

Preface

Eigenvalue problems are ubiquitous in engineering and science. We assume that the reader already knows this and agrees with us that the study of numerical algorithms for solving eigenvalue problems is a worthwhile endeavor.

Prerequisites

We assume that our reader already has some experience with matrix computations and is familiar with the standard notation and terminology of the subject. We also assume knowledge of the basic concepts of linear algebra. A reader who has covered most of [73] or [72] will be in a good position to appreciate this monograph. These books, which were not selected at random, are just two of many from which one could have studied. Some others are the books by Trefethen and Bau [59], Golub and Van Loan [39], Demmel [29], and Björck [18].

This Book

This monograph is about a class of methods for solving matrix eigenvalue problems. Of course the methods are also useful for computing related objects such as eigenvectors and invariant subspaces. We will introduce new algorithms along the way, but we are also advocating a new way of viewing and implementing existing algorithms, notably Francis's implicitly shifted QR algorithm [36].

Our first message is that if we want to compute the eigenvalues of a matrix A , it is often advantageous to store A in QR-decomposed form. That is, we write $A = QR$, where Q is unitary and R is upper triangular, and we store Q and R instead of A . This may appear to be an inefficient approach but, as we shall see, it often is not. Most matrices that arise in applications have some special structures, and these often imply special structures for the factors Q and R . For example, if A is upper Hessenberg, then Q is also upper Hessenberg, and it follows from this that Q can be stored very compactly. As another example, suppose A is unitary. Then $Q = A$, and R is the identity matrix, so we don't have to store R at all.

Every matrix can be transformed to upper Hessenberg form by a unitary similarity transformation. We will study this and related transformations in detail in Chapter 6, but for the early chapters of the book we will simply take the transformation for granted; we will assume that A is already in Hessenberg form. Thus we consider an arbitrary upper Hessenberg matrix in QR-decomposed form and show how to compute its eigenvalues. Our method proceeds by a sequence of similarity transformations that drive the matrix toward upper triangular form. Once the matrix is triangular, the eigenvalues can be read from the main diag-

onal. In fact our method is just a new implementation of Francis's implicitly shifted QR algorithm. The storage space requirement is $O(n^2)$ because we must store the upper triangular matrix R , and the flop count is $O(n^3)$.

Once we know how to handle general matrices, we consider how the procedure can be simplified in special cases. The easiest is the unitary case, where $R = I$. This results in an immediate reduction in the storage requirement to $O(n)$ and a corresponding reduction of the computational cost to $O(n^2)$ flops. A similar case is that of a companion matrix, which is both upper Hessenberg and unitary-plus-rank-one. This results in an R that is unitary-plus-rank-one. Once we have figured out how to store R using only $O(n)$ storage, we again get an algorithm that runs in $O(n^2)$ flops. The unitary case is old [42], but our companion algorithm is new [7].

A structure that arises frequently in eigenvalue problems is symmetry: $A = A^T$. This very important structure does not translate into any obvious structure for the factors Q and R , so it does not fit into our framework in an obvious way. In Section 4.4 we show that the symmetric problem can be solved using our methodology: we turn it into a unitary problem by a Cayley transform [10]. We did not seriously expect that this approach would be faster than all of the many other existing methods for the symmetric eigenvalue problem [73, Section 7.2], but we were pleased to find that it is not conspicuously slow.

Our solution to the symmetric problem serves as a stepping stone to the solution of the symmetric-plus-rank-one problem, which includes comrade and colleague matrices [14] as important special cases. If a polynomial p is presented as a linear combination of Chebyshev or Legendre polynomials, for example, the coefficients can be placed into a comrade matrix with eigenvalues equal to the zeros of p . A Cayley transform turns the symmetric-plus-rank-one problem into a unitary-plus-rank-one problem, which we can solve by our fast companion solver. Our $O(n^2)$ algorithm gives us a fast way to compute the zeros of polynomials expressed in terms of these classic orthogonal polynomial bases.

We also study the generalized eigenvalue problem, for which the Moler–Stewart QZ algorithm is the appropriate variant of Francis's algorithm. We show how to apply this to a companion pencil using the same approach as for the companion matrix, but utilizing two unitary-plus-rank-one upper triangular matrices instead of one [4]. We then extend this methodology to matrix polynomial eigenvalue problems. A block companion pencil is formed and then factored into a large number of unitary-plus-rank-one factors. The resulting algorithm is advantageous when the degree of the matrix polynomial is large [6].

The final chapter discusses the reduction to Hessenberg form and generalizations of Hessenberg form. We introduce generalizations of Francis's algorithm that can be applied to these generalized Hessenberg forms [64].

This monograph is a summary and report on a research project that the five of us (in various combinations) have been working on for several years now. Included within these covers is material from a large number of recent sources, including [4, 5, 6, 7, 8, 9, 10, 12, 54, 62, 63, 64, 74]. We have also included some material that has not yet been submitted for publication. This book exists because the senior author decided that it would be worthwhile to present a compact and unified treatment of our findings in a single volume. The actual writing was done by the senior author, who wanted to ensure uniformity of style and viewpoint (and, one could also say, prejudices). But the senior author is only the scribe, who (of course) benefitted from substantial feedback from the other

authors. Moreover, the book would never have come into existence without the work of a team over the course of years. We are excited about the outcome of our research, and we are pleased to share it with you.

The project is not finished by any means, so a second edition of this book might appear at some time in the future.

This Book's Title

Francis's algorithm, in its standard form, is a bulge-chasing algorithm. Each iteration begins with a matrix in upper Hessenberg form. A tiny bulge is created in the upper Hessenberg form, and that bulge is then chased away. In our reformulation of the algorithm, we do not chase a bulge. Instead of introducing a bulge, we introduce an extraneous *core transformation*. We then chase that core transformation through the matrix until it disappears via a *fusion* operation. Thus our algorithms are core-chasing algorithms. Core transformations, which are defined in Chapter 1, are familiar objects. For example, Givens rotations are core transformations.

Software

Fortran implementations of many of the algorithms described in this monograph are collected in the package EISCOR, which can be accessed at

www.siam.org/books/fa13

Acknowledgment

We thank three anonymous reviewers who read a preliminary version of the book and provided valuable feedback.