for $k = 0$,

$$\sum_{i=1}^n \omega_i = \gamma = \text{arbitrary positive number.} \quad (7.15)$$

However the Vandermonde-like matrix G whose (i, j) element is $\chi_{j-1}(\theta_i)$, $j = 1, \dots, n$, is nonsingular (Exercise 7.8.4), and so the system of linear equations specified by (7.14) and (7.15) has a unique solution $\{\omega_1, \dots, \omega_n\}$ for each positive γ . \square

It is customary to take $\gamma = 1$. In [Golub and Welsch, 1969] it is shown how to adapt the techniques of Chapter 8 to compute the s_{1i} , $i = 1, \dots, n$ without finding all of S . We cover that topic in section 7.9 and find a use for the values s_{ni} , $i = 1, \dots, n$ in Chapter 13.

Exercises on Section 7.8

- 7.8.1. By taking suitable inner products show that if η denotes the identity function $\eta(\xi) \equiv \xi$ then

$$\alpha_{j+1} = (\eta\phi_j, \phi_j)/(\phi_j, \phi_j),$$

$$\beta_j^2 = (\eta\phi_{j-1}, \phi_j)/(\phi_{j-1}, \phi_{j-1}).$$

Why is $(\eta\phi_{j-1}, \phi_j) > 0$?

- 7.8.2. Define $\chi_0(\xi) \equiv 1$ and expand $\det[\xi - T_{1:j}]$ by its last row to discover that the χ_i also obeys (7.11). Hence show that $\phi_i = \chi_i$ for all i . Note that $\phi_n \notin \mathcal{P}_{n-1}$ but, being orthogonal to each ϕ_i , $i < n$, its norm must vanish. Thus ϕ_n vanishes at the abscissae ξ_i of (7.10).
- 7.8.3. Let F be the matrix with $f_{ij} = \theta_i^{j-1}$. Show that F is nonsingular by using the fact the $\det F$ is a polynomial in the θ_i , $i = 1, \dots, n$.
- 7.8.4. Let $G = [\chi_{j-1}(\theta_i)]$. Show that G is nonsingular.
- 7.8.5. What is the inner product corresponding to the second difference matrix $[\cdots -1 \ 2 \ -1 \cdots]$, and what is the name of the associated set of polynomials?

7.9. Eigenvectors of T

There is a remarkable formula for the square of any element of any normalized eigenvector of T , and there are several beautiful relations among the elements of each eigenvector.

In order to derive these results we recall the notion of the (classical) *adjugate* of a matrix, namely, the transpose of the matrix of cofactors. For example, if $|B| = \det B$, then

$$\begin{aligned} \text{adj} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 2 & -1 & 1 \end{bmatrix} &= \begin{bmatrix} \begin{vmatrix} 1 & 1 \\ -1 & 1 \end{vmatrix}, & -\begin{vmatrix} 0 & 1 \\ -1 & 1 \end{vmatrix}, & \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} \\ -\begin{vmatrix} 1 & 1 \\ 2 & 1 \end{vmatrix}, & \begin{vmatrix} 1 & 1 \\ 2 & 1 \end{vmatrix}, & -\begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} \\ \begin{vmatrix} 1 & 1 \\ 2 & -1 \end{vmatrix}, & -\begin{vmatrix} 1 & 0 \\ 2 & -1 \end{vmatrix}, & \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix} \end{bmatrix} \\ &= \begin{bmatrix} 2 & -1 & -1 \\ 1 & -1 & 0 \\ -3 & 1 & 1 \end{bmatrix}. \end{aligned}$$

The importance of the adjugate stems from the famous Cauchy–Binet formula

$$B \cdot \text{adj}[B] = \det[B] \cdot I.$$

We begin with a fact about the symmetric matrices that was proved in [Thompson and McEnteggert, 1968]. It employs the characteristic polynomial χ of A and relates certain adjugates to the spectral projectors $H_i \equiv z_i z_i^*$ described in Chapter 1.

Theorem 7.9.1. Let $A = Z\Lambda Z^*$ be the spectral decomposition of A ; $Z = (z_1, \dots, z_n)$ is orthogonal, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, with each λ_i simple. Then for $i = 1, \dots, n$,

$$\text{adj}[\lambda_i - A] = \chi'(\lambda_i) z_i z_i^*, \quad (7.16)$$

where χ' is the derivative of χ .

Proof. For $\mu \neq \lambda_i$, $\mu - A$ is invertible and so

$$\begin{aligned} \text{adj}[\mu - A] &= \det\mu - A^{-1} \\ &= \chi(\mu)Z(\mu - \Lambda)^{-1}Z^* \\ &= Z \Delta(\mu)Z^*, \quad \Delta = \text{diag}(\delta_1, \dots, \delta_j), \end{aligned} \quad (7.17)$$

where

$$\delta_k = \delta_k(\mu) = \chi(\mu)/(\mu - \lambda_k) = \prod_{j \neq k} (\mu - \lambda_j).$$

The elements of $\text{adj}[B]$ are sums of products of elements of B and thus continuous functions of them. On letting $\mu \rightarrow \lambda_i$ both sides of (7.17) have limits. In particular, since the λ_i are simple,

$$\delta_k(\lambda_i) = \begin{cases} 0, & k \neq i, \\ \chi'(\lambda_i), & k = i, \end{cases}$$

and the result follows. \square

Before applying this result to T we recall the definition of $T_{\mu:\nu}$ in (7.1) and define

$$\chi_{\mu:\nu}(\tau) = \begin{cases} \det[\tau - T_{\mu:\nu}], & \mu \leq \nu, \\ 1, & \mu > \nu. \end{cases}$$

The following corollary of Theorem 7.9.1 was given in [Paige, 1971].

Theorem 7.9.2. Let $T = S\Theta S^*$ be the spectral decomposition of tridiagonal T ; $S = (s_1, \dots, s_n)$ is orthogonal, $\Theta = \text{diag}(\theta_1, \dots, \theta_n)$. Then, for $\mu \leq \nu$ and all j , the elements $s_{\mu j} (\equiv e_{\mu}^* s_j)$ of the normalized eigenvectors of T obey

$$\chi'_{1:n}(\theta_j) s_{\mu j} s_{\nu j} = \chi_{1:\mu-1}(\theta_j) \beta_\mu \cdots \beta_{\nu-1} \chi_{\nu+1:n}(\theta_j). \quad (7.18)$$

In particular, when θ_j is a simple eigenvalue (T unreduced),

$$s_{\mu j}^2 = \chi_{1:\mu-1}(\theta_j) \chi_{\mu+1:n}(\theta_j) / \chi'_{1:n}(\theta_j). \quad (7.19)$$

The proof is left as Exercise 7.9.3. Recall that each χ is monic. Delete row and column μ from T , and the new characteristic polynomial is the $\chi_{1:\mu-1} \chi_{\mu+1:n}$ in (7.19).

On giving special values to μ and ν several valuable relations among the elements of S become visible. Recall that $\chi(\theta) \equiv \chi_{1:n}(\theta)$.

Corollary 7.9.1. For all $j \leq n$,

$$s_{1j} s_{nj} \chi'(\theta_j) = \beta_1 \beta_2 \cdots \beta_{n-1}, \quad (7.20)$$

$$s_{1j}^2 \chi'(\theta_j) = \chi_{2:n}(\theta_j), \quad (7.21)$$

$$s_{nj}^2 \chi'(\theta_j) = \chi_{1:n-1}(\theta_j). \quad (7.22)$$

The next results are useful in establishing convergence for various iterative methods for computing eigenvectors.

Formula (7.22) may be used to give upper and lower bounds on the entries in the last row of the eigenvector matrix of an unreduced tridiagonal matrix.

Corollary 7.9.2. Consider the notation of Theorem 7.9.2 and let $\tau_1 < \tau_2 < \dots < \tau_{n-1}$ be the eigenvalues of $T_{1:n-1}$; then

$$\left\{ \frac{\frac{\tau_1 - \theta_1}{\theta_n - \theta_1}}{\frac{\theta_j - \tau_{j-1}}{\theta_j - \theta_1} \cdot \frac{\frac{\tau_j - \theta_j}{\theta_n - \theta_j}}{\frac{\theta_n - \tau_{n-1}}{\theta_n - \theta_1}}} \right\} < s_{nj}^2 < \begin{cases} \frac{\frac{\tau_1 - \theta_1}{\theta_2 - \theta_1}}{\frac{\theta_j - \tau_{j-1}}{\theta_j - \theta_{j-1}} \cdot \frac{\frac{\tau_j - \theta_j}{\theta_{j+1} - \theta_j}}{\frac{\theta_n - \tau_{n-1}}{\theta_n - \theta_{n-1}}}}, & j = 1, \\ \frac{\frac{\tau_1 - \theta_1}{\theta_2 - \theta_1}}{\frac{\theta_j - \tau_{j-1}}{\theta_j - \theta_{j-1}} \cdot \frac{\frac{\tau_j - \theta_j}{\theta_{j+1} - \theta_j}}{\frac{\theta_n - \tau_{n-1}}{\theta_n - \theta_{n-1}}}}, & j \neq 1, n, \\ \frac{\frac{\tau_1 - \theta_1}{\theta_2 - \theta_1}}{\frac{\theta_n - \tau_{n-1}}{\theta_n - \theta_{n-1}}}, & j = n. \end{cases}$$

Proof. See [Hill and Parlett, 1992, Theorem 3]. \square

Theorem 7.9.3. If T is unreduced then its eigenvector matrix has no zero elements in its first and last rows nor in the columns corresponding to the extreme eigenvalues.

Proof. Since T is unreduced its eigenvalues are distinct by Lemma 7.7.1. Thus $\chi'(\theta_j) \neq 0$. By Corollary 7.9.1, $s_{1j}s_{nj} = \Pi \beta_i / \chi'(\theta_j) \neq 0$. Now consider the two columns. By Cauchy's interlace theorem (section 10.1) the zeros of $\chi_{1,n-1}$ and $\chi_{2,n}$ and all $\chi_{\mu,\nu}$ lie strictly between θ_1 and θ_n . By Paige's formula (7.19) the corresponding eigenvectors have no zero elements. \square

A picture of S for a random 25-by-25 T is given in Example 13.2.1.

Example 7.9.1. An instructive illustration of (7.21) is the case when $n = 100$, $\alpha_i = 0$, $\beta_i = 1$ for all i . The eigenvectors are the same as for the famous second difference matrix ($\alpha_i = -2$, $\beta_i = 1$ for all i), and $\chi_{1,n}(\tau)$ is related to the Chebyshev polynomial T_{n+1} described in Appendix B. Let $\omega = \pi/(n+1)$. In this example it is convenient to order the θ_j by $\theta_1 > \theta_2 > \dots$

Eigenvalues: $\theta_j = 2 \cos j\omega$, $j = 1, \dots, n$.

Eigenvectors: $s_{ij} = \sqrt{\kappa} \sin i j \omega$, $\kappa = 2/(n+1)$.

$(s_{1j} = (-1)^{j+1}s_{nj}; \text{ hence } \chi_{2:n}(\theta_j) = (-1)^{j+1}\beta_1 \cdots \beta_{n-1} = (-1)^{j+1})$
 In fact $\chi(\theta) = \kappa T'_{101}(\theta/2)$.

j	1	2	49	50
θ_j	0.1999×10^1	0.1996×10^1	0.9328×10^{-1}	0.3110×10^{-1}
$\chi'(\theta_j)$	0.5221×10^5	-0.1307×10^5	0.5061×10^2	-0.5051×10^2
$s_{1j}^2 = s_{nj}^2$	0.1915×10^{-4}	0.7654×10^{-4}	0.1976×10^{-1}	0.1980×10^{-1}
$\chi_{2:100}(\theta_j)$	1.0	-1.0	1.0	-1.0

Note the large values for $\chi'(\theta_i)$, $i = 1, 2$, although θ_1 and θ_2 are close.

Exercises on Section 7.9

7.9.1. Compute

$$\text{adj} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

and $\text{adj}[T_{1:3}]$ for a general T .

7.9.2. Show that $\sum_{i=1}^n \text{adj}[\lambda_i - A] = \chi'(A)$.

7.9.3. Prove Paige's Theorem 7.9.2 by evaluating the (μ, ν) element of $\text{adj}[\theta_j - T]$.

7.9.4. Give the proper values to μ and ν to establish Corollary 7.9.1.

7.9.5. Compute the values of s_{1j} and s_{nj} , for all j , for

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 10 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 10 \end{bmatrix}, \quad \begin{bmatrix} 10 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -10 \end{bmatrix}.$$

7.9.6. Compute $\chi'(\lambda_1)z_1 z_1^*$ and $\text{adj}[\lambda_1 - A]$ for $A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$.

7.10. Sturm Sequences

For a given T the sequence of characteristic polynomials $\{\chi_j(\xi), j = 0, 1, \dots, n\}$ defined in (7.12) is a *Sturm sequence*; that is, the zeros of χ_{j-1} interlace those of χ_j . This result is an application of the Cauchy interlace theorem (section 10.1). An interesting property of Sturm sequences is that the number of sign agreements between consecutive terms of the numerical sequence $\{\chi_k(\xi), k = 0, 1, \dots, n\}$, call it $\alpha(\xi)$, equals the number of zeros of χ_n which are less than ξ . If $\xi < \eta$ then

$$\alpha(\eta) - \alpha(\xi)$$

is the number of eigenvalues of T in the half open interval $[\xi, \eta)$. Figure 7.2 illustrates a Sturm sequence for a simple 3-by-3 matrix T .

The interlacing of the zeros is strict if T is unreduced, but the matrix W_{21}^- introduced in section 3.6 shows how weak the strict inequality can be. For W_{21}^- the largest zero of χ_{20} agrees with the largest zero of χ_{21} through the first 14 decimals.

By using the three-term recurrence, shown in section 7.7, the sequence $\{\chi_k(\xi)\}$ can be evaluated at a cost of $2n$ multiplications. By the method of bisection (section 3.5) eigenvalues can be approximated as accurately, or as crudely, as desired subject to the limitations of roundoff.

This ingenious technique was presented by Givens in 1954 and has been used extensively ever since. However, the same information can be obtained by use of triangular factorization and Sylvester's inertia theorem, a method which is both more stable and has wider applicability. It is described in Chapter 3.

The advantage of the three-term recurrence is the absence of division which is relatively slow on many computers. Because the three-term recurrence is prone to overflow and underflow the algorithm has to fuss with rescaling and testing.

The observant reader may have noticed that if θ is an eigenvalue of T then

$$[1, \chi_1(\theta)/\beta_1, \dots, \chi_{n-1}(\theta)/(\beta_1\beta_2 \cdots \beta_{n-1})]^* \quad (7.23)$$

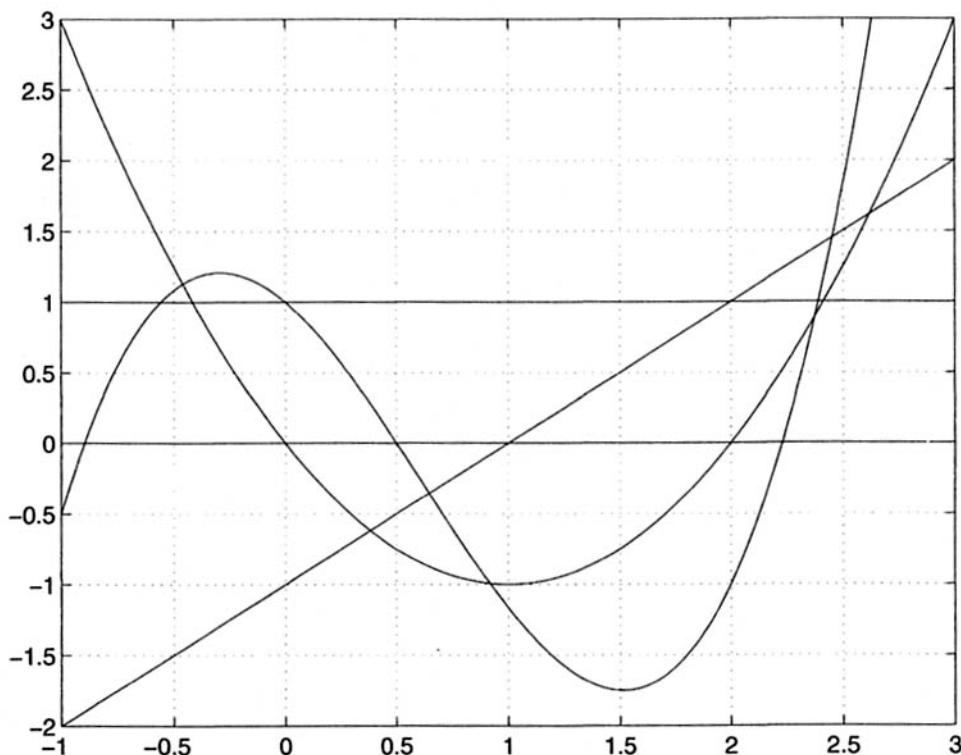
is an eigenvector of T (Exercise 7.10.2). However, in practice, when θ is merely a very accurate approximation the vector given above can be almost orthogonal to the true eigenvector. The technique is violently unstable. A detailed and illuminating discussion is given in [Wilkinson, 1965, p. 316]. The topic is developed in Exercises 7.10.3–7.10.5. However, the algorithm warrants further study because its storage demands are so modest.

Exercises on Section 7.10

- 7.10.1. Show that if $\chi_j(\xi)$ had been defined as $\det[T_{1:j} - \xi]$ then $\alpha(\xi)$ would give the number of eigenvalues of T greater than ξ . However, χ_j would not be monic for odd j .
- 7.10.2. By examining the three-term recurrence for $\{\chi_j(\theta)\}$ show that (7.23) gives an eigenvector of T belonging to the eigenvalue θ .
- 7.10.3. Show that when θ is not an eigenvalue then the vector v given in (7.23) satisfies

$$(T - \theta)v = e_n \chi_n(\theta) / (-\beta_1 \cdots \beta_{n-1}).$$

- 7.10.4. Replace θ by $\theta + \delta\theta$ and expand each $\chi_j(\theta + \delta\theta)$ about θ to obtain an expression for the effect of perturbations in θ on (7.23).



$$A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & -\frac{1}{6} \end{bmatrix}$$

ξ	-1	0	1.5	3
χ_0	1	1	1	1
χ_1	-2	-1	0.5	2
χ_2	3	0	-0.75	3
χ_3	-0.5	1	2	3
α	0	1	2	3

FIG. 7.2. Sturm sequence.

7.10.5. Consider the tridiagonal matrix $(\cdots -1 \ 2 \ -1 \cdots)$ whose eigenvalues are given in Example 7.9.1, and hence find the change in the approximate vector v for various eigenvalues (inner and outer) when $\delta\theta = 10^{-10}\theta$

and $n = 100$. See equation (7.23).

7.11. When to Neglect an Off-Diagonal Element

If $\beta_i = 0$ then T is completely reduced to $\text{diag}(T_1, T_2)$, and the two submatrices T_1 and T_2 can be processed independently with some small gain in efficiency. Of more importance is the fact that some processes (the QL algorithm, for example) *require* that any such reductions be recognized; the process will be marred, if not ruined, by an unexpected zero value for a β .

In practice, therefore, we must recognize when T is reduced “to within working accuracy.” The treatment of this problem is the least satisfactory aspect of the computation of eigenvalues of symmetric tridiagonal matrices. It is useful to distinguish some distinct but related questions.

7.11.1. The Mathematical Problem

Find an upper bound, in terms of T ’s elements, on the change $\delta\theta$ induced in any eigenvalue θ when β_i is replaced by zero.

Simple answer: $|\delta\theta_j| \leq |\beta_i|$ for all j (Exercise 7.11.1).

Wrong answer: $|\delta\theta_j| \leq \beta_i^2 / \min |\alpha_\mu - \alpha_\nu|$ over all $\alpha_\mu \neq \alpha_\nu$.

Example

$$T = \begin{bmatrix} 1 & \sqrt{2} & 0 \\ \sqrt{2} & 2 & \beta \\ 0 & \beta & 0 \end{bmatrix}.$$

Eigenvalues

$$\begin{cases} 3 + O(\beta^2), \pm\beta/\sqrt{3} + O(\beta^2) & \text{when } \beta \text{ is tiny,} \\ 3, \pm 0 & \text{when } \beta = 0. \end{cases}$$

Complicated answer: Take $\beta_0 = \beta_n = 0$ and define

$$\delta_i \equiv (\alpha_{i+1} - \alpha_i)/2, \quad \rho_i^2 \equiv (1 - 1/\sqrt{2})(\beta_{i-1}^2 + \beta_{i+1}^2). \quad (7.24)$$

Then

$$\sum_{j=1}^n (\delta\theta_j)^2 \leq \frac{\beta_i^2}{\delta_i^2 + \rho_i^2} \left[2\rho_i^2 + \frac{\delta_i^2 \beta_i^2}{\delta_i^2 + \rho_i^2} \right] \equiv \omega_i^2. \quad (7.25)$$

The proof is given in [Kahan, 1966]. He applies the Wielandt–Hoffman theorem (see section 1.6) to the result of performing a rotation in the $(i, i + 1)$ plane. The bound is then minimized as a function of the angle of rotation.

There are several ways to weaken this bound in the interests of simplicity. See Exercise 7.11.3.

The simple bound $|\beta_i|$ is far too crude in most applications.

Example 7.11.1. For

$$T = \begin{bmatrix} 3 & 10^{-5} & 0 \\ 10^{-5} & 2 & 10^{-5} \\ 0 & 10^{-5} & 1 \end{bmatrix}$$

the generally wrong answer given above is correct.

7.11.2. Algorithmic Problem 1

Find an inexpensive criterion for neglecting β_i given a tolerance on the permitted disturbance to an eigenvalue θ . If ϵ is the precision of the basic arithmetic operations, then the tolerance may be absolute, $\epsilon\|T\|$, or local, $\epsilon(|\alpha_i| + |\alpha_{i+1}|)$, or relative, $\epsilon\|\theta\|$.

7.11.3. Algorithmic Problem 2

For those T which are derived from full A by similarity transformation, find an inexpensive criterion to decide when suppression of β_i is “equivalent to” the roundoff errors already made in obtaining T . In other words, ensure that T is exactly similar to a matrix which is acceptably close to A .

The desire for a simple test arises from the fact that the QL transform can change T into a more nearly diagonal \hat{T} at a cost of approximately $11n$ multiplications and n square roots. Our test will be applied to all $(n - 1)\beta$'s and will nearly always fail.

Here is the quandary: the simple test $\beta_i < \epsilon\|T\|$ will miss β_i which should be neglected and will occasionally degrade the QL algorithm; yet a test which costs as little as five multiplications will increase execution time of this part of the computation significantly.

The programmer's escape from this quandary is a two-level test:

0. . .
1. If $|\beta_i| \geq \sqrt{\epsilon}\|T\|$ then go to 4.
2. If $|\beta_i| \geq$ favorite test then go to 4.
3. Set $\beta_i = 0$.
4. . .

Several practical tests are suggested in the exercises.

7.11.4. The Last Off-Diagonal Element

In the QL algorithm the last β to be calculated is β_1 and the case for using a refined test seems strong because the goal of the algorithm is to make β_1 negligible.

In this section we have pointed to some problems but have offered no clear-cut solution, and we must report that the 1977 versions of the EISPACK programs use a simple test exclusively; $|\beta_i| < \epsilon(|\alpha_i| + |\alpha_{i+1}|)$. This test is not invariant under translation and that might shock a mathematician.

Exercises on Section 7.11

- 7.11.1. Apply the Weyl monotonicity theorem (section 10.3) to show that the change $\delta\theta$ in any eigenvalue, when β_i is replaced by 0, satisfies

$$|\delta\theta| < |\beta_i|.$$

Hint: A special rank-two matrix is subtracted from T.

- 7.11.2. Find the eigenvalues of

$$\begin{bmatrix} 1 & \sqrt{2} & 0 \\ \sqrt{2} & 2 & \beta \\ 0 & \beta & 0 \end{bmatrix}$$

as functions of β .

- 7.11.3. Using the notation of (7.24) and (7.25) show that

$$\omega_i^2 < (\beta_{i-1}^2 + \beta_i^2 + \beta_{i+1}^2)\beta_i^2/\delta_i^2.$$

- 7.11.4. Apply a Jacobi rotation to annihilate β_i . This introduces $-s\beta_{i-1}$ into element $(i+1, i-1)$ and $s\beta_{i+1}$ into $(i+2, i)$, where $s = \sin \theta$. Hence show that $|\beta_i|(|\beta_{i-1}| + |\beta_{i+1}|) < \epsilon|\delta_i|(|\alpha_i| + |\alpha_{i+1}|)$ is a reasonable criterion for neglecting β_i .

7.12. Inverse Eigenvalue Problems

The fundamental frequency of a uniform vibrating string, clamped at each end, is related to the smallest eigenvalue λ of the differential equation $-\phi'' + \sigma\phi = \lambda\phi$ with boundary conditions $\phi(a) = \phi(b) = 0$. The higher harmonics (overtones) are related to the other eigenvalues. If the string is defective and uneven then the constant σ must be replaced by a function $\sigma(\xi)$ which is related to the density of the string. The eigenvalues of the new problem will depend on σ ,

and the inverse problem asks whether it is possible to locate the defect, i.e., find $\sigma(\xi)$, when all the eigenvalues are known. The answer is that, under certain conditions and with enough extra information, it can be done. One ambitious project in the same vein hopes to determine properties of the earth's core from extensive monitoring of seismic activity at the surface.

When the differential equation is discretized it yields a tridiagonal matrix, and the discrete inverse problem is to find T when given its eigenvalues $\theta_1, \theta_2, \dots, \theta_n$ and some extra information. The previous sections have given enough understanding to solve this problem in a number of important cases.

We have written the spectral decomposition of T as

$$T = S\Theta S^* = (S^*)^* \Theta S^*, \quad (7.26)$$

where S is the matrix of normalized eigenvectors. If Θ is given then the inverse problem can be regarded as the reduction (perhaps we should say expansion) of Θ to tridiagonal form. By the uniqueness of the reduction (Theorem 7.2.1) T and S^* are determined by S^*e_1 and Θ . Sections 7.4 and 7.5 may be applied to $P^*\Theta P$ where P is any orthogonal matrix whose first column is S^*e_1 to compute T . All that remains is to find the s_{1j} , $j = 1, \dots, n$, the top entries of the normalized eigenvectors.

Case 1: symmetry about the midpoint. Suppose that T must be symmetric about the secondary diagonal (top right—bottom left) as well as the main diagonal, i.e.,

$$\alpha_j = \alpha_{n+1-j}, \quad \beta_j = \beta_{n-j} > 0, \quad j = 1, \dots, [n/2].$$

This condition corresponds to a string and associated function $\sigma(\xi)$ which are symmetric about the middle of the interval. The double symmetry of T implies (Exercise 7.12.2) that $s_{1j} = (-1)^{n-j} s_{nj}$, $j = 1, \dots, n$. Now use this in formula (7.20):

$$s_{1j} s_{nj} \chi'(\theta_j) = \beta_1 \beta_2 \cdots \beta_{n-1} \equiv \beta, \quad j = 1, \dots, n, \quad (7.27)$$

where χ is the characteristic polynomial of both T and Θ . So $\chi'(\theta_j) = \prod_{i=1}^n (\theta_j - \theta_i)$, $i \neq j$, and is determined when Θ is given. Moreover β is determined by the normalization condition

$$1 = \sum_{j=1}^n s_{1j}^2 = \beta \sum_{j=1}^n 1/|\chi'(\theta_j)|, \quad (7.28)$$

and so the starting vector $e_1^* S$ is the vector $(|\chi'(\theta_1)|^{-1/2}, |\chi'(\theta_2)|^{-1/2}, \dots, |\chi'(\theta_n)|^{-1/2})$ after normalization. There is no loss of generality in making the elements of $S^* e_1$ positive.

Case 2: submatrix spectrum. Suppose that, in addition to Θ , the eigenvalues of the $T_{2:n}$ are given. These values μ_i , $i = 1, \dots, n - 1$ cannot be arbitrary but, by the Cauchy interlace theorem (section 10.1), must interlace the θ 's; i.e.,

$$\theta_1 < \mu_1 < \theta_2 < \dots < \mu_{n-1} < \theta_n.$$

Formula (7.21) gives

$$s_{1j}^2 \chi'(\theta_j) = \chi_{2,n}(\theta_j), \quad j = 1, \dots, n,$$

where $\chi_{2,n}(\theta_j) = \prod_{i=1}^{n-1} (\theta_j - \mu_i)$ and is computable. As before, $\chi'(\theta_j) = \prod_{i=1}^n (\theta_j - \theta_i)$, $i \neq j$, and the interlace condition guarantees that $\chi_{2,n}/\chi'$ is positive at the θ 's. Consequently, $|s_{1j}|$ is determined for $j = 1, \dots, n$, and a unique T can be constructed with positive β 's.

Case 2 corresponds to the physical problem in which the discretized string is clamped at the first interior mesh point and the new frequencies are then computed.

Example 7.12.1 gives the result for Case 2 on a small and easy problem.

Example 7.12.1.

θ 's	$\left\{ \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1 \right\}$.
μ 's	$\left\{ \frac{9}{40}, \frac{7}{24}, \frac{5}{12}, \frac{3}{4} \right\}$.
s_{1j}	$\{0.633, 0.437, 0.356, 0.333, 0.413\}$.
α 's	$\{0.6, 0.66326, 0.42933, 0.32508, 0.26566\}$.
β 's	$\{0.34046, 0.16216, 0.085823, 0.044658\}$.

The extent to which the continuous function $\sigma(\xi)$ is determined by the eigenvalues of the equation $-\phi'' + \sigma\phi = \lambda\phi$, where $\phi(a) = \phi(b) = 0$, is a more subtle and challenging problem. The reader is referred to [Hald, 1977] and [Hochstadt, 1975]. Both authors also study the matrix cases given above but they use less stable algorithms for computing T .

Our technique of reducing the matrix $P^* \Theta P$ to tridiagonal form is both simple in concept and stable in action, but it is not efficient for large problems. The techniques described in [De Boor and Golub, 1978], essentially Exercise 7.8.1, turn out to be entirely equivalent to using the basic Lanczos algorithm to reduce Θ to tridiagonal form. This is not stable in the face of roundoff error.

The search for the best algorithm to compute T from Θ and $S^* e_1$ continued during the 1980s. It turns out that the effect of roundoff error is reduced

if one replaces Θ by $M = \text{diag}(\mu_1, \dots, \mu_{n-1})$ and S^*e_1 by a related vector $c = (\gamma_1, \dots, \gamma_{n-1})^*$ such that the arrowhead matrix

$$A = \begin{bmatrix} \tau & c^* \\ c & M \end{bmatrix}$$

is similar to Θ . The right choice of c was given by C. Loewner in a course at Stanford University in 1950s. Each entry is a product of quotients of differences of the form $\theta_i - \mu_j$ or $\theta_i - \theta_j$.

There are one or two ingenious ways to reduce A to T without causing the intermediate fill-in that accompanies the methods of sections 7.4 and 7.5. The method is due to W. Kahan and is not yet published.

Exercises on Section 7.12

- 7.12.1. Show that $\chi'(\theta_j) = \Pi_i(\theta_j - \theta_i)$, $i \neq j$, for each zero of the polynomial χ .
- 7.12.2. Let $\tilde{l} = (e_n, \dots, e_1)$ and suppose $T = \tilde{l}T\tilde{l}$. Use the fact that eigenvectors are determined up to a scalar multiple to show that $s_{1j} = \pm s_{nj}$, $j = 1, \dots, n$. Show that if $\beta_j > 0$ for all i then $s_{1j} = s_{nj}(-1)^{n-j}$.
- 7.12.3. Project. Consider the storage requirements for the various ways of computing T when Θ and S^*e_1 are given. Is it necessary to keep an n -by- n array in fast storage?
- 7.12.4. Project. Write a program to compute $H\Theta H$ where $H = H(w)$, $w = f \pm e_1$, where $f = S^*e_1$. Then call the local library subroutines to reduce $H\Theta H$ to T and compute T 's eigenvalues and compare them with the given θ_i . Try various sets of interlacing μ 's for each set of θ 's.

Notes and References

The goal of reducing A to tridiagonal form T instead of finding the coefficients of χ_A is found in two seminal reports [Lanczos, 1950] and [Givens, 1954]. The transformation is described in several books: [Wilkinson, 1965], [Householder, 1964], [Stewart, 1973], [Fox, 1964] to name a few.

The formulas for the eigenvectors of T in terms of the eigenvalues were used in [Paige, 1971] but are not well known. On the other hand the connection of T with orthogonal polynomials is standard material in analysis [Szegő, 1939] and [Golub and Welsch, 1969]. The criteria for neglecting off-diagonal terms come from the unpublished report [Kahan, 1966].

There is an extensive literature on inverse problems for differential equations. Only since 1980 have there been stable and efficient methods for computing T 's with given spectral properties. See [Gragg and Harrod, 1984].

With an eye to the efficient exploitation of parallel computers a new method for computing eigenvalues and eigenvectors of \mathbf{T} was introduced in the 1980s. Over a decade was needed for this divide-and-conquer technique to become accurate as well as fast. To the surprise of many the method is the 1997 champion on conventional serial computers, not just for parallel ones, and even for values of n as small as 50. See the textbook [Demmel, 1997, Chapter 5] for a description and references. This triumph may not last.

In the late 1990s it gradually became clear that the standard representation of tridiagonals by their entries is an unfortunate one. It is much better, for accuracy and efficiency, to know the matrix as a product of bidiagonals. Such a representation does not always exist, and that may have frightened some investigators. However, as the poet said, “Faint heart never won fair lady.”

There is much to be said on this topic. A place to begin is [Parlett, 1995], [R.-C. Li, Parts I and II, 1994], and [LAPACK Working Note #3, 1988].

The QL and QR Algorithms

8.1. Introduction

Although the QL and QR algorithms were not conceived before 1958 they have emerged as today's preferred way of finding all the eigenvalues of a full symmetric matrix. The matrix is first reduced to tridiagonal form by a sequence of rotations or reflections (section 7.4) and then the QL algorithm swiftly reduces the off-diagonal elements until they are negligible. The algorithm repeatedly applies a complicated similarity transformation to the result of the previous transformation, thereby producing a sequence of matrices that converges to a diagonal matrix. What is more, the tridiagonal form is preserved.

The transformation is based on the orthogonal-triangular decomposition of a matrix B , say, and this decomposition is the matrix formulation of the Gram–Schmidt orthonormalization process (section 6.7) applied to the columns of B . If the columns are taken in the order b_1, b_2, \dots, b_n , then the factorization is $B = Q_R R$ where R is right (or upper) triangular and $Q_R^* = Q_R^{-1}$. If the columns are taken in the reverse order b_n, b_{n-1}, \dots, b_1 , then the result is $B = Q_L L$ where L is left (or lower) triangular and $Q_L^* = Q_L^{-1}$.

A strong psychological deterrent to the discovery of the algorithms must have been the apparently high cost of this factorization. However, the orthonormal matrices Q_L and Q_R which loom large in the rest of this chapter will never be formed explicitly. The QL and QR transformations are defined for any square matrix and so we forsake symmetric matrices for those sections which give the formal definitions and basic properties. Next comes the relation to other, simpler methods together with the very satisfactory convergence theory: with Wilkinson's shift, convergence is guaranteed and swift. Finally there is a discussion of various implementations.

8.2. The QL Transformation

Given B and a scalar σ , called the *origin shift*, consider the orthogonal lower-triangular factorization

$$B - \sigma = QL. \quad (8.1)$$

From Q , L , and σ define \hat{B} , the *QL transform* of B , by

$$\begin{aligned} \hat{B} &\equiv LQ + \sigma, \\ &= Q^*(B - \sigma)Q + \sigma, \quad \text{using (8.1),} \\ &= Q^*BQ, \quad \text{since } Q^* = Q^{-1}. \end{aligned} \quad (8.2)$$

It is not clear that \hat{B} is any improvement over B ; no zero elements have been created; nevertheless the shifted QL algorithm doggedly iterates the transformation $B \rightarrow \hat{B}$.

8.2.1. The QL Algorithm

For $k = 1, 2, \dots$ and given $\{\sigma_k\}$, to be discussed later,

1. let $Q_k L_k$ be the QL factorization of $B_k - \sigma_k$;
2. define $B_{k+1} \equiv Q_k^* B_k Q_k$.

The matrices B_k are all orthogonally similar to each other. The relation of B_{k+1} to B_1 is

$$\begin{aligned} B_{k+1} &= Q_k^* B_k Q_k \\ &= Q_k^* (Q_{k-1}^* B_{k-1} Q_{k-1}) Q_k \\ &= \dots \\ &= P_k^* B_1 P_k, \end{aligned} \quad (8.3)$$

where

$$P_k \equiv Q_1 Q_2 \cdots Q_k \quad (8.4)$$

is a product of orthonormal matrices and hence is orthonormal.

The QR transformation is similar but QR replaces QL in (8.1). Of course, the new Q and the new \hat{B} will differ from the ones given in (8.2).

Exercises on Section 8.2

- 8.2.1. Show that $P_k L_k = (B_1 - \sigma) P_{k-1}$ and hence that $P_k e_n$ is a normalized version of $(B_1 - \sigma) P_{k-1} e_n$.

8.2.2. Show that if $\sigma_k = 0$ for all k then

$$P_k(L_k \cdots L_2 L_1) = B_1^k.$$

8.2.3. Perform one step of the QL algorithm with $\sigma_1 = 0$ on

$$B_1 = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}.$$

8.2.4. Perform one step of the QR algorithm on the example of Figure 8.1.

$$B_1 - \sigma_1 = \begin{bmatrix} 6.0 & 0.6 & 0 \\ 1.0 & 4.0 & 2.0 \\ -3.0 & -0.8 & 0 \end{bmatrix} = \begin{bmatrix} 0.8 & 0.6 & 0 \\ 0 & 0 & 1.0 \\ 0.6 & -0.8 & 0 \end{bmatrix} \begin{bmatrix} 3.0 & & \\ 6.0 & 1.0 & \\ 1.0 & 4.0 & 2.0 \end{bmatrix},$$

$$B_2 - \sigma_1 = \begin{bmatrix} 2.4 & 1.8 & 0 \\ 4.8 & 3.6 & 1.0 \\ 2.0 & -1.0 & 4.0 \end{bmatrix} = \begin{bmatrix} 3.0 & & \\ 6.0 & 1.0 & \\ 1.0 & 4.0 & 2.0 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 & 0 \\ 0 & 0 & 1.0 \\ 0.6 & -0.8 & 0 \end{bmatrix}.$$

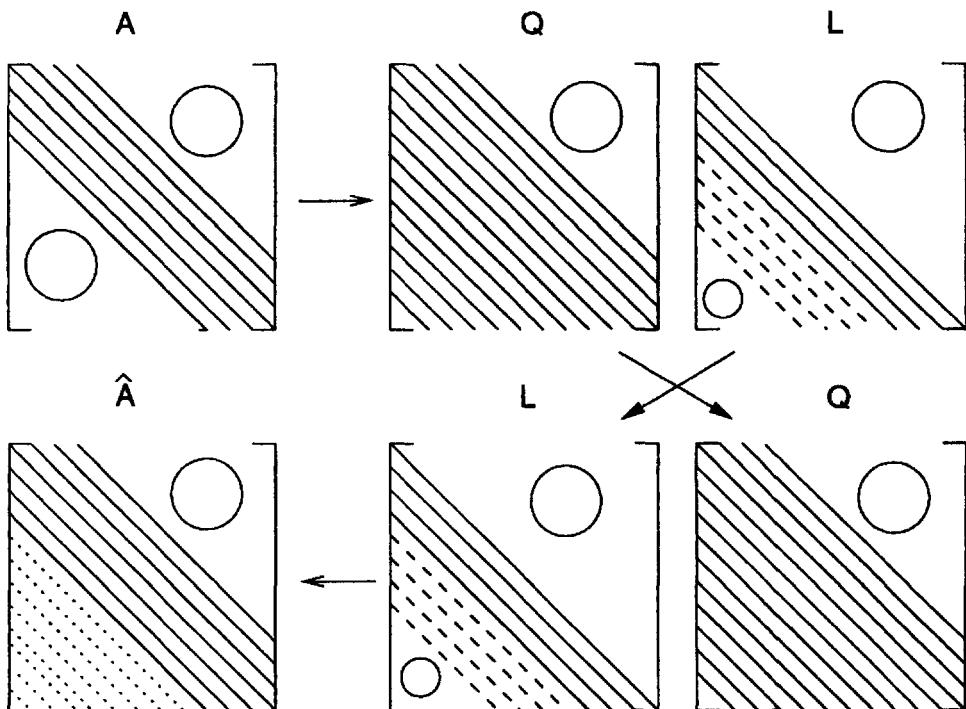
FIG. 8.1. One step of QL.

8.3. PRESERVATION OF BANDWIDTH

If $B = A = A^*$ then symmetry is preserved because the transformations are congruences as well as similarities. The fact that bandwidth is also preserved can be seen from Figure 8.2.

The zero elements *above* the diagonal of \hat{A} are preserved simply by the shape of Q and L ; the elements *below* the diagonal must be zero, through cancellation, since \hat{A} is symmetric.

The preservation of tridiagonal form is very valuable. The fact that such a complicated and far-from-sparse Q_k can preserve the tridiagonal form is quite surprising.



Key: - - - - - Nonzero elements

..... Elements which vanish through cancellation

FIG. 8.2. *Preservation of bandwidth.*

Exercises on Section 8.3

8.3.1. Perform one step of QL with $\sigma_1 = 0$ on

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Exhibit the cancellation in the (3,1) element of \hat{A} .

8.3.2. Show that if $b_{ij} = 0$ for $i < j - 1$ then, under the QL transformation, $\hat{b}_{ij} = 0$ for $i < j - 1$. Hessenberg form is preserved.

8.4. Relation between QL and QR

The QL algorithm is simply a reorganization of the original QR algorithm, and the two methods need not be distinguished (from each other) for theoretical purposes.

To give the formal relationship it is convenient to denote by $Q_R[B]$ the result of orthonormalizing the columns of any square B in the order b_1, b_2, \dots, b_n , while $Q_L[B]$ comes from the reverse order b_n, b_{n-1}, \dots, b_1 . When there is no ambiguity concerning the matrix B we write simply Q_R or Q_L . Also needed is the matrix \tilde{I} obtained by reversing the order of the columns of the identity matrix. Note that $\tilde{I}^* = \tilde{I}^{-1} = \tilde{I}$.

There is no trivial relationship between the orthonormal matrices $Q_R[B]$ and $Q_L[B]$. On the other hand, when B is invertible, we have Lemma 8.4.1.

Lemma 8.4.1.

$$Q_L[\tilde{I} B \tilde{I}] = \tilde{I} Q_R[B] \tilde{I}. \quad (8.5)$$

Proof.

$$\begin{aligned} \tilde{I} B \tilde{I} &= \tilde{I} Q_R[B] R_B \tilde{I} \\ &= (\tilde{I} Q_R[B] \tilde{I})(\tilde{I} R_B \tilde{I}). \end{aligned} \quad (8.6)$$

Since R_B is upper triangular the second line of (8.6) gives the unique QL factorization of $\tilde{I} B \tilde{I}$ as claimed. Uniqueness is proved in section 6.6. \square

Let us return to the two transformations

$$QL : B \rightarrow Q_L^* B Q_L, \quad Q_L = Q_L[B - \sigma],$$

$$QR : B \rightarrow Q_R^* B Q_R, \quad Q_R = Q_R[B - \sigma].$$

Lemma 8.4.2. Let $\{B_k^L\}$ and $\{B_k^R\}$ be sequences generated by the QL and QR algorithms using the same sequence of origin shifts.

If $B_1^R = \tilde{\|B_1^L\|}$ then

$$B_k^R = \tilde{\|B_k^L\|} \text{ for all } k > 1. \quad (8.7)$$

The proof is left as Exercise 8.4.1.

If B_k^L tends to lower-triangular form, as $k \rightarrow \infty$, with the smaller eigenvalues (in absolute value) at the top, then B_k^R tends to upper-triangular form with the smaller eigenvalues at the bottom. For symmetric matrices both the QL and QR algorithm converge to diagonal form. The original reason for introducing the QL algorithm is that certain types of problems yield *graded* matrices in which the smaller elements are already at the top of the matrix and the bigger ones are at the bottom as indicated in section 8.13. Matrices of this form can arise from the generalized problem $A - \lambda M$ when M is ill conditioned but positive definite. The lower-triangular Cholesky factor C satisfying $CC^* = M$ often has columns which shrink steadily,

$$\|c_1\| > \|c_2\| > \|c_3\| > \cdots > \|c_n\|,$$

and the reduced matrix $C^{-1}AC^{-*}$ will be graded like the one in Example 8.13.1.

Exercise on Section 8.4

8.4.1. Prove Lemma 8.4.2.

8.5. QL, the Power Method, and Inverse Iteration

Now we link QL to the more familiar processes described in Chapter 4. Given A and any sequence $\{\sigma_k, k = 1, 2, \dots\}$ of origin shifts the three algorithms of the section heading produce three different sequences:

QL (with $A_1 = A$): $\{A_k\}$ where $A_{k+1} = Q_k^* A_k Q_k$
and $A_k - \sigma_k = Q_k L_k$;

PM (v_1 any unit vector): $\{v_k\}$ where $v_{k+1} = (A - \sigma_k)v_k/\nu_k$,
and $\|v_k\| = 1$;

INVIT (u_1 any unit vector): $\{u_k\}$ where $(A - \sigma_k)u_{k+1} = u_k \tau_k$,
and $\|u_k\| = 1$.

It is interesting that if v_1 and u_1 are chosen appropriately then the three sequences are intimately related. What connects them is the matrix introduced in section 8.2 to connect A_{k+1} to A ; namely,

$$P_k = Q_1 Q_2 \cdots Q_k, \quad P_0 = I. \quad (8.8)$$

Recall, from (8.3), that $A_{k+1} = P_k^* A P_k$.

Theorem 8.5.1. *Assume that no shift σ_k is an eigenvalue of A . If $u_1 = e_1$ and $v_1 = e_n$ then, for $k \geq 1$, $u_k = P_{k-1} e_1$ and $v_k = P_{k-1} e_n$.*

Recall that if A is tridiagonal then so are all the A_k 's and moreover, from section 7.2, A_{k+1} is uniquely determined by A together with either $P_k e_1$ or $P_k e_n$.

Proof. Rewrite $A_{k+1} = P_k^* A P_k$ as

$$P_{k-1} A_k = A P_{k-1}. \quad (8.9)$$

Now

$$\begin{aligned} P_k L_k &= P_{k-1} Q_k L_k, \text{ by (8.8),} \\ &= P_{k-1} (A_k - \sigma_k), \text{ by QL,} \\ &= (A - \sigma_k) P_{k-1} \text{ by (8.9).} \end{aligned} \quad (8.10)$$

Since L_k is lower triangular, equate the last columns of (8.10)

$$P_k e_n = (A - \sigma_k) P_{k-1} e_n / \nu_k, \quad \nu_k = e_n^* L_k e_n. \quad (8.11)$$

Observe that (8.11) defines the power sequence, with shifts, generated by A and $P_0 e_n$. So if in algorithm PM above $v_1 = P_0 e_n (= e_n)$, then $v_{k+1} = P_k e_n$.

To obtain the analogous relation for u_k a little manipulation is needed to turn rows into columns. Transpose (8.10) to obtain

$$L_k^* P_k^* = P_{k-1}^* (A - \sigma_k).$$

Premultiply this by P_{k-1} and postmultiply by P_k to get

$$P_{k-1}L_k^* = (A - \sigma_k)P_k. \quad (8.12)$$

Since L_k^* is upper triangular,

$$P_{k-1}e_1\tau_k = (A - \sigma_k)P_k e_1, \quad \tau_k = e_1^* L_k^* e_1. \quad (8.13)$$

Since $P_0 = I$, (8.13) defines inverse iteration, with shifts, started from e_1 . So if in INVIT above $u_1 = P_0 e_1 = e_1$, then $P_k e_1 = u_{k+1}$. \square

This result raises the question of how these different methods could ever come up with the same shifts σ_k in practice. That question is answered in section 8.7.

8.6. Convergence of the Basic QL Algorithm

When no origin shifts are used, i.e., $\sigma_k = 0$ for all k , then the algorithm is called *basic* and is simultaneously linked to both direct and inverse iteration, without shifts. The convergence of basic QL can be seen as a corollary of the convergence properties of these two simple iterations.

Theorem 8.6.1. *Suppose that A's eigenvalues satisfy*

$$0 < |\lambda_1| < |\lambda_2| \leq \cdots \leq |\lambda_{n-1}| < |\lambda_n| = \|A\|.$$

Let z_i be the normalized eigenvector of λ_i , $i = 1$ and n , and let $\{A_k\}$ be the basic QL sequence derived from $A_1 \equiv A$. If $e_i^ z_i \neq 0$, $i = 1$ and n , then as $k \rightarrow \infty$.*

$$A_k e_1 \rightarrow e_1 \lambda_1, \quad A_k e_n \rightarrow e_n \lambda_n. \quad (8.14)$$

Proof. Let $\{u_k\}$ be the inverse iteration sequence derived from $u_1 = e_1$ with no shifts. The hypotheses, using Theorem 4.2.1 (Corollary 4.2.1), ensure that $u_k = z_1 + O(|\lambda_1/\lambda_2|^k)$ as $k \rightarrow \infty$. By Theorem 8.5.1,

$$e_1 = P_{k-1}^*(P_{k-1} e_1) = P_{k-1}^* u_k = P_{k-1}^* z_1 + O(|\lambda_1/\lambda_2|^k),$$

and after using (8.3),

$$\begin{aligned}\mathbf{A}_k \mathbf{e}_1 &= \mathbf{P}_{k-1}^* \mathbf{A} \mathbf{u}_k \\ &= \mathbf{P}_{k-1}^* [\mathbf{z}_1 \lambda_1 + O(|\lambda_1/\lambda_2|^k)] \\ &= \mathbf{e}_1 \lambda_1 + O(|\lambda_1/\lambda_2|^k), \quad k \rightarrow \infty.\end{aligned}$$

Similarly, using the power sequence $\{\mathbf{v}_k\}$, $\mathbf{A}_k \mathbf{e}_n \rightarrow \mathbf{e}_n \lambda_n$. \square

There are two notions of convergence associated with the QL and QR algorithms. Strictly speaking, convergence would be taken to mean the convergence of the matrix sequence $\{\mathbf{A}_k\}$ to some limit matrix. In practice, however, as soon as $\|\mathbf{A}_k \mathbf{e}_1 - \mathbf{e}_1 a_{11}^{(k)}\|$ is negligible $a_{11}^{(k)}$ is taken as an eigenvalue and the computation then proceeds on the *submatrix* obtained by ignoring the first row and column. In other words, a stable form of deflation is built into the QL and QR algorithms, as discussed in section 5.3. Hence the second notion simply concerns the convergence of the vector sequence $\{\mathbf{A}_k \mathbf{e}_1\}$ to a limit $\mathbf{e}_1 \lambda$.

We shall employ the second meaning exclusively.

Theorem 8.6.1 is important but not exciting. For a full matrix each QL step is expensive (n^3 ops), and convergence is linear with unknown and often very poor convergence factors. The power of the practical algorithm comes from (a) preservation of bandwidth (reducing the cost of a step to $O(n)$ ops for tridiagonals) and (b) the use of origin shifts to reduce the number of steps.

As shown in section 7.9 the hypotheses $\mathbf{e}_1^* \mathbf{z}_1 \neq 0$ and $\mathbf{e}_n^* \mathbf{z}_n \neq 0$ in Theorem 8.6.1 do hold for all unreduced tridiagonal matrices.

We now turn to the choice of origin shifts σ_k .

8.7. The Rayleigh Quotient Shift

The basic QL algorithm on an unreduced tridiagonal \mathbf{A} converges in one step if \mathbf{A} is singular. Thus shifts are chosen to approximate eigenvalues. The convergence of the basic algorithm ensures that $a_{11}^{(k)} = \mathbf{e}_1^* \mathbf{A}_k \mathbf{e}_1 \rightarrow \lambda_1$ as $k \rightarrow \infty$, and so it seems reasonable to use $a_{11}^{(k)}$ as a shift *after* this element has settled down to some extent. However, the shift strategy discussed below casts caution aside and uses $a_{11}^{(k)}$ as the shift right from the start. Formally,

$$\sigma_k = a_{11}^{(k)} = \mathbf{e}_1^* \mathbf{A}_k \mathbf{e}_1, \quad k = 1, 2, \dots \quad (8.15)$$

Surprisingly it turns out that σ_k is the same Rayleigh quotient ρ_k used in RQI (section 4.6). To distinguish RQI from INVIT, of which it is a special case, we write x_k instead of u_k .

Theorem 8.7.1. *If RQI is started with $x_1 = e_1$ and QL uses (8.15) for shifts then ρ_k ($\equiv x_k^* A x_k$) = σ_k for all k .*

Proof. Initially,

$$\sigma_1 = e_1^* A e_1 = x_1^* A x_1 = \rho_1.$$

Assume, as induction hypothesis, that $\sigma_k = \rho_k$ for $k = 1, \dots, j$. Then Theorem 8.5.1 shows that $P_k e_1 = x_{k+1}$ for $k = 1, \dots, j$. Hence

$$\sigma_{j+1} = e_1^* A_{j+1} e_1 = e_1^* P_j^* A P_j e_1 = x_{j+1}^* A x_{j+1} = \rho_{j+1}.$$

By the principle of induction the result holds for all k . \square

All the convergence properties of RQI (sections 4.7, 4.8, and 4.9) can now be translated into statements about QL with this particular shift.

In particular let $r_k \equiv (A - \rho_k)x_k$ denote the residual vector of x_k and let QL, using (8.15), produce

$$A_k = \begin{bmatrix} \alpha_k & b_k^* \\ b_k & M_k \end{bmatrix}.$$

Then (Exercise 8.7.5), $\|b_k\| = \|r_k\|$. Further let $\phi_k = \angle(x_k, z_1)$, the error angle. If $\phi_k \rightarrow 0$ as $k \rightarrow \infty$, then (Exercise 8.7.3)

$$\|r_k\| / |\sin \phi_k| \rightarrow |\lambda_2 - \lambda_1|. \quad (8.16)$$

The ultimate cubic convergence of ϕ_k to zero (Theorem 4.7.1) entails the same behavior for $\|r_k\| (= \|b_k\|)$. The usual proofs that $\|b_{k+1}\| / \|b_k\|^3 \rightarrow \text{constant}$ as $k \rightarrow \infty$ (assuming that $\|b_k\| \rightarrow 0$) are rather complicated.

Turning to the global behavior of QL we conclude that $\|b_k\| \leq \|b_{k-1}\|$ for all k (from section 4.8) and $\|b_k\| \rightarrow 0$ for almost all A (section 4.9). However, nondiagonal matrices do exist which are invariant under QL with the Rayleigh quotient shift. The simplest example is

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

However, the slightest perturbation of this A leads to convergence.

Although such matrices must exist no one has exhibited an A_1 such that, in exact arithmetic, the QL sequence $\{A_k\}$, using (8.15), neither converges nor is stationary. The simple Rayleigh quotient shift strategy makes QL a powerful tool for diagonalizing matrices. In section 8.9 we turn to a strategy which is slightly more complicated and even more satisfactory.

Exercises on Section 8.7

8.7.1. State and prove the analogue of Theorem 8.7.1 for the QR algorithm.

8.7.2. Compute A_2 when

$$A_1 = \begin{bmatrix} 10 & 1 & 0 \\ 1 & 20 & 2 \\ 0 & 2 & 30 \end{bmatrix}$$

and $\sigma_1 = 10$.

8.7.3. Using the expansion $x_k = z_1 \cos \phi_k + w_k \sin \phi_k$, $z_1^* w_k = 0$, $\|w_k\| = 1$, show that

$$\|r_k\|^2 = \sin^2 \phi_k \left[\|(\mathbf{A} - \rho_k)w_k\|^2 + \frac{1}{4}(\lambda_1 - \rho(w_k))^2 \sin^2 2\phi_k \right]$$

and hence establish (8.16).

8.7.4. Prove that, as $k \rightarrow \infty$ and $x_k \rightarrow z_1$,

$$|\lambda_1 - \rho_{k+1}| / |\lambda_1 - \rho_k|^3 \rightarrow 1.$$

8.7.5. Show that although $b_k \neq r_k$ we have $\|b_k\| = \|r_k\|$ using the partition of A_k shown above.

8.7.6. Show that the basic QL algorithm applied to a singular unreduced tridiagonal matrix must converge in one step.

8.8. The Off-Diagonal Elements

In the context of inverse iteration the only vector on hand at the k th step is the current eigenvector approximation u_k , and it is natural to choose its Rayleigh quotient as shift. However, in the context of the QL algorithm A_k is available and so more accurate approximations to λ_1 can be obtained at negligible cost.

Before discussing such shifts in detail we look at the result of a single QL transformation with any shift σ when A is tridiagonal. The formulas are

$$A = T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & & \ddots & \beta_{n-1} \\ & & & & \alpha_n \end{bmatrix},$$

$$T - \sigma = QL, \quad \hat{T} = Q^* T Q. \quad (8.17)$$

By the invariance of bandwidth \hat{T} is also tridiagonal and so, in a formal sense, \hat{T} is the result of “reducing” T to tridiagonal form using Q . By Theorem 7.2.1 \hat{T} is completely determined by T and either q_1 or q_n . Consider the last column on each side of $T - \sigma = QL$ to see that q_n is a multiple of $(0, \dots, 0, \beta_{n-1}, \alpha_n - \sigma)^*$, and this fact is used in *implementing* the transformation as discussed in sections 8.12 and 8.13. It turns out that the description of \hat{T} in terms of q_1 rather than q_n keeps the analysis simple. If σ is an eigenvalue, the algorithm will converge immediately; therefore, we need only consider the case when σ is not an eigenvalue.

The relation with inverse iteration (Theorem 8.5.1, $k = 2$) gives

$$q_1 (= P_1 e_1) = (T - \sigma)^{-1} e_1 \tau$$

or

$$\begin{aligned} (T - \sigma) q_1 &= e_1 \tau, \\ \tau &= 1 / \| (T - \sigma)^{-1} e_1 \| = \| (T - \sigma) q_1 \| . \end{aligned} \quad (8.18)$$

The normalization factor τ plays a central role in what follows. The results of section 7.3 characterize the new off-diagonal elements $\hat{\beta}_i$ in terms of certain monic polynomials. In particular,

$$\begin{aligned} |\hat{\beta}_1| &= \min \|\phi(T) q_1\|, \text{ over all } \phi(\xi) = \xi - \mu, \\ &\leq \| (T - \sigma) q_1 \| = \tau, \text{ by (8.18),} \end{aligned} \quad (8.19)$$

and

$$\begin{aligned} |\hat{\beta}_1 \hat{\beta}_2| &= \min \|\phi(T) q_1\|, \text{ over all monic } \phi \text{ of degree 2,} \\ &\leq \| (T - \alpha_1)(T - \sigma) q_1 \|, \text{ the artful choice,} \\ &= \| (T - \alpha_1) e_1 \tau \|, \text{ by (8.18),} \\ &= \| e_2 \beta_1 \tau \| = |\beta_1| \tau. \end{aligned} \quad (8.20)$$

These relations (8.19) and (8.20), which hold for all $\sigma \neq \lambda_j[\mathbf{T}]$, show the potential usefulness of the normalizing factor τ given in (8.18).

Although exact expressions for $\hat{\beta}_1$ and $\hat{\beta}_1\hat{\beta}_2$ are available (section 8.11), they are more complicated than upper bounds on τ and for our purposes the bounds are adequate.

8.9. Residual Bounds Using Wilkinson's Shift

Given \mathbf{T} , as in (8.17), then Wilkinson's shift ω is that eigenvalue of $\begin{bmatrix} \alpha_1 & \beta_1 \\ \beta_1 & \alpha_2 \end{bmatrix}$ which is closer to α_1 . In case of a tie ($\alpha_1 = \alpha_2$) we choose the smaller, namely, $\alpha_1 - |\beta_1|$. This shift is obviously better than α_1 when either β_1 or β_2 is small. One formula for ω is

$$\omega = (\alpha_1 + \alpha_2)/2 - \text{sign}(\delta)\sqrt{\delta^2 + \beta_1^2} \quad (\text{sign}(0) = 1),$$

where $\delta = (\alpha_2 - \alpha_1)/2$, but a better one (Exercise 8.9.1) is

$$\omega = \alpha_1 - \text{sign}(\delta)\beta_1^2 / \left(|\delta| + \sqrt{\delta^2 + \beta_1^2} \right).$$

A glance at Figure 8.3 shows that

$$|\alpha_1 - \omega| \leq |\alpha_2 - \omega|, \quad (8.21)$$

with equality if and only if $\delta = 0$. By noting that $|\beta_1|$ is the geometric mean of $|\alpha_1 - \omega|$ and $|\alpha_2 - \omega|$ we have

$$\frac{|\alpha_1 - \omega|}{|\beta_1|} = \frac{|\beta_1|}{|\alpha_2 - \omega|} = \sqrt{\frac{|\alpha_1 - \omega|}{|\alpha_2 - \omega|}} \leq 1, \quad (8.22)$$

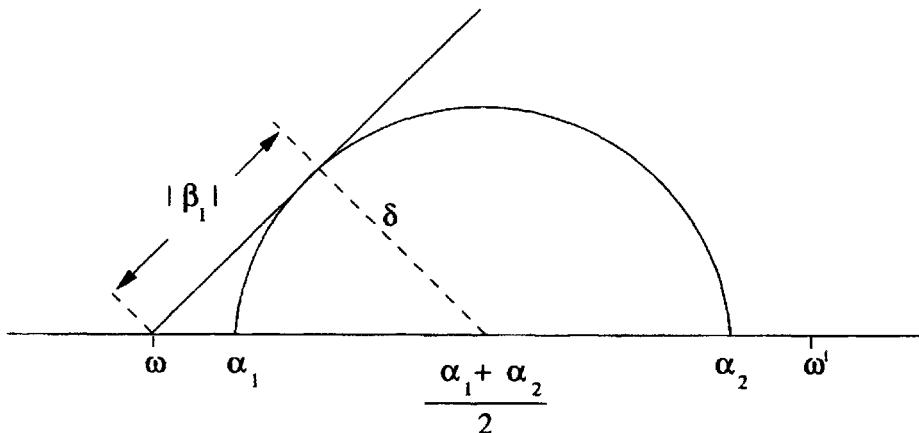
with equality if and only if $\delta = 0$.

Our goal, following (8.20), is to bound $\tau = \|(\mathbf{T} - \omega)\mathbf{q}_1\| = 1/\|\mathbf{p}\|$ where \mathbf{p} is defined by

$$(\mathbf{T} - \omega)\mathbf{p} = \mathbf{e}_1. \quad (8.23)$$

Lemma 8.9.1. *If Wilkinson's shift ω is not an eigenvalue of \mathbf{T} , then the unit vector \mathbf{q}_1 is determined by $(\mathbf{T} - \omega)\mathbf{q}_1 = \mathbf{e}_1\tau$ and satisfies*

$$\|(\mathbf{T} - \omega)\mathbf{q}_1\|^2 = \tau^2 \leq \min\{2\beta_1^2, \beta_2^2, |\beta_1\beta_2|/\sqrt{2}\}. \quad (8.24)$$

FIG. 8.3. Wilkinson's shift ω .

Proof. Let p in (8.23) have elements $\pi_1, \pi_2, \dots, \pi_n$. Then

$$\tau^2 = 1/\|p\|^2 \leq 1/(\pi_1^2 + \pi_2^2 + \pi_3^2). \quad (8.25)$$

For convenience let $\bar{\alpha}_i = \alpha_i - \omega$. Then the first two equations in (8.23) may be written

$$\bar{\alpha}_1 \pi_1 + \beta_1 \pi_2 = 1, \quad (8.26)$$

$$\beta_1 \pi_1 + \bar{\alpha}_2 \pi_2 + \beta_2 \pi_3 = 0. \quad (8.27)$$

Eliminate π_1 and use $\bar{\alpha}_1 \bar{\alpha}_2 = \beta_1^2$ to find

$$0 + 0 + \beta_2 \pi_3 = -\beta_1 / \bar{\alpha}_1. \quad (8.28)$$

It is remarkable that π_3 depends only on $\bar{\alpha}_1, \beta_1, \beta_2$. In contrast π_1 and π_2 depend on all the elements of $T - \omega$. However, a simple bound on $\pi_1^2 + \pi_2^2$ comes from the linear equation (8.26) and elementary geometry (Exercise 8.9.2)

$$\pi_1^2 + \pi_2^2 \geq 1/(\bar{\alpha}_1^2 + \beta_1^2). \quad (8.29)$$

Substitute (8.28) and (8.29) into (8.25) to get

$$\tau^2 \leq \left\{ \frac{1}{\bar{\alpha}_1^2 + \beta_1^2} + \frac{\beta_1^2}{\bar{\alpha}_1^2 \beta_2^2} \right\}^{-1}. \quad (8.30)$$

This is more complicated than necessary. By (8.22)

$$\tau^2 \leq (1/2\beta_1^2 + 1/\beta_2^2)^{-1} \leq \min\{2\beta_1^2, \beta_2^2\}. \quad (8.31)$$

The middle expression in (8.31) is half the harmonic mean of $2\beta_1^2$ and β_2^2 . Finally, because the geometric mean exceeds the harmonic mean, (8.31) also yields

$$\tau^2 \leq \frac{1}{2} \sqrt{(2\beta_1^2)\beta_2^2} \quad (8.32)$$

and the lemma is proved. \square

Exercises on Section 8.9

- 8.9.1. Show that both formulas for ω , above (8.21), are the same in exact arithmetic. Find an example in which the first formula loses half the digits and show that the second formula cannot suffer such a loss.
- 8.9.2. Find the point (ξ, η) on the line $\lambda\xi + \mu\eta = 1$ which is closest to the origin.

8.10. Tridiagonal QL Always Converges

For any unreduced tridiagonal matrix T_1 the QL algorithm produces a sequence of unreduced tridiagonal matrices $\{T_k, k = 1, 2, \dots\}$, and the glorious fact is that, with Wilkinson's shift, $\beta_1^{(k)} \rightarrow 0$ rapidly as $k \rightarrow \infty$. The only assumption is that the arithmetic is done exactly.

Several practical advantages accrue from this theory: (1) there is no need to test for the right moment to switch from one shift strategy to another; (2) there is no need to add statements to the programs to check for, and cope with, rare special cases; and (3) an upper limit, such as 30, can be put on the number of iterations allowed to find any eigenvalue. This last feature guarantees quick execution in finite precision without essentially restricting the algorithm's applicability. See Exercise 8.10.1.

In contrast to the Rayleigh quotient shift strategy where $\sigma_k = \alpha_1^{(k)}$, the off-diagonal element $\beta_1^{(k)}$ need not decrease at each step when Wilkinson's shift is used. Monotonicity has been sacrificed to win guaranteed convergence to zero.

One difficulty in analyzing a nonmonotonic process is to know how many consecutive steps must be considered in order to capture the essential pattern. Fortunately one step suffices in our case because the quantity $\beta_1^2\beta_2$ does decline monotonically to zero and also dominates $\hat{\beta}_1^3$.

Theorem 8.10.1. *The tridiagonal QL algorithm using Wilkinson's shift always converges, i.e., $\beta_1^{(k)} \rightarrow 0$ as $k \rightarrow \infty$.*

Proof. Let T and \hat{T} be consecutive terms in the tridiagonal QL algorithm. By Lemma 8.9.1,

$$\hat{\beta}_1^2 < \tau^2 \leq \min\{2\beta_1^2, \beta_2^2, |\beta_1\beta_2|/\sqrt{2}\}.$$

Now combine the first and third candidates for this minimum,

$$|\hat{\beta}_1^3| = |\hat{\beta}_1| \cdot |\hat{\beta}_1^2| < (\sqrt{2}|\beta_1|)(|\beta_1\beta_2|/\sqrt{2}) = |\beta_1^2\beta_2|. \quad (8.33)$$

Next consider the sequence $\{(\beta_1^{(k)})^2\beta_2^{(k)}\}$. The product of (8.19), $|\hat{\beta}_1| < \tau$, and (8.20), $|\hat{\beta}_1\hat{\beta}_2| < |\beta_1|\tau$, yields

$$|\hat{\beta}_1^2\hat{\beta}_2| < |\beta_1|\tau^2 \leq |\beta_1| \cdot |\beta_1\beta_2|/\sqrt{2}. \quad (8.34)$$

Consequently, as $k \rightarrow \infty$,

$$|\beta_1^{(k+1)}|^3 < |(\beta_1^{(k)})^2\beta_2^{(k)}| < |(\beta_1^{(1)})^2\beta_2^{(1)}|/(\sqrt{2})^{k-1} \rightarrow 0. \quad \square$$

The next section shows that for large enough k , which often means $k > 2$, the actual convergence rate is much better than indicated in the proof given above. The importance of Theorem 8.10.1 is that convergence is guaranteed and is at least linear right from the start.

It is most satisfactory that the modest change from the Rayleigh quotient shift ($\sigma = a_{11}$) to Wilkinson's shift ($\sigma = \omega$) transforms the nature of the convergence from almost always to always. However with the Rayleigh quotient shift there is no restriction to tridiagonal matrices; the theory in Chapter 4 is coordinate-free. One might ask if there is a shift strategy which guarantees convergence of inverse iteration from any starting vector. The answer is yes because the tridiagonal assumption is not a real restriction but rather a convenient normalization. The only task is to formulate Wilkinson's shift in geometric form and this is quickly done.

Let u be the current vector in inverse iteration for a given symmetric matrix A . By the uniqueness of reduction (Theorem 7.2.1) there is a unique orthogonal transformation Q such that $Q^*u = e_1$ and $Q^*AQ = T$ is tridiagonal. Now we pull back the shift ω from the tridiagonal form to the original setting.

Definition 8.10.1. Given A and u , with $\|u\| = 1$, define

$$\alpha_1 = u^* A u, \quad r = A u - u \alpha_1, \quad \beta_1 = \|r\|,$$

$$\alpha_2 = r^* A r / \beta_1^2, \quad W_2 = \begin{bmatrix} \alpha_1 & \beta_1 \\ \beta_1 & \alpha_2 \end{bmatrix}.$$

Then the eigenvalue ω of W_2 which is closer to α_1 is Wilkinson's shift for inverse iteration:

$$(A - \omega) \hat{u} = u \tau.$$

When A is full the computation of ω requires the formation of Ar as well as Au , a heavy price. With the QL algorithm $u = e_1$ and it is only the computation of Ar which discourages the use of ω for general matrices. However, when A has bandwidth $2m + 1$ the cost of Ar is m^2 ops and when $m \ll n$ it seems a pity not to use ω and enjoy the fruits of guaranteed convergence.

Exercises on Section 8.10

- 8.10.1. Let T_0 be your favorite 10-by-10 tridiagonal matrix. Let $T(m, \delta)$ be the tridiagonal matrix obtained by linking m copies of T_0 as indicated.

$$T(3, \delta) = \left[\begin{array}{c|c|c} T_0 & O & O \\ \hline O & \square & O \\ \hline O & T_0 & O \\ \hline O & O & T_0 \end{array} \right] \quad \blacksquare = \delta$$

Use a local QL or QR program to compute the eigenvalues of $T(m, \delta)$ and plot the maximum number of iterations needed for any eigenvalue against small values of δ ($\epsilon/10, \epsilon, 10\epsilon, \dots, 10^5\epsilon$) for various values of m (5, 10, 15, 20). We find that it is possible to have QL converge slowly. It is not clear how to choose the shift to make just one of the δ 's decrease.

- 8.10.2. Justify the given definition of Wilkinson's shift for A 's which are not tridiagonal.

8.11. Asymptotic Convergence Rates

With the Rayleigh quotient shift QL converges almost always, and when it does so the asymptotic convergence rate is cubic; in the tridiagonal case successive values of β_1 might be $10^{-1}, 10^{-3}, 10^{-9}, 10^{-27}$. With Wilkinson's shift the asymptotic rate is better than cubic, almost always, but no one has been able to rule out the possibility of (mere) quadratic convergence to a certain very special limit matrix. In discussing the asymptotic regime it is convenient to suppress the iteration count k on which all quantities depend.

A typical step of QL transforms tridiagonal T into $\hat{T} = Q^*TQ$ where

$$|\hat{\beta}_1| < \tau = 1/\|(\hat{T} - \sigma)^{-1}\mathbf{e}_1\|, \quad (8.35)$$

and σ is the shift. The bound on τ in Lemma 8.9.1, which was crucial for establishing global convergence, is too crude for the asymptotic regime.

First we obtain an exact expression for $|\hat{\beta}_1|$.

Lemma 8.11.1. *Let $\hat{T} = Q^*TQ$ be the QL transform of T with shift σ , i.e.,*

$$(\hat{T} - \sigma)\mathbf{q}_1 = \mathbf{e}_1\tau. \quad (8.36)$$

Then

$$|\hat{\beta}_1| = \tau |\sin \angle(\mathbf{q}_1, \mathbf{e}_1)|. \quad (8.37)$$

Proof. Let $\theta = \angle(\mathbf{q}_1, \mathbf{e}_1)$. Rearrange $Q\hat{T}\mathbf{e}_1 = T\mathbf{q}_1$ to find

$$\begin{aligned} \mathbf{q}_2\hat{\beta}_1 &= T\mathbf{q}_1 - \mathbf{q}_1\hat{\alpha}_1 \\ &= (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^*)T\mathbf{q}_1, \text{ since } \hat{\alpha}_1 = \mathbf{q}_1^*T\mathbf{q}_1, \\ &= (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^*)(\mathbf{q}_1\sigma + \mathbf{e}_1\tau), \text{ by (8.36),} \\ &= \tau(\mathbf{e}_1 - \mathbf{q}_1 \cos \theta). \end{aligned}$$

Since \mathbf{q}_2 is a unit vector

$$\hat{\beta}_1^2 = \tau^2(1 - 2\mathbf{q}_1^*\mathbf{e}_1 \cos \theta + \cos^2 \theta) = \tau^2 \sin^2 \theta. \quad \square$$

Normally as $\beta_1 \rightarrow 0$ so do β_2 and β_3 but much more slowly. Consequently α_1, α_2 , and α_3 all approach eigenvalues, but precisely which eigenvalues we cannot say in general. The outcome depends strongly on the initial shift. However, when the eigenvalues are found in the natural order we can make a precise statement about the asymptotic rate of convergence. As above we suppress the iteration index k .

The element $\hat{\beta}_1$ is completely determined by \mathbf{q}_1 . A convenient multiple of \mathbf{q}_1 is the vector $\mathbf{p} = (\pi_1, \pi_2, \dots, \pi_n)^*$ defined in (8.23) by

$$(\mathbf{T} - \sigma)\mathbf{p} = \mathbf{e}_1. \quad (8.38)$$

Theorem 8.11.1. *Let the QL algorithm with Wilkinson's shift be applied to an unreduced tridiagonal matrix \mathbf{T} . Then, as $k \rightarrow \infty$, $\beta_1 \rightarrow 0$. If, in addition,*

$$\beta_2 \rightarrow 0, \beta_3 \rightarrow 0, \text{ and } \alpha_i \rightarrow \lambda_i[\mathbf{T}], \quad i = 1, 2, 3, \quad (8.39)$$

then, as $k \rightarrow \infty$,

$$|\hat{\beta}_1/\beta_1^3\beta_2^2| \rightarrow 1/|\lambda_2 - \lambda_1|^3|\lambda_3 - \lambda_1| \neq 0. \quad (8.40)$$

It is convenient to let $\bar{\alpha}_i \equiv \alpha_i - \omega$. Then by (8.22), $\bar{\alpha}_1\bar{\alpha}_2 = \beta_1^2$.

Proof. When $\sigma = \omega$ the first three equations in (8.38) may be solved in terms of π_4 to obtain (Exercise 8.11.1),

$$\begin{aligned} \pi_1 &= -\frac{\bar{\alpha}_2^2\bar{\alpha}_3}{\beta_1^2\beta_2^2} + \frac{\bar{\alpha}_2\beta_3\pi_4}{\beta_1\beta_2} + \frac{\bar{\alpha}_2}{\beta_1^2}, \\ \pi_2 &= \frac{\bar{\alpha}_2\bar{\alpha}_3}{\beta_1\beta_2^2} - \frac{\beta_3\pi_4}{\beta_2}, \\ \pi_3 &= -\frac{\bar{\alpha}_2}{\beta_1\beta_2}. \end{aligned} \quad (8.41)$$

The guaranteed convergence of QL ensures that $\beta_1 \rightarrow 0$, $\bar{\alpha}_1 \rightarrow 0$, as $k \rightarrow \infty$, but the fate of β_2 is not certain. Assumption (8.39) gives $\beta_2 \rightarrow 0$ and, what is crucial, also guarantees that all the other elements of \mathbf{p} , namely, π_4, \dots, π_n , are $O(|\beta_3\pi_3|)$ as $k \rightarrow \infty$. The demonstration of

this is left as Exercise 8.11.2. Eventually π_1 and π_2 dominate p and the first terms on the right in (8.41) dominate π_1 and π_2 . So, as $k \rightarrow \infty$,

$$\tau = \frac{1}{\|p\|} \sim \frac{\beta_1^2 \beta_2^2}{\bar{\alpha}_2^2 |\bar{\alpha}_3|},$$

$$|\sin \theta| = \left\{ \frac{\pi_2^2 + \cdots + \pi_n^2}{\pi_1^2 + \pi_2^2 + \cdots + \pi_n^2} \right\}^{\frac{1}{2}} \sim \left| \frac{\pi_2}{\pi_1} \right| \sim \left| \frac{\beta_1}{\bar{\alpha}_2} \right|.$$

The result follows from Lemma 8.11.1. \square

In practice the convergence of β_1 is so rapid that calculations with only 14 decimal digits rarely let β_2 and β_3 enter the asymptotic regime before deflation occurs. Example 8.11.1 illustrates this phenomenon.

Example 8.11.1. Local convergence of QL

$$A = (a_{ij}), \quad i, j = 1, \dots, 10, \quad \text{where } a_{ij} = \begin{cases} 2i - 1 & \text{for } i = j, \\ 1 & \text{for } i - j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

i	1	2	3	4
$\lambda_i[A]$	0.549129	2.95307	4.99785	6.99995

Step	Shift	$ \hat{\beta}_1/\beta_1^2 \beta_2^2 $	β_1	β_2	β_3	β_4
0	—	—	1.0	1.0	1.0	1.0
1	0.585786	0.01724	-0.01724	0.5427	0.6825	0.7589
2	0.549132	0.01434	2×10^{-8}	0.2991	0.4736	0.5801
3	0.549129	—	10^{-21}	0.1617	0.3280	0.4437

$$0.016181 = 1/61.801 = 1/[(\lambda_2 - \lambda_1)^3(\lambda_3 - \lambda_1)]$$

Step	Shift	$ \hat{\beta}_1/\beta_2^2 \beta_3^2 $	β_1	β_2	β_3	β_4
3	—	—	10^{-21}	0.1617	0.3280	0.4437
4	2.95323	0.02778	10^{-21}	-1×10^{-5}	0.1663	0.2976
5	2.95307	—	10^{-21}	10^{-18}	0.08393	0.1994

$$0.028902 = 1/34.599 = 1/(\lambda_3 - \lambda_2)^3(\lambda_4 - \lambda_2)$$

It remains to indicate why assumption (8.39) might fail. Consider the matrix

$$\mathbf{T}_\infty = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & * \\ 0 & 0 & * & * \end{bmatrix}.$$

If a QL program failed to recognize that $\beta_1 = 0$ then QL with Wilkinson's shift would leave \mathbf{T}_∞ invariant. It seems possible that some delicate perturbation of \mathbf{T}_∞ , as starting matrix, might cause QL to converge to \mathbf{T}_∞ . In such a case $\beta_2 \not\rightarrow 0$ while $\bar{\alpha}_2 \rightarrow 0$. However π_1 still dominates p although it is the third term in the expression which brings this about. Then as $k \rightarrow \infty$,

$$\tau = O(\beta_1^2 / |\bar{\alpha}_2|) = O(|\bar{\alpha}_1|),$$

$$|\sin \theta| = O(|\beta_1|),$$

$$|\hat{\beta}_1| = O(|\bar{\alpha}_1 \beta_1|) = O(\beta_1^2).$$

Thus convergence is quadratic even in these circumstances.

Here ends our discussion of the convergence of the QL algorithm. The remaining sections turn to the implementation problems.

Exercises on Section 8.11

8.11.1. Solve (8.38) to obtain (8.41).

8.11.2. Let $\mathbf{p}^* = (\pi_1, \pi_2, \pi_3, \mathbf{s}^*)$. Show that $\mathbf{e}_1 \pi_3 \beta_3 + (\mathbf{T}_{4:n} - \omega) \mathbf{s} = \mathbf{o}$. Use the Cauchy interlace Theorem 10.1.1 to show that $\|(\mathbf{T}_{4:n} - \omega)^{-1}\|$ is bounded as $\omega \rightarrow \lambda_1$. Conclude that $|\pi_n/\pi_3| = O(\beta_3)$ as $k \rightarrow \infty$. Assume that (8.39) holds. See (7.1) for definition of $\mathbf{T}_{4:n}$.

8.11.3. Investigate the asymptotic convergence rate under the assumption that $\beta_2 \rightarrow 0$, $\bar{\alpha}_3 \rightarrow 0$, $\beta_3 \not\rightarrow 0$.

8.12. Tridiagonal QL with Explicit Shift

The tridiagonal form is preserved under the QL transformation (section 8.3), and this section shows how economically \mathbf{T} can be turned into $\hat{\mathbf{T}} \equiv \mathbf{Q}^* \mathbf{T} \mathbf{Q}$. If σ is the shift then $\mathbf{T} - \sigma = \mathbf{Q} \mathbf{L}$ and the orthogonal matrix \mathbf{Q} must be a so-called *lower Hessenberg matrix* (i.e., $q_{ij} = 0$ whenever $i - j < 1$), as a glance at Figure 8.2 reveals. Fortunately \mathbf{Q} need never be formed explicitly as we now show.

Imagine the reduction of $T - \sigma$ to L by a sequence of $n - 1$ plane rotations

$$Q^*(T - \sigma) = R_1 \cdots R_{n-1}(T - \sigma) = L, \quad (8.42)$$

where each rotation $R_j = R(j, j + 1, \theta_j)$ is chosen to annihilate the $(j, j + 1)$ element of the matrix on hand. The plane rotations are described in section 6.4. Let us take a few snapshots of the process. It is convenient to let c_j denote $\cos \theta_j$ and s_j denote $\sin \theta_j$.

We start with the two rows at the bottom and let $\bar{\alpha}_i = \alpha_i - \sigma$.

$$\begin{aligned} & \begin{bmatrix} c_{n-1} & -s_{n-1} \\ s_{n-1} & c_{n-1} \end{bmatrix} \begin{bmatrix} \cdot & \cdot & 0 & \beta_{n-2} & \bar{\alpha}_{n-1} & \beta_{n-1} \\ \cdot & \cdot & \cdot & 0 & \beta_{n-1} & \bar{\alpha}_n \end{bmatrix} \\ &= \begin{bmatrix} \cdot & \cdot & 0 & c_{n-1}\beta_{n-2} & \pi_{n-1} & 0 \\ \cdot & \cdot & \cdot & s_{n-1}\beta_{n-2} & * & \zeta_n \end{bmatrix}, \end{aligned} \quad (8.43)$$

where $\zeta_n^2 = \beta_{n-1}^2 + \bar{\alpha}_n^2$, $\pi_{n-1} = c_{n-1}\bar{\alpha}_{n-1} - s_{n-1}\beta_{n-1}$, $c_{n-1}\beta_{n-1} - s_{n-1}\bar{\alpha}_n = 0$. The last equation determines θ_{n-1} , but of course we only need $s_{n-1} = \beta_{n-1}/\zeta_n$ and $c_{n-1} = \bar{\alpha}_n/\zeta_n$. Next we apply R_{n-2} ; rows $n - 2$ and $n - 1$ become

$$\begin{aligned} & \begin{bmatrix} c_{n-2} & -s_{n-2} \\ s_{n-2} & c_{n-2} \end{bmatrix} \begin{bmatrix} \cdot & \cdot & 0 & \beta_{n-3} & \bar{\alpha}_{n-3} & \beta_{n-1} & 0 \\ \cdot & \cdot & \cdot & 0 & c_{n-1}\beta_{n-2} & \pi_{n-1} & 0 \end{bmatrix} \\ &= \begin{bmatrix} \cdot & \cdot & 0 & c_{n-2}\beta_{n-3} & \pi_{n-2} & 0 & 0 \\ \cdot & \cdot & \cdot & s_{n-2}\beta_{n-3} & * & \zeta_{n-1} & 0 \end{bmatrix}, \end{aligned}$$

and so on until L is obtained.

The second stage builds up \hat{T} from

$$\hat{T} - \sigma = LQ = LR_{n-1}^* \cdots R_1^*. \quad (8.44)$$

The trouble with such a straightforward approach is the need to store all the c_i and s_i , $i = n - 1, n - 2, \dots, 2$, from the first stage given in (8.42). However, when $R_{n-2}R_{n-1}(T - \sigma)$ has been computed the last two *columns* of L are already in their final form. Consequently R_{n-1}^* can be applied on the right at once, without waiting for L to be formed completely. After that is done c_{n-1} and s_{n-1} are no longer needed. The calculation actually proceeds in an even more intertwined fashion than we have suggested.

Because \hat{T} is tridiagonal the $(j, j - 2)$ elements of L must be annihilated eventually. They do not affect any other quantities in the calculation and can be ignored. In fact, by ingenious algebraic manipulations $\hat{T} - \sigma$ can be written over T with the aid of only six temporary words of storage, $(n - 1)$ square roots, $3(n - 1)$ divisions, $10(n - 1)$ multiplications, and $10(n - 1)$ additions. The fluctuation in the ratios of the execution times of the basic arithmetic

operations makes it difficult to summarize the operation count both neatly and accurately.

Full details concerning the algorithm are given in Contribution II/3 in the Handbook, the corresponding sections of the EISPACK guide, and the LAPACK reference manual (see section 2.8).

The shift σ is not restored; it is $\hat{T} - \sigma$ which is computed. Consequently the running sum of the shifts must be maintained during the iteration.

Occasionally an unfortunate early choice of shift can wipe out relevant information in small α 's, and so we turn to an alternative implementation.

8.13. Chasing the Bulge

There is a way to effect the transformation $T \rightarrow \hat{T} = Q^*TQ$ so that the *only* modifications to T are plane rotations and there is no need to subtract σ from the diagonal elements.

The first rotation R_{n-1} is chosen exactly as in the previous section, namely, $\tan \theta_{n-1} = \beta_{n-1}/(\alpha_{n-1} - \sigma)$, and T is premultiplied by R_{n-1} as shown in (8.43). Next, instead of premultiplying by R_{n-2} the matrix is postmultiplied by R_{n-1}^* thus completing a similarity transformation of T . Applying R_{n-1}^* to (8.43) gives

$$\begin{bmatrix} \dots & \alpha_{n-2} & c_{n-1}\beta_{n-2} & s_{n-1}\beta_{n-2} \\ \dots & c_{n-1}\beta_{n-2} & \gamma_{n-1} & s_{n-1}\pi_{n-1} \\ \dots & s_{n-1}\beta_{n-2} & s_{n-1}\pi_{n-1} & \hat{\alpha}_n \end{bmatrix}, \quad (8.45)$$

where $\gamma_{n-1} = c_{n-1}\pi_{n-1}$ and $\hat{\alpha}_n$ can be expressed in terms of c_{n-1} and s_{n-1} . However because the trace is preserved we have $\hat{\alpha}_n + \gamma_{n-1} = \alpha_{n-1} + \alpha_n$.

The tridiagonal form has been spoilt in positions $(n, n-2)$ and $(n-2, n)$. This is the bulge. All the remaining rotations are devoted to restoring the matrix in (8.45) to tridiagonal form by the method of Givens described in section 7.5. The quantity $s_{n-1}\beta_{n-2}$ is annihilated by a rotation $R_{n-2} \equiv R(n-2, n-1, \dot{\theta}_{n-2})$ with $\tan \dot{\theta}_{n-2} = s_{n-1}\beta_{n-2}/s_{n-1}\pi_{n-1} = \beta_{n-2}/\pi_{n-1}$. The similarity yields

$$\dot{R}_{n-2} R_{n-1} T R_{n-1}^* \dot{R}_{n-2}^* \quad (8.46)$$

$$= \begin{bmatrix} & & & 0 & & 0 \\ & \ddots & & 0 & & 0 \\ & & \alpha_{n-3} & c_{n-2}\beta_{n-3} & s_{n-2}\beta_{n-3} & 0 \\ & & c_{n-2}\beta_{n-3} & \gamma_{n-2} & s_{n-2}\pi_{n-2} & 0 \\ & & s_{n-2}\beta_{n-3} & s_{n-2}\pi_{n-2} & \dot{\alpha}_{n-1} & \dot{\beta}_{n-1} \\ & \ddots & 0 & 0 & \dot{\beta}_{n-1} & \dot{\alpha}_n \end{bmatrix} \quad (8.47)$$

where $\gamma_{n-2} = c_{n-2}\pi_{n-2}$ and $\dot{\alpha}_{n-1} + \gamma_{n-2} = \alpha_{n-2} + \gamma_{n-1}$.

The bulge has been chased from $(n, n-2)$ to $(n-1, n-3)$ and (8.47) reveals the essential pattern. The plane rotations are continued until the bulge goes to $(2, 0)$ and vanishes. The dot on \dot{R}_j is to emphasize that, in principle, there is no reason why \dot{R}_j should equal the R_j of section 8.12. The final matrix is \hat{T} , and it is not clear at this point how the transformation $T \rightarrow \hat{T}$ is related to QL. (See section 8.15 for more details concerning the transformation $T \rightarrow \hat{T}$.)

By the reduction uniqueness theorem (section 7.2) \hat{T} is completely determined by T and the first (or last) column of $\dot{Q} = R_{n-1}^* \dot{R}_{n-2}^* \cdots \dot{R}_1^*$. Note that there is no dot over R_{n-1} . By the special form of plane rotations

$$\begin{aligned} \dot{Q}e_n &= R_{n-1}^* \dot{R}_{n-2}^* \cdots \dot{R}_1^* e_n = R_{n-1}^* \dot{R}_{n-2}^* \cdots \dot{R}_2^* e_n \\ &= \cdots \\ &= R_{n-1}^* e_n. \end{aligned} \quad (8.48)$$

Similarly for the transformation matrix of the previous section, and so

$$Qe_n = R_{n-1}^* R_{n-2}^* \cdots R_1^* e_n = R_{n-1}^* e_n = \dot{Q}e_n. \quad (8.49)$$

By (8.49) and Theorem 7.2.1 if either \hat{T} or \dot{T} is unreduced then they must be identical (up to signs of the off-diagonal elements). All that is necessary in the new formulation is that the first rotation angle θ_{n-1} be determined by the QL factorization of $T - \sigma$.

By forming \dot{T} , instead of $\hat{T} - \sigma$, the transformation has been effected *implicitly*.

To complete the theory behind this indirect formulation of QL one more result is needed.

Lemma 8.13.1. Let $\hat{T} = Q^*TQ$ where Q comes from the QL factorization of $T - \sigma$. If T is unreduced and σ is not an eigenvalue then \hat{T} is unreduced too.

Proof. Let $\hat{\pi}_j \equiv \hat{\beta}_j \cdots \hat{\beta}_{n-1}$, $j = n-1, \dots, 1$. We must show that $\hat{\pi}_j \neq 0$ for $j \geq 1$. An easy variation on the results of section 7.3 yields

$$|\hat{\pi}_{n-j}| = \|\tilde{\chi}_j(\hat{T})\mathbf{e}_n\|, \quad 1 \leq j \leq n,$$

where $\tilde{\chi}_j$ are monic polynomials of degree j . So

$$\begin{aligned} |\hat{\pi}_{n-j}| &= \|Q^*\tilde{\chi}_j(T)Q\mathbf{e}_n\|, \quad \text{since } \hat{T} = Q^*TQ, \\ &= \|\tilde{\chi}_j(T)\mathbf{q}_n\|, \quad \text{by orthogonal invariance of } \|\cdot\|, \\ &= \|\tilde{\chi}_j(T)(T - \sigma)\mathbf{e}_n\|/l_{nn}, \quad \text{by the QL factorization,} \end{aligned}$$

and $l_{nn} > 0$, since σ is not an eigenvalue. Moreover polynomials in a matrix commute; so if $\hat{\pi}_{n-j} = 0$, then

$$(T - \sigma)\tilde{\chi}_j(T)\mathbf{e}_n = \mathbf{o},$$

and, since $T - \sigma$ is invertible,

$$\tilde{\chi}_j(T)\mathbf{e}_n = \mathbf{o}.$$

The $(n-j)$ th element of the vector on the left is $\pi_{n-j} \equiv \beta_{n-j} \cdots \beta_{n-1}$. In other words $\hat{\pi}_{n-j} = 0$ if and only if $\pi_{n-j} = 0$. \square

8.13.1. Comparison of Explicit and Implicit Shifts

The only defect of the explicit shift is that if a big σ is subtracted from a small α_j then information in α_j is irretrievably lost.

More details and a complete algorithm are given in Contribution II/4 in the Handbook and in EISPACK. The operation count is essentially the same as for the explicit shift, $9n$ mults, $2n$ divisions, and $(n-1)$ square roots. Usually the subtraction of σ does *no more* damage to the eigenvalues than some of the preceding arithmetic operations. In general small eigenvalues are not determined to as high relative accuracy as are the large ones. However,

for some “graded” or nearly diagonal matrices—as shown below—the small eigenvalues can be determined to high relative accuracy, provided that an appropriate algorithm is used. The implicit shift delivers this bonus always. The explicit shift also yields such accuracy when that the eigenvalues are found in order of increasing absolute value.

One blemish of the implicit code as described above is that, in finite precision, the effect of a small shift can be obliterated by large entries at the bottom of T before its effect is felt at the top. In [Stewart, 1970] an ingenious modification of the implicit algorithm is presented that uses alternative formulas for the rotation: one when $\theta < \pi/4$, another when $\theta \geq \pi/4$. The arithmetic effort increases by about 30%.

Example 8.13.1. A “graded” matrix

$$T = \begin{bmatrix} 1 & 1 \\ 1 & 10^3 & 10^2 \\ & 10^2 & 10^6 & 10^5 \\ & & 10^5 & 10^9 & 10^8 \\ & & & 10^8 & 10^{12} \end{bmatrix}.$$

The small eigenvalues can be determined to high *relative* accuracy.

This section must end with a warning. The theorems invoked to equate the implicit shift algorithm and QL are based on exact arithmetic and the unreduced property of T . In practice it is appropriate to test whether T is unreduced to *working precision*. This is not an easy decision. See section 7.11 for a discussion of various tests.

8.14. Shifts for all Seasons

Experience shows that all the eigenvalues of T can be computed with an average of approximately 1.7 QL transforms per eigenvalue when Wilkinson’s shift is used. A rule of thumb (section 8.15) says that $9n^2$ multiplications suffice to find all the eigenvalues and consequently it is difficult for any modification to make a *significant* improvement in this computation. Recall that reduction of a full A to T already requires $(2/3)n^3$ multiplications.

The situation changes markedly when P , the product of all the Q_k , is to be accumulated during the algorithm. This way of computing P ensures that the computed eigenvectors are orthogonal to working accuracy always, even when some eigenvalues are very close to each other. The computation of P_k ($= P_{k-1}Q_k$) requires $4n(n-1)$ multiplications and raises the cost of one QL

iteration from $24n$ multiplications (section 8.13) to $4n^2 + 21n$ multiplications. The spectral decomposition of T by this method is an $O(n^3)$ process, and there is some incentive to reduce the number of QL transformations.

We will now give brief comments on some strategies which have been considered.

8.14.1. No Shifts

Eigenvalues will be found in monotonic order by absolute value—in exact arithmetic. It is too slow for general use.

8.14.2. Rayleigh Quotient Shift

This is very simple to program and there is very fast convergence but troublesome cases could arise.

8.14.3. Newton's Shift

This is designed to produce the eigenvalues of tridiagonals in monotone order without losing second-order convergence. T must be definite (positive or negative) initially; then $T + \chi_T(0)/\chi'_T(0)$ is still definite. Bauer found a clever way to implement QL so that $-\chi_T(0)/\chi'_T(0)$ is produced as a by-product and can be used as a shift at the next iteration. Details are given in Contribution II/6 in the Handbook and in EISPACK program RATQR. Difficulties with underflow have been reported.

Convergence can be rather slow in difficult cases, and the caution built into the Newton shift is only needed for the stability of the implementation. It is quicker to risk finding a few more eigenvalues than are wanted, using Wilkinson's or some more powerful shift, and then check which eigenvalues have been found by slicing the spectrum as described in Chapter 3.

8.14.4. Saad's Shifts

Since each update of P costs $4n^2$ ops it is not unreasonable to spend $O(n)$ ops to find a better shift than Wilkinson's. [Saad, 1974] makes some specific suggestions. It is possible to evaluate $\chi(\omega)$ and $\chi'(\omega)$ together in $4n$ ops without modifying T and then compute one QL transform with shift $\omega - \chi(\omega)/\chi'(\omega)$ and update P . Of course there is no need to restrict the shift to the first Newton iterate of ω . In fact Saad proposes using Newton's method to compute an eigenvalue to the desired accuracy and only then to perform a QL transform using the eigenvalue as shift. Such techniques bring the total number

of QL iterations close to n and permit the computation of $P = \prod_{k=1}^n Q_k$ in approximately $\sum_1^n 4k(k - 1) = 4(n^3 - n)/3$ ops, as 40% reduction in cost.

8.14.5. The Ultimate Shifts

If two n -vectors of extra storage are available then the root-free QL algorithm of section 8.15 can be applied to a copy of T to compute all the eigenvalues in perhaps $9n^2 + 42n$ ops. The QL transform is then applied to T by using the eigenvalues in monotonic order as shifts and accumulating the product of the Q 's.

However, with finite precision arithmetic, more than one iteration with a given eigenvalue may be needed before deflation occurs.

The advantage of using QL instead of Newton's method to find the eigenvalues is not so much in the operation count as in the guaranteed convergence of QL with Wilkinson's shift. More care is needed with Newton's method to make it converge in all situations.

*8.15. Casting Out Square Roots

The implicit QL transformation of T into \hat{T} requires $9n$ multiplications, $2n$ divisions, and $(n - 1)$ square roots. Usually $(1.7)n$ transformations suffice to compute all the eigenvalues, and the algorithm is rightly regarded as both efficient and stable. It lies at the heart of current eigenvalue programs. Nevertheless it can be quickened when eigenvectors are not wanted.

In 1963 Ortega and Kaiser observed that the QR transformation can be reorganized to eliminate all the square roots. Unfortunately their program occasionally gives inaccurate results. This can happen because the new version no longer literally carries out orthogonal similarity transformations. That paper led to a fascinating sequence of investigations devoted to casting square roots out of the QR algorithm (Rutishauser, Wilkinson, Welsch, Stewart, Glauz, Pereyra, and Sack, to name a few). Each contributor seemed to cure a defect in his predecessor's program only to introduce a subtle blemish of his own. The sequence appeared to terminate with [Reinsch, 1971] who gave a streamlined, stable algorithm called TQLRAT. However, Reinsch's code has a feature which occasionally prevents the calculation of small eigenvalues to their maximum relative accuracy. The quest for an optimal implementation was not over. See Notes and References for recent work.

This section presents an unpublished algorithm developed by Pal and Walker (in 1968–1969!) as a project in a graduate course taught by Kahan at the University of Toronto. We call it the PWK algorithm. It avoids tolerances without

sacrificing either stability or elegance.

Using either the PWK or TQLRAT algorithm it is usually possible to compute the p smallest or p largest eigenvalues of T with approximately $20pn$ multiplications.

The best shift strategies yield the eigenvalues in a loosely monotonic order. After p eigenvalues have been computed it is easy to slice the spectrum (section 3.3) and see whether or not any wanted eigenvalues have been missed. If so the algorithm proceeds until all p have been found. A simple estimate is two iterations per eigenvalue. This is too low if $p = 1$ and too high if $p \geq n/3$.

All eigenvalues of T can be computed in approximately $9n^2$ ops.

Justification: Assume an average of 1.8 transforms for each eigenvalue, the order dropping by one each time.

8.15.1. Derivation of the Algorithm

Our object is to exhibit the QL transform of section 8.13 in such a way that it is clear how to avoid the square root and yet preserve accuracy in all circumstances. We start with $T = T_n$ and gradually build up $\hat{T} = T_1$ by means of a sequence of plane rotations $T_i = R_i T_{i+1} R_i^*$, $i = n-1, \dots, 1$, where $R_i = R(i, i+1, \theta_i)$ and T_i bulges in positions $(i \pm 1, i \mp 1)$.

An important ingredient in this undertaking is to see the structure of the active elements of T_{i+1} . This pattern emerged in section 8.13 and it is shown, in general position, in Figure 8.4. It is helpful to ignore the shift σ now, and it is easy to put it in place at the end.

In order to move the bulge in T_{i+1} the variables c_i and s_i , which determine R_i , must satisfy

$$c_i(s_{i+1}\beta_i) - s_i(s_{i+1}\pi_{i+1}) = 0. \quad (8.50)$$

This dictates

$$\zeta_i = \sqrt{\pi_{i+1}^2 + \beta_i^2}, \quad c_i = \pi_{i+1}/\zeta_i, \quad s_i = \beta_i/\zeta_i. \quad (8.51)$$

Now

$$\hat{\beta}_{i+1} = c_i(s_{i+1}\pi_{i+1}) + s_i(s_{i+1}\beta_i) = s_{i+1}\zeta_i. \quad (8.52)$$

$$\mathbf{T}_{i+1} = \begin{bmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1 & \ddots & & & & \\ \vdots & \alpha_{i-1} & \beta_{i-1} & & & \\ & \beta_{i-1} & \alpha_i & c_{i+1}\beta_i & s_{i+1}\beta_i & \\ & & c_{i+1}\beta_i & c_{i+1}\pi_{i+1} & s_{i+1}\pi_{i+1} & \\ & & s_{i+1}\beta_i & s_{i+1}\pi_{i+1} & \hat{\alpha}_{i+2} & \hat{\beta}_{i+2} \\ & & & & & \ddots \\ & & & & & \hat{\alpha}_n \end{bmatrix}$$

Transformation of the (i, i) element:

$$\alpha_i \rightarrow \gamma_i \rightarrow \hat{\alpha}_i,$$

$$\mathbf{T}_{i+1} \rightarrow \mathbf{T}_i \rightarrow \mathbf{T}_{i-1},$$

$$\gamma_i = c_i \pi_i \text{ from section 8.13.}$$

FIG. 8.4. The QL transform showing the bulge.

The interesting question is how to express the new values of elements (i, i) , $(i, i + 1)$, and $(i + 1, i + 1)$. It helps to lay out the active rows of $\mathbf{R}_i \mathbf{T}_{i+1}$,

$$\begin{bmatrix} \cdot & \alpha_{i-1} & \beta_{i-1} & 0 & 0 & \cdot \\ \cdot & c_i \beta_{i-1} & \pi_i & 0 & 0 & \cdot \\ \cdot & s_i \beta_{i-1} & \kappa_i & c_{i+1} \zeta_i & s_{i+1} \zeta_i & \cdot \\ \cdot & 0 & s_{i+1} \beta_i & s_{i+1} \pi_{i+1} & \hat{\alpha}_{i+1} & \cdot \end{bmatrix}, \quad (8.53)$$

where

$$\pi_i = c_i \alpha_i - s_i (c_{i+1} \beta_i). \quad (8.54)$$

When R_i^* is applied on the right the (i, i) element becomes

$$\begin{aligned}\gamma_i &= c_i \pi_i - s_i 0 \\ &= c_i^2 \alpha_i - c_i s_i c_{i+1} \beta_i, \text{ by (8.54),} \\ &= c_i^2 \alpha_i - s_i^2 c_{i+1} \pi_{i+1}, \text{ by (8.50),} \\ &= c_i^2 \alpha_i - s_i^2 \gamma_{i+1}. \end{aligned}\quad (8.55)$$

Considerable efforts have been made to get rid of either c_i^2 or s_i^2 , but PWK prefer to keep them both for the sake of stability. The constant trace condition gives a nice way to update element $(i+1, i+1)$, namely,

$$\hat{\alpha}_{i+1} = \gamma_{i+1} + \alpha_i - \gamma_i. \quad (8.56)$$

Having updated γ by (8.55) we are free to update π by

$$\pi_i = \gamma_i / c_i. \quad (8.57)$$

This will not do when $c_i = 0$ and it looks as though (8.57) might be unreliable when c_i is tiny. The error analysis shows that the redundancy in (8.55) keeps (8.57) stable in the presence of tiny values of c_i . It is left as Exercise 8.15.1 to show that the case $c_i = 0$ is a harmless interchange: $s_i = 1$, $\hat{\alpha} = \alpha_i$, but $\pi_i = -\beta_i c_{i+1}$. Thus (8.57) becomes

$$\pi_i = \begin{cases} \gamma_i / c_i, & c_i \neq 0, \\ -\beta_i c_{i+1}, & c_i = 0. \end{cases} \quad (8.58)$$

Formulas (8.51), (8.52), (8.55), (8.56), and (8.58) yield a slightly unorthodox implementation of the inner loop of the standard QL algorithm. To get rid of the square root in (8.51) it is only necessary to square formulas (8.51), (8.52), and (8.58). No essential information has been lost since (8.55) uses c_i^2 and s_i^2 . The inner loop is laid out at the end of the section.

The nice feature is that (8.58) embodies a simple zero test to switch between two formulas. The shift appears in (8.55) only since (8.56) is invariant. Before examining stability we look at some other versions.

8.15.2. Ortega–Kaiser Version

A multiplication is saved by writing

$$\gamma_i = \alpha_i - s_i^2(\alpha_i + \gamma_{i+1}) \quad (8.59)$$

instead of $\gamma_i = c_i^2 \alpha_i - s_i^2 \gamma_{i+1}$. When $s_i^2 \doteq 1$ and $\gamma_{i+1} \ll \alpha_i$ then γ_i should be very close to $-\gamma_{i+1}$ but the computed value will have a very large relative error. Of itself this is not serious. The damage is done in the next step (8.58) wherein $\pi_i^2 = \gamma_i^2 / c_i^2$. When $s_i^2 \doteq 1$ the error in π_i^2 will be large relative to $\|\mathbf{T} - \sigma\|$.

8.15.3. Reinsch's Algorithm

Reinsch also wished to avoid the redundancy of using both c_i^2 and s_i^2 and so used (8.59) to update γ . This necessitated rewriting (8.58) in a clever way, as follows:

$$\pi_i^2 = \gamma_i^2/c_i^2 = \gamma_i \cdot \eta_i, \quad (8.60)$$

where

$$\begin{aligned} \eta_i &\equiv \gamma_i/c_i^2 \\ &= \alpha_i - s_i^2 \gamma_{i+1}/c_i^2, \text{ by (8.55),} \\ &= \alpha_i - \gamma_{i+1} \beta_i^2/\pi_{i+1}^2, \text{ by (8.50),} \\ &= \alpha_i - \beta_i^2 c_{i+1}^2/\gamma_{i+1}, \text{ by (8.58),} \\ &= \alpha_i - \beta_i^2/\eta_{i+1}. \end{aligned} \quad (8.61)$$

In fact η_i is the i th diagonal element in the triangular factorization of T from the bottom upward. To modify (8.61) for $\eta_{i+1} = 0$ Reinsch replaces 0 by $\epsilon \max_k |\alpha_k - \sigma|$. This can be justified in terms of backward error analysis as being equivalent to perturbing α_i by no more than $\epsilon \max |\alpha_k - \sigma|$ which is a small perturbation relative to $\|T - \sigma\|$. Nevertheless it is nice to have an algorithm in which such a rigid replacement is not needed.

8.15.4. Slicing the Spectrum

Reinsch observed that ν , the number of negative η_i , is equal to the number of eigenvalues of T that are less than the shift σ . The reasons are given in section 3.3. Since $\gamma_i = \eta_i c_i^2$ the same information can be obtained from the PWK scheme. When σ has converged to working accuracy then $\nu + 1$ will give its index. This is valuable information when only a few eigenvalues of T are wanted.

On the other hand it is slightly quicker to test the index outside the hand-crafted inner loop of the QL scheme.

8.15.5. Stability

There is a place for a formal error analysis of the PWK algorithm but not in this book. See the reference at the end of the Notes and References. The treatment of most of the formulas is straightforward. We will focus on the way γ_i is formed in (8.55). We show that, in practice, the division in (8.57) is not to be feared.

Let overbars denote computed quantities and assume that

$$\bar{f}\ell(\xi + \eta) = \xi(1 + \epsilon) + \eta(1 + \epsilon),$$

$$\bar{f}\ell(\xi\eta) = \xi\eta(1 + \epsilon). \quad (8.62)$$

The quantity ϵ represents a possibly different, tiny value at each appearance.

Despite errors, the algorithm maintains the following relationships:

$$\bar{\pi}_{i+1}\bar{s}_i = \beta_i\bar{c}_i(1 + \epsilon), \text{ by (8.50)}, \quad (8.63)$$

$$\bar{\gamma}_i = \bar{c}_i\bar{\pi}_i(1 + \epsilon), \text{ by (8.58)}. \quad (8.64)$$

Then in (8.55),

$$\begin{aligned} \bar{\gamma}_i &= \bar{f}\ell[(\alpha_i - \sigma)\bar{c}_i^2 - \bar{\gamma}_{i+1}\bar{s}_i^2] \\ &= [\alpha_i(1 + \epsilon) - \sigma(1 + \epsilon)]\bar{c}_i^2(1 + \epsilon)^2 - \bar{\gamma}_{i+1}\bar{s}_i^2(1 + \epsilon)^2, \text{ by (8.62)}, \\ &= \gamma_i + 3\epsilon\alpha_i\bar{c}_i^2 + 3\epsilon\sigma\bar{c}_i^2 + 2\epsilon\bar{\gamma}_{i+1}\bar{s}_i^2, \end{aligned} \quad (8.65)$$

neglecting all terms in ϵ^2 . The last term in (8.65) appears to invite disaster when $\bar{\gamma}_i$ is divided by a small \bar{c}_i . This is not the case because

$$\begin{aligned} \bar{\gamma}_{i+1}\bar{s}_i^2 &= \bar{c}_{i+1}\bar{\pi}_{i+1}(1 + \epsilon)\bar{s}_i^2, \text{ by (8.64)}, \\ &= (1 + \epsilon)\bar{c}_{i+1}\bar{s}_i\beta_i\bar{c}_i(1 + \epsilon), \text{ by (8.63)}. \end{aligned} \quad (8.66)$$

It follows that, after (8.57) is executed,

$$|\bar{\pi}_i - \bar{\pi}| \leq \epsilon\{3(|\alpha_i| + |\sigma|)\bar{c}_i + 2|\bar{c}_{i+1}\beta_i| \cdot |\bar{s}_i| + |\bar{\pi}_i|\} \quad (8.67)$$

and so the error is always tiny relative to neighboring elements in the matrix.

In Table 8.1 we display the inner loop of the algorithm. It operates on a matrix in rows $k, k+1, \dots, m$, and capital letters denote squared quantities, $B_i = \beta_i^2$, $i = k, k+1, \dots, m-1$. Note that the code sets to zero one entry that lies outside the matrix, B_{k-1} for QR, B_m for QL.

Exercise on Section 8.15

- 8.15.1. Examine the case when $c_{i+1} \neq 0$ but $c_i = 0$ and show how the interchange leads to $\pi_i = -\beta_i c_{i+1}$.

TABLE 8.1
PWK inner loops.

	QR	QL
Initialize	$C \leftarrow 1, S \leftarrow 0,$ $\gamma \leftarrow \alpha_k - \sigma, P \leftarrow \gamma \cdot \gamma$	$C \leftarrow 1, S \leftarrow 0,$ $\gamma \leftarrow \alpha_m - \sigma, P \leftarrow \gamma \cdot \gamma$
Loop	$i \leftarrow k, k+1, \dots, m-1;$ $BB \leftarrow B_i$ $R \leftarrow P + BB$ $B_{i-1} \leftarrow S \cdot R$ $OLDC \leftarrow C$ $C \leftarrow P/R$ $S \leftarrow BB/R$ $old\gamma \leftarrow \gamma$ $\alpha \leftarrow \alpha_{i+1}$ $\gamma \leftarrow C \cdot (\alpha - \sigma) - S \cdot old\gamma$ $\alpha_i \leftarrow old\gamma + (\alpha - \gamma)$ $P \leftarrow \begin{cases} (\gamma \cdot \gamma)/C, & C \neq 0, \\ OLDC \cdot BB, & C = 0. \end{cases}$	$i \leftarrow m-1, \dots, k+1, k;$ $BB \leftarrow B_i$ $R \leftarrow P + BB$ $B_{i+1} \leftarrow S \cdot R$ $OLDC \leftarrow C$ $C \leftarrow P/R$ $S \leftarrow BB/R$ $old\gamma \leftarrow \gamma$ $\alpha \leftarrow \alpha_i$ $\gamma \leftarrow C \cdot (\alpha - \sigma) - S \cdot old\gamma$ $\alpha_{i+1} \leftarrow old\gamma + (\alpha - \gamma)$ $P \leftarrow \begin{cases} (\gamma \cdot \gamma)/C, & C \neq 0, \\ OLDC \cdot BB, & C = 0. \end{cases}$
Termination	$B_{m-1} \leftarrow S \cdot P$ $\alpha_m \leftarrow \sigma + \gamma$	$B_k \leftarrow S \cdot P$ $\alpha_k \leftarrow \sigma + \gamma$
Cost	3 divisions, 4 multiplications, 5 adds per iteration	
Storage	$BB, R, P, S, C, OLDC, \gamma, old\gamma$	
Scaling	The matrix should be scaled up, by a power of the radix of the arithmetic unit, before starting in order to avoid unnecessary underflows in the QL transformations.	

8.16. QL for Banded Matrices

If four eigenvalues of a matrix A of order 400 and bandwidth 11 are wanted, then it is somewhat inefficient to reduce A to tridiagonal form as described in section 7.5. Because bandwidth is preserved the shifted QL algorithm can be implemented with economy in both storage requirements and operation count. The algorithm is presented as Contribution II/7 in the Handbook. It is a clever piece of mathematical software that is far from a blind realization of the transformation defined in section 8.2. We will not go into those details here, but we will indicate the main ideas.

With a given shift σ the matrix $\bar{A} = A - \sigma I$ is, in principle, premultiplied by a sequence of reflectors H_n, H_{n-1}, \dots , to reduce it, column by column, to lower-triangular form L . Reflectors are elementary orthogonal matrices described in section 6.3. Thus

$$H_2 \cdots H_n \bar{A} = L, \quad (8.68)$$

where $H_n = H(w_n)$ is chosen to put the n th column into triangular form, H_{n-1} puts the $(n-1)$ st column of $H_n \bar{A}$ into triangular form, and so on. The proper choice of w_i is straightforward and is given in section 6.3.

The second phase of the transformation requires the postmultiplication of L by all the H_i and a difficulty becomes apparent. The vectors w_i defining the H_i must be saved and this requires $n(m+1)$ storage locations. Here $(m+1)$ is the half-bandwidth. The problem goes away when it is observed that there is no need to find L before beginning the second phase. Inspection of Figure 8.5 shows that after m of the H_i have been applied on the left, the second phase can be started. This suggests that one whole QL transformation can be accomplished by $m+n$ minor steps of the following form:

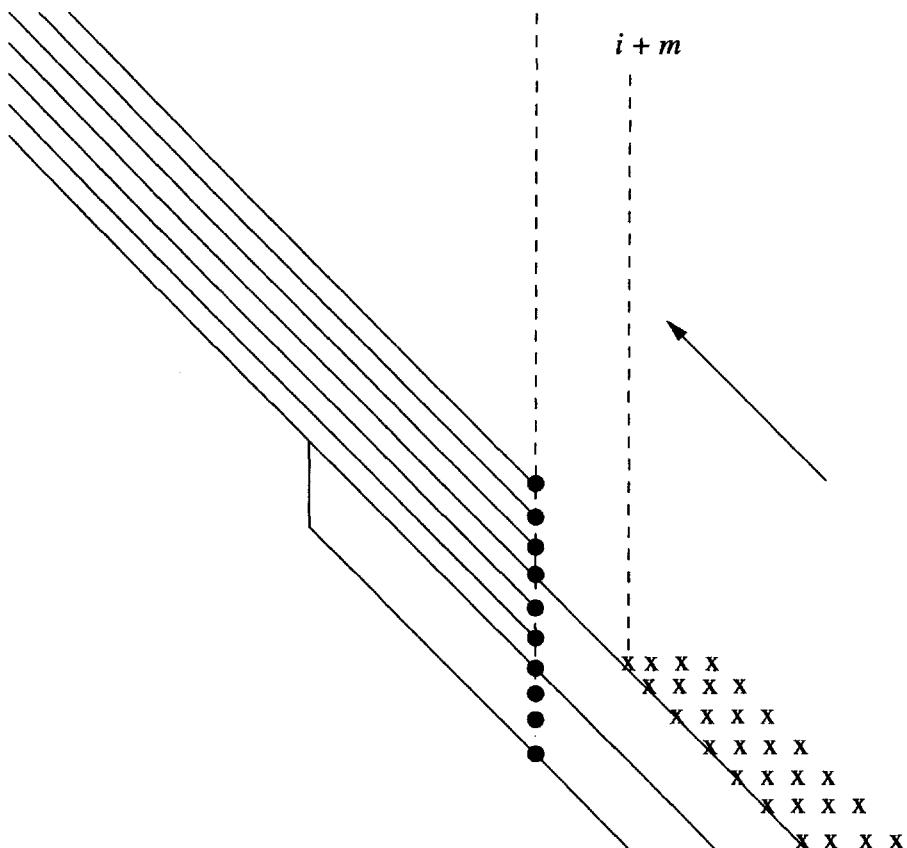
$$H_i \bar{A} H_{i+m}, \quad i = n, n-1, \dots, -m, \quad (8.69)$$

where \bar{A} is the current matrix and $H_j = I$ for $j \leq q$ and $j > n$. Further inspection of Figure 8.5 shows that w_{i+2m} can be discarded at this step.

In fact the algorithm does not implement (8.69) directly. Instead $(A - \sigma e_i)$ is taken straight into the corresponding column of L via

$$H_i \cdots H_{i+2m} (a_i - \sigma e_i),$$

w_i is stored away, other columns of A are *not* touched, and the $(i+m)$ th row of the final matrix \hat{A} is computed and stored. Full advantage is taken of symmetry, and storage is handled very nicely.



Key: — Elements of A (untouched)

● ● ● Active column

X X X Elements of \hat{A}

FIG. 8.5. *The banded algorithm.*

8.16.1. Origin Shift

The algorithm will be used only when a small number of eigenvalues are needed, and so the order in which eigenvalues are found is important. The Handbook uses a two-stage strategy: the origin shift σ is 0 initially; then after the first

off-diagonal row passes a test Wilkinson's tridiagonal shift is used. This is the only inelegant feature of this program.

At the cost of one triangular factorization ($m^2n/2$ ops) one can find the exact position of any computed eigenvalue in the spectrum. This makes it less important to shift with great caution.

8.16.2. Operation Count

$3(m + 1)(2m + 1)$ ops per minor step.

$(m + n)$ minor steps per iteration.

5 iterations (average) for the first eigenvalue.

1 iteration for each later eigenvalue.

8.16.3. Storage

$n \times (m + 1)$ for \mathbf{A} (by diagonals),

$(m + 1) \times (2m + 1)$ for the \mathbf{w}_i ,

$2 \times (2m + 1)$ for work space.

This method is in EISPACK, release 2, under the name BQR.

Notes and References

The QL, or QR, algorithm is the currently preferred way to compute the eigenvalues of small matrices. Yet it was not invented until 1958–1959 and was not appreciated until the mid-1960s. The key idea came from Rutishauser with his construction of a related algorithm called LR in 1958.

Credit also goes to a young systems programmer in England, J.G.F. Francis, who made the three observations required to make the method successful, to wit (1) the basic QR transformation, (2) the invariance of the Hessenberg form, and (3) the use of origin shifts to hasten convergence. It should be added that Francis received some assistance from Strachey and Wilkinson. The original papers are [Francis, 1961–1962]. Independently, and in the same period, Kublanovskaya discovered (1) and (3), but without the improvements induced by (2) the algorithm is very slow. See [Kublanovskaya, 1961].

The intimate connection between the RQI and QR was noticed by both Kahan and Wilkinson. In fact, the observed monotonic decay of the last off-diagonal element in the tridiagonal QR algorithm led Kahan to the global convergence analysis of RQI presented in Chapter 4.

Wilkinson and Reinsch encouraged us to switch from the QR to the QL formulation by using it in the influential Handbook.

In 1968 Wilkinson proved that the tridiagonal QL algorithm can never fail when his shift is used. His proof is based on the monotonic decline of

the product $|\beta_1\beta_2|$. However, the analysis is simpler when $\beta_1^2\beta_2$ is used in place of $\beta_1\beta_2$, and this approach is presented in sections 8.9 and 8.10 which follow [Hoffman and Parlett, 1978] and uses some insights buried in [Dekker and Traub, 1971]. Some of the ideas which led to the discovery of the QR algorithm are described in [Parlett, 1964].

The PWK version of the root-free QL algorithm (section 8.15) has not appeared in the open literature. Backward stability of PWK was proved in [Feng, 1991]. The algorithm was incorporated in the LAPACK library under the name SSTERF using QR rather than QL format. However, in 1996 Gates and Gragg showed how to get rid of one multiplication in the inner loop while preserving stability. The change is easily described. In PWK (QL shifted version) the operations (8.55) and (8.56) are

$$\begin{aligned}\gamma_i &\leftarrow c_i^2(\alpha_i - \sigma) - s_i^2\gamma_{i+1}, \\ \hat{\alpha}_i &\leftarrow \gamma_{i+1} + \alpha_i - \gamma_i.\end{aligned}$$

Gates and Gragg introduce a new variable u_i and rewrite these operations (in QL format) as

$$\begin{aligned}u_i &\leftarrow \alpha_i + \gamma_{i+1}, \\ \gamma_i &\leftarrow c_i^2(u_i - \sigma) - \gamma_{i+1}, \\ \hat{\alpha}_i &\leftarrow u_i - \gamma_i.\end{aligned}$$

It looks so easy now! Note that u_i is a “temporary” variable; the quantity $\alpha_i + \gamma_{i+1}$ should be kept in a register and never sent back to the memory.

In addition Gates and Gragg show how the sign of c_k may be recovered from c_k^2 (assuming $s_k \geq 0$). Observe that $\text{sign}(c_k) = \text{sign}(\pi_{k+1})$, by definition of c_k , and, in addition $\gamma_k = \pi_k c_k$ so that $\text{sign}(\gamma_k) = \text{sign}(\pi_k)\text{sign}(c_k)$. Hence, sequentially,

$$\text{sign}(\pi_k) = \begin{cases} \text{sign}(\gamma_k)\text{sign}(\pi_{k+1}), & \pi_{k+1}^2 \neq 0, \\ -\text{sign}(\pi_{k+2}) & \text{otherwise.} \end{cases}$$

The rapid and global convergence of QL with Wilkinson’s shift is very attractive. It does not follow that this is the best shift. See [Erxiong and Zhenye, 1985] for a better strategy.

We expect that by the year 2000 A.D. the QL and QR algorithms will have been displaced by a new variant of the qr algorithm introduced by Rutishauser in 1953. See the notes at the end of Chapter 7 for references.

Jacobi Methods

9.1. Rotation in the Plane

In two dimensions a rotation through an angle θ is accomplished by premultiplying vectors by

$$R(\theta) = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}.$$

Throughout this chapter we use c , s , and t for the scalar quantities $\cos \theta$, $\sin \theta$, and $\tan \theta$. The associated similarity transformation on a matrix A is

$$\begin{aligned} RAR^* &= \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} \alpha & \gamma \\ \gamma & \beta \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \\ &= \begin{bmatrix} \alpha c^2 - 2\gamma sc + \beta s^2 & (c^2 - s^2)\gamma + (\alpha - \beta)sc \\ (c^2 - s^2)\gamma + (\alpha - \beta)sc & \alpha s^2 + 2\gamma sc + \beta c^2 \end{bmatrix}. \end{aligned} \quad (9.1)$$

The new matrix will be diagonal if

$$\tan 2\theta = \frac{\sin 2\theta}{\cos 2\theta} = \frac{2sc}{c^2 - s^2} = \frac{2\gamma}{\beta - \alpha}. \quad (9.2)$$

There is no need to find θ explicitly. Let

$$\delta = |\beta - \alpha| / 2, \quad \nu = \sqrt{\gamma^2 + \delta^2}. \quad (9.3)$$

Using standard trigonometric identities (Exercise 9.1.1) we find that

$$c^2 = \frac{1}{2}(1 + \delta/\nu), \quad s^2 = \frac{1}{2}(1 - \delta/\nu). \quad (9.4)$$

The expression for s^2 is a classic example of a treacherous formula for finite precision calculation. Whenever θ is small then δ/ν is close to 1 and the

formula would be fine were it easy to calculate δ/ν to extra precision. With fixed length storage for numbers the remedy is to use a better formula.

There are several stable ways of computing $\cos \theta$ and $\sin \theta$ from the data. Perhaps the nicest is found in Rutishauser's Jacobi program in the Handbook. He uses the trigonometric identities for $\tan 2\theta$ in terms of $t = \tan \theta$,

$$\frac{1 - t^2}{2t} = \cot 2\theta = \frac{\beta - \alpha}{2\gamma} \equiv \zeta. \quad (9.5)$$

Thus t is the smaller root, in magnitude, of

$$t^2 + 2\zeta t - 1 = 0 \quad (9.6)$$

and so

$$t = \text{sign}(\zeta) / (|\zeta| + \sqrt{1 + \zeta^2}). \quad (9.7)$$

Then

$$c = 1/\sqrt{1 + t^2}, \quad s = ct. \quad (9.8)$$

By forcing c to be positive we obtain a rotation with $|\theta| \leq \pi/4$. There is one other solution to $\cot 2\theta = \zeta$ and that is the angle $(\pi/2 + \theta)$. The reason for insisting on the small angle is given at the end of section 9.3 on convergence.

Further use of trigonometry (Exercise 9.1.2) shows that the new matrix is

$$\begin{bmatrix} \alpha - \gamma t & 0 \\ 0 & \beta + \gamma t \end{bmatrix}. \quad (9.9)$$

Exercises on Section 9.1

- 9.1.1. Use the relation between trigonometric functions of θ and 2θ to prove (9.4).
- 9.1.2. Show that the diagonal elements for \mathbf{RAR}^* can be written as $\alpha - \gamma t$ and $\beta + \gamma t$.

9.2. Jacobi Rotations

The traditional formula defining the rotation matrix $\mathbf{R}(\theta)$ which diagonalizes a 2-by-2 matrix \mathbf{A} are given above in (9.5), (9.7), and (9.8). The idea behind Jacobi methods is to apply that technique to the case of an n -by- n matrix \mathbf{A} using the plane rotations $\mathbf{R}(i, j, \theta)$ described in section 6.4. By identifying $\alpha = a_{ii}$, $\beta = a_{jj}$, $\gamma = a_{ij}$, the formula (9.5) given above for c and s implicitly defines

the angle θ such that the (i, j) and (j, i) elements of $\mathbf{A}' = \mathbf{R}(i, j, \theta)\mathbf{A}\mathbf{R}(i, j, \theta)^*$ are zero, namely,

$$\tan 2\theta = 2a_{ij}/(a_{jj} - a_{ii}), \quad i < j.$$

With this choice $\mathbf{R}(i, j, \theta)$ is called a *Jacobi rotation matrix* and the associated similarity transformation is a *Jacobi rotation*. It annihilates the (i, j) element and, since θ is fixed by (9.5), we can write \mathbf{R}_{ij} for $\mathbf{R}(i, j, \theta)$ without ambiguity.

What is new when $n > 2$ is that there are other off-diagonal elements in rows i and j which are transformed. Thus, for $k \neq i, j$,

$$\begin{aligned} a'_{ik} &= c \cdot a_{ik} - s \cdot a_{jk}, \\ a'_{jk} &= s \cdot a_{ik} + c \cdot a_{jk}. \end{aligned} \tag{9.10}$$

Note also that for $k \neq i, j$,

$$(a'_{ik})^2 + (a'_{jk})^2 = a_{ik}^2 + a_{jk}^2. \tag{9.11}$$

Cost. By taking advantage of symmetry and working with either the lower or the upper triangle of \mathbf{A} , a traditional Jacobi rotation can be made with $4n$ multiplications and two square roots (Exercise 9.2.1).

Alternatives. Before leaving this topic it is important to emphasize that there are other possible choices for θ when rotating the (i, j) plane.

In the early days when square root was computed in software rather than hardware various rational approximation to c , s , and t were used to avoid the formula for c in (9.8). However, it is important that $c^2 + s^2 = 1$ to working accuracy.

The Jacobi rotation is distinguished by causing the greatest decrease in the sum of squares of the off-diagonal elements. It is locally optimal but not necessarily best for the total computation. A Givens rotation in the (i, j) plane chooses θ to annihilate some element other than a_{ij} .

Example 9.2.1. *Annihilation of a_{13} by rotations in two different planes*

$$\mathbf{A} = \begin{bmatrix} 3.0 & -12.0 & 10.0 \\ -12.0 & 64.0 & -60.0 \\ 10.0 & -60.0 & 60.0 \end{bmatrix}.$$

$$\text{Trace}(\mathbf{A}) = 127.0, \omega(\mathbf{A})(= \sum_{i < j} a_{ij}^2) = 3844.0.$$

Jacobi

$$\mathbf{A}'_J = \mathbf{R}^*(1, 3, \theta) \mathbf{A} \mathbf{R}(1, 3, \theta) = \begin{bmatrix} 61.70 & -61.16 & 0.00 \\ -61.16 & 64.00 & 1.75 \\ 0.00 & 1.75 & 1.30 \end{bmatrix}.$$

$$\theta = 9.67^\circ, \quad \tan(\theta) = 0.17, \quad \omega(\mathbf{A}'_J) = 3744.0.$$

Givens

$$\mathbf{A}'_G = \mathbf{R}^*(2, 3, \theta) \mathbf{A} \mathbf{R}(2, 3, \theta) = \begin{bmatrix} 3.00 & -15.62 & 0.00 \\ -15.62 & 121.38 & -8.85 \\ 0.00 & -8.85 & 2.62 \end{bmatrix}.$$

$$\theta = -39.81^\circ, \quad \tan(\theta) = -0.83, \quad \omega(\mathbf{A}'_G) = 322.37.$$

The values of ω appear to contradict the assertion above that a Jacobi rotation produces the maximal decrease in ω . The resolution of this quandary constitutes Exercise 9.2.3.

9.2.1. Rutishauser's Modifications

In the application we are about to describe the angles θ will often be small, and there are alternative expressions to (9.10) which cost no more and have better roundoff properties. Define

$$\tau \equiv \tan \theta / 2 = s / (1 + c) \tag{9.12}$$

and then (Exercise 9.2.2) for $k \neq i, j$,

$$\begin{aligned} a'_{ik} &= a_{ik} - s(a_{jk} + \tau \cdot a_{ik}), \\ a'_{jk} &= a_{jk} + s(a_{ik} - \tau \cdot a_{jk}). \end{aligned} \tag{9.13}$$

We close this section by mentioning another effective device introduced by Rutishauser. It is simple but no one thought to use it before 1965. The modifications ($\pm ta_{ij}$) to the diagonal elements are accumulated in a separate array of length n , for a whole sweep through all the off-diagonal elements and only then are the totals of these (usually) small quantities added to the diagonal elements, which are also stored in a separate array. Such devices are the essence of good mathematical software.

Exercises on Section 9.2

- 9.2.1. Do the operation count for one Jacobi rotation using (9.5), (9.7), and (9.8) and modifying only the upper triangular part of \mathbf{A} .
- 9.2.2. Derive (9.12) and (9.13) using standard trigonometric identities.
- 9.2.3. The Givens rotation used to annihilate (1, 3) in Example 9.2.1 causes a far greater reduction in ω than does the Jacobi rotation in the (1, 3) plane. Does this contradict the assertion that Jacobi rotations produce the greatest decrease in ω ?
- 9.2.4. Find some formulas for c , s , and t that approximate the solution to (9.5) but do not use a square root. Your formulas should be very accurate when θ is small.

9.3. Convergence

Jacobi methods seek to diagonalize a given \mathbf{A} by a sequence of Jacobi rotations. Zero elements created at one step will be filled in later and any diagonalizing sequence must be, in principle, infinite. Jacobi methods vary solely in their strategies for choosing the next doomed element.

Before discussing particular strategies we present the notions on which their analysis rests. Recall the Frobenius matrix norm

$$\|\mathbf{A}\|_F = \left(\sum_i \sum_j a_{ij}^2 \right)^{1/2}$$

and Fact 1.10: if \mathbf{Q} is orthogonal then $\|\mathbf{Q}\mathbf{A}\mathbf{Q}^*\|_F = \|\mathbf{A}\|_F$. It is helpful to split $\|\mathbf{A}\|_F^2$ into two parts:

$$\delta(\mathbf{A}) \equiv \sum a_{ii}^2,$$

$$2\omega(\mathbf{A}) = \sum \sum_{i \neq j} a_{ij}^2. \quad (9.14)$$

For *any* plane rotation $\mathbf{R}(p, q, \theta)$ it happens that

$$\delta(\mathbf{R}\mathbf{A}\mathbf{R}^*) = \delta(\mathbf{A}) + 2[a_{pq}^2 - (\text{new } a_{pq})^2]. \quad (9.15)$$

In particular, for a Jacobi rotation \mathbf{R}_{ij} (Exercise 9.3.1),

$$\omega(\mathbf{R}_{ij}\mathbf{A}\mathbf{R}_{ij}^*) = \omega(\mathbf{A}) - a_{ij}^2. \quad (9.16)$$

In whatever order the off-diagonal elements are annihilated the quantity ω is monotonic decreasing and the only question is whether the limit is zero or not. Provided only that, at each step, the pair (i, j) is chosen so that

$$a_{ij}^2 \geq \text{the average of } \{a_{pq}^2 : p < q\} = 2\omega(\mathbf{A})/n(n-1); \quad (9.17)$$

then, using (9.16), we get

$$\omega(R_{ij}\mathbf{A}R_{ij}^*) \leq \left(1 - \frac{2}{n(n-1)}\right) \omega(\mathbf{A}) \quad (9.18)$$

and, by (9.18), ω must converge to zero.

Condition (9.17) ensures convergence to diagonal *form* and that is sufficient for practical purposes. Nevertheless it is legitimate to wonder whether the diagonal elements either wander about or converge to specific eigenvalues in the limit.

Fact 1.11, the Wielandt–Hoffman theorem, shows (Exercise 9.3.4) that there is *some* ordering π of the eigenvalues so that

$$|\lambda_{\pi(i)} - a_{ii}|^2 \leq \sum_{\nu} |\lambda_{\pi(\nu)} - a_{\nu\nu}|^2 \leq 2\omega(\mathbf{A}). \quad (9.19)$$

One fear is that this ordering might change at each step. We may apply (9.19) at a late stage in the Jacobi iteration at which

$$\omega(\mathbf{A}) \leq \frac{1}{4} \min |\lambda_p - \lambda_q|^2 \equiv \sigma^2, \quad (9.20)$$

where the minimum is over *distinct* pairs of eigenvalues. Whenever (9.20) holds there is a diagonal element in the interval $[\lambda - \sigma, \lambda + \sigma]$ for each distinct eigenvalue λ , and the number of a_{ii} in this interval is precisely the multiplicity of λ (Exercise 9.3.2).

Can any a_{ii} escape to another interval at a subsequent step? Yes, indeed, because there are *two* angles θ which satisfy the Jacobi condition

$$\tan 2\theta = 2a_{ij}/(a_{jj} - a_{ii}). \quad (9.21)$$

The larger angle forces both a_{ii} and a_{jj} to leave their intervals and the smaller angle prevents their escape.

Example 9.3.1.

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}.$$

Theorem 9.3.1. *If a Jacobi method satisfies (9.17) at each step and chooses θ in the half open interval $(-\pi/4, \pi/4]$ to satisfy (9.21), then the sequence of matrices converges to a diagonal matrix.*

The proof is left as Exercise 9.3.3.

Exercises on Section 9.3

- 9.3.1. Prove (9.16) using (9.11) and the invariance of $2\omega + \delta$.
- 9.3.2. Use (9.19) and (9.20) to show that each interval $[\lambda - \sigma, \lambda + \sigma]$ contains at least one diagonal element.
- 9.3.3. Prove Theorem 9.3.1.
- 9.3.4. Show that (9.19) is a corollary of the Wielandt–Hoffman theorem for suitable choices of A and M .

9.4. Strategies

9.4.1. Classical Jacobi

At each step this strategy seeks and destroys a maximal off-diagonal element. Convergence follows from (9.18).

On some computers it is relatively expensive to find a maximal off-diagonal element. The naive technique would scan all $n(n-1)/2$ candidates. During the 1960s, considerable effort went into the improvement of this task. In [Corbato, 1963] it is shown that one need only search an array of length n . However, the price for this improvement is twofold: (i) an auxiliary array is needed for the maximal elements in each column, and (ii) the attractive simplicity of the program has been compromised (the number of instructions is more than doubled).

The deathknell of the classical strategy was the observation embodied in (9.17): any element that is above average will suffice for convergence and yield the same guaranteed geometric reduction in ω .

9.4.2. The Cyclic Jacobi Methods

The simplest scheme is to annihilate elements, regardless of size, in the order

$$(1, 2), (1, 3), \dots, (1, n), (2, 3), \dots, (2, n), (3, 4), \dots, (n-1, n)$$

and then begin again for another sweep through the matrix.

In [Henrici, 1958] it was shown that if the angles are suitably restricted the method does indeed converge. The difficulty in the analysis is the possibility that the larger off-diagonal elements are always moving around *ahead* of the current element (i, j) in the sequence. If this were to happen then significant reductions in ω would never occur.

In principle the off-diagonal elements could be annihilated in any order at each sweep, provided that no element was missed. The advent of parallel computers has renewed interest in proving convergence under relaxed strategies; see [Modi, 1988], [Veselić and Hari, 1990], and [Hari, 1991].

The cyclic methods waste time annihilating small elements in the early sweeps.

9.4.3. Threshold Methods

The preferred strategy is to use the simple cyclic pattern but to skip rotations when a_{ij} is less than some threshold value τ which can either be fixed or be varied at each rotation. The fixed threshold is changed, of course, when all off-diagonal elements are below it.

Rutishauser chooses to calculate, for each of the first three sweeps, a fixed τ which is an approximation to ω , namely,

$$\tau = \frac{1}{5} \left(\sum \sum_{p < q} |a_{pq}| / n^2 \right).$$

Perhaps the fastest of the known techniques is the *variable* threshold strategy of Kahan and Corneil [Corneil, 1965]: initially one computes $\omega = \sum \sum_{p < q} a_{pq}^2$ at the cost of $N = \frac{n(n-1)}{2}$ multiplications and then $\tau = \sqrt{\omega/N}$, the true root mean square (RMS) of the off-diagonal elements. At each actual rotation ω is reduced by a_{ij}^2 and τ is recomputed; the cost is one multiplication, one division and one square root *per rotation*. The square root is justified when the number of actual rotations per sweep is well below $N/3$. The alternative is to change the test to

$$N a_{ij}^2 > \omega,$$

which costs two multiplications *per test*, i.e., $2N$ multiplications per sweep regardless of the number of rotations.

9.5. Ultimate Quadratic Convergence

After three or four initial sweeps through all the off-diagonal elements, convergence to diagonal form is usually very rapid. Recall ω and σ from (9.14) and (9.20). If at the end of some sweep $\omega/\sigma^2 \leq 2^{-j}$ we can expect $\omega/\sigma^2 \leq 2^{-2j}$ at the end of next sweep.

Rigorous proofs of this quadratic convergence property [Schönhage, 1961] and [Wilkinson, 1962] are too involved to present here. However, the central idea is simple and goes as follows.

Consider a specific position in the matrix, say the $(1, n)$ element, after several sweeps in cyclic order by rows, namely, $(1, 2), (1, 3), \dots, (n-1, n)$. After being annihilated $(1, n)$ stays zero until the Jacobi rotation involving R_{2n} . Its new value, by (9.10), is

$$\begin{aligned} a'_{1n} &= c \cdot a_{1n} + s \cdot a_{12} \\ &= s \cdot a_{12}, \quad \text{since } a_{1n} = 0. \end{aligned} \quad (9.22)$$

Moreover, since $|\theta| < \pi/4$,

$$|s| = |\sin \theta| \leq |\theta| \leq |\tan \theta| \leq \frac{1}{2} |\tan 2\theta| = |a_{2n}| / |a_{nn} - a_{22}|. \quad (9.23)$$

Combining (9.22) and (9.23) yields the key observation

$$|a'_{1n}| \leq |a_{12}a_{2n}| / |a_{nn} - a_{22}|. \quad (9.24)$$

Now $a_{ii} \rightarrow \lambda_i$ as the sweeps continue. After *some stage*

$$|a_{kk} - a_{ll}| > \sigma \equiv \min_{i \neq j} |\lambda_i - \lambda_j|/2 \quad \text{for all } k \neq l, \quad (9.25)$$

and we suppose for the moment that all the λ_i are simple. By the same argument we see that the next time the $(1, n)$ element is changed its new value is

$$\begin{aligned} |a''_{1n}| &= |ca'_{1n} + sa_{13}| \\ &\leq |a'_{1n}| + |sa_{13}| \\ &\leq (|a_{12}a_{2n}| + |a_{13}a_{3n}|) / \sigma, \\ &\quad \text{since } |s| < |a_{3n}| / \sigma. \end{aligned} \quad (9.26)$$

At the end of some later sweep (9.25) will hold and, in addition, $\max_{i \neq j} |a_{ij}| \leq \eta$ for some η smaller than σ . At the end of the following sweep (9.24) and (9.26) suggest that

$$\max_{i \neq j} |\text{new } a_{ij}| \leq \eta^2(n-1)/\sigma = O(\eta^2) \quad \text{as } \eta \rightarrow 0. \quad (9.27)$$

This behavior is called (*asymptotic*) quadratic convergence.

9.5.1. Multiple Eigenvalues

The foregoing analysis suggests that the onset of rapid convergence to diagonal form will be greatly delayed when eigenvalues are close (i.e., when σ is very small) and may not even be ultimately quadratic when multiple eigenvalues are present. However these fears are groundless as the following observation shows.

There is no loss of generality in supposing that all the a_{ii} 's which converge to a multiple eigenvalue λ are at the bottom of A 's diagonal. Now partition A accordingly as

$$A = \begin{bmatrix} A_1 & B \\ B^* & A_2 \end{bmatrix}, \quad A_1 \text{ is } (n-m) \text{ by } (n-m), \quad A_2 \text{ is } m \text{ by } m. \quad (9.28)$$

Our assumption is that $A_2 \rightarrow \lambda I_m$ where m is the multiplicity of λ and that all other eigenvalues of A are separated from λ by 2δ . After a certain number of sweeps all the eigenvalues of A_1 will be separated from λ by δ or more; in other words,

$$\|(A_1 - \lambda I)^{-1}\| = 1/\min |\lambda_i - \lambda| \leq 1/\delta.$$

The argument for quadratic convergence was based on the assumption that the angles of rotation, $|a_{ij}|/|a_{ii} - a_{jj}|$, become small. This assumption *fails* for the off-diagonal elements of A_2 , where each $a_{kk} \rightarrow \lambda$ but *holds* for B . Fortunately the angles for A_2 do not matter because all the off-diagonal elements are *already tiny*, as the next theorem shows.

Theorem 9.5.1. *Let A be partitioned as in (9.28). If $\|(A_1 - \lambda I)^{-1}\| < 1/\delta$ and if λ is an eigenvalue of multiplicity m , then*

$$\|A_2 - \lambda I\| \leq \|B\|^2/\delta.$$

The proof is an application of block factorization.

Proof.

$$\begin{aligned} A - \lambda I &= \begin{bmatrix} A_1 - \lambda I & B \\ B^* & A_2 - \lambda I \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ B^*(A_1 - \lambda I)^{-1} & I_m \end{bmatrix} \begin{bmatrix} A_1 - \lambda I & 0 \\ 0^* & X_\lambda \end{bmatrix} \begin{bmatrix} I & (A_1 - \lambda I)^{-1}B \\ 0^* & I_m \end{bmatrix}, \end{aligned}$$

where

$$\mathbf{X}_\lambda = (\mathbf{A}_2 - \lambda) - \mathbf{B}^*(\mathbf{A}_1 - \lambda)^{-1}\mathbf{B}.$$

Now observe that the rank of $\mathbf{A} - \lambda$ is $n - m$. By Sylvester's law of inertia the rank of the block diagonal factor must also be $n - m$, and so $\mathbf{X}_\lambda = \mathbf{0}$. Thus

$$\|\mathbf{A}_2 - \lambda\| = \|\mathbf{B}^*(\mathbf{A}_1 - \lambda)^{-1}\mathbf{B}\| \leq \|\mathbf{B}\|^2/\delta. \quad \square$$

A corollary of this theorem is that, for $j > n - m$,

$$|a_{jj} - \lambda| \leq \|\mathbf{B}\|^2/\delta, \quad (9.29)$$

so the diagonal elements converge faster to the multiple eigenvalues than to the well-separated ones.

9.6. Assessment of Jacobi Methods

To reject all plane rotations except Jacobi rotations in a procedure to diagonalize \mathbf{A} is like going into a boxing ring with one hand tied behind your back. What happens if we "generalize" Jacobi methods very slightly to allow the use of Givens rotations to annihilate elements?

We can, first of all, use Givens's original method to reduce \mathbf{A} to tridiagonal form \mathbf{T} (see section 7.4). These rotations may not reduce ω^2 but they do preserve the zeros that have been so carefully created. Next we can do a Jacobi rotation $\mathbf{R}_{n-1,n}$. This fills in elements $(n, n - 2)$ and $(n - 2, n)$ and creates a bulge in the tridiagonal form. This bulge can be chased up the matrix and off the top by a sequence of *Givens* rotations as described in section 8.13. At this point we have executed one tridiagonal QL transformation with zero shift. See Chapter 8 for the QL transform. This sequence can be repeated until the off-diagonal elements are negligible. Lo and behold, Jacobi has turned into Givens reduction to tridiagonal form followed by QL without shifts.

If we relax the habit of zeroing a matrix element at *every* step, we can do a plane rotation in the $(n - 1, n)$ plane through an angle other than the Jacobi angle. This permits us to incorporate shifts into the QL algorithm and thus accelerate convergence dramatically.

The preceding remarks reveal how one method can turn into a rival one by making some apparently minor changes in tactics.

Are there reasons for artificially restricting ourselves to rotating through the Jacobi angle instead of permitting other choices of θ ? First we must say that the best Jacobi programs are only about three times slower than the tridiagonal QL methods, but Jacobi's claim to continued attention has been *simplicity*

rather than efficiency. This may be very important in special circumstances (hand-held calculators or computations in space vehicles).

Let us examine the claim to simplicity. Given below are counts of the executable operations (at the assembly languages level) of various codes in the Handbook. None of the codes tried to minimize these op counts.

All eigenvalues and eigenvectors:

Jacobi 121 versus	$\begin{cases} \text{Tred 2 (tridiagonal reduction)} & 81 \\ \text{Tql 2 (QL algorithm)} & 89 \end{cases}$
-------------------	--

All eigenvalues, no eigenvectors:

Jacobi 99 versus	$\begin{cases} \text{Tred 1} & 58 \\ \text{Tql 1} & 87 \end{cases}$
------------------	---

Fetches and stores were ignored when they occurred as part of arithmetic expressions.

The Jacobi advantage is not great, and it must be said that there is some advantage, when fast storage is tight, to having the computation split into two separate stages. A valuable feature of Jacobi is that it never rotates through larger angles than necessary and, consequently, when the matrix permits, small eigenvalues can be computed to high relative accuracy. See [Demmel and Veselić, 1992] and [Slapričar, 1992].

The arrival of parallel computers renewed interest in Jacobi methods because $n/2$ Jacobi rotations may be performed simultaneously if that many processors are available. See [Modi, 1988].

Notes and References

In [Jacobi, 1846] plane rotations were used to diagonalize a real 7-by-7 symmetric matrix. One hundred years later the method was discovered and described in a report [Bargmann, Montgomery, and von Neumann, 1946] and was eventually turned into a clever and effective program in Rutishauser's Contribution II/1 to the Handbook. The variable threshold strategy is presented in the unpublished report [Corneil, 1965].

The quadratic convergence rate was no secret and formal proofs were given in [Schönhage, 1961] and [Wilkinson, 1962]. A more vexing problem was the possibility that the cyclic and other more convenient strategies might not always lead to convergence to a diagonal matrix.

We mention here an important way to implement a Jacobi algorithm. One-sided Jacobi methods update at each rotation two arrays R and $F = AR$ so that column operations are used exclusively.

Eigenvalue Bounds

Useful information about the eigenvalues of A may be obtained from some of its submatrices. In addition A 's eigenvalues can be related to those of neighboring matrices in a way that goes far beyond standard perturbation theory (the effect of small changes). Such investigations began in the nineteenth century and continue to this day. Much recent work extends the matrix theory to cover differential and integral operators.

The first three sections present the classical material named after Cauchy, Courant, Fischer, and Weyl. The Courant–Fischer minmax theorem turned up in two different lecture courses that I took as a graduate student. On each occasion the lecturer tried to present the proof and became confused. As a result I concluded (erroneously) that the theorem was deep and difficult. Many years later I realized that the only hazard in presenting the proof lies with the indices. Are eigenvalues labelled in increasing or decreasing order? Are subspaces given by bases or by constraints?

In [Ikebe, Inagaki, Miyamoto, 1987] it was shown that each of these classical results depends on the elementary fact that, in a vector space of finite dimension n , any two subspaces whose dimensions sum to more than n must have a nontrivial intersection. That is all; the results are now perfectly natural. I have copied those proofs with slight modifications; reference to n has been minimized by the device of describing some subspaces as “constraints” using “orthogonality,” $v \perp \mathcal{G}^j$, rather than “belonging,” $v \in \mathcal{S}^{n-j}$. The attentive reader will notice a certain duality between the results for eigenvalues taken in increasing order and those taken in decreasing order.

Later sections present refinements which have been developed in response to demands for the computation of eigenvalues of larger and ever larger matrices. All the results are inclusion theorems; that is, they describe intervals which are

guaranteed to contain one or more eigenvalues of \mathbf{A} . The later sections show how to exploit the extra information which is likely to be available at the end of a step in an expensive iterative method of the sort described in Chapters 13 and 14. By computing the appropriate optimal intervals the iteration can be stopped at the right moment for the required accuracy.

Closely related error bounds are given in the next chapter but they only make use of *norms* of certain residuals. Consequently they are less elaborate and less expensive to compute than the intervals of this chapter.

The convention of negative indices for the largest eigenvalues proves useful in this chapter; $\alpha_{-j} = \alpha_{n+1-j}$.

It is essential to know the facts about Rayleigh quotients. In Fact 1.8 the eigenvalues $\alpha_1, \alpha_2, \dots, \alpha_n$ of \mathbf{A} were declared to be the stationary values of the Rayleigh quotient function $\rho(\mathbf{x}) \equiv \mathbf{x}^* \mathbf{A} \mathbf{x} / \mathbf{x}^* \mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$. In particular, Rayleigh's principle states that

$$\min_i \alpha_i \leq \rho(\mathbf{x}) \leq \max_j \alpha_j, \quad (10.1)$$

and this is a direct consequence of the fact that $\rho(\mathbf{x}; \mathbf{A})$ is a weighted average of \mathbf{A} 's eigenvalues. For the same reason, for $\mathcal{Z}^j = \text{span}(\mathbf{z}_1, \dots, \mathbf{z}_j)$, where $\mathbf{A}\mathbf{z}_i = \mathbf{z}_i\alpha_i$, $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$,

$$\alpha_j = \max_{\mathbf{x} \in \mathcal{Z}^j} \rho(\mathbf{x}; \mathbf{A}) = \min_{\mathbf{y} \perp \mathcal{Z}^{j-1}} \rho(\mathbf{y}; \mathbf{A}). \quad (10.2)$$

10.1. Cauchy's Interlace Theorem

This section shows how the eigenvalues of \mathbf{A} relate to those of a principal submatrix. The result can be found, in simple form, in A. Cauchy's "Cours d'Analyse" of 1821. These were the lecture notes used at the Ecole Polytechnique of Paris.

Theorem 10.1.1. *Partition $n \times n$ \mathbf{A} as*

$$\mathbf{A} = \begin{bmatrix} \mathbf{H} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{U} \end{bmatrix}, \quad (10.3)$$

\mathbf{H} is $m \times m$, $m < n$. Label the eigenpairs of \mathbf{A} and \mathbf{H} as

$$\begin{aligned} \mathbf{A}\mathbf{z}_i &= \mathbf{z}_i\alpha_i, \quad i = 1, \dots, n, \quad \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n, \\ \mathbf{H}\mathbf{y}_i &= \mathbf{y}_i\theta_i, \quad i = 1, \dots, n, \quad \theta_1 \leq \theta_2 \leq \dots \leq \theta_m. \end{aligned}$$

Then

$$\alpha_k \leq \theta_k \leq \alpha_{k+(n-m)}, \quad k = 1, \dots, m.$$

Proof. In order to relate Rayleigh quotients for \mathbf{H} to Rayleigh quotients for \mathbf{A} define n -vectors \mathbf{x}_i by

$$\mathbf{x}_i = \begin{pmatrix} \mathbf{y}_i \\ \mathbf{o} \end{pmatrix}, \quad i = 1, \dots, m.$$

Thus

$$\rho(\mathbf{x}_i; \mathbf{A}) = \rho(\mathbf{y}_i; \mathbf{H}) = \theta_i, \quad i = 1, \dots, m. \quad (10.4)$$

For each $k = 1, \dots, m$, define subspaces

$$\begin{aligned} \mathcal{Z}^{k-1} &:= \text{span}(\mathbf{z}_1, \dots, \mathbf{z}_{k-1}), \\ \mathcal{X}^k &:= \text{span}(\mathbf{x}_1, \dots, \mathbf{x}_k). \end{aligned}$$

Since $n < \infty$, dimensional arguments show the existence of $\mathbf{v} \neq \mathbf{o}$ in $(\mathcal{Z}^{k-1})^\perp \cap \mathcal{X}^k$. So

$$\begin{aligned} \alpha_k &= \min_{\mathbf{u} \perp \mathcal{Z}^{k-1}} \rho(\mathbf{u}; \mathbf{A}), && \text{by (10.2),} \\ &\leq \rho(\mathbf{v}; \mathbf{A}), && \text{since } \mathbf{v} \perp \mathcal{Z}^{k-1}, \\ &\leq \max_{\mathbf{w} \in \mathcal{X}^k} \rho(\mathbf{w}; \mathbf{A}), && \text{since } \mathbf{v} \in \mathcal{X}^k, \\ &= \max_{\mathbf{t} \in \mathcal{Y}^k} \rho(\mathbf{t}; \mathbf{H}), && \text{by (10.4),} \\ &= \theta_k && \text{by (10.2).} \end{aligned}$$

The second inequality is more naturally written in our descending notation as

$$-\alpha_{-j} \leq -\theta_{-j}, \quad j = 1, \dots, m$$

and is an instance of the inequality just proved applied to $-A$. \square

Remark 10.1.1. The eigenvalues of A and H do not change when rows and corresponding columns are permuted. Consequently Cauchy's theorem applies to *any* principal submatrix H of order $m \leq n$.

Consideration of the valid indices in Cauchy's interlace theorem suggests that there are m intervals $[\alpha_k, \alpha_{k+(n-m)}]$, $k = 1, \dots, m$, each containing its own θ but only $m - (n - m)$ intervals $[\theta_j, \theta_{j+(n-m)}]$, $j = 1, \dots, m - (n - m)$, each containing its own α . Has some information been lost? No, but some interpretation is needed. In what follows (γ, δ) denotes the open interval $\{\xi : \gamma < \xi < \delta\}$.

Corollary 10.1.1. *With the notation of Theorem 10.1.1*

$$\alpha_{j+(n-m)} \in [\theta_j, \theta_{j+(n-m)}], \quad j = 1, \dots, m - (n - m), \quad (10.5)$$

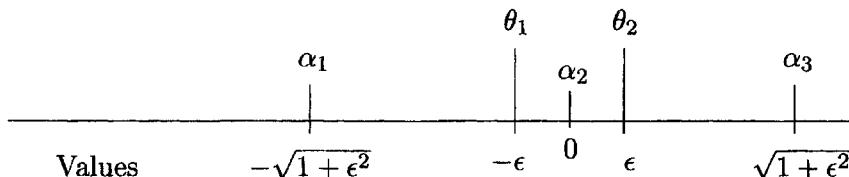
$$\alpha_{m+l} \notin (\theta_l, \theta_{m-(n-m)+l}), \quad l = 1, \dots, (n - m). \quad (10.6)$$

The proof is left as Exercise 10.1.6.

Remark 10.1.2. (10.6) may be written in the form (10.5), with $j = m - (n - m) + 1, \dots, m$, using the following conventions: (i) $\theta_{m+l} = \theta_l$, $l > 0$, and (ii) if $\gamma < \delta$ then $[\delta, \gamma] =$ complement of (γ, δ) . Convention (ii) is equivalent to closing the real line with a single point at infinity.

Example 10.1.1.

$$H = \begin{bmatrix} 0 & \epsilon \\ \epsilon & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & \epsilon & 0 \\ \epsilon & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$



Example 10.1.2. The eigenvalues of the famous tridiagonal matrix T_n with typical row $(\dots -1 \ 2 \ -1 \ \dots)$ are known to be

$$\tau_j^{(n)} = 4 \sin^2(j\phi_n), \quad j = 1, \dots, n, \quad \phi_n = \pi/(2n+2).$$

The interlace bounds yield the following typical results. Here $n = 100$.

$$\tau_{k-(n-m)}^{(m)} \leq \tau_k^{(n)} \leq \tau_k^{(m)} \leq \tau_{k+(n-m)}^{(n)}.$$

Case 1. $m = 99, k = 45, \quad 1.6252 \leq 1.6595 \leq 1.6871 \leq 1.7210.$

Case 2. $m = 90, k = 45, \quad 1.2908 \leq 1.6595 \leq 1.9655 \leq 2.2790.$

As $(n - m)$ increases the bounds become very weak.

Exercises on Section 10.1

- 10.1.1. Consider the case $n - m = 1$ and show, without using the interlace theorem, that $\det(\mathbf{A} - \xi)$ changes sign between distinct eigenvalues θ_j and θ_{j+1} of \mathbf{H} . Assume no b_j vanishes and consider $w(\xi)$ of Exercise 10.1.4.
- 10.1.2. For $n - m = 1$ verify that the inequalities are best possible in the following sense. Given a set of numbers $\{\alpha_i, i = 1, \dots, n\}$ and another set $\{\theta_j, j = 1, \dots, n-1\}$ which interlaces the first, i.e. $\alpha_i < \theta_i < \alpha_{i+1}$, show that there exists a matrix \mathbf{H} with the eigenvalues θ_j and a vector \mathbf{b} and scalar π such that

$$\mathbf{A} = \begin{bmatrix} \mathbf{H} & \mathbf{b} \\ \mathbf{b}^* & \pi \end{bmatrix}$$

has eigenvalues $\{\alpha_i\}$. Note that we have written $\mathbf{B} = \mathbf{b}^*$ here.

- 10.1.3. Extend the result of the previous exercise to the case in which $\alpha_i \leq \theta_i \leq \alpha_{i+1}$; i.e., suppose that $\alpha_j = \theta_j$ for some j and then construct \mathbf{b} and π . (Harder.)
- 10.1.4. For $n - m = 1$ plot $w(\xi) = a_{nn} - \xi I - \mathbf{B}^*(\mathbf{H} - \xi I)^{-1}\mathbf{B}$ as a function of ξ assuming that \mathbf{A} is tridiagonal with no zero subdiagonal elements. Here $\mathbf{B} = \mathbf{e}_m \beta = (0, \dots, \beta)^*$.
- 10.1.5. Find the relation between j at the end of the proof and k in the statement of Theorem 10.1.1.
- 10.1.6. Establish (10.6) from Theorem 10.1.1.

10.2. Minmax and Maxmin Characterization

The blemish in the expression for α_j in (10.2) is that it depends explicitly on z_1, \dots, z_{j-1} . The beautiful result of this section removes this weakness. It is simple, but not trivial, and its origins go back to Poincaré in the late nineteenth century. It was known to E. Fischer [Fischer, 1905] but was made popular by R. Courant [Courant, 1920]. It is often called the Courant–Fischer theorem and is used in proofs of more specialized theorems.

A geometric interpretation is helpful. Consider a 3×3 positive definite matrix \mathbf{A} and the related ellipsoid $\mathcal{A} = \{x : x^* \mathbf{A}^{-1} x = 1\} = \{\mathbf{A}^{1/2} y : \|y\| = 1\}$. The eigenvalues $\alpha_1, \alpha_2, \alpha_3$ are the squares of the lengths of the principal semi-axes of \mathcal{A} . The task is to characterize α_2 without reference to α_1 or α_3 . Each plane through the origin, the center of the ellipsoid, cuts the ellipsoid in an ellipse. For each plane consider the length of the longest ray from \mathbf{o} to the ellipse. The theorem says that the smallest (over all planes) of all these longest rays has length $\sqrt{\alpha_2}$. Conversely $\sqrt{\alpha_2}$ is also the maximum, over all planes on \mathbf{o} , of the shortest ray to the associated ellipse.

We use the symbol \mathcal{S} to recall *subspace* and \mathcal{C} to recall *constraint space*. The presentation is designed to minimize references to the dimension n .

Theorem 10.2.1. *Let \mathcal{S} and \mathcal{C} denote subspaces of \mathcal{E}^n with dimension indicated by the superscript. For $j = 1, \dots, n$,*

$$\alpha_j \equiv \lambda_j[\mathbf{A}] = \min_{\mathcal{S}^j} \max_{\mathbf{u} \in \mathcal{S}^j} \rho(\mathbf{u}; \mathbf{A}) = \max_{\mathcal{C}^{j-1}} \min_{\mathbf{v} \perp \mathcal{C}^{j-1}} \rho(\mathbf{v}; \mathbf{A}).$$

Equivalently,

$$\alpha_{-j} \equiv \lambda_{-j}[\mathbf{A}] = \min_{\mathcal{C}^{j-1}} \max_{\mathbf{u} \perp \mathcal{C}^{j-1}} \rho(\mathbf{u}; \mathbf{A}) = \max_{\mathcal{S}^j} \min_{\mathbf{v} \in \mathcal{S}^j} \rho(\mathbf{v}; \mathbf{A}).$$

Proof. Since n is finite

$$\dim \mathcal{S}^j + \dim (\mathcal{C}^{j-1})^\perp = n + 1 > n$$

and it follows, by the basis theorem, that $\mathcal{S}^j \cap (\mathcal{C}^{j-1})^\perp \neq \mathbf{o}$. Let $\mathbf{w} = \mathbf{w}(\mathcal{S}, \mathcal{C})$ denote any nonzero vector in both subspaces, so

$$\min_{\mathbf{v} \perp \mathcal{C}^{j-1}} \rho(\mathbf{v}) \leq \rho(\mathbf{w}(\mathcal{S}, \mathcal{C})) \leq \max_{\mathbf{u} \in \mathcal{S}^j} \rho(\mathbf{u}).$$

The inequalities hold for *all* choices of \mathcal{S}^j and \mathcal{C}^{j-1} , so try them all to find

$$\max_{\mathcal{C}^{j-1}} \min_{v \perp \mathcal{C}^{j-1}} \rho(v) \leq \min_{\mathcal{S}^j} \max_{u \in \mathcal{S}^j} \rho(u).$$

To show equality use $\mathcal{Z}^j = \text{span}(z_1, \dots, z_j)$ for \mathcal{S}^j and \mathcal{Z}^{j-1} for \mathcal{C}^{j-1} ,

$$\min_{x \perp \mathcal{Z}^{j-1}} \rho(x) \leq \max_{\mathcal{C}^{j-1}} \min_{v \perp \mathcal{C}^{j-1}} \rho(v) \leq \min_{\mathcal{S}^j} \max_{u \in \mathcal{S}^j} \rho(u) \leq \max_{w \in \mathcal{Z}^j} \rho(w),$$

and using (10.2),

$$\alpha_j \leq \max_{\mathcal{C}^{j-1}} \min_{v \perp \mathcal{C}^{j-1}} \rho(v) \leq \min_{\mathcal{S}^j} \max_{u \in \mathcal{S}^j} \rho(u) \leq \alpha_j. \quad \square$$

Exercise on Section 10.2

- 10.2.1. Establish the characterization of α_{-j} without using the result for α_j .
- 10.2.2. Without using the minmax theorem prove that $\lambda_1[A + Y] > \lambda_1[A]$ if Y is positive definite. Give lower and upper bounds for the increase in terms of Y 's eigenvalues.
- 10.2.3. Prove the result in Exercise 10.2.2 by using $-A$ and the minmax theorem.

10.3. The Monotonicity Theorems

A basic fact omitted from Chapter 1 is that $\lambda_i[W]$ is not a linear function of W . So what relations are there between the eigenvalues of W , Y , and $W + Y$? To simplify discussion let $A = W + Y$ and, for $i = 1, \dots, n$,

$$\begin{aligned}\lambda_{\pm i}[A] &= \alpha_{\pm i}, \\ \lambda_{\pm i}[W] &= \omega_{\pm i}, \\ \lambda_{\pm i}[Y] &= \eta_{\pm i}.\end{aligned}$$

Before stating the theorem we look at some special cases. From the minimal property of $\lambda_1[\cdot]$ it follows that

$$\omega_1 + \eta_1 \leq \alpha_1. \tag{10.7}$$

It is also true, but less obviously so,

$$\omega_j + \eta_1 \leq \alpha_j \leq \omega_j + \eta_{-1}, \quad j = 1, \dots, n, \tag{10.8}$$

and this is often put in the form

$$|\alpha_j - \omega_j| \leq \|Y\|, \quad (10.9)$$

which is useful when Y is a small perturbation of W . Less well known is the fact that Y 's rank limits the extent to which α_j can stray from ω_j .

The following result was known in the nineteenth century but appears not to have been written down in full until [Weyl, 1912].

The proof turns on the elementary fact that, for a given v , the Rayleigh quotient *is* a linear functional of its matrix argument;

$$\rho(v; W) + \rho(v; Y) = \rho(v; W + Y). \quad (10.10)$$

Theorem 10.3.1. *Let $A = W + Y$ with eigenvalues given above. For any i, j satisfying $1 \leq i + j - 1 \leq n$, the following inequalities hold*

$$\omega_i + \eta_j \leq \alpha_{i+j-1} \quad \text{and} \quad \alpha_{-(i+j-1)} \leq \omega_{-i} + \eta_{-j}.$$

Proof. Denote the eigenvectors of A, W , and Y by

$$Az_i = z_i \alpha_i, \quad Ww_i = w_i \omega_i, \quad Yy_i = y_i \eta_i, \quad i = 1, 2, \dots, n$$

and define

$$\mathcal{W}^i := \text{span}(w_1, \dots, w_i), \quad \mathcal{Y}^j := \text{span}(y_1, \dots, y_j).$$

For any three subspaces $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ of \mathcal{E}^n ,

$$\begin{aligned} \dim(\mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{S}_3) &= \dim(\mathcal{S}_1) + \dim(\mathcal{S}_2) + \dim(\mathcal{S}_3) \\ &\quad - \dim(\mathcal{S}_2 + \mathcal{S}_3) - \dim(\mathcal{S}_1 + (\mathcal{S}_2 \cap \mathcal{S}_3)) \\ &\geq \dim(\mathcal{S}_1) + \dim(\mathcal{S}_2) + \dim(\mathcal{S}_3) - n - n. \end{aligned}$$

Now select $\mathcal{S}_1 = (\mathcal{W}^{i-1})^\perp$, $\mathcal{S}_2 = (\mathcal{Y}^{j-1})^\perp$, $\mathcal{S}_3 = \mathcal{Z}^{i+j-1}$. A little arithmetic shows that the triple intersection must contain a nonzero vector v . So

$$\begin{aligned} \omega_i + \eta_j &= \min_{x \perp \mathcal{W}^{i-1}} \rho(x; W) + \min_{g \perp \mathcal{Y}^{j-1}} \rho(g; Y) \\ &\leq \rho(v; W) + \rho(v; Y), \quad \text{since } v \perp \mathcal{W}^{i-1}, v \perp \mathcal{Y}^{j-1}, \\ &= \rho(v; W + Y) \\ &\leq \max_{f \in \mathcal{Z}^{i+j-1}} \rho(f; A) = \alpha_{i+j-1}, \quad \text{since } v \in \mathcal{Z}^{i+j-1}. \end{aligned}$$

The second inequality may be written

$$-\omega_{-i} - \eta_{-j} \leq -\alpha_{-(i+j-1)}$$

and is an instance of the first one applied to $-A$. \square

This is perturbation theory without the restriction that Y be small! Table 10.1 illustrates the monotonicity theorem applied to some small matrices.

Example 10.3.1.

$$W = \begin{bmatrix} 5 & 2 & -1 \\ 2 & 3 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \quad Y = \begin{bmatrix} -1 & & \\ & 1 & \\ & & 2 \end{bmatrix}, \quad A = \begin{bmatrix} 4 & 2 & -1 \\ 2 & 4 & 1 \\ -1 & 1 & 3 \end{bmatrix}.$$

$$\begin{aligned} \{\lambda_i[W]\} &= \{-0.0571, 2.7992, 6.2579\}, \\ \{\lambda_i[Y]\} &= \{-1.0, 1.0, 2.0\}, \quad \{\lambda_i[A]\} = \{1.0, 4.0, 6.0\}. \end{aligned}$$

TABLE 10.1
Table for Example 10.3.1.

i	j	$\lambda_i[W] + \lambda_j[Y]$	\leq	$\lambda_{i+j-1}[A]$	$-i$	$-j$	$\lambda_i[W] + \lambda_j[Y]$	\geq	$\lambda_{1-i-j}[A]$
1	1	-1.0571		1.0	1	1	8.2579		6.0
2	1	1.7792		4.0	2	1	4.7792		4.0
1	2	0.9427			1	2	7.2579		
3	1	5.2579		6.0	3	1	1.9424		1.0
2	2	3.7792			2	2	3.7792		
1	3	1.9429			1	3	5.2579		

The more that is known of Y the more precise the inference we can draw. Let the inertia (defined in section 1.5) of Y be (π, ν, ζ) and its rank be ρ ; then, with $A = W + Y$ and $\rho < n$ as the next corollary shows,

every interval containing $\rho + 1$ ω 's contains at least one α .

Corollary 10.3.1 (the rank theorem). *With the notation of Theorem 10.3.1 let \mathbf{Y} 's inertia be (π, ν, ζ) and let $\rho = \pi + \nu = \text{rank}(\mathbf{Y})$. For all k such that $\nu < k \leq n - \pi$,*

$$\omega_{k-\rho} \leq \omega_{k-\nu} \leq \alpha_k \leq \omega_{k+\pi} \leq \omega_{k+\rho}.$$

Proof. By definition of ν , $\eta_{\nu+1} \geq 0$. Now choose $i = k - \nu$, $j = \nu + 1$ in Theorem 10.3.1 to find

$$\omega_{k-\nu} \leq \omega_{k-\nu} + \eta_{\nu+1} \leq \alpha_k.$$

Then pick $i = n + 1 - k - \pi$, $j = \pi + 1$ and apply the other inequality in Theorem 10.3.1, using $\eta_{-(\pi+1)} \leq 0$, to get

$$\alpha_{-(n+1-k)} \leq \omega_{-(n+1-k-\pi)} + \eta_{-(\pi+1)} \leq \omega_{-(n+1-k-\pi)}.$$

On using $\alpha_{-(n+1-k)} = \alpha_k$ the assertion is verified. \square

In particular, when \mathbf{Y} is positive semidefinite, i.e., $\nu(\mathbf{Y}) = 0$, then Weyl's original *monotonicity* result is obtained

$$\omega_k \leq \alpha_k \leq \omega_{k+\rho}, \quad k = 1, \dots, n - \rho.$$

In words, all eigenvalues increase, but the increase is limited by $\text{rank}(\mathbf{Y})$ as well as by $\|\mathbf{Y}\|$.

Example 10.3.2. Let $\mathbf{Y} = \mathbf{y}\mathbf{y}^* \neq \mathbf{O}$ be a positive semidefinite rank-one matrix; then $\rho = 1$ in the preceding result and

$$\omega_k \leq \alpha_k \leq \omega_{k+1}.$$

On the other hand we can pick $i = n + 1 - k$, $j = 1$ in the monotonicity theorem to obtain

$$\alpha_k \leq \omega_k + \eta_{-1} = \omega_k + \|\mathbf{Y}\|$$

which is the second inequality in (10.9).

Applications of the theorem occur in sections 4.5 and 11.5.

Exercise Section 10.3

10.3.1. When can equality occur in the monotonicity theorem?

10.4. The Residual Interlace Theorem

This section embarks on a presentation of work begun in the 1960s which has been inspired by demands to estimate eigenvalues of larger and larger matrices. Cauchy's interlace theorem is too general to be directly applicable to the task. Recall from section 10.1 that

$$A = \begin{bmatrix} H & B^* \\ B & Y \end{bmatrix}, \left\{ \begin{array}{ll} H \text{ is } m \text{ by } m, & \lambda_i[H] = \theta_i \\ A \text{ is } n \text{ by } n, & \lambda_i[A] = \alpha_i \end{array} \right\}. \quad (10.11)$$

Cauchy brackets each α_j using some θ_i and $\pm\infty$. These bounds are inevitably very weak when $m \leq n/2$, and yet interest centers on cases with $n = 1000$, $m = 100$.

What is lacking? Cauchy's theorem ignores B . Yet when $B = O$ each θ_i is an α_j for some j . It is plausible that when B is small, in some sense, then each θ_i must be close to one of the α_j .

In many applications B is available, and the theorems which follow can be regarded as providing the best possible inferences about the $\{\alpha_i\}$ when B is known in addition to the $\{\theta_i\}$. Actually B is a *residual* matrix and Chapter 11 explains how it arises in the Rayleigh–Ritz method.

B is usually long and thin, and for the remaining results of this chapter it must be put into the more useful form $\begin{bmatrix} C \\ 0 \end{bmatrix}$, where C has only $k[\equiv \text{rank}(B)]$ rows and satisfies $C^*C = B^*B$. There are several satisfactory ways in which this can be done, as is indicated in Exercise 11.5.6. Error bounds utilizing only H and $\|B\|$ are presented in Chapter 11.

From now on we study A 's of the form

$$A = \begin{bmatrix} H & C^* & O^* \\ C & V & Z^* \\ O & Z & W \end{bmatrix}; \quad \begin{array}{l} H \text{ is } m \text{ by } m, \\ V \text{ is } k \text{ by } k, \\ A \text{ is } n \text{ by } n, \end{array} \quad (10.12)$$

but only H and C are available. In some applications $k = m$; in others $k \ll m \ll n$; $k = 3$, $m = 40$, $n = 1000$ is typical. The results are directly usable when C is small. However in sections 10.6 and 10.7 it will be shown that with additional, very crude knowledge of the missing submatrix significantly tighter bounds can be obtained. It is quite surprising that useful information about the spectrum of a 1000-by-1000 matrix can be derived from the eigenvalues of a few matrices of order 50. (See Chapters 13, 14, and 15.)

Being ignorant of the k -by- k submatrix V we replace it by a k -by- k symmetric matrix X of our choice, and in what follows X is a parameter in an auxiliary matrix of order $m + k$:

TABLE 10.2
Table for Example 10.4.1.

X	μ_1	μ_2	μ_3	μ_4
diag(10,10)	9.382	9.9	10.1	11.618
diag(11,11)	9.9901	10.0	11.010	12.0
diag(10,11.21)	9.9	10.1	10.1	12.11
diag(9.85,11.1)	9.8	10.05	10.05	12.05

$$M(X) \equiv \begin{bmatrix} H & C^* \\ C & X \end{bmatrix}. \quad (10.13)$$

Its eigenvalues are denoted by $\mu_i = \mu_i(X) \equiv \lambda_i[M(X)]$, $i = 1, \dots, m+k$. Appropriate choices for X will be discussed shortly. With apt choices for X the μ 's give much more information than the θ 's alone provide. Cauchy's interlace theorem applied to the three pairs (A, H) , $(A, M(V))$, and $(M(X), H)$ yields (Exercise 10.4.3),

$$\alpha_i \leq \theta_i \leq \mu_{i+k}(X); \quad \mu_{-i-k}(X) \leq \theta_{-i} \leq \alpha_{-i}; \quad \alpha_i \leq \mu_i(V). \quad (10.14)$$

It is not always true that $\mu_i(X) \leq \alpha_i$; yet the next theorem shows that the interval $[\mu_i, \mu_{i+k}]$ must contain an eigenvalue of A . To say which one is more difficult.

Example 10.4.1. $H = \text{diag}(10,11)$, $C = \text{diag}(0.1,1)$. The simple bounds in section 4.5 dictate that $[9.9, 10.1]$ and $[10, 12]$ each contain an α , *possibly the same one* since the intervals overlap. However, any choice of X yields intervals $[\mu_1, \mu_3]$ and $[\mu_2, \mu_4]$ each containing its own α . This tiny auxiliary calculation removes the fear that 10 and 11 are approximating a single eigenvalue α . The last two X 's in Table 10.2 were chosen in the light of Theorem 10.5.1.

The generic case (Case 2 below) was established by W. Kahan (unpublished) in 1967. R. O. Hill and I removed the hypothesis of Case 2 in the 1990s.

Theorem 10.4.1. Consider \mathbf{A} as given in (10.12) together with eigenvalues $\mu_i(X)$, $i = 1, \dots, m+k$, of the auxiliary matrix $\mathbf{M}(X)$ of (10.13) with any k by k \mathbf{X} . Each interval $[\mu_j, \mu_{j+k}]$, $j = 1, \dots, m$ contains a different eigenvalue α_j of \mathbf{A} . In addition, there is a different eigenvalue α_I outside each open interval (μ_i, μ_{i+m}) , $i = 1, \dots, k$.

The distinction between the interior and exterior intervals disappears if the real line is closed ($-\infty = +\infty$) and if each subscript j on μ is read as $j - m - k$ if $j > m + k$.

Proof.

1. If $\mathbf{V} = \mathbf{X}$ it suffices to apply Cauchy's interlace theorem. The submatrices \mathbf{Z} and \mathbf{W} of (10.2) are unknown but, just as \mathbf{B} was transformed into $\begin{pmatrix} \mathbf{C} \\ \mathbf{O} \end{pmatrix}$, so may \mathbf{Z} be transformed into $\begin{pmatrix} \mathbf{O} \\ \mathbf{F} \end{pmatrix}$, where \mathbf{F} is $\sigma \times k$, $\sigma = \text{rank}(\mathbf{Z}) \leq k$. Thus \mathbf{A} is similar to

$$\mathbf{A}' = \begin{bmatrix} \mathbf{H} & \mathbf{C}^* & \mathbf{O}^* & \mathbf{O}^* \\ \mathbf{C} & \mathbf{V} & \mathbf{O}^* & \mathbf{F}^* \\ \mathbf{O} & \mathbf{O} & \tilde{\mathbf{W}}_{11} & \tilde{\mathbf{W}}_{21}^* \\ \mathbf{O} & \mathbf{F} & \tilde{\mathbf{W}}_{21} & \tilde{\mathbf{W}}_{22} \end{bmatrix}, \quad \tilde{\mathbf{W}} \text{ is similar to } \mathbf{W}.$$

Delete the last σ rows and columns. By Theorem 10.1.1

$$\alpha_j \leq \lambda_j[\mathbf{M} \oplus \tilde{\mathbf{W}}_{11}] \leq \alpha_{j+\sigma} \leq \lambda_{j+\sigma}[\mathbf{M} \oplus \tilde{\mathbf{W}}_{11}], \quad j = 1, \dots, n - 2\sigma.$$

Since each $\mu_j(\mathbf{V})$ is an eigenvalue of $[\mathbf{M} \oplus \tilde{\mathbf{W}}_{11}]$ each interval $[\mu_j(\mathbf{V}), \mu_{j+\sigma}(\mathbf{V})]$, $j = 1, \dots, m - \sigma + k$ contains its own α , say, α_J . By (10.6) there are σ other α 's, say, α_I , satisfying

$$\alpha_I \notin (\mu_i(\mathbf{V}), \mu_{m+i-\sigma+k}(\mathbf{V})), \quad i = 1, \dots, \sigma.$$

When $\sigma < k$ these conclusions are slightly stronger (smaller interval) than the claims in Theorem 10.4.1, but in general σ will not be known so take the worst case $\sigma = k$.

2. If $\text{rank}(\mathbf{V} - \mathbf{X}) = k$ then $\mathbf{V} - \mathbf{X}$ is invertible and \mathbf{A} may be split as

$$\mathbf{A} = \begin{bmatrix} \mathbf{H} & \mathbf{C}^* & \mathbf{O} \\ \mathbf{C} & \mathbf{X} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \tilde{\mathbf{W}} \end{bmatrix} + \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{V} - \mathbf{X} & \mathbf{Z}^* \\ \mathbf{O} & \mathbf{Z} & \mathbf{Z}(\mathbf{V} - \mathbf{X})^{-1}\mathbf{Z}^* \end{bmatrix}$$

$$\begin{aligned} &= U(X) + T(X), \\ \tilde{W} &= W - Z(V - X)^{-1}Z^*. \end{aligned}$$

By construction $T(X)$ is congruent to

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & V - X & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and so has rank k . By Corollary 10.3.1 every interval containing $k+1$ (consecutive) eigenvalues of $M \oplus \tilde{W}_{11}$ contains its own α . Since each $\mu_i(X)$ is an eigenvalue of $M \oplus \tilde{W}_{11}$ there are m α_J 's with

$$\alpha_J \in [\mu_j(X), \mu_{j+k}(X)], \quad j = 1, \dots, m.$$

By (10.6) there are k other eigenvalues, α_I with

$$\alpha_I \notin (\mu_i(X), \mu_{m+i}(X)), \quad i = 1, \dots, k.$$

3. Let $\nu = \text{rank}(V - X)$ with $0 < \nu < k$. Then

$$V - X = Q \begin{pmatrix} Y & 0 \\ 0 & 0 \end{pmatrix} Q^*$$

with $Q^* = Q^{-1}$ and $\nu \times \nu$ Y is invertible. The similarity $A' = (I \oplus Q^* \oplus I)A(I \oplus Q \oplus I)$ leaves unchanged the eigenvalues of A and $M(X)$. So there is no loss in redefining Q^*C as C , Q^*XQ as X , and ZQ as $(Z_1 Z_2)$ to begin again with the splitting

$$\begin{aligned} A' &= \begin{bmatrix} H & C^* & \begin{pmatrix} 0 \\ O^* \\ Z_2^* \end{pmatrix} \\ C & X & \tilde{W} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \begin{pmatrix} Y & O^* \\ O & 0 \end{pmatrix} & \begin{pmatrix} Z_1^* \\ O^* \end{pmatrix} \\ 0 & (Z_1 O) & Z_1 Y^{-1} Z_1^* \end{bmatrix} \\ &= U(X) + T(X), \end{aligned}$$

where

$$\tilde{W} = W - Z_1 Y^{-1} Z_1^*.$$

By construction $T(X)$ is congruent to $(O \oplus Y \oplus O)$ and has rank ν . By Corollary 10.3.1 there are α 's satisfying

$$\begin{aligned} \alpha_J &\in [\lambda_j[U], \lambda_{j+\nu}[U]], \quad j = 1, \dots, n - \nu, \quad (10.15) \\ \alpha_I &\notin (\lambda_i[U], \lambda_{n-\nu+i}[U]), \quad i = 1, \dots, \nu. \end{aligned}$$

Now consider $U = U(X)$. If $Z_2 = O$ then $U = M \oplus \tilde{W}$ and every eigenvalue μ of M is an eigenvalue of U . Hence every interval $[\mu_j, \mu_{j+\nu}], j = 1, \dots, m+k-\nu$ contains its own α , not necessarily α_J , and there are ν α 's outside the open intervals $(\mu_i, \mu_{m+k-\nu+i})$, $i = 1, \dots, \nu$. If $Z_2 \neq O$ then, just as B was transformed into $\begin{pmatrix} O \\ C \end{pmatrix}$, so can Z_2 be written $Z_2 = P \begin{pmatrix} O \\ F \end{pmatrix}$ with $P^* = P^{-1}$ and F is $\sigma \times (k-\nu)$ where

$$\sigma = \text{rank}(Z_2) \leq k - \nu. \quad (10.16)$$

Define \hat{W} and partition it conformably with F by

$$\hat{W} := P^* \tilde{W} P = \begin{pmatrix} \hat{W}_{11} & \hat{W}_{21} \\ \hat{W}_{12} & \hat{W}_{22} \end{pmatrix}$$

so that $U(X)$ is similar to

$$\begin{bmatrix} H & C^* & O \\ C & X & \begin{pmatrix} O^* & O^* \\ O^* & F^* \end{pmatrix} \\ O & \begin{pmatrix} O & O \\ O & F \end{pmatrix} & \begin{pmatrix} \hat{W}_{11} & \hat{W}_{21}^* \\ \hat{W}_{12} & \hat{W}_{22} \end{pmatrix} \end{bmatrix}.$$

Deletion of the last σ rows and columns leaves $M \oplus \hat{W}_{11}$ and, by Theorem 10.1.1,

$$\begin{aligned} \lambda_j[M \oplus \hat{W}_{11}] &\leq \lambda_{j+\sigma}[U], \quad j = 1, \dots, n - \sigma, \\ \lambda_i[U] &\leq \lambda_i[M \oplus \hat{W}_{11}], \quad i = 1, \dots, n - \sigma. \end{aligned} \quad (10.17)$$

Combine (10.15) and (10.17) to find separate α 's in each interval $[\lambda_j[M \oplus \hat{W}_{11}], \lambda_{j+\nu+\sigma}[M \oplus \hat{W}_{11}]]$, $j = 1, \dots, n - 2\sigma - \nu$ and $\sigma + \nu$ more α 's outside the open intervals

$$(\lambda_i[M \oplus \hat{W}_{11}], \lambda_{n-2\nu-\sigma+i}[M \oplus \hat{W}_{11}]), \quad i = 1, \dots, \sigma + \nu.$$

A fortiori, each interval

$$[\mu_j, \mu_{j+\nu+\sigma}], \quad j = 1, \dots, m + k - \sigma - \nu$$

contains its own α , and there are $\nu + \sigma$ more α 's outside the open intervals

$$(\mu_i, \mu_{m+k-\nu-\sigma+i}) \quad i = 1, \dots, \sigma + \nu.$$

When $\sigma + \nu < k$ these conclusions are slightly stronger (smaller intervals) than the claims in Theorem 10.4.1, but in general σ and ν will be unknown so take the worst case $\sigma + \nu = k$. \square

How should X be chosen? The answer depends on whether or not there is any adscititious (= received from outside) information about A besides H and C . Various cases are considered in the following sections.

Exercises on Section 10.4

10.4.1. Show that if $X_1 \geq X_2$, in the sense that $X_1 - X_2$ is nonnegative definite, then $\mu_j(X_1) \geq \mu_j(X_2)$, $j = 1, \dots, m+k$.

10.4.2. Consider the simple choice $X = \xi$ for large scalar ξ . Use Sylvester's theorem to show that

$$\text{as } \xi \rightarrow -\infty, \mu_{i+k}(\xi) \rightarrow \begin{cases} -\infty & \text{for } 1-k \leq i \leq 0, \\ \theta_i & \text{for } 1 \leq i \leq m, \end{cases}$$

$$\text{as } \xi \rightarrow +\infty, \mu_{-i-k}(\xi) \rightarrow \begin{cases} +\infty & \text{for } 1-k \leq i \leq 0, \\ \theta_{-i} & \text{for } 1 \leq i \leq m. \end{cases}$$

10.4.3. Use Cauchy's theorem to derive all the inequalities in (10.14).

*10.5. Lehmann's Optimal Intervals

In the event of no extra outside information about A other than the submatrices H and C of (10.12) the best choice for the matrix X of (10.13) was given, implicitly, in [Lehmann, 1949] and [Lehmann, 1966]. In our development this result, namely, Theorem 10.5.2, is an immediate corollary of Kahan's residual interlace Theorems 10.4.1 and 10.5.1.

The word optimal has the special, but reasonable, meaning that for *any real number* ζ one can describe the X which yields the tightest bounds for A 's eigenvalues α on either side of ζ . For simplicity we require that $\zeta \neq \theta_i (\equiv \lambda_i[H])$, $i = 1, \dots, m$. Recall from Theorem 10.4.1 that all the inclusion intervals derived from $M(X)$ are of the form $[\mu_j, \mu_{j+k}]$ where k is the rank of C . Consequently, in order to have a pair of inclusion intervals abutting at ζ , say $[\mu', \zeta], [\zeta, \mu'']$, it is necessary that ζ be a k -fold eigenvalue of $M(X)$. The proper choice turns out to be the k -by- k matrix

$$X_\zeta \equiv \zeta + C(H - \zeta)^{-1}C^*. \quad (10.18)$$

Lemma 10.5.1. *The parameter ζ is a k -fold eigenvalue of $M(X_\zeta)$. Further, each eigenvalue $\mu_i(X_\zeta)$ is a monotone nondecreasing function of ζ , $i = 1, 2, \dots, m+k$, between eigenvalues of H .*

Proof. Consider the block triangular factorization of $M - \zeta$

$$\begin{aligned} M(X_\zeta) - \zeta &= \begin{bmatrix} H - \zeta & C^* \\ C & C(H - \zeta)^{-1}C^* \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ C(H - \zeta)^{-1} & I \end{bmatrix} \begin{bmatrix} H - \zeta & 0 \\ 0^* & 0 \end{bmatrix} \begin{bmatrix} I & (H - \zeta)^{-1}C^* \\ 0^* & I \end{bmatrix}. \end{aligned}$$

By Sylvester's inertia theorem $M - \zeta$ has the same nullity (namely, order $- \text{rank}$) as $\text{diag}(H - \zeta, 0)$ which is k since $\zeta \neq \lambda_i[H]$. In other words ζ is a k -fold eigenvalue of $M(X_\zeta)$.

To show monotonicity we invoke the minmax characterization (section 10.2) to find

$$\mu_i(X_\zeta) = \min_{S^i} \max_{s \in S^i} \frac{s^* M(X_\zeta) s}{s^* s},$$

where S^i is any subspace of dimension i . Partition s^* as (u^*, v^*) where u is m by 1, v is k by 1. Then, by the definition of M_ζ in (10.13),

$$s^* M_\zeta s = u^* Hu + u^* C^* v + v^* Cu + v^* (\zeta + C(H - \zeta)^{-1}C^*) v$$

and, for $\zeta \neq \lambda_i[H]$,

$$\begin{aligned} \frac{d}{d\zeta} (s^* M_\zeta s) &= v^* v + v^* C(H - \zeta)^{-2} C^* v \\ &= \|v\|^2 + \|(H - \zeta)^{-1} C^* v\|^2 \geq 0. \quad \square \end{aligned}$$

The best interval depends on the location of ζ .

Theorem 10.5.1. *Consider (10.12) and (10.18). Let ζ be any number satisfying $\theta_j < \zeta < \theta_{j+1}$. Each interval $[\mu_j(X_\zeta), \zeta]$ and $[\zeta, \mu_{j+k+1}(X_\zeta)]$ contains at least one of A 's eigenvalues. Moreover there is an A with eigenvalues only at the endpoints of these intervals.*

Proof. The interlace Theorem 10.1.1 applied to $M(X_\zeta)$ yields

$$\mu_{j+l}(X_\zeta) \leq \theta_{j+l} \leq \mu_{j+k+l}(X_\zeta), \quad l = 0, 1.$$

Since ζ is a k -fold eigenvalue of $M(X_\zeta)$ these inequalities tell us which of M 's eigenvalues equals ζ ;

$$\mu_j \leq \theta_j < \mu_{j+1} = \mu_{j+2} = \cdots = \mu_{j+k} = \zeta < \theta_{j+1} \leq \mu_{j+k+1}.$$

The conclusion is now a corollary of Kahan's Theorem 10.4.1 and simply exploits the coincidence of the μ 's. The A which proves optimality is $\text{diag}[M(X_\zeta), \zeta I_{n-m-k}]$; all its eigenvalues, other than the μ 's, are at ζ . \square

It is not obvious from this result how to pick ζ to reduce the width of $[\mu_j, \zeta]$ to a minimum. Indeed no general formula is known, but Figure 10.1 shows how the width varies with ζ . An application was given in Example 10.4.1. The original form in which Lehmann presented his results confirms that ζ be chosen not too far from θ_j . How that formulation is derived from the preceding theorem is indicated in Exercises 10.5.5, 10.5.6, and 10.5.7.

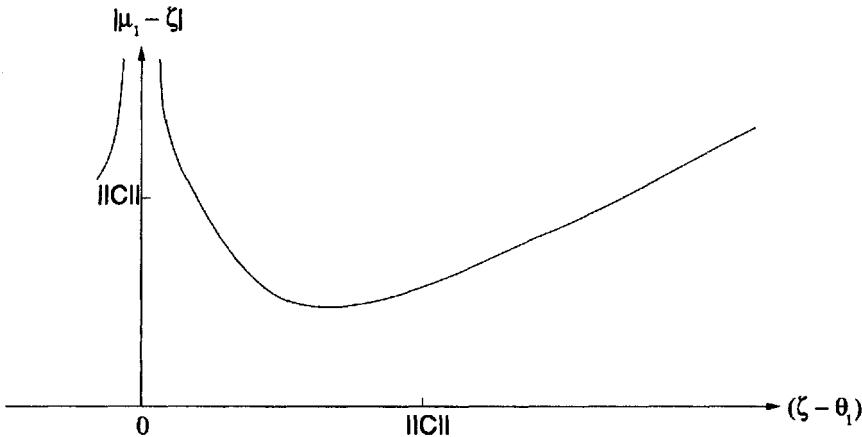


FIG. 10.1. The width of an optimal interval as a function of $\zeta - \theta_1$.

For any real number ξ Lehmann defines

$$\delta_i = \delta_i(\xi) \equiv \sqrt{\lambda_i[(H - \xi)^2 + C^*C]}, \quad i = 1, \dots, m. \quad (10.19)$$

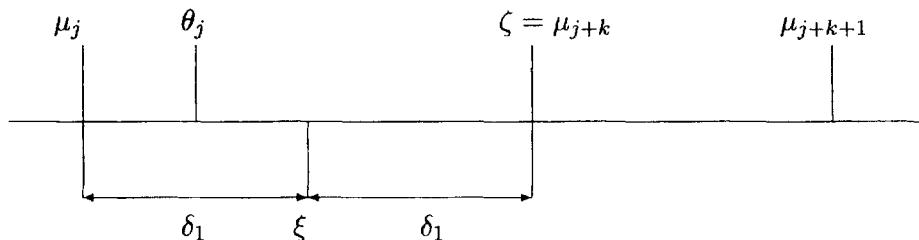
Here is the original formulation of Theorem 10.5.1 from [Lehmann, 1949].

Theorem 10.5.2. For every ξ and for $i = 1, 2, \dots, m$, \mathbf{A} has at least i eigenvalues in $[\xi - \delta_i, \xi + \delta_i]$, \mathbf{A} has at least $m+1-i$ eigenvalues not in $(\xi - \delta_i, \xi + \delta_i)$, and there is an \mathbf{A} with no eigenvalue in this open interval.

In Lehmann's formulation the goal is to minimize $\delta_1(\xi)$ and by Weyl's monotonicity theorem applied to (10.19),

$$\delta_1(\xi)^2 \leq \lambda_1[(\mathbf{H} - \xi)^2] + \|\mathbf{C}\|^2.$$

In many applications $\|\mathbf{C}\|$ is small compared with $\|\mathbf{H}\|$ and, in any case, this bound on δ^2 is minimized when $\xi = \theta_j$ for some j . However, the choice $\xi = \theta_j$ does not in general minimize δ_1 . It turns out (Exercise 10.5.5) that ξ and $\delta_1(\xi)$ are, respectively, the midpoint and half-width of $[\mu_j, \mu_{j+k}]$ as indicated in the next figure.



Which formulation is preferable? The matrix $(\mathbf{H} - \xi)^2 + \mathbf{C}^* \mathbf{C}$ is m by m while $\mathbf{M}(\mathbf{X}_\xi)$ is $(m+k)$ by $(m+k)$. However, k of \mathbf{M} 's eigenvalues are known and the corresponding subspace is known too (Exercise 10.5.8). This information can be used to facilitate the calculation of the rest of \mathbf{M} 's eigenvalues, but such efficiency considerations must take second place to those of accuracy.

Unfortunately the δ formulation forces the calculation of $\delta_1^2(\xi)$ in order to find $\delta_1(\xi)$. Now δ_1^2 is the smallest eigenvalue and so (see section 2.7) will be computed with an error which is tiny compared with $\|(\mathbf{H} - \xi)^2 + \mathbf{C}^* \mathbf{C}\|$. The cases of greatest interest will have $\delta_1/|\xi|$ small, say 10^{-5} , and although δ_m^2 will be correct to working accuracy we must expect 10 fewer correct decimals in δ_1^2 and hence in δ_1 . This loss is enough to destroy the value of the computation just when its potential is greatest. Since $\delta_1/|\xi|$ is small there is no need for very high relative accuracy in δ_1 but there is need for some correct figures. With the \mathbf{M} formulation the computed values μ_j and $\mu_{j+k} (= \zeta)$ may be very

close, but that does not prevent a good program from computing them to as much accuracy as their separation warrants. To summarize, the formulation (10.19) inevitably discards nearly twice as many figures as is necessary. Exercise 10.5.10 illustrates this phenomenon.

On the other hand since the 1980s the preferred way to compute $\delta_1(\xi)$ is as the smallest singular value of $[\mathbf{H} - \xi, \mathbf{C}^*]$ and this approach restores Theorem 10.5.2 to favor.

Exercises on Section 10.5

- 10.5.1. What happens to $\mathbf{X}_\zeta = \zeta + \mathbf{C}(\mathbf{H} - \zeta)^{-1}\mathbf{C}^*$ and $\mu_i(\mathbf{X}_\zeta)$, $i = 1, \dots, m+k$ as $\zeta \rightarrow \theta_j$?
- 10.5.2. Show that for $i \leq j$ the interval $[\mu_i(\mathbf{X}_\zeta), \zeta]$ contains at least as many eigenvalues α of \mathbf{A} as $[\mu_i(\mathbf{X}_\zeta), \zeta]$ contains eigenvalues μ of $\mathbf{M}(\mathbf{X}_\zeta)$. Similarly for $i > j$ the interval $[\zeta, \mu_i(\mathbf{X}_\zeta)]$ contains at least as many eigenvalues α as $[\zeta, \mu_i(\mathbf{X}_\zeta)]$ contains μ 's.
- 10.5.3. What is the smallest number of α 's which can lie in the union of $[-\infty, \mu_i(\mathbf{X}_\zeta)]$ and (ζ, ∞) ? *Hint:* Use the last part of Theorem 10.5.1.
- 10.5.4. Show that the interval in Exercise 10.5.2 is optimal in the sense that no subinterval can be proved to contain that many of \mathbf{A} 's eigenvalues.
- 10.5.5. (Difficult.) Let $\theta_j < \zeta < \theta_{j+1}$ and let μ be any eigenvalue of $\mathbf{M}(\mathbf{X}_\zeta)$ other than ζ . Define $\xi = (\zeta + \mu)/2$, $\delta = (\zeta - \mu)/2$. Show that

$$\det[(\mathbf{H} - \xi)^2 + \mathbf{C}^*\mathbf{C} - \delta^2] = 0.$$

- 10.5.6. Let $\delta_i^2(\xi) \equiv \lambda_i[(\mathbf{H} - \xi)^2 + \mathbf{C}^*\mathbf{C}]$, $i = 1, \dots, m$. Use Cauchy's interlace theorem applied to $(\mathbf{A} - \xi)^2$ to prove that $[\xi - \delta_j(\xi), \xi + \delta_j(\xi)]$ contains at least j of \mathbf{A} 's eigenvalues.
- 10.5.7. Derive Lehmann's original formulation from Theorem 10.5.1.
- 10.5.8. Verify that for any k -vector v the $(m+k)$ -vector $[v^*\mathbf{C}(\mathbf{H} - \zeta)^{-1}, -v^*]^*$ is an eigenvector of $\mathbf{M}(\mathbf{X}_\zeta)$ belonging to ζ .
- 10.5.9. (Difficult.) Deflate $\mathbf{M}(\mathbf{X}_\zeta)$ analytically to obtain a more complicated m -by- m matrix whose eigenvalues are $\mu_1, \dots, \mu_j, \mu_{j+k+1}, \dots, \mu_{m+k}$. *Hint:* Use the matrix $\begin{bmatrix} (\mathbf{H} - \zeta)^{-1}\mathbf{C}^* \\ -\mathbf{I}_k \end{bmatrix} \mathbf{R}$, where \mathbf{R} is k -by- k upper triangular chosen so that the columns of the product are orthonormal; i.e., $\mathbf{R}^{-*}\mathbf{R}^{-1}$ is the Cholesky decomposition (see Chapter 3) of

$$\mathbf{C}(\mathbf{H} - \zeta)^2\mathbf{C}^* + \mathbf{I}.$$

10.5.10. Consider the case $\xi = 1$, $C = (0, \gamma^2)$, and $H = [\begin{smallmatrix} \gamma & 1 \\ 1 & \gamma \end{smallmatrix}]$. Suppose that the sum $1 + \gamma^4$ is computed as 1 (or try $\gamma = 10^{-3}$). Study the computation of $\delta_1(\xi)$ as given in (10.19). Then consider the computation of the $\mu_i(X_\zeta)$ when $\zeta = 1 + \gamma^2/\sqrt{1 + \gamma^2}$.

*10.6. The Use of Bounds on the Missing Submatrix

Recall that we are studying matrices of the form

$$A = \begin{bmatrix} H & C^* & O^* \\ C & Y \\ O & & \end{bmatrix}; \quad A \text{ is } n \text{ by } n, \quad H \text{ is } m \text{ by } m, \quad (10.20)$$

and Y is unknown. Lehmann's bounds can be improved a lot if just a little is known about Y , a bound on $\|Y\|$ or $\|Y^{-1}\|$ for instance. To derive these smaller intervals it is helpful to see what can be said when Y 's eigenvalues $\{\eta_i, i = 1, \dots, n-m\}$ are known. To this end set the matrix X in (10.13) to ηI_k ;

$$M(\eta) \equiv \begin{bmatrix} H & C^* \\ C & \eta I_k \end{bmatrix}, \quad \mu_i(\eta) \equiv \lambda_i[M(\eta)], \quad i = 1, \dots, m+k. \quad (10.21)$$

Convention: $\mu_i = -\infty$ when $i \leq 0$; $\mu_i = +\infty$ when $i > m+k$.

The following blizzard of results comes from Weyl's monotonicity theorem 10.3.1. They were obtained by Kahan in 1957 but have been derived independently and published in [Weinberger, 1959 and 1974].

Lemma 10.6.1. *For $0 \leq i \leq m+1$, $0 \leq j \leq n-m$, and $\alpha_i = \lambda_i[A]$,*

$$\min\{\mu_i(\eta_{j+1}), \eta_{j+1}\} \leq \alpha_{i+j} \leq \max\{\mu_{i+k}(\eta_j), \eta_j\}, \quad (10.22)$$

$$\begin{aligned} \min\{\mu_{-i-k}(\eta_{-j}), \eta_{-j}\} &\leq \alpha_{-i-j} \\ &\leq \max\{\mu_{-i}(\eta_{-i-j}), \eta_{-j-1}\}, \end{aligned} \quad (10.23)$$

$$\alpha_i \leq \mu_i(\eta_{-1}), \quad (10.24)$$

$$\mu_{-i}(\eta_1) \leq \alpha_{-i}.$$

Proof. Split \mathbf{A} appropriately and apply the monotonicity theorem in section 10.3 adroitly to get

$$\begin{aligned}\alpha_{i+j} &\geq \lambda_i \left[\begin{bmatrix} \mathbf{H} & \mathbf{C}^* \mathbf{O}^* \\ \mathbf{C} & \eta_{j+1} \end{bmatrix} \right] + \lambda_{j+1} \left[\begin{bmatrix} \mathbf{O} & \mathbf{O}^* & \mathbf{O}^* \\ \mathbf{O} & \mathbf{Y} - \eta_{j+1} & \mathbf{O} \end{bmatrix} \right] \\ &= \lambda_i \left[\begin{pmatrix} \mathbf{M}(\eta_{j+1}) & \mathbf{O}^* \\ \mathbf{O} & \eta_{j+1} \end{pmatrix} \right] + 0 \\ &\geq \min\{\mu_i(\eta_{j+1}), \eta_{j+1}\}.\end{aligned}$$

Note that the second term is still 0 when $j = n - m$ and $\eta_{j+1} = +\infty$. The result is vacuously true when $i = 0$. The dual part of the monotonicity theorem leads to

$$\alpha_{-p-q} \leq \max\{\mu_{-q}(\eta_{-q-1}), \eta_{-q-1}\}.$$

Now translate this, using $\alpha_{-l} = \alpha_{n+1-l}$, $\mu_{-l} = \mu_{m+k+1-l}$, $\eta_{-l} = \eta_{n-m+1-l}$, with $p = m + 1 - i$, $q = n - m - j$, to get

$$\alpha_{i+j} \leq \max\{\mu_{i+k}(\eta_j), \eta_j\}.$$

Thus (10.22) is proved and (10.23) is equivalent to (10.22).

The last result, (10.25), is left as Exercise 10.6.1. \square

The preceding results are academic; if the η_i were known then the role of \mathbf{Y} and \mathbf{H} could be reversed to good effect. In practice if anything is known of \mathbf{Y} it is likely to be a crude bound. Recall that $\eta \leq \mathbf{Y}$ means that $\eta \leq \rho(\mathbf{x}; \mathbf{Y})$ for all $\mathbf{x} \neq \mathbf{o}$.

Corollary 10.6.1. For $i = 1, \dots, m$,

if $\eta' \leq \mathbf{Y}$ then

$$\min\{\mu_i(\eta'), \eta'\} \leq \alpha_i \text{ and } \mu_{-i}(\eta') \leq \alpha_{-i},$$

if $\mathbf{Y} \leq \eta''$ then

$$\alpha_i \leq \mu_i(\eta'') \text{ and } \alpha_{-i} \leq \max\{\mu_{-i}(\eta''), \eta''\}. \quad (10.25)$$

Proof. By monotonicity of the μ_i , Lemma 10.5.1,

$$\mu_i(\eta') \leq \mu_i(\eta_1), \quad \mu_i(\eta_{-1}) \leq \mu_i(\eta'').$$

Now apply (10.22) with $j = 0$ for the lower bound on α_i and (10.25) for the upper bound. The treatment of α_{-i} is similar. \square

It is also possible to exploit a known gap in \mathbf{Y} 's spectrum to find some inclusion intervals.

Corollary 10.6.2. Suppose that

$$\eta_\pi \leq \eta' < \eta'' \leq \eta_{\pi+1} \text{ for some } \pi, \text{ and} \quad (10.26)$$

$$\theta_j \leq \eta' \leq \theta_{j+1} \leq \dots \leq \theta_{j+l} \leq \eta'' < \theta_{j+l+1}; \quad (10.27)$$

then

$$\mu_i(\eta'') \leq \alpha_{\pi+i} \leq \mu_{i+k}(\eta'), \quad j+1 \leq i \leq j+l. \quad (10.28)$$

The proof is left as Exercise 10.6.3. Note that it is not necessary that the index π be known for these bounds to be of use.

Example 10.6.1.

$$\mathbf{A} = \begin{bmatrix} \theta & \gamma & 0 \\ \gamma & \mathbf{Y} & \\ 0 & & \end{bmatrix}, \quad m = k = 1, \\ \mathbf{H} = \theta, \quad \mathbf{C} = \gamma.$$

$$\mathbf{M}(\xi) = \begin{bmatrix} \theta & \gamma \\ \gamma & \xi \end{bmatrix}, \quad 0 < \gamma \ll \theta.$$

The residual norm bound of section 4.5 using e_1 as an approximate eigenvector shows that there is an α in $[\theta - \gamma, \theta + \gamma]$. Assuming nothing of \mathbf{Y} the optimal Lehmann-Kahan interval using the value $\zeta = \theta + \gamma$ also turns out to be $[\theta - \gamma, \theta + \gamma]$ and all other ζ 's produce bigger bounds. Now we try some other \mathbf{M} 's and let $\sigma \equiv \sqrt{\theta^2 + 4\gamma^2} \approx \theta + 2\gamma^2/\theta$.

ξ	μ_1	μ_2
0	$\frac{1}{2}(\theta - \sigma) \approx -\gamma^2/\theta$	$\frac{1}{2}(\theta + \sigma) \approx \theta + \gamma^2/\theta$
θ	$\theta - \gamma$	$\theta + \gamma$
2θ	$\theta + \frac{1}{2}(\theta - \sigma) \approx \theta - \gamma^2/\theta$	$\theta + \frac{1}{2}(\theta + \sigma) \approx 2\theta + \gamma^2/\theta$

Assumption: $0 \leq Y \leq 2\theta$.

Inference from (10.25):

$$-\gamma^2/\theta \leq \alpha_1 \leq \theta - \gamma^2/\theta, \quad \theta + \gamma^2/\theta \leq \alpha_3 \leq 2\theta + \gamma^2/\theta.$$

The α 's have a lot of freedom.

Assumption: $\theta \leq Y \leq 2\theta$.

Inference from (10.25):

$$\theta - \gamma \leq \alpha_1 \leq \theta - \gamma^2/\theta, \quad \theta + \gamma \leq \alpha_3 \leq 2\theta + \gamma^2/\theta.$$

Note that α_1 is tightly constrained.

Assumption: $\eta_1 \leq 0 < 2\theta \leq \eta_2$.

Inference from Corollary 10.6.2:

$$\alpha_2 = \theta[1 + \gamma^2/\theta^2 + O(\gamma^4/\theta^4)]!!$$

This concludes Example 10.6.1.

What has not been shown is that the bounds in (10.25) and Corollary 10.6.2 are an improvement over Lehmann's. The difference is strongest as $\|C\| \rightarrow 0$ and $\theta_j \rightarrow \alpha_{j+p}$ for some p . The width of Lehmann's interval is $2\delta_1(\xi)$ and from the definition of δ_j in (10.19),

$$\delta_1(\xi)^2 \geq \lambda_1[H - \xi]^2. \quad (10.29)$$

Thus $\delta_1(\xi) = O(\|C\|)$ if $|\xi - \theta_j| = O(\|C\|)$ as $\|C\| \rightarrow 0$.

In contrast

$$\mu_j(\eta) \geq \theta_j - \frac{2\|C\|^2}{\eta - \theta_j + \sqrt{(\eta - \theta_j)^2 + 4\|C\|^2}} \quad (10.30)$$

and so, for example, the intervals $[\mu_i(\eta'), \mu_i(\eta'')]$ shrink down on α_i as $O(\|C\|^2)$ provided that η', η'' are separated from θ_j and are independent of $\|C\|$. A proof of (10.30) is given in [Wielandt, 1967]. Example 10.6.1 illustrates this phenomenon.

Exercises on Section 10.6

- 10.6.1. By using the Weyl monotonicity result Theorem 10.3.1 shows that $\alpha_i \leq \mu_i(\eta_{-1})$ and $\mu_{-i}(\eta_1) \leq \alpha_{-i}$.

10.6.2. Show that if $\eta' \leq Y \leq \eta''$ then

$$\mu_{-i}(\eta') \leq \alpha_{-i} \leq \max\{\mu_{-i}(\eta''), \eta''\}.$$

10.6.3. Prove Corollary 10.6.2 by applying Lemma 10.6.1 with j replaced by π .

10.6.4. If the eigenvalues η_i of Y are known then they can be related to the eigenvalues θ_i of H . Prove that if $\theta_i \leq \eta_{j+1}$ then $\mu_i(\eta_{j+1}) \leq \alpha_{i+j}$ and that if $\eta_j \leq \theta_i$ then $\alpha_{i+j} \leq \mu_{i+k}(\eta_j)$.

*10.7. The Use of Gaps in A's Spectrum

Consider the example in section 10.6 in which there is a good approximation θ to an eigenvalue α ,

$$A = \begin{bmatrix} \theta & \gamma & 0 \\ \gamma & Y & \\ 0 & & \end{bmatrix}, \quad M(\xi) = \begin{bmatrix} \theta & \gamma \\ \gamma & \xi \end{bmatrix}.$$

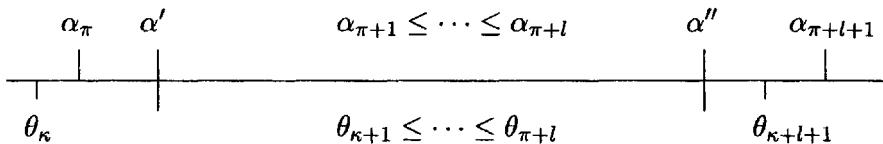
With no knowledge of Y the best inference is that α lies in $[\theta - \gamma, \theta + \gamma]$. Suppose, however, that adscititious knowledge comes in the form that α is the only eigenvalue of A in a *larger* interval $[\theta - \beta, \theta + \beta]$ where $\gamma < \beta$. The strong conclusion is that α actually lies in $[\theta - \gamma^2/\tau, \theta + \gamma^2/\tau]$ where $\tau = (\beta + \sqrt{\beta^2 + 4\gamma^2})/2 \approx \beta + \gamma^2/\beta$. If $\beta/\theta = 10^{-1}$, $\gamma/\theta = 10^{-3}$ then θ agrees with α to five decimal figures. Observations such as this permit prompt termination of expensive iterative procedures for calculating α 's.

These better bounds come from the use of two samples of the auxiliary matrix M .

ξ	$\mu_1(\xi)$	$\mu_2(\xi)$
$\theta - \beta = \alpha'$	$\theta - \tau$	$\theta + \gamma^2/\tau$
$\theta + \beta = \alpha''$	$\theta - \gamma^2/\tau$	$\theta + \tau$

In 1929 Temple proved essentially that $\mu_1(\alpha'') \leq \alpha \leq \mu_2(\alpha')$ and the theorem below generalizes it and similar results of Kato. See [Temple, 1933] and [Temple and Bickley, 1993].

Consider A in the form (10.20) with $\|C\|$ small enough that an interval, $[\alpha', \alpha'']$, say, is known to contain the *same* number of α 's (A 's eigenvalues) as θ 's (H 's eigenvalues) as indicated in the figure.



The optimal intervals are given by eigenvalues μ of the auxiliary matrix $M(X_\zeta)$ of (10.13) and (10.18) with appropriate choice of ζ . Theorem 10.7.1 is a corollary of Theorem 10.5.1 and was first stated in [Lehmann, 1949].

Theorem 10.7.1. Suppose that for some indices π, κ , and l

$$\alpha_\pi < \alpha' \leq \alpha_{\pi+1} \leq \cdots \leq \alpha_{\pi+l} \leq \alpha'' < \alpha_{\pi+l+1}, \quad (10.31)$$

$$\theta_\kappa < \alpha' < \theta_{\kappa+1} \leq \cdots \leq \theta_{\kappa+l} < \alpha'' < \theta_{\kappa+l+1}, \quad (10.32)$$

then

$$\kappa \leq \pi, \quad \text{and} \quad (10.33)$$

$$\mu_{\kappa+j}(X_{\alpha''}) \leq \alpha_{\pi+j} \leq \mu_{\kappa+j+k}(X_{\alpha'}) \quad \text{for } 1 \leq j \leq l. \quad (10.34)$$

Recall that $k = \text{rank}[C]$. These inequalities are best possible inferences from the data.

It is not necessary to know π in order to apply the theorem.

Proof. By Cauchy's theorem $\alpha_\kappa \leq \theta_\kappa$ and, by (10.31) and (10.32), $\theta_\kappa < \alpha' < \alpha_{\pi+1}$, which establishes (10.33). Now for the harder part. By (10.32) $\theta_\kappa < \alpha' < \theta_{\kappa+1}$ and so the residual interlace Theorem 10.5.1 may be invoked with $\zeta = \alpha'$ to deduce that $[\alpha', \mu_{\kappa+k+1}(X_{\alpha'})]$ contains at least one α and moreover $\alpha' = \mu_{\kappa+1}(X_{\alpha'}) = \cdots = \mu_{\kappa+k}(X_{\alpha'})$. By (10.31) $\alpha_{\pi+1}$ must be in that interval while $\alpha_{\pi+2}$ may or may not be included. Thus

$$\alpha_{\pi+1} \leq \mu_{\kappa+k+1}(X_{\alpha'})$$

which establishes the second inequality in (10.34) with $j = 1$.

To deal with the other values of j use either Lehmann's formulation Theorem 10.5.2 with suitable ξ or Exercise 10.5.2 to find a larger interval, $[\alpha', \mu_{\kappa+k+j}(X_{\alpha'})]$, containing at least j α 's. By (10.31) it must include $\alpha_{\pi+1}, \dots, \alpha_{\pi+j}$. This establishes the second inequality in (10.34) for all j up to $m - \kappa$, but only for $j \leq l$ will the first inequality also hold.

To establish the first inequality pick $\zeta = \alpha''$ and use Exercise 10.5.2 to conclude that $\mu_{\kappa+l}(X_{\alpha''}) = \dots = \mu_{\kappa+k+l}(X_{\alpha''}) = \alpha''$ and that $[\mu_{\kappa+j}(X_{\alpha''}), \alpha'']$ contains at least $l+1-j$ α 's for all j from l down to $-(\kappa-1)$. By (10.31) it must include $\alpha_{\pi+j}, \dots, \alpha_{\pi+l}$. The only values of j for which both inequalities hold is $1 \leq j \leq l$.

Equality in (10.34) holds on the left for $A = \text{diag}[M(X_{\alpha''}), \alpha'' + 1]$ and on the right for $A = \text{diag}[M(X_{\alpha''}), \alpha' - 1]$. \square

Exercise on Section 10.7

- 10.7.1. Suppose that $\|C\|$ is so small that $\alpha' + \|C\| \leq \theta_{q+1}$ in (10.32) of Theorem 10.7.1. Show that for $j = 1, \dots, l$,

$$\theta_{q+j} \leq \mu_{q+j+k}(X_{\alpha'}) \leq \theta_{q+j} + \|C\|^2 / (\theta_{q+j} - \alpha').$$

Note and References

In the 1930s Temple began to work on error bounds for approximate eigenpairs using residuals. Others who contributed to the subject, and extended it to differential operators, were Weinstein, Kato, and Wielandt. By bringing A 's spectrum into the picture, interesting a priori error bounds can be derived. A recent comprehensive account of work in this field is [Weinberger, 1974]. More recent still is [Chatelin and Lemordant, 1978].

This chapter singles out one theme: the exploitation of the residual matrix itself—not just its norm—and tries to present the ideas simply and yet completely. [Lehmann, 1949, 1963] was the first to obtain the optimal bounds which can be derived from the given information, but the approach taken in these unpublished notes of Kahan unifies all the results and has the advantage of being in English.

Well after the first edition of this book appeared the experts in eigenvalue calculations came to associate the importance of relative perturbation theory: bounds on the relative changes in nonzero eigenvalues caused by small relative changes in the parameters that define the matrix. The reader is referred to the textbook [Demmel, 1997, chapter 5] for an introduction and for references.

This page intentionally left blank

Approximations from a Subspace

11.1. Subspaces and Their Representation

Heavy use of the abstract notions of vector space and subspace in a discussion of numerical methods may seem to some readers unnecessarily abstruse. In fact, however, the language of subspaces simplifies such discussions by suppressing distracting details. This section reviews the way in which subspaces are handled and shows how the choosing of a basis corresponds to certain explicit matrix manipulations.

A subspace \mathcal{S} of \mathcal{E}^n is a subset which happens to be closed under the operation of taking linear combinations. A more useful definition is that \mathcal{S} is the totality of *all* linear combinations of some small set of vectors in \mathcal{S} . Any small set which generates \mathcal{S} in this way is called a *spanning set* and there are (infinitely) many spanning sets for each \mathcal{S} other than the trivial subspace $\{\mathbf{0}\}$. It is convenient to order the m vectors in a spanning set as columns of a matrix $\mathbf{S} = (\mathbf{s}_1, \dots, \mathbf{s}_m)$ and to say, briefly and improperly, that \mathbf{S} *spans* \mathcal{S} , although it is really the columns of \mathbf{S} that span \mathcal{S} . If one considers \mathbf{S} as a linear transformation, not just a collection of columns, then \mathcal{S} is called the *range* of \mathbf{S} or the *column space* of \mathbf{S} . $\mathbf{S}\mathbf{x}$ is a neat way to denote a linear combination of the columns of \mathbf{S} . There are several different ways of describing \mathcal{S} ,

$$\mathcal{S} = \text{range } \mathbf{S} = \{\mathbf{S}\mathbf{x} : \mathbf{x} \in \mathcal{E}^m\} = \mathbf{S}\mathcal{E}^m. \quad (11.1)$$

The introduction of \mathcal{E}^m in (11.1) is important; as \mathbf{x} ranges over *all* m -vectors, i.e., over *all* of \mathcal{E}^m , $\mathbf{S}\mathbf{x}$ ranges over *part* of \mathcal{E}^m , namely, \mathcal{S} (Exercise 11.1.2).

It pays to keep spanning sets as a small as possible. The minimal ones are called *bases* and all bases of \mathcal{S} have the same number of vectors in them. This number is \mathcal{S} 's dimension. From now on the dimension of a subspace will

be denoted automatically by a superscript; thus $\mathcal{S} \equiv \mathcal{S}^m$. One advantage of using an n -by- m matrix S for \mathcal{S}^m is that there is a one-to-one correspondence between the long n -vectors of \mathcal{S}^m and the short auxiliary vectors x of \mathcal{E}^m given by $s = Sx$. In other words, it is a basis that gives this advantage.

A need arises in section 11.4 to describe typical subspaces \mathcal{G}^j of the subspace \mathcal{S}^m of \mathcal{E}^n . How is this to be done?

Example 11.1.1. Let

$$\mathcal{S}^3 \equiv \text{range} \left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \right) \subset \mathcal{E}^4.$$

With respect to this basis the subspace

$$\mathcal{G}^2 \equiv \text{range} \left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \text{range} \left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \subset \mathcal{S}^3$$

corresponds to the subspace

$$\hat{\mathcal{G}}^2 \equiv \text{range} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \subset \mathcal{E}^3.$$

Note the difference between \mathcal{S}^3 and \mathcal{E}^3 . A different, more interesting, example shows that

$$\mathcal{F}^2 \equiv \left\{ \alpha \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \gamma \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} : \alpha + \beta + \gamma = 0 \right\} \subset \mathcal{S}^3$$

corresponds to the subspace

$$\hat{\mathcal{F}}^2 \equiv \left\{ \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} : \alpha + \beta + \gamma = 0 \right\} \subset \mathcal{E}^3.$$

And so it is in general. Once a basis S for \mathcal{S}^m is chosen, there is a natural correspondence between subspaces \mathcal{G}^j of the subspace \mathcal{S}^m and subspaces $\hat{\mathcal{G}}^j$ of \mathcal{E}^m given by $\mathcal{G}^j = S\hat{\mathcal{G}}^j = \{Su : u \in \hat{\mathcal{G}}^j\}$ (Exercise 11.1.4). Figure 11.1 illustrates the matter in another way,

$$\begin{array}{ccc} \mathcal{E}^n & & \\ \cup & & \\ S = S\mathcal{E}^m, & & S \text{ is } n \text{ by } m, \\ \cup \quad \cup & & \\ \boxed{\mathcal{G}^j = S\hat{\mathcal{G}}^j = SGS^j}, & & G \text{ is } m \text{ by } j. \end{array}$$

One-to-one correspondence
between subspaces \mathcal{G}^j of \mathcal{S}^m and $\hat{\mathcal{G}}^j$ of \mathcal{E}^m

FIG. 11.1. Subspaces of a subspace.

The subspace \mathcal{S}^m is in \mathcal{E}^n , not just \mathbb{R}^n , and so it is both meaningful and convenient to use orthonormal bases to describe it. See Exercise 11.1.5.

Exercises on Section 11.1

- 11.1.1. Let S be a nonempty subset of \mathcal{E}^n with the property that if u and v are in S then so is $\alpha u + \beta v$. Why must S also be the totality of all linear combinations of some finite subset of vectors in S ?

- 11.1.2. Let

$$S = \begin{bmatrix} 2 & 4 \\ 1 & 2 \\ 0 & 0 \end{bmatrix}$$

be a (wasteful) spanning set for S . Show that every vector s in S can be written as Su for some u in \mathcal{E}^2 . How many u for each s ?

- 11.1.3. Describe the second pair of subspaces \mathcal{F}^2 and $\hat{\mathcal{F}}^2$ in the example as the column spaces of certain matrices F and \hat{F} .
- 11.1.4. Let S be a basis for \mathcal{S}^m . Prove that the mapping between the subspaces of \mathcal{S}^m and those of \mathcal{E}^m induced by S is indeed one-to-one.
- 11.1.5. By using Gram–Schmidt, or otherwise, give orthonormal bases of the subspaces \mathcal{G}^2 and \mathcal{F}^2 given in section 11.1.

11.2. Invariant Subspaces

An eigenvector z of A may be normalized to have any convenient nonzero norm, and we usually say, somewhat loosely, that z , $2z$, and $-z$ are the same eigenvector. It is sometimes more convenient to speak of the subspace $\mathcal{S}^1 = \text{span}(z)$. This subspace of \mathcal{E}^n enjoys two remarkable properties:

- (1) \mathcal{S}^1 is mapped into itself by A , $A\mathcal{S}^1 \subset \mathcal{S}^1$. In fact $A\mathcal{S}^1 = \mathcal{S}^1$ if $\lambda \neq 0$.
- (2) The image under A of any z in \mathcal{S}^1 is simply a fixed multiple of z , $Az = z\lambda$ and λ depends on \mathcal{S}^1 alone, not z .

Any nonzero subspace obeying (2) is called an *eigenspace*.

Now let $Z = (z_1, \dots, z_m)$ be an n -by- m matrix whose columns are eigenvectors of A . Then $\text{range}(Z)$ enjoys (1) but not (2), unless $\lambda_1 = \lambda_2 = \dots = \lambda_m$. Subspaces of \mathcal{E}^n satisfying (1) are called *invariant*. Conversely, any invariant subspace has a basis of eigenvectors (Exercise 11.2.1). Eigenspaces are special invariant subspaces.

For any given n -by- m $F = (f_1, \dots, f_m)$ it is desirable to have a test for $\text{range}(F)$'s invariance. By (1), $Af_j = \sum f_i c_{ij}$ for each $j = 1, \dots, m$, for some unknown coefficients c_{ij} . These relations can be expressed more neatly in terms of F 's *residual matrix*

$$R \equiv \begin{array}{c|c} A & \\ \hline F & - F \end{array} = C = O. \quad (11.2)$$

When F has full rank m then (11.2) can be solved for a unique C given by $C = (F^*F)^{-1}F^*AF$. If $\text{rank}(F) < m$ then there are many solutions C of (11.2), but it is wasteful and usually unnecessary to work with such F 's.

When an orthonormal basis is used to represent a space then most calculations are simplified. For example, if Q satisfies $Q^*Q = I$ then $\text{range}(Q)$ is invariant if

$$R(Q) \equiv AQ - QH = O, \quad H \equiv Q^*AQ. \quad (11.3)$$

If $\text{range}(Q) = \text{range}(F)$ is invariant then both C and H represent the *restriction* of A to $\text{range}(F)$, but H has the advantage of being symmetric. Moreover

each eigenvector of C or of H determines an eigenvector of A .

If $Cy = y\lambda$ then Fy is an eigenvector of A with the same eigenvalue λ . If $Hx = x\lambda$ then Qx is an eigenvector of A with eigenvalue λ .

Example 11.2.1.

$$F = \begin{bmatrix} 3 & 0 \\ 1 & 1 \\ -1 & -1 \\ -3 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ -1 & -3 \\ -3 & 1 \end{bmatrix} \frac{1}{2\sqrt{5}},$$

$$A = \begin{bmatrix} 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & -2 \\ -2 & 0 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{bmatrix}.$$

The computation of C and y is left as an exercise, but

$$H = \frac{2}{5} \begin{bmatrix} 7 & 6 \\ 6 & -2 \end{bmatrix}, \quad x = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

Exercises on Section 11.2

- 11.2.1. Prove that if S is invariant under A then S has a basis of eigenvectors.
Use results from section 1.4.
- 11.2.2. Let $F = QL^*$ with Q orthonormal. Express C in terms of H and L^* .
- 11.2.3. In the example compute C and find its eigenvectors and the corresponding ones of A .

Warning. It is a common practice to say that a rectangular matrix Q , $n \times m$, $m \leq n$, satisfying $Q^*Q = I_m$ is *orthonormal* when it would be more precise to say that Q 's columns form an orthonormal set. This practice, which we follow, would say that a matrix G , $n \times m$, $m \leq n$ is *orthogonal* if G^*G is diagonal and invertible.

However, in matrix theory U is said to be orthogonal only when $U^*U = UU^* = I$; i.e., $U^* = U^{-1}$. This definition is ubiquitous in the mathematical world and is *not* consistent with practice among specialists in matrix computations. Nevertheless, after the warning is absorbed it is not difficult to live with the inconsistency.

11.3. The Rayleigh–Ritz Procedure

Usually the subspace \mathcal{S}^m on hand turns out not to be invariant under A . If it is nearly invariant then it should contain good approximations to some eigenvectors of A . Three ingredients are required for computing the best set of approximate eigenvectors from \mathcal{S}^m to eigenvectors of A :

1. S , an n -by- m full rank matrix whose columns are a basis for \mathcal{S}^m .
2. A subprogram, call it OP (for operator), which returns Ax for any given x .
3. Utility programs for orthonormalizing sets of vectors and computing eigensystems of m -by- m symmetric matrices.

There is no need for A to be known explicitly. In some, but not all, applications n is large ($n > 1000$) and $m \ll n$. Sometimes only a subset of p of the m approximate eigenpairs are wanted. The well-known Rayleigh–Ritz procedure (specified in Table 11.1) computes these approximations.

Exercises on Section 11.3

Apply the RR procedure in the following contexts:

- 11.3.1. The matrix A of section 11.2, $\mathcal{S} = \text{span}[(2 \ 1 \ -1 \ -2)^*, (2 \ -1 \ -1 \ 2)^*]$.
- 11.3.2. The matrix A of section 11.2, $\mathcal{S} = \text{span}[e_1, e_2, e_3]$.
- 11.3.3. The matrix A of section 11.5, $\mathcal{S} = \text{span}(Q)$.
- 11.3.4. Verify the operation counts in Table 11.1 giving, whenever possible, a second term in the expression. Suppose that $n = 10^3$, $m = 10^2$, $p = 10^1$. Calculate the OP ratios for each step.
- 11.3.5. How do the operation counts in steps 3 and 4 change if H happens to be tridiagonal?

11.4. Optimality

There are three (related) ways of justifying the claim that the RR approximations $\{\theta_i, y_i\}$ are optimal for the given information. The first is a natural corollary of the minmax characterization of eigenvalues. From section 10.2,

$$\alpha_j = \lambda_j[A] = \min_{\mathcal{F}^j \subset \mathcal{E}^n} \max_{f \in \mathcal{F}^j} \rho(f; A) \quad (f \neq 0). \quad (11.4)$$

TABLE 11.1
Procedure RR (Rayleigh-Ritz).

Action	Cost (in ops)
1. Orthonormalize the columns of S , if necessary, to get an orthonormal n -by- m Q written over S .	$m(m+1)n$
2. Form AQ by m calls to the subprogram OP . Often A is known only by the user's program and OP is not a simple matrix-vector product.	m calls on OP (See Chapter 15.)
3. Form m -by- m $H = \rho(Q) \equiv Q^*(AQ)$, the (matrix) Rayleigh quotient of Q .	$\frac{1}{2}m(m+1)n$
4. Compute the p ($\leq m$) eigenpairs of H which are of interest, say $Hg_i = g_i\theta_i$, $i = 1, \dots, p$. The θ_i are the <i>Ritz values</i> .	$\approx m^3$ (less if $p \ll m$)
5. If desired compute the p <i>Ritz vectors</i> $y_i = Qg_i$, $i = 1, \dots, p$. The full set $\{(\theta_i, y_i)\}$, $i = 1, \dots, m$ is the best set of approximations to eigenpairs of A which can be derived from S^m alone.	pmn
6. Residual error bounds. Form the p residual vectors $r_i = r(y_i) = Ay_i - y_i\theta_i = (AQ)g_i - y_i\theta_i$ using the last expression for computation. Also compute $\ r_i\ $. Each <i>Ritz interval</i> $[\theta_i - \ r_i\ , \theta_i + \ r_i\]$ contains an eigenvalue of A . If some of the intervals overlap, then a bit more work is required to guarantee approximations to p eigenvalues of A . See section 11.5.	$p(m+2)n$
7. A nontraditional extra step is described in section 11.8.	$\frac{1}{2}p(p+1)n + O(p^3)$

Recall that dimensions of spaces are denoted by superscripts and $\rho(f; A) = f^* Af / f^* f$ for $f \neq 0$. Consequently the natural definition of the best approximation β_j to α_j from the given subspace S^m is to replace E^n by S^m in (11.4) to get

$$\beta_j \equiv \min_{G^j \subset S^m} \max_{g \in G^j} \rho(g; A) \quad (g \neq 0). \quad (11.5)$$

The only difficulty in the proof of the theorem below is characterizing the subspaces G^j of the subspace S^m . If Q is a fixed orthonormal basis in S^m then $S^m = QE^m \equiv \{Qs : s \in E^m\}$. The key point is that the subspaces of S^m are generated by the subspaces of E^m from the same correspondence. This was established in section 11.1; namely,

$$G^j \subset S^m \text{ if and only if } G^j = Q\hat{G}^j \text{ and } \hat{G}^j \subset E^m. \quad (11.6)$$

Theorem 11.4.1. *Let Q be any orthonormal basis for S^m . With β_j defined in (11.5)*

$$\beta_j = \lambda_j [Q^* A Q], \quad j = 1, \dots, m.$$

Proof. Applying (11.6) and $Q^* Q = I_m$ it follows that

$$\begin{aligned} \beta_j &= \min_{G^j \subset S^m} \max_{g \in G^j} \rho(g; A), \text{ and } 0 \neq g = Qs, \\ &= \min_{\hat{G}^j \subset E^m} \max_{s \in \hat{G}^j} \rho(s; H), \text{ since } G^j = Q\hat{G}^j \text{ and } s^* s = g^* g \neq 0, \\ &= \lambda_j [H] \equiv \theta_j. \quad \square \end{aligned}$$

The optimality result given above concerns only the θ 's.

The second way in which the approximations are optimal concerns Q . For any m -by- m matrix B , a residual matrix $R(B) \equiv A - QB$ is associated. The minimizing property of the familiar Rayleigh quotient $\rho(q)$ is inherited by the matrix $H = Q^* A Q \equiv \rho(Q)$. Recall that we favor the spectral norm $\|S\| = [\lambda_{\max}(S^* S)]^{1/2}$.

Theorem 11.4.2. *For given orthonormal n -by- m Q and $H \equiv Q^*AQ$,*

$$\|R(H)\| \leq \|R(B)\| \quad \text{for all } m\text{-by-}m B.$$

Proof. $R(B)^*R(B) = Q^*A^2Q - B^*(Q^*AQ) - (Q^*AQ)B + B^*B$. The trick is to see that the last three terms are just $(H - B)^*(H - B) - H^2$ and so

$$\begin{aligned} R(B)^*R(B) &= Q^*A^2Q - H^2 + (H - B)^*(H - B) \\ &= R(H)^*R(H) + (H - B)^*(H - B). \end{aligned}$$

Since $(H - B)^*(H - B)$ is positive semidefinite, Corollary 10.3.1 shows that

$$\|R(B)\|^2 = \lambda_{-1}[R(B)^*R(B)] \geq \lambda_{-1}[R(H)^*R(H)] = \|R(H)\|^2. \quad \square$$

For uniqueness of minimizing B see Exercises 11.4.4 and 11.4.5.

Now let S be any orthonormal basis in \mathcal{S}^m and let Δ be any diagonal matrix. Thus the pairs $\{(\delta_i, s_i), i = 1, \dots, m\}$ are rival eigenpair approximations to $\{(\theta_i, y_i)\}$. However, from Theorem 11.4.2 $\|AS - S\Delta\|$ is minimized over S and Δ when $s_i = y_i$, $\delta_i = \theta_i$, $i = 1, \dots, m$. The verification of this is left as an important exercise (Exercise 11.4.1). A related and useful characterization of the y_i is given in Exercise 11.4.6. By Exercise 11.4.4 the RR approximations are the unique minimizers of $\|AS - S\Delta\|_F$.

A third way in which the RR approximations are optimal is in the spirit of backward error analysis (see Chapter 2). Since \mathcal{S}^m is not invariant it is meaningless to speak of the restriction of A to \mathcal{S}^m . The next best thing is A 's *projection* onto \mathcal{S}^m . See section 1.4 for a discussion of projections. If P_S is the orthogonal projector onto \mathcal{S}^m , i.e., $P_S g$ is the closest vector in \mathcal{S}^m to g , then \mathcal{S}^m is invariant under $P_S A$ and so it is meaningful to speak of its restriction to \mathcal{S}^m , namely, $P_S A|_S$, and this is the desired projection. It is intimately related to the matrix $P_S A P_S$ which acts on the whole of \mathcal{E}^n and is also called A 's projection. No harm results from this ambiguity because the two projections

have the same action on \mathcal{S}^m . Next comes the third characterization of the Ritz pairs.

Lemma 11.4.1. *The (θ_i, y_i) , $i = 1, \dots, m$ are the eigenpairs for \mathbf{A} 's projection onto \mathcal{S}^m .*

The proof constitutes Exercise 11.4.2.

Example 11.4.1. If \mathbf{A} rotates the plane \mathcal{E}^2 through 45^0 counterclockwise, then \mathbf{A} 's projection onto any \mathcal{S}^1 simply shrinks \mathcal{S}^1 by a factor $\sqrt{2}$ as shown in Figure 11.2.

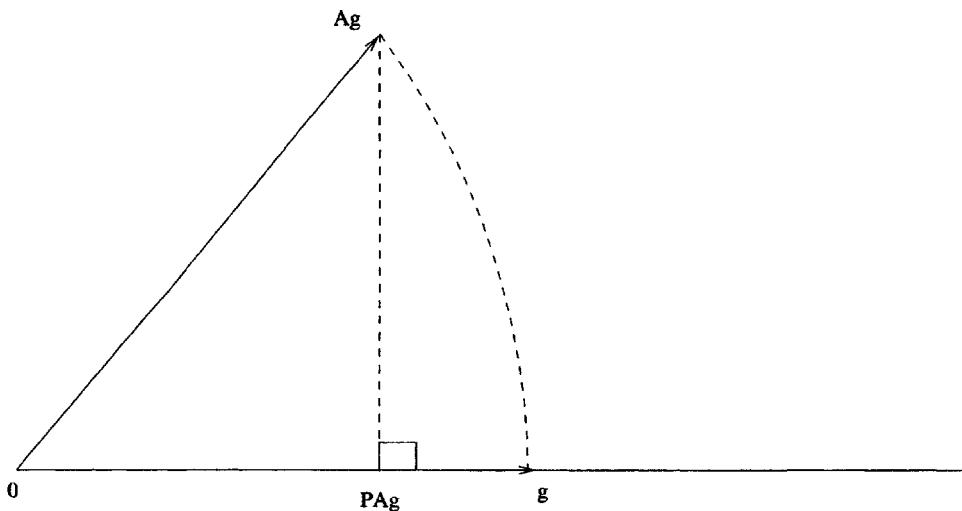


FIG. 11.2. \mathbf{A} 's projection on the one-dimensional subspace.

There are two ways in which the RR approximations are not optimal. In general no Ritz vector y_j is the closest unit vector in \mathcal{S}^m to any eigenvector of \mathbf{A} . Even more surprising perhaps is that the error bound $\|\mathbf{Av} - v\rho(v)\|/\|v\|$ is not minimized over \mathcal{S}^m by any of the Ritz vectors when $m > 1$. Example 11.5.1 illustrates both assertions. To summarize, by achieving *collective optimality* for all m pairs the RR approximations usually relinquish optimality for any particular eigenpair.

There is little profit in approximations which are good but not known to be good. The rest of this chapter focuses on the assessment of the RR approximations.

Exercises on Section 11.4

- 11.4.1. Use Theorem 11.4.2 to show that for any orthonormal basis S of \mathcal{S}^m , $\min \|AS - S\Delta\|$ over diagonal Δ and orthonormal S is achieved when $s_i = y_i$, $\delta_i = \theta_i$.
- 11.4.2. Verify that each pair (θ_i, y_i) is an eigenpair $P_S A|_S$.
- 11.4.3. Show that \mathcal{S} is also invariant under $\bar{A} \equiv A - R(H)Q^* - QR(H)^*$. Verify that $\|\bar{A} - A\| = \|R(H)\|$. Are the (θ_i, y_i) eigenpairs of \bar{A} ? Does $\bar{A} = P_S A P_S$?
- 11.4.4. Define $\|\cdot\|_F$ by $\|B\|_F^2 = \text{trace}(B^*B)$. With respect to Theorem 11.4.2 show that $\|R(B)\|_F \geq \|R(H)\|_F$ with equality only when $B = H$.
- 11.4.5. Find A, Q , and a 2-by-2 matrix $B \neq H$ such that $\|R(B)\| = \|R(H)\|$.
- 11.4.6. Show that, for x in \mathcal{S}^m ,

$$Ax - x\rho(x) \perp \mathcal{S}^m \text{ if and only if } x = y_i \text{ for some } i \leq m.$$

11.5. Residual Bounds on Clustered Ritz Values

At the completion of step 6 in the RR procedure there are on hand $\theta_i, y_i, r_i, \|r_i\|$, for $i = 1, \dots, p$. The simple error bounds in section 4.5 guarantee an eigenvalue α of A in each interval $[\theta_i - \|r_i\|, \theta_i + \|r_i\|]$. If the intervals are disjoint then the θ_i provide approximations to p different eigenvalues of A , as desired. Better bounds require either more knowledge of A or more work as described in section 11.8 and Chapter 10.

If two or more intervals overlap then it is possible that two or more different θ_i are approximating a single α as Example 11.5.1 and Figure 11.3 show.

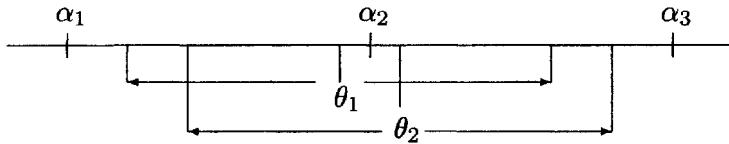
Example 11.5.1.

$$A = \begin{bmatrix} 0 & \gamma & 0 \\ \gamma & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

$$\text{Take } \gamma \approx 0.1, \quad \alpha_{\pm 1} = \mp \sqrt{1 + \gamma^2}, \quad \alpha_2 = 0.$$

Then compute H and

$$\theta_{\pm 1} = \mp \gamma, \quad y_1^* = (1, -1, 0)/\sqrt{2}, \quad y_2^* = (1, 1, 0)/\sqrt{2}.$$

FIG. 11.3. *Clustered Ritz values.*

$$R(Y) = (\mathbf{r}_1, \mathbf{r}_2) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 1 \end{bmatrix} \frac{1}{\sqrt{2}}, \quad \|\mathbf{r}_1\| = \|\mathbf{r}_2\| = \frac{1}{\sqrt{2}},$$

$$\|R(Y)\| = 1, \quad \mathbf{r}_i = \mathbf{A}\mathbf{y}_i - \mathbf{y}_i\theta_i.$$

Note that θ_1 and θ_2 provide spurious evidence that there are two α 's in the interval $[\frac{-1}{\sqrt{2}} - \gamma, \frac{1}{\sqrt{2}} + \gamma]$. This concludes Example 11.5.1.

The way out of the overlap difficulty is to use the norm of the associated residual matrix \mathbf{R} .

Theorem 11.5.1. *Let \mathbf{Q} be any orthonormal n -by- m matrix. Associated with it are $\mathbf{H} (\equiv \mathbf{Q}^* \mathbf{A} \mathbf{Q})$ and $\mathbf{R} (\equiv \mathbf{A} \mathbf{Q} - \mathbf{Q} \mathbf{H})$. There are m of \mathbf{A} 's eigenvalues $\{\alpha_j, j = 1, \dots, m\}$ which can be put in one-to-one correspondence with the eigenvalues θ_j of \mathbf{H} in such a way that*

$$|\theta_j - \alpha_j| \leq \|\mathbf{R}\|, \quad j = 1, \dots, m.$$

Proof. Orthonormal columns can be appended to \mathbf{Q} to fill out a square orthonormal matrix $\mathbf{P} \equiv (\mathbf{Q}, \tilde{\mathbf{Q}})$. Then

$$\mathbf{P}^* \mathbf{A} \mathbf{P} \equiv \begin{bmatrix} \mathbf{Q}^* \mathbf{A} \mathbf{Q} & \mathbf{Q}^* \mathbf{A} \tilde{\mathbf{Q}} \\ \tilde{\mathbf{Q}} \mathbf{A} \mathbf{Q} & \tilde{\mathbf{Q}} \mathbf{A} \tilde{\mathbf{Q}} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{H} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{W} \end{bmatrix},$$

where only \mathbf{H} will be known explicitly. Yet

$$\begin{aligned} \mathbf{P}^* \mathbf{R} &= \mathbf{P}^* \mathbf{A} \mathbf{Q} - \mathbf{P}^* \mathbf{Q} \mathbf{H} \\ &= (\mathbf{P}^* \mathbf{A} \mathbf{P})(\mathbf{P}^* \mathbf{Q}) - (\mathbf{P}^* \mathbf{Q}) \mathbf{H} \\ &= \begin{bmatrix} \mathbf{H} & \mathbf{B}^* \\ \mathbf{B} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{O} \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \mathbf{O} \end{bmatrix} \mathbf{H} = \begin{bmatrix} \mathbf{O} \\ \mathbf{B} \end{bmatrix}. \end{aligned} \quad (11.7)$$

By the orthogonal invariance of $\|\cdot\|$ (Fact 1.10 in section 1.6)

$$\|R\| = \|P^*R\| = \left\| \begin{bmatrix} O \\ B \end{bmatrix} \right\| = \|B\|.$$

Now

$$P^*AP = \begin{bmatrix} H & O^* \\ O & W \end{bmatrix} + \begin{bmatrix} O & B^* \\ B & O \end{bmatrix} \quad (11.8)$$

and by the Weyl monotonicity theorem (section 10.3) for $i = 1, \dots, n$

$$\alpha_i = \lambda_i[P^*AP] \leq \lambda_i \left[\begin{pmatrix} H & O^* \\ O & W \end{pmatrix} \right] + \lambda_{-1} \left[\begin{pmatrix} O & B^* \\ B & O \end{pmatrix} \right]. \quad (11.9)$$

Now the θ_j appear somewhere in the ordered list of eigenvalues of H and W . So there exist indices j' such that

$$\lambda_{j'} \left[\begin{pmatrix} H & O^* \\ O & W \end{pmatrix} \right] = \theta_j, \quad j = 1, \dots, m.$$

To evaluate the second term on the right in (11.9), square the matrix

$$\begin{bmatrix} O & B^* \\ B & O \end{bmatrix}^2 = \begin{bmatrix} B^*B & O^* \\ O & BB^* \end{bmatrix}. \quad (11.10)$$

Since B^*B and BB^* have the same nonzero eigenvalues (Exercise 11.5.2),

$$\lambda_{-1} \left[\begin{pmatrix} O & B^* \\ B & O \end{pmatrix} \right] = \sqrt{\lambda_{-1}[B^*B]} = \sqrt{\|B\|^2} = \|R\|. \quad (11.11)$$

With $i = j'$ in (11.9) the theorem's inequality is obtained. \square

When $m > 2$, $\|R\|$ has a small but nonnegligible cost. It can be majorized via $\|R\|^2 \leq \|R\|_F^2 = \sum_{i=1}^m \|r_i\|^2$, and the $\|r_i\|^2$ are already available from step 6 in the RR procedure. If $\|R\|_F$ is to be used there is a corresponding result whose proof constitutes Exercise 11.5.3.

Theorem 11.5.2. *There are m indices j' , $j = 1, 2, \dots, m$ such that*

$$\sum_{i=1}^m (\theta_j - \alpha_{j'})^2 \leq 2\|R\|_F^2.$$

In practice Theorem 11.5.1 will usually be applied not to $R(Q)$ but to a matrix $\bar{R} \equiv (r_1, \dots, r_\nu)$ which corresponds to a subset of θ 's whose residual intervals overlap. In the proof Q is replaced by $\bar{Q} = (y_1, \dots, y_\nu)$ and H by $\bar{H} = \text{diag}(\theta_1, \dots, \theta_\nu)$. Then it follows that each interval $[\theta_i - \|\bar{R}\|, \theta_i + \|\bar{R}\|]$ contains its own α for these clustered θ_i , $i = 1, \dots, \nu$.

Exercises on Section 11.5

- 11.5.1. Let $H = \text{diag}(\theta_1, \dots, \theta_4)$, $C = \text{diag}(\gamma_1, \dots, \gamma_4)$, and let $Q = (e_1, \dots, e_4)$. Apply Theorem 11.5.1 to $A = \begin{bmatrix} H & C^* \\ C & H \end{bmatrix}$. What does this example show about possible strengthening of the results?
- 11.5.2. There are several ways to prove that B^*B and BB^* have the same nonzero eigenvalues. Show that $\begin{bmatrix} B^*B & 0 \\ B & 0 \end{bmatrix}$ and $\begin{bmatrix} 0 & 0 \\ B & BB^* \end{bmatrix}$ are similar.
- 11.5.3. Apply the Wielandt–Hoffman theorem, Fact 1.11 in section 1.6, to (11.8) to prove Theorem 11.5.2.
- 11.5.4. Use the example in section 11.2 to evaluate $R(Q)$. How much worse is $\|R(Q)\|$ than either of the residual vector bounds?
- 11.5.5. Apply Theorem 11.5.2 to a tridiagonal matrix taking $Q = (e_1, \dots, e_m)$. Which theorem in Chapter 10 gives the same result?
- 11.5.6. For any n -by- m F of full rank $m \ll n$ define the “block reflector that reverses F ” by

$$H(F) \equiv I - 2F(F^*F)^{-1}F^*.$$

Verify that suitable choices for the matrix P in (11.7) are $H_{\pm} \equiv H(Q \pm E_1)$ where n -by- m $E_1 \equiv (e_1, e_2, \dots, e_m)$. Partition Q and R as $(\begin{smallmatrix} Q_{11} \\ Q_{21} \end{smallmatrix})$ and $(\begin{smallmatrix} R_{11} \\ R_{21} \end{smallmatrix})$ where Q_{11} and R_{11} are m -by- m . Verify that

$$B = R_{21} - Q_{21}(I_m + \text{sym } Q_{11})^{-1}R_{11}$$

where $\text{sym } J \equiv (J + J^*)/2$. Finally C is computed from B by the methods in sections 6.7.2 or 6.8.

11.6. No Residual Bounds on Ritz Vectors

How well does a Ritz vector y_i approximate some eigenvector z_i ? The following 2-by-2 example shows that without adscititious information (i.e., not readily computable information) no bound can be placed on the error in the Ritz vectors. Consider

$$A = \begin{bmatrix} \nu + \delta & \gamma \\ \gamma & \nu - \delta \end{bmatrix}, \quad \alpha_1 = \nu - \sigma, \quad \alpha_2 = \nu + \sigma, \quad \text{where } \sigma^2 = \delta^2 + \gamma^2.$$

Take $Q = e_1$, so that $H = \theta = \nu + \delta$. Then

$$|\theta - \alpha_1'| = \sigma - |\delta| = \gamma^2 / (\sigma + |\delta|) \leq |\gamma| = \|R\|,$$

where $1' = 2$ if $\delta \geq 0$, $1' = 1$ if $\delta \leq 0$.

The “true” eigenvector and the Ritz vector are, respectively,

$$\begin{bmatrix} 1 \\ \pm\gamma/(\sigma + |\delta|) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

If ϕ is the acute angle between them (the error angle) then

$$\cos \phi = (1 + \gamma^2 / (\sigma + |\delta|)^2)^{-1/2} \longrightarrow \begin{cases} 1 & \text{if } \delta \neq 0 \\ 1/\sqrt{2} & \text{if } \delta = 0 \end{cases} \text{ as } \gamma \rightarrow 0.$$

So when $\delta = 0$ the “true” eigenvector is $(1, \pm 1)^*$ for all nonzero γ and the error angle is $\pi/4$. That is why there is no bound, in terms of $\|R\|$, on the error in the Ritz vector.

What has gone wrong? Why does the method seem to break down when $|\delta|$ is small? *The answer is that the method does not break down, but the question does.* As $\delta \rightarrow 0$ the request for the error $\|e_1 - z_1\|$ in the Ritz vector becomes sillier and sillier—a perfect example of the danger of a purely formal approach.

When $|\delta/\theta|$ is very small then θ is almost as good an approximation to α_1 as to α_2 . Which of the two eigenvectors should the Ritz vector approximate? Since they are mutually orthogonal no single vector can be close to both and, with wisdom no less than Solomon’s, the Ritz vector splits the difference between the two rivals: in the limit, as $\delta \rightarrow 0$

$$e_1 = \frac{1}{2} \left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right] = \lim(z_1 + z_2)/\sqrt{2}.$$

This example reflects the general situation in which the eigenvectors are not necessarily continuous functions of the matrix elements in the neighborhood of matrices with multiple eigenvalues; see section 1.4. In such cases the useful objects are the invariant subspaces associated with each cluster of very close eigenvalues.

The user’s difficulty has changed but it is still there. Instead of asking whether a Ritz vector is good or bad he or she may ask whether there are several of A’s eigenvalues close to one of the computed θ ’s. The Ritz method cannot answer this question.

When extra information, beyond R, is available then it can often be used to give an error bound. In its absence we cannot do better than Theorem 11.5.1.

That admirable scholar N. J. Higham found out that the word adscititious in line 2 of this section was dropped from the *Concise Oxford Dictionary* in 1990.

11.7. Gaps in the Spectrum

Consider now any unit vector y and its Rayleigh quotient $\theta = y^*Ay$. The simple error bound in section 4.5 guarantees that there is at least one eigenvalue α of A satisfying

$$|\alpha - \theta| \leq \|Ay - y\theta\| = \|r(y)\|.$$

However, if θ is known to be well separated from all eigenvalues other than the closest one α , then not only may the error bound on θ be improved but upper and lower bounds can be put on the accuracy with which y approximates α 's eigenvector z . To this end we introduce two useful quantities:

$$\begin{aligned} \text{gap} &= \text{gap}(\theta) := \min |\lambda_i[A] - \theta| \text{ over all } \lambda_i \neq \alpha, \\ \text{spread} &= \text{spread}(A) := \lambda_{-1}[A] - \lambda_1[A]. \end{aligned}$$

Theorem 11.7.1. *Let y be a unit vector with $\theta = y^*Ay$ and residual $r(y) = Ay - y\theta$. Let α be the eigenvalue of A closest to θ , let z be its normalized eigenvector, and let $\psi := \angle(y, z)$. Then*

$$\frac{\|r(y)\|}{\text{spread}} \leq |\sin \psi| \leq \frac{\|r(y)\|}{\text{gap}} \quad (11.12)$$

and

$$0 \leq |\theta - \alpha| \leq \frac{\|r(y)\|^2}{\text{gap}}. \quad (11.13)$$

The upper bound in (11.12) holds for any value of θ .

The proof is quite subtle; its origins are hard to trace but are discussed in [Davis and Kahan, 1970].

Proof. If $y = \pm z$ there is nothing to prove, otherwise decompose y in the form $y = z \cos \psi + w \sin \psi$, where w is the unit vector in the y - z plane

orthogonal to \mathbf{z} . Hence, for any value of θ ,

$$\mathbf{r}(\mathbf{y}) = (\mathbf{A} - \theta)\mathbf{y} \quad (11.14)$$

$$= \mathbf{z}(\alpha - \theta) \cos \psi + (\mathbf{A} - \theta)\mathbf{w} \sin \psi \quad (11.15)$$

since $\mathbf{A}\mathbf{z} = \mathbf{z}\alpha$. Happily, $\mathbf{z}^*(\mathbf{A} - \theta)\mathbf{w} = (\alpha - \theta)\mathbf{z}^*\mathbf{w} = 0$ and so, by Pythagoras,

$$\|\mathbf{r}(\mathbf{y})\|^2 = (\alpha - \theta)^2 \cos^2 \psi + \|(\mathbf{A} - \theta)\mathbf{w}\|^2 \sin^2 \psi. \quad (11.16)$$

This yields the upper bound on $\sin \psi$ since $\|(\mathbf{A} - \theta)\mathbf{w}\| \geq \text{gap}$. See Exercise 11.7.1. The other results require that $\theta = \rho(\mathbf{y})$ and hence $\mathbf{r}(\mathbf{y}) \perp \mathbf{z}$ (see section 1.5);

$$0 = \mathbf{y}^*\mathbf{r}(\mathbf{y}) = (\alpha - \theta) \cos^2 \psi + \mathbf{w}^*(\mathbf{A} - \theta)\mathbf{w} \sin^2 \psi. \quad (11.17)$$

Thus $\cos^2 \psi$ and $\sin^2 \psi$ are in the ratio $\mathbf{w}^*(\mathbf{A} - \theta)\mathbf{w} : \theta - \alpha$ and so

$$\sin^2 \psi = \frac{\theta - \alpha}{\mathbf{w}^*(\mathbf{A} - \theta)\mathbf{w} + (\theta - \alpha)} = \frac{\theta - \alpha}{\mathbf{w}^*(\mathbf{A} - \alpha)\mathbf{w}}.$$

Eliminate $\cos^2 \psi$ from (11.16) and (11.17) and simplify to find

$$\|\mathbf{r}(\mathbf{y})\|^2 = \mathbf{w}^*(\mathbf{A} - \theta)(\mathbf{A} - \alpha)\mathbf{w} \sin^2 \psi. \quad (11.18)$$

The Cauchy–Schwartz inequality gives the lower bound

$$\begin{aligned} \|\mathbf{r}(\mathbf{y})\|^2 &\leq \|(\mathbf{A} - \theta)\mathbf{w}\| \cdot \|(\mathbf{A} - \alpha)\mathbf{w}\| \sin^2 \psi \\ &\leq \text{spread}^2 \sin^2 \psi, \end{aligned}$$

and the formula for $\sin^2 \psi$ gives

$$\|\mathbf{r}(\mathbf{y})\|^2 = \mathbf{w}^*(\mathbf{A} - \theta)(\mathbf{A} - \alpha)\mathbf{w} \cdot \frac{\theta - \alpha}{\mathbf{w}^*(\mathbf{A} - \alpha)\mathbf{w}}. \quad (11.19)$$

To obtain the bound on $|\theta - \alpha|$ one must use the assumption that there is no eigenvalue separating θ and α . Consequently $(\mathbf{A} - \theta)(\mathbf{A} - \alpha)$ is positive definite and, writing $\mathbf{w} = \sum \mathbf{z}_i \xi_i$ over $\mathbf{z}_i \neq \mathbf{z}$, one obtains the crucial equality

$$\mathbf{w}^*(\mathbf{A} - \theta)(\mathbf{A} - \alpha)\mathbf{w} = \sum |\alpha_i - \alpha| \cdot |\alpha_i - \theta| \xi_i^2. \quad (11.20)$$

The sum implied by \sum is over the spectrum of \mathbf{A} complementary to our chosen α closest to θ . By definition of gap ,

$$\begin{aligned} \mathbf{w}^*(\mathbf{A} - \theta)(\mathbf{A} - \alpha)\mathbf{w} &\geq \text{gap} \sum |\alpha_i - \alpha| \xi_i^2 \\ &\geq \text{gap} \left| \sum (\alpha_i - \alpha) \xi_i^2 \right| \\ &= \text{gap} |\mathbf{w}^*(\mathbf{A} - \alpha)\mathbf{w}|. \end{aligned} \quad (11.21)$$

Substitute inequality (11.21) into (11.19) and the proof is complete. \square

Corollary 11.7.1. *When $\theta = \rho(y)$ is closest to α_1 or to α_{-1} then*

$$\tan \psi \leq \frac{\|r(y)\|}{\text{gap}} \quad \text{and} \quad \frac{\|r(y)\|^2}{\text{spread}} \leq |\theta - \alpha|. \quad (11.22)$$

Proof. In these two cases $\text{sign}(\alpha_i - \mu)$ is constant for any $\mu \in [\theta, \alpha]$ with $\alpha = \alpha_1$ or α_{-1} . So

$$|w^*(A - \mu)w| = \sum |\alpha_i - \mu|\xi_i^2, \quad (11.23)$$

where $w = \sum z_i \xi_i$, as given in (11.20). With $\mu = \theta$ and using the definition of gap, (11.23) yields

$$|w^*(A - \theta)w| \geq \text{gap}. \quad (11.24)$$

Next observe that $|\alpha_i - \theta| \leq |\alpha_i - \alpha| \leq \text{spread}$, so (11.20) together with (11.23) with $\mu = \alpha$ yields

$$w^*(A - \theta)(A - \alpha)w \leq \text{spread}|w^*(A - \alpha)w|. \quad (11.25)$$

Now (11.17) gives

$$\tan^2 \psi = \frac{\theta - \alpha}{w^*(A - \theta)w}$$

and then apply (11.13) and (11.24) to find

$$\tan^2 \psi \leq \left(\frac{\|r(y)\|^2}{\text{gap}} \right) / \text{gap}$$

which delivers the first inequality. The second result appears when (11.25) is applied to (11.19); namely,

$$\|r(y)\|^2 \leq \frac{\text{spread} |w^*(A - \alpha)w| |\theta - \alpha|}{|w^*(A - \alpha)w|}. \quad \square$$

In many applications, but not all, gap is unknown and the bounds in the gap theorem are theoretical. In some circumstances, in the Lanczos algorithm, for instance, there comes a time when it is very unlikely that there remain undetected α 's hidden among the known α 's. In such circumstances the gap for θ_j can be replaced by the computable quantity $\min_{\theta_i \neq \theta_j} (|\theta_i - \theta_j| - \|r_j\|)$ and the bounds become good estimates.

Theorem 11.7.1 is very satisfactory for pinning down isolated eigenvalues. It is of no use when a cluster of close α 's is approximated by a cluster of θ 's with overlapping interval bounds. Theorem 11.7.1 suggests, and experience corroborates, that the Ritz vectors for such close θ 's are sometimes poor eigenvector approximations. To be specific, suppose that y_1, y_2, y_3 are three such Ritz vectors. It turns out that if the cluster of θ 's is well separated from all the α 's not in the cluster then $\text{span}(y_1, y_2, y_3)$ is a much better approximation to the associated invariant subspace, Z^3 , say, than is any of the y 's as an individual eigenvector approximation. In other words, the mismatch between the bases does not prevent the two subspaces from being close. What is needed is a measure of the closeness of two subspaces.

11.7.1. Gap Theorems for Subspaces

Our physical confinement to three-dimensional space gives us no intuitive feeling for the way that even a pair of planes can be related in E^4 , let alone a pair of p -dimensional subspaces nestling in E^n . A full treatment of this topic is beyond the scope of this book. An excellent summary of the material is [Davis and Kahan, 1969] but the proofs and the Hilbert space setting are to be found in the formidable but important paper [Davis and Kahan, 1970]. What follows is a summary of a part of that work.

If f and g are unit vectors in E^n then

$$\angle(\text{span } f, \text{span } g) = \cos^{-1} |f^* g| = \arccos |f^* g|.$$

Now let F and G be orthonormal n -by- p matrices and let $\mathcal{F}^p = \text{range}(F)$, $\mathcal{G}^p = \text{range}(G)$. It turns out that the proper measure of the closeness of \mathcal{F}^p and \mathcal{G}^p is a set of p numbers called the *canonical angles* between \mathcal{F}^p and \mathcal{G}^p . However, we shall only need the smallest of them and will take that as the angle between the spaces

$$\angle(\mathcal{F}^p, \mathcal{G}^p) \equiv \arccos \|F^* G\|. \quad (11.26)$$

Note that $0 \leq \|F^* G\| \leq 1$, Exercise 11.7.6 and, further, Exercise 11.7.7,

$$\angle(\mathcal{F}^p, \mathcal{G}^p) = \max_{f \in \mathcal{F}^p} \min_{g \in \mathcal{G}^p} \angle(f, g). \quad (11.27)$$

The next task is to define the gap. By Theorem 11.5.1 there are p eigenvalues α_i 's of a given A which can be paired with the eigenvalues θ_i of $\rho(F) = F^*AF$ so that $|\alpha_i - \theta_i| \leq \|AF - F\rho(F)\|$, $i = 1, \dots, p$. If there are more than p α 's which satisfy the inequality then any selection of p of them will do. The remaining $n - p$ α 's constitute the spectrum of A *complementary* to the spectrum of $\rho(F)$. Let the indices of these complementary α 's form the set \mathcal{J} . The gap between the spectrum of $\rho(F)$ and the complementary spectrum of A is defined by

$$\text{gap} \equiv \min\{|\theta_i - \alpha_j| : 1 \leq i \leq p, j \in \mathcal{J}\}. \quad (11.28)$$

The object of all this preparation is to compare \mathcal{F}^p with the invariant subspace \mathcal{Z}^p belonging to the p α 's paired with the θ 's. Davis and Kahan give the following elegant generalization of Theorem 11.7.1.

Theorem 11.7.2. *With the notation developed above,*

$$\text{gap} \cdot \sin \angle(\mathcal{F}^p, \mathcal{Z}^p) \leq \|AF - F\rho(F)\|. \quad (11.29)$$

One important application replaces F by the set of Ritz vectors associated with a cluster of close Ritz values θ_i and $\rho(F)$ by $\text{diag}(\theta_1, \dots, \theta_p)$. When there are more than p eigenvalues α bunched close together then the bound yields very little; otherwise it is very satisfactory.

Exercises on Section 11.7

- 11.7.1. Obtain a lower bound for $\|(A - \theta)w\|^2$, discard some information in (11.16), and deduce Theorem 11.7.1's first inequality.
- 11.7.2. Show that when $\theta = \rho(y)$, then $\sin^2 \psi = (\theta - \alpha)/w^*(A - \alpha)w$.
- 11.7.3. Derive (11.19) using the fact that $w^*w = 1$.
- 11.7.4. Show that $|w^*(A - \alpha)w| \geq \text{gap}$ when θ is closest to the greatest eigenvalue $\alpha_{-1} = \|A\|$.
- 11.7.5. Does Theorem 11.7.1 continue to hold if α is a multiple eigenvalue but gap is the distance from θ to eigenvalues other than α ? If not, where does the proof break down?

11.7.6. F and G are n by p and orthonormal. Show that $\|F^*G\| \leq 1$.

Hint: $\|C\| = \max_{u,v} |u^* Cv| / (\|u\| \cdot \|v\|)$. When can equality occur?

11.7.7. Prove (11.27).

11.8. Condensing the Residual

Step 6 in the RR procedure produced the p columns of the matrix $R(Y) = AY - Y\Theta$ where Θ is the diagonal matrix of Ritz values and $Y = (y_1, \dots, y_p)$ contains the Ritz vectors. The previous error bounds involved $\|\mathbf{r}_i\|$, $i = 1, \dots, p$, or possibly $\|R(Y)\|$ if the θ 's were tightly bunched. Improved bounds can be obtained by using the matrix $R(Y)$ itself, not just its norm.

If the Gram-Schmidt process is applied to $R(Y)$ it produces n -by- p orthonormal S and p -by- p upper triangular C such that $R(Y) = SC$. We point out that $R(Y)^*R(Y) = C^*S^*SC = C^*C$, so C is the Cholesky factor of the p -by- p matrix $R(Y)^*R(Y)$. It is C which provides the extra information needed to improve the previous bounds. In practice information is lost when forming $R(Y)^*R(Y)$ explicitly and it is preferable to reduce $R(Y)$ to C by premultiplying $R(Y)$ by well chosen orthogonal matrices, either plane rotations or reflections. To this end the RR procedure should be supplemented with an extra step.

Step 7 of RR: reduce $R(Y)$ to its upper-triangular factor C . The optimal bounds developed in sections 10.6, 10.7, and 10.8 can now be applied with Θ in place of H . The cost of computing C is $p(p+1)n$ and the cost of the optimal bounds is $O(p^3)$ ops.

Note that if $R(Y)$ does not have full rank then C will have to be in upper echelon form, the error bounds of Chapter 10 require a little less computation, and the derivation of C should be done with a rank revealing QR factorization.

Justification for step 7 comes from

Lemma 11.8.1. A is orthogonally similar to

$$\bar{A} = \begin{bmatrix} \Theta & C^* & O^* \\ C & & \\ O & & U \end{bmatrix}$$

where U is unknown.

The proof constitutes Exercise 11.8.2.

Exercises on Section 11.8

- 11.8.1. Use the fact that Θ is the (matrix) Rayleigh quotient of \mathbf{Y} to show that $\mathbf{Y}^* \mathbf{S} = \mathbf{O}$ when $\mathbf{R}(\mathbf{Y}) = \mathbf{S}\mathbf{C}$ has full rank.
- 11.8.2. Let $\mathbf{P} = (\mathbf{Y}, \mathbf{S}, \mathbf{J})$ be an n -by- n orthonormal matrix. Use it to prove Lemma 11.8.1.
- 11.8.3. Show that $\|\mathbf{R}(\mathbf{Y})\| = \|\mathbf{R}(\mathbf{Q})\|$ when $p = m$. Here $\mathbf{Y} = \mathbf{Q}\mathbf{G}$, $\mathbf{G}^* = \mathbf{G}^{-1}$.
- 11.8.4. Compare the operation counts for computing \mathbf{C} by (a) forming $\mathbf{R}(\mathbf{Y})^* \mathbf{R}(\mathbf{Y})$ and doing a Cholesky factorization and (b) using modified Gram-Schmidt and discarding the columns of \mathbf{S} . What about storage requirements?

*11.9. A Priori Bounds for Interior Ritz Approximations

The subspace \mathcal{S}^m yields RR approximations (θ_i, \mathbf{y}_i) to eigenpairs $(\alpha_{i'}, \mathbf{z}_{i'})$ for $i = 1, \dots, m$. The results of this section are of interest when \mathcal{S}^m is sufficiently well placed in \mathcal{E}^n that α_i itself is the closest eigenvalue to θ_i . The bounds are a priori and are not computable. Their value lies in assessing the *potential* accuracy of the RR approximations from \mathcal{S}^m (see Chapter 12). The idea is to use the error in the extremal Ritz vector \mathbf{y}_1 to obtain a simple error bound for θ_2 , and then to use the errors in \mathbf{y}_1 and θ_2 to bound the error in \mathbf{y}_2 , and so on, moving toward the interior of the spectrum.

By the minmax characterization of eigenvalues, $\alpha_1 \leq \theta_1 \leq \rho(\mathbf{s})$ for any \mathbf{s} in \mathcal{S}^m . A clever choice of \mathbf{s} (near the eigenvector \mathbf{z}_1) will lead to as good a bound as the circumstances warrant. However, the corresponding bounds for θ_2 , $\alpha_2 \leq \theta_2 \leq \rho(\mathbf{s})$, hold only if $\mathbf{s}^* \mathbf{y}_1 = 0$ and $\mathbf{s} \in \mathcal{S}^m$ (Exercise 11.9.1). And there lies the difficulty because the condition $\mathbf{s}^* \mathbf{y}_1 = 0$ demands exact knowledge of \mathbf{y}_1 and that is not allowed in a strict a priori analysis. The remedy is to take two usable conditions (1) $\mathbf{s}^* \mathbf{z}_1 = 0$ and (2) a bound on $\angle(\mathbf{y}_1, \mathbf{z}_1)$ and then derive modified bounds of the form

$$\alpha_2 \leq \theta_2 \leq \rho(\mathbf{s}) + \text{"a little something."}$$

Let $\phi_i = \angle(\mathbf{y}_i, \mathbf{z}_i)$, $i = 1, \dots, m$.

Lemma 11.9.1. *For each $j \leq m$ and for any unit vector $s \in S^m$ which satisfies*

$$s^* z_i = 0, \quad i = 1, \dots, j-1,$$

then

$$\begin{aligned} \alpha_j &\leq \theta_j \leq \rho(s) + \sum_{i=1}^{j-1} (\alpha_{-1} - \theta_i) \sin^2 \phi_i \\ &\leq \rho(s) + \sum_{i=1}^{j-1} (\alpha_{-1} - \alpha_i) \sin^2 \phi_i. \end{aligned} \quad (11.30)$$

Proof. The first and third inequalities follow from the Cauchy interlace theorem. To establish the middle one take s to be a unit vector and decompose it in the Ritz vector basis of S^m as

$$s = t + \sum_{i=1}^{j-1} y_i \gamma_i, \quad \text{where } t^* y_i = 0, \quad i = 1, \dots, j-1. \quad (11.31)$$

The hypothesis $s^* z_i = 0$ leads straight to a bound on γ_i ,

$$|\gamma_i| = |s^* y_i| = |s^*(y_i - z_i \cos \phi_i)| \leq \|s\| \cdot |\sin \phi_i| = |\sin \phi_i|. \quad (11.32)$$

Since $t^* y_i = 0$, $i < j$, and $y_i^* y_k = 0$, $k \neq i$,

$$\rho(s) = t^* A t + \sum_{i=1}^{j-1} (y_i^* A y_i) \gamma_i^2. \quad (11.33)$$

In order to turn (11.33) into the desired inequality all the terms must be negative, so we shift A by α_{-1} ;

$$\begin{aligned} \rho(s) - \alpha_{-1} &= t^* (A - \alpha_{-1}) t + \sum (\theta_i - \alpha_{-1}) \gamma_i^2 \\ &\geq t^* (A - \alpha_{-1}) t / (t^* t) + \sum (\theta_i - \alpha_{-1}) \gamma_i^2, \\ &\quad \text{since } \|t\| \leq \|s\| \leq 1, \\ &\geq \rho(t) - \alpha_{-1} + \sum_{i=1}^{j-1} (\theta_i - \alpha_{-1}) \sin^2 \phi_i, \quad \text{by (11.32).} \end{aligned}$$

Finally note that $\rho(\mathbf{t}) \geq \theta_j = \min_{\mathbf{u} \in \mathcal{S}^m} \rho(\mathbf{u})$ over $\mathbf{u} \in \mathcal{S}^m$, $\mathbf{u} \perp \mathbf{y}_i$, $i < j$. Add α_{-1} to each side and the middle inequality is established. \square

The next task is to see how each angle ϕ_j can be estimated in terms of the previous ones. To do this we introduce, temporarily, $\phi_{ij} = \angle(\mathbf{z}_i, \mathbf{y}_j)$, $i = 1, \dots, n$, $j = 1, \dots, m$. Then $\phi_i = \phi_{ii}$. The following trigonometric facts are needed: for each j ,

$$\mathbf{y}_j = \sum_{i=1}^n \mathbf{z}_i \cos \phi_{ij}, \quad (11.34)$$

$$|\cos \phi_{ij}| \leq |\sin \phi_i|, \text{ see Exercise 11.9.3,} \quad (11.35)$$

$$\sum_{i=j+1}^n \cos^2 \phi_{ij} = \sin^2 \phi_j - \sum_{i=1}^{j-1} \cos^2 \phi_{ij}. \quad (11.36)$$

Lemma 11.9.2. *For each $j = 1, \dots, m$,*

$$\sin^2 \phi_j \leq \left[(\theta_j - \alpha_j) + \sum_{i=1}^{j-1} (\alpha_{j+1} - \alpha_i) \sin^2 \phi_i \right] / (\alpha_{j+1} - \alpha_j).$$

Proof. By (11.34) and Exercise 11.9.2

$$\rho(\mathbf{y}_j; \mathbf{A} - \alpha_j) = \theta_j - \alpha_j = \sum_{i=1}^n (\alpha_i - \alpha_j) \cos^2 \phi_{ij}.$$

Take terms involving the previous ϕ_{ij} to the other side and use the ordering $\alpha_1 \leq \alpha_2 \leq \dots$ to find

$$\begin{aligned} (\theta_j - \alpha_j) &+ \sum_{i=1}^{j-1} (\alpha_j - \alpha_i) \cos^2 \phi_{ij} \\ &= \sum_{i=j+1}^n (\alpha_i - \alpha_j) \cos^2 \phi_{ij} \\ &\geq (\alpha_{j+1} - \alpha_j) \sum \cos^2 \phi_{ij} \\ &= (\alpha_{j+1} - \alpha_j) \left(\sin^2 \phi_j - \sum_{i=1}^{j-1} \cos^2 \phi_{ij} \right) \text{ by (11.36).} \end{aligned}$$

Solve for $\sin^2 \phi_j$ and use (11.35) to obtain desired inequality. \square

The case $j = 1$ yields $\sin^2 \phi_1 \leq (\theta_1 - \alpha_1)/(\alpha_2 - \alpha_1)$. This is a disguised form of the error estimate for Rayleigh quotients. True a priori bounds are obtained by choosing suitable vectors s and then using (11.30) and Lemma 11.9.2 alternately to majorize $\sin^2 \varphi_j$ and $\theta_j - \alpha_j$. The price paid for spurning explicit use of θ_j is rather high as section 12.4 reveals.

Exercises on Section 11.9

11.9.1. Prove that $\alpha_2 \leq \theta_2 \leq \rho(s)$ if and only if $s \in S^m$ and $s^*y_1 = 0$ by using the minmax characterization of eigenvalues.

11.9.2. Show that $\rho(y_i) = \theta_i$ and $y_i^*y_j = 0$ by using the definition of y_i .

11.9.3. Establish (11.35) by using the results of Exercise 11.9.2.

11.9.4. Prove that $\alpha_{-1} \geq \theta_{-1} \geq \rho(s) - \sum_{i=1}^{j-1} (\alpha_i - \alpha_1) \sin^2 \phi_{-i}$ where

$$\phi_{-i} = \angle(y_{-i}, z_{-i}) \text{ and } s^*z_{-i} = 0, \quad i = 1, \dots, j-1.$$

11.9.5. What is the analogue of Lemma 11.9.2 for $\sin^2 \phi_{-j}$?

11.9.6. Show that Lemma 11.9.1 continues to hold when α_{-1} is replaced by θ_{-1} .

*11.10. Nonorthogonal Bases

In principle it is always possible to choose an orthonormal basis for the subspace S^k but in practice it is not always convenient to do so. Some techniques, such as inverse iteration, produce approximate eigenvectors that fail to be mutually orthogonal when the eigenvalues are huddled close together. When n is large it is tempting to skip the precaution of reorthogonalizing computed eigenvectors, particularly when they are nearly orthogonal. What is there to lose?

In order to answer the question quantitatively let S be an n -by- k matrix with normalized columns, probably a matrix of “Ritz” vectors, and let H be any k -by- k symmetric matrix, probably $H = \text{diag}(\theta_1, \dots, \theta_k)$. In any case $\theta_i = \lambda_i[H]$. The residual matrix is $AS - SH$.

The only condition needed on S is that it should have full column rank. Although it is not strictly necessary (for Theorem 11.10.1) we assume that the columns of S all have unit length. It follows that when $\sigma_{\min}(S) = 1$ then $\sigma_{\max}(S) = 1$ (why?) and thus S is orthonormal. In other words the descent

from orthogonality to linear dependence is measured by the decline of $\sigma_{\min}(S)$ from 1 to 0.

The goal of this section is to show that Theorem 11.5.1 degrades gracefully as $\sigma_{\min}(S)$ declines from 1 toward 0. Theorem 11.5.1 pairs up the eigenvalues $\{\theta_j\}_1^k$ of the Rayleigh quotient matrix $T = Q^*AQ$, where $S^k = \text{range}(Q)$, with k eigenvalues $\{\alpha_j\}$ of A . As we shall see T may be replaced by any m -by- m real symmetric matrix H (with eigenvalues θ_j) without invalidating the result.

In 1967 Kahan proved that, given A , H , and S as described above, there is a pairing such that

$$|\alpha_j - \theta_j| \leq c \|AS - SH\| / \sigma_{\min}(S),$$

with $c = \sqrt{2}$. A proof was given in the 1980 edition of this book. Kahan conjectured that the constant c should be 1, but despite efforts by several researchers the conjecture was not established until 1995. Three Chinese mathematicians, Cao, Xie, and Li (see [Cao, Xie, and Li, 1996]), came up with a subtle induction argument but the proof given here is more straightforward than theirs is. The first two lemmas are standard results. As usual $\|X\|^2 = \lambda_{\max}(X^*X)$.

Lemma 11.10.1. *Let $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$ have full column rank. If $\|X_2\| < \sigma_{\min}(X)$ then X_1 has full column rank.*

Proof.

$$X_1^*X_1 = X^*X - X_2^*X_2.$$

By Weyl's monotonicity theorem in section 10.3

$$\lambda_{\min}(X_1^*X_1) \geq \lambda_{\min}(X^*X) - \lambda_{\max}(X_2^*X_2).$$

The hypothesis asserts that the right side exceeds 0. So X_1 has full column rank. \square

Lemma 11.10.2. Suppose that the spectrum of H lies in $[\mu, \nu]$ and the spectrum of A lies outside $[\mu - \delta, \nu + \delta]$, $\delta > 0$. For any conformable B there is a unique matrix S of the same shape as B satisfying $AS - SH = B$. Moreover,

$$\delta\|S\| < \|B\|.$$

Proof. For any τ , $(A - \tau I)S - S(H - \tau I) = B$. Consequently, by shifting to the midpoint of $[\mu, \nu]$ one may assume that

$$-\gamma \leq \lambda_{\min}(H) \leq \lambda_{\max}(H) \leq \gamma \quad (2\gamma = \nu - \mu) \quad (11.37)$$

and

$$\gamma + \delta < \min_i |\lambda_i[A]|,$$

so

$$\|A^{-1}\| < (\gamma + \delta)^{-1}. \quad (11.38)$$

Now

$$\begin{aligned} \|B\| &= \|AS - SH\| \\ &\geq \|AS\| - \|SH\| \\ &\geq \|A^{-1}\|^{-1}\|S\| - \|S\| \cdot \|H\| \\ &> (\gamma + \delta)\|S\| - \|S\|\gamma = \delta\|S\|. \end{aligned}$$

The last line invokes (11.37) and (11.38). \square

The long delay in removing the factor $\sqrt{2}$ from Kahan's theorem is puzzling. One factor that may have contributed to the delay is the convention that multiple eigenvalues are repeated according to their multiplicity, and when pairing eigenvalues of A with those of H it may be necessary to assign to different θ 's some eigenvalues of A that are the same!

Let us recall the difficulty. Suppose that $H = \text{diag}(\theta_1, \dots, \theta_k)$ and $R = AS - SH = [r_1, \dots, r_k]$. For each $i = 1, 2, \dots, k$ there is an eigenvalue α of A such that $|\alpha - \theta_i| \leq \|r_i\|$, but Example 11.5.1 shows that there may not be k distinct α 's, one for each i . So we ask for the smallest η such that a distinct α may be found in each interval $[\theta_i - \eta, \theta_i + \eta]$. It turns out that, in the absence of more

information, $\eta = \|R\|$ when $\sigma_{\min}(S) = 1$ and this section shows that when S has full rank then $\eta = \|R\|/\sigma_{\min}(S)$.

In computer science it is often necessary to distinguish a “cell” or “address” in the memory from its contents. In order to clarify the proof that follows we shall think of $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ as an array of cells or addresses in which are stored the real values $\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n$ in A 's spectrum. By definition the locations α_i , $i = 1, \dots, n$ are distinct. Abusing language slightly, but harmlessly, we shall also call α_i an eigenvalue of A . The thrust of the theorem is to find k locations in α to match the k locations $\theta_1, \dots, \theta_k$ in the array θ of H 's eigenvalues. The analysis has to cover extreme situations in which, perhaps, both A and H are identity matrices.

The next result is the key.

Lemma 11.10.3. *Let $A = \text{diag}(\bar{\alpha}_1, \dots, \bar{\alpha}_n)$, $H = \text{diag}(\bar{\theta}_1, \dots, \bar{\theta}_k)$, where*

$$\begin{aligned}\bar{\alpha}_1 &\leq \bar{\alpha}_2 \leq \dots \leq \bar{\alpha}_n, \\ \bar{\theta}_1 &\leq \bar{\theta}_2 \leq \dots \leq \bar{\theta}_k.\end{aligned}$$

For any n -by- k matrix S define $R := AS - SH$. Assume that S has full column rank k , so $1 \leq k \leq n$ and let $\eta := \|R\|/\sigma_{\min}(S)$.

There are at least k locations $\alpha_{1'}, \alpha_{2'}, \dots, \alpha_{k'}$ such that, for $j = 1', \dots, k'$,

$$\bar{\alpha}_j \in [\bar{\theta}_1 - \eta, \bar{\theta}_k + \eta].$$

Proof. Let A_- be the principal submatrix of A containing all α_l such that $\bar{\alpha}_l < \bar{\theta}_1 - \eta$, let A_+ be principal submatrix of A containing all α_m such that $\bar{\alpha}_m > \bar{\theta}_k + \eta$, and let A_0 have the remaining $\bar{\alpha}$'s, so that

$$A = A_- \oplus A_0 \oplus A_+.$$

The spectrum of H lies in $[\bar{\theta}_1, \bar{\theta}_k]$ and the spectrum of $A_- \oplus A_+$ lies outside $[\bar{\theta}_1 - \eta, \bar{\theta}_k + \eta]$. Partition S and R conformably with A :

$$S = \begin{pmatrix} S_- \\ S_0 \\ S_+ \end{pmatrix}, \quad R = \begin{pmatrix} R_- \\ R_0 \\ R_+ \end{pmatrix}. \quad (11.39)$$

The equation $AS - SH = R$ includes the equation

$$\begin{pmatrix} R_- \\ R_+ \end{pmatrix} = \begin{pmatrix} A_- & O \\ O & A_+ \end{pmatrix} \cdot \begin{pmatrix} S_- \\ S_+ \end{pmatrix} - \begin{pmatrix} S_- \\ S_+ \end{pmatrix} H. \quad (11.40)$$

Apply Lemma 11.10.2 to (11.40) using the full rank hypothesis to conclude that

$$\eta \left\| \begin{pmatrix} S_- \\ S_+ \end{pmatrix} \right\| < \left\| \begin{pmatrix} R_- \\ R_+ \end{pmatrix} \right\|. \quad (11.41)$$

By the definition of η and (11.41),

$$\frac{\|R\|}{\sigma_{\min}(S)} = \eta < \frac{\left\| \begin{pmatrix} R_- \\ R_+ \end{pmatrix} \right\|}{\left\| \begin{pmatrix} S_- \\ S_+ \end{pmatrix} \right\|} \leq \frac{\|R\|}{\left\| \begin{pmatrix} S_- \\ S_+ \end{pmatrix} \right\|}. \quad (11.42)$$

The last inequality holds because $\begin{pmatrix} R_- \\ R_+ \end{pmatrix}$ is a submatrix of R . From (11.42)

$$\sigma_{\min}(S) > \left\| \begin{pmatrix} S_- \\ S_+ \end{pmatrix} \right\|. \quad (11.43)$$

Use Lemma 11.10.1 with (11.43) to see that the perturbation

$$S \longrightarrow \begin{pmatrix} O \\ S_0 \\ O \end{pmatrix}$$

cannot change the rank. Hence

$$\text{row size}(S_0) \geq \text{rank} \begin{pmatrix} O \\ S_0 \\ O \end{pmatrix} = \text{rank}(S) = k. \quad (11.44)$$

So there are at least k $\bar{\alpha}$'s in $[\bar{\theta}_1 - \eta, \bar{\theta}_k + \eta]$ as claimed. \square

With this preparation the proof of the theorem is straightforward.

Theorem 11.10.1. Let A be n by n , H be k by k , and let $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ hold the eigenvalues of A and $\theta = (\theta_1, \dots, \theta_k)$ hold the eigenvalues of H . Let S be any n -by- k matrix with full column rank, so $1 \leq k \leq n$. Then there are at least k locations $\alpha_{1'}, \alpha_{2'}, \dots, \alpha_{k'}$ in α such that, for $j = 1, \dots, k$,

$$|\bar{\alpha}_{j'} - \bar{\theta}_j| \leq \|AS - SH\|/\sigma_{\min}(S).$$

Proof.

1. For any unitary matrices P and Q the substitutions

$$A \longrightarrow P^*AP, \quad H \longrightarrow Q^*HQ, \quad S \longrightarrow P^*SQ$$

leave the hypotheses and conclusions unchanged because the spectral norm is unitarily invariant (Fact 10 in Chapter 1). Consequently there is no loss of generality in assuming that both A and H are diagonal with any ordering of the eigenvalues along the diagonal that is convenient. Both A and H are ordered monotonically:

$$\begin{aligned} A &= \text{diag}(\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n), \quad \bar{\alpha}_i \leq \bar{\alpha}_{i+1}, \quad i = 1, \dots, n-1, \\ H &= \text{diag}(\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_k), \quad \bar{\theta}_i \leq \bar{\theta}_{i+1}, \quad i = 1, \dots, k-1. \end{aligned}$$

2. Consider the spectrum of H and the region $\mathcal{C} = \mathcal{C}(\{\theta_i\}_1^k) := \bigcup_{i=1,k} [\bar{\theta}_i - \eta, \bar{\theta}_i + \eta]$ that surrounds it. In general \mathcal{C} may consist of several connected components and there may be $\bar{\alpha}$'s in between the components. The required pairing of α 's with θ 's will be done separately for each connected component. To be specific suppose there are q such components (necessarily disjoint) and partition H accordingly:

$$H = H_1 \oplus H_2 \oplus \cdots \oplus H_q.$$

It is convenient, for this section of the proof, to relabel the spectrum of H_i as $(\theta_1^{(i)}, \dots, \theta_{k_i}^{(i)})$ where $\sum_{i=1}^q k_i = k$ and $\bar{\theta}_j^{(i)} \leq \bar{\theta}_{j+1}^{(i)}$, $j = 1, \dots, k_i - 1$.

Each H_i is associated with a component, say, \mathcal{C}_i , of \mathcal{C} . By choice of H_i , $i = 1, \dots, q$,

$$\mathcal{C}_i = \mathcal{C}(\theta_1^{(i)}, \dots, \theta_{k_i}^{(i)}) = [\bar{\theta}_1^{(i)} - \eta, \bar{\theta}_{k_i}^{(i)} + \eta]. \quad (11.45)$$

A fortiori, the analogue of (11.45) holds for every subsequence of $(\theta_1^{(i)}, \dots, \theta_{k_i}^{(i)})$ consisting of *consecutive terms*. Recall that subsets of linearly independent sets are linearly independent. Thus Lemma 11.10.3 applied to any such subsequence guarantees that

for $0 \leq j < j + l \leq k_i$,

there are at least l $\bar{\alpha}$'s in $\mathcal{C}(\theta_{j+1}^{(i)}, \dots, \theta_{j+l}^{(i)})$. (11.46)

3. It remains to show how (11.46) guarantees that each $\mathcal{C}(\theta_j^{(i)})$ has its own α and the analysis does not involve matrix theory. There is no harm in dropping the index i for the remainder of this section. Property (11.46) with $l = 1$ shows that there is at least one $\bar{\alpha}$ in each interval $\mathcal{C}(\theta_m)$, $m = 1, \dots, k$, but because the intervals may overlap this fact alone does not give a matching. It does permit valid definitions,

$$\begin{aligned}\alpha_{1'} &= \min \{\alpha_m : \bar{\alpha}_m \in \mathcal{C}(\theta_1)\}, \\ \alpha_{k'} &= \max \{\alpha_m : \bar{\alpha}_m \in \mathcal{C}(\theta_k)\}.\end{aligned}$$

The following argument may appear fussy but that level of detail seems to be needed. Consider $\mathcal{C}(\theta_2)$. If $\bar{\alpha}_{1'} \notin \mathcal{C}(\theta_2)$ then there must be an $\alpha_m (> \alpha_{1'})$ with $\bar{\alpha}_m \in \mathcal{C}(\theta_2)$. There must be at least two $\bar{\alpha}$'s in $\mathcal{C}(\theta_1, \theta_2)$, so whether or not $\bar{\alpha}_{1'} \in \mathcal{C}(\theta_2)$ it is valid to define

$$\alpha_{2'} = \min \{\alpha_m : \alpha_m > \alpha_{1'}, \bar{\alpha}_m \in \mathcal{C}(\theta_2)\}.$$

Similarly

$$\alpha_{(k-1)'} = \max \{\alpha_m : \alpha_m < \alpha_{k'}, \bar{\alpha}_m \in \mathcal{C}(\theta_{k-1})\}.$$

Now consider the typical case: $\alpha_{1'}, \alpha_{2'}, \dots, \alpha_{(j-1)'}$ have all been defined and attention is focused on $\mathcal{C}(\theta_j)$, $j \leq [k/2]$. Let l , depending on j , be the smallest index such that $\alpha_{l'}, \alpha_{(l+1)'}, \dots, \alpha_{(j-1)'}$ are *consecutive locations* in α . Either $l = 1$ or there is an α_k satisfying

$$\alpha_{(l-1)'} < \alpha_k < \alpha_{l'}.$$

By the minimality in the definition of $\alpha_{l'}$ it is impossible that $\bar{\alpha}_k \in \mathcal{C}(\theta_l)$. Consequently, whether $l = 1$ or not, $\alpha_{l'}$ is the smallest α with $\bar{\alpha} \in \mathcal{C}(\theta_l)$. Now consider $\mathcal{C}(\theta_l, \dots, \theta_j)$. If there were no $\alpha_m > \alpha_{(j-1)'}$ in $\mathcal{C}(\theta_j)$ then $\alpha_{l'}, \alpha_{(l+1)'}, \dots, \alpha_{(j-1)'}$ would be the only $\bar{\alpha}$'s

in $\mathcal{C}(\theta_l, \dots, \theta_j)$ contradicting (11.46) which guarantees $j - l + 1$ of them. Consequently it is valid to define

$$\alpha_{j'} = \min \left\{ \alpha_m : \alpha_m > \alpha_{(j-1)'}, \bar{\alpha}_m \in \mathcal{C}(\theta_j) \right\}.$$

Similarly

$$\alpha_{(k-j)'} = \max \left\{ \alpha_m : \alpha_m < \alpha_{(k-j+1)'}, \bar{\alpha}_m \in \mathcal{C}(\theta_{k-j}) \right\}.$$

If there is a middle θ , say, θ_p , then $\alpha_{p'}$ may be defined using either min or max to complete the matching for all θ 's associated with H_i .

4. By repeating the matching process for each i , $i = 1, \dots, q$, one obtains the required one-to-one correspondence between the eigenvalues of H and some eigenvalues of A . \square

Notes and References

The RR approximation procedure has become a standard tool in many branches of mathematics and engineering. The original references are [Rayleigh, 1899] and [Ritz, 1909]. It is easy to get confused over the senses in which the approximations are optimal and the senses in which they are not. We have not found a text which sets the matter out clearly.

The residual bound on clustered eigenvalues appears in the unpublished report [Kahan, 1967]. The eigenvalue bounds which are based on gaps in the spectrum have their origins in [Temple, 1933], [Weinstein, 1935], and [Kato, 1949], but the material for section 11.7 came from [Davis and Kahan, 1970].

In a different vein, complementing the a posteriori results discussed so far, come the inequalities which show how the Ritz approximations to inner eigenvalues are affected by the errors in the approximations to outer eigenvalues. The source is [Kaniel, 1966] and some unpublished work of Paige. The application comes in the next chapter.

Results on the orthogonality of Ritz vectors to complementary eigenvectors (those belonging to eigenvalues other than the one associated with the Ritz vector) are given in [Knyazev, 1994].

Krylov Subspaces

12.1. Introduction

Of considerable importance in the theory of various methods for computing eigenpairs of A is a simple type of subspace which is determined by a single nonzero vector, say, f . Krylov matrices $K^m(f)$ and Krylov subspaces $\mathcal{K}^m(f)$ are defined by

$$\begin{aligned} K^m(f) &= (f, Af, \dots, A^{m-1}f), \\ \mathcal{K}^m(f) &= \text{range } K^m(f). \end{aligned}$$

The dimension of \mathcal{K}^m will usually be m unless either f is specially related to A or $m > n$.

When started from f , the power method described in Chapter 4 will compute the columns of $K^m(f)$ one by one. However, each column is written over its predecessor and so only the latest column is retained, thereby economizing on storage. In principle the whole of $K^m(f)$ could be saved and the RR (Rayleigh–Ritz; see Chapter 11) approximations from $\mathcal{K}^m(f)$ could be computed. For $m \geq 2$ the RR approximations will be better than the one from the power method, but they will be more expensive. Are they cost effective? That is a nice technical question involving f , the $\alpha_i (\equiv \lambda_i[A])$, storage capacity, and the ease with which A may be manipulated, but the answer briefly is a resounding yes.

Estimates of the comparative accuracy of the two methods will be taken up in this chapter and then will come a description of how RR approximations from $\mathcal{K}^m(f)$ can be computed far more economically than appears possible at first sight. The following examples may provide incentive for tackling the details of the analysis. We hope that the whole theory will seem to emerge as a *natural* consequence of using Krylov subspaces.

Example 12.1.1. $m = 2, n = 3$. “A plane is better than its axes.” $A = \text{diag}(3, 2, 1)$, $f = (1, 1, \eta)^*$, η small. Let $(\theta_i^{(m)}, y_i^{(m)})$, $i = 1, \dots, m$ be the RR approximations from $\mathcal{K}^m(f)$. $\angle(u, v)$ denotes the acute angle between u and v . Then for $\eta < 0.01$,

$m = 2$	$\angle(y_2^{(2)}, e_1) \leq \eta\sqrt{2}$ is better than	$\angle(A^2 f, e_1) \approx 4/9$
$m = 3$	$\angle(y_3^{(3)}, e_1) = 0$ is better than	$\angle(A^3 f, e_1) \approx 8/27$

This simple example brings out the fact that the power method requires, in principle, an infinite number of iterations to capture an eigenvector whereas the RR approximations are exact for $\mathcal{K}^n(f)$. The case $m = 2$ shows how much better the plane $\mathcal{K}^2(f)$ can be than either of its given axes, f and Af , as Figure 12.1 suggests.

Example 12.1.2.

$$\alpha_i = \lambda_i[A] = i, \quad i = 0, \dots, n+1.$$

The power method will converge to z_{-1} , the dominant eigenvector. This problem is moderately difficult when n is large. The following table gives the number of steps m required to guarantee that, for any f , (final error angle) < (initial error angle)/100. The eigenvalue error will then be reduced by a factor of 10^4 approximately. The expressions come from section 12.6.

The first number allows for perversely chosen f 's such as $z_1 + z_2 + 10^{-6}z_{-1}$. The second number, in parentheses, omits the common factor in the error bounds in section 12.6 and gives a good estimate of m when f is a *random* starting vector.

n	PM: $(A - \alpha_1)^m f$	$\mathcal{K}^m(f)$
10^2	693 (463)	38 (27)
10^3	8061 (4607)	139 (84)
10^4	92105 (46054)	500 (265)
n	$n \ln[100\sqrt{n}](n \ln 100)$	$\frac{1}{2}\sqrt{n} \ln[200\sqrt{n}] (\frac{1}{2}\sqrt{n} \ln 200)$

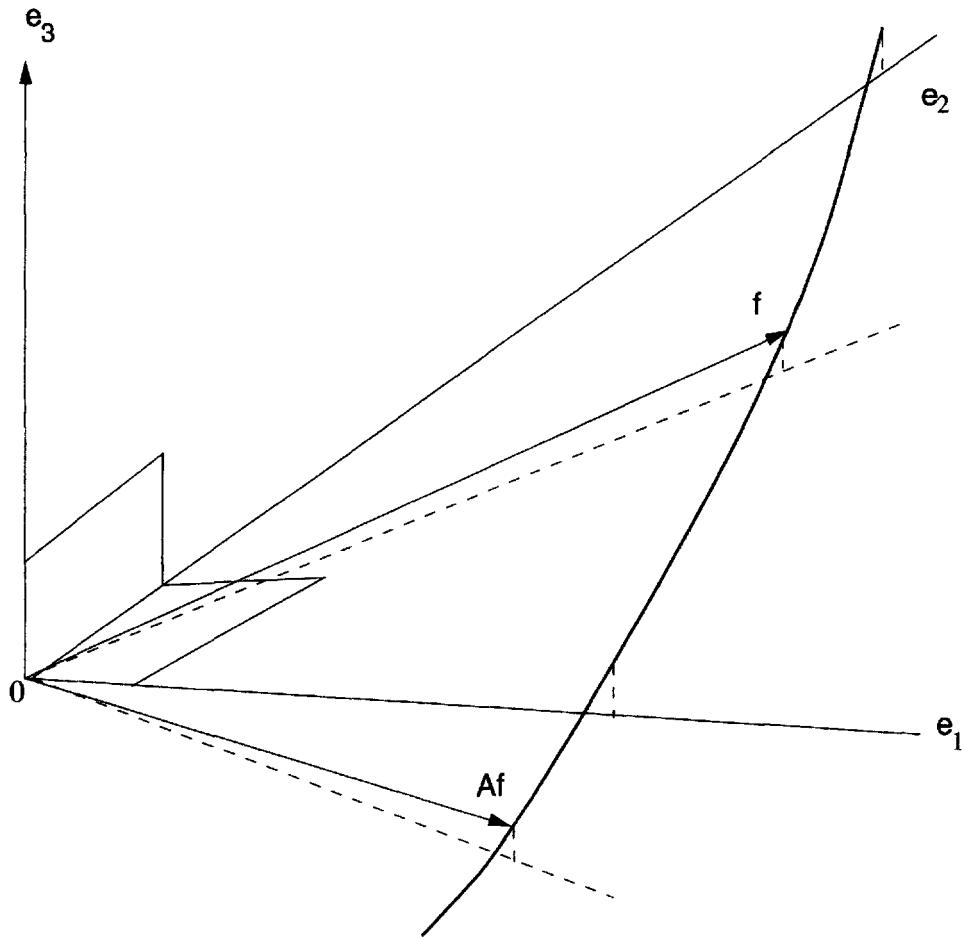


FIG. 12.1. *A plane is better than its axes.*

12.2. Basic Properties

12.2.1. A Theoretical Limitation

The starting vector f for $\mathcal{K}(f)$ might be orthogonal to one or more eigenvectors of A . These eigenvectors are orthogonal to $\mathcal{K}^j(f)$ for all j , in exact arithmetic (Exercise 12.2.1), and so can never be detected by the power method, or any Krylov subspace method, based on f .

In order to describe the situation formally it helps to look at Exercise 12.2.2 and introduce (a) $\Sigma(f)$, the spectrum of A associated with f and defined by

$\Sigma(f) = \{\alpha : \alpha = \lambda[A] \text{ and } H_\alpha f \neq 0\}$ and (b) $\mathcal{J}(f)$, the smallest invariant subspace of \mathcal{E}^n which contains f . It can be shown that $\mathcal{J}(f) = \text{span}\{A_\alpha f : \alpha \in \Sigma(f)\}$ and that A_f , the restriction of A to $\mathcal{J}(f)$, has simple eigenvalues (Exercise 12.2.3). Moreover the Krylov subspaces eventually fill up $\mathcal{J}(f)$; thus for some $l \leq n$,

$$\mathcal{K}^1(f) \subset \mathcal{K}^2(f) \subset \cdots \subset \mathcal{K}^l(f) = \mathcal{K}^{l+1}(f) = \mathcal{J}(f).$$

There are two ways of adjusting the approximation theory to this limitation. Either assume, in fine academic style, that $\mathcal{J}(f) = \mathcal{E}^n$ and $A_f = A$ or complicate the statement of all results by referring to A_f instead of A . The fact remains that numerical methods based on powering may, in exact arithmetic, fail to detect some eigenvectors of some A 's and they must, again in exact arithmetic, fail to detect the multiplicity of *any* eigenvalue they find.

Fortunately roundoff errors make the assumption $A_f = A$ a realistic one in practice. This comment will be amplified during the discussion of the Lanczos algorithm in Chapter 13.

12.2.2. Invariance Properties

The subspace $\mathcal{K}^m(f)$ depends on A and, when necessary, it is also denoted by $\mathcal{K}^m(f; A)$. The following invariance properties are valuable.

- I. Scaling: $\mathcal{K}^m(\sigma f; \tau A) = \mathcal{K}^m(f; A), \quad \sigma \neq 0, \tau \neq 0.$
- II. Translation: $\mathcal{K}^m(f; A - \sigma) = \mathcal{K}^m(f; A).$
- III. Change of Basis: $\mathcal{K}^m(Pf; PAP^*) = P\mathcal{K}^m(f; A), \quad P^* = P^{-1}.$

Verification is left to the reader but some comments are in order.

The error bounds associated with approximation from $\mathcal{K}^m(f; A)$ should also be invariant in the same way.

- Property II is puzzling at first because it fails completely for the associated Krylov matrices $\mathcal{K}^m(f; A)$. Thus the power method depends strongly on any shift σ , but each σ merely induces a different basis for the same subspace $\mathcal{K}^m(f; A)$. One consequence of properties I and II is that there is no loss of generality in supposing that $\lambda_1[A] = -1$ and $\lambda_{-1}[A] = +1$. This normalization reveals why, as m increases, the space $\mathcal{K}^m(f; A)$ moves close to the eigenvectors

belonging to extreme eigenvalues. Not surprisingly the error bounds involve ratios of differences of A 's eigenvalues; such quantities are properly invariant. We call them gap ratios.

- Property III reflects the fact that an orthogonal basis change in \mathcal{E}^n induces an orthogonal similarity transformation of A . There is no loss of generality in studying diagonal matrices A for theoretical purposes but such matrices do not capture the effect of roundoff error.

Exercises on Section 12.2

- 12.2.1. Show that if $Az = z\alpha$ and $z \perp f$ then $z \perp \mathcal{K}^m(f)$ for all $m > 1$.
- 12.2.2. Show that $\mathcal{J}(f) = \text{span } \{H_\alpha f : \alpha \in \Sigma(f)\}$ by using $AH_\alpha = H_\alpha A = \alpha H_\alpha$.
- 12.2.3. Show that $A_f \equiv A|_{\mathcal{J}(f)}$ has simple eigenvalues.
- 12.2.4. Consider $A = \text{diag}(-1, -0.5, 0, 0.5, 1)$ and $f = (1, 1, \dots, 1)^*$. Compute the angle between $A^k f$ and the (e_1, e_5) plane for $k = 0, 1, 2, 3, 4$.
- 12.2.5. Verify the invariance properties I, II, and III.

12.3. Representation by Polynomials

Each element s in $\mathcal{K}^m(f)$ has the special form

$$s = \sum_{i=0}^{m-1} (A^i f) \gamma_i = \sum_{i=0}^{m-1} (\gamma_i A^i) f = \pi(A) f \quad (12.1)$$

where $\pi(\xi) \equiv \sum \gamma_i \xi^i$ is a polynomial of degree $< m$. This chapter will make heavy use of the vector space \mathcal{P}^k of all real polynomials of degree not exceeding k . For example (12.1) has the nice interpretation

$$\mathcal{K}^m(f) = \{\pi(A)f : \pi \in \mathcal{P}^{m-1}\} \quad (12.2)$$

and provides a one-to-one correspondence between \mathcal{K}^m and \mathcal{P}^{m-1} thanks to our tacit assumption that $\dim \mathcal{K}^m = m$.

Not surprisingly polynomials play a large role in describing approximations from \mathcal{K}^m . For future reference we describe the Ritz vectors. Recall from Chapter 11 that for any subspace \mathcal{S}^m the Ritz vectors $y_i, i = 1, \dots, m$ are mutually orthogonal and are characterized by the property

$$Ay_i - y_i \theta_i \perp \mathcal{S}^m, \quad i = 1, \dots, m. \quad (12.3)$$

In Krylov subspaces it is easy to characterize vectors orthogonal to a particular y_k .

Lemma 12.3.1. *Let (θ_i, y_i) , $i = 1, \dots, m$ be the RR approximations from $\mathcal{K}^m(f)$. If $\omega \in \mathcal{P}^{m-1}$ then*

$$\omega(A)f \perp y_k \text{ if and only if } \omega(\theta_k) = 0.$$

Proof. Suppose first that $\omega \in \mathcal{P}^m$, not just \mathcal{P}^{m-1} , and that $\omega(\xi) = (\xi - \theta_k)\pi(\xi)$ where $\pi \in \mathcal{P}^{m-1}$. Thus

$$\pi(A)f \in \mathcal{K}^m(f) \quad (12.4)$$

and

$$\begin{aligned} y_k^* \omega(A)f &= y_k^*(A - \theta_k)\pi(A)f \\ &= [(\mathbf{A} - \theta_k)y_k]^* \pi(A)f, \quad \text{since } \mathbf{A} = \mathbf{A}^*, \\ &= 0, \quad \text{by (12.3) and (12.4).} \end{aligned}$$

This establishes sufficiency with a little extra information since $\omega(A)f$ covers some vectors outside \mathcal{K}^m . In particular $y_k \perp S_k$ where

$$S_k \equiv (A - \theta_k)\mathcal{K}^{m-1}(f) = \{\tau(A)f : \tau \in \mathcal{P}^{m-1}, \tau(\theta_k) = 0\}$$

is a subspace of \mathcal{K}^m of dimension $m - 1$. Since the set of all vectors in \mathcal{K}^m which are orthogonal to y_k is a subspace of dimension $m - 1$ it must coincide with S_k . This establishes necessity. \square

This result yields a description of y_k . It is natural to define

$$\mu(\xi) \equiv \prod_{i=1}^m (\xi - \theta_i) \quad \text{and} \quad \pi_k(\xi) \equiv \mu(\xi)/(\xi - \theta_k). \quad (12.5)$$

From the result established in the proof of Lemma 12.3.1 follows Corollary 12.3.1.

Corollary 12.3.1. *With the definitions in (12.5)*

$$\begin{aligned} \mathbf{y}_k &= \pi_k(\mathbf{A})\mathbf{f}/\|\pi_k(\mathbf{A})\mathbf{f}\|, \\ \|\mu(\mathbf{A})\mathbf{f}\| &= \min \|\omega(\mathbf{A})\mathbf{f}\|, \end{aligned} \quad (12.6)$$

where the minimum is over all monic polynomials ω of degree m .

The proof is left as an exercise. Naturally μ is called the *minimal polynomial* of \mathbf{f} of degree m . Note also that $\|\mu(\mathbf{A})\mathbf{f}\|$ is the distance of $\mathbf{A}^m\mathbf{f}$ from \mathcal{K}^m . The Lanczos algorithm (introduced in the next chapter) computes quantities $\beta_1, \beta_2, \beta_3, \dots$ when applied to \mathbf{A} . When \mathbf{f} is the starting vector then

$$\|\mu(\mathbf{A})\mathbf{f}\| = \beta_1 \beta_2 \cdots \beta_m.$$

This quantity appeared in Corollary 7.3.1 and will occur again.

The next step is to establish a basic lemma which will be used to derive bounds on $(\theta_j - \alpha_j)$ for each $j = 1, \dots, m$. The role of \mathbf{f} in these bounds can be captured adequately by two numbers: $\angle(\mathbf{f}, \mathbf{z}_j)$ is one and the other is $\angle(\mathbf{f}, \mathcal{Z}^j)$ where $\mathcal{Z}^j = \text{span}(\mathbf{z}_1, \dots, \mathbf{z}_j)$. When $j > 1$ the latter angle may be much smaller than the former as is indicated in Figure 12.1.

Lemma 12.3.2. *Let \mathbf{h} be the normalized projection of \mathbf{f} , $\|\mathbf{f}\| = 1$, orthogonal to \mathcal{Z}^j . For each $\pi \in \mathcal{P}^{m-1}$ and each $j \leq m$ the Rayleigh quotient ρ satisfies*

$$\rho(\pi(\mathbf{A})\mathbf{f}; \mathbf{A} - \alpha_j) \leq (\alpha_n - \alpha_j) \left[\frac{\sin \angle(\mathbf{f}, \mathcal{Z}^j)}{\cos \angle(\mathbf{f}, \mathbf{z}_j)} \cdot \frac{\|\pi(\mathbf{A})\mathbf{h}\|}{\pi(\alpha_j)} \right]^2.$$

Proof. Let $\psi = \angle(\mathbf{f}, \mathcal{Z}^j)$ and let \mathbf{g} be the normalized projection of \mathbf{f} onto \mathcal{Z}^j so that

$$\mathbf{f} = \mathbf{g} \cos \psi + \mathbf{h} \sin \psi$$

is an orthogonal decomposition of f . Since \mathcal{Z}^j is invariant under A ,

$$s \equiv \pi(A)f = \pi(A)g \cos \psi + \pi(A)h \sin \psi$$

is an orthogonal decomposition of s . A little calculation yields

$$\begin{aligned} \rho(s; A - \alpha_j) &= [g^*(A - \alpha_j)\pi^2(A)g \cos^2 \psi \\ &\quad + h^*(A - \alpha_j)\pi^2(A)h \sin^2 \psi] / \|\pi(A)f\|^2. \end{aligned} \quad (12.7)$$

The eigenvalues of A are labeled so that $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$ and

- (a) $v^*(A - \alpha_j)v \leq 0$ for all $v \in \mathcal{Z}^j$, in particular for $v = \pi(A)g$,
- (b) $w^*(A - \alpha_j)w \leq (\alpha_n - \alpha_j)\|w\|^2$ for all $w \perp \mathcal{Z}^j$, in particular for $w = \pi(A)h$.

When (a) and (b) are used to simplify (12.7) it becomes

$$\rho(s; A - \alpha_j) \leq (\alpha_n - \alpha_j)[\|\pi(A)h\| \sin \psi / \|\pi(A)f\|]^2. \quad (12.8)$$

The proof is completed by using an eigenvector expansion of f

$$\begin{aligned} \|s\|^2 &= \|\pi(A)f\|^2 \\ &= \sum_{i=1}^n \pi^2(\alpha_i) \cos^2 \angle(f, z_i) \\ &\geq \pi^2(\alpha_j) \cos^2 \angle(f, z_j). \quad \square \end{aligned}$$

The inequalities (a) and (b) used to obtain (12.8) are crude, but without dragging in more information they cannot be improved. Since $\angle(f, \mathcal{Z}^j) \leq \angle(f, z_j)$ it is valid to replace the sine and cosine in the lemma by $\tan \angle(f, z_j)$ but when $j > 1$ this increases the bound unnecessarily. There is no loss of generality in assuming that all our angles are acute.

Exercises on Section 12.3

12.3.1. Prove both parts of Corollary 12.3.1. *Hint:* Consider the direction of $\mu(A)f$.

12.3.2. Show that for each $\pi \in \mathcal{P}^{m-1}$ and $j \leq m$,

$$|\tan \angle(z_j, \mathcal{K}^m)| \leq |\tan \angle(z_j, f)| \cdot \|\pi(A)f_j\| / |\pi(\alpha_j)|$$

where f_j is the normalized projection of f orthogonal to z_j .

*12.4. The Error Bounds of Kaniel and Saad

This section develops a priori error bounds on the RR approximations (θ_i, y_i) , $i = 1, \dots, m$, from $\mathcal{K}^m(\mathbf{f}; \mathbf{A})$ to the corresponding eigenpairs (α_i, \mathbf{z}_i) of \mathbf{A} . These results should be contrasted with the computable residual error bounds (Chapter 11) which relate each θ_i to some close eigenvalue $\alpha_{i'}$ of unknown index. It is not unusual to have $i' \neq i$, and other methods (see Chapter 3) are needed to determine i' . There are analogous bounds relating (θ_{-i}, y_{-i}) to $(\alpha_{-i}, \mathbf{z}_{-i})$ for $i = 1, \dots, m$ and consequently each θ_i is compared tacitly with α_i and with α_{i-m-1} for $i = 1, \dots, m$. At most one of the bounds will be small.

The error bound on $\theta_j - \alpha_j$ depends on several quantities: the starting vector \mathbf{f} , the $\angle(y_i, \mathbf{z}_i)$ for $i < j$, and the *spread* $(\alpha_n - \alpha_1)$ of \mathbf{A} 's spectrum. However, as will be seen shortly, the leading role is played by the Chebyshev polynomial T_{m-j} whose steep climb outside the interval $[-1, 1]$ helps to explain the excellent approximations obtained from Krylov subspaces. Appendix B is devoted to these polynomials.

The error bounds come from choosing a polynomial π in Lemma 12.3.2 such that, among other things,

- I. $|\pi(\alpha_j)|$ is large while $\|\pi(\mathbf{A})\mathbf{h}\|$ is small, and
- II. $\rho(\mathbf{s}; \mathbf{A} - \alpha_j) \geq 0$ where $\mathbf{s} = \pi(\mathbf{A})\mathbf{f}$.

It is I that brings in the Chebyshev polynomials. Note that

$$\begin{aligned} \|\pi(\mathbf{A})\mathbf{h}\|^2 &= \sum_{i=j+1}^n \pi^2(\alpha_i) \cos^2 \angle(\mathbf{f}, \mathbf{z}_i) / \sum_{i=j+1}^n \cos^2 \angle(\mathbf{f}, \mathbf{z}_i) \\ &\leq \max_{i>j} \pi^2(\alpha_i). \end{aligned}$$

There are $(n - j)$ α 's exceeding α_j but π has, it turns out, only $m - j$ disposable zeros and usually $m \ll n$. The best π will depend strongly on the actual distribution of \mathbf{A} 's eigenvalues. In the interests of simplicity it is customary to majorize $\|\pi(\mathbf{A})\mathbf{h}\|$ still further by

$$\|\pi(\mathbf{A})\mathbf{h}\|^2 \leq \max_{i>j} \pi^2(\alpha_i) \leq \max \pi^2(\tau) \text{ over all } \tau \text{ in } [\alpha_{j+1}, \alpha_n].$$

It is a scaled Chebyshev polynomial which minimizes the term on the right when $\pi(\alpha_j)$ is given.

Requirement II concerns the left side of the inequality in Lemma 12.3.2, namely, $\rho(\mathbf{s}; \mathbf{A} - \alpha_j)$. The following facts are known:

- (A) $0 \leq \theta_j - \alpha_j$ (from section 10.1),

(B) $\theta_j - \alpha_j \leq \rho(\mathbf{s}; \mathbf{A} - \alpha_j)$ if $\mathbf{s} \perp \mathbf{y}_i$ for all $i < j$ (from section 11.4),

(C) $\theta_j - \alpha_j \leq \rho(\mathbf{s}; \mathbf{A} - \alpha_j) + \sum_{i=1}^{j-1} (\alpha_n - \alpha_i) \sin^2 \angle(\mathbf{y}_i, \mathbf{z}_i),$

if $\mathbf{s} \perp \mathbf{z}_i$ for all $i < j$ (from section 11.9).

It is clear from (A) that if $\rho(\mathbf{s}; \mathbf{A} - \alpha_j) < 0$ then, a fortiori, $\rho(\mathbf{s}; \mathbf{A} - \alpha_j) < \theta_j - \alpha_j$ and Lemma 12.3.2 cannot be used to bound $\theta_j - \alpha_j$. Hence condition II above turns our attention to either (B), which yields Saad's bounds, or (C) which yields Kaniel's bound. We take (B) first.

Theorem 12.4.1. *Let $\theta_1 \leq \dots \leq \theta_m$ be the Ritz values derived from $\mathcal{K}^m(\mathbf{f})$ and let (α_i, \mathbf{z}_i) be the eigenpairs of \mathbf{A} . For each $j = 1, \dots, m$*

$$0 \leq \theta_j - \alpha_j \leq (\alpha_n - \alpha_j) \left[\frac{\sin \angle(\mathbf{f}, \mathcal{Z}^j)}{\cos \angle(\mathbf{f}, \mathbf{z}_j)} \cdot \frac{\prod_{\nu=1}^{j-1} \left(\frac{\theta_\nu - \alpha_n}{\theta_\nu - \alpha_j} \right)}{T_{m-j}(1+2\gamma)} \right]^2$$

and

$$\tan \angle(\mathbf{z}_j, \mathcal{K}^m) \leq \frac{\sin \angle(\mathbf{f}, \mathcal{Z}^j)}{\cos \angle(\mathbf{f}, \mathbf{z}_j)} \cdot \frac{\prod_{\nu=1}^{j-1} \left(\frac{\alpha_\nu - \alpha_n}{\alpha_\nu - \alpha_j} \right)}{T_{m-j}(1+2\gamma)},$$

where $\gamma \equiv (\alpha_j - \alpha_{j+1}) / (\alpha_{j+1} - \alpha_n)$.

Proof. Apply Lemma 12.3.2. To ensure (B) the trial vector $\pi(\mathbf{A})\mathbf{f}$ must be orthogonal to $\mathbf{y}_i, \dots, \mathbf{y}_{j-1}$. By Lemma 12.3.1 it suffices to consider polynomials π of the form

$$\pi(\xi) = (\xi - \theta_1) \cdots (\xi - \theta_{j-1}) \tilde{\pi}(\xi), \quad \tilde{\pi} \in \mathcal{P}^{m-j}.$$

Note that for such π

$$\begin{aligned}
\frac{\|\pi(A)h\|}{|\pi(\alpha_j)|} &\leq \frac{\|(A - \theta_1) \cdots (A - \theta_{j-1})\| \cdot \|\tilde{\pi}(A)h\|}{|(\alpha_j - \theta_1) \cdots (\alpha_j - \theta_{j-1})| \cdot |\tilde{\pi}(\alpha_j)|} \\
&\leq \prod_{i=1}^{j-1} \left[\frac{\alpha_n - \theta_i}{\alpha_j - \theta_i} \right] \max_{\tau} \frac{|\tilde{\pi}(\tau)|}{|\tilde{\pi}(\alpha_j)|} \\
&\quad \text{over } \tau \text{ in } [\alpha_{j+1}, \alpha_n], \tag{12.9}
\end{aligned}$$

since $h \perp \mathcal{Z}^j$. The problem has been reduced to finding $\tilde{\pi} \in \mathcal{P}^{m-j}$ that minimizes the ratio on the right side. The well-known solution (see Appendix B) is the Chebyshev polynomial adapted to $[\alpha_{j+1}, \alpha_n]$. In fact

$$\begin{aligned}
\min_{\tilde{\pi}} \max_{\tau} \frac{|\tilde{\pi}(\tau)|}{|\tilde{\pi}(\alpha_j)|} &= \frac{\max_{\tau} T_{m-j}(\tau; [\alpha_{j+1}, \alpha_n])}{T_{m-j}(\alpha_j; [\alpha_{j+1}, \alpha_n])} \\
&= \frac{1}{T_{m-j}(1 + 2\gamma)}, \tag{12.10}
\end{aligned}$$

where the *gap ratio* γ is defined in the statement of the theorem. On combining (B), Lemma 12.3.2, and the relations (12.9), (12.10) the bound on $\theta_j - \alpha_j$ is obtained. The angle bound comes from decomposing f as

$$f = \bar{g} \cos \angle(f, \mathcal{Z}^{j-1}) + z_j \cos \angle(f, z_j) + h \sin \angle(f, \mathcal{Z}^j).$$

This time π is chosen to satisfy $\pi(\alpha_i) = 0$ for $i = 1, \dots, j-1$, so that

$$s = \pi(A)f = o + z_j \pi(\alpha_j) \cos \angle(f, z_j) + \pi(A)h \sin \angle(f, \mathcal{Z}^j).$$

This is an orthogonal decomposition of s and therefore

$$\tan \angle(s, z_j) = \frac{\sin \angle(f, \mathcal{Z}^j) \|\pi(A)h\|}{\cos \angle(f, z_j) |\pi(\alpha_j)|}.$$

The proof is completed by taking the same $\tilde{\pi}$ as above; i.e.,

$$\pi(\tau) = \tilde{\pi}(\tau) \prod_{i=1}^{j-1} (\tau - \alpha_i). \quad \square$$

In order to avoid the use of the Ritz values θ_i in the bound on $\theta_j - \alpha_j$ Kaniel makes explicit reference to the Ritz vectors y_i , $i = 1, \dots, j-1$. He substitutes the same s as was used for the angle bound above, namely,

$$s = \pi(A)f = (A - \alpha_1) \cdots (A - \alpha_{j-1})\tilde{\pi}(A)f$$

into Lemma 11.9.1 which compensates, as shown in condition (C) that appears before Theorem 12.4.1, for the fact that this s does not satisfy $s^*y_i = 0$, $i < j$. The same $\tilde{\pi}$ is used as in the proof of Theorem 12.4.1 to obtain Theorem 12.4.2.

Theorem 12.4.2. *The RR approximations (θ_j, y_j) from $\mathcal{K}^m(f)$ to (α_j, z_j) satisfy*

$$\begin{aligned} 0 \leq \theta_j - \alpha_j \leq (\alpha_n - \alpha_j) & \left[\frac{\sin \angle(f, Z^j)}{\cos \angle(f, z_j)} \cdot \frac{\prod_{\nu=1}^{j-1} \left(\frac{\alpha_\nu - \alpha_n}{\alpha_\nu - \alpha_j} \right)}{T_{m-j}(1+2\gamma)} \right]^2 \\ & + \sum_{\nu=1}^{j-1} (\alpha_n - \alpha_\nu) \sin^2 \angle(y_\nu, z_\nu), \end{aligned}$$

where

$$\gamma = (\alpha_j - \alpha_{j+1}) / (\alpha_{j+1} - \alpha_n), \text{ and, from Lemma 11.9.2,}$$

$$\begin{aligned} \sin^2 \angle(y_\nu, z_\nu) \leq & \left[(\theta_\nu - \alpha_\nu) + \sum_{\mu=1}^{\nu-1} (\alpha_{\nu+1} - \alpha_\mu) \sin^2 \angle(y_\mu, z_\mu) \right] \\ & / (\alpha_{\nu+1} - \alpha_\nu). \end{aligned}$$

Theorem 12.4.1 does not give an explicit bound on $\angle(y_\nu, z_\nu)$, and to repair that omission Saad relates this angle to the smaller one $\angle(z_\nu, \mathcal{K}^m)$ which was majorized in Theorem 12.4.1. Reference to Figure 12.2 shows that

$$\sin^2 \angle(y_j, z_j) = PQ^2 + QR^2 = \sin^2 \phi + \sin^2 \omega \cos^2 \phi. \quad (12.11)$$

As indicated the scene is three dimensional— $\{y_j, u_j, w_j\}$ is an orthonormal set of vectors of which y_j and u_j are in \mathcal{K}^m while $w_j \perp \mathcal{K}^m$. The normalized projection of z_j onto \mathcal{K}^m is v_j , and the following relations are needed for the proof of the next theorem:

$$z_j = v_j \cos \phi + w_j \sin \phi, \quad (12.12)$$

$$v_j = y_j \cos \omega + u_j \sin \omega, \quad (12.13)$$

$$\phi = \angle(z_j, v_j), \omega = \angle(v_j, y_j).$$

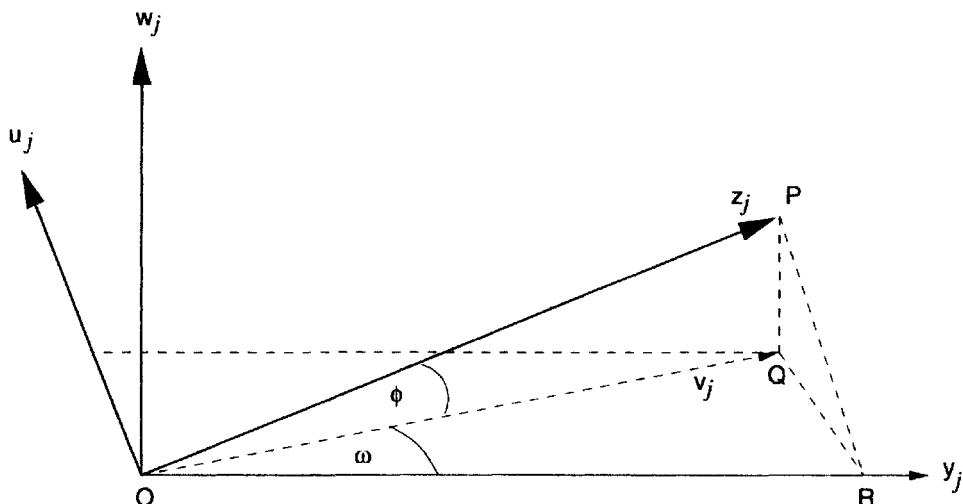


FIG. 12.2. *Projection of z_j onto \mathcal{K}^m and $\text{span}(y_j)$.*

In order to discuss Ritz vectors such as y_j we need the (orthogonal) projector onto \mathcal{K}^m . This is represented by the matrix $Q_m Q_m^*$ in section 11.4 but we will call it H here. Recall that (θ_j, y_j) is an eigenpair of the projection of A on \mathcal{K}^m , namely, the operator HA restricted to \mathcal{K}^m . The analysis which follows gives a nice example of the importance of the domain because shall also be interested in HA restricted to $(\mathcal{K}^m)^\perp$. The distinction between the two operators can be made by writing HAH for the former and $HA(I - H)$ for the latter. Theorem 12.4.3 below makes use of an interesting quantity called the *variation of \mathcal{K}^m by A* ,

$$\beta_1 \cdots \beta_m \equiv \|HA(I - H)\| = \|HA \text{ restricted to } (\mathcal{K}^m)^\perp\|. \quad (12.14)$$

It is left as Exercise 12.4.2 to show the connection to Corollary 12.3.1.

Theorem 12.4.3. Let z_j be the normalized projection of f onto the eigenspace of α_j for some $j \leq m$ and let $(\theta_i, y_i), i = 1, \dots, m$ be the Ritz approximations from \mathcal{K}^m . Then

$$\sin^2 \angle(z_j, y_j) \leq \left(1 + \frac{\beta_1^2 \cdots \beta_m^2}{\text{gap}(j; m)^2}\right) \sin^2 \angle(z_j, \mathcal{K}^m),$$

where $\text{gap}(j; m) \equiv \min |\alpha_j - \theta_i|$ over $i \neq j$, $i = 1, \dots, m$.

Proof. Using (12.12) rewrite the defining property of z_j , namely, $(A - \alpha_j)z_j = 0$ in the form

$$(A - \alpha_j)v_j \cos \phi = -(A - \alpha_j)w_j \sin \phi.$$

This vector is not in \mathcal{K}^m and the trick is to consider its projection onto \mathcal{K}^m . Hence

$$\|\mathbf{H}(A - \alpha_j)v_j \cos \phi\| = \|\mathbf{H}(A - \alpha_j)w_j \sin \phi\|. \quad (12.15)$$

The right side of (12.15) is easily majorized using (12.14) and $w_j = (I - \mathbf{H})v_j$:

$$\begin{aligned} \|\mathbf{H}(A - \alpha_j)w_j \sin \phi\| &\leq \|\mathbf{H}(A - \alpha_j)(I - \mathbf{H})\| \cdot \|w_j\| \sin \phi \\ &= \beta_1 \cdots \beta_m \sin \phi. \end{aligned} \quad (12.16)$$

Next observe that y_j is an eigenvector of $\mathbf{H}(A - \alpha_j)$ and so (see Exercise 12.4.3) premultiplication of (12.13) by $\mathbf{H}(A - \alpha_j)$ yields an orthogonal decomposition of the vector on the left of (12.15), namely,

$$\mathbf{H}(A - \alpha_j)v_j = (\theta_j - \alpha_j)y_j \cos \omega + \mathbf{H}(A - \alpha_j)u_j \sin \omega. \quad (12.17)$$

Hence

$$\|\mathbf{H}(A - \alpha_j)v_j\| \geq \|\mathbf{H}(A - \alpha_j)u_j\| \sin \omega. \quad (12.18)$$

Finally note that $u_j \in (y_j^\perp \cap \mathcal{K}^m)$ which is invariant under $\mathbf{H}(A - \alpha_j)$. The restriction of $\mathbf{H}(A - \alpha_j)\mathbf{H}$ to this subspace has eigenvalues $\theta_i - \alpha_j$, $i = 1, \dots, m$, $i \neq j$ and so

$$\|\mathbf{H}(A - \alpha_j)u_j\|^2 = \rho(u_j; [\mathbf{H}(A - \alpha_j)\mathbf{H}]^2) \geq \min_{i \neq j} (\theta_i - \alpha_j)^2 = \text{gap}(j; m)^2. \quad (12.19)$$

The inequalities (12.16), (12.18), and (12.19) applied to (12.15) yield

$$\text{gap}(j; m) \sin \omega \cos \phi \leq \beta_1 \cdots \beta_m \sin \phi \quad (12.20)$$

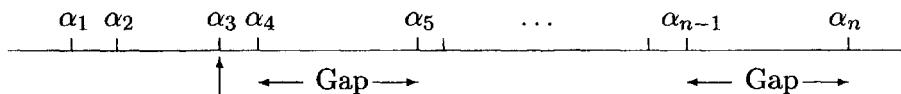
and the result follows from (12.11). \square

Exercises on Section 12.4

- 12.4.1. State carefully and prove the results dual to Theorems 12.4.1 and 12.4.2. Does γ need to be redefined?
- 12.4.2. Show that $\beta_1 \cdots \beta_m = \|\mu(\mathbf{A})f\| = \|\mathbf{H}\mathbf{A}(1 - \mathbf{H})\|$ where μ was defined in section 12.3. Since \mathcal{K}^m is a Krylov subspace the range of $\mathbf{H}\mathbf{A}(1 - \mathbf{H})$ has dimension 1.
- 12.4.3. Show that $y_j \perp \mathbf{H}(\mathbf{A} - \alpha_j)\mathbf{u}_j$ in (12.17).
- 12.4.4. Justify (12.22).
- 12.4.5. Assume that $\alpha_1 = 0, \alpha_3 = 1, \alpha_n = 1001$. For $m = 10$ and $m = 30$ determine the values of α_2 such that the bound of $\theta_1 - \alpha_1$ in Theorem 12.4.1 (or Theorem 12.4.2) is as good as the bound from (12.22) with $\mathcal{I} = \{2\}$.

*12.5. Better Bounds

The simplest choice for $\tilde{\pi}$, as made in Theorems 12.4.1 and 12.4.2, will not always give the best bound as the following figure indicates. Take $j = 3$, take α_4 close to α_3 , and take α_{n-1} well separated from α_n .



In this situation it is better to force $\tilde{\pi}(\alpha_4) = 0, \tilde{\pi}(\alpha_n) = 0$, and then fit T_{m-j-2} to $[\alpha_5, \alpha_{n-1}]$. The more that is known about \mathbf{A} , the better $\tilde{\pi}$ can be chosen. In general let $\mathcal{I} = \{j+1, j+2, \dots, k-1, l+1, \dots, n\}$ be the index set of those α 's to be taken as zeros of $\tilde{\pi}$, and define the *gap ratio* γ_{jkl} by

$$\gamma_{jkl} \equiv (\alpha_j - \alpha_k)/(\alpha_k - \alpha_l), \quad j < k < l. \quad (12.21)$$

In both theorems in section 12.4 it is legitimate to replace $T_{m-j}(1 + 2\gamma_{j,j+1,n})$ by

$$\prod_{\mu=j+1}^{k-1} \left(\frac{\alpha_j - \alpha_\mu}{\alpha_\mu - \alpha_l} \right) \prod_{\nu=l+1}^n \left(\frac{\alpha_j - \alpha_\nu}{\alpha_k - \alpha_\nu} \right) T_{m-j-|\mathcal{I}|}(1 + 2\gamma_{jkl}), \quad (12.22)$$

where $|\mathcal{I}|$ denotes the number of elements in \mathcal{I} .

All the bounds become worthless as j increases too much because there are only m θ 's to n α 's and quite soon α_j will not be the closest eigenvalue to θ_j . By the invariance properties of $\mathcal{K}^m(f)$ there is no preferred end to the spectrum. The results dual to Theorems 12.4.1 and 12.4.2 involve little more than negating indices. The precise formulation of them is a useful exercise.

Example 12.5.1. The bounds of Saad (Theorem 12.4.1) and Kaniel (Theorem 12.4.2) will be compared on an example close to one used by Kaniel in his original paper. For simplicity we majorize $\sin \angle(f, Z^j)/\cos \angle(f, z_j)$ by $\tan \phi_j$, where $\phi_j \equiv \angle(f, z_j)$ and f is the starting vector for \mathcal{K}^m with $m = 53$. The data on A 's eigenvalues are $\alpha_1 = 0$, $\alpha_2 = 0.01$, $\alpha_3 = 0.04$, $\alpha_4 = 0.1$, $\alpha_n = 1.0$.

Saad's bounds are of the form $\theta_j - \alpha_j \leq (\alpha_n - \alpha_j)[(\tan \phi_j)\kappa_j^{(m)} / T_{m-j}(1 + 2\gamma)]^2$, where $\kappa_j^{(m)}$ is a function of θ 's and α 's. In this example $\kappa_j^{(m)}$ equals the corresponding factor κ_j in Kaniel's bounds to the given accuracy. Consequently Saad's bounds are simpler and tighter.

There is no loss in taking all angles to be acute.

$$j = 1$$

$$\gamma_{12n} = \frac{1}{99}, \quad T_{52}(1 + 2\gamma) = 1.73 \times 10^4, \quad \kappa_1^{(m)} = \kappa_1 = 1.0,$$

$$\theta_1 - \alpha_1 \leq (\tan \phi_1 \times 5.77 \times 10^{-5})^2,$$

$$\sin^2 \angle(y_1, z_1) \leq (\theta_1 - \alpha_1) / (\alpha_2 - \alpha_1) \leq (\tan \phi_1 \times 5.77 \times 10^{-4})^2.$$

$$j = 2$$

$$\gamma_{23n} = \frac{1}{32}, \quad T_{51}(1 + 2\gamma) = 3.39 \times 10^7,$$

$$\kappa_2 \equiv (\alpha_n - \alpha_1) / (\alpha_2 - \alpha_1) = 100 = \kappa_2^{(m)}.$$

$$\begin{aligned} \text{Saad: } \theta_2 - \alpha_2 &\leq 0.99(\tan \phi_2 \times 2.95 \times 10^{-6})^2. \\ \text{Kaniel: } \theta_2 - \alpha_2 &\leq 0.99(\tan \phi_2 \times 2.95 \times 10^{-6})^2 \\ &\quad + 1.0(\tan \phi_1 \times 5.77 \times 10^{-4})^2. \end{aligned}$$

$$\begin{aligned} \sin^2 \angle(y_2, z_2) &\leq 33.3(\theta_2 - \alpha_2) + 1.33 \sin^2 \angle(y_1, z_1) \\ &\leq (\tan \phi_2 \times 1.70 \times 10^{-5})^2 + (\tan \phi_1 \times 3.40 \times 10^{-3})^2. \end{aligned}$$

 $j = 3$

$$\gamma_{34n} = \frac{1}{15}, \quad T_{50}(1 + 2\gamma) = 8.17 \times 10^{10},$$

$$\kappa_3 = (\alpha_n - \alpha_1)(\alpha_n - \alpha_2)/(\alpha_3 - \alpha_1)(\alpha_3 - \alpha_2) = 825 = \kappa_3^{(m)}.$$

$$\begin{aligned} \text{Saad: } \theta_3 - \alpha_3 &\leq 0.96(\tan \phi_3 \times 1.00 \times 10^{-8})^2. \\ \text{Kaniel: } \theta_3 - \alpha_3 &\leq 0.96(\tan \phi_2 \times 2.95 \times 10^{-6})^2 \\ &\quad + 0.99 \sin^2 \angle(y_2, z_2) + 1.0 \sin^2 \angle(y_1, z_1) \\ &\leq (\tan \phi_3 \times 0.98 \times 10^{-8})^2 + (\tan \phi_2 \times 1.70 \times 10^{-5})^2 \\ &\quad + (\tan \phi_1 \times 3.45 \times 10^{-3})^2. \end{aligned}$$

$$\sin^2 \angle(y_3, z_3) \leq 16.7(\theta_3 - \alpha_3) + 1.5 \sin^2 \angle(y_2, z_2) + 1.67 \sin^2 \angle(y_1, z_1).$$

Please note how the a priori bounds suffocate the a posteriori bounds under crude bounds (see Theorem 12.4.2) on the $\sin^2 \angle(y_i, z_i)$, $i = 1, \dots, j-1$.

Saad does not provide an a priori bound on (y_j, z_j) , but his a posteriori bound (Theorem 12.4.3) is

$$\sin^2 \angle(y_j, z_j) \leq (1 + \bar{\beta}_m^2 / \text{gap}^2)(\kappa_j^{(m)} / T_{m-j})^2 \tan^2 \phi_j$$

where

$$\bar{\beta}_m = \beta_1 \beta_2 \cdots \beta_m,$$

and in this example $(1 + \bar{\beta}_m^2 / \text{gap}^2)$ is very likely to be around 1 in magnitude. Chapter 13 will show how $\bar{\beta}_m$ will be computed in the course of the Lanczos algorithm. Thus Theorem 12.4.3 is much better than the bound in Theorem 12.4.2.

The next example shows that all the bounds so far are likely to be extreme overestimates.

Example 12.5.2. Improved error bounds.

This example continues the previous one but employs the more complicated choice of π described in (12.22) which replaces T_{m-j} by σT_{m-j-v} . Extra datum: $\alpha_{n-1} = 0.9$ (to illustrate the effect of forcing π to vanish at isolated zeros at *both* ends of the spectrum).

$$\boxed{j = 1}$$

$$\gamma_{1,4,n-1} = 0.125, \quad T_{49}(1 + 2\gamma) = 5.58 \times 10^{14},$$

$$\begin{aligned}\sigma &= \frac{(\alpha_1 - \alpha_2)(\alpha_1 - \alpha_3)(\alpha_1 - \alpha_n)}{(\alpha_{n-1} - \alpha_2)(\alpha_{n-1} - \alpha_3)(\alpha_4 - \alpha_n)} \\ &= \frac{1 \times 2 \times 10}{89 \times 43 \times 9} = 5.81 \times 10^{-4},\end{aligned}$$

$$\sigma \times T_{49} = 3.24 \times 10^{11},$$

$$\theta_1 - \alpha_1 \leq (\tan \phi_1 \times 3.09 \times 10^{-12})^2,$$

$$\sin^2 \angle(y_1, z_1) \leq (\tan \phi_1 \times 3.09 \times 10^{-11})^2.$$

$$\boxed{j = 2}$$

$$\gamma_{2,4,n-1} = \frac{9}{80}, \quad T_{49}(1 + 2\gamma) \approx 9.38 \times 10^{13},$$

$$\begin{aligned}\sigma &= \frac{(\alpha_3 - \alpha_2)(\alpha_n - \alpha_2)}{(\alpha_{n-1} - \alpha_3)(\alpha_n - \alpha_4)} = \frac{33}{860}, \quad \sigma \times T_{49} \approx 3.6 \times 10^{12}, \\ \kappa_2 &= \left(\frac{\alpha_n - \alpha_1}{\alpha_2 - \alpha_1} \right) = 100.\end{aligned}$$

$$\text{Saad: } \theta_2 - \alpha_2 \leq 0.99(\tan \phi_2 \times 2.76 \times 10^{-11})^2.$$

$$\begin{aligned}\text{Kaniel: } \theta_2 - \alpha_2 &\leq 0.99(\tan \phi_2 \times 2.76 \times 10^{11})^2 \\ &\quad + (\tan \phi_1 \times 3.09 \times 10^{-11})^2.\end{aligned}$$

$$\sin^2 \angle(y_2, z_2) \leq \frac{1}{3}[100(\theta_2 - \phi_2) + 4 \sin^2 \angle(y_1, z_1)].$$

$$j = 3$$

$$\gamma_{3,4,n-1} = \frac{6}{80}, \quad T_{49}(1 + 2\gamma) = 2.26 \times 10^{11},$$

$$\sigma = \left(\frac{\alpha_3 - \alpha_n}{\alpha_4 - \alpha_n} \right) = \frac{16}{15}, \quad \sigma \times T_{49} \approx 2.41 \times 10^{11},$$

$$\kappa_3 = \frac{(\alpha_n - \alpha_1)(\alpha_n - \alpha_2)}{(\alpha_3 - \alpha_1)(\alpha_3 - \alpha_2)} = 825.$$

$$\text{Saad: } \theta_3 - \alpha_3 \leq 0.96(\tan \phi_3 \times 3.41 \times 10^{-9})^2.$$

$$\begin{aligned} \text{Kaniel: } \theta_3 - \alpha_3 &\leq 0.96(\tan \phi_3 \times 3.41 \times 10^{-9})^2 \\ &+ [\tan \phi_2 \times 2.76 \times 10^{-10}]^2 + [\tan \phi_1 \times 6.18 \times 10^{-11}]^2. \end{aligned}$$

$$\sin^2 \angle(y_3, z_3) \leq 16.7(\theta_3 - \alpha_3) + 1.5 \sin^2 \angle(y_2, z_2) + 1.67 \sin^2 \angle(y_1, z_1).$$

12.6. Comparison with the Power Method

After $(m - 1)$ steps of the power method the best approximation to an eigenvalue is $\rho(A^{m-1}f)$. Lemma 12.3.2 can be applied to give an error bound provided that $\pi(\xi) = \xi^{m-1}$. The power method is not translation invariant and $A^{m-1}f/\|A^{m-1}f\|$ converges to either z_1 or z_n as $m \rightarrow \infty$. When $\|A\| = -\alpha_1$ the limit is z_1 , and the bounds analogous to Theorem 12.4.2 are

$$\begin{aligned} \rho(A^{m-1}f) - \alpha_1 &\leq (\alpha_n - \alpha_1)[\tan \angle(f, z_1)/(\alpha_1/\alpha_2)^{m-1}]^2, \\ \sin^2 \angle(A^{m-1}f, z_1) &\leq (\rho(A^{m-1}f) - \alpha_1)/(\alpha_2 - \alpha_1). \end{aligned} \quad (12.23)$$

A cleaner comparison with Krylov subspaces emerges if we consider a matrix in which $\|A\| = \alpha_n$ and the smallest eigenvalue α_1 is approximated by using the power method with $A - \alpha_n$. The bounds become

$$\rho((A - \alpha_n)^{m-1}f) - \alpha_1 \leq (\alpha_n - \alpha_1)[\tan \angle(f, z_1)/(1 + \gamma_{12n})^{m-1}]^2,$$

where γ_{12n} is defined in (12.21), and

$$\sin^2 \angle((A - \alpha_n)^{m-1}f, z_1) \leq \{\rho[(A - \alpha_n)^{m-1}f] - \alpha_1\}/(\alpha_2 - \alpha_1).$$

(12.24)

Comparison of Theorem 12.4.1 with (12.24) shows that both angle bounds contain the factor $(\alpha_n - \alpha_1)/(\alpha_2 - \alpha_1)$ which is needed to cope with perversely chosen or unfortunate f 's. It is possible to replace the simple expression $(\alpha_n - \alpha_1)/(\alpha_2 - \alpha_1)$ with a more complicated one involving all the $\angle(f, z_i)$, $i = 1, \dots, n$, and when f is reasonably well chosen the simple expression is a severe overestimate. Table 12.1 summarizes the bounds on the ratio (final error angle)/ $\angle(f, z_j)$ in terms of the relative gap ratio γ . Appendix B gives the estimates for the Chebyshev polynomials T_m . Recall that for small γ

$$T_k(1 + 2\gamma) \approx \frac{1}{2}(1 + 2\sqrt{\gamma} + 2\gamma)^k.$$

It is the $2\sqrt{\gamma}$ which makes the difference in difficult problems when γ is small (like 10^{-4}). When γ is large (like 10^2) then m will be small (like 1 or 2) and the methods coalesce.

TABLE 12.1
Power method versus Krylov subspace method.

<i>Power method: $(A - \alpha_n)^m f$</i>
$(1 + \gamma)^{-m} \approx \begin{cases} e^{-m\gamma} & (\gamma \rightarrow 0) \\ \gamma^{-m} & (\gamma \rightarrow \infty) \end{cases}$
$\mathcal{K}^m(f)$
$[T_m(1 + 2\gamma)]^{-1} \approx \begin{cases} 2e^{-2m\sqrt{\gamma}} & (\gamma \rightarrow 0) \\ 2(4\gamma)^{-m} & (\gamma \rightarrow \infty) \end{cases}$

The basis has been laid for a comparison of approximations from $\mathcal{K}^m(f)$ with various other methods. The block power method starts with an n -by- p matrix F and, for an extra cost, computes approximations from $\text{span}((A -$

$\alpha_n)F$). The effect on the bounds in (12.24) is to replace γ_{12n} by $\gamma_{1,p+1,n}$ and $\angle(f, z_1)$ by $\angle(f, Z^p)$. If the change in angle is ignored it is clear that the block power method produces results as good as does $K^m(f)$ when $\gamma_{1,p+1,n}/\gamma_{1,2,n} = 2/\sqrt{\gamma_{1,2,n}}$. For equispaced α_i equality is achieved only when $p = 2n/(\sqrt{n} + 2) \approx 2\sqrt{n}$, an impractically large blocksize.

It is also possible to compare inverse iteration with $K^m(f)$ although such a comparison is somewhat unnatural because if it is possible to compute $(A - \sigma)^{-m}f$ then it is also possible to use $K^m(f; (A - \sigma)^{-1})$. If σ is close to an eigenvalue, $\sigma < \alpha_1$, say, then the error bound (12.24) still applies provided that γ_{12n} is replaced by $\tilde{\gamma} = (\alpha_2 - \alpha_1)/(\alpha_1 - \sigma)$. (Why?) So inverse iteration will produce approximations as good as those from $K^m(f; A)$, for equispaced α_i , whenever $\ln \tilde{\gamma} > 2/\sqrt{n}$. If subspace iteration, or block inverse iteration, is used in the same equispaced problem then its approximation, from $\text{span}[(A - \sigma)^{-m+1}F]$ will better $y_1^{(m)}$ whenever $\ln p\hat{\gamma} > 2/\sqrt{n}$. This indicates how powerful inverse iteration can be.

So far the cost of the various techniques has been ignored, as has the important fact that $K^m(f)$ produces approximations to several eigenpairs of A . There may be cases in which A is so large and intractable that vector multiplication is the only thing that can be done with A ; there certainly are cases in which A , though large, does permit the occasional solution of $(A - \sigma)u = v$ for u but at a cost significantly greater than one product Av ; there are also cases in which the solution of $(A - \sigma)u = v$ takes less time than say three products Av . Another aspect is the storage requirement and it is time to take up the problem of computing the RR approximations from $K^m(f)$; see section 12.7. The table in section 12.1 indicates that for fairly difficult cases m will have to exceed 25 and may rise into the 100's.

Some might say that we have been unfair to the power method. If the optimal shift $(\alpha_2 + \alpha_{-1})/2$ were used instead of α_{-1} then the convergence factor would change from $(1 + \gamma)^{-1}$ to $(1 + 2\gamma)^{-1}$. In practice this optimal shift is more difficult to estimate than is α_{-1} and only changes the convergence rate by a factor of 2.

Exercise on Section 12.6

- 12.6.1. Let $\alpha_i = \bar{\beta}^i$, $i = 1, \dots, n$, $\bar{\beta} > 0$. Take $n = 100$ and determine m such that

$$\sin(y_i, z_i) \leq \sin(z_i, f)/100 \text{ for (a) } i = 1, 2 \text{ and (b) } i = -1, -2.$$

12.7. Partial Reduction to Tridiagonal Form

There is an intimate connection between Krylov subspaces and tridiagonal matrices. Let us start with $\mathcal{K}^m(f, A)$. For theoretical work, such as the Saad–Kaniel theory, the natural basis for \mathcal{K}^m is the Krylov basis $K^m(f, A)$ of section 12.1. For practical work there is a *distinguished orthonormal basis*, $Q_m \equiv (q_1, \dots, q_m)$, which is the result of applying the Gram–Schmidt orthonormalizing process to the columns of $K^m(q_1)$ in the natural order f, Af, \dots . Here the dimension of \mathcal{K}^m must be m ; i.e., $K^m(q_1)$ must have full rank. We will not consider any m with $\mathcal{K}^{m+1} = \mathcal{K}^m$. For reasons which become clear in Chapter 13 we call Q_m the *Lanczos basis* of $\mathcal{K}^m(f; A)$. In matrix terminology we can write

$$K_m(f) = Q_m C_m^{-1} \quad (12.25)$$

as the QR factorization of K_m . The columns of the upper-triangular matrix C_m contain the coefficients of the Lanczos polynomials introduced in Chapter 7 (but not in monic form), although this fact will not be exploited here.

For general vector sequences the Gram–Schmidt process is quite expensive and becomes more so as the number of vectors increases. In contrast, for Krylov sequences $K^m(f)$ the process simplifies dramatically to yield a three-term recurrence connecting the columns of $Q_m \equiv (q_1, \dots, q_m)$.

Before tackling the general case we consider an important example. Observe first that since $q_3 \in \mathcal{K}^3(q_1)$,

$$\begin{aligned} \text{span}(q_1, q_2, q_3, Aq_3) &= \text{span}(q_1, q_2, q_3, A(\gamma_2 A^2 + \gamma_1 A + \gamma_0 I)q_1) \\ &= \text{span}(q_1, q_2, q_3, A^3 q_1) \\ &= \mathcal{K}^4(q_1). \end{aligned}$$

Thus to complete the basis for \mathcal{K}^4 it suffices to orthogonalize Aq_3 against q_3, q_2 , and q_1 . It turns out that Aq_3 is already orthogonal to q_1 (Exercise 12.7.1).

Theorem 12.7.1. *When $\mathcal{K}^m(f; A)$ has full rank and Q_m is defined by (12.25) then $Q_m^* A Q_m$ is an unreduced tridiagonal matrix.*

Proof. The characteristic property of $\mathcal{K}^m(f)$ is that for each $j < m$, $A\mathcal{K}^j \subset \mathcal{K}^{j+1}$. In particular $q_i \perp \mathcal{K}^{i-1}$ and $Aq_j \in \mathcal{K}^{j+1}$. Consequently

$$q_i^*(Aq_j) = 0 \quad \text{for each } i > j + 1. \quad (12.26)$$

By the symmetry of A , $q_j^*(Aq_i) = q_i^*(Aq_j) = 0$ for all $j < i - 1$. This establishes the tridiagonal nature of $Q_m^*AQ_m$. Note that

$$\begin{aligned} \mathcal{K}^{j+1} &= \text{span}(K^j, A^j f) \\ &= \text{span}(K^j, Aq_j) \\ &= \text{span}(K^j, q_{j+1}). \end{aligned}$$

If K^m has full rank and $j < m$ then Aq_j and q_{j+1} cannot be orthogonal; i.e., $q_{j+1}^*(Aq_j) \neq 0$. \square

If we write $\alpha_j = q_j^*Aq_j$, $\beta_j = q_{j+1}^*Aq_j$, then the tridiagonal matrix $Q_m^*AQ_m$ will be denoted by

$$T_m = T_{1:m} = \left[\begin{array}{cccccc} \alpha_1 & \beta_1 & & & & & \\ \beta_1 & \alpha_2 & \beta_2 & & & & \\ & \beta_2 & \ddots & \ddots & & & \\ & & \ddots & \ddots & \beta_{m-1} & & \\ & & & & \beta_{m-1} & \alpha_m & \end{array} \right]. \quad (12.27)$$

With this extra notation we can give a useful consequence of the theorem in Corollary 12.7.1.

Corollary 12.7.1 For each $j < m$,

$$\boxed{\mathbf{A}} \quad \boxed{\mathbf{Q}_j} \quad - \quad \boxed{\mathbf{Q}_j} \quad = \quad \boxed{\mathbf{T}_j} \quad \boxed{\mathbf{0}}$$

where the last column on the right is $\mathbf{r}_j \equiv \mathbf{q}_{j+1}\beta_j$. (12.28)

Proof. The verification that for $i \leq j$,

$$\mathbf{A}\mathbf{q}_i = \mathbf{q}_{i-1}\beta_{i-1} + \mathbf{q}_i\alpha_i + \mathbf{q}_{i+1}\beta_i, \quad (12.29)$$

is left as Exercise 12.7.2. The corollary merely expresses this in compact form. It is only necessary to note that there are two different formulas for β_j ; $\beta_j = \mathbf{q}_{j+1}^* \mathbf{A} \mathbf{q}_j = \|\mathbf{r}_j\|$. \square

The matrix on the right in (12.28) can be written as $\mathbf{r}_j \mathbf{e}_j^*$ where $\mathbf{e}_j^* = (0, \dots, 0, 1)$ has only j elements while all the other vectors have n elements.

Observe that the fundamental Krylov matrix $\mathbf{K}^m(\mathbf{f})$ has faded from the picture and \mathbf{Q}_j is directly related to \mathbf{A} in (12.28). The corollary suggests a way in which all the \mathbf{Q}_j and \mathbf{T}_j , $j = 1, \dots, m$ can be built up from \mathbf{A} and $\mathbf{Q}_1 = \mathbf{q}_1 = \mathbf{f}/\|\mathbf{f}\|$. At the beginning of the j th step, $\mathbf{T}_{j-1}, \beta_{j-1}$, and \mathbf{Q}_j are on hand.

From (12.28),

$$\begin{aligned} \mathbf{r}_j &= \mathbf{r}_j \mathbf{e}_j^* \mathbf{e}_j = (\mathbf{A} \mathbf{Q}_j - \mathbf{Q}_j \mathbf{T}_j) \mathbf{e}_j \\ &= \mathbf{A} \mathbf{q}_j - \mathbf{q}_{j-1} \beta_{j-1} - \mathbf{q}_j \alpha_j, \end{aligned} \quad (12.30)$$

where

$$\alpha_j = \mathbf{q}_j^* (\mathbf{A} \mathbf{q}_j), \quad \text{since the } \mathbf{q}'\text{s are orthogonal.} \quad (12.31)$$

Next

$$\beta_j = \|\mathbf{q}_{j+1}\beta_j\| = \|\mathbf{r}_j\| \quad \text{from (12.28).} \quad (12.32)$$

If $\beta_j > 0$ then \mathbf{q}_{j+1} is just \mathbf{r}_j/β_j and the step is complete. If $\beta_j = 0$ then $\mathbf{A}\mathbf{Q}_j = \mathbf{Q}_j\mathbf{T}_j$ and the algorithm halts with

$$\text{span } \mathbf{Q}_j = \mathcal{K}^j(\mathbf{f}) = \mathcal{K}^{j+1}(\mathbf{f}) = \mathcal{J}(\mathbf{f}), \quad (12.33)$$

the smallest invariant subspace containing \mathbf{f} . In the 1950s, when the goal was to compute \mathbf{T}_n , the possibility that $\beta_m = 0$ for $m < n$ was regarded as a mild nuisance. Today, seeking a few eigenvectors, early termination is an outcome devoutly to be wished for because then each eigenvalue of \mathbf{T}_m is an eigenvalue of \mathbf{A} . In the extreme case, when \mathbf{f} is an eigenvector the algorithm will halt after one step.

Formula (12.29) is the well-known three-term recurrence relating the columns of \mathbf{Q}_m .

The cost of the step is dominated by the formation of $\mathbf{A}\mathbf{q}_j$, the only appearance of \mathbf{A} in the algorithm.

Exercises on Section 12.7

- 12.7.1. Show directly, using $\mathbf{A}^* = \mathbf{A}$ and $\mathbf{q}_i^* \mathbf{q}_k = 0$, $i \neq k$, that $\mathbf{A}\mathbf{q}_3$ is orthogonal to \mathbf{q}_1 .
- 12.7.2. Show that (12.28) and (12.29) are equivalent and also express Theorem 12.7.1 with the notation of (12.27). Show that $\beta_j = \mathbf{q}_{j+1}^* \mathbf{A}\mathbf{q}_j = \|\mathbf{r}_j\|$.
- 12.7.3. Compute \mathbf{T}_3 when

$$\mathbf{A} = \begin{bmatrix} 7 & 5 & 3 \\ 5 & 3 & 1 \\ 3 & 1 & 0 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}.$$

Does $\beta_3 = \|\mathbf{r}_3\| = 0$?

- 12.7.4. Let the matrix \mathbf{C}_m that is implicitly defined in (12.25) be written as $\Delta_m \mathbf{U}_m$ where \mathbf{U}_m is upper triangular with 1's on the diagonal and $\Delta_m = \text{diag}(\mathbf{C}_m)$. Show that the j th column of \mathbf{U}_m holds the coefficient of the j th Lanczos polynomial defined in Chapter 7, section 7.3.

Notes and References

The idea of the power method is a very natural one. It arises, in disguised form, in many branches of science and engineering. The civil engineers call it *Stodola's iteration*. In [Krylov, 1931] the sequence $\{\mathbf{x}, \mathbf{Ax}, \mathbf{A}^2\mathbf{x}, \dots\}$ is actually used as a clever way to find the coefficients of the characteristic polynomial and,

despite that unfortunate goal, Krylov's name has become securely attached to the sequence.

The chapter presents the bounds in [Kaniel, 1966] (with corrections from [Paige, 1971]) which show what very good eigenvalue approximations can be obtained from Krylov subspaces of modest dimension. Our aim was to make the results intelligible as well as quotable. The original proofs in [Saad, 1980] of Theorem 12.4.1 have been simplified in order to blend with the neighboring material.

Lanczos Algorithms

13.1. Krylov + Rayleigh–Ritz = Lanczos

The Lanczos algorithm has had a checkered history since its debut in 1950. Although Lanczos pointed out that his method could be used to find a few eigenvectors of a symmetric matrix it was heralded at that time as a way to reduce the whole matrix to tridiagonal form. In this capacity it flopped unless very expensive modifications were incorporated. Twenty years later Paige showed that despite its sensitivity to roundoff the simple Lanczos algorithm is nevertheless an effective tool for computing some outer eigenvalues and their eigenvectors.

The exact algorithm can be presented in various ways. In fact it has already made its appearance in section 7.2 where the constructive proof that $\mathbf{q}_1 = \mathbf{Q}\mathbf{e}_1$ completely determines the reduction of \mathbf{A} to tridiagonal form $\mathbf{T} (= \mathbf{Q}^*\mathbf{A}\mathbf{Q})$ is simply a description of the Lanczos algorithm.

A different approach is given in section 12.7 which shows that the columns of $\mathbf{Q}_m = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m)$ form a distinguished basis for the Krylov subspace $\mathcal{K}^m(\mathbf{q}_1)$ whose virtues are extolled at length in Chapter 12. In this basis \mathbf{A} 's projection onto $\mathcal{K}^m(\mathbf{q}_1)$ is represented by a tridiagonal matrix \mathbf{T}_m . The algorithm is summarized by two equations, at step j , namely,

$$\mathbf{A}\mathbf{Q}_j - \mathbf{Q}_j\mathbf{T}_j = \mathbf{r}_j\mathbf{e}_j^*, \quad \text{where } \mathbf{r}_j = \mathbf{q}_{j+1}\beta_j, \quad (13.1)$$

and

$$\mathbf{I} - \mathbf{Q}_j^*\mathbf{Q}_j = \mathbf{O}. \quad (13.2)$$

A third approach is to see the Lanczos algorithm as the natural way to implement the Rayleigh–Ritz procedure (RR in section 11.3) on the *sequence* of Krylov subspaces $\mathcal{K}^j(\mathbf{f})$, $j = 1, 2, \dots$. At each step the subspace dimension grows by one, and the best approximate eigenvectors in the subspace are

computable in a straightforward manner. The general and rather costly RR procedure is dramatically simplified when not one but a sequence of Krylov subspaces is used. Let us see how this economy comes about.

The first move in RR is to determine an orthonormal basis for the subspace but, in our case, at step j the basis $\{q_1, \dots, q_{j-1}\}$ of \mathcal{K}^{j-1} is already computed and only one vector need be added. This vector q_j has to be the component of Aq_{j-1} orthogonal to \mathcal{K}^{j-1} (see section 12.7) and, as we shall see below, q_j is already on hand but not in normalized form. Thus the first step is a simple normalization.

The next two steps in RR compute the Rayleigh quotient matrix $\rho(Q_j) \equiv Q_j^*(AQ_j)$. In our case, by Theorem 12.7.1, $\rho(Q_j)$ (i.e., T_j) is tridiagonal and, by (12.31) and (12.32), is formed from T_{j-1} by adding the elements β_j and α_j in the appropriate positions. Again it turns out that β_{j-1} is on hand and so the first real computation is the formation of $\alpha_j = q_j^* u_j$ and $u_j = Aq_j$. We mention here that in most applications A stands for a complicated program, not a square array. Typically the program might solve $(M - \sigma)u_j = q_j$ for u_j given M and σ .

Next in RR comes the computation of as many eigenvalues and eigenvectors $(\theta_i^{(j)}, s_i^{(j)})$ of T_j as are required. The tridiagonal form facilitates this step considerably (see Chapter 8). In section 11.4 it is shown that the collectively best approximate eigenpairs from \mathcal{K}^j are the Ritz pairs $(\theta_i^{(j)}, y_i^{(j)})$ where $y_i^{(j)} = Q_j s_i^{(j)}$, $i = 1, \dots, j$. We shall drop the superscript (j) whenever there is no risk of confusion.

That is the end of the RR procedure except for assessing the accuracy of the Ritz pairs, the topic of section 13.2. For the assessment it is necessary to orthogonalize u_j ($= Aq_j$) against q_j and q_{j-1} and so obtain a useful residual vector r_j . It turns out that β_j ($\equiv \|r_j\|$) is needed both in section 13.2 and at the next step of the Lanczos algorithm. As a bonus r_j is a multiple of q_{j+1} . Now everything is ready for the RR approximations from \mathcal{K}^{j+1} .

For completeness we lay out the computations in one step of the process. It is left as Exercise 13.1.1 to explain why the algorithm differs slightly from the description given above.

13.1.1. Simple Lanczos

r_0 is given, $\beta_0 = \|r_0\| \neq 0$. For $j = 1, 2, \dots$ repeat:

1. $q_j \leftarrow r_{j-1}/\beta_{j-1}$.
2. $u_j \leftarrow Aq_j$.

3. $\mathbf{r}_j \leftarrow \mathbf{u}_j - \mathbf{q}_{j-1}\beta_{j-1}$ ($\mathbf{q}_0 = \mathbf{o}$).

4. $\alpha_j \leftarrow \mathbf{q}_j^* \mathbf{r}_j.$

5. $\mathbf{r}_j \leftarrow \mathbf{r}_j - \mathbf{q}_j\alpha_j.$

6. $\beta_j \leftarrow \|\mathbf{r}_j\|.$

7. Compute θ_i , \mathbf{s}_i , \mathbf{y}_i as desired.

8. If satisfied then stop.

The important observations for large matrix calculations are given in italics below.

- After substep 3 above \mathbf{q}_{j-1} may be put into a secondary storage device. It is not needed again until some Ritz vector $\mathbf{y}_i^{(m)}$ is to be formed, from $\mathbf{y}_i^{(m)} \equiv Q_m \mathbf{s}_i$, at say the m th Lanczos step when $\mathbf{y}_i^{(m)}$ has converged sufficiently.

Only three n -vectors are needed in the fast store, an attractive feature when $n > 10^3$. Paige pointed out that it is often possible to get away with two n -vectors. See Exercise 13.1.2.

- Frequently \mathbf{A} is not a conventional array of the form $\mathbf{A}(1 : n, 1 : n)$ but the symbol for a user-written program, perhaps very complicated, that returns a vector \mathbf{Ax} for any given \mathbf{x} . Thus \mathbf{A} is indeed a linear operator.

The Kaniel-Saad theory (section 12.4) suggests that good RR approximations to the outer eigenvalues and eigenvectors will emerge for j as small as $2\sqrt{n}$, and it is for such calculations that the Lanczos algorithm is ideally suited. It is not necessary to fix the last step $j (= m)$ in advance. The process goes on until the wanted Ritz pairs $(\theta_i, \mathbf{y}_i^{(j)})$, $i = 1, \dots, p$ are deemed satisfactory. In exact arithmetic this must occur by $j = n$ but usually it will be much sooner. Typical values to bear in mind are $p = 10$, $m = 200$, $n = 10^4$.

- The initial vector \mathbf{f} (or \mathbf{q}_1) is best selected by the user to embody any available knowledge concerning \mathbf{A} 's wanted eigenvectors. In the absence of such knowledge either a random vector or $(1, 1, \dots, 1)^*$ is used for \mathbf{f} .

In order to specify the Lanczos process completely we must say when the algorithm should bother to compute some eigenvalues of \mathbf{T}_j . The rather surprising answer is that it should do so at every step. The next section, which continues to assume that exact arithmetic is being used, explains why.

Exercises on Section 13.1

- 13.1.1. Steps 3 to 5 of the algorithm do not correspond to the verbal description of the Lanczos process in section 13.1. Show that the two versions are equivalent in exact arithmetic. Explain the differences. *Hint:* Modified Gram–Schmidt (section 6.7) versus Gram–Schmidt.
- 13.1.2. Assume that the operation $u \leftarrow Av$ is implemented in the form $u \leftarrow u + Av$. Rearrange the simple algorithm to use only two n -vectors (q and r say). *Hint:* A swap operation on vectors will be needed.

13.2. Assessing Accuracy

The Kaniel–Saad theorems (Chapter 12) tells us to expect increasingly good approximations to the outer eigenvalues from $\mathcal{K}^j(q_1)$ as j increases. In practice we need a posteriori bounds to apply in each specific case. The gap theorems in section 11.7 show that the residual norm $\|Ay - y\theta\|$ is a good measure of the accuracy of the RR pair (θ, y) .

In principle it is possible to compute θ_i and y_i from T_j at each step in the Lanczos algorithm. Fortunately it is possible to compute $\|Ay_i - y_i\theta_i\|$ without computing y_i ! To see how let us drop the subscript i and observe that

$$\begin{aligned}\|Ay - y\theta\| &= \|AQs - Qs\theta\|, \quad \text{since } y = Qs, \\ &= \|(AQ - QT)s\|, \quad \text{since } s\theta = Ts, \\ &= \|(\beta_j q_{j+1} e_j^*)s\|, \quad \text{using (13.1),} \\ &= \beta_j |e_j^* s|, \quad \text{since } \|q_{j+1}\| = 1.\end{aligned}\tag{13.3}$$

So the bottom elements of the normalized eigenvectors of T_j signal convergence and there is no need to form y until its accuracy is satisfactory. This result explains why some Ritz values can be very accurate even when β_j is not small.

Example 13.2.1. Dwindling eigenvectors of T

$S = (s_{ij})$ is the matrix of normalized eigenvectors of a random symmetric tridiagonal matrix.

$$\tilde{S} = (\tilde{s}_{ij}); \quad \tilde{s}_{ij} = -\log_{10} |s_{ij}|.$$

If each entry ν in \tilde{S} , shown below, is replaced by $10^{-\nu}$ then the new table gives the absolute values of the elements of S . The bottom row of this S has some elements as small as 10^{-13} and these are associated with the isolated eigenvalues λ_1 and λ_{25} as shown in Table 13.1 and Figure 13.1. This phenomenon is typical.

TABLE 13.1
The matrix \tilde{S} .

j	1	2	3	4	5	...	11	12	13	14	15	...	21	22	23	24	25
\tilde{s}_j	6	3	1	4	1	.	2	2	1	1	1	.	2	7	3	4	6
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	
	4	3	4	2	2	.	2	1	1	1	1	.	3	2	4	4	5
	5	3	4	2	3	.	2	1	1	1	1	.	3	2	4	4	5
	6	4	5	1	4	.	2	1	1	2	1	.	4	1	5	5	7
	7	4	5	1	4	.	2	1	1	2	1	.	4	1	5	5	7
	7	5	5	1	4	.	2	1	1	2	1	.	4	1	6	6	8
	8	5	5	1	3	.	2	1	1	3	1	.	5	2	7	7	9
	10	7	6	2	5	.	3	2	1	1	1	.	6	3	8	9	11
	11	7	7	3	5	.	3	2	1	1	1	.	6	4	9	9	12
	11	7	7	3	5	.	3	2	1	1	1	.	6	4	10	10	12
25	13	9	9	4	7	.	4	2	1	1	1	.	8	6	11	12	14

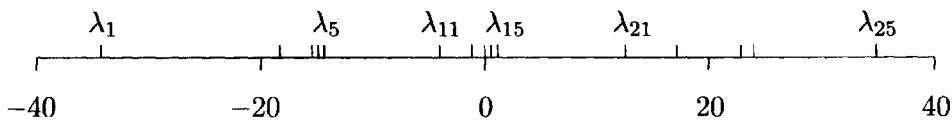


FIG. 13.1. *Eigenvalues.*

Let $s_{ji}^{(j)} \equiv \mathbf{e}_j^* \mathbf{s}_i$ and define the useful numbers

$$\beta_{ji} \equiv \beta_j |s_{ji}^{(j)}|, \quad i = 1, \dots, j. \quad (13.4)$$

Without loss of generality we may take $s_{ji}^{(j)} > 0$ for all $i \leq j$. At the j th step of Lanczos, by Theorem 4.5.1

$$|\theta_i - \lambda[\mathbf{A}]| \leq \beta_{ji} \quad (13.5)$$

for some eigenvalue $\lambda[\mathbf{A}]$ which depends on i . Moreover, by using the θ_k as approximate eigenvalues in order to compute a gap $\gamma_i \equiv \min_k |\theta_i - \theta_k|$ over $k \neq i$, Theorem 11.7.1 can be invoked to give ultimately realistic estimates (not bounds)

$$|\sin \angle(y_i^{(j)}, \mathbf{z})| \doteq \beta_{ji} / \gamma_i \quad (13.6)$$

for the Ritz vectors, and

$$|\theta_i - \lambda| \doteq \beta_{ji}^2 / \gamma_i \quad (13.7)$$

for the Ritz values, where (λ, z) is an eigenpair of A .

When several intervals $[\theta_i - \beta_{ji}, \theta_i + \beta_{ji}]$, $i = l, l+1, l+2$, say, happen to overlap then Theorem 11.5.1 can and should be used; namely, let $\sigma \equiv \sum_{i=l}^{l+2} \beta_{ji}$; then each interval $[\theta_i - \sigma, \theta_i + \sigma]$, $i = l, l+1, l+2$ contains its own eigenvalue (Exercise 13.2.1).

These estimates and bounds are attractive, but they suggest that some eigenvectors $s_i^{(j)}$ of T_j must be computed in order to form β_{ji} . Although such a computation requires only $O(j)$ ops it is worth mentioning that there are at least two ways of computing $s_{ji}^{(j)}$ without computing all of $s_i^{(j)}$. Of course these alternatives also require $O(j)$ ops and all three techniques are satisfactory. See Exercises 13.2.2 and 13.2.3.

A key issue is the cost of computing the Ritz values θ_i . By using the root-free QL algorithm with judicious shifts *all* the θ 's can be computed in approximately $9j^2$ multiplications. On the other hand by combining the QL algorithm with spectrum slicing the p extreme eigenvalues at each end can be computed in a total of approximately $36pj$ multiplications. Here we allow 1.8 QL iterations per eigenvalue and $10j$ multiplications per QL iteration. The count of $36pj$ is so low that it is a negligible fraction of the cost of one Lanczos step when $j \ll n$.

It is possible to create more sophisticated procedures that *update* the outermost Ritz values $\theta_i^{(j)}$, at each step, with $O(1)$ ops per given Ritz value. See [Nour-Omid and Parlett, 1985].

These a posteriori error estimates help keep down the number of Lanczos steps by terminating the algorithm as soon as it attains the desired accuracy.

Exercises on Section 13.2

- 13.2.1. Show why $\sigma = \Sigma \beta_{ji}$ is a bound on the norm of the residual matrix $AY - Y\Theta$ where $Y = (y_l, y_{l+1}, y_{l+2})$, $\Theta = \text{diag } (\theta_l, \theta_{l+1}, \theta_{l+2})$.
- 13.2.2. Use the results in section 7.9 to compute s_{ji}^2 assuming that the old θ 's $\theta_i^{(j-1)}$, $i = 1, \dots, j-1$ are saved along with the $\theta_i^{(j)}$.
- 13.2.3. It is possible to modify the inner loop of the standard QL algorithm so that the vector $e_j^* S$ is updated by each plane rotation. How many ops will be added to the inner loop?

13.3. The Effects of Finite Precision Arithmetic

The preceding sections paint a very rosy picture of the Lanczos algorithm. However, it was known to Lanczos, when he presented the algorithm in 1950, that the computed quantities could deviate greatly from their exact counterparts. The second basic equation, $Q_j^* Q_j = I_j$, is utterly destroyed by roundoff and the algorithm has been described, somewhat unfairly, as unstable. What makes the practical algorithm interesting is that despite this gross deviation from the exact model, it nevertheless delivers fully accurate RR approximations. Its fault is in not quitting while it is ahead because it continues to compute many redundant copies of each Ritz pair.

In order to analyze the process without drowning in irrelevant details we make an important, and standard, change of notation. From now on Q_j and T_j denote the quantities stored in the computer under these names. No further attention will be paid to their Platonic images. Moreover the vector $y_i^{(j)}$ ($\equiv Q_j s_i^{(j)}$) will be called a “Ritz vector” even when Q_j is far from orthonormal. The quotes remind us that it is not a true Ritz vector from $\text{span } Q_j$.

We will now describe the curious way the Lanczos algorithm behaves in practice and also give an example, before embarking on the analysis.

For the first few steps, maybe 3, maybe 30, the results are indistinguishable from the exact process. Then a new Lanczos vector q fails to be orthogonal, to working precision, to its predecessors. A few steps later Q_j does not even have full rank; i.e., the Lanczos vectors are linearly dependent. This looks like disaster because there is then no guarantee that T_j will bear any useful relation to A . Nevertheless an odd thing happens; at the same time as orthogonality among the $\{q_i, i = 1, \dots, j\}$ disappears a “Ritz pair” $(\theta_i^{(j)}, y_i^{(j)})$, for some i , converges to an eigenpair of A . As the algorithm proceeds further it “forgets” that it has found that eigenpair and starts to compute it again. Soon there will be two “Ritz pairs” accurately approximating that single eigenpair of A and hence the two Ritz vectors must be multiples of each other—almost—and this can only happen if Q_j has linearly dependent columns—almost.

The exact Lanczos algorithm must terminate ($\beta_j = 0$) for some $j \leq n$, but in practice the process grinds on forever, computing more copies of outer eigenvectors for each new inner pair it discovers.

The loss of orthogonality makes itself felt in all parts of the algorithm. For example, the relation $\|Ay_i - y_i\theta_i\| = \beta_{ji}$ no longer holds. Instead we have

$$|\theta_i - \lambda[A]| \leq \|Ay_i - y_i\theta_i\| / \|y_i\| \leq (\beta_{ji} + \|F_j\|) / \|y_i\| \quad (13.8)$$

where F_j accounts for roundoff and is harmless. See (13.10) for more details. Recall that we do not want to compute y_i at each step and are faced with the real possibility that y_i might be small since the only bound we have is

$$\|y_i\|^2 \geq \lambda_1[Q_j^* Q_j]. \quad (13.9)$$

We shall return to this matter later. Of course it is still necessary that β_{ji} be small for (θ_i, y_i) to be a good approximation. Except in rare cases no β_j will ever be tiny.

This odd behavior of the algorithm is certainly not what we wanted but neither is it a disaster.

The following example [Scott, 1978] was rigged to have an isolated dominant eigenvalue in order to induce rapid convergence and the consequent arrival of a duplicate “Ritz pair.” The linear independence of the q_i is best measured by $\sigma_1(Q_j)$, the smallest singular value of Q_j , and $\sigma_1(Q_j)^2 \equiv \lambda_1[Q_j^* Q_j]$. See Tables 13.2 and 13.3.

Example 13.3.1. *Quick loss of orthogonality.*

$$A = \text{diag}(0, 1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 4 \times 10^{-4}, 1.0),$$

$$q_1 = (1, 1, 1, 1, 1, 1)^*/\sqrt{6}, \quad \text{roundoff unit} = 10^{-7}.$$

TABLE 13.2
History of the Lanczos run.

j	α_j	β_j	$\sigma_1(Q_j)$
1	0.1668333	0.3726035	1.0
2	0.8333665	0.0003464	0.9999999
3	0.0002004	0.0003094	0.9997097
4	0.1464297	0.3532944	0.0760186
5	0.9998344	0.0001098	0.0000004

TABLE 13.3
Selected Ritz values and residual norms.

j	Ritz value $\theta_i^{(j)}$	$\ Ay_i - y_i\theta_i\ $
3	$\theta_3^{(3)} = 1.000000$	0.48×10^{-7}
5	$\theta_4^{(5)} = 0.9999996$	0.41×10^{-4}
	$\theta_5^{(5)} = 1.0000001$	0.68×10^{-5}

Exercise on Section 13.3

13.3.1. Show that $y_i (\equiv Q_j s_i)$ satisfies (13.9).

13.4. Paige's Theorem

Examination of the elements of $Q_j^* Q_j$ in a variety of cases suggests that orthogonality loss among Q_j 's columns is both widespread and featureless. Such observations promoted the expensive remedy of explicitly orthogonalizing each new q_{j+1} against *all* the previous q 's. However when the “Ritz vectors” $y_i (\equiv Q_j s_i^{(j)})$, $i = 1, \dots, j$, are examined the hidden pattern becomes clearer.

The theorem below, which embodies the insights (13.15) and (13.16), is due to C. C. Paige. A little more preparation is needed before presenting his results.

By the end of its j th step the Lanczos algorithm has produced Q_j —the matrix of Lanczos vectors, T_j —the tridiagonal matrix embodying the three-term recurrence, and the residual vector $r_j \equiv (A Q_j - Q_j T_j) e_j$. Information about the algorithm can be condensed into two fundamental relations which govern the computed quantities. They are

$$A Q_j - Q_j T_j = r_j e_j^* + F_j \quad (13.10)$$

and

$$I_j - Q_j^* Q_j = C_j^* + \Delta_j + C_j \quad (13.11)$$

where j -by- j C_j is strictly upper triangular and Δ_j is diagonal. F , C , and Δ account for the effects of roundoff error, but at this stage there is no need to see precisely what they are like. It turns out that $\|F_j\|$ remains tiny (like ε) relative to $\|A\|$, for all j , but $\|C_j\|$ rises to 1 as soon as a “Ritz pair” is duplicated in the Lanczos process.

The effects of most of the roundoff errors are negligible and the way to keep the analysis clean is to ignore those which are inconsequential. We will give a rigorous analysis of a model of the computation, a model which captures all the important features. The first assumption is that S_j and Θ_j are exact, namely,

$$T_j = S_j \Theta_j S_j^*, \quad S_j^* = S_j^{-1}, \quad \Theta_j = \text{diag}(\theta_1, \dots, \theta_j). \quad (13.12)$$

The second assumption is that *local* orthogonality is maintained, i.e.,

$$q_{i+1}^* q_i = 0, \quad i = 1, \dots, j-1, \quad \text{and } r_j^* q_j = 0, \quad (13.13)$$

or, equivalently, in terms of (13.11), $(C_j)_{i,i+1} = 0$. The justification for (13.13) is that α_i is chosen to force the condition to working accuracy. Were it not for a later application we would also assume that $\Delta_j = 0$ in (13.11), i.e., that $\|q_i\| = 1$ for all i .

It is important to remember that j , the number of Lanczos steps, can increase without bound. Only in exact arithmetic is $j \leq n$. We are now ready to establish Paige's interesting results for this model.

Theorem 13.4.1. *Assume that the simple Lanczos algorithm satisfies relations (13.10) through (13.13) above. Let K_j and N_j , respectively, be the strictly upper-triangular parts of the skew symmetric matrices*

$$F_j^* Q_j = Q_j^* F_j \quad \text{and} \quad \Delta_j T_j = T_j \Delta_j, \quad (13.14)$$

and let $G_j = S_j^*(K_j + N_j)S_j$. Then the "Ritz vectors" y_i ($\equiv Q_j s_i$), $i = 1, \dots, j$ satisfy

$$y_i^* q_{j+1} = \gamma_{ii}^{(j)} / \beta_{ji} \quad (13.15)$$

and for $i \neq k$,

$$(\theta_i - \theta_k) y_i^* y_k = \gamma_{ii}^{(j)} \left(\frac{s_{jk}}{s_{ji}} \right) - \gamma_{kk}^{(j)} \left(\frac{s_{ji}}{s_{jk}} \right) - (\gamma_{ik}^{(j)} - \gamma_{ki}^{(j)}), \quad (13.16)$$

where $G_j \equiv (\gamma_{ik}^{(j)})$ and $\beta_{ji} \equiv \beta_j s_{ji}$.

Proof. Drop the subscript j on which all quantities depend and premultiply (13.10) by Q^* to get

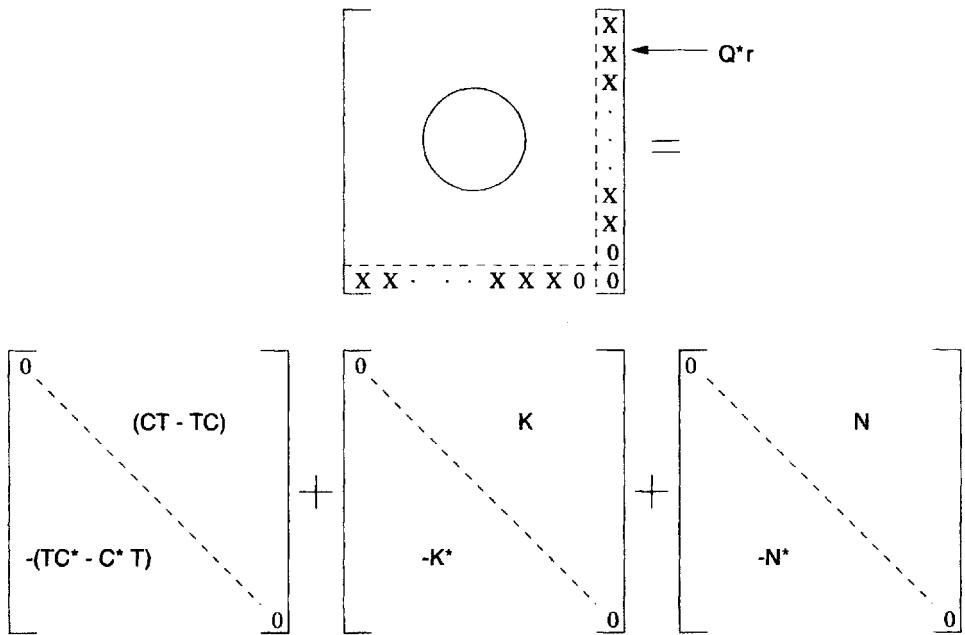
$$Q^* A Q - Q^* Q T = Q^* r e^* + Q^* F. \quad (13.17)$$

To eliminate A subtract (13.17) from its transpose and then apply (13.11) and (13.13) to find

$$\begin{aligned} (Q^* r)^* - e(Q^* r)^* &= (I - Q^* Q)T - T(I - Q^* Q) + F^* Q - Q^* F \\ &= (C^* T - T C^*) + (CT - TC) \\ &\quad + (\Delta T - T \Delta) + F^* Q - Q^* F. \end{aligned} \quad (13.18)$$

This important relation is illustrated in Figure 13.2.

Since $\Delta T - T \Delta$ and $F^* Q - Q^* F$ are skew symmetric they may be represented as in Figure 13.2. By (13.13) $CT - TC$ is strictly upper triangular,

FIG. 13.2. *The structure of Q^*r .*

as is the rank-one matrix $(Q^*r)e^*$. The strictly upper-triangular part of (13.18) is the key relation, i.e.,

$$(Q^*r)e^* = CT - TC + N + K. \quad (13.19)$$

Now $s_i^*(13.19)s_i$ gives

$$\begin{aligned} y_i^* q_{j+1} \beta_{ji} &= (s_i^* Q^*) r (e^* s_i), \quad \text{since } r = q_{j+1} \beta_j, \\ &= s_i^* (CT - TC) s_i + s_i^* (N + K) s_i, \quad \text{by (13.19),} \\ &= s_i^* C s_i \theta_i - \theta_i s_i^* C s_i + \gamma_{ii}, \quad \text{by (13.12),} \\ &= \gamma_{ii}. \end{aligned} \quad (13.20)$$

This gives (13.15).

To obtain (13.16) consider $s_i^*(13.17)s_k$, $i \neq k$,

$$y_i^* A y_k - y_i^* y_k \theta_k = y_i^* q_{j+1} \beta_{jk} + s_i^* Q^* F s_k. \quad (13.21)$$

To eliminate $y_i^* A y_k$ form $s_i^*(13.17)s_k - s_k^*(13.17)s_i$ and then use (13.15) to remove $y_i^* q_{j+1}$; finally (13.16) emerges. \square

These results are of most use when $\|G_i\|$ is tiny, like $\varepsilon\|A\|$, and we know of no case where $\|G_j\| > \varepsilon\|A\|$ and ε is the unit roundoff. For the simple Lanczos algorithm $\|q_i\| = 1$ and so

$$\|Q_j\|^2 = \|Q_j^* Q_j\| \leq \text{trace}(Q_j^* Q_j) = j, \quad (13.22)$$

but this is too crude, especially when $j > n$. A good computable estimate is $\|Q\| = \sqrt{m}$, where m is the size of the biggest cluster of duplicate Ritz vectors (Exercise 13.4.1). In any case, when $\Delta_j = O$ in (13.11), then $N_j = O$ in (13.14) and, by Exercise 13.4.3,

$$\|G_j\| = \|K_j\| \leq \|K_j\|_F = (1/\sqrt{2})\|K_j - K_j^*\|_F \leq \sqrt{2}\|Q_j\|\|F_j\|_F \quad (13.23)$$

and so $\|F_j\|_F$ determines whether or not $\|G_j\|$ is small. Recall that for any B , $\|B\|_F \equiv \sqrt{\text{trace}(B^* B)}$. In [Paige, 1976] a careful error analysis shows that

$$\|F_j\|_F \leq (7 + \alpha)\sqrt{j\varepsilon\|A\|}, \quad (13.24)$$

where α accounts for the error contributed by the user's program for computing Ax by a traditional matrix–vector product. However, we know of no exception to the stronger assertion

$$\|F_j\| \leq \varepsilon\|A\|. \quad (13.25)$$

Another way in which the practical Lanczos algorithm deviates from the exact one is that the “Ritz vectors” y_i do not all remain of unit length as j increases. A complicated analysis of Paige, based on (13.15) and some results from Chapter 7, shows that it is *only* the presence of other “Ritz values” which are extra copies of θ_i that permits $\|y_i\|$ to shrink to values like $\sqrt{\varepsilon}$. Thus a set of duplicate Ritz pairs can be expected to have associated y 's of very different lengths.

Exercises on Section 13.4

- 13.4.1. Assume that, for each j , any two Ritz vectors y_i, y_k are either orthogonal or parallel. Deduce that $\|Q\| \leq \sqrt{m}$, where m is the size of the biggest cluster of “Ritz values.”
- 13.4.2. Derive a three-term recurrence dominating the quantities $\|Q_i^* q_{i+1}\|$; namely, if $\|Q_i^* q_{i+1}\| \leq \xi_{i+1}$ for $i = 1, \dots, j-1$, then $\|Q_j^* q_{j+1}\| \leq \xi_{j+1} \equiv \{\|T_j - \alpha_j\| \xi_j + \beta_{j-1} \xi_{j-1} + (1 + \|Q_j\|) \|F_j\|\}/\beta_j$.
- 13.4.3. Prove that $\|FG\|_F \leq \|F\|\|G\|_F$ whenever FG is defined. Then derive all the relations in (13.23). The monotonicity theorem in Chapter 10 may be useful.

13.5. An Alternative Formula for β_j

There is a variant of the Lanczos algorithm which produces a nonsymmetric tridiagonal matrix J_j instead of T_j . That version has been used extensively but this section shows why it is inferior to the simple implementation given above. The observation is due to Paige. The analysis, which may be found in [Scott, 1978], is a neat application of Paige's theorem and serves to make formulas (13.15) and (13.16) more familiar.

The algorithm. Pick r_0 with $\beta'_0 = \|r\| \neq 0$. For $j = 1, 2, \dots$,

1. $q_j \leftarrow r_{j-1} / \beta'_{j-1}$.
2. $u_j \leftarrow Aq_j$.
3. $\eta_{j-1} \leftarrow q_{j-1}^* u_j \quad (q_0 = 0)$.
4. $\alpha_j \leftarrow q_j^* u_j$.
5. $r_j \leftarrow \mu_j - q_j \alpha_j - q_{j-1} \eta_{j-1}$.
6. $\beta'_j \leftarrow \|r_j\|$.
7. Compute θ_i , y_i , β'_{ji} as desired.
8. If satisfied then stop.

The goal of this version is to obtain extended local orthogonality, $q_{i+1}^* q_{i-1} = O(\epsilon)$, and $q_{i+1}^* q_i = O(\epsilon)$.

A negative value of η_{j-1} should not be permitted, but wildly disparate positive values of η_i and β'_i (they are equal in exact arithmetic) will aggravate the effect of roundoff errors. We suppose here that $\eta_i > 0$ for all i . The i th row of J_j is $(\dots \beta'_{i-1} \ \alpha_i \ \eta_i \dots)$.

Lemma 13.5.1. *There exists $\Omega_j = \text{diag}(\omega_1, \dots, \omega_{j-1}, 1)$ such that*

$$\Omega_j^{-1} J_j \Omega_j = T_j \text{ and } \beta_i = \sqrt{\eta_i \beta'_i}, \quad i = 1, \dots, n-1.$$

The proof constitutes Exercise 13.5.1.

The equation governing the algorithm can now be rewritten

$$\begin{aligned} \mathbf{A}\mathbf{Q}_j\Omega_j &= \mathbf{Q}_j\Omega_j(\Omega_j^{-1}\mathbf{J}_j\Omega_j) + \mathbf{r}_j\mathbf{e}_j^*\Omega_j + \mathbf{F}_j\Omega_j \\ &= \mathbf{Q}_j\Omega_j\mathbf{T}_j + \mathbf{r}_j\mathbf{e}_j^* + \mathbf{F}_j\Omega_j, \end{aligned} \quad (13.26)$$

since $\omega_j = 1$.

We want to apply Paige's Theorem 13.4.1 as before but now the column lengths of $\mathbf{Q}_j\Omega_j$ are not 1 but $\omega_1, \dots, \omega_{j-1}, 1$ instead; i.e., $\Delta_j = \Omega_j^2$ in (13.11). Moreover \mathbf{N}_j , the upper part of $\Delta_j\mathbf{T}_j - \mathbf{T}_j\Delta_j$, is zero except that, for $i = 1, \dots, j-1$,

$$\begin{aligned} n_{i,i+1} &= \beta_i(\omega_{i+1}^2 - \omega_i^2) \\ &= \beta_i\omega_{i+1}^2(1 - \beta_i'/\eta_i), \quad \text{by Exercise 13.5.1,} \\ &= \frac{\beta'_{j-1} \cdots \beta'_{i+1}}{\eta_{j-1} \cdots \eta_{i+1}} \left(1 - \frac{\beta'_i}{\eta_i}\right) \beta_i, \quad \text{eliminating the } \omega\text{'s.} \end{aligned}$$

Thus the greater the asymmetry in \mathbf{J}_j the greater is $(\|\mathbf{K}_j\| + \|\mathbf{N}_j\|)$ as well as the numerators γ_{ij} in (13.15) and (13.16). No such troubles plague the symmetric variant.

Exercise on Section 13.5

13.5.1. Prove Lemma 13.5.1 and find the formulas for the ω_i , $i = 1, \dots, j-1$.

13.6. Convergence \Rightarrow Loss of Orthogonality

Paige's formulas (13.15) and (13.16) substantiate the claims made earlier (section 13.3) about the behavior of the simple Lanczos algorithm in finite precision arithmetic. The appearance of the ratio s_{ji}/s_{jk} , together with its reciprocal, in (13.16), clarified the whole situation. Despite the worst-case bounds (13.23) and (13.24), all the evidence suggests that the elements of the matrix $\mathbf{G}_j \equiv \mathbf{S}_j^*(\mathbf{K}_j + \mathbf{N}_j)\mathbf{S}_j$ which appear in these bounds satisfy

$$|\gamma_{ik}^{(j)}| \leq \varepsilon \|\mathbf{A}\| \quad \text{for all } i, k, j, \quad (13.27)$$

and so $|\mathbf{y}_i^* \mathbf{q}_{j+1}|$ is governed entirely by β_{ji} ($= \beta_j s_{ji}$).

We leave it as an important exercise for the reader to draw the following conclusions:

- Orthogonality among the \mathbf{q}_i , $i = 1, \dots, j$, is well maintained until one of the "Ritz vectors" begins to converge ($\beta_{ji} \doteq \sqrt{\varepsilon \|\mathbf{A}\|}$).

2. Each new Lanczos vector \mathbf{q}_{j+1} and each bad “Ritz vector” ($\beta_{ji} > \|\mathbf{A}\|/j$) has a significant component in the direction of each good “Ritz vector” ($\beta_{ji} < \sqrt{\varepsilon}\|\mathbf{A}\|$).
3. The emergence of (almost) duplicate copies of previously converged “Ritz pairs” is quite consistent with (13.16).

For example, consider the step at which a third “Ritz value” $\bar{\theta}$ first joins the cluster formed by $\dot{\theta}$ and $\ddot{\theta}$ as a clearly recognizable new member. It will be found that $\dot{\theta}$ and $\ddot{\theta}$ are perturbed away from the common eigenvalue λ by the intrusion of $\bar{\theta}$. The same phenomenon occurs to the “Ritz vectors” $\dot{\mathbf{y}}$ and $\ddot{\mathbf{y}}$. On subsequent Lanczos steps $\dot{\theta}$, $\ddot{\theta}$, and $\bar{\theta}$ all converge back on λ until a fourth copy of λ “condenses out,” barges in on the cluster, and elbows the others out of the way. And so it would continue indefinitely.

Paige’s theorem says nothing about the frequency with which the duplicate copies of eigenvalues may appear. The cycle time for each individual eigenvector seems to be fairly constant and to depend strongly on the gaps in the spectrum relative to the spread. The subject appears not to have been investigated.

The tables below illustrate (13.15) and (13.16) in the context of Example 13.3.1 considered above.

Example 13.6.1.

$$\mathbf{A} = \text{diag}(0, 1 \times 10^{-3}, 2 \times 10^{-3}, 3 \times 10^{-3}, 4 \times 10^{-3}, 1.0),$$

$$\mathbf{q}_1 = (1, 1, 1, 1, 1, 1)/\sqrt{6}.$$

At $j = 3$ $\beta_3 = 0.0003094$, and

i	θ_i	$\ \mathbf{y}_i\ $	β_{ji}	$ \mathbf{y}_i^* \mathbf{q}_4 $	
1	0.587×10^{-4}	0.9999745	0.2225×10^{-3}	0.4297×10^{-3}	
2	0.3415×10^{-3}	1.000025	0.2228×10^{-3}	0.4297×10^{-3}	
3	1.000000	1.000001	0.48×10^{-7}	0.923981	↔

In each row the product of the entries in the last two columns is approximately $10^{-7} = \varepsilon\|\mathbf{A}\|$ as required by (13.15). In addition mutual orthogonality between good and bad Ritz vectors is poor as required by (13.16).

$$\mathbf{Y}_3^* \mathbf{Y}_3$$

1.0	0.5×10^{-6}	-0.2×10^{-3}	\Leftarrow
	1.0	0.2×10^{-3}	\Leftarrow
Symmetric		1.0	

At $j = 5$ $\beta_5 = 0.0001098$ and

$$\mathbf{Y}_5^* \mathbf{Y}_5$$

1.0	-0.89×10^{-3}	0.10×10^{-5}	-0.40×10^{-6}	-0.35×10^{-6}	\Leftarrow
	1.0	0.85×10^{-3}	0.54×10^{-6}	0.60×10^{-6}	
		1.0	-0.42×10^{-6}	-0.35×10^{-6}	
			0.83	0.52×10^{-1}	
Symmetric				1.3	

As a result of his analysis and experience Paige suggested that the simple Lanczos process be continued as long as storage permitted. At the end of the run all the relevant Ritz pairs are computed and the duplicate copies are simply discarded. This scheme has been used with success, but in the remaining sections we consider a modification of the Paige-style Lanczos algorithm which is more efficient and, more significantly, can be used automatically without expert judgment.

i	θ_i	$\ \mathbf{y}_i\ $	β_{ji}	$ \mathbf{y}_i^* \mathbf{q}_6 $
1	0.157×10^{-4}	1.000490	0.486×10^{-4}	0.33997×10^{-2}
2	0.2001×10^{-3}	1.000477	0.778×10^{-4}	0.21557×10^{-2}
3	0.3845×10^{-3}	0.999509	0.486×10^{-4}	0.35172×10^{-2}
4	0.9999996	0.75043	0.414×10^{-4}	0.70092×10^{-2}
5	1.000000	1.148672	0.681×10^{-5}	0.70535×10^{-2}
		↑	↑	

13.7. Maintaining Orthogonality

As we have seen, the Lanczos vectors $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots$ lose mutual orthogonality as the number of steps increases. The original cure for this condition was proposed by Lanczos himself and required the explicit orthogonalization of \mathbf{q}_{j+1} against all previous \mathbf{q} 's. The following step is added after step 5 in the algorithm in section 13.1.

$$5\frac{1}{2}. \quad \mathbf{r}_j \leftarrow \mathbf{r}_j - \mathbf{q}_\nu (\mathbf{q}_\nu^* \mathbf{r}_j), \quad \nu = j, j-1, \dots, 2, 1.$$

(Note that \mathbf{r}_j is explicitly orthogonalized against \mathbf{q}_j and \mathbf{q}_{j-1} .) Consequently all the \mathbf{q} 's must be kept handy and the arithmetic cost of each step soars (Exercise 13.7.1). This variant of the algorithm is called Lanczos *with full reorthogonalization*.

It is of some (mainly academic) interest to note that reorthogonalization of itself cannot guarantee to produce \mathbf{q} 's which are orthogonal to working accuracy. The reason is given in section 6.9 and, as suggested there, the remedy is to test the decrease in norm of each vector and repeat an orthogonalization whenever necessary.

Another fact worth noting is that the cost of reorthogonalization can be halved by keeping the matrix \mathbf{Q}_j in factored form $\mathbf{H}_1 \mathbf{H}_2 \cdots \mathbf{H}_j \mathbf{E}_j$ where $\mathbf{E}_j = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_j)$ and each \mathbf{H}_i is a reflector matrix $\mathbf{H}(\mathbf{w}_i)$ as described in section 6.2. It is only necessary to keep the vectors \mathbf{w}_i , $i = 1, \dots, j$, and the first $i-1$ elements of \mathbf{w}_i are zero. See [Golub, Underwood, and Wilkinson, 1972] for more details. Despite this improvement there is strong incentive to avoid the storage and arithmetic costs of reorthogonalization.

Paige's result (Theorem 13.4.1) shows that in the simple Lanczos method \mathbf{q}_{j+1} tilts most in the direction of those "Ritz vectors" $\mathbf{y}_i^{(j)}$, if any, which are fairly good approximations to eigenvectors; more precisely,

$$\mathbf{y}_i^{(j)*} \mathbf{q}_{j+1} = \gamma_{ii}^{(j)} / \beta_{ji}, \quad (13.28)$$

where $\gamma_{ik}^{(j)} \leq \varepsilon \|\mathbf{A}\|$ for all i, j, k , by (13.27), and $\beta_{ji} \equiv \beta_j s_{ji} \doteq \|\mathbf{A}\mathbf{y}_i^{(j)} - \mathbf{y}_i^{(j)}\theta_i^{(j)}\|$ by (13.3). Moreover the tilting can be monitored *without computing* $\mathbf{y}_i^{(j)}$ because each β_{ji} can be computed with approximately $4j$ ops once the "Ritz values" $\theta_i^{(j)}$ are known.

Formula (13.28) suggests a more discriminating way of maintaining a good measure of orthogonality than full reorthogonalization; namely, orthogonalize \mathbf{q}_{j+1} against $\mathbf{y}_i^{(j)}$ when and only when β_{ji} becomes small. To do this $\mathbf{y}_i^{(j)}$ must

be computed and stored and this itself costs half as much as full reorthogonalization at one step. When convergence is slow the new orthogonalizations do not occur very often and the ever growing Q_j does not have to be kept on hand for each step.

To see whether there is merit in the idea we examine two simple examples in each of which a vector \bar{q} is orthogonalized against y_2 to produce

$$q = (I - y_2 y_2^*) \bar{q} / \| (I - y_2 y_2^*) \bar{q} \|.$$

The global orthogonality of the q_i and the $y_i^{(j)}$, $i = 1, \dots, j$ is measured by

$$\kappa_j \equiv \| I - Q_j^* Q_j \| = \| I - Y_j^* Y_j \|.$$
 (13.29)

In the examples $\bar{\kappa}_3$ is the value before orthogonalization and κ_3 is the value afterward.

Example 13.7.1. *Nothing gained.*

$$y_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad y_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \frac{1}{\sqrt{2}}, \quad \bar{q} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{\sqrt{3}}, \quad q = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \frac{1}{\sqrt{6}}.$$

Then

$$\begin{aligned} y_1^* q &= -1/\sqrt{6}, & y_1^* y_2 &= 1/\sqrt{2}, & y_2^* q &= 0, \\ \kappa_2 &= 1/\sqrt{2}, & \bar{\kappa}_3 &= 1/\sqrt{2}, & \kappa_3 &\geq 1/\sqrt{2}. \end{aligned}$$

Example 13.7.2. *Previous level of orthogonality restored.*

$$y_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad y_2 = \begin{bmatrix} 10^{-4} \\ 1 \\ 0 \end{bmatrix}, \quad \bar{q} = \begin{bmatrix} 0 \\ 10^{-2} \\ 1 \end{bmatrix}, \quad q = \begin{bmatrix} -10^{-6} \\ 0 \\ 1 \end{bmatrix}.$$

Then

$$\begin{aligned} y_1^* q &= -10^{-6}, & y_1^* y_2 &= 10^{-4}, & y_2^* q &= 10^{-10}, \\ \kappa_2 &= 10^{-4}, & \bar{\kappa}_3 &= 10^{-2}, & \kappa_3 &= 10^{-4}. \end{aligned}$$

Example 13.7.1 shows that orthogonalization, *by itself*, need not improve the situation. Example 13.7.2 is more realistic and suggests that the technique is beneficial whenever $|y_i^* q_{j+1}|$ exceeds κ_j significantly.

Rewards for maintaining strong linear independence among the Lanczos vectors, i.e., $\| I - Q_j^ Q_j \| < 0.01$.*

1. Troublesome, redundant Ritz pairs cannot be formed.
2. The number of Lanczos steps is kept to a minimum and, in any case, can never exceed n .
3. Multiple eigenvalues can be found, one by one. The reason is given in the next section.
4. The method can be used as a black box and requires no delicate parameters to be set by the user beyond indicating the desired accuracy and the amount of fast storage available.

On the debit side the program must compute and store any good Ritz vectors to be used for purging the new \mathbf{q} 's whether or not they are of interest to the user. For instance, if three (algebraically) small eigenvalues are wanted the algorithm may well be obliged to compute three or more Ritz vectors belonging to large Ritz values simply because some of them converge quickly. Scott is exploring the consequences of orthogonalizing the new \mathbf{q} 's against only those Ritz vectors which are wanted.

In what follows we shall sometimes say \mathbf{q} is *purged of \mathbf{y}* as a synonym for \mathbf{q} is orthogonalized against \mathbf{y} .

Suppose that we permit κ_j (defined in (13.29)) to grow slowly, as j increases, from an initial value $\kappa_1 \doteq \varepsilon$ but never to exceed a certain value κ . If $\kappa \doteq n\varepsilon$ then orthogonalization will be forced almost all the time and the cost may exceed that of full reorthogonalization. At the other extreme if $\kappa \geq 1$ then the Paige-style Lanczos process is recovered. Thus κ lets us interpolate between the two extreme versions of the Lanczos process.

Exercise on Section 13.7

- 13.7.1. Apart from the cost of computing $\mathbf{A}\mathbf{q}_j$ the op count for simple Lanczos is $5n$ and for full reorthogonalization is $(j + 3)n$ if it is assumed that \mathbf{Q}_j is not held in factored form as described in [Golub, Underwood, Wilkinson, 1972]. Verify these counts.

***13.8. Selective Orthogonalization (SO)**

A small modification to the simple Lanczos algorithm ensures that the vectors $\mathbf{q}_1, \mathbf{q}_2, \dots$ maintain a reasonable, preset level of linear independence. Let $\kappa_j \equiv \| -\mathbf{Q}_j^* \mathbf{Q}_j \|$ and suppose that it is required to keep $\kappa_j \leq \kappa$ for some κ in the interval $(n\varepsilon, 0.01)$. If $\varepsilon = 10^{-14}$ and $n = 10^4$ then the interval still spans eight orders of magnitude. Both experience and informal analysis suggest that κ

should be chosen near $\sqrt{\varepsilon}$ (to within an order of magnitude) in order to reap the fruits of full reorthogonalization at a cost close to, and sometimes less than, that of the simple algorithm.

The new version modifies the vector r_j of the simple algorithm of section 13.1 before normalizing it, and some notation is needed to distinguish r_j before from r_j afterward. We reserve r_j for the final form and use r'_j for the original. From (13.10)

$$r'_j \equiv Aq_j - q_j\alpha_j - q_{j-1}\beta_{j-1} - f_j,$$

where f_j accounts for roundoff errors during the j th step and $\|f_i\|$ remains below $n\varepsilon\|A\|$ for all j . We are interested in $\angle(r'_j, q_i)$, $i = 1, \dots, j$, or, equivalently, in $\angle(r'_j, y_i^{(j)})$, $i = 1, \dots, j$. Recall that $y_i^{(j)} = Q_j s_i$ and (θ_i, s_i) is an eigenpair of T_j . While the angles remain close to $\pi/2$ there is no need to depart from the simple algorithm and the new algorithm sets $r_j \equiv r'_j$. By Paige's Theorem 13.4.1

$$\cos \angle(r'_j, y_i^{(j)}) = \gamma_{ii}^{(j)} / \beta'_{ji} \|y_i^{(j)}\|,$$

and $|\gamma_{ii}^{(j)}| \leq \varepsilon\|A\|$ for all i, j , and $\beta'_{ji} \equiv \beta'_j s_{ji}$, $\beta'_j = \|r'_j\|$, $s_{ji} = e_j^* s_i$. By the final remarks in section 13.4 the restriction $\kappa \leq 0.01$ keeps $|1 - \|y_i^{(j)}\|| \ll \kappa$ for all i, j , and we shall ignore the factor $\|y_i^{(j)}\|$ throughout this section. The important consequence is that the angles can be monitored by keeping track of the easily computed quantities β'_{ji} .

As soon as the Lanczos vectors begin to lose orthogonality attention centers on the set of indices

$$\mathcal{L}(j) = \{i : |\cos \angle(y_i^{(j)}, r'_j)| \geq \kappa/\sqrt{j}\}.$$

Let $|\mathcal{L}(j)|$ denote the number of indices in $\mathcal{L}(j)$. For most values of j it turns out that $|\mathcal{L}| = 0$, but when $|\mathcal{L}(j)| > 0$ then the idea is to purge r'_i of the associated $y_i^{(j)}$, $i \in \mathcal{L}(j)$, which we call the *threshold* "Ritz vectors." The hope is that the resulting vector r_i will satisfy $|\cos \angle(y_k^{(j)}, r_j)| < \kappa/\sqrt{j}$ for all the values $k = 1, \dots, j$. Then r_j is normalized to become q_{j+1} and we have

$$\|Q_j^* q_{j+1}\| = \|Y_j^* q_{j+1}\| < \sqrt{j} \frac{\kappa}{\sqrt{j}} = \kappa.$$

The factor \sqrt{j} is a crude overbound and could not be attained. A more realistic bound is given in Exercise 13.8.1.

In order to purge r'_j it is necessary to compute $y_i^{(j)}$ for $i \in \mathcal{L}(j)$. This involves bringing the old q 's back from secondary storage, and we say that the

modified algorithm *pauses* whenever $|\mathcal{L}(j)| > 0$. On the other hand if $\kappa \geq \sqrt{\varepsilon}$ then $\beta'_{ji} \leq \varepsilon \|A\| / (\kappa / \sqrt{j}) = \sqrt{j\varepsilon} \|A\|$ and it follows that $\theta_i^{(j)}$, $i \in \mathcal{L}(j)$, will often agree with an eigenvalue of A to working accuracy. For some, but not all applications, these threshold “Ritz vectors” will already be acceptable and so there is no reluctance to compute them.

It is worth repeating that when κ exceeds $n\varepsilon$ then the $y_i^{(j)}$ are neither orthonormal (to working accuracy) nor the true Ritz vectors from $\text{span } Q_j$. That is why we persist in using quotes when referring to them. Before proceeding with the description of the algorithm we must make an observation about (true) Ritz vectors.

There is an annoying identification problem as j varies because the set of Ritz vectors changes completely at each step. In general there is no natural association between $y_i^{(j)}$ and $y_i^{(j+1)}$ for a given i . This is evident in Example 10.1.1 with $j = 1$ and $j = 2$. However as soon as a sequence $\{\theta_i^{(j)}\}$, for some fixed i such as ± 1 or ± 2 , settles down in its first few digits, i.e., as soon as convergence becomes apparent, then it is meaningful to speak about y_i and its (vector) values at various steps.

With these remarks in mind we follow the history of a typical Ritz vector in the modified algorithm. At step 25, say, $\theta_i^{(25)}$ emerges from a crowd of undistinguished “Ritz values” in the middle of the spectrum and settles down, $|\theta_i^{(24)} - \theta_i^{(25)}| / |\theta_i^{(25)}| < 0.1$. Later $y_i^{(40)}$ becomes a threshold vector; there is a pause wherein $y_i^{(40)}$ is computed and possibly some other threshold vectors as well. Next it always happens that $y_i^{(41)}$ is a threshold vector and agrees with $y_i^{(40)}$ to several figures (depending on κ). It is tempting to forego the expense of pausing to compute $y_i^{(41)}$ and to use $y_i^{(40)}$ in its place for purging r'_{41} . We will return to this point shortly.

The crucial fact, which is not immediately obvious, is that for $j = 42, 43, \dots$, $i \notin \mathcal{L}(j)$. The effects of roundoff may or may not put i into \mathcal{L} again before the computation is over. In any case what matters is that $i \notin \mathcal{L}(j)$ for most values of j . That is part of the reason that the modified algorithm is as economical as the simple one.

Let us return to $y_i^{(40)}$ and $y_i^{(41)}$. True Ritz vectors are orthonormal. Moreover $y_i^{(40)}$ will deviate most from a true Ritz vector because of components of those $y_\nu^{(40)}$ which crossed the threshold earlier and are therefore better converged; we call these the *good* “Ritz vectors” at step 40. Since the Lanczos vectors q_1, q_2, \dots, q_j must be brought back from secondary storage there is little extra trouble in recomputing *all* the good $y_\nu^{(40)}$ (more precisely those which

have not yet converged to working accuracy). It is also possible to orthonormalize the good $y_\nu^{(40)}$ and produce new vectors which we call y_1, \dots, y_i without a superscript. In that case y_i is used for purging both r'_{40} and r'_{41} . There will not be a pause at $j = 41$ unless another “Ritz vector” crosses the threshold at that step. The only pairs (θ_μ, s_μ) which need be computed at each step are the extreme (outermost) θ ’s that are not yet good. There is no need to recompute good θ ’s at each step, even if they have not converged.

Note that the modification is independent of the user’s accuracy requirements. Results to working accuracy can be obtained but so can rough approximations. There is no anomaly if a Ritz vector is acceptable before it is good because the adjectives good and bad pertain to the degree of orthogonality which is to be maintained. There are strong arguments for taking $\kappa = \sqrt{\varepsilon}$ (see the next section) and so no extra choice is thrust on the user.

Suppose that A has a multiple eigenvalue λ_1 . At step 30, say, $y_1^{(30)}$ will be one of λ_1 ’s eigenvectors to within working accuracy. After step 30 the SO keeps $y_1^* r_j = O(\varepsilon \|A\|)$ for $j > 30$. Roundoff error will introduce nonzero components of other eigenvectors for λ_1 which are orthogonal to y_1 . In time a new Ritz vector, say, $y_8^{(70)}$, will converge to one of these eigenvectors. Then $y_1^* r_j = O(\varepsilon \|A\|)$, $y_8^* r_j = O(\varepsilon \|A\|)$, for $j > 70$ and so on until λ_1 ’s eigenspace is spanned.

Section 13.8.1 describes the SO algorithm in more detail.

13.8.1. LanSO Flowchart (Lanczos Algorithm with SO)

Recall that a good Ritz vector is one that has not yet been accepted as an eigenvector but its Ritz value has converged to an eigenvalue to working precision.

Parameters

lc = index of last fully converged Ritz vector ($\beta_{ji} < j\varepsilon \|A\|$) from the left end of the spectrum.

lg = index of last good Ritz vector [$lg \in \mathcal{L}(k)$ for some $k \leq j$] from the left.

rc, rg as above for the right end of the spectrum.

Initialize

$lc = lg = rc = rg = |\mathcal{L}| = 0$. $\mathcal{L} = \emptyset$ (empty). $q_0 = o$. Pick $r'_0 \neq o$.

Loop

For $j = 1, 2, \dots, n$ repeat steps 1 through 5.

1. If $|\mathcal{L}| > 0$ then purge \mathbf{r}' of threshold vectors to get \mathbf{r} and set $\beta_{j-1} \leftarrow \|\mathbf{r}\|$.
2. If $\beta_{j-1} = 0$ then stop else normalize \mathbf{r} to get \mathbf{q}_j .
3. Take a Lanczos step to get α_j , \mathbf{r}' , β'_j .
4. $\theta_i^{(j)} \leftarrow \lambda_i[\mathbf{T}_j]$ for $i = lg + 1, lg + 2$ and $i = -(rg + 1), -(rg + 2)$. Compute associated s_{ji} . Set $|\mathcal{L}| = 0$.
5. If $\beta'_{ji} (= \beta'_j s_{ji}) < \sqrt{\varepsilon} \|\mathbf{T}_j\|$ for any of the i in step 4 then pause.

Pause

1. Form $\mathcal{L} (\equiv \{i : \beta'_{ij} < \sqrt{\varepsilon} \|\mathbf{T}_j\|\})$. Update lg, rg .
2. Summon \mathbf{Q}_j and compute $\mathbf{y}_l^{(j)} = \mathbf{Q}_j \mathbf{s}_l$ for $l = lc, \dots, lg$ and $l = -rc, \dots, -rg$.
3. *Optional step:* Perform modified Gram–Schmidt on the new $\mathbf{y}_l^{(j)}$; use the most accurate first. Update \mathbf{s}_{jl} accordingly.
4. If enough \mathbf{y} 's are acceptable then stop.
5. Compute $\mathbf{y}_l^* \mathbf{r}'$ for each good \mathbf{y}_l ; if too big add l to \mathcal{L} .
This step allows \mathbf{y}_l to be refined.

It is only necessary to retain the threshold vectors after the pause; the Lanczos vectors can be rewound.

Example 13.8.1. *Example of SO.*

$$n = 6.$$

$$\mathbf{A} = \text{diag}(0., 0.00025, 0.0005, 0.00075, 0.001, 10.).$$

$$\mathbf{q}_1 = 6^{-1/2}(1., 1., 1., 1., 1., 1.)^*$$

Unit roundoff $\doteq 10^{-14}$. Note that 0.75×10^{-6} is written $.75E - 06$.

Simple Lanczos was run for six steps.

$$\mathbf{Q}_6^* \mathbf{Q}_6$$

$$\begin{bmatrix} .10E + 01 & .75E - 14 & -.30E - 10 & .25E - 06 & .97E - 02 & .41E + 00 \\ .75E - 14 & .10E + 01 & .33E - 10 & .55E - 06 & .22E - 01 & .91E + 00 \\ -.30E - 10 & .33E - 10 & .10E + 01 & -.97E - 10 & .19E - 05 & .79E - 04 \\ .25E - 06 & .55E - 06 & -.97E - 10 & .10E + 01 & .11E - 09 & .23E - 08 \\ .97E - 02 & .22E - 01 & .19E - 05 & .11E - 09 & .10E + 01 & -.12E - 12 \\ .41E + 00 & .91E + 00 & .79E - 04 & .23E - 08 & -.12E - 12 & .10E + 01 \end{bmatrix}$$

The Lanczos algorithm with SO was run for six steps. It paused after four steps and computed a good Ritz vector for the eigenvalue 10. It then took two more steps orthogonalizing against this vector.

$Q_6^* Q_6$ for SO

$$\begin{bmatrix} .10E + 01 & .75E - 14 & -.30E - 10 & .25E - 06 & .11E - 09 & .92E - 10 \\ .75E - 14 & .10E + 01 & .33E - 10 & .55E - 06 & .51E - 10 & -.36E - 10 \\ -.30E - 10 & .33E - 10 & .10E + 01 & -.97E - 10 & .44E - 10 & -.37E - 07 \\ .25E - 06 & .55E - 06 & -.97E - 10 & .10E + 01 & .24E - 07 & -.64E - 08 \\ -.11E - 09 & .51E - 10 & -.44E - 10 & .24E - 07 & .10E + 01 & .10 - 13 \\ .92E - 10 & -.36E - 10 & -.37E - 07 & -.64E - 08 & .10E - 13 & .10E + 01 \end{bmatrix}$$

Note that the leading 4-by-4 principal minor is the same in both matrices. Note that the robust linear independence has been maintained by the SO scheme.

Exercises on Section 13.8

- 13.8.1. Assume that $y_k^{(j)*} r_j = y_k^{(j)} r'_j$ for $k \notin \mathcal{L}(j)$. Use Paige's Theorem 13.4.1 to obtain

$$\|Q_j^* q_{j+1}\| \leq (\varepsilon \|A\| / \beta_j) \left(\sum_{k \notin \mathcal{L}(j)} s_{jk}^{-2} \right)^{-1/2}.$$

Assume that $\|y_k^{(j)}\| = 1$ and $|\gamma_{ii}^{(j)}| \leq \varepsilon \|A\|$.

- 13.8.2. Assume that $q_{j+1} \beta_j = A q_j - q_j \alpha_j - q_{j-1} \beta_{j-1}$ for all j . Let $Az = z\lambda$ and suppose that $z^* q_{40} = z^* q_{41} = 0$. Show that, in exact arithmetic, $z^* q_j = 0$ for $j > 41$.

*13.9. Analysis of SO

This section discusses various aspects of the SO procedure described in section 13.8.

The simple Lanczos algorithm will continue indefinitely in a finite precision environment and Paige was concerned with the question of convergence, as $j \rightarrow \infty$, of certain Ritz pairs $(\theta_i^{(j)}, y_i^{(j)})$ to eigenpairs of A . The modified algorithm preserves a strong measure of linear independence among the q_i and so the procedure must terminate ($\beta_j \leq n\varepsilon\|A\|$) with $j \leq n$. Consequently our attention turns away from convergence to the influence of the value of κ on the execution of the modified algorithm.

13.9.1. Orthonormalizing the Good Ritz Vectors

This extra feature is not a necessary part of selective orthogonalization but it has merit when there is no subroutine available to refine crude output from a Lanczos process and supply careful a posteriori error bounds on the lines of Chapter 10.

For simplicity relabel the Ritz vectors so that $y_1^{(j)}, \dots, y_{i-1}^{(j)}$ are the good ones and suppose that $\{i\} = \mathcal{L}(j)$. Without loss of generality we suppose that $\beta_{j1} \leq \beta_{j2} \leq \dots \leq \beta_{ji}$. Since \mathbf{Q}_j must be brought back to compute $y_i^{(j)}$ there may be little extra expense to recompute the good Ritz vectors. Some might be accepted as eigenvectors and frozen.

During the pause the algorithm computes, for $l = 1, 2, \dots, i$,

$$\begin{aligned} y_l^{(j)} &\leftarrow \mathbf{Q}_j s_l, \\ \tilde{y}_l &\leftarrow y_l^{(j)} - \sum_{\nu=1}^{l-1} y_\nu^* (y_\nu^* y_l^{(j)}), \\ y_l &\leftarrow \tilde{y}_l / \|\tilde{y}_l\|. \end{aligned}$$

What Ritz value should be associated with y_l ? Naturally the Rayleigh quotient $\rho(y_l)$ would be best but we do not want to compute $\mathbf{A}y_l$. The answer is suggested by the following result.

Lemma 13.9.1.

$$\rho(y_l) = \theta_l^{(j)} + O(\kappa^2 \|\mathbf{A}\|), \quad l \leq i < j. \quad (13.30)$$

The proof is left as Exercise 13.9.2. Note that the best bound we can put on the coefficients $\gamma_{\nu l} \equiv y_\nu^* y_l^{(j)}$ is $|\gamma_{\nu l}| \leq \kappa_j + O(\kappa_j^2)$ (Exercise 13.9.1). If $\kappa^2 \leq \varepsilon$ there is no loss in using $\theta_l^{(j)}$ as the Ritz value for y_l .

In the remaining subsection we will not consider the use of the vectors y_l , $l \leq i$ but will continue to work with the unimproved $y_l^{(j)}$ for purging r'_j .

13.9.2. The Effect of Purging on Angles

We drop the superscript j on each “Ritz vector” y_k . We need to compare $\angle(y_k, r_j)$ and $\angle(y_k, r'_j)$ for all $k \notin \mathcal{L}(j)$. Of course $y_i^* r_j = O(\varepsilon \|\mathbf{A}\|)$ for $i \in \mathcal{L}(j)$,

by construction. After the purge

$$r_j \equiv r'_j - \sum_{\nu \in \mathcal{L}(j)} y_\nu \xi_\nu, \quad \xi_\nu = y_\nu^* r'_j,$$

and so

$$y_k^* r_j = y_k^* r'_j - \sum_{\nu} (y_k^* y_\nu) \xi_\nu. \quad (13.31)$$

In exact arithmetic both ξ_ν and $y_k^* y_\nu$ vanish, but in practice it is only necessary that their product be tiny, like $\varepsilon \|A\|$, to ensure that the purgings do not degrade the small inner products $y_k^* r'_j$.

Now ξ_ν is definitely not small since

$$\begin{aligned} |\xi_\nu| &= |y_\nu^* r'_j| / \|y_\nu\| \\ &= |\cos \angle(y_\nu, r'_j)| \beta'_j \quad (\beta'_j \equiv \|r'_j\|), \\ &\geq \beta'_j \kappa / \sqrt{j} \quad \text{by definition of } \mathcal{L}(j). \end{aligned}$$

The other factor, $y_k^* y_\nu$, in (13.31) is bounded by κ_j and this bound is realistic for some values of k . Some terms in the sum in (13.31) will be larger than $\beta'_j \kappa \kappa_j / j$. Moreover $\kappa_j \doteq \kappa$ eventually. So, in order to preserve $\angle(y_k, r'_j)$ to working accuracy it seems necessary to have

$$\kappa^2 \leq \varepsilon. \quad (13.32)$$

On the other hand if $|\xi_\nu|$ can be much greater than $\kappa \|A\|$ then (13.32) will not protect the configuration from a gradual acceleration in the loss of orthogonality. If Paige's Theorem 13.4.1 continues to hold then the *sudden* convergence of one of the y_i , indicated by $\beta_{ji} \ll \kappa \|A\|$, might provoke such a large ξ_ν . The following result [Scott, 1978] shows that this fear is unfounded.

Lemma 13.9.2. *If $r'_j = Aq_j - q_j \alpha_j - q_{j-1} \beta_{j-1} - f_j$ then*

$$|y_i^* r'_j| \leq \|Ay_i - y_i \theta_i\| + \kappa_j [(\alpha_j - \theta_i)^2 + \beta_{j-1}^2]^{1/2} + \|f_j\|. \quad (13.33)$$

The proof is left as Exercise 13.9.3.

The effect of SO is to spoil the nice bound $\|\mathbf{A}y_k - y_k\theta_k\| \leq \beta_{jk} + O(\varepsilon\|\mathbf{A}\|)$ for the bad “Ritz vectors” y_k . The next subsection suggests that for the good y_i ($i \in \mathcal{L}(\nu)$ for some $\nu \leq j$) the convenient bound still holds. Moreover, by the definition of $\mathcal{L}(j)$, $\beta_{ji} \leq \varepsilon\sqrt{j}\|\mathbf{A}\|/\kappa$. Thus Lemma 13.9.2 yields the following interesting bound:

$$|y_i^* r'_j| \leq \varepsilon\sqrt{j}\|\mathbf{A}\|/\kappa + \kappa(2\|\mathbf{A}\|) + O(\varepsilon\|\mathbf{A}\|), \quad i \in \mathcal{L}(j).$$

The right-hand side is minimized (approximately) by the choice $\kappa = \sqrt{\varepsilon}$ and then $|\xi_i|$ can never rise much above $\sqrt{\varepsilon}\|\mathbf{A}\|$.

13.9.3. The Governing Formula

Suppose that, for the first time, a “Ritz vector” y_1 crosses the threshold at step j , i.e., $|\cos \angle(y_1, r'_j)| > \kappa/\sqrt{j}$. After the purge

$$r_j = r'_j - y_1\xi_1, \quad \xi_1 = y_1^* r'_j / \|y_1\|. \quad (13.34)$$

The basic relation given in (13.10) is

$$\mathbf{A}Q_j - Q_j T_j = r'_j e_j^* + F_j, \quad (13.35)$$

and when r'_j is eliminated from (13.34) and (13.35) we find

$$\mathbf{A}Q_j - Q_j T_j - \xi_1 y_1 e_j^* = r_j e_j^* + F_j. \quad (13.36)$$

It is helpful to bring $y_1 e_j^*$ to the left side because y_1 is in $\text{span } Q_j$ and $y_1^* r_j = O(\varepsilon\|\mathbf{A}\|)$ by construction. Recall that $y_1 = Q_j s_1$ and $T_j s_1 = s_1 \theta_1$; so (13.36) becomes

$$\mathbf{A}Q_j - Q_j(T_j + \xi_1 s_1 e_j^*) = r_j e_j^* + F_j. \quad (13.37)$$

The rank-one perturbation of T_j compensates for the fact that T_j is not the projection of \mathbf{A} on $\text{span } Q_j$.

It may be verified (Exercise 13.9.4) that the eigenpairs of the perturbed matrix are

$$\begin{cases} \theta_1 + \xi_1 s_{j1}, & s_1, \\ \theta_k, & s_k + s_1[\xi_1 s_{jk}/(\theta_k - \theta_1)] + O(\varepsilon), \quad k > 1. \end{cases} \quad (13.38)$$

By Paige’s theorem $y_1^* r'_j = \gamma_{11}/\beta'_{j1}$ where $\gamma_{11} = O(\varepsilon\|\mathbf{A}\|)$ and $\beta'_{j1} = \beta'_j s_{j1}$. So the quantity ξ_1 in (13.34) may be written

$$\xi_1 = y_1^* r'_j / \|y_1\| = \gamma_{11}/(\beta'_j \|y_1\|)$$

and the perturbation $\xi_1 s_{j1}$ to θ_1 in (13.38) satisfies $\xi_1 s_{j1} = \gamma_{11}/(\beta'_j \|y_1\|)$ and thus is negligible unless β'_j itself is small. However, a small β'_j is not consistent with our assumption that only one Ritz vector crossed the threshold at step j .

On the other hand the significant changes to the other eigenvectors s_k come as no surprise. The formulas indicate precisely how to remove the component of y_1 that has crept into the other "Ritz vectors" y_k . Since we are not interested in computing the true bad Ritz vectors from $\text{span } Q_j$ there seems to be no point in recording the modifications to the s_k . This is a relief.

To see the general pattern more clearly we go on to the next step and assume that no new threshold vectors appear. The algorithm takes the lazy way out and uses $y_1^{(j)}$ instead of $y_1^{(j+1)}$ to purge r'_{j+1} . A good estimate for $\sin \angle(y_1^{(j)}, y_1^{(j+1)})$ is $s_{j+1,1}$, the $(j+1, 1)$ element of S_{j+1} , and we expect that $s_{j+1,1}^{(j+1)} < s_{j1}^{(j)} = O(\sqrt{\varepsilon})$. In any case whether or not $y_1^{(j)}$ and $y_1^{(j+1)}$ are close, we find

$$\begin{aligned} \mathbf{A}Q_{j+1} - Q_{j+1} \left[T_{j+1} + \begin{bmatrix} s_1^{(j)} \\ \mathbf{o} \end{bmatrix} (0, \dots, 0, \xi_1^{(j)}, \xi_1^{(j+1)}) \right] \\ = r_{j+1} e_{j+1}^* + F_{j+1}. \end{aligned}$$

The vector $s_1^{(j)}$ must be endowed with an extra zero element at the bottom to be conformable with T_{j+1} . The picture shows the nonnegligible entries.

$$\begin{bmatrix} x & x & & \square & \triangle \\ x & x & x & \square & \triangle \\ & x & x & \square & \triangle \\ & & x & \square & \triangle \\ & & & x & x \\ & & & & x & x \end{bmatrix}$$

$\square = \text{elements of } s_1 \xi_1^{(j)}$,
 $\triangle = \text{elements of } s_1 \xi_1^{(j+1)}$.

The bottom two elements of the perturbing vectors are $O(\varepsilon \|T_{j+1}\|)$.

It might be supposed that a similar pattern occurs at step $j+2$ but that is not the case. After two purgings r'_{j+2} will be orthogonal to y_1 to working

accuracy (thanks to the three-term recurrence):

$$\begin{aligned} \mathbf{y}_1^* \mathbf{r}'_{j+2} &= \mathbf{y}_1^* (\mathbf{A} \mathbf{q}_{j+2} - \mathbf{q}_{j+2} \alpha_{j+2} - \mathbf{q}_{j+1} \beta_{j+1} - \mathbf{f}_{j+2}) \\ &= (\mathbf{A} \mathbf{y}_1)^* \mathbf{q}_{j+2} - O(\varepsilon) - O(\varepsilon) - O(\varepsilon) \\ &= (\mathbf{y}_1 \theta_1 + \mathbf{q}_{j+1} \beta_{j1})^* \mathbf{q}_{j+2} - O(\varepsilon) \\ &= O(\varepsilon). \end{aligned}$$

In fact \mathbf{y}_1 will not become a threshold vector again unless roundoff boosts latent components of \mathbf{y}_1 in the current \mathbf{r} -vector up to the required level. There will be no more purgings until a new \mathbf{y}_i , say \mathbf{y}_2 , crosses the threshold at step m . Since \mathbf{Q}_m must be called in to compute $\mathbf{y}_2^{(m)}$ ($\equiv \mathbf{Q}_m \mathbf{s}_2^{(m)}$) there is little extra cost to computing $\mathbf{y}_1^{(m)}$ and storing it over $\mathbf{y}_1^{(j)}$ at this time.

At this point we remember the perturbations to \mathbf{T}_{j+1} associated with the purging at steps j and $j+1$. Thus $\mathbf{y}_2^{(m)}$ should be computed from

$$\mathbf{Q}_m \hat{\mathbf{s}}_2 = \mathbf{Q}_m \left(\mathbf{s}_2^{(m)} - \begin{bmatrix} \mathbf{s}_1^{(j)} \\ \mathbf{o} \end{bmatrix} \mu \right).$$

The effect of this correction is to make $\mathbf{y}_2^{(m)}$ more nearly orthogonal to $\mathbf{y}_1^{(m)}$. Consequently it is simpler, and more effective, to *ignore the perturbations* and simply orthonormalize $\mathbf{Q}_m \mathbf{s}_2^{(m)}$ against normalized $\mathbf{y}_1^{(m)}$ to get a new orthonormal pair when necessary. One possibility is to do this immediately as discussed at the beginning of this section; another possibility is to wait until the end.

A rigorous analysis of the SO process is beyond the scope of this book. Details concerning implementation of the method are given in [Parlett and Scott, 1979].

Section 13.11 points to developments since 1980.

Exercises on Section 13.9

13.9.1. Prove, by induction or otherwise, that

$$|\gamma_{\nu l}| \equiv |\mathbf{y}_{\nu}^* \mathbf{y}_l^{(j)}| \leq \kappa_j + O(\kappa_j^2).$$

13.9.2. Assume that $\mathbf{y}_l^{(j)*} \mathbf{r}_j = 0$, use Exercise 13.9.1, and use $\rho(\mathbf{y}_l; \mathbf{A}) = \theta_l^{(j)} + \rho(\mathbf{y}_l; \mathbf{A} - \theta_l^{(j)})$ to prove Lemma 13.9.1.

13.9.3. Prove Lemma 13.9.2.

13.9.4. Verify that the eigenvectors of $\mathbf{T}_j + \xi_1 \mathbf{s}_1 \mathbf{e}_j^*$ are given by (13.38).

13.9.5. Define, for each computed Ritz pair (θ, \mathbf{y}) , the sequence $\{\tau_j\}$ given by

$$\tau_{j+1} \equiv [\tau_j|\theta - \alpha_j| + \tau_{j-1}\beta_{j-1} + 3\varepsilon\|\mathbf{A}\|]/\beta'_j.$$

Show that if $|\mathbf{y}^*\mathbf{q}_{j-1}| \leq \tau_{j-1}$ and $|\mathbf{y}^*\mathbf{q}_j| \leq \tau_j$ then $|\mathbf{y}^*\mathbf{q}_{j+1}| \leq \tau_{j+1}$. Assume that $\mathbf{Ay} - \mathbf{y}\theta = \mathbf{q}\beta$ where $\mathbf{q} = \mathbf{q}_l$ for some $l < j - 1$ and $\beta < \sqrt{\varepsilon}\|\mathbf{A}\|$. Assume that there is no pause so that $\mathbf{r}'_j = \mathbf{q}_{j+1}\beta'_j$ is given as in Lemma 13.9.2 and use (13.25) for \mathbf{f}_j .

13.9.6. Write the governing equation for SO as

$$\mathbf{AQ}_j - \mathbf{Q}_j(\mathbf{T}_j + \mathbf{J}_j) = \mathbf{r}_j \mathbf{e}_j^* + \mathbf{F}_j,$$

where \mathbf{J}_j is strictly upper triangular and contains the appropriate multiples of eigenvectors \mathbf{s}_l used in computing threshold ‘‘Ritz vectors’’; thus $\mathbf{J}_j = \sum_l \mathbf{s}_l \xi_l \mathbf{e}_l^*$. Assume that $\kappa = \sqrt{\varepsilon}$, that $\mathbf{I} - \mathbf{Q}_j^* \mathbf{Q}_j = \mathbf{C}_j^* + \mathbf{C}_j$ with $\|\mathbf{C}_j\| \leq \kappa$, and that $\|\mathbf{J}_j\| \leq \kappa\|\mathbf{A}\|$. Imitate the proof of Paige’s theorem, neglect all quantities which are $O(\varepsilon\|\mathbf{A}\|)$, and deduce that, for $i = 1, \dots, j$,

$$\mathbf{y}_i^* \mathbf{q}_{j+1} \beta_{ji} = \gamma_{ii}^{(j)} - \mathbf{s}_i^* \mathbf{J}_j \mathbf{s}_i.$$

Show that for threshold vectors $\mathbf{s}_i^* \mathbf{J}_i \mathbf{s}_i$ is negligible.

13.10. Band (or Block) Lanczos

Even when used with full reorthogonalization the basic Lanczos algorithm cannot detect the multiplicity of the eigenvalues which it computes. The reasons are given in section 12.2. This limitation prompted the development of the block version of the Lanczos process which is capable of determining multiplicities up to the blocksize.

The idea is not to start with a single vector \mathbf{q}_1 but with a set of mutually orthonormal vectors which we take as the columns of a starting n -by- ν matrix \mathbf{Q}_1 . Typical values for ν are 2, 3, and 6. The generalization of the algorithm of section 13.1 to this new situation is straightforward and we shall describe it briefly. Associated with \mathbf{Q}_1 is the big Krylov subspace

$$\hat{\mathcal{K}}^{\nu j}(\mathbf{Q}_1) \equiv \text{span} (\mathbf{Q}_1, \mathbf{AQ}_1, \dots, \mathbf{A}^{j-1} \mathbf{Q}_1).$$

(We assume, for simplicity, that $A^{j-1}Q_1$ has rank ν but, as we shall see, the failure of this assumption causes no profound difficulties. It merely complicates the description of the process.) The Rayleigh-Ritz procedure applied to $\mathcal{K}^{\nu j}$ for $j = 1, 2, \dots$ produces the distinguished orthonormal basis $\hat{Q}_j \equiv (Q_1, Q_2, \dots, Q_j)$ and, in this basis, the projection of A is the block tridiagonal matrix

$$\hat{T}_j \equiv \begin{bmatrix} A_1 & B_1^* \\ B_1 & A_2 & & \\ & \ddots & \ddots & \\ & & \ddots & B_{j-1}^* \\ & & & B_{j-1} & A_j \end{bmatrix}, \quad A_i \text{ is } \nu \text{ by } \nu.$$

The B_i may be chosen to be upper triangular. By our assumption that $A^{j-1}Q_1$ has full rank it follows (Exercise 13.10.2) that each B_i is invertible.

13.10.1. Block Lanczos

R_0 , n by ν , is given. For $j = 1, 2, \dots$ repeat

1. $R_{j-1} = Q_j B_{j-1}$, the QR factorization of R_{j-1} (section 6.7).
2. $R_j \leftarrow A Q_j - Q_{j-1} B_{j-1}^*$, $Q_0 = O$.
3. $A_j \leftarrow Q_j^* R_j$.
4. $R_j \leftarrow R_j - Q_j A_j$.
5. Compute and test Ritz pairs. If satisfied then stop.

This algorithm is considerably more complicated than the simple Lanczos algorithm; in particular the computation of eigenpairs of \hat{T}_j has a cost proportional to $(\nu^2)(\nu j)$. The reward is that \hat{T}_j can have eigenvalues of multiplicity up to and including ν and thus can deliver approximations to multiple eigenvalues of A .

In practice, however, the Q_i 's lose mutual orthogonality as soon as convergence sets in. The remedies described earlier are available: either full re-orthogonalization or SO.

An attractive and successful alternative is to run the algorithm until the first Ritz pair converges and orthogonality is lost. Then start again with a new R_0 which is orthogonal to all known eigenvectors and iterate until all the wanted eigenvectors have been found. Much of the power of Lanczos is lost by restarting, and the apt choice of a new R_0 which uses information from previous runs is not easy. Nevertheless the method has been useful.

At this point we go back and, following an idea in [Ruhe, 1979], reformulate block Lanczos in a way which puts it on the same footing as the simple Lanczos algorithm. The original A is directly reduced to *band* matrix form.

13.10.2. Band Lanczos

Pick orthonormal $\mathbf{q}_1, \dots, \mathbf{q}_\nu$. Set $\mathbf{r} = \mathbf{q}_\nu$, $t_{\nu,0} = 1$, and for $j = 1, 2, \dots$, repeat:

1. $\mathbf{q}_{j+\nu+1} \leftarrow \mathbf{r}/t_{j+\nu-1,j-1}$.
2. $\mathbf{r} \leftarrow A\mathbf{q}_j - \sum_{i=j-\nu}^{j-1} \mathbf{q}_i t_{ij} \quad (\mathbf{q}_k = \mathbf{o} \text{ if } k < 1)$.

3. For $i = j, \dots, j + \nu - 1$,
$$\begin{cases} t_{ij} \leftarrow \mathbf{q}_i^* \mathbf{r}, \\ \mathbf{r} \leftarrow \mathbf{r} - \mathbf{q}_i t_{ij}. \end{cases}$$

4. $t_{i+\nu,j} \leftarrow \|\mathbf{r}\|$. If $t_{j+\nu,j} = 0$ reduce ν .
5. Compute and test Ritz pairs. If satisfied then stop.

In exact arithmetic the band algorithm is identical to the block algorithm. In practice it has the virtue of requiring no special QR factorization subprograms.

At the end of step j the computed quantities satisfy

$$A\hat{\mathbf{Q}}_j - \hat{\mathbf{Q}}_j \hat{\mathbf{T}}_j = \mathbf{R}_j \mathbf{E}_j^* + \hat{\mathbf{F}}_j \quad (13.39)$$

where $\mathbf{E}_j^* = (0, \dots, 0, \mathbf{l}_\nu)$ and $\hat{\mathbf{F}}_j$ accounts for roundoff in executing step j . The Ritz vectors are of the form $\mathbf{y}_i = \hat{\mathbf{Q}}_j \mathbf{s}_j$ where $\hat{\mathbf{T}}_j \mathbf{s}_i = \mathbf{s}_i \theta_i$. The accuracy of a Ritz pair may be assessed in the usual way; multiplying (13.39) by \mathbf{s}_i we find

$$\|A\mathbf{y}_i - \mathbf{y}_i \theta_i\| \doteq \beta_{ji} \equiv \|\mathbf{B}_j(\mathbf{E}_j^* \mathbf{s}_i)\|. \quad (13.40)$$

The band version is somewhat more complicated than the single vector algorithm. However if many of the wanted eigenvalues are known to have multiplicity ν then band Lanczos with SO will be more efficient than simple SO because all copies of a multiple eigenvalue will be found at the same step instead of one by one. On the other hand if all eigenvalues are simple then simple SO is preferable.

Exercises on Section 13.10

- 13.10.1. Do an operation count for one step of the band Lanczos algorithm.

Assume that the band QL algorithm is used to compute 2ν eigenvalues of \mathbf{T} (section 8.16) at two QL transformations per eigenvalue. Each \mathbf{s}_i requires $(\nu + 1)^2 \nu j$ ops.

13.10.2. Assume that $t_{i+\nu,j} > 0$, $j = 1, \dots, (j-1)\nu$. Why does T have no eigenvalues of multiplicity greater than ν ?

13.10.3. In exact arithmetic $Q_j^* Q_j = I_\nu$, and $A \hat{Q}_j - \hat{Q}_j \hat{T}_j = R_j E_j^*$. Prove directly from these relations and block Lanczos that

$$\hat{Q}_{j-1}^* R_j = 0$$

and

$$\text{if } A_j = Q_j^* A Q_j \quad \text{then } \hat{Q}_j^* R_j = 0.$$

13.11. Partial Reorthogonalization (PRO)

The studies of Paige showed that the simple Lanczos algorithm run in finite precision arithmetic always found the desired eigenvalues eventually. An excellent implementation and discussion can be found in [Cullum and Willoughby, 1985]. Recent studies [Greenbaum, 1989], [Greenbaum and Strakos, 1992] [Druskin and Knisherman, 1991], [Knisherman, 1995] consider how long one must wait, but the results are problem dependent and complicated. The obvious remedy (see section 13.7) is to explicitly orthogonalize each new Lanczos vector against all the preceding ones. The cost in storage and arithmetic is high and virtually precludes long Lanczos runs.

The compromise is to demand semiorthogonality among the computed vectors; $\cos(q_i, q_j) \leq \sqrt{\text{macheps}}$, $i \neq j$. The rewards are substantial; see [Parlett, 1992]. The next question is the cost of maintaining semiorthogonality.

The work of Scott described in sections 13.8 and 13.9 was only a first step in finding the answer. In 1983 Simon found and exploited a recurrence relation that governs the loss of orthogonality, namely, the vector $Q_j^* q_{j+1}$, and makes no reference to the Ritz vectors themselves. Whenever semiorthogonality is lost then the current vectors q_j and q_{j+1} are explicitly purged of their components in preceding Lanczos vectors. This process is called “partial reorthogonalization” (PRO) to distinguish it from Scott’s SO. See [Simon, 1984a and 1984b]. A feature of PRO is that no partially converged Ritz vectors need be computed.

At first SO and PRO were thought of as rivals, but Nour-Omid and I later realized that the two techniques are complementary. More precisely,

- (A) PRO is well suited to correct the orthogonality loss when a Ritz value first “converges” to an eigenvalue;
- (B) SO is well suited to correct for the “return of banished eigenvectors” described in section 13.8.

In addition SO is an efficient way to keep Lanczos vectors semiorthogonal to any known eigenvectors (either rigid body modes in structural analysis or eigenvectors computed during an earlier Lanczos run with a different shift). Both techniques are employed in the Boeing-Lanczos code which is perhaps the best implementation so far. See [Grimes, Lewis, and Simon, 1994].

13.12. Block Versus Simple Lanczos

A number of applications produce operators \mathbf{A} with some multiple eigenvalues. Some experts have concluded, a little hastily, that block Lanczos is the way to go. However a simple Lanczos algorithm that maintains semiorthogonality or strong linear independence can compute multiple eigenvalues although their copies arrive one at a time. If the multiplicity of an eigenvalue does not exceed the blocksize then the block algorithm will find all copies at the same step, an attractive feature.

On the other hand for sheer approximating power simple Lanczos is best. To fix ideas consider the problem of finding a smooth function of ξ that is 1 at $\xi = 0$ and whose maximal value on an interval $[\alpha, \beta]$ is as small as possible. One polynomial of degree 60 is, in general, preferable to a linear combination of six polynomials each of degree 10.

Researchers who have experimented with various blocksizes find the smaller sizes, such as 2 or 3, are most efficient. At the present time there has been no sophisticated comparison of a block version with a simple Lanczos code that maintains semiorthogonality.

Notes and References

The paper [Lanczos, 1950] began it all. Various connections with other methods are given in [Householder, 1964] and [Wilkinson, 1965]. A deeper understanding of the simple Lanczos algorithm emerged from the pioneering Ph.D. thesis [Paige, 1971]. Some important facets of that work were published in [Paige, 1972 and 1976] but a proof of the theorem proved in section 13.4 has not appeared in open literature.

The use of the bottom element of an eigenvector of T_j in assessing the accuracy of a Ritz vector was shown by Paige and also in [Kahan and Parlett, 1976] but was not picked up by the community of users. Nevertheless engineers, chemists, and physicists have used the simple Lanczos process with success in large, difficult problems by incorporating a variety of checks on the progress of the computation. See [Whitehead, 1972], [Davidson, 1975], and [van Kats and van der Vorst, 1976]. The idea of using the algorithm iteratively, i.e., restarting periodically, goes back to some of the early attempts to use Lanczos

on the computers available in the 1950s. The use of blocks is described in [Golub, 1973], [Cullum and Donath, 1974], and [Underwood, 1975]. Selective orthogonalization was introduced in [Parlett and Scott, 1979] and Scott now advocates that SO be applied only at the wanted end of the spectrum. The Lanczos algorithm is now being adapted to compute the whole spectrum of large A ; see [Cullum and Willoughby, 1985].

A Ritz value can appear to converge for several iterations to a number that is not an eigenvalue and then abruptly move away before settling down to a true eigenvalue. This “misconvergence” is explained in [Parlett, 1990].

We mention two recent developments related to the Lanczos algorithm. Ruhe takes the idea of shift-and-invert Lanczos, i.e., use $(A - \sigma)^{-1}$ as the operator, and extends it to present us with rational Krylov subspace methods. See [Ruhe, 1984]. The second idea was developed for the nonsymmetric eigenvalue problem but extends readily to the symmetric Lanczos algorithm. Saad developed the idea of explicitly restarting a Lanczos run when storage is limited. However, it seems preferable to restart the Lanczos algorithm implicitly by explicitly postmultiplying the matrix Q_j of Lanczos vectors by cleverly chosen orthogonal matrices. The price to be paid for this convenience is that some, but not all, of the recent Lanczos vectors must be discarded. See [Sorensen, 1992].

This page intentionally left blank

Subspace Iteration

14.1. Introduction

Subspace iteration is a straightforward generalization of both the power method and inverse iteration which were presented in Chapter 4. Given A and a subspace \mathcal{S} of \mathbb{E}^n there is no difficulty in *defining* a new subspace

$$A\mathcal{S} \equiv \{As : s \in \mathcal{S}\}.$$

Repetition of the idea produces the Krylov sequence of subspaces

$$\mathcal{K}(\mathcal{S}) = \{\mathcal{S}, A\mathcal{S}, A^2\mathcal{S}, \dots\}$$

and the questions are, how can the sequence be represented and is it worthwhile?

Before taking up these problems we must dispose of a very reasonable objection. In practice \mathcal{S} is given indirectly via an orthonormal basis $S = (x_1, x_2, x_3)$ say. Then $A^k S$ is spanned by $A^k S = (A^k x_1, A^k x_2, A^k x_3)$. Even if these columns are normalized they are simply the k th terms in three separate power sequences each of which will converge (slowly) to the dominant eigenvector z_n where $Az_n = z_n \lambda_n$ and $\lambda_n = \|A\|$. How can three slowly convergent sequences be preferable to one?

The answer is that $A^k S$ is a bad basis for a good subspace $A^k \mathcal{S}$. In section 14.4 we shall see that $A^k \mathcal{S}^p$ does converge to span $(z_n, z_{n-1}, \dots, z_{n-p+1})$, the dominant invariant subspace of dimension p , and, for large enough k , one application of the RR procedure (Chapter 11) produces good approximations to the individual eigenvectors.

If it is feasible to solve linear systems such as $(A - \sigma)x = b$, either by factoring $A - \sigma$ or by iteration, then subspace iteration can be effected with $(A - \sigma)^{-1}$ to obtain the p eigenvalues closest to σ together with their eigenvectors. This is the most common way in which the technique is used.

The main point is that the reward for working with several columns at once, and orthonormalizing then from time to time, is an improved factor in the linear convergence of successive subspaces. When several clustered eigenvalues are wanted the improvement is dramatic and more than makes up for the extra work incurred by working with a bigger subspace than is really wanted. In other configurations the method can be very slow. The difficulty is that the distribution of eigenvalues is usually unknown. Consequently it is not clear how large p , the dimension of \mathcal{S} , should be taken. Moreover the efficiency of the method often depends strongly on the value p .

Sophisticated versions of subspace iteration were developed during the 1960s and 1970s when the Lanczos algorithm was under a cloud. Now that easy-to-use, reliable Lanczos programs are available it is pertinent to ask whether subspace iteration should be laid to rest. The answer is no because there are a few circumstances in which its use is warranted.

1. If no secondary storage is available and the fast store can hold only a few n -vectors at a time, then there seems to be no choice but to discard previous vectors in the Krylov sequence and employ subspace iteration.
2. If the relative gap between the wanted eigenvalues and the others is enormous, as in inverse iteration with a good shift, then only one power step, or a few, will be needed for convergence. The situation is so good that the advantages of the Lanczos method are not needed. However Lanczos also works very well in these favorable circumstances.

The finest example of how far subspace iteration can be taken toward the goal of *automatic* computation is Rutishauser's program *ritzit* which is presented in Contribution II/9 in the Handbook. The program is complicated enough to be efficient in a wide variety of applications, but it is intelligible and every effort was made to keep down the number of decisions thrust on the user. The next three sections describe some of that work.

14.2. Implementations

The exposition is simplified by supposing that \mathbf{A} is positive definite and that its dominant p eigenpairs (α_i, \mathbf{z}_i) , $i = n - p + 1, \dots, n$ are to be found. The matrix \mathbf{A} is not modified and need not be known explicitly because its role in the program is that of an operator receiving vectors \mathbf{u} and returning vectors \mathbf{Au} . Any special features of \mathbf{A} which permit economies in storage or arithmetic operations should be exploited by the user in coding the subprogram which Rutishauser calls OP (for operator).

The ν th step in each implementation transforms an orthonormal basis $S_{\nu-1}$ of $A^{\nu-1}S$ into an orthonormal basis S_ν of $A^\nu S$. In addition there must be a test for convergence. (The rectangular matrix S_ν bears no relation to T_j 's eigenvector matrix S_j of Chapter 13.)

TABLE 14.1
Implementation 1: Simple subspace iteration.

	Action	Cost
(a)	Compute $C_\nu = AS_{\nu-1}$.	p calls on OP.
(b)	Test each column for convergence.	pn ops.
(c)	Orthonormalize $C_\nu = Q_\nu R_\nu$ (by modified Gram-Schmidt, section 6.7 or by QR).	$p(p+1)n$ ops.
(d)	Set $S_\nu = Q_\nu$.	0.

Remarks on Table 14.1. The columns of S_ν are *not* optimal approximations to the target eigenvectors from $\text{span } S_\nu$. Even if $S = \text{span}(z_n, z_{n-1})$ the columns of S_ν will converge only linearly to z_n and z_{n-1} as $\nu \rightarrow \infty$, despite the fact that they are already in $S = \text{span } S_0$ (Exercise 14.2.1). This defect suggests that the RR procedure (section 11.3) should be applied frequently to S_ν .

Figure 14.1 gives a picture of one step of Implementation 2.

TABLE 14.2
Implementation 2: Subspace iteration + RR.

	Action	Cost
(a)	Compute $C_\nu = AS_{\nu-1}$.	p calls on OP, C overwrites S .
(b)	Orthonormalize $C_\nu = Q_\nu R_\nu$ by modified Gram-Schmidt.	$p(p+1)n$ ops, Q overwrites C .
(c)	Form $\hat{H}_\nu = Q_\nu^*(AQ_\nu)$.	p calls on OP, $\frac{1}{2}p(p+1)n$ ops for \hat{H} .
(d)	Factor $\hat{H}_\nu = G_\nu \Theta_\nu G_\nu^*$.	κp^3 ($\kappa \approx 5$) (but κ depends on the spectral decomposition method).
(e)	Form $S_\nu = Q_\nu G_\nu$, the Ritz vectors for $A^\nu S$.	$k p^2 n$ ops, S overwrites Q .

Remarks on Table 14.2. If G_ν is found as a product of orthonormal matrices, say $G_\nu = P_1 \cdots P_k$ then (e) can be carried out at the same time as (d) by the

1. Compute

$$\begin{array}{ccc} C_\nu & = & A_\nu \\ \boxed{} & & \boxed{} \\ & & \boxed{\phantom{S_{\nu-1}}} \end{array}$$

2. Orthonormalize

$$\begin{array}{ccc} C_\nu & = & Q_\nu \\ \boxed{} & & \boxed{} \\ & & \boxed{} \end{array}$$

3. Form Rayleigh quotient

$$\begin{array}{ccc} \hat{H}_\nu & = & Q_\nu^* \\ \boxed{\phantom{\hat{H}_\nu}} & & \boxed{} \\ & & \boxed{} \end{array}$$

4. Factor

$$\begin{array}{ccc} \hat{H}_\nu & = & G_\nu^* \quad \boxed{} \\ \boxed{\phantom{\hat{H}_\nu}} & & \boxed{} \end{array}$$

θ_ν

5. Form

$$\begin{array}{ccc} S_\nu & = & Q_\nu \quad \boxed{} \\ \boxed{} & & \boxed{} \end{array}$$

FIG. 14.1. One step of Implementation 2.

following algorithm: $S_\nu = Q_\nu$, then for $i = 1, \dots, k$, $S_\nu \leftarrow S_\nu P_i$. There is no need for G_ν to be formed explicitly and a single n -by- p array suffices for S , C , and Q .

S_ν is the best basis in $A^\nu S$ and its columns converge to the z 's. However, the price is high; in particular the p extra calls to subroutine OP in (c) help to double the cost of each step as compared with Implementation 1. Fortunately there is a clever way to avoid these extra calls based on the fact that $\phi(A)$ has the same eigenvectors as A for any analytic function ϕ that does not coalesce distinct eigenvalues.

For the moment leave the function ϕ unspecified and seek the RR approximations to $\phi(A)$ from $A^\nu S$. As in the previous implementations let

$$C_\nu = AS_{\nu-1}. \quad (14.1)$$

We seek a new basis, say, $C_\nu F_\nu$ with the p -by- p matrix F_ν satisfying two conditions, namely

$$(C_\nu F_\nu)^*(C_\nu F_\nu) = I_p \quad (\text{orthonormality}) \quad (14.2)$$

and

$$(C_\nu F_\nu)^* \phi(A) (C_\nu F_\nu) \equiv \Delta_\nu^{-2} = \text{diagonal} \quad (\text{giving Ritz vectors}). \quad (14.3)$$

The right choice is $\phi(\xi) = \xi^{-2}$ so that (14.3) collapses into

$$(F_\nu^* S_{\nu-1}^* A) A^{-2} (A S_{\nu-1} F_\nu) = F_\nu^* F_\nu = \Delta_\nu^{-2}. \quad (14.4)$$

Thus $F_\nu \Delta_\nu$ will be orthonormal and (14.2) becomes

$$\begin{aligned} C_\nu^* C_\nu &= (F_\nu^{-*} \Delta_\nu^{-1}) \Delta_\nu^2 (\Delta_\nu^{-1} F_\nu^{-1}) \\ &= (F_\nu \Delta_\nu) \Delta_\nu^2 (F_\nu \Delta_\nu)^* \quad \text{by (14.4)}. \end{aligned} \quad (14.5)$$

Thus F_ν and Δ_ν are determined by the spectral decomposition of $C_\nu^* C_\nu$.

TABLE 14.3
Implementation 3: 2 applied to A^{-2} .

	Action	Cost
(a)	Compute $C_\nu = AS_{\nu-1}$.	p calls on OP, C overwrites S .
(b)	Compute $\overset{\circ}{H}_\nu = C_\nu^* C_\nu$.	$\frac{1}{2}p(p+1)n$ ops.
(c)	Factor $\overset{\circ}{H}_\nu = B_\nu \Delta_\nu^2 B_\nu^*$, the spectral decomposition.	κp^3 ops (κ depends on the method).
(d)	Form $S_\nu = C_\nu B_\nu \Delta_\nu^{-1}$ ($= C_\nu F_\nu$). kp^2n ops, S overwrites C .	

Remarks on Table 14.3. $\overset{\circ}{H}_\nu$ is the projection of A^2 onto $\text{span } S_{\nu-1}$ (Exercise 14.2.2). The new S_ν will differ from the S_ν of the previous implementation. Moreover it is not expressed as a product of orthonormal matrices. In 1971 Reinsch found a clever way to remove this blemish as follows. Let the QR factorization of C_ν be $Q_\nu R_\nu$ as in Implementation 2. Then, from (b) in Implementation 3,

$$\overset{\circ}{H}_\nu = C_\nu^* C_\nu = R_\nu^* Q_\nu^* Q_\nu R_\nu = R_\nu^* R_\nu.$$

The LR transformation was the forerunner of the QR transformation and when applied to $\overset{\circ}{H}_\nu$ it produces a more nearly diagonal matrix H_ν defined by the reversing the factors of $\overset{\circ}{H}_\nu$,

$$\begin{aligned} H_\nu &\equiv R_\nu R_\nu^* = R_\nu \overset{\circ}{H}_\nu R_\nu^{-1} \\ &= R_\nu B_\nu \Delta_\nu^2 B_\nu^* R_\nu^{-1}, \quad \text{by (c) of Implementation 3,} \\ &= (R_\nu B_\nu \Delta_\nu^{-1}) \Delta_\nu^2 (\Delta_\nu B_\nu^* R_\nu^{-1}). \end{aligned}$$

It may be verified (Exercise 14.2.3) that $P_\nu \equiv R_\nu B_\nu \Delta_\nu^{-1}$ is the orthonormal matrix of eigenvectors of H_ν . So, from (d) in Implementation 3,

$$S_\nu = C_\nu B_\nu \Delta_\nu^{-1} = Q_\nu R_\nu B_\nu \Delta_\nu^{-1} = Q_\nu P_\nu.$$

At last we have the preferred implementation, called ritzit, given in Table 14.4.

TABLE 14.4
Implementation 4: [in ritzit].

	Action	Cost
(a)	Compute $C_\nu = AS_{\nu-1}$.	p calls on OP, C over S.
(b)	Compute $C_\nu = Q_\nu R_\nu$.	$p(p+1)n$ ops, Q over C.
(c)	Form $H_\nu = R_\nu R_\nu^*$.	$\frac{1}{3}p^3$ ops.
(d)	Factor $H_\nu = P_\nu \Delta_\nu^2 P_\nu^*$.	kp^3n ops, ($\kappa \approx 5$).
(e)	Form $S_\nu = Q_\nu P_\nu$.	kp^2n ops, S over Q.

Note: In Table 14.4 steps (d) and (e) should be done together.

Exercises on Section 14.2

- 14.2.1. Take $S_0 = (s_1, s_2)$ with $s_i = (z_n \pm z_{n-1})/\sqrt{2}$ and verify that $S_\nu e_1$ converges to z_n . What is the convergence factor?

- 14.2.2. Show that $\overset{\circ}{H}_\nu$ is the projection of A^2 onto $\text{span}(S_{\nu-1})$. See sections 1.4 and 11.4 for definitions and discussion of projections.
- 14.2.3. Show that $P_\nu \equiv R_\nu B_\nu \Delta_\nu^{-1}$ is actually orthogonal.
- 14.2.4. Since $\phi(A)$'s eigenvectors are the same as A 's why are the Ritz vectors from S_ν for $\phi(A)$ not the same as the Ritz vectors from S_ν for A ? Hint: See section 11.4.
- 14.2.5. Make a table showing the total op count and storage requirement for each of the four implementations.

14.3. Improvements

14.3.1. Chebyshev Acceleration

The technique described in this section is used in many branches of numerical analysis and so warrants more than a brief mention.

Even with Implementation 4 the RR procedure is expensive and so it is tempting to take a few steps of the basic power method between each invocation of RR. To be specific suppose that the algorithm computes, in turn, $S_{\nu+j} = AS_{\nu+j-1}$, $j = 1, \dots, m$. However, it is no more trouble to incorporate shifts σ_j and compute instead $S_{\nu+j} = (A - \sigma_j)S_{\nu+j-1}$, $j = 1, \dots, m$. Thus the algorithm could compute

$$S_{\nu+m} = \phi(A)S_\nu$$

for *any* monic polynomial ϕ of the degree m . The problem is to select a helpful ϕ .

After the first application of the RR procedure the program possesses approximations $\theta_{-1}, \dots, \theta_{-p}$ to dominant eigenvalues. The θ_{-i} are the eigenvalues of H_ν or $\overset{\circ}{H}_\nu$ or \hat{H}_ν . A little reflection shows that a useful, but by no means optimal, choice is a ϕ which is as small as possible on the interval $[\theta_1, \theta_{-p}]$. Since θ_1 is unknown it is customary to use $[0, \theta_{-p}]$ if A is positive semidefinite and $[-\theta_{-p}, \theta_{-p}]$ otherwise. This problem in approximation theory is solved by the Chebyshev polynomials adapted to the appropriate interval. These polynomials are described briefly in Appendix B. The beautiful fact is that there is no need to know (and then use) the zeros of the appropriate Chebyshev polynomial $T_m(\xi)$ as shifts σ_j because a simple three-term recurrence with constant coefficients permits the calculation of $T_j(A)$ from $T_{j-1}(A)$ and $T_{j-2}(A)$. Specifically if T_m is adapted to the interval $[-e, e]$ then, for $j = 2, \dots, m$,

$$T_{\nu+j}(\xi) = \frac{2\xi}{e} T_{\nu+j-1}(\xi) - T_{\nu+j-2}(\xi),$$

$$S_{\nu+j} = \frac{2}{e} AS_{\nu+j-1} - S_{\nu+j-2}.$$

Moreover there is no need to save $S_{\nu+j-2}$ provided that each column is updated from S_ν all the way to $S_{\nu+m}$ in turn. This means that only two extra n -vectors of storage are needed to effect this acceleration of the convergence of S_ν .

How should m be chosen? An important advantage of the recurrence is that it is independent of m and so m can be freely changed during the computation. In section 14.4 it will be shown that the convergence factor for the dominant Ritz vector is $\theta_{-p-1}/\theta_{-1}$. If this ratio is small, like 0.1, then there is little need for acceleration and m can be held at 1. However when that ratio is close to 1, like 0.98, then there is much to be gained from a large m . The only constraint is that the columns of $S_{\nu+m}$ must be kept fully independent. In his *ritzit* Rutishauser requires

$$\|S_{\nu+m}\| \approx T_m(\theta_{-1}/\theta_{-p}) \leq \cosh 8 < 1500.$$

The program which embodies this powerful device is remarkably simple and even elegant. However, to keep a proper perspective it must be recalled that the Lanczos algorithm, when run for m steps, gives even more than Chebyshev acceleration using the *unknown optimal* interval $[\alpha_1, \alpha_{-p-1}]$. This observation is based on the theory of section 12.4.

14.3.2. Randomization

This is a device to protect the algorithm from an unhappy choice of an initial subspace \mathcal{S} which is effectively orthogonal to one of the desired eigenvectors.

After each orthogonalization of the basis vectors the one with the innermost Rayleigh quotient θ_{-p} is replaced by a random vector orthogonal to the rest of the basis. This is a simple way of making it *most* unlikely that any wanted eigenvectors will be missed.

It is advisable to wait until the Rayleigh quotients have settled down and Rutishauser waits until three RR approximations have been made. Here is an example of one of the rare ad hoc parameters that occur in the program. The alternative of monitoring the Rayleigh quotients for stabilization of θ_{-m} seems to be more complicated than is warranted.

Note that this device impinges on Chebyshev acceleration because the quantity θ_{-p} , the p th Rayleigh quotient, determines the interval for the Chebyshev polynomial. Some care is needed in the precise designation of the ends of the interval to keep them free from spurious current values of θ_{-p} . The reader is referred to Contribution II/9 in the Handbook for details.

14.3.3. Termination Criteria

The RR approximations θ_i should move monotonically to their limits α_i ($i = \pm 1, \pm 2, \dots$), in the absence of roundoff, as subspace iteration continues. In *ritzit* Rutishauser accepts a θ_i as soon as it stagnates (thus achieving accuracy close to working precision). No test is made on the Ritz vectors y_i until θ_i has been accepted.

Recall from section 11.7 that

$$|\sin \angle(y_i, z_i)| \leq \frac{\|r_i\|}{\text{gap}},$$

where $r_i = (A - \theta_i)y_i$ is known but the gap, namely, $\min\{|\alpha_{i-1} - \alpha_i|, |\alpha_i - \alpha_{i+1}|\}$, is not. Rutishauser uses the θ 's to approximate the gap and thus obtains a computable estimate of the error angle which can be tested for the required accuracy.

Those experienced in numerical computation know that preset tolerances can occasionally fail to be met. Hence Rutishauser builds into the simple error measure ($\|r_i\|/\text{gap}$) a slow but sure decay which ensures that the program will terminate even in those cases in which the desired accuracy is greater than the program can achieve. However, Rutishauser did not make use of the residual norm for clustered eigenvalues (see section 11.5) nor of the refined error bounds in Chapter 10.

*14.4. Convergence

Let $Z \equiv (z_1, \dots, z_m)$, where $Az_i = z_i\alpha_i$, $i = 1, \dots, m$ be the matrix of wanted eigenvectors. To be specific we assume that

$$0 < \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_m < \alpha_{m+1} \leq \dots, \quad (14.6)$$

so that $Z \equiv Z^m \equiv \text{span } Z$ is the dominant invariant subspace under A^{-1} . Let S be any m -dimensional subspace of E^m and $\{A^{-k}S : k = 0, 1, 2, \dots\}$ the associated sequence generated by subspace iteration. The quantity $\angle(Z, A^{-k}S)$, defined in section 11.7, is too crude a measure of how well $A^{-k}S$ approximates Z because the user wants to know how well specific vectors, such as z_1 , can be approximated from $A^{-k}S$. So the natural objects to study are

$$\psi_j^{(k)} \equiv \angle(z_j, A^{-k}S) \equiv \min \angle(z_j, x) \quad \text{over } x \in A^{-k}S.$$

We want to see how fast $\psi_j^{(k)} \rightarrow 0$ and, at the same time, make the proof as similar to the one-dimensional case as possible. The reader might wish to refer to section 4.2.

In order to have convergence of $\{A^{-k}S\}$ to Z , and not some other invariant subspace, as $k \rightarrow \infty$ it is necessary to assume that $\angle(Z, S) < \pi/2$, or, equivalently, that for any orthonormal basis S of S

$$Z^*S \text{ is invertible.} \quad (14.7)$$

A useful notion in the analysis given below is the (matrix) angle Ψ between Z and S defined by

$$\Psi = \cos^{-1}(Z^*SS^*Z)^{1/2}. \quad (14.8)$$

Functions of matrices can be defined in various ways, but in our case Ψ is only needed to give meaning to matrices such as $\sin \Psi$ and $\tan \Psi$ which are messy when expressed in terms of S and Z . Assumption (14.7) ensures that Ψ is well defined. It is not diagonal.

In analogy with the one-dimensional case there exists an orthonormal basis S such that $Z^*S = S^*Z = \cos \Psi$ (Exercise 14.4.1). This basis S can be expressed in terms of Z as follows:

$$\begin{array}{c|c|c|c|c} S & = & Z & + & \cos \Psi \\ \hline & & & & J \\ \hline & & & & \sin \Psi \end{array} \quad (14.9)$$

where (Exercise 14.4.2)

$$J^*J = I_m, \quad Z^*J = O. \quad (14.10)$$

Theorem 14.4.1. *Under assumptions (14.6) and (14.7) on S and Z each eigenvector z_i , $i \leq m$, satisfies*

$$\tan \angle(z_i, A^{-k}S) \leq \left(\frac{\alpha_i}{\alpha_{m+1}} \right)^k \tan \angle(Z, S). \quad (14.11)$$

Proof. Premultiply (14.9) by \mathbf{A}^{-k} and postmultiply by $(\sec \Psi)\Lambda^k$ to find

$$\begin{aligned}\mathbf{A}^{-k}\mathcal{S}(\sec \Psi)\Lambda^k &= \mathbf{A}^{-k}\mathbf{Z}\Lambda^k + \mathbf{A}^{-k}\mathbf{J}(\tan \Psi)\Lambda^k \\ &= (\mathbf{Z}\Lambda^{-k})\Lambda^k + \mathbf{A}^{-k}\mathbf{J}(\tan \Psi)\Lambda^k,\end{aligned}\quad (14.12)$$

where $\Lambda = \text{diag}(\alpha_1, \dots, \alpha_m)$. The key fact is that \mathbf{Z} is orthogonal to $\mathbf{A}^{-k}\mathbf{J}$ (Exercise 14.4.3) and it is convenient to rewrite $\mathbf{A}^{-k}\mathbf{J}$ as

$$\mathbf{A}^{-k}\mathbf{J} = \mathbf{J}_k\Omega_k, \quad \Omega_k = (\mathbf{J}^*\mathbf{A}^{-2k}\mathbf{J})^{\frac{1}{2}}, \quad (14.13)$$

so that \mathbf{J}_k is orthonormal. To bound Ω_k note that

$$\|\Omega_k\|^2 = \max_{\|v\|} v^*\mathbf{J}^*\mathbf{A}^{-2k}\mathbf{J}v \leq \alpha_{m+1}^{-2k}, \quad (14.14)$$

since $\mathbf{Z}^*\mathbf{J} = \mathbf{O}$. Now consider the j th columns in (14.12) to find

$$\mathbf{x}_j^{(k)} \equiv \mathbf{A}^{-k}\mathcal{S}(\sec \Psi)\mathbf{e}_j\lambda_j^{-k} = \mathbf{z}_j + \mathbf{u}_j, \quad \mathbf{u}_j = \mathbf{J}_k\Omega_k(\tan \Psi)\mathbf{e}_j\alpha_j^k,$$

$$\begin{aligned}\tan \angle(\mathbf{z}_j, \mathbf{A}^{-k}\mathcal{S}) &\leq \tan \angle(\mathbf{z}_j, \mathbf{x}_j^{(k)}) = \|\mathbf{u}_j\|/1 \\ &= \|\Omega_k(\tan \Psi)\mathbf{e}_j\|\alpha_j^k, \quad \text{using (14.13),} \\ &\leq \left(\frac{\alpha_j}{\alpha_{m+1}}\right)^k \tan \angle(\mathbf{z}_j, \mathcal{S}(\sec \Psi)\mathbf{e}_j), \\ &\quad \text{by (14.14) and Exercise 14.4.4,} \\ &\leq \left(\frac{\alpha_j}{\alpha_{m+1}}\right)^k \tan \angle(\mathcal{Z}, \mathcal{S}). \quad \square\end{aligned}\quad (14.15)$$

One way to appreciate this result is to contrast the computation of \mathbf{z}_1 by subspace iteration and inverse iteration (Exercise 14.4.5).

Theorem 14.4.1 shows that a certain sequence $\mathbf{x}_j^{(k)} : \mathbf{x}_j^{(k)} = \mathbf{A}^{-k}\mathcal{S}(\sec \Psi)\mathbf{e}_j \rightarrow \mathbf{z}_j$ as $k \rightarrow \infty$, but it does not address the behavior of the sequence $\mathbf{y}_j^{(k)}$ actually computed by, say, Implementation 4. Some authors invoke the “optimality” of Ritz vectors to conclude that $\mathbf{y}_j^{(k)}$ must converge at least as quickly as does $\mathbf{x}_j^{(k)}$. Such an argument is mistaken because $\mathbf{y}_j^{(k)}$ is *not* the closest unit vector in $\mathbf{A}^{-k}\mathcal{S}$ to \mathbf{z}_j . Thus the \mathbf{x} ’s might converge quicker than the \mathbf{y} ’s.

Rutishauser shows that $y_j^{(k)} \rightarrow x_j^{(k)}$ at the same asymptotic rate as $x_j^{(k)} \rightarrow z_j$ and consequently the same improved factor α_j/α_{m+1} governs the linear convergence of $\{y_j^{(k)}\}$ to z_j . The difficulty lies in the complicated nature of Ritz vectors and a brief digression is needed before establishing these claims. If B is a nonorthonormal basis for $\text{span } B$ then each Ritz approximation for A^2 from $\text{span } B$ is of the form Bt where t satisfies

$$(B^*A^2B - \mu^2B^*B)t = o, \quad (14.16)$$

and μ is the associated Ritz value (Exercise 14.4.6).

Theorem 14.4.2. *When subspace iteration uses Implementation 4 then each Ritz vector $y_i^{(k)}$ is related to the vector $x_i^{(k)}$ of Theorem 14.4.1 as $k \rightarrow \infty$, by*

$$\begin{aligned} \sin \angle(y_i^{(k)}, x_i^{(k)}) &= O\left(\left(\frac{\alpha_i}{\alpha_{m+1}}\right)^k\right), \\ i &= 1, \dots, m. \end{aligned}$$

Proof. In the basis $B \equiv A^{-k}S(\sec \Psi)\Lambda^k$, given in (14.12), $x_i^{(k)}$ is represented by e_i and $y_i^{(k)}$ is represented by a solution t_i of (14.16). On substituting the right side of (14.12) for B the formula (14.16) becomes

$$[\Lambda^2 - \mu^2 + (\alpha_{m+1}^{-1}\Lambda)^k H_k(\alpha_{m+1}^{-1}\Lambda)^k] t = o, \quad (14.17)$$

where, by Exercise 14.4.7,

$$\|H_k\| \leq (\alpha_{m+1} \|\tan \Psi\|)^2. \quad (14.18)$$

The perturbation to $\Lambda^2 - \mu^2$ in (14.17) vanishes as $k \rightarrow \infty$, and for large enough k there is a μ_i close to α_i and a t_i close to e_i . Next consider k so large that

$$|\mu_i - \alpha_i| \leq \delta \equiv \frac{1}{2} \min |\alpha_i - \alpha_j| \quad \text{over } \alpha_j \neq \alpha_i.$$

For such k $(\alpha_i^2, \mathbf{e}_i)$ is a good approximate eigenpair to (μ_i^2, \mathbf{t}_i) and the residual vector \mathbf{r}_i for (14.17) satisfies

$$\begin{aligned}\|\mathbf{r}_i\| &= \left\| \left[\Lambda^2 + (\alpha_{m+1}^{-1} \Lambda)^k \mathbf{H}_k (\alpha_{m+1}^{-1} \Lambda)^k - \alpha_i^2 \right] \mathbf{e}_i \right\| \\ &= \left\| (\alpha_{m+1}^{-1} \Lambda)^k \mathbf{H}_k \mathbf{e}_i \right\| \left(\frac{\alpha_i}{\alpha_{m+1}} \right)^k \\ &\leq \left(\frac{\alpha_m}{\alpha_{m+1}} \right)^k \left(\frac{\alpha_i}{\alpha_{m+1}} \right)^k (\alpha_{m+1} \|\tan \Psi\|)^2.\end{aligned}\quad (14.19)$$

The last inequality uses (14.18). By Theorem 11.7.1 (the gap results)

$$\sin \angle(\mathbf{t}_i, \mathbf{e}_i) \leq \frac{\|\mathbf{r}_i\|}{\delta}. \quad (14.20)$$

For large k the basis \mathbf{B} is almost orthonormal,

$$|\mathbf{b}_i^* \mathbf{b}_j| \leq \left(\frac{\alpha_i}{\alpha_{m+1}} \right)^k \left(\frac{\alpha_j}{\alpha_{m+1}} \right)^k \|\tan \Psi\|^2, \quad i \neq j. \quad (14.21)$$

When the effects of (14.20) and (14.21) are combined (Exercise 14.4.8) it turns out that (14.21) dominates in the limit and

$$\sin \angle(\mathbf{y}_i^{(k)}, \mathbf{x}_i^{(k)}) \leq \left(\frac{\alpha_i}{\alpha_{m+1}} \right)^k \|\tan \Psi\| \quad \text{as } k \rightarrow \infty, \quad (14.22)$$

which completes the proof. \square

Exercises on Section 14.4

- 14.4.1. Start with any orthonormal basis \mathbf{S} of \mathcal{S} and then find m -by- m orthogonal \mathbf{G} such that $\hat{\mathbf{S}} = \mathbf{SG}$ satisfies $\mathbf{Z}^* \hat{\mathbf{S}} = \hat{\mathbf{S}}^* \mathbf{Z} = \cos \Psi$, a positive definite matrix.
- 14.4.2. Continuing the previous exercise verify that $\mathbf{J} = \hat{\mathbf{S}} \operatorname{cosec} \Psi - \mathbf{Z} \cot \Psi$.
- 14.4.3. Show that $\mathbf{Z} \perp \mathbf{A}^{-k} \mathbf{J}$.
- 14.4.4. Show that $\tan \angle(\mathbf{z}_j, \mathbf{S} \sec \Psi \mathbf{e}_j) = \|(\tan \Psi) \mathbf{e}_j\|$.
- 14.4.5. Find expressions for the number of steps required to reduce the error angle by a factor of 1000 by subspace iteration. Divide your answer by m and compare with the result for inverse iteration. Is this comparison fair?

14.4.6. An orthonormal basis for $\text{span } \mathbf{B}$ is $\mathbf{B}(\mathbf{B}^*\mathbf{B})^{-\frac{1}{2}}$. Use this to establish (14.16).

14.4.7. Show that $H_k = \tan \Psi (\Omega_{k-1}^2 - \mu^2 \Omega_k^2) \tan \Psi \alpha_{m+1}^{2k}$ and then confirm (14.18) for large enough k .

14.4.8. Write

$$\mathbf{t}_i = \mathbf{e}_i + \eta_i \mathbf{f}_i, \quad \|\mathbf{f}_i\| = 1, \quad |\eta_i| = O\left(\left(\frac{\alpha_i}{\alpha_{m+1}} \frac{\alpha_m}{\alpha_{m+1}}\right)^k\right).$$

Also

$$\cos \angle(\mathbf{y}_i, \mathbf{x}_i) = \mathbf{t}_i^* \mathbf{B} \mathbf{B} \mathbf{e}_i / \|\mathbf{B} \mathbf{t}_i\| \cdot \|\mathbf{B} \mathbf{e}_i\|.$$

Use these two facts to establish (14.22).

14.5. Sectioning

There is a variant of the implementation given in sections 14.2 and 14.3 which is aimed at the subclass of large problems in which all the eigenvalues in a given interval (α, β) must be computed, however many there may be. The interval may be quite wide. Of course the spectrum could be sliced at α and β to determine this number, but what makes the problem interesting is the desire to keep the number of factorizations to a minimum, preferably to one.

The method uses only one application of the RR procedure. The iterative part is transferred to determining starting vectors which actually span the invariant subspace associated with (α, β) . The expensive step is factoring $\mathbf{A} - \mu$ for some μ at, or close to, the midpoint $(\alpha + \beta)/2$, but the factors permit a relatively swift execution of inverse iteration. The goal is to distinguish rapidly between three possible configurations: (a) no eigenvalues in (α, β) , (b) one or more eigenvalues close to μ , (c) eigenvalues bunched near α and β but not near μ . The distinction is made by monitoring carefully the rate of convergence of inverse iteration on a single vector which is kept orthogonal to directions already found to be in the invariant subspace associated with (α, β) . Inverse iteration is efficient for cases (a) and (b) but not for (c). By deleting (c) early the program can abandon the current shift and, *only when warranted*, start again with a well-chosen shift near α and another near β . This brings the total to three factorizations for difficult cases.

In case (b) the iteration is stopped as soon as the vector is seen to be lying in the desired subspace; there is no waiting about for convergence. Since the program is given α and β it can make use of Chebyshev acceleration to further reduce the components in those eigenvectors belonging to eigenvalues

outside (α, β) . In this way columns are added to the starting matrix until case (a) is obtained, signaling that the invariant subspace has been captured. One application of the RR procedure, using \mathbf{A} , yields the wanted eigenvectors. The all-important details are given in [Jensen, 1972] where sectioning was first presented.

A different type of sectioning has been proposed recently in [Wilson, 1978], apparently without knowledge of Jensen's work. By minimizing an operation count and making certain simplifying assumptions Wilson concludes that a good choice for blocksize in subspace iteration is $\sqrt{m/2}$, where m is the half-bandwidth of \mathbf{A} . To be specific, suppose that the optimal blocksize for the given matrix is declared to be 10.

Wilson's strategy makes freer use of factorizations than does Jensen's and uses spectrum slicing to find subintervals of (α, β) each containing about seven eigenvalues. Wilson then begins with the subinterval nearest β (i.e., the innermost end of the big interval) and uses subspace iteration to find the eigenvalues and eigenvectors in it. After that he works down toward α using some vectors from the previous iteration as starting vectors and making sure that all starting vectors are orthogonal to the eigenvectors found previously.

The usual practice had been to work from α to β .

When the bandwidth w is very small relative to n then factorization is comparable in cost to other vector operations such as orthogonalization. In that case it pays to use spectrum slicing and the techniques described in section 3.5 to locate an eigenvalue accurately and then find the eigenvector with one or two steps of inverse iteration. When the eigenpairs are found one by one there is no need to use blocks which are larger than the number of wanted eigenvectors.

Notes and References

An early work analyzing block methods was [Bauer, 1957], but the most important references on subspace iteration are [Rutishauser, 1969 and 1971b]. The geometrical aspects are discussed in [Parlett and Poole, 1973].

The civil engineers used and developed the method somewhat independently of the numerical analysts and the language used is therefore different. For example, Jennings uses the term "interaction matrix" for what we call the projection of \mathbf{A} onto a subspace. Recent references which describe tests of several variants are [Bathé and Wilson, 1976] and [Jennings and McKeown, 1992].

The development of sectioning, [Jensen, 1972], was very valuable for large problems. See [Jennings and Agar, 1978] for recent developments.

This page intentionally left blank

The General Linear Eigenvalue Problem

15.1. Introduction

This chapter takes up the task of computing some, or all, of the pairs (λ, z) such that $(A - \lambda M)z = 0$, $z \neq 0$ given two symmetric matrices A and M . The scalar λ is called an *eigenvalue* (or *root*) of the pair (A, M) and z is an eigenvector. In [Gantmacher, 1959] the matrix $A - \lambda M$ is called a matrix *pencil*. The rather strange use of the word “pencil” comes from optics and geometry: an aggregate of (light) rays converging to a point does suggest the sharp end of a pencil and, by a natural extension, the term came to be used for any *one parameter family* of curves, spaces, matrices, or other mathematical objects. In structural analysis A is the stiffness matrix (usually written as K) and M is the mass matrix.

Two pencils (A_1, M_1) and (A_2, M_2) are said to be *equivalent* if there exist invertible matrices E and F such that

$$A_2 = EA_1F, \quad M_2 = EM_1F. \quad (15.1)$$

In some contexts (15.1) is called *strict equivalence*. The roots of two equivalent pencils are the same and the eigenvectors are simply related (Exercise 15.1.1). Moreover the roots $\lambda_1, \lambda_2, \dots$ are zeros of the *characteristic polynomial*

$$\chi(\tau) \equiv \det[\tau M - A]. \quad (15.2)$$

Symmetry is too precious a property to surrender lightly and so we shall consider only *congruent* pencils, that is, equivalent pencils in which $E = F^*$. It is not necessary that $F^* = F^{-1}$ in order to preserve eigenvalues, but in practice orthonormal F are popular because by Fact 1.10 in Chapter 1, $\|A_1\| = \|A_2\|$, $\|M_1\| = \|M_2\|$ and so no dangerous element growth can occur in carrying out the congruence transformation explicitly.

It is natural to seek the analogue to the spectral theorem, Fact 1.4 in Chapter 1, and hence to find the canonical form (i.e., simplest pair) in each class of congruent pencils. The answer is known as the simultaneous reduction to diagonal form.

For some pencils (A, M) there is an invertible F such that

$$F^*AF = \Phi = \text{diag}(\phi_1, \dots, \phi_n),$$

$$F^*MF = \Psi = \text{diag}(\psi_1, \dots, \psi_n).$$

There are two important departures from the spectral theorem: (1) although the ratios ϕ/ψ , $i = 1, \dots, n$ are unique the matrices Φ and Ψ are not and (2) the reduction is not always possible.

If $\psi_i \neq 0$, $i = 1, \dots, n$ then it is possible to normalize the pair Φ, Ψ by making $\Psi = I$ and $\Phi = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, but this is not always advisable. This point is elaborated in sections 15.2 and 15.3 which present basic material on matrix pencils or, equivalently, on pairs of quadratic forms.

With that preparation in hand sections 15.4, 15.5, 15.6, and 15.7 discuss the numerical reduction of (A, M) to (Λ, I) while the rest of the chapter is concerned with extending the methods of earlier chapters to large pencils (A, M) .

Exercise on Section 15.1

- 15.1.1. Let $A_2 = EA_1F$, $M_2 = EM_1F$. Show that the roots of (A_1, M_1) and (A_2, M_2) are the same and show how the eigenvectors are related.

15.2. Symmetry Is Not Enough

The generalized eigenvalue problem is, in principle, more difficult than the standard one because of three new phenomena which can occur. Fortunately they can be illustrated on 2-by-2 pencils.

	Eigenpairs
I. $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$	$(1, e_1); \quad (\frac{0}{0}, e_2).$
II. $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad M = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$	$(\frac{1}{0}, e_1); \quad (\frac{0}{1}, e_2).$
III. $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$	$\left[i, \begin{pmatrix} i \\ -1 \end{pmatrix} \right]; \quad \left[i, \begin{pmatrix} i \\ 1 \end{pmatrix} \right].$
	(Here $i^2 = -1.$)

In (I) all scalars are eigenvalues for e_2 . In (II) ∞ is an eigenvalue for a well-defined eigenvector e_1 . In (III) there are complex eigenvalues even though A and M are symmetric. This last phenomenon is clarified by the following surprising result.

Theorem 15.2.1. *Any real square matrix B can be written as $B = AM^{-1}$ or $B = M^{-1}A$ where A and M are suitable symmetric matrices.*

The proof is the subject of Exercise 15.2.3. It follows that any difficulty arising in the computation of B 's eigenvalues can afflict the task of solving $(A - \lambda M)x = (AM^{-1} - \lambda)Mx = 0$.

Pencils, like (I), which have $\chi(t) = 0$ for all t are called *singular*. Often, but not always, pencils are singular because A and M have some null vectors in common, $Ax = 0, Mx = 0, x \neq 0$. Such x are, strictly speaking, eigenvectors and any number whatsoever is a matching eigenvalue. This behavior is unorthodox to say the least and the first part of any analysis of a pencil should be to find any common null space and get rid of it. Theoretically such a subspace is removed by *deflation*, that is, by restricting A and M to the *complementary* invariant subspace in \mathcal{E}^n .

For practical work the danger which suggests itself is that there may be vectors x which are nearly annihilated by both A and M . In such a case an eigenvalue program may compute some innocent looking eigenvalues which are not only hypersensitive to perturbations in A and M but whose very presence degrades the stability of the other eigenvalues. An example will give substance

to these remarks.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-8} \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \times 10^{-8} \end{bmatrix} : \quad \text{Solution } (1, \mathbf{e}_1), (\frac{1}{2}, \mathbf{e}_2). \quad (15.3)$$

Now perturb \mathbf{A} by elements of the order 10^{-8} . Let

$$\mathbf{A}' = \begin{bmatrix} 1 & \sqrt{2} \times 10^{-8} \\ \sqrt{2} \times 10^{-8} & 2 \times 10^{-8} \end{bmatrix}. \quad (15.4)$$

The roots of $(\mathbf{A}', \mathbf{M})$ are approximately 1 ± 10^{-4} with eigenvectors approximately $\mathbf{e}_2 \mp 10^{-4}\mathbf{e}_1$. Thus a change of 10^{-8} in \mathbf{A} has changed the root 1 by 10^{-4} , a magnification of 10,000, while the root $\frac{1}{2}$ changes completely. Eigenvalues such as $\frac{1}{2}$ are *ill disposed*, an apt term coined by G.W. Stewart. Observe that (15.3) is equivalent to

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} : \quad \text{Solution } (1, \mathbf{e}_1), (\frac{1}{2}, \mathbf{e}_2) \quad (15.5)$$

which is perfectly conditioned.

Mathematically, in exact arithmetic, we cannot distinguish between (15.3) and (15.5). It may be vital for certain applications to recognize that (15.3) and (15.5) are not equally permissible representations of a problem, but such knowledge is external to the standard theory of matrix pencils and must be supplied as an extra element of the problem in hand. It is best if this can be done by designating the scaling which is appropriate for the data.

It is worth emphasizing that infinite eigenvalues, Case II, are not necessarily ill disposed. In fact, the infinite eigenvalues of (\mathbf{A}, \mathbf{M}) are the zero eigenvalues of (\mathbf{M}, \mathbf{A}) and this is one more hint that each eigenvalue of a pencil should be presented by a pair and not reduced to a single number. Our normal measure of the separation of two eigenvalues, namely, $|\lambda_i - \lambda_j|$, is also called into question. Since the roles of \mathbf{A} and \mathbf{M} are interchangeable it would seem natural to use a measure of the separation of λ_i and λ_j which is invariant under reciprocation.

The *chordal metric*,

$$\chi(\lambda, \mu) \equiv |\lambda - \mu| / \sqrt{1 + \lambda^2} \sqrt{1 + \mu^2},$$

enjoys this property (Exercises 15.2.4 and 15.2.5) and crops up in the study of non-Euclidean geometry.

With this tool in hand it should be possible to generalize the error bounds in Chapters 4, 10, and 11 to cover the pencils (\mathbf{A}, \mathbf{M}) in a properly invariant manner. This has been done in masterly fashion in [Stewart and Sun, 1990].

However, the error bounds in this book do not require that the matrices being compared be close in any sense.

Exercises on Section 15.2

15.2.1. Verify the given solutions of (I), (II), and (III). Give a 2-by-2 symmetric pencil with only one eigenvector.

15.2.2. Find a singular 2-by-2 pencil, not symmetric, in which \mathbf{A} and \mathbf{M} do not share a common null space.

15.2.3. Note that

$$\begin{bmatrix} \rho & 1 & \mu & 0 \\ 0 & \rho & 0 & \mu \\ -\mu & 0 & \rho & 1 \\ 0 & -\mu & 0 & \rho \end{bmatrix} = \begin{bmatrix} 0 & \mu & 1 & \rho \\ \mu & 0 & \rho & 0 \\ 1 & \rho & 0 & -\mu \\ \rho & 0 & -\mu & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Let $\mathbf{B} = \mathbf{F}\mathbf{J}_r\mathbf{F}^{-1}$ where \mathbf{J}_r is the real Jordan form. Show that \mathbf{J}_r can be written as $\mathbf{J}_r = \tilde{\mathbf{A}}\tilde{\mathbf{M}}^{-1}$. Then conclude that $\mathbf{B} = \mathbf{AM}^{-1}$ where $\mathbf{A} = \tilde{\mathbf{A}}\tilde{\mathbf{F}}^*$, $\mathbf{M} = \tilde{\mathbf{F}}\tilde{\mathbf{M}}\tilde{\mathbf{F}}^*$.

15.2.4. Show that $\chi(\lambda, \mu) = \chi(1/\lambda, 1/\mu)$. Let $\lambda = \alpha_1/\beta_1$, $\mu = \alpha_2/\beta_2$, and express $\chi(\lambda, \mu)$ in terms of the α 's and β 's. Note that χ is invariant under orthogonal transformations of the form

$$\begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

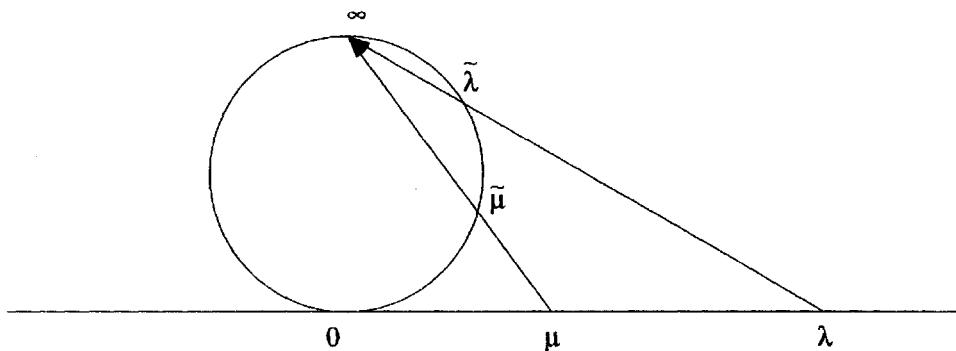
15.2.5. Show that $\chi(\lambda, \mu)$, the chordal metric, is the length of the chord joining $\tilde{\lambda}$ and $\tilde{\mu}$ in Figure 15.1.

15.3. Simultaneous Diagonalization of Two Quadratic Forms

In exact arithmetic all three phenomena revealed in section 15.2 can be banished by confining attention to those cases in which either \mathbf{A} or \mathbf{M} or some combination $\alpha\mathbf{A} + \mu\mathbf{M}$ is positive definite. This restriction seems natural when one notes that the standard eigenvalue problem corresponds to $\mathbf{M} = \mathbf{I}$, and \mathbf{I} is the prototype of all positive definite matrices.

In fact when given the pencil (\mathbf{A}, \mathbf{M}) there is no loss in considering the equivalent pencil $(\hat{\mathbf{A}}, \hat{\mathbf{M}}) = (\gamma\mathbf{A} + \sigma\mathbf{M}, -\sigma\mathbf{A} + \gamma\mathbf{M})$ for any pair (γ, σ) such that $\gamma^2 + \sigma^2 = 1$. What is the best pair to choose? To answer this consider

$$\mu(\mathbf{A}, \mathbf{M}) \equiv \inf_{\|\mathbf{x}\|=1} \{(\mathbf{x}^*\mathbf{A}\mathbf{x})^2 + (\mathbf{x}^*\mathbf{M}\mathbf{x})^2\}^{1/2}. \quad (15.6)$$

FIG. 15.1. *The chordal metric.*

The following interesting result is the subject of [Uhlig, 1979].

Theorem 15.3.1. *If $\mu(A, M) > 0$ then there is a pair (γ, σ) such that $\hat{M} = \gamma M - \sigma A$ is positive definite provided $n > 2$.*

Even when M is positive definite it might be preferable to work with (\hat{A}, \hat{M}) if $\lambda_1(\hat{M}) \gg \lambda_1(M)$. Although it is not discussed in the literature the most attractive choice of (γ, σ) in the definite case is the one that minimizes the condition number of \hat{M} ; $\min \lambda_{-1}(\hat{M})/\lambda_1(\hat{M})$. See [Parlett, 1991].

At present there is no economical way of finding the best pair and exploiting Theorem 15.3.1. Pencils with $\mu > 0$ are called *definite* and in theory the remainder of this chapter applies to definite pencils, but we shall confine ourselves to those in which definiteness is explicit.

It is the definiteness of the pencil (A, M) which guarantees the simultaneous reduction to diagonal form.

Theorem 15.3.2. *If M is positive definite then there are many invertible matrices F such that F^*AF and F^*MF are both diagonal and real.*

Proof. By the spectral theorem and the hypothesis,

$$\mathbf{M} = \mathbf{G}\Delta^2\mathbf{G}^*$$

where \mathbf{G} is orthonormal and $\Delta = \text{diag}(\delta_1, \dots, \delta_n)$ is real. The only, but crucial, use of positive definiteness is that \mathbf{M} 's eigenvalues are positive and allow the definition of a real Δ . Now reduce the pencil to standard form as follows:

$$\mathbf{A} \rightarrow \Delta^{-1}\mathbf{G}^*\mathbf{A}\mathbf{G}\Delta^{-1} = \mathbf{H}, \quad \mathbf{M} \rightarrow \Delta^{-1}\mathbf{G}^*\mathbf{M}\mathbf{G}\Delta^{-1} = \mathbf{I}.$$

By the spectral theorem for \mathbf{H} there is an orthonormal \mathbf{P} so that

$$\mathbf{H} = \mathbf{P}\Lambda\mathbf{P}^*$$

and so, with $\mathbf{F} \equiv \mathbf{G}\Delta^{-1}\mathbf{P}$, a product of invertible matrices,

$$\mathbf{A} \rightarrow \mathbf{F}^*\mathbf{A}\mathbf{F} = \Lambda, \quad \mathbf{M} \rightarrow \mathbf{F}^*\mathbf{M}\mathbf{F} = \mathbf{I}.$$

For any nonsingular diagonal matrix Ω , reduction by $\mathbf{F}\Omega$ also yields a diagonal pencil. \square

The proof suggests a simple way to compute the roots of (\mathbf{A}, \mathbf{M}) but that topic is postponed until the next section. Another form of Theorem 15.3.2 is Theorem 15.3.3.

Theorem 15.3.3. *If \mathbf{M} is positive definite then the symmetric pencil (\mathbf{A}, \mathbf{M}) has n real roots $\lambda_1, \lambda_2, \dots, \lambda_n$ in the interval $[-\|\mathbf{M}^{-1}\mathbf{A}\|, \|\mathbf{M}^{-1}\mathbf{A}\|]$ and, to match them, n linearly independent eigenvectors $\mathbf{z}_1, \dots, \mathbf{z}_n$. Moreover \mathbf{z}_i and \mathbf{z}_j are \mathbf{M} -orthogonal if $\lambda_i \neq \lambda_j$, i.e.,*

$$\mathbf{z}_i^*\mathbf{M}\mathbf{z}_j = 0.$$

If $\lambda_i = \lambda_j$ then \mathbf{z}_i and \mathbf{z}_j may be chosen to be \mathbf{M} -orthogonal.

The proof is left as Exercise 15.3.3.

Another bonus from the positive definiteness of \mathbf{M} is that both bilinear forms

$$(\mathbf{x}, \mathbf{y})_{\mathbf{M}} \equiv \mathbf{y}^*\mathbf{M}\mathbf{x}, \quad (\mathbf{x}, \mathbf{y})_{\mathbf{M}^{-1}} \equiv \mathbf{y}^*\mathbf{M}^{-1}\mathbf{x}, \quad (15.7)$$

are genuine *inner product* functions. The vector space \mathbb{R}^n supplemented with either inner product becomes a genuine inner product space \mathcal{M}^n and almost all properties of \mathcal{E}^n , such as the Cauchy–Schwarz inequality, carry over to \mathcal{M}^n . Exercises 15.3.3 and 15.3.4 explore some of this material.

A pencil is positive definite only if both \mathbf{A} and \mathbf{M} are positive definite.

Exercises on Section 15.3

15.3.1. Prove Theorem 15.3.3 by imitating the proofs of analogous results for the standard problem.

15.3.2. Show that (15.7) defines an inner product by verifying the following:

1. $(\mathbf{x}, \mathbf{y})_{\mathbf{M}} = (\mathbf{y}, \mathbf{x})_{\mathbf{M}}$,
2. $(\alpha\mathbf{x} + \beta\mathbf{y}, \mathbf{z})_{\mathbf{M}} = \alpha(\mathbf{x}, \mathbf{z})_{\mathbf{M}} + \beta(\mathbf{y}, \mathbf{z})_{\mathbf{M}}$,
3. $(\mathbf{x}, \mathbf{x})_{\mathbf{M}} > 0$ if $\mathbf{x} \neq \mathbf{o}$.

15.3.3. Verify that $\|\mathbf{u}\|_{\mathbf{M}} \equiv \sqrt{\mathbf{u}^* \mathbf{M} \mathbf{u}}$ satisfies the axioms for a norm:

1. $\|\mathbf{u}\|_{\mathbf{M}} > 0$ if $\mathbf{u} \neq \mathbf{o}$,
2. $\|\alpha\mathbf{u}\|_{\mathbf{M}} = |\alpha| \|\mathbf{u}\|_{\mathbf{M}}$,
3. $\|\mathbf{u} + \mathbf{v}\|_{\mathbf{M}} \leq \|\mathbf{u}\|_{\mathbf{M}} + \|\mathbf{v}\|_{\mathbf{M}}$.

For which properties is it essential that \mathbf{M} be positive definite?

15.3.4. Prove the Cauchy–Schwarz inequality

$$(\mathbf{x}, \mathbf{y})_{\mathbf{M}}^2 \leq (\mathbf{x}, \mathbf{x})_{\mathbf{M}} (\mathbf{y}, \mathbf{y})_{\mathbf{M}}$$

by considering $(\mathbf{x} + \xi\mathbf{y}, \mathbf{x} + \xi\mathbf{y})_{\mathbf{M}}$ for all real ξ .

15.3.5. Find θ such that $(\cos \theta)[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}] - (\sin \theta)[\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}]$ is positive definite. What is the best choice for θ ?

15.4. Explicit Reduction to Standard Form

Thanks to the Handbook and EISPACK (see section 2.8) high-quality programs are available for small standard eigenvalue problems. So there is much to be said for reducing the general problem $(\mathbf{A} - \lambda\mathbf{M})\mathbf{x} = \mathbf{o}$ to an equivalent standard form $(\mathbf{A} - \lambda\mathbf{I})\mathbf{y} = \mathbf{o}$. There are several ways to do it.

1. Form $\mathbf{M}^{-1}\mathbf{A}$ and lose both symmetry and sparsity.

2. Solve the standard eigenvalue problem for M to get orthonormal G and diagonal Δ such that $M = G\Delta^2G^*$. Then

$$A - \lambda M = G\Delta(\hat{A} - \lambda I)\Delta G^*, \quad (15.8)$$

where

$$\hat{A} = \Delta^{-1}G^*AG\Delta^{-1}. \quad (15.9)$$

3. Compute the Cholesky decomposition $M = LL^*$. Then

$$A - \lambda M = L(\overset{o}{\hat{A}} - \lambda I)L^* \quad (15.10)$$

and

$$\overset{o}{\hat{A}} = L^{-1}AL^*. \quad (15.11)$$

See Contribution II/10 in the Handbook or the EISPACK guide for more details.

15.4.1. Remarks

1. If eigenvectors are not wanted then the transform matrices L or G need not be preserved. If eigenvectors are wanted then either the transformations or the original pairs must be saved.
2. Some of the transformations on A and M can be done simultaneously, for example, when G and L are expressed as products of simple matrices (Exercise 15.4.1).
3. An extension and refinement of method 2 is described in section 15.5.
4. Wilkinson gives an elegant way to compute $\overset{o}{\hat{A}}$. Initially n -by- n arrays hold A and M . Copy the diagonals of A and M into one-dimensional arrays DA and DM . Then compute L and store its lower-triangular part in the lower triangle of M . Finally compute $(L^{-1}A)L^{-*}$ in two stages by using only the lower triangle of A which, on completion, holds the lower-triangular part of $\overset{o}{\hat{A}}$. The details are left as Exercise 15.4.2.
5. In order to simplify discussion assume that both A and M are positive definite and that the eigenvalues of the pair (A, M) are ordered by

$$0 < \lambda_1 \leq \cdots \leq \lambda_n.$$

Note that

$$\|\overset{o}{A}\| = \lambda_n \leq \|M^{-1}\| \|A\|.$$

In many cases some elements of $\overset{o}{A}$ are of the same magnitude as $\|M\|^{-1} \|A\|$ and much greater than $\|A\|$. This happens when M is ill conditioned for inversion (nearly singular) and then the eigenvalues of $\overset{o}{A}$ are spread over many orders of magnitude, for example, from 10^3 to 10^{20} . It is likely that the small eigenvalues will be computed with far less relative accuracy than the large ones. This is because the computed eigenvalues are exact for a matrix \tilde{A} and often $\|\tilde{A} - \overset{o}{A}\| \doteq \varepsilon \|\overset{o}{A}\|$, where ε is the unit roundoff. A crude bound on the eigenvalue change is

$$\frac{|\tilde{\lambda}_i - \overset{o}{\lambda}_i|}{|\overset{o}{\lambda}_i|} \leq \frac{\|\tilde{A} - \overset{o}{A}\|}{|\overset{o}{\lambda}_i|} \doteq \varepsilon \left(\frac{\|\overset{o}{A}\|}{\overset{o}{\lambda}_i} \right), \quad i = 1, \dots, n.$$

For $i = 1, 2, 3$ it is quite possible to have $\|\overset{o}{A}\| / \overset{o}{\lambda}_1 \doteq 1/\varepsilon$ and hence the possibility that $\overset{o}{\lambda}_1$ will have no correct figures.

This is the flaw in explicit reduction when M is nearly singular and the arithmetic is done with numbers of limited precision. It is serious because in many cases the original pair A, M does determine the small eigenvalues to high relative accuracy. See below.

6. If M , and therefore L , is moderately ill conditioned for inversion and if, in addition, the columns of L decrease monotonically then $\overset{o}{A}$ will usually be a graded matrix whose rows increase in norm (last row is biggest). Graded matrices usually define their smallest eigenvalues with greater relative accuracy than could be expected from standard norm estimates (absolute error $< \varepsilon \|\overset{o}{A}\|$). An example is given in Figure 15.2.

15.4.2. Reduction of Banded Pencils

Section 7.5 presented an ingenious algorithm for reducing a banded matrix A to tridiagonal form T without temporarily enlarging the bandwidth in the process. In [Crawford, 1973] this approach is extended to transform banded pencils (A, M) into (T, I) without the use of extra storage.

We shall not describe the algorithm in detail for the following reasons. When A and M are large it is unlikely that all, or even a majority, of the eigenvalues will be wanted, and methods which avoid explicit reduction are

$$\begin{aligned}
 A &= \begin{bmatrix} 4 & 1 & 0 & 1 \\ 1 & 4 & 2 & 0 \\ 0 & 2 & 4 & 2 \\ 1 & 0 & 2 & 3 \end{bmatrix} & M &= \begin{bmatrix} 10000 & 4000 & 1000 & 100 \\ 4000 & 1700 & 430 & 60 \\ 1000 & 430 & 110 & 16.5 \\ 100 & 60 & 16.5 & 5.26 \end{bmatrix} \\
 L &= \begin{bmatrix} 100 & & & \\ 40 & 10 & & \\ 10 & 3 & 1 & \\ 1 & 2 & .5 & .1 \end{bmatrix} \\
 \overset{o}{A} &= \begin{bmatrix} .0004 & -.0006 & -.0002 & .119 \\ -.0006 & -.0384 & .0908 & -1.616 \\ -.0022 & .0908 & 3.150 & 2.658 \\ .119 & -1.616 & 2.658 & 223.8 \end{bmatrix}
 \end{aligned}$$

FIG. 15.2. Reduction of (A, M) to $(\overset{o}{A}, I)$.

preferable. Moreover large A and M of narrow bandwidth are ideally suited to spectrum slicing (section 3.3) or the Lanczos algorithm (section 15.11).

Exercises on Section 15.4

- 15.4.1. Find the operation counts for the spectral decomposition and the Cholesky factorization of M . Exploit symmetry. Give op counts for reductions (15.8), (15.9) and (15.10), (15.11).
- 15.4.2. Let $B = L^{-1}A$. Find an algorithm which computes and writes the lower part of BL^{-*} over the lower part of A , losing A 's diagonal in the process. Exploit symmetry where possible.
- 15.4.3. Consider a 2-by-2 pencil. If M is ill conditioned and if $\overset{o}{A}$ is ill conditioned, then $\overset{o}{A}$ must be strongly graded. True or false?

*15.5. The Fix–Heiberger Reduction

This section describes a careful reduction to standard form designed specifically to cope with those cases in which M is, for practical purposes, positive semidefinite (at least one eigenvalue is zero). In other words M is either singular or so close to a singular matrix that it is preferable to work with the latter

and face up to the consequences. The aim is to find any infinite or ill-disposed eigenvalues (see section 15.2) before computing the respectable ones.

In what follows the notation

$$A, M \xrightarrow{P} \tilde{A}, \tilde{M}$$

means

$$\tilde{A} = P^* A P, \quad \tilde{M} = P^* M P.$$

Also $P \oplus Q$ stands for $\text{diag}(P, Q)$. Let η be a user-given criterion for negligibility.

There are three steps of increasing complexity.

Step 1. Find the spectral decomposition of M , $M = G(\Delta_1^2 \oplus \Delta_2^2)G^*$ where Δ_2^2 has the tiny eigenvalues of M , $\|\Delta_2\|^2 < \eta\|\Delta_1\|^2$. Replace Δ_2^2 by O . Partition G^*AG to match Δ_1 to find

$$A, M \xrightarrow{G} \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{12}^* & \tilde{A}_{22} \end{bmatrix}, \quad \begin{bmatrix} \Delta_1^2 & O \\ O & O \end{bmatrix}.$$

Step 2. Reduce Δ_1^2 to I and find the spectral decomposition of \tilde{A}_{22} , $\tilde{A}_{22} = F(\Phi \oplus \Psi)F^*$ where $\Phi \oplus \Psi$ is diagonal and $\|\Psi\| < \eta\|\Phi\|$. Put $\Psi = O$. Write $\Delta_1 \tilde{A}_{12} F$ as $(\overset{\circ}{A}_{12}, \overset{\circ}{A}_{13})$ to find

$$(\cdot, \cdot) \xrightarrow{\Delta_1^{-1} \oplus F} \begin{bmatrix} \overset{\circ}{A}_{11} & \overset{\circ}{A}_{12} & \overset{\circ}{A}_{13} \\ \overset{\circ}{A}_{12}^* & \Phi & O \\ \overset{\circ}{A}_{13} & O & O \end{bmatrix}, \quad \begin{bmatrix} I & O & O \\ O & O & O \\ O & O & O \end{bmatrix}.$$

When M is well conditioned the last two rows and columns will be empty and the result is $(\overset{\circ}{A}_{11}, I)$ which is reduction 2 given in section 15.4. If \tilde{A}_{22} is ill conditioned then the final form is

$$\begin{bmatrix} \overset{\circ}{A}_{11} & \overset{\circ}{A}_{12} \\ \overset{\circ}{A}_{12}^* & \Phi \end{bmatrix}, \quad \begin{bmatrix} I & O \\ O & O \end{bmatrix}.$$

The further reduction of this pencil to obtain the finite eigenvalues is left as Exercise 15.5.1.

In the general case $\overset{\circ}{A}_{13}$ needs further analysis.

Step 3. Trouble comes when $\overset{\circ}{A}_{13}$ does not have full rank, but this is one of the cases when Fix–Heiberger is warranted and so we allow for this possibility. At this point it is necessary to compute the singular value decomposition of $\overset{\circ}{A}_{13}$, namely,

$$\overset{\circ}{A}_{13} = Q \begin{bmatrix} \Sigma \\ O \end{bmatrix} P^*,$$

where Q and P are orthonormal (of different sizes) and Σ is diagonal with non-negative elements. The matrices Q and P are not unique, Q is an eigenvector matrix for $\overset{o}{A}_{13}\overset{o}{A}_{13}^*$, while P is an eigenvector matrix for $\overset{o}{A}_{13}^*\overset{o}{A}_{13}$. The diagonal elements σ_i of Σ are given by

$$\sigma_i = \sqrt{\lambda_{-i} \begin{bmatrix} \overset{o}{A}_{13} & \overset{o}{A}_{13} \\ \overset{o}{A}_{13}^* & \overset{o}{A}_{13} \end{bmatrix}}$$

and are called the singular values of $\overset{o}{A}_{13}$.

Next Σ is divided into $\Theta \oplus \Omega$ where Θ has the larger singular values and $\|\Omega\| \leq \eta \|\Theta\|$. Then put $\Omega = 0$ and define

$$\begin{aligned} n_1 &\equiv \text{the number of rows in } \Theta, \\ n_1 + n_2 &\equiv \text{the number of rows in } \overset{o}{A}_{13}, \\ n_3 &\equiv \text{the number of rows in } \Phi. \end{aligned}$$

The submatrices in the Fix-Heiberger form are given by

$$n_1 \quad n_2$$

$$Q^* \overset{o}{A}_{11} Q = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix}, \quad Q^* \overset{o}{A}_{12} = \begin{bmatrix} A_{13} \\ A_{23} \end{bmatrix}.$$

Then finally

$$(\dots) \xrightarrow{Q \oplus I \oplus P} \begin{bmatrix} A_{11} & A_{12} & A_{13} & \Theta & 0 \\ A_{22} & A_{23} & 0 & 0 \\ & \Phi & 0 & 0 \\ \text{sym} & & 0 & 0 \\ & & & 0 \end{bmatrix}, \quad I \oplus I \oplus O \oplus O \oplus O.$$

If $\Theta \neq \Sigma$ then the fifth (block) row and column of O 's is actually present and the pencil (A, M) is singular. A warning should be given that any number is an eigenvalue. The fifth row and column may then be dropped and this deflates the troublesome null space. The genuine eigenvalues and eigenvectors can be found from the remaining four rows and columns as in the case when $\overset{o}{A}_{13}$ has full rank in step 2.

Let $(x_1^*, x_2^*, x_3^*, x_4^*)$ be a partitioned eigenvector of the 4-by-4 leading submatrix of the canonical form shown above. Solving these equations from the bottom row up yields

$$\begin{aligned}x_1 &= 0, \\x_3 &= -\Phi^{-1}A_{23}^*x_2, \\(A_{22} - A_{23}\Phi^{-1}A_{23}^* - \lambda)x_2 &= 0, \\x_4 &= -\Theta^{-1}(A_{12}x_2 + A_{13}x_3).\end{aligned}$$

Thus the n_2 finite eigenvalues come from the standard eigenvalue problem for x_2 in line 3. Derivation of the eigenvectors with infinite eigenvalue is left as Exercise 15.5.2. The formulas for the eigenvectors of the original problem are the subject of Exercise 15.5.3.

The purpose of the extra care exercised in the above reduction is the accurate determination of those small eigenvalues which happen to be well determined by the original data. If a suitable value of η is not apparent the calculation should be made with two different values of η (say $\eta = n\varepsilon$ and $\eta = \sqrt{\varepsilon}$) and the results compared.

In [Fix and Heiberger, 1972] bounds are given on the error caused by annihilating Δ_2 , Ψ , and Ω .

Exercises on Section 15.5

- 15.5.1. Find a standard eigenvalue problem which gives the finite eigenvalues of

$$\begin{bmatrix} \overset{\circ}{A}_{11} & \overset{\circ}{A}_{12} \\ \overset{\circ}{A}_{12}^* & \Phi \end{bmatrix}, \quad \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}.$$

Recall that Φ is diagonal and invertible.

- 15.5.2. Describe the eigenvectors of the Fix–Heiberger form which have infinite eigenvalues using the partition x_1, x_2, x_3 , and x_4 employed in section 15.5.
- 15.5.3. Give the formulas which turn the eigenvectors of the canonical form into those of original pencil assuming that $\overset{\circ}{A}_{13}$ has full rank.
- 15.5.4. At the end of step 1 we have

$$\begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 1 & 1 & 0 \\ 2 & 1 & 2 & 3 & 0 & 0 \\ 2 & 3 & 4 & 0 & 0 & 0 \\ 10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{\text{sym}}, \quad \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 10^{-2} & & \\ & & & 10^{-4} & \\ & & & & 0 \\ & & & & & 0 \end{bmatrix}.$$

Find the Fix–Heiberger form and, if a calculator is available, find all eigenvectors and eigenvalues.

15.6. The QZ Algorithm

A stable generalization of the QR algorithm to arbitrary square pencils (B, N) was presented in [Moler and Stewart, 1973]. The method destroys symmetry and changes (A, M) to the equivalent pencil (J, K) , where J and K are upper triangular. Orthogonal transformations are employed exclusively so that $\|A\| = \|J\|$, $\|M\| = \|K\|$, and the final accuracy is impervious to hazards such as M being singular or even indefinite. Ill-disposed eigenvalues are revealed as the quotients of small diagonal elements of J and K .

For small pencils QZ is an alternative to the Fix–Heiberger reduction and has the advantage of being generally available in computer center program libraries. Approximately $20n^3$ ops are required to find all the eigenvalues by QZ. The time penalty is less irksome than the need for two auxiliary n -by- n arrays. However, for small n neither penalty is too serious unless a very large number of pencils are to be processed.

The loss of symmetry is displeasing and we turn next to a method which employs congruencies rather than equivalence transformations.

15.7. Jacobi Generalized

The natural extension of Jacobi's idea (Chapter 9) is to find a congruence in the (i, j) plane to annihilate the (i, j) elements of both A and M . If M is positive definite this can always be done and there is considerable freedom in the choice of congruence.

The problem is essentially two dimensional. Restricting attention to the typical (i, j) plane and using the simplest congruencies yields

$$\begin{bmatrix} 1 & -\alpha \\ \beta & 1 \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} 1 & \beta \\ -\alpha & 1 \end{bmatrix} \quad (15.12)$$

$$= \begin{bmatrix} a_{ii} - 2\alpha a_{ij} + \alpha^2 a_{jj}, & \beta a_{ii} + (1 - \alpha\beta)a_{ij} - \alpha a_{jj} \\ \beta a_{ii} + (1 - \alpha\beta)a_{ij} - \alpha a_{jj}, & \beta^2 a_{ii} + 2\beta a_{ij} + a_{jj} \end{bmatrix} \quad (15.13)$$

and similarly for M . In order to diagonalize both matrices a special pair of quadratic equations must be solved for α and β . Fortunately there is a closed form solution (see Exercise 15.7.1); the nonlinear term can be eliminated to reveal $\alpha = \delta_i/\nu$, $\beta = \delta_j/\nu$, where

$$\delta_i \equiv \det \begin{bmatrix} a_{ii} & a_{ij} \\ m_{ii} & m_{ij} \end{bmatrix}, \quad \delta_j \equiv \det \begin{bmatrix} a_{jj} & a_{ij} \\ m_{jj} & m_{ij} \end{bmatrix}, \quad (15.14)$$

and ν satisfies the quadratic equation

$$\nu^2 - \delta_{ij}\nu - \delta_i\delta_j = 0, \quad (15.15)$$

$$\delta_{ij} \equiv \det \begin{bmatrix} a_{ii} & a_{jj} \\ m_{ii} & m_{jj} \end{bmatrix}. \quad (15.16)$$

When \mathbf{M} is positive definite, then there is a nonzero solution to the quadratic (Exercise 15.7.2), and the congruence is proper, i.e., the determinant $1 + \alpha\beta \neq 0$. To keep α and β small it is essential to choose as ν the root of (15.15) which is farthest from 0.

The asymptotic quadratic convergence of the regular Jacobi process extends to this case [Zimmerman, 1969] provided that the process does converge. However, convergence has not been proven.

These techniques have been used with some success on small \mathbf{A} and \mathbf{M} which are diagonally dominant, that is $a_{ii} > \sum_{j=1}^n |a_{ij}|$, $j \neq i$, for each i . See [Bathé and Wilson, 1976].

Exercises on Section 15.7

15.7.1. Find a closed form solution (α, β) to

$$\beta a_{ii} + (1 + \alpha\beta)a_{ij} - \alpha a_{jj} = 0,$$

$$\beta m_{ii} + (1 + \alpha\beta)m_{ij} - \alpha m_{jj} = 0.$$

15.7.2. Show that (15.15) has a nonzero solution when \mathbf{M} is positive definite.

15.7.3. Show that with the proper choice of ν in (15.15) the parameters α and β satisfy $0 \leq \alpha\beta \leq 1$.

15.8. Implicit Reduction to Standard Form

The three reductions described in section 15.4 can be used implicitly. This observation is important in the treatment of large sparse \mathbf{A} and \mathbf{M} . Let us reconsider the three techniques.

1. The matrix $\mathbf{M}^{-1}\mathbf{A}$ is neither symmetric nor sparse but it is self-adjoint with respect to the \mathbf{M} inner product;

$$(\mathbf{M}^{-1}\mathbf{A}\mathbf{x}, \mathbf{y})_{\mathbf{M}} = (\mathbf{x}, \mathbf{M}^{-1}\mathbf{A}\mathbf{y})_{\mathbf{M}} = \mathbf{y}^*\mathbf{A}\mathbf{x}.$$

The vector $\mathbf{w} = \mathbf{M}^{-1}\mathbf{A}\mathbf{u}$ is computed in two steps.

- (a) Form $\mathbf{v} = \mathbf{A}\mathbf{u}$, exploiting sparsity.
 - (b) Either solve $\mathbf{M}\mathbf{w} = \mathbf{v}$ iteratively, if \mathbf{M} cannot be factored, or solve $\mathbf{L}\mathbf{x} = \mathbf{v}$; then $\mathbf{L}^*\mathbf{w} = \mathbf{x}$ if $\mathbf{M} = \mathbf{LL}^*$.

Each technique in (b) can exploit sparsity.

2. In principle the matrix $\hat{A} = \Delta^{-1}G^*AG\Delta^{-1}$ can be used without forming \hat{A} . However, the matrix G rarely inherits any sparse structure in M (except when $M = \Delta^2$ and $G = I$) and so this decomposition is not used implicitly to our knowledge.
3. The product $x = \overset{\circ}{\hat{A}}u$ where $\overset{\circ}{\hat{A}} = L^{-1}AL^{-*}$ can be formed in three steps:

Solve $L^*v = u$ for v ,
 Form $w = Av$,
 Solve $Lx = w$ for x .

Any band or profile structure in M is inherited by its Cholesky factor L . The sparse matrix L need not be inverted. If A and M have bandwidth $2m + 1$, $m \ll n$, then the method given above requires approximately $(4m + 3)n$ ops to form x as against n^3 using the full form $\overset{\circ}{\hat{A}}$. So both time and storage are saved by the implicit reduction when only a few eigenvalues and eigenvectors are required.

Figure 15.3 shows how L inherits A 's sparsity pattern.

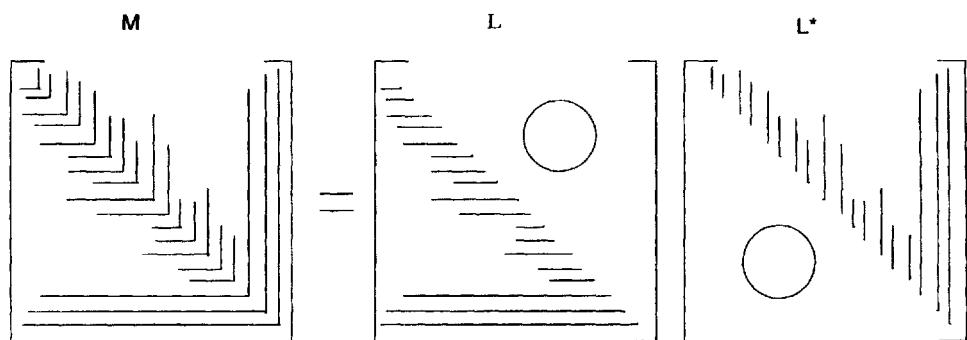


FIG. 15.3. *Preservation of profile in triangular factorization.*

15.9. Simple Vector Iterations

The previous section may have given the impression that if M can be factored into LL^* then there is nothing better to do than apply the most appropriate method to the standard problem $(\overset{\circ}{A}, I)$ with $\overset{\circ}{A}$ either in explicit or implicit form. This approach is certainly in the mathematical tradition of reducing a new problem to one previously solved. However, in the present context the standard reduction may not be warranted.

In order to examine this question we reconsider the power method and inverse iteration of Chapter 4. The governing equations of the two methods, with shift σ , are now

$$\text{PM: } Mv_{k+1} = (A - \sigma_k M)v_k \tau_k, \quad (15.17)$$

$$\text{INVIT: } (A - \sigma_k M)u_{k+1} = Mu_k \nu_k, \quad (15.18)$$

where τ_k and ν_k are normalizing constants. The new fact is that both iterations require the solution of a system of equations.

Case 1. If A and M are so large that factorization is infeasible then some inner iteration must be used to solve (15.17) or (15.18) and the two techniques are equal in cost—in the absence of special qualities in M .

Case 2. If A and M can be factored then we can compare two versions of INVIT, namely,

1. Factor $A - \sigma_k M$ in order to solve (15.18) for u_{k+1} .

2. Solve $(\overset{\circ}{A} - \sigma_k I)w_{k+1} = w_k \pi_k$ for w_{k+1} in three steps:

$$x_k = Lw_k, \quad (A - \sigma_k M)y_k = x_k, \quad w_{k+1} = (L^*y_k)\pi_k. \quad (15.19)$$

If u_{k+1} is normalized in the same way as w_{k+1} then the two versions have the same cost and the first version is certainly more natural than the second version.

The preceding remarks show that for large matrices there is merit in retaining the original form of the problem, namely,

$$(A - \lambda M)x = o.$$

1. The spectrum slicing technique and secant iterations of Chapter 3 extend directly and are well suited to those cases in which A and M have narrow

bandwidth and no eigenvectors are wanted. See [Bathé and Wilson, 1976, Chapter 11] for more details.

2. Most of the results of Chapter 4 extend to the pencil (A, M) provided that an appropriate norm is used in place of $\|\cdot\|$. Theoretically, the extension is fine but in practice the new norm increases the cost considerably. Perhaps we have been spoiled by the extreme simplicity of the Euclidean norm. The proofs of the results given below are close in spirit to the corresponding proofs in Chapter 4 and will be omitted. Some of them are quite straightforward; others are less so. Recall that $\|x\|_{M^{-1}} \equiv \sqrt{x^* M^{-1} x}$.

Theorem 15.9.1. *For arbitrary $u \neq 0$ and σ there is an eigenvalue λ of (A, M) such that*

$$|\lambda - \sigma| \leq \|(A - \sigma M)u\|_{M^{-1}} / \|Mu\|_{M^{-1}}.$$

As in the standard case an excellent choice for σ in INVIT is Rayleigh's quotient

$$\rho(u) = \rho(u; A, M) \equiv u^* A u / u^* M u, \quad u \neq 0.$$

Theorem 15.9.2. *When M is positive definite the Rayleigh quotient enjoys the following properties:*

Homogeneity: $\rho(\alpha u) = \rho(u)$, $\alpha \neq 0$ (degree 0).

Boundedness: $\rho(u)$ ranges over $[\lambda_1, \lambda_{-1}]$ as u ranges over the unit sphere.

Stationarity: $\text{grad } \rho(u) \equiv \nabla \rho(u) = 2(Au - \rho(u)Mu)^*/u^*Mu$. Thus ρ is stationary at, and only at, the eigenvectors of (A, M) .

Minimum Residual:

$\|(A - \sigma M)u\|_{M^{-1}}^2 \geq \|Au\|_{M^{-1}}^2 - |\rho(u)|^2 \|Mu\|_{M^{-1}}^2$ with equality when and only when $\sigma = \rho(u)$.

Numerical methods have been proposed to remedy the “defect” that $\rho(u)$ does not minimize $\|(A - \sigma M)u\|$ over all σ . This is rather like forgetting to change currency when visiting a foreign country.

As before we define the residual vector $r(u)$ by

$$Au = \rho(u)Mu + r(u)$$

and observe that this is again an orthogonal decomposition of Au in the M^{-1} inner product; $(r(u), Mu)_{M^{-1}} = 0$. Algebraically it happens that $r^*u = 0$ but that is not so significant in the present context.

The RQI produces the sequence $\{x_k\}$ using

$$(A - \rho_k M)x_{k+1} = Mx_k \tau_k, \quad (15.20)$$

where τ_k is chosen so that $\|Mx_k\|_{M^{-1}} = 1$ for all k and $\rho_k = \rho(k)$. The convergence theory of RQI is given in the following three theorems.

Theorem 15.9.3. Suppose that $\{x_k\} \rightarrow z$, an eigenvector, as $k \rightarrow \infty$. Let $\psi_k = \angle(Mx_k, Mz)$ in the M^{-1} inner product. Then

$$\lim_{k \rightarrow \infty} |\tan \psi_{k+1} / \tan^3 \psi_k| \leq 1.$$

Theorem 15.9.4.

$$\|(A - \rho_{k+1} M)x_{k+1}\|_{M^{-1}} \leq \|(A - \rho_k M)x_k\|_{M^{-1}} \text{ for all } k.$$

Theorem 15.9.5. *For any nonzero starting vector x_0*

1. $\rho_k \rightarrow \rho$, as $k \rightarrow \infty$, and either
2. $(\rho_k, x_k) \rightarrow (\lambda, z)$, an eigenpair, cubically, or
3. $\{x_{2k}\} \rightarrow x_+$, $\{x_{2k+1}\} \rightarrow x_-$, linearly, and x_\pm are bisectors of eigenvectors belonging to eigenvalues $\rho \pm \tau$,

$$\tau = \lim_k \|(\mathbf{A} - \rho_k \mathbf{M})x_k\|_{\mathbf{M}^{-1}}.$$

The regime in 3 is unstable in the face of perturbation.

15.9.1. Other Iterative Techniques

The power method and RQI are techniques for solving the homogeneous system of equations $(\mathbf{A} - \lambda \mathbf{M})z = 0$. Of course λ is unknown and so the problem is not linear. Nevertheless almost every known technique for solving linear systems yields an analogous iteration for the eigenvalue problem. For example, there is a successive overrelaxation (SOR) method which can be very effective for special problems when triangular factorization is not possible. The reader is referred to [Ruhe, 1975 and 1977] for a full treatment of these ideas. It is our belief that no iteration of the form $x_{k+1} = \phi_k(x_k)$ can compete with the Lanczos algorithm which discards no previous information.

For the same reason we have not described those methods which seek to minimize $\|(\mathbf{A} - \mu \mathbf{M})x\|$ by various gradient methods and have proved useful in the more general problem of minimizing nonlinear functions. In our case the Euclidean norm $\|\cdot\|$ is not natural to the problem, and so there is no reason why such iterations should converge rapidly nor any evidence that they do.

One of the simplest iterative techniques is coordinate overrelaxation. The idea is simply to minimize $(\rho(x + \alpha e_j); \mathbf{A}, \mathbf{M})$ over α for each coordinate vector e_j in turn. Here x is the current approximate eigenvector whose j th element is to be changed. In practice it turns out advantageous to overrelax and replace x by $x + \omega \alpha e_j$ for some ω in $(1, 2)$. Each step requires two matrix–vector multiplications and, inevitably, the convergence properties are not very satisfactory.

The usual argument in favor of these simple iterative techniques is that they require no factorization and consequently are the *only* methods available

for those problems in which A and M are so huge that triangular factorization is supposedly not possible. This sentiment flouts the old adage, "Where there's a will there's a way." The example of the structural engineers suggests that there is no limit on the size of matrices which can be factored. Of course secondary storage must be used heavily when $n > 5000$ and the proper control of transfers between the various storage hierarchies is neither easy nor within the scope of this book.

Even when factorization is rejected the equation $Mx = b$ can be solved, either by iteration or by the conjugate gradient technique, and so the Lanczos algorithm and subspace iteration can be used as described in the remaining sections.

Exercises on Section 15.9

15.9.1. Prove Theorem 15.9.1.

15.9.2. Prove Theorem 15.9.2.

15.9.3. Do an operation count for one step of RQI assuming that A and M have half-bandwidth m . Assume that $A - \rho_k M$ permits triangular factorization and use the Table 3.1 for relevant operation counts.

15.9.4. Prove Theorem 15.9.3.

15.9.5. Prove Theorem 15.9.4.

15.10. RR Approximations

Let S be an n -by- m matrix of full rank $m \leq n$. We want to find formulas for those linear combinations of S 's columns which are collectively the best approximations from $\text{span}S$. Our criterion for "best" will be imported directly from the standard problem discussed in Chapter 11. This is somewhat unsatisfactory as regards an independent development of the theory of matrix pencils, but it does avoid a rather complicated discussion of the proper geometric setting for (A, M) .

Every symmetric positive definite matrix has a unique positive definite square root and, for theoretical purposes, it is convenient to examine the reduction of (A, M) to standard form by means of M 's square root $M^{\frac{1}{2}}$. The given basic equation

$$Az - Mz\lambda = o \quad (15.21)$$

is rewritten as

$$(M^{-\frac{1}{2}}AM^{-\frac{1}{2}} - \lambda)M^{\frac{1}{2}}z = M^{-\frac{1}{2}}o = o. \quad (15.22)$$

Consequently the objects of interest are $\hat{A} \equiv M^{-\frac{1}{2}}AM^{-\frac{1}{2}}$, $\hat{z} = M^{\frac{1}{2}}z$, and the modified trial vectors contained in the columns of $M^{\frac{1}{2}}S$.

For the reduced problem we can invoke the RR approximations discussed in section 11.4. The approximations are easiest to describe when the basis is orthonormal, i.e., when $(S^*M^{\frac{1}{2}})(M^{\frac{1}{2}}S) = I_m$. For theoretical work the easiest way to normalize S is to use

$$\hat{S} = M^{\frac{1}{2}}S(S^*MS)^{-\frac{1}{2}}. \quad (15.23)$$

Thus $\hat{S}^*\hat{S} = I_m$ and the classical RR theory says that

$$\min \|\hat{A}\hat{S} - \hat{S}\hat{H}\|_F, \quad \text{over } m\text{-by-}m \hat{H}, \quad (15.24)$$

is achieved uniquely by

$$\hat{H} \equiv \rho(\hat{S}) \equiv \hat{S}^*\hat{A}\hat{S}. \quad (15.25)$$

Note the use of $\|\cdot\|_F$ instead of $\|\cdot\|$. Furthermore the best approximations to eigenvectors of \hat{A} from $\text{span } \hat{S}$ are $\hat{S}\hat{g}_i$, $i = 1, \dots, m$, where $\hat{H}\hat{g}_i = \hat{g}_i\theta_i$ and $\|\hat{g}_i\| = 1$. In brief, the Ritz approximations are $\hat{S}\hat{G}$ where $\hat{G}\Theta\hat{G}^*$ is the spectral factorization of \hat{H} .

Now let us translate this characterization into expressions involving the original matrices A and M . From (15.25) and (15.23)

$$\hat{H} = (S^*MS)^{-\frac{1}{2}}(S^*AS)(S^*MS)^{-\frac{1}{2}}. \quad (15.26)$$

Fortunately \hat{H} is not needed explicitly; its eigenpairs (θ_i, \hat{g}_i) yield the Ritz approximations. From (15.26) we derive

$$(S^*AS - \theta_i S^*MS)(S^*MS)^{-\frac{1}{2}}\hat{g}_i = 0. \quad (15.27)$$

This equation is usually written as

$$(A_S - \theta_i M_S)g_i = 0, \quad i = 1, \dots, m. \quad (15.28)$$

The best approximations to \hat{z}_i is $\hat{S}\hat{g}_i$ and so, by (15.22), the best approximations to the original z 's are the vectors

$$M^{-\frac{1}{2}}\hat{S}\hat{g}_i = S(S^*MS)^{-\frac{1}{2}}\hat{g}_i = Sg_i, \quad i = 1, \dots, m. \quad (15.29)$$

To summarize,

The RR approximations to the pencils (A, M) from $\text{span } S$ are the pairs (θ_i, Sg_i) , $i = 1, \dots, m$, where (θ_i, g_i) satisfy (15.28). They are optimal in the sense of minimizing

$$\|M^{-\frac{1}{2}}R\|_F^2 = \text{trace}(R^*M^{-1}R)$$

over all the residual matrices R where $r_i = Ax_i - Mx_i\mu_i$, $i = 1, \dots, m$, and $x_i^*Mx_j = \delta_{ij}$ (Kronecker symbol), $x_i \in \text{span } S$.

Exercises on Section 15.10

- 15.10.1. Let $\langle \cdot, \cdot \rangle$ be an inner product function on \mathbb{R}^n . Find the scalar μ which minimizes, for given x , $\langle Ax - Mx\mu, Ax - Mx\mu \rangle$.
- 15.10.2. Derive (15.27).

15.11. Lanczos Algorithms

Familiarity with the contents of Chapter 13 is strongly recommended.

Given A and M (positive definite) and M 's Cholesky factor L then the Lanczos algorithm, with or without SO, can be applied to the implicitly defined matrix $\tilde{A} = L^{-1}AL^{-*}$ as described in section 15.7. The user supplied matrix–vector product program OP must incorporate L but the actual Lanczos program need not be modified at all. At the end each computed eigenvector y must be mapped back to an eigenvector z of (A, M) by solving $L^*z = y$ for z . This usage is very satisfactory and standard working practice when M has narrow bandwidth.

On the other hand it is possible to use Lanczos on $M^{-1}A$, again defined implicitly, and this option is valuable when M cannot be factored conveniently. In this case the algorithm must be reformulated to take account of M and so the algorithms of Chapter 13 must be modified a little.

Mathematically the extension is immediate. The basic three-term recurrence becomes

$$Mq_{j+1}\beta_{j+1} = Aq_j - Mq_j\alpha_j - Mq_{j-1}\beta_j.$$

The indices of the β 's have been shifted up by one. The Lanczos vectors q_1, q_2, \dots are not mutually orthonormal but they are M -orthonormal, i.e., $q_j^*Mq_j = \delta_{ij}$. The tridiagonal matrix T_j is the same as before and if the algorithm is continued for n steps it produces an invertible matrix Q which

reduces (A, M) to (T, I) . By the end of step j the algorithm has produced $Q_j = (q_1, \dots, q_j)$ and T_j , the leading principal j -by- j submatrix of T . If (θ, s) is an eigenpair of T_j then $(\theta, Q_j s)$ is the corresponding Ritz pair for the pencil (A, M) (see section 15.10).

For theoretical and computational purposes it is helpful to introduce the auxiliary vectors p_1, p_2, \dots defined by $p_i = M q_i$. Note that the sequences $\{q_i\}$ and $\{p_i\}$ are biorthonormal in \mathcal{E}^n , i.e., $p_k^* q_j = \delta_{kj}$. SO generalizes in a straightforward manner. The simple algorithm is recovered by omitting the first step of the algorithm given below.

15.11.1. Selective Lanczos Algorithm for (A, M)

Pick $u_1 \neq 0$. Compute $r_1 \leftarrow Mu_1$, $\beta_1 \leftarrow \sqrt{u_1^* r_1} > 0$. For $j = 1, 2, \dots$ repeat steps 1 through 9:

1. If there are any threshold vectors, then purge u_j of threshold vectors, and recompute $r_j \leftarrow Mu_j$, $\beta_j \leftarrow \sqrt{u_j^* r_j}$.
2. $q_j \leftarrow u_j / \beta_j$.
3. $\bar{u}_j \leftarrow Aq_j - p_{j-1}\beta_j$ ($p_0 = 0$).
4. $\alpha_j \leftarrow q_j^* \bar{u}_j$.
5. $p_j \leftarrow r_j / \beta_j$.
6. $r_{j+1} \leftarrow \bar{u}_j - p_j \alpha_j$.
7. Solve $Mu_{j+1} = r_{j+1}$ for u_{j+1} ($= q_{j+1}\beta_{j+1}$).
8. $\beta_{j+1} \leftarrow \sqrt{u_{j+1}^* r_{j+1}}$.
9. Compute eigenpairs (θ_i, s_i) of T_j as needed. Test for threshold vectors and for a pause. Test for convergence. If satisfied then stop.

15.11.2. Remarks

1. It has been said that Lanczos cannot be used unless M can be factored. However, iterative methods in step 7 can make maximal use of sparsity, and a good starting vector is $[\text{diag}(M)]^{-1}r_{j+1}$.
2. Steps 5 and 7 are additions to the algorithms of Chapter 13. Step 5 increases the vector operation count from five to six. Only four n -vectors of storage are needed; u, r, p, q , an increase of one.

3. q_j can be put into sequential storage after step 4. There is no need to save p_j , although the old p_i are useful in step 1.
4. The quantities $\beta_{ji} = \beta_j |e_j^* s_i|$ still measure convergence but in the new norm. See Exercise 15.11.2.
5. The algorithm given above can be used to advantage even when M can be factored as indicated below.

15.11.3. How to Use a Lanczos Program

A task which occurs frequently is to compute the eigenvectors belonging to all the eigenvalues in an interval $[\sigma - \omega, \sigma + \omega]$. Both A and M will be large and sparse but one or two triangular factorizations may be feasible. In such a case it will be advantageous to apply the Lanczos algorithm to the pair $(M, A - \sigma M)$ instead of (A, M) ; see [Ruhe and Ericsson, 1980].

If the computation of Aq or the solution of $Mu = r$ involves heavy use of secondary storage then there may be no extra cost in computing AQ or solving $MU = R$ for n -by- m matrices Q , U , and R for some $m > 1$. In these circumstances the block Lanczos algorithm with SO would seem to be the most effective procedure for computing eigenvalues and eigenvectors.

Exercises on Section 15.11

- 15.11.1. Assume that A and M have half-bandwidth m and that M has been factored into LL^* . Do an operation count for one step of the algorithm assuming that Aq requires $(2m + 1)n$ ops.
- 15.11.2. Of what residual is β_{ji} the norm?
- 15.11.3. Verify that the same matrix T_j is produced, in exact arithmetic, by the algorithms of Chapter 13 acting on \hat{A} ($\equiv L^{-1}AL^{-*}$) and by the algorithm of this section applied to (A, M) .

15.12. Subspace Iteration

The methods described in Chapter 14 extend readily to the problem $(A - \lambda M)x = o$ when M is positive definite. As pointed out in section 15.9 the cost of one step of the power method (or direct iteration) is approximately the same as one step of inverse iteration (unless M is diagonal) and the improved convergence rate of the latter makes it the preferred version for computing the small eigenvalues of large, sparse pencils (A, M) . The basic idea of subspace iteration

is to combine block inverse iteration, using occasional shifts of origin, with the RR approximations at each step.

Sophisticated implementations of the method have become the standard tool in the dynamic analysis of structures of all kinds. Typical problems have $n > 1000$, but fortunately \mathbf{A} and \mathbf{M} usually have a half-bandwidth around 100 and the pencil (\mathbf{A}, \mathbf{M}) is positive definite. Perhaps the 10 smallest eigenvalues are wanted together with their eigenvectors or, for earthquake calculations, all the eigenvalues less than some designated value.

At each step k the algorithm computes an n -by- m matrix $\mathbf{S}^{(k)}$ whose columns are the current eigenvector approximations. As shown in Chapter 14 some columns will converge faster than others and in practice it is important to take advantage of this phenomenon although we shall discuss it no further. One disadvantage of subspace iteration is that m should be chosen larger than the number of eigenvectors actually wanted. The improved convergence rate more than makes up for the extra work, but there is no really satisfying way to decide a priori on how many extra columns to carry. What is more the efficiency of the program depends quite sensitively on the choice of blocksize.

The algorithm is set out in Table 15.1.¹⁴

15.12.1. Comments on Table 15.1

(I). The selection of σ is discussed in section 14.5. In many applications there is extra information available to guide the user. Here is an instance of the difficulty in making subspace iteration into a black box program. The efficiency depends strongly on the choice of this parameter.

(II). See Chapter 3 for more details. Check $\nu(\Delta)$ to see where σ actually slices the spectrum.

(III). Sometimes knowledge of the application can guide the selection of starting vectors. The columns of $\mathbf{S}^{(0)}$ need not be orthonormal with respect to \mathbf{M} but they should be strongly linearly independent. In the absence of adscititious information a reasonable choice is to take column 1 as the sum of the first $[n/m]$ columns of I_n , column 2 as the sum of the second $[n/m]$ columns, and so on. Vectors with random elements do not seem to be very effective. The starting vectors must be orthogonalized against any known eigenvectors.

1. When eigenvectors are computed in batches (l at a time) it is important to keep the current subspace orthonormal (with respect to \mathbf{M}) to all known eigenvectors. However, it is not necessary to orthogonalize at each iteration. The frequency with which a known eigenvector \mathbf{z}_i should be purged depends

¹⁴ Adapted from [Wilson, 1978]

TABLE 15.1
Summary of subspace iteration.

Data A, M n -by- n , average half-bandwidth ω for both A and M . $S^{(k)}$ n -by- m , approximate eigenvectors at step k . $\Theta^{(k)}$ m -by- m , diagonal matrix of Ritz values. $1 \leq m < \omega < n$.	
INITIAL CALCULATIONS:	OPERATION COUNT
I. Select origin shift σ . II. Factor $A - \sigma M$ into $L\Delta L^*$. III. Select m starting vectors to be the columns of $S^{(0)}$. Set $\Theta^{(0)} = I_m$.	$n(\omega + 1)(\omega + 2)/2$ $\approx nm$
ITERATION: FOR $k = 1, 2, \dots$	
1. Orthogonalize $S^{(k-1)}$ against known eigenvectors, when necessary. 2. Form right-hand side, scaled by $\Theta^{(k-1)}$, $R^{(k)} = MS^{(k-1)}\Theta^{(k-1)}$. 3. Solve $L\Delta L^* \bar{S}^{(k)} = R^{(k)}$. 4. "project" A and M on $\text{span } \bar{S}^{(k)}$ to get $A^{(k)} = \bar{S}^{(k)*}R^{(k)}$, $M^{(k)} = \bar{S}^{(k)}M\bar{S}^{(k)}$. 5. Solve the m -by- m eigenvalue problem $(A^{(k)} - \sigma M^{(k)})G^{(k)} = M^{(k)}G^{(k)}\Theta^{(k)}$ for shifted Ritz values $\Theta^{(k)}$ and eigenvector matrix $G^{(k)}$ which is orthonormal with respect to $M^{(k)}$. 6. Form new basis $S^{(k)} = \bar{S}^{(k)}G^{(k)}$. 7. Test for convergence.	$2nm(\omega + 1)$ $2nm\omega$ $2nm^2 + nm(2\omega + 1)$ $10m^3$ nm^2
SUBTOTAL	$3nm(2\omega + m + 1) + \dots$

on the ratio $\max_j |\theta_j^{(k-1)}| / |\lambda_i - \sigma|$. Thus if λ_i is very close to σ then \mathbf{z}_1 will converge in one or two steps and thereafter it can be deflated by restricting the columns of $\mathbf{S}^{(k)}$ to be orthogonal to \mathbf{z}_1 . Of course m is reduced with resulting gains in efficiency.

4. As k increases, the full m -by- m matrices $\mathbf{A}^{(k)}$ and $\mathbf{M}^{(k)}$ are increasingly dominated by their diagonal elements. This makes the Jacobi method of described in section 15.7 very attractive, despite the absence of any proof of convergence, because the number of sweeps required to make the off-diagonal elements negligible is usually between 3 and 6, depending on the working precision.

15.13. Practical Considerations

Those who are concerned with the computation of eigenvalues and eigenvectors of large pencils (\mathbf{A}, \mathbf{M}) are well aware that success depends on the happy combination of techniques for (a) generating \mathbf{A} and \mathbf{M} , (b) moving information in and out of the fast store, (c) computing the results, and (d) handling input/output. For large problems it becomes increasingly difficult to isolate the numerical method from the rest of the environment. An important consequence is the difficulty of measuring, *a priori*, the cost of a method and hence its efficiency.

When it is possible to execute m matrix–vector products $\mathbf{Ax}_1, \mathbf{Ax}_2, \dots, \mathbf{Ax}_m$ almost as quickly as a single one then the relative speeds of contending methods may be strongly affected and our habitual judgments are upset. At a lower level we observe that the times required for a fetch, a store, an add, a multiply, and a divide are all getting closer to each other and this trend will make many conventional operation counts misleading. On the other hand the importance of the divide/multiply ratio has simply moved to a higher level of abstraction: the problem-dependent ratio of the cost of solving $\mathbf{Ax} = \mathbf{b}$ to the cost of forming \mathbf{Ab} . This is where the sparsity structure and the computer system enter the picture in a crucial way.

This book has concentrated on those aspects of the numerical methods which are independent of the vagaries of the computing system but there is no intention of belittling the others. As numerical methods become better adapted to the current facilities so will the bottleneck in these routine calculations move to the data management facets of the task.

Notes and References

Matrix pencils are discussed in some detail in [Gantmacher, 1959, Vol. II, Chapter 2]. Most of the material in section 15.2 comes from [Moler and

Stewart, 1973]. Definite quadratic forms are discussed in many books: [Strang, 1976], [Stewart, 1973], and [Franklin, 1968] to name a few.

The explicit reduction to standard form is covered in [Wilkinson, 1965] but the various options are not usually placed in close proximity. Section 15.5 comes from [Fix and Heiberger, 1972], section 15.6 from [Moler and Stewart, 1973], and section 15.7 from [Zimmermann, 1969] via [Bathé and Wilson, 1976].

The implicit reduction to standard form is well known but is not usually presented in this way. The extension of the Rayleigh quotient iteration and the RR procedure to pencils is given in some detail because the material did not seem to be readily available elsewhere. It was tempting to present the RR procedure from a more abstract point of view, letting it emerge naturally from the proper geometric setting. However, the necessary preparation seems to exceed the reward.

References for coordinate relaxation are [Schwarz, 1974 and 1977] for theory and [Shavitt, Bender, and Pipano, 1973] on the practical aspects. This book does not treat relaxation methods and the reader is referred to the timely and comprehensive survey [Ruhe, 1977].

References for the Lanczos algorithm applied to (A, M) are [Weaver and Yoshida, 1971], [Golub, Underwood, and Wilkinson, 1972], and [Cullum and Donath, 1974]. The idea of using $M^{-1}A$ implicitly can be found in [Wiberg, 1973] and [McCormick and Noe, 1977]. A termination criterion based on watching the behavior of the Ritz value $\theta_i^{(j)}$ is give in [van Kats and van der Vorst, 1976]. The idea of the band Lanczos comes from [Ruhe, 1979].

The structural engineers have done such extensive work with subspace iteration that the tale of all the gimmicks and modifications they have tried would be a long one. Two references, [Bathé and Wilson, 1976] and [Jennings and McKeown, 1992], seem to synthesize all that experience. The whole field of large eigenvalue computations is surveyed in admirable style in [Stewart, 1976a].

Perhaps the best Lanczos code available in the 1990s is described in [Grimes, Lewis, and Simon, 1994].

Appendix A

Rank-One and Elementary Matrices

The elementary row operations on a matrix B are (a) interchange two rows, (b) multiply a row by a nonzero scalar, and (c) add a multiple of one row to another. They can be effected by premultiplying B by appropriate rather simple matrices which are traditionally called elementary. The concept of an elementary matrix gains significance in the light of the important result that any invertible matrix can be written as a product of elementary matrices (in many ways).

In the 1950s Householder revised the definition to make it both simpler and more general.

Definition. *An elementary matrix is any square matrix of the form*

$$I + a \text{ rank-one matrix.} \quad (\text{A-1})$$

Elementary matrices can be stored in a compact form and their inverses are easy to write down as we now show.

All *rank-one* matrices are of the form xy^* , since every column is a multiple of x and every row is a multiple of y^* . It is standard practice to normalize x and y and write a rank-one matrix as

$$u\sigma v^* \quad (\text{A-2})$$

where $\|u\| = \|v\| = 1$. Consequently (A-1) can be written as

$$E = I - u\sigma v^*. \quad (\text{A-3})$$

It is easy to verify that E^{-1} exists if and only if $v^* \sigma u \neq 1$, in which case

$$E^{-1} = I - u\tau v^*, \quad (\text{A-4})$$

where $\tau^{-1} + \sigma^{-1} = v^* u$. Thus (A-4) confirms that E^{-1} is also elementary. Observe that u, v, σ , and τ define both E and E^{-1} and, in practical work, the product EB can be formed with only $5n^2$ ops as against n^3 if the structure of E is ignored. Observe that, in general, E would be a full matrix if it were to be formed explicitly.

Elementary matrices are indeed useful tools.

Exercises on Appendix A

- A-1. Find the values of u and v and σ which yield the traditional elementary matrices described in the first sentence.
- A-2. Verify that (A-2), (A-3), and (A-4) continue to hold when u and v are allowed to be n -by- m matrices ($m \leq n$) satisfying $u^* u = v^* v = I_m$. Of course σ and τ become m -by- m matrices.

Appendix B

Chebyshev Polynomials

For a given positive integer k the function $\cos k\phi$ is *not* a polynomial in ϕ but it does happen to be a polynomial of degree k in $\cos \phi$. This polynomial is, of course, well defined for values of its argument outside the interval $[-1, 1]$ but it cannot be expressed in terms of cosines. If $\cos \phi = \xi$ then $\phi = \cos^{-1} \xi = \arccos \xi$ and

$$\begin{aligned} T_k(\xi) &= \begin{cases} \cos(k \arccos \xi) = \cos k\phi, & -1 \leq \xi \leq 1, \\ \cosh(k \arccosh \xi), & |\xi| \geq 1, \end{cases} \\ &= \frac{1}{2} \left[\left(\xi + \sqrt{\xi^2 - 1} \right)^k + \left(\xi + \sqrt{\xi^2 - 1} \right)^{-k} \right], \quad |\xi| > 1, \\ &\approx \frac{1}{2}(2\xi)^k \text{ as } \xi \rightarrow \infty, \text{ } k \text{ fixed.} \end{aligned}$$

Moreover

$$\begin{aligned} T_k(1) &= 1 \text{ for all } k, \\ T_k(1 + 2\epsilon) &\approx \frac{1}{2} (1 + 2\sqrt{\epsilon} + 2\epsilon)^k \quad \text{for } 0 \leq \epsilon < 0.1, \\ &\approx \frac{1}{2} \exp(2k\sqrt{\epsilon}) \quad \text{for } k\sqrt{\epsilon} > 1. \end{aligned}$$

It is the rapid growth of $T_k(1 + \epsilon)$ for small ϵ and large k that makes Chebyshev acceleration so useful. See Table B-1.

A simple three-term recurrence permits the evaluation of $T_k(\xi)$ without knowledge of T_k 's coefficients. The recurrence is a disguised form of

$$\cos(k + 1)\phi + \cos(k - 1)\phi = 2 \cos \phi \cos k\phi,$$

TABLE B-1
Representative values of $T_m(1 + 2\gamma)$.

γ m	10^{-4}	10^{-3}	10^{-2}	10^{-1}
10	1.0201	1.2067	3.7502	2.5227×10^2
10^2	3.7621	2.7876×10^2	2.3466×10^8	5.3436×10^{26}
2×10^2	2.7306×10^1	1.5542×10^5	1.1014×10^{17}	5.7107×10^{53}
10^3	2.4250×10^8	1.4507×10^{27}	2.5927×10^{86}	9.7179×10^{269}

namely,

$$T_{k+1}(\xi) = 2\xi T_k(\xi) - T_{k-1}(\xi).$$

The polynomial $T_k(\xi)/2^{k-1}$ is the smallest monic polynomial of degree k provided that the size of a function is taken as its maximal absolute value on $[-1, 1]$. Of more importance is the related fact that of all polynomials p of degree $\leq k$ which satisfy $p(\gamma) = \delta$ for some $|\gamma| > 1$ the smallest on $[-1, 1]$ is

$$\bar{p}(\xi) = \frac{\delta T_k(\xi)}{T_k(\gamma)}$$

and

$$\|\bar{p}\|_\infty = \max_{-1 \leq \xi \leq 1} |\bar{p}(\xi)| = \frac{|\delta|}{T_k(\gamma)}.$$

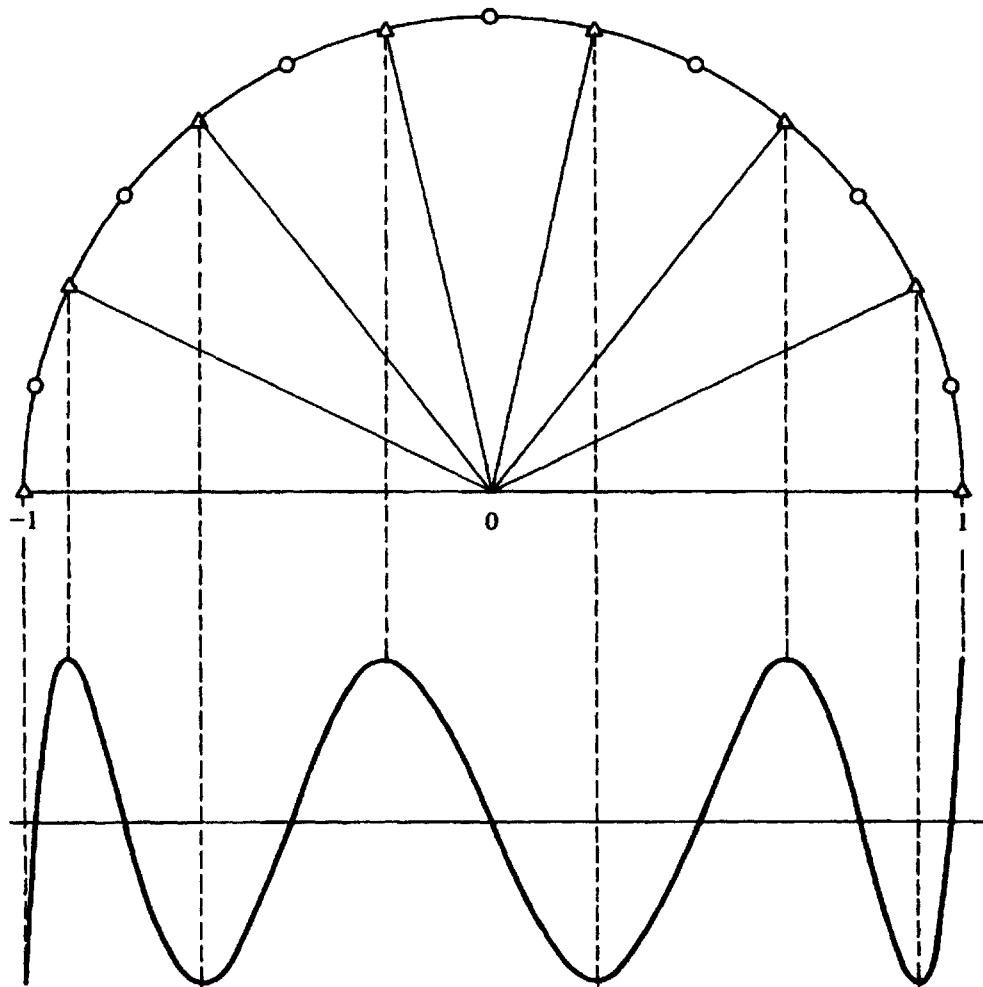
In order to get the analogous results for an interval $[\alpha, \beta]$ use the adapted Chebyshev polynomial

$$\begin{aligned} T_k(\xi : \alpha, \beta) &\equiv T_k \left(\frac{2\xi - \alpha - \beta}{\beta - \alpha} \right) \\ &= T_k \left(1 + 2 \frac{\xi - \beta}{\beta - \alpha} \right). \end{aligned}$$

The leading coefficient of $T_k(\xi : \alpha, \beta)$ is $\frac{1}{2}[4/(\beta - \alpha)]^k$.

The optimal characteristics of T_k stem from its famous *equioscillation* property: on $[-1, 1]$ T_k switches between its maximum absolute value with alternating signs exactly k times. No other polynomial of degree k or less can equioscillate more than k times.

Extrema of T_7 are $\cos(j\pi/7)$, $j = 0, \dots, 7$



Key: Δ Extrema
 \circ Zeros

FIG. B-1. Graph of T_7 . Δ extrema, \circ zeros.

This page intentionally left blank

Annotated bibliography

Matrix Theory

Bhatia, R. 1987. *Perturbation Bounds for Matrix Eigenvalues*. New York: Longman Scientific and Technical.

A slim, well-written monograph which begins with the Hermitian case and extends Weyl's monotonicity theorem, described in Chapter 10, to other matrix classes such as the group of n -by- n orthogonals and the strange set of normal matrices.

Franklin, J. N. 1968. *Matrix Theory*. Englewood Cliffs, N.J.: Prentice-Hall.

Clear, attractive to mathematicians, brings out the use of matrix theory in the study of differential equations. The Jordan form is treated thoroughly and intelligibly.

Horn, R. and C. Johnson. 1985. *Matrix Analysis*, New York: Cambridge University Press.

I think of this as the successor to the Marcus and Minc compendium of results on matrix theory *A Survey of Matrix Theory and Matrix Inequalities*. Boston: Allyn and Bacon, 1964.

Horn, R. and C. Johnson. 1991. *Topics in Matrix Analysis*. New York: Cambridge University Press.

Noble, B., and J. W. Daniel. 1977. *Applied Linear Algebra*. 2d ed. Englewood Cliffs, N.J.: Prentice-Hall.

Popular, broad in scope, not too demanding of the reader. The new edition brings several improvements and removes several errors.

Strang, G. 1976. *Linear Algebra and Its Applications*. New York: Academic Press.

My favorite. Compact, stylish, every example and exercise brings out a worthwhile point. The thrust is toward applications, but here mathematics is used to enhance understanding, not build an elaborate theory.

Matrix Computations (Introduction)

Fox, L. 1964. *An Introduction to Numerical Linear Algebra*. New York: Oxford University Press.

A very clear exposition at a fairly elementary level. Despite its age it still makes a very good text for a first course in the subject.

Gourlay, A. R., and G. A. Watson. 1973. *Computational Methods for Matrix Eigenproblems*. New York: John Wiley.

A beautiful little book. Presents the essentials simply and directly. No programs.

Schwarz, H. R., H. Rutishauser, and E. Stiefel. 1973. *Numerical Analysis of Symmetric Matrices*. Englewood Cliffs, N.J.: Prentice-Hall.

The flavor is quite different from both Fox and Stewart. The presentation of classical, background material is masterly, but some of the numerical methods described with such care are surprisingly out of date. Algol programs are used to illustrate the methods.

Stewart, G. W. 1973. *Introduction to Matrix Computations*. New York: Academic Press.

Comprehensive, authoritative, and readable. More demanding than Fox. Programs are included.

Watkins, D. S. 1991. *Fundamentals of Matrix Computations*. New York: John Wiley and Sons.

The students and I were pleased with this well-written textbook.

Matrix Computations (Specialized)

Demmel, J. W. 1997. *Applied Numerical Linear Algebra*. Philadelphia: SIAM.

This book superseded my old notes as the text for a graduate level course on matrix computations. It is an impressive example of the huge amount of information that one fine mind can digest and organize.

- Duff, I. S., A. M. Erisman, and J. K. Reid. 1987. *Direct Methods for Sparse Matrices*. New York: Oxford University Press.
An up-to-date comprehensive account of the subject.
- George, A. and J. W. Liu. 1980. *Computer Solution of Large Sparse Positive Definite Systems*. Englewood Cliffs, N.J.: Prentice-Hall.
A detailed, error-free discussion of the Cholesky factorization.
- Golub, G. H. and C. F. van Loan. 1997. *Matrix Computations*. 3d ed. Baltimore and London: The Johns Hopkins University Press.
This successful compendium covers far more than eigenvalue problems. It is a textbook as well as a reference.
- Householder, A. S. 1964. *The Theory of Matrices in Numerical Analysis*. New York: Blaisdell.
Not surprisingly, the numerical methods covered are quite out of date. However, the first three chapters present valuable background topics from matrix theory: projections, factorizations, norms and convex bodies, the Perron-Frobenius theory, and inequalities involving eigenvalues. This material and the extensive problems and exercises retain their value. The book is densely written, as for professional mathematicians, and was very influential in the field.
- Jennings, A. 1977. *Matrix Computation for Engineers and Scientists*. New York: John Wiley.
Most methods in current use for matrix problems, and some obsolete techniques, are described in good detail. The book is strong on examples and experiments and is less concerned with providing a theory to relate and explain the behavior of various methods. The language is aimed at structural engineers.
- Jennings, A. and J. J. McKeown. 1992. *Matrix Computation*. 2d. ed., New York: John Wiley.
This is an updated version of the first edition.
- Saad, Y. 1992. *Numerical Methods for Large Eigenvalue Problems*. New York: Halsted; 1996. Boston: PWS Publishing Co.
- Stewart, G. W. and J. Sun. 1990. *Matrix Perturbation Theory*. San Diego: Academic Press.

All you need to know (nearly) about the sensitivity of the numbers we compute to small uncertainties in the data.

Wilkinson, J. H. 1965. *The Algebraic Eigenvalue Problem*. New York: Oxford University Press.

Those parts of matrix theory and perturbation theory which bear on numerical methods are presented first. Next comes roundoff error analysis. The major part of the book is a masterly and detailed account of all the important eigenvalue techniques. The analysis covers both exact arithmetic and noisy arithmetic. Several important examples illustrate the often surprising effects of round-off error. Effective use is made of "backward" error analyses, for which Wilkinson is justly renowned, to explain in a simple and compelling way the behavior of each numerical method.

The author is not concerned with formalizing his understanding into theorems, and his style makes the book frustrating as a reference manual. Despite these blemishes, this book is "the bible" for those who work with matrix computations.

Wilkinson, J. H., and C. Reinsch, Eds. 1971. *Handbook for Automatic Computation*. Vol. II Linear Algebra. New York: Springer-Verlag.

This is a collection of some 82 programs in the Algol language and, in a sense, synthesizes 20 years of work aimed at relegating the basic analysis of matrices to the computer. Along with each program is valuable supporting documentation. Not all the contributions are by the two authors but each program bears the stamp of their influence and advice.

Many of these algorithms are available in the EISPACK collection.

I have chosen to point to the appropriate, readily available, well-tested programs in the Handbook rather than present versions of my own.

References

- Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. 1995. *LAPACK Users' Guide, Release 2.0*, 2d ed., Philadelphia: SIAM.
- Bargmann, V., C. Montgomery, and J. von Neumann. 1946. *Solution of Linear Systems of High Order*. Princeton, N.J.: Institute for Advanced Study.
- Bathé, K. -J., and E. Wilson. 1976. *Numerical Methods in Finite Element Analysis*. Englewood Cliffs, N.J.: Prentice-Hall.
- Bauer, F. L. 1957. Das Verfahren der Treppeniteration und Verwandte Verfahren zur Lösung Algebraischer Eigenwertprobleme. *ZAMP*, 8:214–235.
- Bischof, C. and C. F. van Loan. 1987. The WY representation for products of Householder matrices. *SIAM J. Sci. Statist. Comput.* 8:s2–s13.
- Bleher, J. H., S. M. Rump, U. Kulisch, M. Metzger, Ch. Ullrich, and W. Walter. 1987. A study of a FORTRAN extension for engineering/scientific computation with access to ACRITH. *Computing* 39:93–110.
- Brent, R. P. 1973. *Algorithms for Minimization without Derivatives*. Englewood Cliffs, N.J.: Prentice-Hall.
- Browne, E. T. 1930. On the separation property of the roots of the secular equation. *Amer. J. Math.* 52:841–850.
- Bus, J. C., and T. J. Dekker. 1975. Two efficient algorithms with guaranteed convergence for finding a zero of a function. *Trans. Math. Software* 1:330–345.

- Cao, Z.-H., J.-J. Xie, and R.-C. Li. 1996. A sharp version of Kahan's theorem on clustered eigenvalues. *Linear Algebra Appl.* 245:147–155.
- Cauchy, A. 1829. *Sur l'équation à l'aide de laquelle on determine les inégalités séculaire des Mouvements des Planètes.* Oeuvres Complètes, Series II, Tome IX (Gauthiers Villars, Paris, 1891).
- Chatelin, F., and J. Lemordant. 1978. Error bounds in the approximation of the eigenvalues of differential and integral operators. *J. Math. Anal. Appl.* 22:257–271.
- Cline, A. K., G. H. Golub, and G. W. Platzman. 1976. Calculation of normal modes of oceans using a Lanczos method. In *Sparse Matrix Computations*. Edited by J. R. Bunch and D. J. Rose, pp. 409–426. New York: Academic Press.
- Corbato, F. J. 1963. On the coding of Jacobi's method for computing eigenvalues and eigenvectors of real symmetric matrices. *J. Assoc. Comput. Mach.* 10:123–125.
- Corneil, D. 1965. *Eigenvalues and Orthogonal Eigenvectors of Real Symmetric Matrices.* Master's thesis, Toronto: University of Toronto, Dept. of Computer Science.
- Courant, R. 1920. Über die Eigenwert bei den Differentialgleichungen der Mathematischen Physik. *Math. Z.* 7:1–57.
- Crandall, S. H. 1951. Iterative procedures related to relaxation methods for eigenvalue problems, *Proc. Roy. Soc. London Ser. A*, 207:416–423.
- Crawford, C. R. 1973. Reduction of a band-symmetric generalized eigenvalue problem. *Comm. ACM* 16:41–44.
- Cullum, J., and W. E. Donath. 1974. *A Block Generalization of the Symmetric S-step Lanczos Algorithm.* Report #RC 4845 (#21570). Yorktown Heights, N.Y.: IBM Thomas J. Watson Research Center.
- Cullum, J. K., and R. A. Willoughby. 1985. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations.* Vol. I Theory, Vol. II Programs. Boston: Birkhäuser.
- Daniel, J. W., W. B. Gragg, L. Kaufman, and G. W. Stewart. 1976. Re-orthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization. *Math. Comp.* 30:772–795.

- Davidson, E. R. 1975. The iterative calculation of a few of the lowest eigenvalues of corresponding eigenvectors of large real symmetric matrices. *J. Comput. Phys.* 17:87–94.
- Davis, C. 1958. Separation of two linear subspaces. *Acta Sci. Math. Szeged* 19:172–189.
- Davis, C., and W. Kahan. 1969. Some new bounds on perturbation of subspaces. *Bull. Amer. Math. Soc.* 75:863–868.
- . 1970. The rotation of eigenvectors by a perturbation-III. *SIAM J. Numer. Anal.* 7:1–46.
- De Boor, C., and G. H. Golub. 1978. The numerically stable reconstruction of a Jacobi matrix from spectral data. *Linear Algebra Appl.* 21:245–260.
- Dekker, T. J., and J. F. Traub. 1971. The shifted QR algorithm for Hermitian matrices, *Linear Algebra Appl.* 11:137–154.
- Demmel, J., and K. Veselić. 1992. Jacobi's method is more accurate than QR. *SIAM J. Matrix Anal. Appl.* 13:1204–1246.
- Dhillon, I. S. 1998. Current inverse iteration software can fail. *BIT*. to appear.
- Dodson, D., and J. Lewis. 1985. *Issues Relating to Extension of the Basic Linear Algebra Subprograms*, ACM SIGNUM Newsletter, 20:2–18.
- Dongarra, J., J. Du Croz, I. Duff, and S. Hammarling. 1990. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Software* 16:1–17.
- Druskin, V. L., and L. A. Knisherman. 1991. Error estimates for the simple Lanczos process when calculating functions of symmetric matrices and eigenvalues, *J. Comput. Math. Math. Phys.* 31:970–983.
- Dubrulle, A. A. 1998. *QR Algorithm with Variable Iteration Multiplicity*. Cupertino, Cal.: Hewlett Packard. Linear Algebra Appl., to appear.
- Duff, I. S., R. G. Grimes, and J. G. Lewis. 1992. *Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release 1)*, Tech. Report RAL-92-086. Oxfordshire, England: Rutherford Appleton Laboratories.

- Erxiong, J., and Z. Zhenye. 1985. A new shift of the QL algorithm for irreducible symmetric tridiagonal matrices. *Linear Algebra Appl.* 65:261–272. (Authors name published here erroneously as Erxiong J. instead of Jiang E.)
- Faddeev, D. D., and V. N. Faddeeva. 1963. *Computational Methods of Linear Algebra*. Translated by R. C. Williams. San Francisco: W. H. Freeman.
- Feng, Y. 1991. *Backward Stability of Root Free QR Algorithms*, Ph.D. Thesis. Berkeley: University of California.
- Fischer, E. 1905. Über quadratische Formen mit reelen Koeffizienten, *Monatsch Math. Phys.* 16:234–249.
- Fix, G., and R. Heiberger. 1972. An algorithm for the ill-conditioned generalized eigenvalue problem. *SIAM J. Numer. Anal.* 9:78–88.
- Forsythe, G., and C. B. Moler. 1967. *Computer Solutions of Linear Algebraic Systems*, Englewood Cliffs, N.J.: Prentice-Hall.
- Francis, J. G. F. 1961 and 1962. The QR transformation, Parts I and II. *Computer J.* 4:265–271; 4:332–345.
- Franklin, J. N. 1968. *Matrix Theory*, Englewood Cliffs, N.J.: Prentice-Hall.
- Gantmacher, F. R. 1959. *The Theory of Matrices*, Vol. II, New York: Chelsea Publishing Co.
- Garbow, B. S., J. M. Boyle, J. J. Dongarra, and C. B. Moler. 1977. *Matrix Eigensystem Routines—EISPACK Guide*, Lecture Notes in Computer Science 6, Berlin: Springer-Verlag.
- Gentleman, W. M. 1973. Least squares computations by Givens transformations without square roots. *J. Inst. Math. Appl.* 12:329–336.
- Gill, P. E., G. H. Golub, W. Murray, and M. A. Saunders. 1974. Methods for modifying matrix factorizations. *Math. Comp.* 28:505–535.
- Gill, P. E., W. Murray, and M. A. Saunders. 1975. Methods for computing and modifying the LDU factors of a matrix. *Math. Comp.* 29:1051–1077.
- Givens, W. 1954. Numerical Computation of the Characteristic Values of a Real Symmetric Matrix. ORNL-1574. Oak Ridge, Tenn.: Oak Ridge National Laboratory.

- Glauz, G. 1974. Private communication.
- Golub, G. H. 1973. Some uses of the Lanczos algorithm in numerical linear algebra. In *Topics in Numerical Analysis*. Edited by J. J. H. Miller. pp. 173–184. New York: Academic Press.
- Golub, G. H., R. Underwood, and J. H. Wilkinson. 1972. *The Lanczos Algorithm for the Symmetric Ax = λBx Problem*. Tech. Report STAN-CS-72-270. Stanford, Cal.: Stanford University, Computer Science Dept.
- Golub, G. H., and J. H. Welsch. 1969. Calculation of Gauss quadrature rules. *Math. Comp.* 23:221–230.
- Gragg, W. B. and W. J. Harrod. 1984. The numerically stable reconstruction of Jacobi spectral data. *Numer. Math.* 44:317–336.
- Greenbaum, A. 1989. Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra Appl.* 113:7–63.
- Greenbaum, A. and Z. Strakos. 1996. Predicting the behavior of finite-precision Lanczos and conjugate gradient computations. *SIAM J. Matrix Anal. Appl.* 13:121–137.
- Grimes, R., J. G. Lewis, and H. Simon. 1994. A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM J. Matrix Anal. Appl.* 15:228–272.
- Hald, O. H. 1977. Inverse eigenvalue problems for Jacobi matrices. *Linear Algebra Appl.* 14:63–85.
- . 1977. Discrete inverse Sturm-Liouville problems. *Numer. Math.* 27:249–256.
- Hammarling, S. 1974. A note on modification to the Givens plane rotation. *J. Inst. Math. Appl.* 13:215–218.
- Hari, V. 1991. On sharp quadratic convergence bounds for the serial Jacobi methods. *Numer. Math.* 60:375–406.
- Henrici, P. 1958. On the speed of convergence of cyclic and quasicyclic Jacobi methods for computing eigenvalues of Hermitian matrices. *J. SIAM*, 6:144–162.
- Hill, R. O., Jr., and B. N. Parlett. 1992. Refined interlacing properties. *SIAM J. Matrix Anal. Appl.* 13:239–247.

- Hochstadt, H. 1975. On inverse problems associated with Sturm-Liouville operators. *J. Differential Equations* 17:220–235.
- Hoffman, W., and B. N. Parlett. 1978. A new proof of global convergence for the tridiagonal QL algorithm. *SIAM J. Numer. Anal.* 15:929–937.
- Hotelling, H. 1943. Some new methods in matrix calculation. *Ann. Math. Stat.* 14:1–34.
- Householder, A. S. 1958. A class of methods for inverting matrices. *J. SIAM*, 6:189–195.
- . 1961. On deflating matrices. *SIAM J. Appl. Math.* 9:89–93.
- Ikebe, Y., T. Inagaki, and S. Miyamoto. 1987. The monotonicity theorem, Cauchy's theorem, and the Courant-Fischer theorem. *Amer. Math. Monthly* 94:352–354.
- Jacobi, C. G. J. 1846. Concerning an easy process for solving equations occurring in the theory of secular disturbances. *J. Reine Angew. Math.* 30:51–94.
- Jennings, A., and T. J. A. Agar. 1978. Hybrid Sturm sequence and simultaneous iteration methods. In *Proc. of Symp. on Applic. of Computer Methods in Eng.* Los Angeles: University of Southern California Press.
- Jensen, P. S. 1972. The solution of large symmetric eigenproblems by sectioning. *SIAM J. Numer. Anal.* 9:534–545.
- Kahan, W. 1966. *When to Neglect Off-Diagonal Elements of Symmetric Tridiagonal Matrices*. Tech. Report CS42. Stanford, Cal.: Stanford University, Computer Science Dept.
- . 1967. *Inclusion Theorems for Clusters of Eigenvalues of Hermitian Matrices*. Tech. Report CS42. Toronto: University of Toronto, Computer Science Dept.
- Kahan, W., and B. N. Parlett. 1976. How far should you go with the Lanczos algorithm? In *Sparse Matrix Computations*. Edited by J. R. Bunch and D. J. Rose, pp. 131–144. New York: Academic Press.
- Kaniel, S. 1966. Estimates for some computational techniques in linear algebra. *Math. Comp.* 20:369–378.

- Kato, T. 1949. On the upper and lower bounds of eigenvalues. *J. Phys. Soc. Japan* 334–339.
- . 1966. *Perturbation Theory for Linear Operators*. New York: Springer-Verlag. For general perturbation theory see Chap. 2.
- Knisherman, L. A. 1995. The quality of approximations to a well separated eigenvalue, and the arrangement of “Ritz Numbers” in the simple Lanczos process. *Comput. Math. and Math. Phys.* 35:1175–1197.
- Knuth, D. E. 1969. *The Art of Computer Programming. Fundamental Algorithms*. Vol. I. Reading, Mass.: Addison-Wesley Publishing Co.
- Knyazev, A. 1994. New estimates for Ritz vectors. In *Proc. of the Fifth SIAM Conf. on Appl. Linear Algebra*. Snowbird, Utah.
- Krylov, A. N. 1931. On the numerical solution of equations which in technical questions are determined by the frequency of small vibrations of material systems. *Izv. Akad. Nauk. S.S.R. Otdel. Mat. Estest.* 1:491–539.
- Kublanovskaya, V. N. 1961. On some algorithms for the solution of the complete eigenvalue problem. *Zh. Vychisl. Mat.* 1:555–570.
- Lanczos, C. 1950. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards, Sec. B* 45:225–280.
- LAPACK Working Note #3, 1988. *Computing Small Singular Values of Bidiagonal Matrices with Guaranteed High Relative Accuracy*. Argonne, Ill.: Argonne National Laboratory.
- Lawson, C., R. Hanson, D. Kincaid, and F. Krogh. 1979. Basic linear algebra subprograms for Fortran usage. *ACM Trans. Math. Software* 5:308–323; 5:324–325.
- Lehmann, N. J. 1949. Calculation of eigenvalue bounds in linear problems. *Arch. Math.* 2:139–147.
- . 1963. Optimal eigenvalue localization in the solution of symmetric matrix problems. *Numer. Math.* 5:246–272.
- . 1966. On optimal eigenvalue localization in the solution of symmetric matrix problems. *Numer. Math.* 8:42–55.

- Li, Ch.-K., and R. Mathias. 1998. On the Lidskii-Mirsky-Wielandt theorem. *Numer. Math.* to appear.
- Li, R.-C. 1994. *Relative Perturbation Theory: (I) Eigenvalue Variations*, Tech. Report CS-94-252, LAPACK Working Note #84. Knoxville: University of Tennessee, Computer Science Dept.
- . 1994. *Relative Perturbation Theory: (II) Eigenspace Variations*, Tech. Report CS-94-252, LAPACK Working Note #85. Knoxville: University of Tennessee, Computer Science Dept.
- McCormick, S. F., and T. Noe. 1977. Simultaneous iteration for the matrix eigenvalue problem. *Linear Algebra Appl.* 16:43–56.
- Modi, J. J. 1988. *Parallel Algorithms and Matrix Computation*, Oxford: Clarendon Press.
- Moler, C. B., and G. W. Stewart. 1973. An algorithm for generalized matrix eigenvalue problems. *SIAM J. Numer. Anal.* 10:241–256.
- Nour-Omid, B. and B. N. Parlett. 1985. The use of refined error bounds when updating eigenvalues of tridiagonals. *Linear Algebra Appl.* 68:179–220.
- Ortega, J. M., and H. F. Kaiser. 1963. The \mathbf{LL}^T and QR methods for symmetric tridiagonal matrices. *Numer. Math.* 5:211–225.
- Ostrowski, A. M. 1958, 1959. On the convergence of the Rayleigh quotient iteration for the computation of characteristic roots and vectors, I and II. *Arch. Rational Mech. Anal.* 1:233–241; 2:423–428.
- Paige, C. C. 1971. *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*. Ph.D. thesis. London: Univ. of London.
- . 1972. Computational variants of the Lanczos method for the eigenproblem. *J. Inst. Math. Appl.* 10:373–381.
- . 1976. Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix. *J. Inst. Math. Appl.* 18:341–349.
- Parlett, B. N. 1964. The origin and development of methods of LR type. *SIAM Rev.* 6:275–295.
- . 1971. Analysis of algorithms for reflections in bisectors. *SIAM Rev.* 13:197–208.

- . 1974. The Rayleigh quotient iteration and some generalizations for nonnormal matrices. *Math. Comp.* 28:679–693.
- . 1990. Misconvergence in the Lanczos algorithm. In *Reliable Numerical Computation*. Edited by M. G. Cox and S. Hammarling. Oxford: Clarendon Press.
- . 1991. Symmetric matrix pencils. *J. Comput. Appl. Math.* 38:373–385.
- . 1992. The rewards for maintaining semi-orthogonality among Lanczos vectors. *Numer. Linear Algebra Appl.* 1:243–267.
- . 1995. The new qr algorithms. In *Acta Numerica 1995*. pp. 459–491. Cambridge: Cambridge University Press.
- Parlett, B. N., and W. Kahan. 1969. On the convergence of a practical QR algorithm. In *Information Processing 68* (Proc. IFIP Congress, Edinburgh, 1968). *Mathematical Software*, sect. 1, pp. 114–118. Amsterdam: North-Holland.
- Parlett, B. N., and W. G. Poole. 1973. A geometric convergence theory for the QR, LU, and power iterations. *SIAM J. Numer. Anal.* 10:389–412.
- Parlett, B. N., and D. S. Scott. 1979. The Lanczos algorithm with selective orthogonalization. *Math. Comp.* 33:217–238.
- Rayleigh, Lord (J. W. Strutt). 1899. On the calculation of the frequency of vibration of a system in its gravest mode, with an example from hydrodynamics. *Philos. Mag.* 47:556–572.
- Reinsch, C. H. 1971. A stable rational QR algorithm for the computation of the eigenvalues of an Hermitian, tridiagonal matrix. *Numer. Math.* 25:591–597.
- Ritz, W. 1909. Über eine neue Method zur Lösung Gewisser Variationsprobleme der Mathematischen Physik. *J. Reine. Angew. Math.* 135:1–61.
- Ruhe, A. 1975. Iterative eigenvalue algorithms based on convergent splittings. *J. Comput. Phys.* 19:110–120.
- . 1977. Computation of eigenvalues and eigenvectors. In *Sparse Matrix Techniques, Copenhagen, 1976, LNM*, pp. 130–184. New York: Springer-Verlag.

- . 1979. Implementation aspects of band Lanczos algorithm for computation of eigenvalues of large sparse matrices. *Math. Comp.* 33:680–687.
- . 1984. Rational Krylov sequence methods for eigenvalue computations. *Linear Algebra Appl.* 58:391–405.
- Ruhe, A., and T. Ericsson. 1980. The spectral transformation Lanczos method in the numerical solution of large, sparse, generalized, symmetric eigenvalue problems. *Math. Comp.* 35:1251–1268.
- Rutishauser, H. R. 1969. Computational aspects of F. L. Bauer's simultaneous iteration method. *Numer. Math.* 13:4–13.
- . 1971a. The Jacobi method for real symmetric matrices. In *Handbook for Automatic Computation (Linear Algebra)*. Edited by J. H. Wilkinson and C. H. Reinsch, pp. 202–211. New York: Springer-Verlag.
- . 1971b. Simultaneous iteration method for symmetric matrices. In *Handbook for Automatic Computation (Linear Algebra)*. Edited by J. H. Wilkinson and C. H. Reinsch, pp. 284–302. New York: Springer-Verlag.
- Saad, Y. 1974. *Shifts of Origin for the QR Algorithm*. Proceedings International Federation of Information Processing Societies Congress, Toronto.
- . 1980. Error bounds on the interior Rayleigh–Ritz approximations from Krylov subspaces. *SIAM J. Numer. Anal.* 17:687–706.
- Sack, R. A. 1972. A fully stable rational version of the QR algorithm for tridiagonal matrices. *Numer. Math.* 18:432–441.
- Schönhage, A. 1961. On the convergence of the Jacobi process. *Numer. Math.* 3:374–380.
- Schwarz, H. R. 1970. The method of conjugate gradients in least squares fitting. *Z. Vermessungsweser* 95:130–140.
- . 1974. The eigenvalue problem $(A - \lambda B)x = 0$ for symmetric matrices of high order. *Comput. Methods Appl. Mech. Engrg.* 3:11–28.
- . 1977. More results on $(A - \lambda B)$. *Comput. Methods Appl. Mech. Engrg.* 12:181–199.
- Scott, D. 1978. *Analysis of the Symmetric Lanczos Algorithm*. Ph.D. dissertation, Berkeley: Univ. of California, Dept. of Mathematics.

- Shavitt, I., C. F. Bender, and A. Pipano. 1973. The iterative calculation of several of the lowest or highest eigenvalues and corresponding eigenvectors of very large symmetric matrices. *J. Comput. Phys.* 11:90–108.
- Simon, H. 1984a. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Linear Algebra Appl.* 61:101–132.
- . 1984b. The Lanczos algorithm with partial reorthogonalization. *Math. Comp.* 42:115–142.
- Slapričar, I. 1992. *Accurate Symmetric eigenreduction by a Jacobi Method*. Ph.D. thesis, Hagen, Germany: Ferruniversität.
- Sorensen, D. C. 1992. Implicit application of polynomial filters in a k -step Arnoldi method. *SIAM J. Matrix Anal. Appl.* 13:357–385.
- Stewart, G. W. 1970. Incorporating origin shifts into the QR algorithm for symmetric tridiagonal matrices. *Comm. Assoc. Comput. Mach.* 13:365–367.
- . 1976a. A bibliographic tour of the large, sparse generalized eigenvalue problem. In *Sparse Matrix Computations*, Edited by J. R. Bunch and D. J. Rose. New York: Academic Press.
- . 1976b. The economic storage of plane rotations. *Numer. Math.* 25:137–138.
- Szegő, G. 1939. *Orthogonal Polynomials*. Am. Math. Soc. Colloq. Publ. No. 23. New York.
- Temple, G. 1933. The computation of characteristic numbers and characteristic functions. *Proc. London Math. Soc.* 2:257–280.
- . 1952. The accuracy of Rayleigh's method of calculating the natural frequencies of vibrating systems. *Proc. Roy. Soc. London Ser. A.* 211:204–224.
- Temple, G., and W. G. Bickley. 1933. *Rayleigh's Principle and Its Applications to Engineering*. London: Constable.
- Thompson, R. C., and P. McEnteggert. 1968. Principal submatrices II, The upper and lower quadratic inequalities. *Linear Algebra Appl.* 1:211–243.
- Traub, J. 1964. *Iterative Methods for the Solution of Equations*. Englewood Cliffs, N.J.: Prentice-Hall.

- Uhlig, F. 1979. A recurring theorem about pairs of quadratic forms and extensions: A survey. *Linear Algebra Appl.* 25:219–238.
- Underwood, R. 1975. *An Iterative Block Lanczos Method for the Solution of Large Sparse Symmetric Eigenproblems*. Ph.D. dissertation, STAN-CS-75-496, Stanford, Cal.: Stanford University.
- van Kats, J. M., and H. A. van der Vorst, 1976. *Numerical Results of the Paige-Style Lanczos Method for the Computation of Extreme Eigenvalues of Large Sparse Matrices*. Tech. Report 3. Utrecht, Netherlands: Academic Computer Center.
- Veselić, K. and V. Hari. 1990. A note on a one-sided Jacobi algorithm. *Numer. Math.* 56:627–633.
- Weaver, W., and D. M. Yoshida. 1971. The eigenvalue problem for banded matrices. *Comput. & Structures* 1:651–664.
- Weinberger, H. F. 1954. *A Rayleigh-Ritz Procedure Giving Upper and Lower Bounds for Eigenvalues*. Tech. Note BN-41, College Park: University of Maryland, Inst. for Fluid Dynamics and Applied Mathematics.
- . 1959. *A Theory of Lower Bounds for Eigenvalues*. Tech. Note BN-183. College Park: University of Maryland, Inst. for Fluid Dynamics and Applied Mathematics.
- . 1974. *Variational Methods for Eigenvalue Approximation*. Philadelphia: SIAM.
- Weinstein, A. 1935. Sur la Stabilité des Plaques Encastrées. *C. R. Acad. Sci. Paris* 200:107–109.
- Weyl, H. 1912 (submitted for publication in 1911). The laws of asymptotic distribution of the eigenvalues of linear partial differential equations. *Math. Ann.* 71:441–479.
- Whitehead, R. R. 1972. A numerical approach to nuclear shell-model calculations. *Nuclear Phys.* A182:290–300.
- Wiberg, T. 1973. *A Combined Lanczos and Conjugate Gradient Method for the Eigenvalue Problem of Large Sparse Matrices*. Tech. Report UMIN-4273. Umeå, Sweden: Dept. of Information Processing, S-90187.
- Wielandt, H. 1967. *Topics in the Theory of Matrices*. (Lecture notes prepared by R. R. Meyer.) Madison: University of Wisconsin Press.

- Wilkinson, J. H. 1960. Householder's method for the solution of the algebraic eigenproblem. *Comput. J.* 3:23-27.
- . 1962. Note on the quadratic convergence of the cyclic Jacobi process. *Numer. Math.* 4:296-300.
- . 1964. *Rounding Errors in Algebraic Process*. Englewood Cliffs, N.J.: Prentice-Hall.
- Wilson, E. 1978. Private communication.
- Zeeman, E. C. 1976. Catastrophe theory. *Sci. Amer.* 234:65-83.
- Zimmerman, K. 1969. *On the Convergence of a Jacobi Process for Ordinary and Generalized Eigenvalue Problems*. Dissertation 4305. Zurich: Eidgenossische Technische Hochschule.

This page intentionally left blank

Index

- Accumulation of inner products, 37
Adjoint matrix, 4
 self-adjoint, 7
Adjugate matrix, 137
Agar, T. J. A., 337
Anderson, E., 41
Arithmetic, 27
- Bai, Z., 41
Band QL and QR, 185
Bandwidth, 21
 preservation, 153
Bargmann, V., 118, 200
Basis, 229
 nonorthogonal, 253
Bathé, K.-J., 57, 337, 354, 357, 368
Bauer, F. L., 337
 Bauer–Fike theorem, 18
Bender, C. F., 368
Bessel functions, 120
Bickley, W. G., 225
Bischof, C., 107
Bleher, J. H., 38
Boyle, J. M., 41
Brent, R. P., 57, 60
Browne, E. T., 55
Bus, J. C., 57
- Cancellation, 30
Cao, Z.-H., 254
- Cauchy, A., 202
 –Binet formula, 137
 –Schwarz inequality, 3, 346
 interlace theorem, 52, 54, 202
Cayley–Hamilton theorem, 124
Characteristic polynomial, 6, 54, 60,
 339
- Chatelin, F., 227
Chebyshev
 acceleration, 329
 polynomials, 371
Cholesky factorization, 12, 24, 106
Chordal metric, 342
Cline, A. K., 24
Clustered Ritz values, 239
Condition number, of eigenvalue, 16
Congruence transformation, 11, 51,
 93, 102, 339
- Corbato, F. J., 195
Corneil, D., 196, 200
Courant, R., 206
Courant–Fischer, *see* Minmax char-
acterization
Crandall, S. H., 85
Crawford, C. R., 348
Cullum, J., 23, 319, 321, 368
- Daniel, J. W., 118
Davidson, E. R., 320
Davis, C., 98, 244, 247, 260

- De Boor, C., 148
- Deflation, 87
- Dekker, T. J., 57, 188
- Demmel, J., 150, 200
- Dhillon, I. S., 72
- Divided difference, 57
- Dodson, D., 40
- Donath, W. E., 321, 368
- Dongarra, J. J., 41
- Druskin, V. L., 319
- Dubrulle, A. A., 97, 118
- Duff, I. S., 22, 46
- Eigenspace, 8
- Eigenvalue
 - definition, 5
 - hidden, 58
 - multiplicity, 7
- Eigenvector
 - definition, 5
 - duplicate, in Lanczos, 301
 - dwindling, 290
 - tridiagonals, 137
- EISPACK, 41, 68, 72
- Elementary matrix, 369
- Equivalence transformation, 339
- Ericsson, T., 364
- Error
 - analysis, 29, 30, 33, 48
 - backward, 29, 34, 35, 105, 106
 - bounds, 73, 201–227
 - equivalent perturbation, 103
 - forward, 30, 106
- Erxiong, J., 188
- Euclidean space, 2
- Faddeev, D. D., 60
- Faddeeva, V. N., 60
- Feng, Y., 188
- Fill-in, matrix, 45–47, 68
- Fischer, E., 206
- Fix, G., 352, 368
- Floating point binary arithmetic, 30
- Forsythe, G., 49, 69
- Fox, L., 92, 149
- Francis, J. G. F., 187
- Franklin, J. N., 368
- Gantmacher, F. R., 339, 367
- Gap error bounds, 224, 244
- Garbow, B. S., 41
- Gauss transformation, 45
- Gentleman, W. M., 118
- George, A., 46
- Gill, P. E., 118
- Givens, W., 55, 128, 142, 149
 - fast, scaled rotations, 108
 - rotation, 100, 109, 199
- Glauz, G., 178
- Golub, G. H., 24, 118, 136, 148, 149, 303, 305, 321, 368
- Graded matrix, 156, 176
- Gragg, W. B., 149
- Gram–Schmidt process, 106, 107
- Greenbaum, A., 319
- Grimes, R., 22, 320
- Hald, O. H., 148
- Hammarling, S., 109, 118
- Hari, V., 196
- Harrod, W. J., 149
- Heiberger, R., 352, 368
- Henrici, P., 196
- Hermitian matrix, 7
- Hilbert matrix, 88
- Hill, R. O., 140
- Hochstadt, H., 148
- Hoffman, W., 188
- Hotelling, H., 92

- Householder, A. S., 84, 92, 118, 149, 320
method, 107
- Ikebe, Y., 201
- Ill-disposed root, 342
- Inagaki, T., 201
- Inertia
definition, 11, 12
Sylvester's inertia theorem, 11, 14, 51
- Inner product
definition, 2, 18
space, 2, 18, 346
- Interlace theorems
Cauchy, 202
residual, 211
- Invariant norms, 15
- Invariant subspace, 10
- Inverse iteration, 62–73
- Inverse problems, 146
- Jacobi, C. G. J., 100, 118, 200
convergence, 193
generalized, 353
methods, 190
rotation, 100, 189, 190
- Jennings, A., 337, 368
- Jensen, P. S., 337
- Kahan, W., 85, 98, 149, 244, 247, 253, 260, 320
double secant, 57
eigenvalue bounds, 201, 227
neglect off-diagonal elements, 144
orthogonalization and roundoff, 115
RQI convergence theorem, 80
- Kaiser, H. F., 178
- Kaniel, S., 260, 286
error bounds, 269
- Kato, T., 98, 227, 260
- Knisherman, L. A., 319
- Knuth, D. E., 95
- Knyazev, A., 260
- Krylov, A. N., 285
subspace, 261
and polynomials, 265
and power method, 279
error bounds, 269
- Kublanovskaya, V. N., 187
- Kulisch, U., 38
- Lanczos, C., 320
assessing accuracy, 290
for (A, M) , 362
polynomials, 125
reduction to tridiagonal form, 123, 149
- LAPACK, 41, 72
- Lawson, C. R., 40
- LDU theorem, 44
- Lehmann, N. J., 216, 218, 226, 227
optimal intervals, 216
- Lemordant, J., 227
- Lewis, J. G., 22, 40, 320
- Li, Ch.-K., 16
- Li, R.-C., 16, 150, 254
- LINPACK, 40
- Liu, J. W., 46
- Mathias, R., 16
- McCormick, S. F., 368
- McEnteggert, P., 137
- McKeown, J. J., 337, 368
- Minmax characterization, 206
- Miyamoto, S., 201
- Modi, J. J., 196, 200
- Moler, C. B., 41, 49, 69, 353, 368
- Monotonic residuals, 79
- Monotonicity theorems, 207

- Montgomery, C., 118, 200
- Murray, W., 118
- Neglect off-diagonal elements, 144
- Noe, T., 368
- Norms, 3, 15, 346
- Nour-Omid, B., 292
- Ortega, J. M., 178
- Orthogonal matrices, 93
 - errors in a sequence, 102
- Orthogonal polynomials, 134
- Orthogonal vectors, 3
- Orthogonalization
 - Gram-Schmidt, 106
 - re-, in Lanczos, 303
 - roundoff, 113
 - selective, in Lanczos, 305
- Ostrowski, A. M., 85
- Paige, C. C., 138, 149, 286, 298, 320
 - theorem, 295
- Parlett, B. N., 85, 118, 140, 150, 188, 292, 315, 319-321, 337, 344
- Pencil
 - matrix, 339
 - singular, 341
- Permutation matrix, 94
- Perturbation theory, 343
- Pipano, A., 368
- Platzman, G. W., 24
- Poole, W. G., 337
- Positive definite, 12
- Power method
 - convergence, 63
 - definition, 356
 - versus Krylov, 279
- Programs, 39-41
- Projectors, 8
- QR and QL
 - and inverse iteration, 156
 - convergence, 158, 165, 168
 - implementation, 171-187
- Quadratic forms, 11
- QZ algorithm, 353
- Rank, 11
- Rayleigh, Lord, 260
 - iteration, 75-85
 - quotients, 12, 18, 61, 73, 357
 - residual, 237
 - Ritz, 234-236, 360
- Reduction to tridiagonal, 125
- Reflections, 96
- Reinsch, C. H., 40, 41, 131, 173, 175, 177, 178, 185
- Residual error bounds, 73-75
- Residual interlace theorem, 211
- Residual matrix, 232
- Ritz, W., 260
 - interior approximation, 250
 - values and vectors, 234-242
- Ritzit, 328
- Rotations
 - fast, scaled, 108
 - plane, 99
- Roundoff error, 27
- Ruhe, A., 318, 321, 325, 359, 364, 368
- Rump, S. M., 38
- Rutishauser, H. R., 178, 188, 190, 194, 324, 337
- Saad, Y., 177, 286
 - error bounds, 269
- Sack, R. A., 178
- Saunders, M. A., 118
- Schönhage, A., 197, 200
- Schur complement, 45

- Schwarz, H. R., 131, 368
Scott, D. S., 294, 299, 312, 315, 321
Secant method, 56
Sectioning, 336
Separation theorems, *see* Interlace theorems
Shavitt, I., 368
Signature, 11
Similarity transformation, 6, 91
Simon, H., 319, 320
Simultaneous diagonalization of forms, 343
Simultaneous iteration, 323, 364
Slapričar, I., 200
Slicing the spectrum, 50
SOR methods, 359
Sorensen, D. C., 321
Sparse matrix, 21, 46
Spectral theorem, 8
Spectrum, 7
 slicer, 52
Stewart, G. W., 102, 149, 176, 178,
 342, 353, 368
Strakos, Z., 319
Strang, G., 368
Strutt, J. W., *see* Rayleigh, Lord
Sturm sequences, 54, 141
Subspace
 angles, between, 247
 definition, 229
 eigenspace, 232
 invariant, 10, 232
 iteration, 323, 364
 convergence, 331
 representation, 229
 spanning set, 229
Szegő, G., 149
Temple, G., 85, 225, 227, 260
Thompson, R. C., 137
Traub, J. F., 60, 188
Triangular factorization, 43
 error analysis, 48
Tridiagonal form, 119–150
 partial reduction to, 282
Uhlig, F., 344
Underwood, R., 303, 305, 321, 368
Unitary matrix, 5
Unreduced tridiagonal, 119
van der Vorst, H. A., 320, 368
van Kats, J. M., 320, 368
van Loan, C. F., 107
Vector iterations, 61
Veselić, K., 196, 200
von Neumann, J., 118, 200
Weaver, W., 368
Weinberger, H. F., 221, 227
Weinstein, A., 227, 260
Welsch, J. H., 136, 149, 178
Weyl, H., 208
 monotonicity theorem, 54
Whitehead, R. R., 320
Wiberg, T., 368
Wielandt, H., 224, 227
 –Hoffman inequality, 16, 144
Wilkinson, J. H., 16, 30, 40, 41, 55,
 60, 84, 92, 118, 131, 134,
 142, 149, 173, 175, 177, 178,
 185, 197, 200, 303, 305, 320,
 368

Wilson, E., 57, 337, 354, 357, 365,
368

Xie, J.-J., 254

Yoshida, D. M., 368

Zeeman, E. C., 30

Zhenye, Z., 188

Zimmerman, K., 354, 368