POLYNOMIAL ACCELERATION FOR LARGE NONSYMMETRIC EIGENPROBLEMS

加速付反復固有値解法

by

Akira Nishida

西田 晃

A Thesis

Submitted to

The Graduate School of

The University of Tokyo

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Science

December, 1997

ABSTRACT

In this thesis, we propose a highly efficient accelerating method for the restarted Arnoldi iteration to compute the eigenvalues of a large nonsymmetric matrix. Its effectiveness is proved by various numerical experiments and comparisons with other approaches. Several new results on the characteristics of the polynomial acceleration are also reported.

The Arnoldi iteration has been the most popular method for nonsymmetric large eigenproblems. Its defect of increasing computational complexity per iteration step was improved with the explicitly restarting technique, by which the dimensions of the Krylov subspaces are kept modest. Although the restarted Arnoldi iteration is a quite effective approach, the dimension of the subspace becomes inevitably large, especially when the required eigenvalues are clustered. Furthermore, it favors the convergence on the envelope of the spectrum. In this paper, we seek a polynomial such that the components in the direction of unwanted eigenvectors are damped, using the approximate eigensolution estimates obtained in the previous step. Although the Chebyshev acceleration, which defines an elliptic area in the complex plane containing the unwanted Ritz values to be damped, can be combined with the original explicitly restarted Arnoldi iteration, it is restrictive and ineffective if the shape of the convex hull of the unwanted eigenvalues bears little resemblance with an ellipse. In our study, an accelerating polynomial is chosen to minimize an L_2 norm of the polynomial on the boundary of the convex hull with respect to some suitable weight function. A new simple algorithm is proposed for the efficient computation of the mini-max polynomial to accelerate the convergence of the Arnoldi iteration.

From the numerical results, we can derive the strong dependency of the polynomial acceleration on the distribution of spectrum, which proves the better performance of our algorithm than the ellipse-based methods, in the cases where the moduli of the wanted eigenvalues are considerably larger than those of the unwanted eigenvalues, and the faster convergence than those of all the other approaches, especially with the non-clustered distribution of the spectrum.

Finally, we propose a new parallelization technique for the nonsymmetric double shifted QR algorithm with perfect load balance and uninterrupted pipelining on distributed memory parallel architectures, which is strongly required from the viewpoint of complexity of the Arnoldi iteration. Its parallel efficiency is much higher than those reported in other papers.

Table of Contents

Chapter 1

Introduction

1.1 Overview

In the last few years, there have been great progress in the developments of the methods for the standard eigenproblem. Arnoldi's method, which have the disadvantage of increasing computational complexity per iteration step, was improved with the restarting technique, by which the dimensions of the Krylov subspaces is kept modest. Although the Arnoldi iteration is a considerably effective solution, the dimension of the subspace becomes excessively large, especially when the required eigenvalues are clustered. Furthermore, outer eigenvalues on the envelope of the spectrum show faster convergence. This difficulty has been overcome by using the polynomial acceleration technique, which is an extension of the similar technique for symmetric matrices. In the nonsymmetric case, we consider the distribution of the eigenvalues in the complex plane. Suppose $A \in \mathbf{R}^{n \times n}$ is a diagonalizable matrix with eigensolutions (u_j, λ_j) for $j = 1, ..., n$. Letting $p(\cdot)$ be some polynomial, the current starting vector x_0 can be expanded as $p(A)x_0 = c_1 p(\lambda_1)u_1 + \cdots + c_n p(\lambda_n)u_n$ in terms of the basis of eigenvectors. Then if we assume that the eigenvalues are ordered so that the wanted k ones are located at the beginning of the expansion, we seek a polynomial such that $\max_{i=k+1,...,n} |p(\lambda_i)| < \min_{i=1,...,k} |p(\lambda_i)|$ holds. Acceleration techniques attempt to improve the restarted Arnoldi iteration by solving this min-max problem, and applying the accelerating polynomials to its restart vectors.

1.2 Projection Methods

Most algorithms for solving large eigenvalue problems employ a projection technique, an approximation of the exact eigenvector u by a vector $\tilde{u}$ belonging to some subspace K referred to as the subspace of approximants. The subspace iteration algorithm, which will be described in Chapter 3, is a block generalization of the power method and is the most simplest approach, although it is not competitive with other projection methods.

The Krylov subspace methods extract approximations from a subspace of the form

$$\mathsf{K}_m = \text{span}\{v, Av, A^2v, ..., A^{m-1}v\} \tag{1.1}$$

referred to as a Krylov subspace. Arnoldi's method and Lanczos' method are classified in the orthogonal projection methods, while the nonsymmetric Lanczos algorithm in the oblique projection method. Although it is an extension of the simple power method, the projection method turns out to be one of the most successful methods for extracting eigenvalues of large matrices.

Arnoldi's method is an orthogonal projection method onto K_m for general non-Hermitian matrices, which was introduced originally as a means of reducing a dense matrix into the Hessenberg form and later discovered that it leads to a valuable method for approximating the eigenvalues of large sparse matrices. Furthermore, if we are interested in only a few eigensolutions of A, we can restart the algorithm and avoid the difficulty of its high storage and computational requirement as m increases, that is, we compute the approximate eigenvectors and use it as an initial sequence of vectors for the next run of the Arnoldi's method.

1.3 Accelerating Techniques

Although these algorithms are attractive because of its simplicity, their convergence rate may be unacceptably slow in some cases. Polynomial accelerating techniques are useful tools for speeding up the convergence of these methods. A polynomial iteration takes the form of $x_k = p_k(A)x_0$ where p_k is a polynomial which is determined from some knowledge on the distribution of the eigenvalues of A. The polynomial p is selected to be optimal in some sense, and this leads to the use of Chebyshev polynomials.

Given a set of approximate eigenvalues of a nonsymmetric matrix A, a simple region which encloses the spectrum of A can be constructed in the complex plane. One of these ideas is to

use an ellipse that encloses an approximate convex hull of the spectrum. If we consider an ellipse centered at δ with focal distance ϑ, the shifted and scaled Chebyshev polynomials defined by

$$t_k(\zeta) = \frac{T_k(\frac{\delta-\zeta}{\delta})}{T_k(\frac{\delta}{\vartheta})} \tag{1.2}$$

are nearly optimal.

1.4 Least Squares Arnoldi Method

The choice of ellipses as enclosing regions in Chebyshev acceleration, however, may be overly restrictive and ineffective if the shape of the convex hull of the unwanted eigenvalues bears little resemblance to an ellipse. In this thesis, we propose a simple method for determining the accelerating polynomial, which is chosen so as to minimize an L_2 norm of the polynomial p on the boundary of the convex hull of the unwanted eigenvalues with respect to some suitable weight function w. The only restriction with this technique is that the degree of the polynomial is limited because of cost and storage requirements. This can be overcome by compounding low degree polynomials, and the stability of the computation is enhanced by employing a Chebyshev basis. We prove the optimality of our method and confirm its validity by various experiments.

From the results on the complexity of our method, we can see that the number of floating point operations rapidly increases with the size of the subspace dimension m, which indicates that we need to take m as small as possible if we want to avoid QR to become a bottleneck. In this thesis, we propose a new data mapping method with best possible scalability for the parallelization of the double shifted QR algorithm, in which the loads including the lookahead step are balanced, and the computations are pipelined by hiding the communication latency. Our implementation on a Fujitsu AP1000+ attains parallel efficiency higher than 90% without matrix size reduction, and 70–80% for the whole process including the matrix size reduction.

1.5 Organization

This thesis is constructed by the following topics: Chapter 2 describes the basic concepts which formulate the algebraic eigenvalue problem. Preliminary definitions are also given in this chapter. Chapter 3 introduces some orthodox approaches to solve the eigenproblem, which can be considered as the derivatives of the concept of vector iterations. In Chapter 4, we will outline the fundamental results on the Arnoldi process, obtained by the preceding researches. Chapter 5 will be devoted to

the polynomial acceleration. We will begin with the basic idea of the polynomial acceleration, which was originally developed for solving linear systems, and show that they can be more effectively applied to eigenproblems. Chapter 6 describes the details of our method. We introduce the idea of acceleration using the least squares polynomials, and propose an efficient method for determining its parameters. Our algorithm is evaluated by sufficient number of test problems from various applicational fields in Chapter 7. We will also discuss some inherent problems, lying on the more effective computation. Our current goal is the integration of iterative eigensolver and parallel direct methods for nonsymmetric matrices. In this thesis, it is partially implemented in Chapter 7. Full integration requires more thorough research but it will be realized in the very near future. The concluding remarks in Chapter 8 will include our perspectives on the problem.

Chapter 2

Basic Facts in Linear Algebra

2.1 Definitions

2.1.1 Matrices and Eigenvalues

Definition 2.1. 1 (General Definitions) *By an m by n matrix, we mean an array of mn elements a_{ik}, where $i = 1, ..., m$ and $k = 1, ..., n$, arranged in a rectangular form*

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}. \tag{2.1}$$

An n by n matrix is called a square matrix of degree n. Each horizontal n-tuple of an $m \times n$ matrix is called a row, and each vertical m-tuple is called a column of the matrix. A square matrix is called a diagonal matrix if $a_{ik} = 0$ for $i \neq k$. An $n \times n$ matrix whose (i, k)-element is equal to δ_{ik} is called the identity matrix of degree n, where δ_{ik} is the Kronecker delta. A $1 \times n$ matrix $(a_1, a_2, ..., a_n)$ is called a row vector of dimension n, and an $m \times 1$ matrix

$$\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \tag{2.2}$$

is called a column vector of dimension m.

Definition 2.1. 2 *Let A be a square matrix. If there exists a matrix A^{-1} such that $AA^{-1} = A^{-1}A = I$, then A^{-1} is called the inverse matrix of A, and A is called a nonsingular matrix.*

Definition 2.1. 3 *Let $A = (a_{ik})$ be an $m \times n$ matrix. Then the $n \times m$ matrix (b_{ik}) such that $b_{ik} = a_{ki}$ for all i and k is called the transposed matrix of A and is denoted by A^T. Let $A = (a_{ik})$ be a square matrix with elements in the complex number field $\mathbf{C}$. Then the adjoint matrix A^H of A is the transposed conjugate $\bar{A}^T = (\bar{a}_{ki})$.*

Definition 2.1. 4 *A complex scalar λ is called an eigenvalue of the square matrix A if a nonzero vector u of a complex vector space $\mathbf{C}^n$ exists such that $Au = \lambda u$. The vector u is called an eigenvector of A associated with λ, and the pair (u, λ) an eigensolution. The set of all the eigenvalues of A is called the spectrum of A and is denoted by $\sigma(A)$.*

Proposition 2.1. 1 *If λ is an eigenvector of A, then $\bar{\lambda}$ is an eigenvalue of $A^H = \bar{A}^T$. An eigenvector v of A^H corresponding to the eigenvalue $\bar{\lambda}$ is called a left eigenvector of A.*

Definition 2.1. 5 *The polynomial*

$$\psi(\lambda) = \det(A - \lambda I) \tag{2.3}$$

is called the characteristic polynomial of A. The equation $\psi(\lambda) = \det(A - \lambda I)$ is called the characteristic equation of A.

The maximum absolute value of the eigenvalue of A is called the spectral radius and denoted by $\rho(A)$.

2.1.2 Selfadjoint and Unitary Matrices

Definition 2.1. 6 *If $A = A^T$ holds, A is called symmetric. If $A = \bar{A}$ holds, A is called real. If $A = A^H = \bar{A}^T$ holds, then A is called Hermitian or selfadjoint.*

Definition 2.1. 7 *The innerproduct (u, v) of two vectors u and v are defined as*

$$(u, v) = u^H v = \bar{u}^T v = \sum \bar{u}_j v_j \tag{2.4}$$

Definition 2.1. 8 (Vector Norms) *The vector whose components are all 0 is called the zero vector and is denoted by the same symbol 0. A vector norm on $\mathbf{C}^n$ is a real-valued function which satisfies the following three conditions:*

1. $\| ku \| = |k| \, \| u \|, \ \forall x \in \mathbf{C}^n, \ \forall k \in \mathbf{C}$.

2. $\| u \| > 0$ *unless* $u = 0$; $\quad \| u \| = 0$ *implies* $u = 0$.

3. $\| u + v \| \leq \| u \| + \| v \|, \quad \forall u, v \in \mathbf{C}^n$.

The Euclidean norm of a complex vector $u \in \mathbf{C}^n$ defined by

$$\| u \|_2 = (u, u)^{\frac{1}{2}} \tag{2.5}$$

is a special case of the Hölder norms

$$\| u \|_p = \left(\sum_{i=1}^{n} |u_i|^p \right)^{\frac{1}{p}}. \tag{2.6}$$

We define a special set of matrix norms for a general matrix A in $\mathbf{C}^{n \times m}$ by

$$\| A \|_{pq} = \max_{x \in \mathbf{C}^n, x \neq 0} \frac{|Ax|_p}{|x|_q} \tag{2.7}$$

Definition 2.1. 9 (Matrix Norms) *The matrix whose components are all* 0 *is called the zero matrix and is denoted by the symbol* O. *A matrix norm on* $\mathbf{C}^{n \times m}$ *is a real-valued function which satisfies the following three conditions:*

1. $\| kA \| = |k| \, \| A \|, \ \forall x \in \mathbf{C}^{n \times m}, \ \forall k \in \mathbf{C}$.

2. $\| A \| > 0$ *unless* $A = O$; $\quad \| A \| = 0$ *implies* $A = O$.

3. $\| A + B \| \leq \| A \| + \| B \|, \quad \forall A, B \in \mathbf{C}^{n \times m}$.

Definition 2.1. 10 *A is said to be orthogonal if A is real and* $A^T A = I$ *holds, and unitary if* $A^H A = I$ *holds.*

2.1.3 Subspaces

Definition 2.1. 11 *The range of the mapping* $A : \mathbf{R}^m \to \mathbf{R}^n$ *is defined by*

$$\mathsf{R}(A) = \{ Ax | x \in \mathbf{C}^m \}, \tag{2.8}$$

and its kernel or null space by

$$\mathsf{N}(A) = \{ x \in \mathbf{C}^m | Ax = 0 \}. \tag{2.9}$$

The rank of a matrix A is equal to the dimension of its range, i.e., to the number of linearly independent columns. A subspace S is said to be *invariant* under a matrix A if $A\mathsf{S} \subset \mathsf{S}$. The subspace $\mathsf{N}(A - \lambda I)$, which is invariant under A, is called the *eigenspace* associated with λ.

2.1.4 Canonical Forms of Matrices

Definition 2.1. 12 (Similarity Transformation) *Two matrices A and XAX^{-1}, where X is a nonsingular matrix, is said to be similar. The mapping $A \rightarrow XAX^{-1}$ is called a similarity transformation.*

Definition 2.1. 13 *An eigenvalue λ of a matrix A is said to have the algebraic multiplicity μ, if it is a root of multiplicity μ of the characteristic polynomial of A. The eigenvalue of algebraic multiplicity one is called simple. A nonsimple eigenvalue is called multiple.*

Definition 2.1. 14 *An eigenvalue λ of A is said to have the geometric multiplicity γ, if the maximum number of independent eigenvalues associated with λ is γ.*

For the proofs of the following two theorems, see Chatelin [10] and Halmos [26].

Lemma 2.1. 1 *For every integer l and each eigenvalue λ_i, we have*

$$\mathsf{N}(A - \lambda_i I)^{l+1} \subset \mathsf{N}(A - \lambda_i I)^{l}. \tag{2.10}$$

In a finite dimensional space, there is a smallest integer l_i such that

$$\mathsf{N}(A - \lambda_i I)^{l_i+1} = \mathsf{N}(A - \lambda_i I)^{l_i}. \tag{2.11}$$

Here, l_i is called the index of λ_i. The subspace $\mathsf{M}_i = \mathsf{N}(A - \lambda_i I)^{l_i}$ is invariant under A and the space $\mathbf{C}^n$ is the direct sum of the subspace M_i's for $i = 1, ..., p$. $\dim(\mathsf{M}_i)$ is denoted by m_i.

Theorem 2.1. 1 (Jordan Canonical Form) *Any square matrix A has a nonsingular matrix X that reduces the matrix to a block diagonal matrix called the Jordan canonical form:*

$$J = X^{-1}AX = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_p \end{pmatrix}, \tag{2.12}$$

where each submatrix J_k corresponds to the subspace M_i associated with the distinct eigenvalue λ_i and is of size m_i. Each of the blocks has itself a block diagonal structure:

$$J_i = \begin{pmatrix} J_{i1} & & & \\ & J_{i2} & & \\ & & \ddots & \\ & & & J_{i\gamma_i} \end{pmatrix}, \quad \textit{with} \quad J_{ik} = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{pmatrix}. \tag{2.13}$$

Each sub-block, refereed to as the Jordan block, is a upper bidiagonal matrix of size not exceeding l_i, and corresponds to a different eigenvector associated with the eigenvalue λ_i.

Each vector can be written uniquely as

$$x = x_1 + x_2 + \cdots + x_p \quad \text{where } x_i \in \mathsf{M}_i, \tag{2.14}$$

and the linear transformation

$$P_i : x \to x_i \tag{2.15}$$

is a projector onto M_i. The family of projectors P_i for $i = 1, ..., p$ satisfies the following properties:

$$\mathsf{R}(P_i) = \mathsf{M}_i, \tag{2.16}$$

$$P_i P_j = P_j P_i = 0, \quad i \neq j, \tag{2.17}$$

$$\sum_{i=1}^{p} P_i = I. \tag{2.18}$$

Theorem 2.1. 2 *Every matrix admits the decomposition*

$$A = \sum_{i=1}^{p} (\lambda_i P_i + D_i) \tag{2.19}$$

where $D_i = (A - \lambda_i I)P_i$ is a nilpotent operator of index l_i, i.e., $D_i^{l_i} = 0$.

Proof. From (2.19), we have

$$AP_i = \lambda_i P_i + D_i, \quad i = 1, 2, ..., p, \tag{2.20}$$

which are summed up into

$$A \sum_{i=1}^{p} P_i = A = \sum_{i=1}^{p} (\lambda_i P_i + D_i). \tag{2.21}$$

□

Theorem 2.1. 3 (Schur Canonical Form) *Any square matrix A has a unitary matrix Q such that*

$$Q^H A Q = R \tag{2.22}$$

is upper triangular.

Proof. The proof is by induction over the dimension of A. □

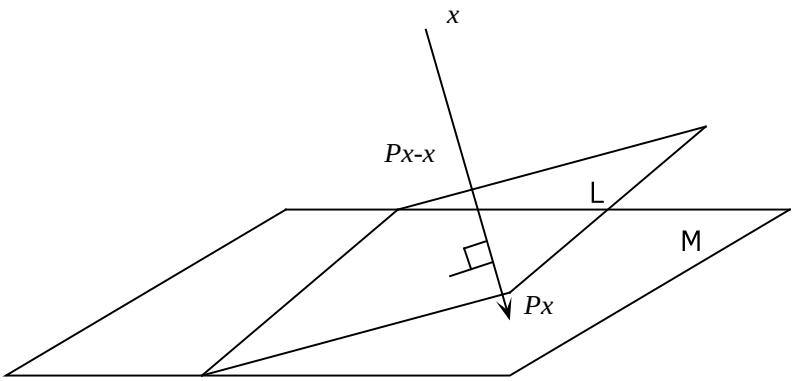

Figure 2.1. Oblique projection of x

2.1.5 Projection Operators

2.1.5.1 Range and Null Space

Definition 2.1. 15 *A linear mapping from* $\mathbf{C}^n$ *to itself, i.e., such that*

$$P^2 = P, \tag{2.23}$$

is called a projector.

If P is a projector, so is $(I - P)$, and the relation

$$\mathsf{N}(P) = \mathsf{R}(I - P) \tag{2.24}$$

holds. The space $\mathbf{C}^n$ is decomposed as the direct sum

$$\mathbf{C}^n = \mathsf{N}(P) \oplus \mathsf{R}(P). \tag{2.25}$$

Conversely, every pair of subspaces M and S which forms a direct sum of $\mathbf{C}^n$ defines a unique projector such that $\mathsf{R}(P) = \mathsf{M}$ and $\mathsf{N}(P) = \mathsf{S}$. For any x, the vector Px satisfies the conditions

$$Px \in \mathsf{M} \quad \text{and} \quad x - Px \in \mathsf{S}. \tag{2.26}$$

These relations define a projector P onto M and orthogonal to the subspace $\mathsf{L} = \mathsf{S}^{\perp}$.

2.1.5.2 Matrix Representations

The matrix representation of a projector is obtained from two bases, a basis $V = [v_1, ..., v_m]$ for the subspace M and and $W = [w_1, ..., w_m]$ for L. Denoting by Vy the representation in the V basis

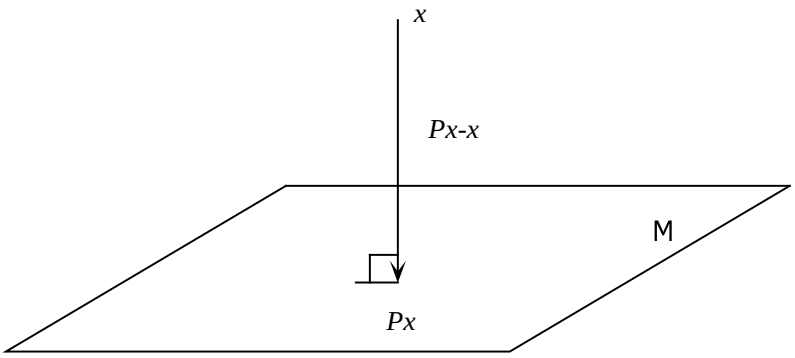

Figure 2.2. Orthogonal projection of x

of Px, the constraint $x - Px \perp \mathsf{L}$ is equivalent to the condition

$$W^T(x - Vy) = 0, \tag{2.27}$$

which derives the relation

$$P = V(W^H V)^{-1} W^H. \tag{2.28}$$

If the two bases are biorthogonal, i.e. $(v_i, w_j) = \delta_{ij}$, we have $y = W^H x$, which leads to the matrix representation of the projector P,

$$P = VW^H. \tag{2.29}$$

2.1.5.3 Orthogonal and Oblique Projectors

Definition 2.1. 16 *The projector P is said to be orthogonal onto M in the case when the subspace L is equal to M, i.e., when*

$$\mathsf{N}(P) = \mathsf{R}(P)^{\perp}, \tag{2.30}$$

and oblique in the nonorthogonal case.

For any vector x, an orthogonal projector is defined through the condition

$$Px \in \mathsf{M} \quad \text{and} \quad (I - P)x \perp \mathsf{M}. \tag{2.31}$$

A projector is orthogonal if and only if it is Hermitian.

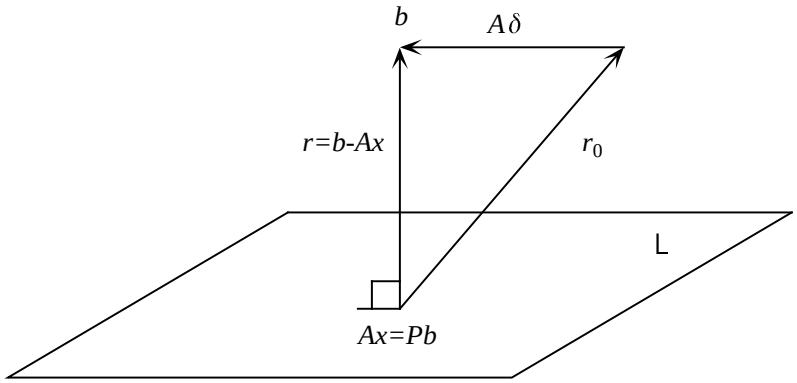

Figure 2.3. Projection of x

2.2 Projection Methods

2.2.1 Projection Methods for Linear Systems

Consider a linear system of equations having n unknowns. If L is the m-dimensional subspace of the candidate approximants to the problem, m constraints must be imposed to extract an approximation. A projection technique for the linear system is defined as a process to find an approximate solution $\tilde{x}$ of the linear system $Ax = b$,

$$\text{Find } \tilde{x} \in x_0 + \mathsf{K} \text{ such that } b - A\tilde{x} \perp \mathsf{L}, \tag{2.32}$$

where x_0 is an initial guess to the solution. If $\tilde{x}$ is written in the form $\tilde{x} = x_0 + \delta$, the approximate solution can be defined as

$$\tilde{x} = x_0 + \delta,\ \delta \in \mathsf{K}, \tag{2.33}$$

$$(r_0 - A\delta, w) = 0,\ \forall w \in \mathsf{L}. \tag{2.34}$$

Most standard techniques use a succession of such projections, in which a new projection steps uses a new pair of subspace K and L, with an initial guess x_0 obtained from the previous projection step.

2.2.1.1 Matrix Representation

Let $V = [v_1, ..., x_m]$ and $W = [w_1, ..., w_m]$ be $n \times m$ matrices, whose column-vectors form the bases of K and L respectively. If the approximate solution is written as

$$\tilde{x} = x_0 + Vy, \tag{2.35}$$

the orthogonality condition leads to the following system for the vector y:

$$W^T AVy = W^T r_0. \tag{2.36}$$

If the matrix $W^T AV$ is nonsingular, we have

$$\tilde{x} = x_0 + V(W^T AV)^{-1} W^T r_0. \tag{2.37}$$

2.2.1.2 Krylov Subspace Methods

A Krylov subspace method for linear systems is a method for which the subspace K_m is the Krylov subspace of dimension m

$$\mathsf{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, ..., A^{m-1} r_0\}, \tag{2.38}$$

where $r_0 = b - Ax_0$. The approximations obtained from a Krylov subspace method are of the form

$$A^{-1} b \approx x_m = x_0 + q_{m-1}(A) r_0, \tag{2.39}$$

in which q_{m-1} is a certain polynomial of degree $m - 1$.

The choice of L_m will have an important effect on the iterative technique. The first is simply $\mathsf{L}_m = \mathsf{K}_m$ and the minimum-residual variation $\mathsf{L}_m = A\mathsf{K}_m$. The second class of methods is based on defining $\mathsf{L}_m = \mathsf{K}_m(A^T, r_0)$.

2.2.1.3 Arnoldi's Method for Linear Systems

We now consider an orthogonal projection method as defined before, which takes $\mathsf{L} = \mathsf{K} = \mathsf{K}_m(A, r_0)$, in which $r_0 = b - Ax_0$. If Arnoldi vector $v_1 = r_0 / \parallel r_0 \parallel_2$ in Arnoldi's method, and set $\beta = \parallel r_0 \parallel_2$, then

$$V_m^T AV_m = H_m \tag{2.40}$$

and

$$V_m^T r_0 = V_m^T (\beta v_1) = \beta e_1. \tag{2.41}$$

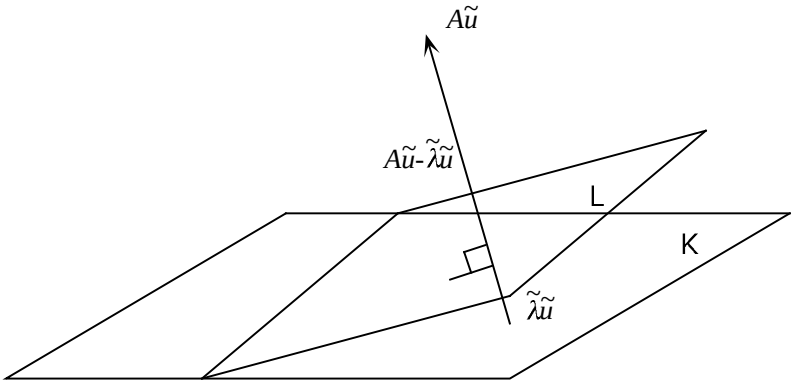

Figure 2.4. Projection Methods for Eigenproblems

As a result, the approximate solution using the above m-dimensional subspaces is given by

$$x_m = x_0 + V_m y_m, \tag{2.42}$$

$$y_m = H_m^{-1}(\beta e_1). \tag{2.43}$$

2.2.2 Projection Methods for Eigenvalue Problems

A projection method for eigenvalue problems is defined as a technique to approximate the desired eigenvector u by a vector $\tilde{u}$, belonging to a subspace K, by imposing the above Petrov-Galerkin condition that the residual vector $\tilde{u}$ be orthogonal to another subspace L.

2.2.2.1 Orthogonal Projection Methods

Let K be an m-dimensional subspace of $\mathbf{C}^n$ and consider an eigenvalue problem:

$$\text{Find } u \in \mathbf{C}^n \text{ and } \lambda \in \mathbf{C} \text{ such that } Au = \lambda u. \tag{2.44}$$

In an orthogonal projection technique onto the subspace K, the following condition is satisfied,

$$A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathsf{K}, \tag{2.45}$$

for an approximate eigenpair $\tilde{\lambda}$ and $\tilde{u}$.

Let $V = [v_1, ..., v_m]$ be $n \times m$ matrix, whose column-vectors form the base of K. If the approximate eigenvector is written as

$$\tilde{u} = Vy, \tag{2.46}$$

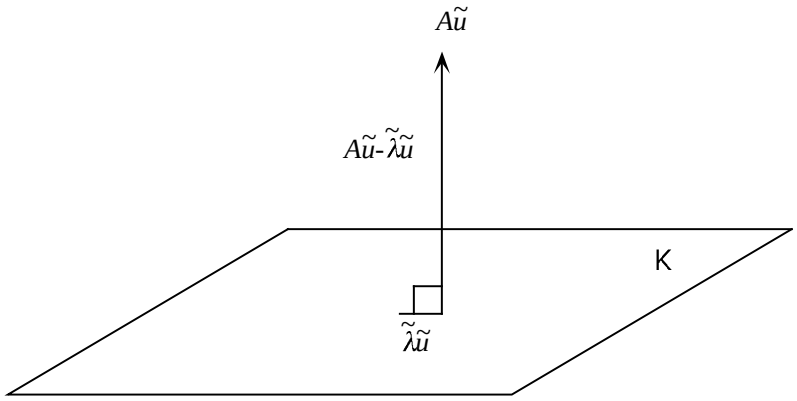

Figure 2.5. Orthogonal Projection Methods

the Galerkin condition becomes

$$(AVy - \tilde{\lambda}Vy, v_j) = 0, \quad j = 1, ..., m. \tag{2.47}$$

Then y and $\tilde{\lambda}$ satisfy

$$By = \tilde{\lambda}y \quad \text{with} \quad B = V^H AV. \tag{2.48}$$

B is the matrix representation of the linear transformation $A_m = PAP$, where we denote by P the orthogonal projector VV^H onto the subspace K. The Galerkin condition can be rewritten as

$$P(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0, \quad \tilde{\lambda} \in \mathbf{C}, \quad \tilde{u} \in \mathsf{K}, \tag{2.49}$$

i.e.,

$$PA\tilde{u} = \tilde{\lambda}\tilde{u}, \quad \tilde{\lambda} \in \mathbf{C}, \quad \tilde{u} \in \mathsf{K}. \tag{2.50}$$

It can be represented as

$$PAP\tilde{u} = \tilde{\lambda}\tilde{u}, \quad \tilde{\lambda} \in \mathbf{C}, \quad \tilde{u} \in \mathbf{C}^n, \tag{2.51}$$

considering the linear transformation of the original problem.

Chapter 3

Vector Iterations

For the outline of the methods referred to in this thesis, we begin with the eigensolvers for real non-Hermite matrices, classified under the concept of vector iterations. See also Golub and Van Loan [24], Saad [58], or Wilkinson [78] for further studies.

3.1 Single Vector Iterations

3.1.1 The Power Method

Suppose $A \in \mathbf{R}^{n\times n}$ and denote its eigenvalues by $\lambda_1, \lambda_2, \cdots, \lambda_n$.

Theorem 3.1. 1 *Suppose that the condition*

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0 \tag{3.1}$$

holds. If we denote by x_1 and y_1 the right and left eigenvectors of λ_1, the sequence

$$u \neq 0, \quad q_0 = \frac{u}{\| u \|_2}, \quad q_k = \frac{Aq_{k-1}}{\| Aq_{k-1} \|_2}, \quad k \geq 1 \tag{3.2}$$

is such that

$$|(Aq_k, q_k) - \lambda_1| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right) \tag{3.3}$$

if and only if $y_1^T u \neq 0$.

Proof. See Chatelin [10] or Wilkinson [78], for example. □

3.1.2 The Inverse Iteration

Let σ be an approximation to a simple eigenvalue λ, with right eigenvector x. If σ is not close to the eigenvalues of A, other than λ, then the dominant eigenvalue of $(A-\sigma I)^{-1}$ is $1/(\lambda-\sigma)$. This fact is exploited in the method of inverse iteration designed to compute the eigenvector x associated with λ whose approximation σ is known. We put

$$q_0 = \frac{u}{\| u \|_2}, \quad (A-\sigma I)z_k = q_{k-1}, \quad q_k = \frac{z_k}{\| z_k \|_2}, \quad k \geq 1. \tag{3.4}$$

3.2 Subspace Iteration Methods

Suppose a subspace $A^k S$ generated by r vectors $A^k u_1, \cdots, A^k u_r$. The r vectors tend to become parallel as $k \to \infty$. The subspace iteration method constructs an orthogonal basis Q_k of $A^k S$ as follows:

1. $U = Q_0 R_0$,
2. for $k \geq 1$, let $U_k = AQ_{k-1} = Q_k R_k$,

where the R_k are upper triangular matrices of degree r. Schmidt's orthogonalization $U_k = Q_k R_k$ can be carried out by the Householder method (see Appendix B).

The spectrum of the matrix $A_k = Q_k^H A Q_k$ of degree r converges to the r dominant eigenvalues of A.

3.3 The QR Algorithm

3.3.1 Principle of the QR Algorithm

The QR algorithm consists of the construction of a sequence $\{A_k\}$ of unitarily similar matrices:

$$A_1 = A = Q_1 R_1, \quad A_{k+1} = R_k Q_k = Q_{k+1} R_{k+1}, \quad k \geq 1 \tag{3.5}$$

where the Q_k are unitary and the R_k are upper triangular matrices. Since $R_k = Q_k^H A_k$, we have

$$A_{k+1} = Q_k^H A_k Q_k = (Q_1 \cdots Q_k)^H A_1 (Q_1 \cdots Q_k) \tag{3.6}$$

and

$$A^k = \hat{Q}_k \hat{R}_k, \tag{3.7}$$

where

$$\hat{Q}_k = Q_1 \cdots Q_k, \quad \hat{R}_k = R_k \cdots R_1. \tag{3.8}$$

The following results were proved by Francis [18].

Lemma 3.3. 1 *Let H be a Hessenberg matrix and let $H = QR$ be its Schmidt factorization. Then both Q and RQ are Hessenberg matrices.*

Theorem 3.3. 1 *Under the hypothesis that*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0, \tag{3.9}$$

the QR algorithm, when applied to an irreducible Hessenberg matrix, produces a sequence of unitary similar Hessenberg matrices which converges (modulo a unitary diagonal matrix) to an upper triangular matrix whose diagonal consists of the eigenvalues $\{\lambda_i\}_1^n$ in this order.

Theorem 3.3. 2 *On the assumption of the previous theorem, the QR algorithm, when applied to A, produces a sequence of unitary similar matrices whose limit form is an upper triangular matrix having $\{\lambda_i\}_1^n$ as its diagonal elements in this order, under the necessary and sufficient condition that the $n-1$ principal minors of X^{-1} are non-zero.*

3.3.2 Relation between the Subspace Iteration and the QR algorithm

The QR algorithm is equivalent to the subspace iteration applied to a full set of $r = n$ initial vectors $Q_0 = I$. We denote by $\hat{Q}_k$ the Q matrices of the subspace iteration, in order to distinguish them from those of the QR algorithm.

The subspace iteration is defined as

$$\hat{Q}_0 = I, \tag{3.10}$$

$$U_k = A\hat{Q}_{k-1}, \tag{3.11}$$

$$U_k = \hat{Q}_k R_k, \tag{3.12}$$

$$A_k = \hat{Q}_k^H A \hat{Q}_k, \tag{3.13}$$

while the QR algorithm by

$$A_0 = A, \tag{3.14}$$

$$A_{k-1} = Q_k R_k, \tag{3.15}$$

$$A_k = R_k Q_k, \tag{3.16}$$

where we define $\hat{Q}_k$ and $\hat{R}_k$ as

$$\hat{Q}_k = Q_1 Q_2 \cdots Q_k \tag{3.17}$$

and

$$\hat{R}_k = R_k R_{k-1} \cdots R_1. \tag{3.18}$$

Theorem 3.3. 3 *The above two processes generate identical sequences of matrices $\hat{R}_k$, $\hat{Q}_k$, and*

$$A^k = \hat{Q}_k \hat{R}_k, \tag{3.19}$$

with

$$A_k = \hat{Q}_k^H A \hat{Q}_k. \tag{3.20}$$

Proof. The proof is obtained by induction in k. The above processes suggest $A^0 = \hat{Q}_0 = \hat{R}_0 = I$ and $A_0 = A$, which satisfy the equations (3.19) and (3.20). For the case of $k \geq 1$, we need to prove (3.19) for the subspace iteration, and (3.19) and (3.20) for the QR method. As for the subspace iteration, the relation is verified by

$$A^k = A\hat{Q}_{k-1}\hat{R}_{k-1} = \hat{Q}_k R_k \hat{R}_{k-1} = \hat{Q}_k \hat{R}_k, \tag{3.21}$$

using the relation (3.11), (3.12), and (3.18).

As for the QR algorithm, we can verify the relation (3.19) by

$$A^k = A\hat{Q}_{k-1}\hat{R}_{k-1} = \hat{Q}_{k-1} A_{k-1} \hat{R}_{k-1} = \hat{Q}_k \hat{R}_k, \tag{3.22}$$

using hypothesis on (3.19) and (3.20), with the relations (3.15), (3.17), and (3.18), while the second by the sequence

$$A_k = Q_k^H A_{k-1} Q_k = \hat{Q}_k^H A \hat{Q}_k, \tag{3.23}$$

using the relation (3.15), (3.16), and the hypothesis on (3.20). □

3.3.3 The Shifted QR Algorithm

Suppose A is a real Hessenberg matrix. The QR algorithm with shifts of origin is described by

$$Q_s(A_s - k_s I) = R_s, \quad A_{s+1} = R_s Q_s^T + k_s I, \quad \text{giving} \quad A_{s+1} = Q_s A_s Q_s^T, \tag{3.24}$$

where Q_s is orthogonal, R_s is upper triangular and k_s is the shift of origin. A_{s+2} would be real even in the case where k_s is complex, if we perform the above transformation with shifts k_s and $\bar{k}_s$ respectively. Francis also proposed an economical method without using complex arithmetic. If B is non-singular and

$$BQ = QH, \tag{3.25}$$

with unitary Q and upper-Hessenberg H which has positive sub-diagonal elements, it can be shown that whole of Q and H are determined uniquely by the first column of Q. We have

$$A_s(Q_s^T Q_{s+1}^T) = (Q_s^T Q_{s+1}^T)A_{s+2} \tag{3.26}$$

and

$$(Q_s^T Q_{s+1}^T)(R_{s+1}R_s) = (A_s - k_s I)(A_s - k_{s+1}I). \tag{3.27}$$

We write

$$Q_{s+1}Q_s = Q, \quad R_{s+1}R_s = R, \quad (A_s - k_s I)(A_s - k_{s+1}I) = M. \tag{3.28}$$

Since $QM = R$ holds, Q is the matrix which triangularizes the matrix product M. The triangularization is performed by the Givens' method, i.e.,

$$Q = R_{n-1,n} \cdots R_{2,n} \cdots R_{2,3}R_{1,n} \cdots R_{1,3}R_{1,2}, \tag{3.29}$$

where R_{ij} is the plane rotation in the plane (i, j). Its first row is determined by $R_{1,n} \cdots R_{1,3}R_{1,2}$, while $R_{1,2}, R_{1,3}, \cdots, R_{1,n}$ are determined by the first column of M, whose nonzero elements are

$$x_1 = (a_{11} - k_1)(a_{11} - k_2) + a_{12}a_{21}, \quad y_1 = a_{21}(a_{11} - k_2) + (a_{22} - k_1)a_{21}, \quad z_1 = a_{32}a_{21}. \tag{3.30}$$

$R_{1,4}, \cdots, R_{1,n}$ are then the identity matrices. If we define C_1 by

$$R_{1,3}R_{1,2}A_s R_{1,2}^T R_{1,3}^T = C_1 \tag{3.31}$$

for some orthogonal matrix S_1 whose first row is $e_1 = (1, 0, \cdots, 0)^T$, we have

$$S_1^T C_1 S_1 = B, \tag{3.32}$$

where B is an upper Hessenberg matrix. B must be A_{s+2}, since the first row of $\tilde{Q}^T = S_1^T R_{1,3}R_{1,2}$, where $B = \tilde{Q}^T A_s \tilde{Q}$, is equal to the first row of Q^T. These computations can be performed more efficiently by $8n^2$ real multiplications.

Chapter 4

The Arnoldi Process

In this chapter, we present the basic ideas of the Arnoldi process, which is a variant of the Krylov subspace method. Section 4.1 derives some important properties. The iterative version of the Arnoldi process is described in Section 4.2.

4.1 Arnoldi's Method

Suppose $A \in \mathbf{R}^{n\times n}$. The Arnoldi approach involves the column-by-column generation of an orthogonal matrix Q such that $Q^TAQ = H$ is the Hessenberg reduction. If we write Q as $[q_1, ..., q_m] \in \mathbf{R}^{n\times m}$ and isolate the last term in the summation $Aq_m = \sum_{i=1}^{m+1} h_{im}q_m$, then we have

$$h_{m+1,m}q_{m+1} = Aq_m - \sum_{i=1}^{m} h_{im}q_m \equiv r_m \tag{4.1}$$

where $h_{im} = q_i^TAq_m$ for $i = 1, ..., m$. We assume that q_1 is a given 2-norm starting vector.

Proposition 4.1. 1 *The Arnoldi process computes an orthonormal basis for the Krylov subspace* $\mathsf{K}_m(A, q_1)$

$$\text{span}\{q_1, ..., q_m\} = \text{span}\{q_1, Aq_1, ..., A^{m-1}q_1\}, \tag{4.2}$$

in which the map is represented by an upper Hessenberg matrix H_m.

Proof. The vectors q_j for $j = 1, 2, ..., m$ are orthonormal by construction. That they span K_m follows from the fact that each vector v_j is of the form $p_{j-1}(A)v_1$ where p_{j-1} is a polynomial of degree $j - 1$, which can be shown by induction on j. □

Algorithm 4.1. 1 (Arnoldi)

1. $h_{1,1} = (Aq_1, q_1)$
2. for $j = 1, ..., m-1$, put
3. $\quad r_j = Aq_j - \sum_{i=1}^{j} h_{ij} q_i, \quad h_{j+1,j} = \| r_j \|_2$
4. $\quad q_{j+1} = h_{j+1}^{-1} r_j, \quad h_{i,j+1} = (Aq_{j+1}, q_i), \quad i \leq j+1$

Proposition 4.1. 2 *Denote by H_m the $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations*

$$AQ_m = Q_m H_m + r_m e_m^T \tag{4.3}$$

$$Q_m^H A Q_m = H_m \tag{4.4}$$

hold, where $e_m = (0, ..., 0, 1)^T$.

Proof. (4.3) holds from the equality

$$Aq_j = \sum_{i=1}^{j+1} h_{ij} q_i, \quad j = 1, 2, ..., m. \tag{4.5}$$

(4.4) follows by multiplying both sides of (4.3). □

The algorithm terminates when $r_j = 0$, which is impossible if the minimal polynomial of A with respect to q_1 is of degree $\geq m$. If this condition is satisfied, H_m is an irreducible Hessenberg matrix.

A complete reduction of A to Hessenberg form can be written as $A = QHQ^H$. Consider the first $m < n$ columns of $AQ = QH$. Let Q_m be the $n \times m$ matrix whose columns are the first m columns of Q, and let $\tilde{H}_m$ be the $(m+1) \times m$ upper-left section of H_{m+1}. We have $AQ_m = Q_{m+1}\tilde{H}_m$ and the mth column of this equation can be written as

$$Aq_m = h_{1m} q_1 + \cdots + h_{mm} q_m + h_{m+1,m} q_{m+1}. \tag{4.6}$$

Then the vectors $\{q_i\}$ form bases of the successive Krylov subspaces generated by A and b, defined as

$$\mathsf{K}_m = \text{span}\{b, Ab, ..., A^{m-1}b\} = \text{span}\{q_1, q_2, ..., q_m\} \subseteq \mathbf{C}^n. \tag{4.7}$$

Since the vectors q_i are orthonormal, these are orthonormal bases. Let us define K_m to be a $n \times m$ Krylov matrix

$$K_m = [b, Ab, \cdots, A^{m-1}b]. \tag{4.8}$$

Then K_m have a reduced QR factorization

$$K_m = Q_m R_m, \tag{4.9}$$

where Q_m is the same matrix as above, and might be expected to contain good information about the eigenvalues of A with largest modulus.

Proposition 4.1. 3 *Let $y_i \in \mathbf{C}^m$ be an eigenvector of H_m associated with the eigenvalue $\tilde{\lambda}_i$ and $\tilde{x}_i = Q_m y_i$. Then*

$$(A - \tilde{\lambda}_i I)\tilde{x}_i = h_{m+1,m} e_m^H y_i q_{m+1} \tag{4.10}$$

and, therefore,

$$\| (A - \tilde{\lambda}_i I)\tilde{x}_i \|_2 = h_{m+1,m} |e_m^H y_i|. \tag{4.11}$$

Proof. (4.10) follows from multiplying both sides of (4.3) by y_i. □

4.1.1 Arnoldi and Polynomial Approximation

Let x be a vector in the Krylov subspace K_m. Since x can be written as a linear combination of powers of A times b

$$x = c_0 b + c_1 Ab + c_2 A^2 b + \cdots + c_{m-1} A^{m-1} b, \tag{4.12}$$

we have

$$x = q(A)b, \tag{4.13}$$

where $q(z)$ the polynomial $q(z) = c_0 + c_1 z + \cdots + c_{m-1} z^{m-1}$. The Arnoldi process solves the problem

$$\text{Find } p^m \in P^m \text{ such that } \| p^m(A)b \|_2 = \text{ minimum} \tag{4.14}$$

exactly, where we denote by P^m the set of monic polynomials of degree m.

Theorem 4.1. 1 *As long as the Arnoldi process does not break down, the problem (4.14) has a unique solution p^m, which is the characteristic polynomial of H_m.*

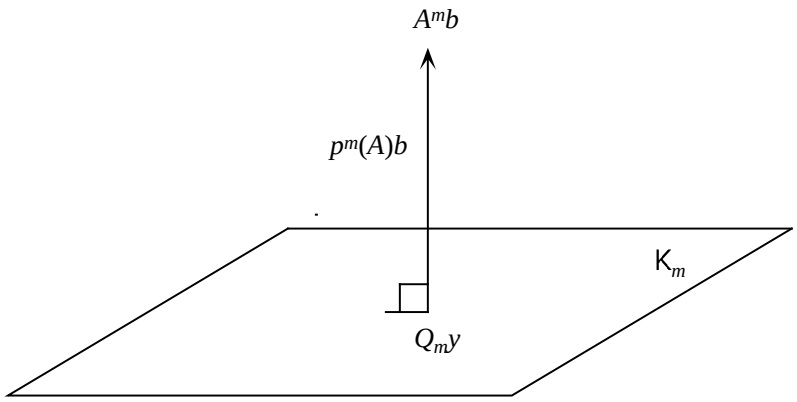

Figure 4.1. Least squares polynomial approximation

Proof. Let us write $p(A)b$ as $p(A)b = A^m b - Q_m y$ for some $y \in \mathbf{C}^m$. Then $\| p^m(A)b \|_2 =$ minimum is equivalent to a linear least squares problem

$$\text{Find } y \text{ such that } \| A^m b - Q_m y \|_2 = \text{ minimum.} \tag{4.15}$$

The solution is characterized by the orthogonality condition $p^m(A)b \perp \mathsf{K}_m$ as illustrated in the figure 4.1, namely, $Q^H p^m(A)b = 0$.

At step m of the Arnoldi process, we have computed the first m columns of Q and H. Then a factorization exists with

$$Q = \begin{pmatrix} Q_m & U \end{pmatrix}, \quad H = \begin{pmatrix} H_m & X_1 \\ X_2 & X_3 \end{pmatrix} \tag{4.16}$$

for some $n \times (n-m)$ matrix U with orthogonal columns that are also orthogonal to the columns of Q_m and some matrices X_1, X_2, and X_3 of dimensions $n \times (n-m)$, $(n-m) \times m$, and $(n-m) \times (n-m)$, respectively, with all but the upper-right entry of X_2 equal to 0. The orthogonality condition becomes $Q_m^H Q p^m(H) Q^H b = 0$, which amounts to the condition that the first m entries of the first column of $p^m(H)$ are zero. Because of the structure of H, these are also the first m entries of the first column of $p^n(H_m)$. By the Cayley-Hamilton theorem, these are zero if p^m is the characteristic polynomial of H_m. Conversely, suppose there were another polynomial p^m with $p^m(A)b \perp \mathsf{K}_m$. Taking the difference would give a nonzero polynomial q of degree $m-1$ with $q(A)b = 0$, violating the assumption that K_m is of full rank. □

Let the Arnoldi process be applied to a matrix $A \in \mathbf{C}^{n\times n}$.

Corollary 4.1. 1 (Translation-invariance) *If A is changed to $A + \sigma I$ for some $\sigma \in \mathbf{C}$, and b is left unchanged, then the Ritz values $\{\theta_j\}$ change to $\{\sigma + \theta_j\}$.*

Corollary 4.1. 2 (Scale-invariance) *If A is changed to σA for some unitary $\sigma \in \mathbf{C}$, and b is left unchanged, then the Ritz values $\{\theta_j\}$ do not change.*

Corollary 4.1. 3 (Invariance under unitary similarity transformation) *If A is changed to UAU^H for some unitary matrix U, and b is changed to Ub, the Ritz values $\{\theta_j\}$ do not change.*

In all three cases, the Ritz vectors, namely, the vectors $Q_m y_j$ corresponding to the eigenvectors y_j of H_m, do not change under the indicated transformation.

4.1.2 Block Arnoldi

Suppose that we are interested in computing the r eigenvalues of a matrix $A \in \mathbf{R}^{n\times n}$. Assume that $V_1 \in \mathbf{R}^{n\times r}$ is a rectangular matrix having r orthonormal columns. Then the algorithm of the block-Arnoldi method can be described as follows:

Algorithm 4.1. 2 (Block Arnoldi)

1. For $k = 1, ..., m-1$, do
2. $\quad W_k = AV_k$
3. $\quad$ For $i = 1, ..., k$, do
4. $\quad\quad H_{i,k} = V_i^T W_k; \quad W_k = W_k - V_i H_{i,k}$
5. $\quad Q_k R_k = W_k$
6. $\quad V_{k+1} = Q_k; \;\; H_{k+1,k} = R_k$

Letting $U_m = [V_1, ..., V_m]$, the restriction of the matrix A to the Krylov subspace is written as

$$H_m = U_m^T A U_m = \begin{pmatrix} H_{1,1} & H_{1,2} & & \cdots & H_{1,m} \\ H_{2,1} & H_{2,2} & & & H_{2,m} \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & & & \\ 0 & \cdots & 0 & H_{m,m-1} & H_{m,m} \end{pmatrix}. \tag{4.17}$$

The above algorithm gives

$$AV_k = \sum_{i=1}^{k} V_i H_{i,k} + V_{k+1} H_{k+1,k}, \quad k = 1, ..., m, \tag{4.18}$$

which can be written in a form as

$$AU_m = U_m H_m + [0, ..., 0, V_{m+1} H_{m+1,m}]. \tag{4.19}$$

Letting $\tilde{\Lambda}_m = \text{diag}(\tilde{\lambda}_1, ..., \tilde{\lambda}_{mr})$ be the diagonal matrix of eigenvalues of H_m corresponding to the eigenvectors $Y_m = [y_1, ..., y_{mr}]$, the above relation gives

$$AU_m Y_m - U_m H_m Y_m = [0, ..., 0, V_{m+1} H_{m+1,m}] Y_m. \tag{4.20}$$

If we denote by $\tilde{X}_m = U_m Y_m$ the matrix of approximate eigenvectors of A and by $Y_{m,r}$ the last r block of Y_m, we have

$$\| A\tilde{X}_m - \tilde{X}_m \tilde{\Lambda}_m \|_2 = \| H_{m+1,m} Y_{m,r} \|_2, \tag{4.21}$$

which will be used for the stopping criterion in the following numerical evaluation in Appendix A.

4.2 The Arnoldi Iteration

The Arnoldi process, which have the disadvantage of increasing computational complexity per iteration step, can be improved with the restarting technique, by which the dimensions of the Krylov subspaces is kept modest (see Saad [55]). In the iterative variant, we start with an initial vectors V_1 and fix a moderate value m, then compute the eigenvectors of H_m. We begin again, using new starting vectors computed from the approximate eigenvectors.

4.2.1 Explicitly Restarted Arnoldi Iteration

The algorithm of the explicitly restarted Arnoldi iteration is summarized below. The choice of m is usually a tradeoff between the length of the reduction that may be tolerated and the rate of convergence. The accuracy of the Ritz values typically increases as m does. For most problems, the size of m is determined experimentally.

Algorithm 4.2. 1 (Explicitly Restarted Arnoldi)

1. Choose $V_1 \in \mathbf{R}^{n \times r}$
2. For $j = 1, ..., m - 1$, do

3. $W_j = AV_j$

4. For $i = 1, ..., j$, do

5. $\quad H_{i,j} = V_i^T W_j; \quad W_j = W_j - V_i H_{i,j}$

6. $Q_j R_j = W_j; \quad V_{j+1} = Q_j; \quad H_{j+1,j} = R_j$

7. Compute the eigenvalues of $H_m = (H_{i,j}) \in \mathbf{R}^{mr \times mr}$ and select $\{\tilde{\lambda}_1, ..., \tilde{\lambda}_r\}$ of largest real parts

8. Stop if their Ritz vectors $\tilde{X}_0 = [\tilde{x}_1, ..., \tilde{x}_r]$ satisfy the convergence criteria

9. Define the iteration polynomial $\psi_k(\lambda)$ of degree k by $\mathrm{Sp}(H_m) - \{\tilde{\lambda}_1, ..., \tilde{\lambda}_r\}$

10. $\tilde{X}_k = \psi_k(A)\tilde{X}_0; \quad Q_k R_k = \tilde{X}_k; \quad V_1 = Q_k$

4.2.2 Other Approaches

The ARPACK software package by Lehoucq and Sorensen [36] implements an implicitly restarted Arnoldi method. The scheme is called *implicit* because the starting vector is updated with an implicitly shifted QR algorithm on the Hessenberg matrix H_m. This method is motivated by the following result:

Let $AV_m = V_m H_m + r_m e_m^T$ be a length m Arnoldi method and $\phi(\cdot)$ a polynomial of degree $p = m - k$ where $k < m$. Since

$$\phi(A)V_k = V_m \phi(H_m)[e_1, e_2, ..., e_k] \tag{4.22}$$

holds, if we compute the QR factorization of $\phi(H_m)[e_1, e_2, ..., e_k] = Q_k R_k$ then the columns of $V_m Q_k$ are an orthogonal basis for $\mathsf{R}(\phi(A)V_k)$, where we denote by $\mathsf{R}(A)$ the range of matrix A.

We give the basic algorithm as implemented by ARPACK below. Note that the convergence rate of the method does not depend on the distribution of the spectrum.

Algorithm 4.2. 2 (Implicitly Restarted Arnoldi)

1. Build a length m Arnoldi method $AV_m = V_m H_m + r_m e_m^T$ with the starting vector v_1

2. Until convergence, do

3. $\quad$ Compute the eigensystem $H_m S_m = S_m D_m$ ordered with the k wanted eigenvalues located in the leading portion of the quasi-diagonal matrix D_m

4. $\quad$ Perform $m - k = p$ steps of the QR iteration with the unwanted eigenvalues of

D_m as shifts to obtain $H_m Q_m = Q_m H_m^+$

5. Restart: Postmultiply the length m Arnoldi method with Q_k to obtain the length k Arnoldi method $AV_m Q_k = V_m Q_k H_k^+ + r_k^+ e_k^T$ (Q_k represents the matrix consisting of the leading k columns of Q_m, and H_k^+ is the leading principal submatrix of degree k of H_m^+)

6. Extend the length k Arnoldi method to a length m one

4.3 Polynomial Accelerations Techniques

Suppose $A \in \mathbf{R}^{n \times n}$ is a diagonalizable matrix with eigensolutions (x_j, λ_j) for $j = 1, ..., n$. Letting $\psi(\cdot)$ be some polynomial, the current starting vector v_1 can be expanded as $\psi(A)v_1 = x_1\psi(\lambda_1)\zeta_1 + \cdots + x_n\psi(\lambda_n)\zeta_n$ in terms of the basis of eigenvectors. Then if we assume that the eigenvalues are ordered so that the wanted k ones are located at the beginning of the expansion, we seek a polynomial such that $\max_{i=k+1,...,n} |\psi(\lambda_i)| < \min_{i=1,...,k} |\psi(\lambda_i)|$, where the components in the direction of unwanted eigenvectors are dumped.

The acceleration techniques attempt to improve the restarted Arnoldi iteration by solving this min-max problem, where a Chebyshev polynomial $\psi(A)$ on an ellipse containing the unwanted Ritz values is applied to the restart vector to accelerate convergence of the restarted Arnoldi iteration. The full theory on these techniques will be described in Chapter 5 and 6.

Chapter 5

Polynomial Acceleration

The idea of using preconditioners has been developed for solving linear equations to improve the properties of problems. Although it has been developed more recently, preconditioning is an effective approach for eigenvalue problems. We present some important properties for basic iterative methods in Section 5.1 and introduce the ideas of the polynomial acceleration in the preceding sections.

5.1 General Theory

5.1.1 Basic Iterative Methods and Their Rates of Convergence

Consider a linear system of n equations

$$Ax = b. \tag{5.1}$$

Definition 5.1. 1 *For a vector x, the residual of (5.1) is denoted by*

$$r = r(x) = Ax - b. \tag{5.2}$$

A *basic iterative method* is written in the form

$$Cd_{l+1} = r_l, \quad x_{l+1} = x_l + d_{l+1}, \quad l = 0, 1, 2, ... \tag{5.3}$$

where $r_l = Ax_l - b$ is the residual and d_l is the correction at stage l. x_0 is an arbitrary initial approximation and C is called the *preconditioning matrix*, which is chosen variously for accelerating convergence. $A = C - R$ is called a *splitting* of A and (5.3) can be rewritten as

$$Cx_{l+1} = Rx_l + b, \quad l = 0, 1, 2, ... \tag{5.4}$$

Lemma 5.1. 1 *For an arbitrary square matrix A,*

$$\lim_{k\to\infty} A^k = 0 \Longleftrightarrow \rho(A) < 1. \tag{5.5}$$

Theorem 5.1. 1 *The sequence of vectors x_l in (5.4) converges to the solution if and only if $\rho(C^{-1}R) < 1$.*

Proof. Letting B be the *iteration matrix* $B = C^{-1}R$ and $e_l = x - x_l$, we have $Ce_{l+1} = Re_l$ and $e_m = B^m e_0$ by recursion. By Lemma 5.1.1, we have $e_m \to 0,\ m \to \infty$ if and only if $\rho(B) < 1$. □

Definition 5.1. 2 $\parallel B^m \parallel$ *is called the convergence factor for m steps and* $R_m = \parallel B^m \parallel^{\frac{1}{m}}$ *is called the average convergence factor for this norm.*

Definition 5.1. 3 $r_m = -\log_{10} R_m$ *is called the average rate of convergence and* $r = -log_{10}\rho(B) = -\log_{10} R_\infty$ *is called the asymptotic rate of convergence.*

5.1.2 Stationary Iterative Methods

Definition 5.1. 4 *A first-order iterative method for the solution of $Ax = b$ is defined by*

$$Cd_{l+1} = -\tau_l r_l,\ x_{l+1} = x_l + d_{l+1}, \quad l = 0, 1, ..., \tag{5.6}$$

where $\{\tau_l\}$ is a sequence of parameters. If $\tau_l = \tau$, $l = 0,\ 1,\ 2, ...$, then the method is called stationary and otherwise nonstationary.

Definition 5.1. 5 *A second-order method is defined by*

$$Cs_l = r_l,\ x_{l+1} = \alpha_l x_l + (1 - \alpha_l)x_{l-1} - \beta_l s_l, \quad l = 0, 1, ..., \tag{5.7}$$

where $\{\alpha_l\}$, $\{\beta_l\}$ are sequences of parameters with $\alpha_0 = 1$.

Letting $e_l = x - x_l$ the errors, we have

$$e_{l+1} = (I - \tau_l A)e_l \tag{5.8}$$

from the above result and

$$e_m = \Pi_{l=0}^{m-1}(I - \tau_l A)e_0 = P_m(A)e_0 \tag{5.9}$$

$$r_m = P_m(A)r_0 \tag{5.10}$$

for some polynomial P_m of degree m such that $P_m(0) = 1$. Similar relations hold for the second-order iterative method. Then we have

$$\| e_m \|_2 = \| P_m(A)e_0 \|_2 \leq \| P_m(A) \|_2 \| e_0 \|_2 = \max_{1 \leq j \leq n} |P_m(\lambda_j)| \| e_0 \|_2 \tag{5.11}$$

(see Dunford and Schwartz [15]).

Definition 5.1. 6 *Let A be s.p.d (symmetric positive definite). The weighted vector-norm is defined by*

$$\| x \|_{A^\nu} = (x^T A^{2\nu} x)^{\frac{1}{2}}, \tag{5.12}$$

where ν is a real number.

Theorem 5.1. 2 *Assume that C and A are s.p.d. and consider the first-order stationary method with a fixed parameter τ. If we assume that $C^{-1}A$ has eigenvalues λ_j with extreme eigenvalues λ_1, λ_n where $0 < \lambda_1 < \lambda_n$, the method converges if $0 < \tau < 2/\lambda_n$.*

Proof. The former part is the consequence of the relations

$$e_m = P_m(C^{-1}A)e_0 = (I - \tau C^{-1}A)^m e_0 \tag{5.13}$$

and

$$\| e_m \|_{A^{\frac{1}{2}}} / \| e_0 \|_{A^{\frac{1}{2}}} \leq \rho(I - \tau C^{-\frac{1}{2}} A C^{-\frac{1}{2}})^m = \max\{|1 - \tau\lambda_1|, |1 - \tau\lambda_n|\}^m. \tag{5.14}$$

Note that the minimum value of $\rho(I - \tau C^{-\frac{1}{2}} A C^{-\frac{1}{2}})^m$ is taken for $\tau = 2/(\lambda_1 + \lambda_n)$, which follows from

$$\min_\tau \rho(I - \tau A^{-\frac{1}{2}} C A^{-\frac{1}{2}}) = \min_\tau \{|1 - \tau\lambda_1|, |1 - \tau\lambda_n|\} = (1 - \lambda_1/\lambda_n)/(1 + \lambda_1/\lambda_n), \tag{5.15}$$

where $1 - \tau\lambda_1 = \tau\lambda_n - 1$ holds. □

Similar results can be obtained for the second-order iterative method.

Theorem 5.1. 3 *Assume that the eigenvalues λ_i of $C^{-1}A$ are positive, with extreme eigenvalues $\lambda_1, \lambda_n, \lambda_1 < \lambda_n$. If we consider a stationary iterative method with the fixed parameters α, β, the*

method converges if and only if $0 < \alpha < 2,\ 0 < \beta < 2\alpha/\lambda_n$ *and the asymptotic convergence factor is smallest and equal to*

$$(\alpha_{\text{opt}} - 1)^{\frac{1}{2}} = \left(\frac{1 - \sqrt{1 - \rho_0^2}}{1 + \sqrt{1 - \rho_0^2}}\right)^{\frac{1}{2}} = \frac{1 - \sqrt{\frac{\lambda_1}{\lambda_n}}}{1 + \sqrt{\frac{\lambda_1}{\lambda_n}}} \tag{5.16}$$

for

$$\alpha = \alpha_{\text{opt}} = \frac{2}{1 + (1 - \rho_0^2)^{\frac{1}{2}}}, \quad \beta = \beta_{\text{opt}} = \frac{2\alpha}{\lambda_1 + \lambda_n} \tag{5.17}$$

where $\rho_0 = [1 - \lambda_1/\lambda_n] \,/\, [1 + \lambda_1/\lambda_n]$.

Proof. See Axelsson [3]. □

5.2 The Chebyshev Iterative Method

5.2.1 The First-Order Chebyshev Iterative Method

Consider a first-order iterative method

$$x_{l+1} = x_l - \tau_l(Ax_l - b) \tag{5.18}$$

where A is s.p.d. The smallest spectral radius

$$\rho(I - \tau A) = (1 - \lambda_1/\lambda_n)/(1 + \lambda_1/\lambda_n) \tag{5.19}$$

is taken with $\tau = 2/(\lambda_n + \lambda_1)$.

We consider to choose a suitable set of parameters $\{\tau_l\}$ to accelerate the convergence by minimizing the norm $\| e_p \|_{A^\nu}$ of the errors $e_p = x - x_p$ after p iterations, that is,

$$e_p = Q_p(A)e_0, \tag{5.20}$$

where

$$Q_p(\zeta) = \Pi_{l=0}^{p-1}(1 - \tau_l\zeta), \tag{5.21}$$

Then we have

$$\| e_p \|_{A^\nu} \leq \| Q_p(A) \|_{A^\nu} \| e_0 \|_{A^\nu} . \tag{5.22}$$

Since

$$\| Q_p(A) \|_{A^\nu} = \max_{y \neq 0} \frac{\| Q_p(A)y \|}{\| y \|} = \max |\mu_i| \tag{5.23}$$

where $\{\mu_i\}$ are the eigenvalues of $Q_p(A)$, we have

$$\| Q_p(A) \|_{A^\nu} = \max_i |Q_p(\lambda_i)| \tag{5.24}$$

to be minimized. So we can simplify the approximation problem to the corresponding problem

$$\| Q_p(A) \|_{A^\nu} = \max_i |Q_p(\lambda_i)| \leq \max_{\lambda_1 \leq \zeta \leq \lambda_n} |Q_p(\zeta)| \tag{5.25}$$

for the continuous interval.

Theorem 5.2. 1 *The least maximum of (5.25) is achieved by the Chebyshev polynomials, namely,*

$$\begin{aligned} \min_{Q_p \in \pi_p^1} \max_{\lambda_1 \leq \zeta \leq \lambda_n} |Q_p(\zeta)| &= \max_{\lambda_1 \leq \zeta \leq \lambda_n} \frac{\left| T_p\left(\frac{\lambda_n + \lambda_1 - 2\zeta}{\lambda_n - \lambda_1}\right)\right|}{T_p\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)} \\ &= \frac{1}{T_p\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)}, \end{aligned} \tag{5.26}$$

where π_p^1 is the set of polynomials of degree at most p which take the values unity at the origin, and T_p is the Chebyshev polynomial of first kind

$$T_p(\zeta) = \frac{1}{2}\left[(\zeta + \sqrt{\zeta^2 - 1})^p + (\zeta - \sqrt{\zeta^2 - 1})^p\right]. \tag{5.27}$$

Proof. Let

$$R(\zeta) = Q_p(0)\frac{T_p\left(\frac{\lambda_n + \lambda_1 - 2\zeta}{\lambda_n - \lambda_1}\right)}{T_p\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)} - Q_p(\zeta) \tag{5.28}$$

be a polynomial of degree p, which takes on zero at $\zeta = 0$. If we assume that

$$\max_{\lambda_1 \leq \zeta \leq \lambda_n} |Q_p(\zeta)/Q_p(0)| < \max_{\lambda_1 \leq \zeta \leq \lambda_n} \left| T_p\left(\frac{\lambda_n + \lambda_1 - 2\zeta}{\lambda_n - \lambda_1}\right)\right|. \tag{5.29}$$

$R(\zeta)$ changes sign in each interval (ζ_i, ζ_{i+1}), since $T_p((\lambda_n + \lambda_1 - 2\zeta_i)/(\lambda_n - \lambda_1))) = (-1)^i$, where $\zeta_i = \cos(i\pi/k)$, and $\max_{\lambda_1 \leq \zeta \leq \lambda_n} \left|T_p\left(\frac{\lambda_n + \lambda_1 - 2\zeta}{\lambda_n - \lambda_1}\right)\right| = 1$. So $R(\zeta)$ has p zeros in addition to the zero at $\zeta = 0$, which is in contradiction to its degree p. Hence assumption (5.29) is false, which proves the theorem. □

By the relation (5.21), the optimal parameters τ_l are given by

$$\tau_l = \frac{1}{\frac{\lambda_n-\lambda_1}{2}\cos\theta_l + \frac{\lambda_n+\lambda_1}{2}}, \quad \theta_l = \frac{2l+1}{2p}\pi, \quad l = 0, 1, ..., p-1. \tag{5.30}$$

Since the extreme eigenvalues are not known, lower and upper bounds $a(a > 0)$ and b of these eigenvalues are used in practice. Consequently, we can see that

$$\min_{Q_p \in \pi_p^1} \max_{\lambda_1 \le \zeta \le \lambda_n} |Q_p(\zeta)| = \frac{1}{T_p\left(\frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1}\right)} = 2\frac{\sigma^p}{1+\sigma^{2p}} \tag{5.31}$$

where $\sigma = (1-\sqrt{\lambda_1/\lambda_n})/(1+\sqrt{\lambda_1/\lambda_n})$, and that the asymptotic average reduction rate is $\lim_{p\to\infty}(\min\max|Q_p(\zeta)|)^{\frac{1}{p}} = \sigma$. Thus, for any $\varepsilon,\ 0 < \varepsilon < 1$,

$$\frac{\| e_p \|_{A^\nu}}{\| e_0 \|_{A^\nu}} \le \varepsilon \tag{5.32}$$

holds if

$$2\frac{\sigma^p}{1+\sigma^{2p}} \le \varepsilon, \tag{5.33}$$

which holds for $p \ge p^*$ where

$$p^* = \ln\left(\frac{1}{\varepsilon} + \sqrt{\frac{1}{\varepsilon^2}-1}\right) / \ln\sigma^{-1}. \tag{5.34}$$

If we assume that $\varepsilon \ll 1$ and $\lambda_n/\lambda_1 \gg 1$,

$$p^* \le \lceil \ln\frac{2}{\varepsilon} / \ln\sigma^{-1} \rceil \le \frac{1}{2}\left(\frac{\lambda_n}{\lambda_1}\right)^{\frac{1}{2}} \ln\frac{2}{\varepsilon}. \tag{5.35}$$

The above technique can be applied to the first order iterative method with a preconditioning matrix C, if $C^{-1}A$ has positive eigenvalues. Note that the number of iterations increases at most as the square root of the condition number of $C^{-1}A$.

5.2.2 The Second-Order Chebyshev Iterative Method

For the second-order iterative method, the parameter sets $\{\alpha_l, \beta_l\}$ of

$$x_{l+1} = \alpha_l x_l + (1-\alpha_l)x_{l-1} - \beta_l r_l, \quad l = 1, 2, ..., \tag{5.36}$$

$$x_1 = x_0 - \frac{1}{2}\beta_0 r_0. \tag{5.37}$$

can be determined independently of rounding errors.

By comparing the relation

$$Q_{l+1}(A) - \alpha_l Q_l(A) + \beta_l A Q_l(A) + (\alpha_l - 1)Q_{l-1}(A) = 0, \quad l = 1, 2, ... \tag{5.38}$$

with the recursion formula

$$T_0(z) = 1,\ T_1(z) = z,\ T_{l+1}(z) - 2zT_l(z) + T_{l-1}(z) = 0, \quad l = 1, 2, ... \tag{5.39}$$

for the Chebyshev polynomials, we can derive a Chebyshev iteration method

$$Q_l(A) = T_l(Z)/T_l(\tilde{b}), \quad Z = \frac{1}{b-a}[(b+a)I - 2A], \tag{5.40}$$

where

$$\alpha_l = \frac{2\tilde{b}T_l(\tilde{b})}{T_{l+1}(\tilde{b})} = 1 + \frac{T_{l-1}(\tilde{b})}{T_{l+1}(\tilde{b})}, \quad \beta_l = \frac{4}{b-a}\frac{T_l(\tilde{b})}{T_{l+1}(\tilde{b})}, \tag{5.41}$$

and

$$0 < a \leq \lambda_1, b \geq \lambda_n, \quad \tilde{b} = (b+a)/(b-a). \tag{5.42}$$

The parameters are defined by the following recursions:

$$a_l = \frac{a+b}{2}\beta_l,\ \beta_l = \frac{1}{\frac{a+b}{2} - \left(\frac{b-a}{4}\right)^2 \beta_{l-1}}, \quad l = 1, 2, ..., \tag{5.43}$$

$$\beta_0 = 4/(a+b). \tag{5.44}$$

Note that the sequence $\{\beta_l\}$ decreases monotonically and converges to $\tilde{\beta} = 4/(\sqrt{b} + \sqrt{a})^2, l \to \infty$, which is equal to the optimal parameter β_{opt} in the stationary second-order method.

5.2.3 The Chebyshev Iterative Method for Nonsymmetric Matrices

Assume that the spectrum of a real nonsymmetric matrix A is contained in an ellipse in the righthalf complex plane,

$$\mathcal{S} = \left\{\zeta \mid \zeta = \frac{b+a}{2} - \frac{b-a}{2}(\cos\theta + i\delta\sin\theta)/\sqrt{1-\delta^2},\ 0 \leq \theta \leq 2\pi\right\}, \quad 0 < a < b \tag{5.45}$$

where $(a, 0)$ and $(b, 0)$ are the foci of the ellipse, δ is the eccentricity. Since the ellipse does not contain the origin, we have

$$\delta < \frac{2\sqrt{\frac{a}{b}}}{1 + \frac{a}{b}} \tag{5.46}$$

from the relation $(b+a)/2 > (b-a)/(2\sqrt{1-\delta^2})$. Note that for $\delta = 0$, $\mathcal{S}$ is the interval $[a, b]$.

The transformation

$$P_1(\zeta) = z = \frac{b+a-2\zeta}{b-a} \tag{5.47}$$

takes the given ellipse $\mathcal{S}$ into a new ellipse

$$\mathcal{E} = \{z \mid z = (\cos\theta + i\delta\sin\theta)/\sqrt{1-\delta^2},\ 0 \le \theta \le 2\pi\}, \tag{5.48}$$

the foci of which are $(-1, 0)$ and $(1, 0)$. Letting $\rho_1 = \sqrt{[(1+\delta)/(1-\delta)]}$, we have

$$\mathcal{E} = \{z \mid z = \frac{1}{2}(\rho_1 + \rho_1^{-1})\cos\theta + i\frac{1}{2}(\rho_1 - \rho_1^{-1})\sin\theta = \frac{1}{2}(\rho_1 e^{i\theta} + \rho_1^{-1}e^{-i\theta}),\ 0 \le \theta \le 2\pi\}. \tag{5.49}$$

Consider the Chebyshev polynomial

$$T_k(z) = \frac{1}{2}\{[z + \sqrt{z^2-1}]^k + [z + \sqrt{z^2-1}]^{-k}\}, \tag{5.50}$$

which is a polynomial of degree k in $z \in \mathbf{C}$. Then

$$T_k(z) = \frac{1}{2}(\rho_1^k e^{ik\theta} + \rho_1^{-k}e^{-ik\theta}) \tag{5.51}$$

and

$$\max_{z\in\mathcal{E}} |T_k(z)| = \frac{1}{2}(\rho_1^k + \rho_1^{-k}), \tag{5.52}$$

where the maximum is taken for $\theta = 0$, for example. Therefore, we get

$$\max_{z\in\mathcal{E}} |T_k(z)| = T_k(\frac{\rho_1 + \rho_1^{-1}}{2}) = T_k(\frac{1}{\sqrt{1-\delta^2}}). \tag{5.53}$$

Note that the parameters

$$\tau_l = \frac{1}{\frac{b-a}{2}\cos\theta_l + \frac{b+a}{2}},\ \theta_l = \frac{2l+1}{2p}\pi, \quad l = 0.,,,.p-1 \tag{5.54}$$

for the first-order method and those given in theorem for the second-order method are same as for the real interval $[a, b]$, corresponding to $\delta = 0$. Considering the normalized polynomial, we find the *average asymptotic convergence factor*

$$\begin{aligned}
\rho \le \lim_{k\to\infty}\left\{\max_{\zeta\in\mathcal{S}}\frac{|T_k(P_1(\zeta))|}{|T_k(P_1(0))|}\right\}^{\frac{1}{k}} &= \lim_{k\to\infty}\left\{\frac{T_k(\frac{1}{\sqrt{1-\delta^2}})}{T_k\left(\frac{(b+a)/(b-a)}{(b+a)(b-a)}\right)}\right\}^{\frac{1}{k}} \\
&= \lim_{k\to\infty}\frac{\rho_1}{T_k(\alpha)^{\frac{1}{k}}} \\
&= \frac{\rho_1}{\alpha + \sqrt{\alpha^2-1}},
\end{aligned} \tag{5.55}$$

of the corresponding iterative method, where

$$\alpha = \frac{b+a}{b-a},\ \rho_1 = \frac{1+\delta}{\sqrt{1-\delta}}. \tag{5.56}$$

Thus, we get

$$\rho \leq \frac{1+\delta}{\sqrt{1-\delta^2}} \left(1 - \sqrt{\frac{a}{b}}\right) \left(1 + \sqrt{\frac{a}{b}}\right). \tag{5.57}$$

Note that $\rho < 1$ holds under the above condition on δ. Since the factor ρ gives only the asymptotic rate of convergence

$$\left(\frac{\| r_k \|}{\| r_0 \|}\right)^{\frac{1}{k}} \to \rho, \quad k \to \infty. \tag{5.58}$$

The convergence may not be monotone for nonsymmetric problems.

5.3 Optimal Parameters for the Chebyshev Polynomials

5.3.1 The Mini-Max Problem

Consider the equivalent form of the Chebyshev polynomials (5.27)

$$T_p(z) = \cosh(p \cosh^{-1}(z)) \tag{5.59}$$

and let $\mathcal{F}(\delta, \vartheta)$ be the member of the family of ellipses in the complex plane centered at δ with the focal points at $\delta + \vartheta$ and $\delta - \vartheta$, where δ and ϑ are complex numbers.

Lemma 5.3. 1 *Suppose $z_i \in \mathcal{F}_i(0,1)$ and $z_j \in \mathcal{F}_j(0,1)$. Then we have*

$$\mathrm{Re}\,(\cosh^{-1}(z_i)) < \mathrm{Re}\,(\cosh^{-1}(z_j)) \quad \Leftrightarrow \quad \mathcal{F}_i(0,1) \subset \mathcal{F}_j(0,1), \tag{5.60}$$

$$\mathrm{Re}\,(\cosh^{-1}(z_i)) = \mathrm{Re}\,(\cosh^{-1}(z_j)) \quad \Leftrightarrow \quad \mathcal{F}_i(0,1) = \mathcal{F}_j(0,1). \tag{5.61}$$

Proof. See Manteuffel [38]. □

For the scaled and translated Chebyshev polynomials $P_n(\zeta) = T_n((\delta - \zeta)/\vartheta)/T_n(\delta/\vartheta)$, we can see that

$$\begin{aligned} P_n(\zeta) &= \frac{e^{n \cosh^{-1}(\frac{\delta-\zeta}{\vartheta})} + e^{-n \cosh^{-1}(\frac{\delta-\zeta}{\vartheta})}}{e^{n \cosh^{-1}(\frac{\delta}{\vartheta})} + e^{-n \cosh^{-1}(\frac{\delta}{\vartheta})}} \\ &\doteq e^{n \cosh^{-1}(\frac{\delta-\zeta}{\vartheta}) - n \cosh^{-1}(\frac{\delta}{\vartheta})} \end{aligned} \tag{5.62}$$

for large n, using the definition of the cosh. Therefore, letting

$$\rho(\zeta) = \lim_{n\to\infty} |P_n(\zeta)^{\frac{1}{n}}| \tag{5.63}$$

be the asymptotic convergence factor of $P_n(\zeta)$ at the point ζ, we get

$$\rho(\zeta) = e^{\mathrm{Re}\,(\cosh^{-1}(\frac{\delta-\zeta}{\vartheta})-\cosh^{-1}(\frac{\delta}{\vartheta}))}. \tag{5.64}$$

From the Lemma 5.3.1 and the above definition,

$$\rho(\zeta_i) < \rho(\zeta_j) \quad \Leftrightarrow \quad \mathcal{F}_i(\delta,\vartheta) \subset \mathcal{F}_j(\delta,\vartheta), \tag{5.65}$$

$$\rho(\zeta_i) = \rho(\zeta_j) \quad \Leftrightarrow \quad \mathcal{F}_i(\delta,\vartheta) = \mathcal{F}_j(\delta,\vartheta), \tag{5.66}$$

$$\rho(\zeta) = 1 \quad \Leftrightarrow \quad \zeta \in \hat{\mathcal{F}}(\delta,\vartheta) \tag{5.67}$$

holds, where $\zeta_i \in \mathcal{F}_i(\delta,\vartheta)$, $\zeta_j \in \mathcal{F}_j(\delta,\vartheta)$ and $\hat{\mathcal{F}}(\delta,\vartheta)$ is the member of the family passing through the origin. Thus, we have

$$\lim_{n\to\infty} P_n(\zeta) = \begin{cases} 0 & \zeta \text{ is inside of } \hat{\mathcal{F}}(\delta,\vartheta) \\ \infty & \zeta \text{ is outside of } \hat{\mathcal{F}}(\delta,\vartheta) \end{cases}. \tag{5.68}$$

The following theorem is useful for the subsequent discussion.

Theorem 5.3. 1 *There exists a unique polynomial $t_n \in \pi_n^1$ such that*

$$\max_{\zeta\in\mathcal{T}} |t_n(\zeta)| = \min_{s_n\in\pi_n^1} \max_{\zeta\in\mathcal{T}} |s_n(\zeta)|, \tag{5.69}$$

where $\mathcal{T}$ is a closed and bounded infinite set in the complex plane.

Proof. See Hille [29]. □

If the region is bounded by an ellipse $\mathcal{E}$ with real foci $\delta+\vartheta$ and $\delta-\vartheta$ that does not contain the origin in its interior, we have the following result.

Theorem 5.3. 2 *Let $0 < \vartheta \le a \le \delta$. If $t_n \in \pi_n^1$ satisfies the above condition then*

$$t_n(\zeta) = P_n(\zeta) = \frac{T_n(\frac{\delta-\zeta}{\vartheta})}{T_n(\frac{\delta}{\vartheta})}. \tag{5.70}$$

Proof. See Clayton [11]. □

The result remains asymptotically true when δ and ϑ are complex values:

Proposition 5.3. 1

$$\lim_{n\to\infty} \min_{s_n\in\pi_n^1} \max_{\zeta\in\mathcal{E}} |s_n(\zeta)|^{1/n} = \lim_{n\to\infty} \max_{\zeta\in\mathcal{E}} |P_n(\zeta)|^{1/n}. \tag{5.71}$$

Proof. We begin by showing that

$$\min_{\zeta\in\mathcal{E}}|P_n(\zeta)| \le \min_{s_n\in\pi_n^1}\max_{\zeta\in\mathcal{E}}|s_n(\zeta)| \le \max_{\zeta\in\mathcal{E}}|P_n(\zeta)|. \tag{5.72}$$

The inequality on the right is trivial. Suppose that the inequality on the left is false:

$$\min_{s_n\in\pi_n^1}\max_{\zeta\in\mathcal{E}}|s_n(\zeta)| \le \min_{\zeta\in\mathcal{E}}|P_n(\zeta)|. \tag{5.73}$$

It implies that

$$\min_{s_n\in\pi_n^1} s_n(\zeta) \le P_n(\zeta) \tag{5.74}$$

when $\zeta\in\mathcal{E}$. By Rouchés theorem, $P_n(\zeta) - \min_{s_n\in\pi_n^1} s_n(\zeta)$ has as many zeros in the interior of $\mathcal{E}$ as $\min_{s_n\in\pi_n^1} s_n(\zeta)$. $P_n(\zeta)$ has n zeros on the segment joining the foci $\delta-\vartheta$ and $\delta+\vartheta$. On the other hand, $P_n(0) - \min_{s_n\in\pi_n^1} s_n(0) = 0$ where the origin is exterior to $\mathcal{E}$, which proves that $P_n - \min_{s_n\in\pi_n^1} s_n$ is the zero polynomial, since its degree does not exceed n and it has at least $n+1$ distinct zeros. Thus $P_n(\zeta) = \min_{s_n\in\pi_n^1} s_n(\zeta)$ on $\mathcal{E}$, which contradicts our hypothesis. □

If we use the log form of $\cosh^{-1}$

$$\cosh^{-1}(w) = \ln(w + (w^2-1)^{\frac{1}{2}}), \tag{5.75}$$

we have

$$\rho(\zeta) = \left|\frac{(\frac{\delta-\zeta}{\vartheta}) + ((\frac{\delta-\zeta}{\vartheta})^2-1)^{\frac{1}{2}}}{(\frac{\delta}{\vartheta}) + ((\frac{\delta}{\theta})^2-1)^{\frac{1}{2}}}\right| = \left|\frac{(\delta-\zeta) + ((\delta-\zeta)^2-\vartheta^2)^{\frac{1}{2}}}{\delta^2 + (\delta^2-\vartheta^2)^{\frac{1}{2}}}\right| \tag{5.76}$$

and, by (5.20), the mini-max problem for a matrix A with eigenvalues λ_i is written as

$$\min_{\delta,\vartheta}\max_{\lambda_i}\rho(\lambda_i) = \min_{\delta,\vartheta}\max_{\lambda_i}\left|\frac{(\delta-\lambda_i) + ((\delta-\lambda_i)^2-\vartheta^2)^{\frac{1}{2}}}{\delta^2 + (\delta^2-\vartheta^2)^{\frac{1}{2}}}\right|. \tag{5.77}$$

Definition 5.3. 1 *Let* $\mathcal{H} = \{\lambda_i \mid \lambda_i$*is a vertex of the smallest convex polygon enclosing the spectrum of* $A\}$ *and* $\mathcal{H}^+ = \{\lambda_i\in\mathcal{H} \mid \mathrm{Im}\,(\lambda_i) \ge 0\}$.

Lemma 5.3. 1 *The mini-max problem can be written in the form*

$$\max_{\lambda_i}\rho(\lambda_i,\delta,\vartheta) = \max_{\lambda_i\in\mathcal{H}}\rho(\lambda_i,\delta,\vartheta) = \max_{\lambda_i\in\mathcal{H}^+}\rho(\lambda_i,\delta,\vartheta). \tag{5.78}$$

Proof. The lemma holds from (5.67). □

5.3.2 The Mini-Max Solution

For the above mini-max problem, we use the following theorem from the functional analysis.

Theorem 5.3. 3 (Alternative Theorem) *Let $\{f_i(x,y)\}$ be a finite set of the real valued functions of two real variables, each of which is continuous on a closed and bounded region $\mathcal{S}$, and let*

$$m(x,y) = \max_i f_i(x,y). \tag{5.79}$$

Then $m(x,y)$ takes on a minimum at some point (x_0,y_0) in the region $\mathcal{S}$. If (x_0,y_0) is in the interior of $\mathcal{S}$, one of the following holds:

1. *The point (x_0,y_0) is a local minimum of $f_i(x,y)$ for some i, such that $m(x_0,y_0) = f_i(x_0,y_0)$.*
2. *The point (x_0,y_0) is a local minimum among the locus $\{(x,y) \in \mathcal{S} \mid f_i(x,y) = f_j(x,y)\}$ for some i and j, such that $m(x_0,y_0) = f_i(x_0,y_0) = f_j(x_0,y_0)$.*
3. *The point (x_0,y_0) is such that $m(x_0,y_0) = f_i(x_0,y_0) = f_j(x_0,y_0) = f_k(x_0,y_0) for some i, j and k$.*

Proof. See Bartle [6]. □

Denote by ζ_0 the smaller intersection point of the ellipse and the real axis. Since $\rho(\zeta,\delta,\vartheta^2)$ takes on the same value at each $\zeta \in \mathcal{F}_i(\delta,\vartheta)$,

$$\rho(\lambda_i,\delta,\vartheta^2) = \rho(\zeta_0,\delta,\vartheta^2) = \frac{(\delta-\zeta_0) + ((\delta-\zeta_0)^2 - \vartheta^2)^{\frac{1}{2}}}{\delta + (\delta^2-\vartheta^2)^{\frac{1}{2}}}. \tag{5.80}$$

Note that if we let $(\delta-\zeta_0)^2 = a^2$,

$$\rho(\lambda_i,\delta,\vartheta^2) = \frac{a + (a^2-\vartheta^2)^{\frac{1}{2}}}{\delta + (\delta^2-\vartheta^2)^{\frac{1}{2}}}, \quad \frac{(\delta-x_i)^2}{a^2} + \frac{y_i^2}{a^2-\vartheta^2} = 1. \tag{5.81}$$

We have the following results for the problem (see Manteuffel [37]):

1. Suppose the positive hull $\mathcal{H}^+$ contains only one eigenvalue $\lambda_1 = x_1 + iy_1$. Then the only local minimum of the function $\rho(\lambda_1,\delta,\vartheta^2)$ occurs at $\delta = x_1$, $\vartheta^2 = -y_1^2$, that is, the member of the family passing through λ_1 is the degenerate ellipse. In this case, we have

$$\rho(\lambda_1,x_1,-y_1^2) = \frac{y_1}{x_1 + (x_1^2+y_1^2)^{\frac{1}{2}}}. \tag{5.82}$$

2. Suppose the positive hull $\mathcal{H}^+$ contains two eigenvalues $\lambda_1 = x_1 + iy_1$ and $\lambda_2 = x_2 + iy_2$. The Alternative Theorem yields that the solution must occur along the intersection of the two surfaces

$$\rho(\lambda_1, \delta, \vartheta^2) = \rho(\lambda_2, \delta, \vartheta^2) \tag{5.83}$$

using the relation

$$\rho(\lambda_1, x_1, -y_1^2) < \rho(\lambda_2, x_1, -y_1^2) \tag{5.84}$$

$$\rho(\lambda_1, x_2, -y_2^2) > \rho(\lambda_2, x_2, -y_2^2). \tag{5.85}$$

Since λ_1 and λ_2 satisfy the equation of the same member of ellipses,

$$\frac{(\delta - x_1)^2}{a^2} + \frac{y_1^2}{a^2 - \vartheta^2} = 1 \tag{5.86}$$

$$\frac{(\delta - x_2)^2}{a^2} + \frac{y_2^2}{a^2 - \vartheta^2} = 1. \tag{5.87}$$

Let

$$A = \frac{x_2 - x_1}{2},\ B = \frac{x_2 + x_1}{2},\ S = \frac{y_2 - y_1}{2},\ T = \frac{y_2 + y_1}{2} \tag{5.88}$$

and assume that $x_2 > x_1$. If $S = 0$, then

$$\delta = B, \quad \vartheta^2 = \frac{a^2(a^2 - (A^2 + T^2))}{a^2 - A^2}. \tag{5.89}$$

If $S \neq 0$, then

$$\vartheta^2 = \frac{(\delta - (B + \frac{ST}{A}))(\delta - (B - A\frac{T}{S}))(\delta - (BA\frac{S}{T}))}{\delta - B} \tag{5.90}$$

$$a^2 = (\delta - (B - A\frac{T}{S}))(\delta - (B - A\frac{T}{S})). \tag{5.91}$$

If $S = 0$, the only local minimum is found in terms of $y = a^2$ as the only real root of the cubic polynomial

$$q_1 y^3 + q_2 y^2 + q_3 y + q_4 = 0 \tag{5.92}$$

in the interval (A^2, B^2), where coefficients are

$$q_1 = B^2 + T^2, \tag{5.93}$$

$$q_2 = -3A^2B^2, \tag{5.94}$$

$$q_3 = 3A^4B^2, \tag{5.95}$$

$$q_4 = -A^4B^2(A^2 + T^2). \tag{5.96}$$

If $S \neq 0$, then the only local minimum can be found in terms of $z = \delta - B$ as the root of the polynomial

$$p_1 z^5 + p_2 z^4 + p_3 z^3 + p_4 z^2 + p_5 z + p_6 = 0 \tag{5.97}$$

in the interval

$$\begin{aligned} (0, A) \quad & \text{for} \quad S > 0 \\ (-A, 0) \quad & \text{for} \quad S < 0. \end{aligned} \tag{5.98}$$

where the coefficients are

$$\begin{aligned}
p_1 &= (2B - A(\frac{T}{S} + \frac{S}{T}))(2B + \frac{ST}{A} - A(\frac{T}{S} + \frac{S}{T})), && (5.99)\\
p_2 &= (2B + \frac{ST}{A} - A(\frac{T}{S} + \frac{S}{T}))((2AB + ST)(\frac{T}{S} + \frac{S}{T}) + 4A^2) \\
&\quad + B^2(2B - A(\frac{T}{S} + \frac{S}{T})) + B(B^2 - A^2), && (5.100)\\
p_3 &= 4A^2 - 4A^3B(\frac{T}{S} + \frac{S}{T}) + A^2ST((\frac{T^3}{S^3} + \frac{S^3}{T^3}) - 3(\frac{T}{S} + \frac{S}{T})) \\
&\quad + A^2B^2(\frac{T^2}{S^2} + \frac{S^2}{T^2} + 3), && (5.101)\\
p_4 &= AST((B - A\frac{T}{S})(B - 3A\frac{T}{S}) + (B - A\frac{S}{T})(B - 3A\frac{S}{T})), && (5.102)\\
p_5 &= -3A^3ST(2B - A(\frac{T}{S} + \frac{S}{T})), && (5.103)\\
p_6 &= -3A^3ST(B^2 - A^2). && (5.104)
\end{aligned}$$

The best point is called a *pair-wise best point* and the associated ellipse is called the *pair-wise best point*. Its convergence factor is as in (5.81).

3. Suppose that the positive hull $\mathcal{H}^+$ contains three or more eigenvalues. Since the mini-max solution must be a pair-wise best point or a point of intersection of three surfaces, the pair-wise best point of λ_1 and λ_2 is the mini-max solution if and only if the pair-wise best ellipse contains the other eigenvalues in the closure of its interior. Let $\lambda_1 = x_1 + iy_1$, $\lambda_2 = x_2 + iy_2$, and $\lambda_3 = x_3 + iy_3$, where $x_1 < x_2 < x_3$, be the three eigenvalues in the positive hull. Then there is the unique point (δ, ϑ^2) such that

$$\rho(\lambda_1, \delta, \vartheta^2) = \rho(\lambda_2, \delta, \vartheta^2) = \rho(\lambda_3, \delta, \vartheta^2) \tag{5.105}$$

only if

$$(x_2 - x_1)(y_3^2 - y_1^2) < (x_3 - x_1)(y_2^2 - y_1^2), \tag{5.106}$$

where the parameters are

$$\delta = \frac{1}{2}\frac{y_1^2(x_2^2 - x_3^2) + y_2^2(x_3^2 - x_1^2) + y_3^2(x_1^2 - x_2^2)}{y_1^2(x_2 - x_3) + y_2^2(x_3 - x_1) + y_3^2(x_1 - x_2)}, \tag{5.107}$$

$$a^2 = \delta^2 - \frac{y_1^2 x_2 x_3(x_2 - x_3) + y_2^2 x_1 x_3(x_3 - x_1) + y_3^2 x_1 x_2(x_1 - x_2)}{y_1^2(x_2 - x_3) + y_2^2(x_3 - x_1) + y_3^2(x_1 - x_3)}, \tag{5.108}$$

$$\vartheta^2 = a^2\left(1 - \frac{y_1^2(x_2 - x_3) + y_2^2(x_3 - x_1) + y_3^2(x_1 - x_2)}{(x_1 - x_2)(x_2 - x_3)(x_3 - x_1)}\right). \tag{5.109}$$

Such a point, referred to as a three way point, can be the mini-max solution only if the associated ellipse passing through λ_1, λ_2, and λ_3, referred to as a three-way ellipse, contains the spectrum in the closure of its interior. Its convergence factor is as in (5.81).

Algorithm 5.3. 1 (Chebyshev Acceleration)

1. For each pair of eigenvalues in the positive hull, find the pair-wise best point
2. If the pair-wise best point contains the other members of the positive hull in the closure of its interior, then it is the mini-max solution
3. If no pair-wise best point is the solution, find the three-way point, if it exists, for each set of three eigenvalues in the positive hull
4. If the associated three-way ellipse contains the other members of the positive hull in the closure of its interior, this point is a candidate
5. The three-way candidate with the smallest convergence factor is the mini-max solution

5.4 The Chebyshev Arnoldi Method

5.4.1 Application to the Nonsymmetric Eigenproblems

We consider here the application of the above restarting techniques to the Arnoldi process. Although the drawbacks of the Arnoldi process can be solved by using the method iteratively as was seen in Chapter 4, in some cases the minimum number of the steps m that must be performed in each inner iteration in order to ensure convergence of the process becomes too large.

These difficulties may be overcome by taking a large enough m, but it can become expensive and impractical. In order to avoid these shortcomings, we consider the use of the iterative Arnoldi process in conjunction with the Chebyshev iteration. The main part of this hybrid algorithm is a Chebyshev iteration, which computes a vector of the form $z_i = p_i(A)z_0$, where we denote by p_i

a polynomial of degree i and by z_0 an initial vector. The polynomial p_i is chosen to amplify the components of z_0 in the direction of the wanted eigenvectors, while damping those of the unwanted eigenvectors. Once $z_i = p_i(A)z_0$ is computed, a few steps of the Arnoldi iteration, starting with an initial vector $v_1 = z_i / \parallel z_i \parallel$, are carried out in order to extract the wanted eigenvectors.

5.4.2 The Chebyshev Iteration

Assume that the parameters of the best mini-max polynomial

$$p_n(\zeta) = \frac{T_n(\frac{\zeta-\vartheta}{\delta})}{T_n(\frac{\zeta_1-\vartheta}{\delta})} \tag{5.110}$$

is given beforehand. Letting $\rho_n = T_n((\zeta_1 - \vartheta)/\delta)$ for $n = 0, 1, \cdots$, we have

$$\rho_{n+1}p_{n+1}(\zeta) = T_{n+1}\left(\frac{\zeta-\vartheta}{\delta}\right) = 2\frac{\zeta-\vartheta}{\delta}\rho_n p_n(\zeta) - \rho_{n-1}p_{n-1}(\zeta), \tag{5.111}$$

which can be transformed into

$$p_{n+1}(\zeta) = 2\sigma_{n+1}\frac{\zeta-\vartheta}{\delta}p_n(\zeta) - \sigma_n\sigma_{n-1}(\zeta) \tag{5.112}$$

by setting $\sigma_{n+1} = \rho_n/\rho_{n+1}$, where σ_i, $i = 1, 2, \cdots$ is computed from the recursion

$$\sigma_1 = \frac{\delta}{\lambda_1 - \vartheta}, \tag{5.113}$$

$$\sigma_{n+1} = \frac{1}{2/\sigma_1 - \sigma_n}, \quad n = 1, 2, ... \tag{5.114}$$

The basic algorithm for the Chebyshev iterative method is as follows:

Algorithm 5.4. 1 (Chebyshev Iteration)

1. Choose an arbitrary initial vector z_0
2. $\sigma_1 = \frac{\delta}{\zeta-\vartheta}$
3. $z_1 = \frac{\sigma_1}{\delta}(A - \vartheta I)z_0$
4. For $n = 1, 2, \cdots$, do
5. $\quad \sigma_{n+1} = \frac{1}{2/\sigma_1 - \sigma_n}$
6. $\quad z_{n+1} = 2\frac{\sigma_{n+1}}{\delta}(A - \vartheta I)z_n - \sigma_n\sigma_{n+1}z_{n-1}$

The algorithm of the Chebyshev Arnoldi Method is as follows:

Algorithm 5.4. 2 (Chebyshev Arnoldi)

1. Choose an orthonormal set of r initial vectors V_1, a number of Arnoldi steps m, and a number of Chebyshev steps n

2. Until Convergence, do

3. Perform the m steps of the Arnoldi algorithm starting with V_1

4. Compute the m eigenvalues of the resulting Hessenberg matrix H_m and select the r wanted eigenvalues $\tilde{\lambda}_1, ..., \tilde{\lambda}_r$

5. If satisfied stop, otherwise continue

6. Using $\mathrm{Sp}(H_m) - \{\tilde{\lambda}_1, ..., \tilde{\lambda}_r\}$, obtain the new estimates of the parameters δ and ϑ of the best ellipse

7. Compute the sequence of r initial vectors Z_0 for the Chebyshev iteration from the approximate eigenvectors $\tilde{x}_1, ..., \tilde{x}_r$

8. Perform n steps of the Chebyshev iteration to obtain Z_n

9. Take V_1 as the orthonormalized vector set computed from Z_n and go back to 1

Chapter 6

Least Squares Based Polynomial Acceleration

The choice of ellipses as enclosing regions in Chebyshev acceleration presented in Chapter 5 is overly restrictive and ineffective, especially when the shape of the convex hull of the unwanted eigenvalues bears little resemblance to an ellipse (see Smolarski and Saylor [65] for various examples). In this chapter, we introduce the idea of acceleration using the least squares polynomials, and propose an efficient method for determining its parameters to solve this problem.

We begin with some definitions of notations in Section 6.1. The mini-max polynomial, which can be derived from the definition of a new norm on the boundary, is introduced and combined with the Arnoldi method as an accelerator in Section 6.2.

6.1 Basic Approach

Let $\mathcal{T}$ be a simply connected region in the complex ζ-plane.

Theorem 6.1. 1 (The Maximum Principle) *If a function $f(\zeta)$ is defined and continuous on a closed bounded set $\mathcal{T}$ and analytic on the interior of $\mathcal{T}$, then the maximum of $|f(\zeta)|$ on $\mathcal{T}$ is assumed on the boundary of $\mathcal{T}$.*

Proof. See Ahlfors [1], for example. □

Using the above property, we can regard the mini-max problem introduced in Chapter 5 as being defined on the boundary of the region which contains the spectrum. Let the boundary C of the region be a continuum consisting of a finite number of rectifiable Jordan arcs. The integrals

considered will have the form

$$\int_C f(\zeta)|d\zeta|, \tag{6.1}$$

where $f(\zeta)$ is a Lebesgue-integrable function defined on C and $|d\zeta|$ is the arc element on C.

Definition 6.1. 1 *The scalar product of two functions $f(\zeta)$ and $g(\zeta)$, ζ on C, is defined by the integral*

$$\langle f, g\rangle = \int_C f(\zeta)\bar{g}(\zeta)w(\zeta)|d\zeta|. \tag{6.2}$$

We introduce here the least squares residual polynomial minimizing an L_2 norm with respect to some weight $w(\zeta)$ on the boundary of a convex hull formed from the approximate eigenestimates. Note that the constraint $p_n(\zeta) \in \pi_n^1$ is not necessary for the eigenproblems (see Section 6.3).

The contour considered here is the finite union of line segments:

Definition 6.1. 2 *Denote by $\mathcal{H}$ the convex hull constituted from the μ vertices $h_0, \cdots, h_\mu$, and by*

$$\vartheta_\nu = \frac{1}{2}(h_\nu + h_{\nu-1}) \tag{6.3}$$

$$\delta_\nu = \frac{1}{2}(h_\nu - h_{\nu-1}), \tag{6.4}$$

the center and the half width on each edge C_ν, $\nu = 1, 2, ..., \mu$, respectively. We can define the Chebyshev weight

$$w_\nu(\zeta) = \frac{2}{\pi}\left[\delta_\nu^2 - (\zeta - \vartheta_\nu)^2\right]^{-\frac{1}{2}}, \quad \zeta \in C_\nu \tag{6.5}$$

on each edge C_ν, and the inner product

$$\langle p, q\rangle = \int_C p(\zeta)\bar{q}(\zeta)w(\zeta)|d\zeta| \tag{6.6}$$

$$\equiv \sum_{\nu=1}^{\mu}\int_{C_\nu} p(\zeta)\bar{q}(\zeta)w_\nu(\zeta)|d\zeta| = \sum_{\nu=1}^{\mu}\langle p, q\rangle_\nu \tag{6.7}$$

on the boundary C. A norm is defined by $\| p \|_w^2 = \langle p, p\rangle$.

Thus, we can rewrite the problem as

$$\min_{p\in\pi_n}\max_{\zeta\in\mathcal{H}}|p(\zeta)| = \min_{p\in\pi_n,\ \zeta\in C}\| p(\zeta) \|_w . \tag{6.8}$$

An algorithm using explicitly the modified moments $\langle t_i(\zeta), t_j(\zeta)\rangle$, where $\{t_j\}$ is some suitable basis of polynomials, is developed for the problem of computing the least squares polynomials in

the complex plane. The set of Chebyshev polynomials suitably shifted and scaled is reasonable as the basis $\{t_j\}$ rather than the power basis $\{1, \zeta, \zeta^2, \cdots, \zeta^{n-1}\}$, which forces unstable computation. However, the matrix M_n whose elements $m_{i,j}$ are defined by

$$m_{i,j} = \langle t_{j-1}, t_{i-1} \rangle, \quad i, j = 1, 2, ..., n+1 \tag{6.9}$$

is still likely to become increasingly ill-conditioned as its size $n+1$ increases.

We express the polynomial $t_j(\zeta)$ in terms of the Chebyshev polynomials

$$t_j(\zeta) = \sum_{i=0}^{j} \varphi_{i,j}^{(\nu)} T_i(\xi_\nu) \quad \text{where } \xi_\nu = \frac{\zeta - \vartheta_\nu}{\delta_\nu} \text{ is real.} \tag{6.10}$$

The expansion coefficients $\varphi_{i,j}^{(\nu)}$ can be computed easily from the three term recurrence of the polynomials

$$\beta_{k+1} t_{k+1}(\zeta) = (\zeta - \alpha_k) t_k(\zeta) - \gamma_k t_{k-1}(\zeta). \tag{6.11}$$

6.2 Least Squares Arnoldi

In this section, we propose a new algorithm to get the mini-max polynomial for the accelerating the Arnoldi iteration.

We can orthogonalize the system and lead to a set of polynomials $\{p_n(\zeta)\}$, where

(a) $p_n(\zeta)$ is a polynomial of degree n in which the coefficients of ζ^n is real and positive;

(b) the system $\{p_n(\zeta)\}$ is orthonormal, that is,

$$\int_C p_n(\zeta) \bar{p}_m(\zeta) w(\zeta) |d\zeta| = \delta_{nm}, \quad n, m = 0, 1, 2, ... \tag{6.12}$$

Definition 6.2. 1 *Let $f(\zeta)$ be a continuous function defined on C and let there correspond the formal Fourier expansion*

$$f(\zeta) \sim f_0 p_0(\zeta) + f_1 p_1(\zeta) + \cdots + f_n p_n(\zeta) + \cdots. \tag{6.13}$$

The coefficients f_n, called the Fourier coefficients of $f(\zeta)$ with respect to the given system, are defined by

$$f_n = \langle f, p_n \rangle = \int_C f(\zeta) \bar{p}_n(\zeta) w(\zeta) |d\zeta|, \quad n = 0, 1, 2, ... \tag{6.14}$$

Theorem 6.2. 1 (Bessel's Inequality) *The partial sums $s_n(\zeta)$ of (6.13) minimize the integral*

$$\int_C |f(\zeta) - \rho(\zeta)|^2 w(\zeta) |d\zeta| \tag{6.15}$$

if $\rho(\zeta)$ ranges over the class of all π_n. The minimum is

$$\int_C |f(\zeta)|^2 w(\zeta)|d\zeta| - |f_0|^2 - |f_1|^2 - \cdots |f_n|^2. \tag{6.16}$$

This also yields Bessel's inequality

$$|f_0|^2 + |f_1|^2 + \cdots + |f_n|^2 \leq \| f(\zeta) \|_w = \int_C |f(\zeta)|^2 w(\zeta)|d\zeta|. \tag{6.17}$$

Proof. For any finite system of complex numbers $c_0, c_1, c_2, ..., c_n$, we have

$$\begin{aligned} \left\| f - \sum_{i=1}^{n} c_i p_i \right\|_w^2 &= \left(f - \sum_{i=1}^{n} c_i p_i, f - \sum_{i=1}^{n} c_i p_i \right) \\ &= \| f \|_w^2 - \sum_{i=1}^{n} c_i \bar{f}_i - \sum_{i=1}^{n} \bar{c}_i f_i + \sum_{i=1}^{n} |c_i|^2 \\ &= \| f \|_w^2 - \sum_{i=1}^{n} |f_i|^2 + \sum_{i=1}^{n} |f_i - c_i|^2 \end{aligned} \tag{6.18}$$

by orthonormality of $\{p_n\}$. Since the minimum of (6.18) is attained when $c_i = f_i$ $(i = 1, 2, ..., n)$, we have $\| f - \sum_{i=1}^{n} c_i p_i \|_w^2$, and hence $\sum_{i=1}^{n} |f_i|^2 \leq \| f \|_w^2$. □

The Theorem 6.2.1 has the following important consequence:

Corollary 6.2. 1 (Nishida 1994) *An orthonormal system $\{p_n(\zeta)\}$ satisfies the condition (6.8).*

Using the above properties of orthogonal polynomials, we can describe the new method to generate the coefficients of the ortho-normal polynomials

$$p_n(\zeta) = \sum_{i=0}^{n} \varphi_{i,n}^{(\nu)} T_i \left(\frac{\zeta - \vartheta_\nu}{\delta_\nu} \right), \tag{6.19}$$

in terms of the Chebyshev weight.

From the condition (6.12) of orthonormality on $p(\zeta)$, we have

$$\langle p_0, p_0 \rangle = \sum_{\nu=1}^{\mu} \langle p_0, p_0 \rangle_\nu = 2 \sum_{\nu=1}^{\mu} \left| \varphi_{0,0}^{(\nu)} \right|^2 = 1, \tag{6.20}$$

$$\langle p_1, p_1 \rangle = \sum_{\nu=1}^{\mu} \langle p_1, p_1 \rangle_\nu = \sum_{\nu=1}^{\mu} \left[2 \left| \varphi_{0,1}^{(\nu)} \right|^2 + \left| \varphi_{1,1}^{(\nu)} \right|^2 \right] = 1, \tag{6.21}$$

$$\langle p_0, p_1 \rangle = \sum_{\nu=1}^{\mu} \langle p_0, p_1 \rangle_\nu = 2 \sum_{\nu=1}^{\mu} \varphi_{0,0}^{(\nu)} \bar{\varphi}_{1,1}^{(\nu)} = 0 \tag{6.22}$$

on C. From (6.20), we have

$$\left| \varphi_{0,0}^{(\nu)} \right| = \frac{1}{2\mu}, \quad \nu = 1, 2, ..., \mu, \tag{6.23}$$

and we can choose $1/\sqrt{2\mu}$ as $\varphi_{0,0}^{(\nu)}$.

Note that each expansion of $p_i(\zeta)$ at any edge must be consistent. The consistency of

$$p_1(\zeta) = \varphi_{0,1}^{(\nu)} + \varphi_{1,1}^{(\nu)}(\zeta - \vartheta_\nu)/\delta_\nu = (\varphi_{1,1}^{(\nu)}/\delta_\nu)\zeta + \varphi_{0,1}^{(\nu)} - \varphi_{1,1}^{(\nu)}\vartheta_\nu/\delta_\nu \tag{6.24}$$

between any edges C_ν and $C_{\nu'}$ where $\nu \neq \nu'$ derives the relation

$$\varphi_{1,1}^{(\nu)}/\delta_\nu = \varphi_{1,1}^{(\nu')}/\delta_{\nu'}, \tag{6.25}$$

$$\varphi_{0,1}^{(\nu)} - \varphi_{1,1}^{(\nu)}\vartheta_\nu/\delta_\nu = \varphi_{0,1}^{(\nu')} - \varphi_{1,1}^{(\nu')}\vartheta_{\nu'}/\delta_{\nu'}, \tag{6.26}$$

which can be rewritten as

$$\varphi_{1,1}^{(\nu)} = \delta_\nu t, \quad \varphi_{0,1}^{(\nu)} - \vartheta_\nu t = \varphi_{0,1}^{(\nu')} - \vartheta_{\nu'} t \tag{6.27}$$

where t is a real number. The condition (6.22) yields the relations

$$\sum_{\nu=1}^{\mu} \left(\varphi_{0,1}^{(\nu)} - \vartheta_\nu t\right) = -\sum_{\nu=1}^{\mu} \vartheta_\nu t = \mu \left(\varphi_{0,1}^{(\nu')} - \vartheta_{\nu'} t\right), \quad 1 \leq \nu' \leq \mu, \tag{6.28}$$

which derives

$$\varphi_{0,1}^{(\nu)} = \vartheta_\nu t - \left(\sum_{\nu'=1}^{\mu} \vartheta_{\nu'}\right) t/\mu. \tag{6.29}$$

Putting (6.29) into (6.22), we get

$$2\sum_{\nu=1}^{\mu} \left|\vartheta_\nu - \left(\sum_{\nu'=1}^{\mu} \vartheta_{\nu'}\right)/\mu\right|^2 t^2 + \sum_{\nu=1}^{\mu} |\delta_\nu|^2 t^2 = 1, \tag{6.30}$$

which determines the value of t as

$$t = \frac{1}{\sqrt{S}}, \quad S = \sum_{\nu=1}^{\mu} \left[2\left|\vartheta_\nu - \left(\sum_{\nu'=1}^{\mu} \vartheta_{\nu'}\right)/\mu\right|^2 + |\delta_\nu|^2\right]. \tag{6.31}$$

Thus, we can compute the values of all the coefficients of the polynomial using the values of $\delta_\nu, \vartheta_\nu$,and μ.

Using the expansion (6.10) and (6.19), the three term recurrence

$$\beta_{k+1} p_{k+1}(\zeta) = (\zeta - \alpha_k) p_k(\zeta) - \gamma_k p_{k-1}\ (\zeta) \tag{6.32}$$

on the $p_i(\zeta)$ can be rewritten as

$$\beta_{k+1} p_{k+1}(\zeta) = (\delta_\nu \xi_\nu + \vartheta_\nu - \alpha_k) \sum_{i=0}^{k} \varphi_{i,k}^{(\nu)} T_i(\xi_\nu) - \gamma_k \sum_{i=0}^{k-1} \varphi_{i,k-1}^{(\nu)} T_i(\xi_\nu). \tag{6.33}$$

From the relations

$$\xi_\nu T_i(\xi_\nu) = \frac{1}{2}\left[T_{i+1}(\xi_\nu) + T_{i-1}(\xi_\nu)\right] \text{ for } i > 0, \quad \xi_\nu T_0(\xi_\nu) = T_1(\xi_\nu), \tag{6.34}$$

(6.33) is expressed by

$$\begin{aligned} & \sum \varphi_i \xi_\nu T_i(\xi_\nu) = \frac{1}{2}\varphi_1 T_0(\xi_\nu) + \left(\varphi_0 + \frac{1}{2}\varphi_2\right) T_1(\xi_\nu) + \cdots \\ + \quad & \frac{1}{2}(\varphi_{i-1} + \varphi_{i+1}) T_i(\xi_\nu) + \cdots + \frac{1}{2}(\varphi_{n-1} + \varphi_{n+1}) T_n(\xi_\nu), \quad \varphi_{n+1} = 0, \end{aligned} \tag{6.35}$$

which is arranged into

$$\begin{aligned} \beta_{n+1} p_{n+1}(\zeta) &= \delta_\nu \left[\frac{\varphi_{1,n}^{(\nu)}}{2} T_0(\xi_\nu) + \left(\varphi_{0,n}^{(\nu)} + \frac{\varphi_{2,n}^{(\nu)}}{2}\right) T_1(\xi_\nu) + \cdots + \sum_{i=2}^{n} \left(\frac{\varphi_{i-1,n}^{(\nu)}}{2} + \frac{\varphi_{i+1,n}^{(\nu)}}{2} \right) T_i(\xi_\nu) \right] \\ &+ (\vartheta_\nu - \alpha_n) \sum_{i=0}^{n} \varphi_{i,n}^{(\nu)} T_i(\xi_\nu) - \gamma_n \sum_{i=0}^{n-1} \varphi_{i,n-1}^{(\nu)} T_i(\xi_\nu), \quad T_{-1} = T_1. \end{aligned} \tag{6.36}$$

Proposition 6.2. 1 *Comparing this equation with (6.19), we find the following relations*

$$\beta_{n+1}\varphi_{0,n+1}^{(\nu)} = \frac{1}{2}\delta_\nu \varphi_{1,n}^{(\nu)} + (\vartheta_\nu - \alpha_n)\varphi_{0,n}^{(\nu)} - \gamma_n \varphi_{0,n-1}^{(\nu)}, \tag{6.37}$$

$$\beta_{n+1}\varphi_{1,n+1}^{(\nu)} = \delta_\nu \left(\varphi_{0,n}^{(\nu)} + \frac{\varphi_{2,n}^{(\nu)}}{2}\right) + (\vartheta_\nu - \alpha_n)\varphi_{1,n}^{(\nu)} - \gamma_n \varphi_{1,n-1}^{(\nu)}, \tag{6.38}$$

$$\beta_{n+1}\varphi_{i,n+1}^{(\nu)} = \delta_\nu \left(\frac{\varphi_{i+1,n}^{(\nu)}}{2} + \frac{\varphi_{i-1,n}^{(\nu)}}{2}\right) + (\vartheta_\nu - \alpha_n)\varphi_{i,n}^{(\nu)} - \gamma_n \varphi_{i,n-1}^{(\nu)}, \ i = 2, ..., n+1, \tag{6.39}$$

where

$$\varphi_{-1,n}^{(\nu)} = \varphi_{1,n}^{(\nu)}, \quad \varphi_{i,n}^{(\nu)} = 0 \text{ for } i > n. \tag{6.40}$$

From (6.32) and the orthogonality of the Chebyshev polynomials, we can derive

$$\begin{aligned} \beta_{k+1} &= \langle p_{k+1}, p_{k+1} \rangle^{1/2} \\ &= \sum_{\nu=1}^{\mu} \int_{C_\nu} p_{k+1} \bar{p}_{k+1} w_\nu(\zeta) |d\zeta| \\ &= \sum_{\nu=1}^{\mu} {\sum_{i=0}^{k+1}}' \varphi_{i,k+1}^{(\nu)} \bar{\varphi}_{i,k+1}^{(\nu)}, \end{aligned} \tag{6.41}$$

denoting by $\sum'$ *a modified sum* $\sum_{i=0}^{\prime n} a_i = 2a_0 + \sum_{i=1}^{n} a_i$.

α and γ are computed by

$$\alpha_k = \langle \zeta p_k, p_k \rangle = \sum_{\nu=1}^{\mu} \left(\vartheta_\nu {\sum_{i=0}^{k}}' \varphi_{i,k}^{(\nu)} \bar{\varphi}_{i,k}^{(\nu)} + \delta_\nu {\sum_{i=0}^{k}}' \varphi_{i,k}^{(\nu)} \bar{\varphi}_{i+1,k}^{(\nu)} \right), \tag{6.42}$$

$$\gamma_k = \langle \zeta p_k, p_{k-1} \rangle = \sum_{\nu=1}^{\mu} \delta_\nu \upsilon_\nu, \tag{6.43}$$

where

$$\upsilon_\nu = \varphi_{1,k}^{(\nu)}\bar{\varphi}_{0,k-1}^{(\nu)} + \left(\varphi_{0,k}^{(\nu)} + \frac{\varphi_{2,k}^{(\nu)}}{2}\right)\bar{\varphi}_{1,k-1}^{(\nu)} + \sum_{i=2}^{k-1}\left(\frac{\varphi_{i-1,k}^{(\nu)}}{2} + \frac{\varphi_{i+1,k}^{(\nu)}}{2}\right)\bar{\varphi}_{i,k-1}^{(\nu)}. \tag{6.44}$$

The polynomial obtained in the above procedure is applied to a matrix of the problem in the following way.

$$p_0(A)v_0 = \varphi_{0,0}^{(\nu)} I v_0, \quad 1 \le \nu \le \mu, \tag{6.45}$$

$$p_1(A)v_0 = \varphi_{0,1}^{(\nu)} I v_0 + \varphi_{1,1}^{(\nu)}/\delta_\nu \cdot (A - \vartheta_\nu I)v_0, \tag{6.46}$$

$$p_{i+1}(A)v_0 = \left[(A - \alpha_i I)p_i(A)v_0 - \gamma_i p_{i-1}(A)v_0\right]/\beta_{i+1}. \tag{6.47}$$

Denoting $p_i(A)v_0$ by w_i, the above recurrence is transformed into

$$w_0 = \varphi_{0,0}^{(\nu)} v_0, \tag{6.48}$$

$$w_1 = \varphi_{0,1}^{(\nu)} v_0 + \varphi_{1,1}^{(\nu)}/\delta_\nu \cdot (Av_0 - \vartheta_\nu v_0) = \left(\varphi_{0,1}^{(\nu)} - \varphi_{1,1}^{(\nu)}\vartheta_\nu/\delta_\nu\right) v_0 + \varphi_{1,1}^{(\nu)}/\delta_\nu \cdot Av_0, \tag{6.49}$$

$$w_{i+1} = \left[Aw_i - \alpha_i w_i - \gamma_i w_{i-1}\right]/\beta_{i+1}, \quad i = 2, ..., n_T. \tag{6.50}$$

6.3 Other Approaches

Consider the problem

$$\min_{s \in \pi_{n-1}} \| \, 1 - (\zeta - \lambda_1)s_n(\zeta) \, \|_w, \tag{6.51}$$

which is to find $\eta = (\eta_0, \eta_1, \cdots, \eta_{n-1})^T$ of $s_n(\zeta) = \sum_{i=0}^{n-1} \eta_i t_i(\zeta)$ so that

$$J(\eta) = \| \, 1 - (\zeta - \lambda_1)s_n(\zeta) \, \|_w \tag{6.52}$$

is minimum (see Saad [57, **?**]). The polynomial

$$\zeta s_n(\zeta) = \sum_{i=0}^{n-1} \eta_i \left[\beta_{i+1}t_{i+1}(\zeta) + \alpha_i t_i(\zeta) + \gamma_i t_{i-1}(\zeta)\right] \tag{6.53}$$

is represented in the basis t_i for $i = 0, 1, \cdots, n$ by the vector $T_n\eta$ where T_n is the $n+1$ by n tridiagonal matrix

$$T_n = \begin{pmatrix} \alpha_0 & \gamma_1 & & & \\ \beta_1 & \alpha_1 & \gamma_2 & & \\ & \beta_2 & \alpha_2 & \ddots & \\ & & & \ddots & \gamma_{n-1} \\ & & & \ddots & \alpha_{n-1} \\ & & & & \beta_n \end{pmatrix}. \tag{6.54}$$

We have

$$J(\eta)^2 = [e_1 - (T_n - \lambda_1 I)\eta]^H M_n [e_1 - (T_n - \lambda_1 I)\eta] \tag{6.55}$$

where $e_1 = (1, 0, \cdots, 0)^T$ and the coefficients of the moment matrix are given by

$$m_{i+1,j+1} = 2\,\mathrm{Re}\left[\sum_{\nu=1}^{\mu}\left(2\varphi_{0,j}^{(\nu)}\bar{\varphi}_{0,i}^{(\nu)} + \sum_{k=1}^{i}\varphi_{k,j}^{(\nu)}\bar{\varphi}_{k,i}^{(\nu)}\right)\right], \quad i = 0, 1, ..., j. \tag{6.56}$$

Let $M_n = LL^T$ be the Choleski factorization of M_n. Then we can compute the minimum of

$$J(\eta) = \parallel L^T [e_1 - T_n\eta] \parallel . \tag{6.57}$$

As will been seen in Chapter 7, this approach requires some excessive computation.

Chapter 7

Implementation

7.1 Modeling

7.1.1 The Least Squares Arnoldi Method

We begin with the complexity of the QR algorithm. We use the number of multiplications as the measure of complexity. The QR algorithm requires $4n^2$ multiplications in one complete step, where we denote by n the degree of the matrix. The double-shifted QR algorithm, which we use for our problem, requires $8n^2$ multiplications in one step in which the two shifts concerning a conjugate pair are performed (see Wilkinson [78]). Hence, denoting the number of the steps of the QR algorithm by n_{QR}, we see that $4n^2 n_{\mathrm{QR}}$ real multiplications are necessary to solve the eigenvalue problem of the matrix A, which can roughly estimated at $10n^3$ (see Golub and Van Loan [24]).

The complexity of the Arnoldi method with the re-orthogonalization, which uses the QR algorithm to compute the eigenvalues of the Hessenberg matrix, is estimated using the above result. It follows the algorithm of the simultaneous least squares Arnoldi method.

We require in the rth step of the computation of $h_{ir}, \quad i = 1, \cdots, r+1,$

$$n^2 + 2nr + 2nr + n \tag{7.1}$$

real multiplications. The total complexity required to obtain the Hessenberg matrix H of degree m is, therefore,

$$\begin{aligned} & mn^2 + 2m(m+1)n + (m-1)n \\ = \quad & mn^2 + (2m^2 + 3m - 1)n \end{aligned} \tag{7.2}$$

real multiplications. Adding this to the computation of the eigenvalues of H by the QR algorithm, the complexity of the Arnoldi method is approximately given by

$$mn^2 + (2m^2 + 3m - 1)n + 10m^3 \tag{7.3}$$

real multiplications.

The evaluation of the complexity of the least squares Arnoldi method is as follows:

1. The computation of the eigenvalue estimates requires

$$mn^2 + (2m^2 + 3m - 1)n + 10m^3 \tag{7.4}$$

real multiplications.

2. Suppose that we have μ vertices for the convex hull. The computation of the coefficients $\varphi_{i,j}^{(\nu)}$ requires

$$\mu(6 + \sum_{n=1}^{k}(3 + 4 + 3n)) \approx \mu[\frac{3}{2}k^2 + \frac{17}{2}k] \tag{7.5}$$

complex multiplications where k is the degree of the polynomial. Each complexity of the other coefficients i.e., β, α, and γ, is

$$\mu \sum_{i=2}^{k}(i+1) \approx \mu[k^2 + 3k], \tag{7.6}$$

$$\mu \sum_{i=1}^{k-1}[(i+1) + (i+1)] \approx \mu[2k^2 + 6k], \tag{7.7}$$

$$\mu\{\sum_{i=1}^{k-1}[1 + 1 + 2 + 2(i-2)]\} = \mu[2k^2 - 2k] \tag{7.8}$$

complex multiplications, respectively. The total number of the complex multiplications is approximately

$$\frac{1}{2}\mu[13k^2 + 31k]. \tag{7.9}$$

3. The polynomial iteration requires

$$n + 1 + n + n^2 + (k-1)(n^2 + n + n) \approx kn^2 + 2kn \tag{7.10}$$

complex multiplications.

The total computation of the least squares Arnoldi method for an iteration consists of the sum of those of the three parts, i.e.,

$$mn^2 + (2m^2 + 3m - 1)n + 10m^3 \tag{7.11}$$

real multiplications and

$$kn^2 + 2kn + \frac{1}{2}\mu[13k^2 + 31k] \tag{7.12}$$

complex multiplications.

Moreover, denoting the number of nonzero entries in A by $n_{\rm nz}$ and the number of required eigenvalues in the block Arnoldi iteration by r, the cost of the block Arnoldi can be defined as $\mathcal{O}(rmn_{\rm nz} + m^2r^2n)$ flops. $10r^3m^3$ flops are required for the computation of the eigenvalues of H_m of degree mr by the QR algorithm, $r^3\mathcal{O}(m^2)$ for the corresponding eigenvectors by the inverse iteration, and $2kr\, n_{\rm nz} + \mathcal{O}(n)$ for the polynomial acceleration. The computation of the coefficients costs approximately $\mathcal{O}(\mu k^2)$ flops, where μ is the number of the vertices of the convex hull. Table 7.1 shows that the complexity of the least squares Arnoldi is roughly $\mathcal{O}(n^2)$, while that of the QR algorithm is $\mathcal{O}(n^3)$.

7.1.2 The Additional Cost of the Saad's Method

The complexity of Saad's method is more large. We need the additional computation of the least squares polynomial using the mini-max polynomials of degree $i = 0, \cdots, n$, which are obtained by our method, as an ortho-normal basis.

The computation of the elements of the matrix M requires

$$\sum_{j=0}^{n}\sum_{i=0}^{j}(i+1) = \frac{1}{6}(n+1)(n+2)(n+3) \tag{7.13}$$

complex multiplications. We need $\frac{1}{3}n(n-1)(n+1)$ multiplications to decompose M into LL^T. We compute then the optimal η^* which makes

$$J(\eta) = \| L^T[e_1 - T_n\eta] \| = \| l_{11}e_1 - F_n\eta \| \tag{7.14}$$

minimum by deforming F_n into an upper triangular matrix by the plane rotations. The plane rotations require $\sum_{i=1}^{n} i = n(n+1)/2$ real multiplications. The η is determined by the least squares method. The total superfluous cost of the computation of the least squares polynomial is then $\frac{1}{6}(n+1)(n+2)(n+3)$ complex multiplications and $\frac{1}{3}n(n-1)(n+1) + \frac{1}{2}n(n+1) = \frac{1}{6}n(n+1)(2n+1)$ real multiplications.

7.1.3 The Complexity of the Manteuffel's Method

The complexity of the computation of the best ellipse is rather complicated. It depends on the distribution of the eigenvalues obtained by the Arnoldi method and classified into several cases.

1. When a pair-wise best point is the mini-max solution, the required computation per a pair of eigenvalues is at most 7 real multiplications and the solution of a cubic equation, if the imaginary parts of the two points are equal. If they are not, 78 real multiplications and the solution of the equation of the fifth degree are required. Moreover we need the judgment whether the other eigenvalues are in the ellipse or not. The number of pairs is $\frac{1}{2}m(m-1)$ where we denote by m the number of the eigenvalues.

2. If no pair-wise best point is the solution, we need the computation of the candidate ellipse for every combination of three points, which contains 48 real multiplications for $\frac{1}{6}m(m-1)(m-2)$ combinations. Then the ellipse with the smallest convergence factor must be chosen.

The complexity of the Newton's method for the nonlinear equations depends on the initial value. Considering that it is used for every combination of the eigenvalues, we can conclude that the least squares Arnoldi method, whose complexity of the corresponding part is $\mathcal{O}(\mu k^2)$ is better.

7.1.4 Other Arguments

The speed of linear convergence of the QR algorithm is controlled by $\max_{r=1,\cdots,n-1}\left|\frac{\lambda_{r+1}}{\lambda_r}\right|$. With shifts of origin, the convergence of $a_{nn}^{(k)}$ to an eigenvalue is asymptotically quadratic.

The study of the convergence of the Arnoldi method is far less sufficient than that of the Lanczos method, since the theory of the uniform approximation on a compact set in the complex plane is not so advanced (see Chatelin [10]).

7.2 Numerical Results

We solved some test problems from the Harwell-Boeing sparse matrix collection (see Duff, Grimes and Lewis [14]), the computed spectra of which are shown in Figure A.7 in Appendix A, using the block Arnoldi iteration. Manteuffel's algorithm was used for reference. Table 7.2 and Table 7.3 indicate that our algorithm shows better performance than Manteuffel's method in the cases where the moduli of the wanted eigenvalues are considerably larger than those of the unwanted eigenvalues.

Table 7.4 shows the comparative results on the ARNCHEB package by Braconnier [9], the ARPACK software package by Lehoucq and Sorensen [36], and the Harwell Subroutine Library code EB13 by Scott [63] and Sadkane [60]. ARNCHEB provides the subroutine ARNOL, which implements the explicitly restarted Arnoldi iteration and the Chebyshev polynomial acceleration. EB13 implements the similar algorithm and also uses Manteuffel's Chebyshev polynomial acceleration. ARPACK provides subroutine DNAUPD that implements the implicitly restarted Arnoldi iteration.

From the results of Table 7.4, we can derive the strong dependency of the polynomial acceleration on the distribution of spectrum. Figure A.7 indicates that the non-clustered distribution of spectra causes the slow convergence, since the approximate spectra may completely differ from the accurate ones. Although ARNCHEB gives reasonable results for computing a single eigensolution, it can struggle on problems for which several eigenvalues are requested. ARPACK displays monotonic consistency and is generally faster and more dependable for small convergence tolerances and large departures from normality. However, its restarting strategy can be more expensive.

maximum	least squares Arnoldi					Arnoldi				QR	
eigenvalues	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
2	2	5	15	3.6E–15	0.38	2	15	8.9E–16	0.57	5.1E–15	1.87
1.5	3	5	20	3.0E–15	0.70	3	15	3.7E–15	0.82	3.6E–15	1.85
1.1	5	10	20	2.9E–14	1.6	1	50	7.5E–13	3.93	5.2E–15	18.8

Table 7.1. Random matrices of degree 50, for the cases of $\lambda_{\max} = 2, 1.5$, and 1.1, while the distribution of the other eigenvalues is $\operatorname{Re}\lambda \in [0,1]$, and $\operatorname{Im}\lambda \in [-1,1]$. m, k, and n_{iter} denote the degree of the Arnoldi method, the maximum degree of the Chebyshev polynomials, and the number of the iterations, respectively. CPU times (in seconds) by HP9000/720.

7.3 Parallelization of the QR algorithm

The above results on the complexity of our method indicate the necessity of more efficient computation of the Arnoldi iteration. Although the speed of convergence increases which the subspace size m is chosen larger, the number of floating-point operations, and therefore the time required by the algorithm, rapidly increases with the subspace dimension m. To avoid QR to become a bottleneck, we propose here a new data mapping method and a schedule of the computation for

problem	WEST0497		WEST0655		WEST0989		WEST2021	
degree of matrix	497		655		989		2021	
number of entries	1727		2854		3537		7353	
number of multiplications	924	440	275	120	13751	*	767	320
number of restarts	14	10	3	2	162	*	12	7
CPU time (sec.)	0.37	0.22	0.17	0.12	8.71	*	1.28	0.67

Table 7.2. Test problems from CHEMWEST, a library in the Harwell-Boeing Sparse Matrix Collection, which was extracted from modeling of chemical engineering plants. The results by Manteuffel's algorithm (right) versus those by the least squares Arnoldi method (left), with size of the basis 20, degree of the polynomial 20, and block size 1, respectively, are listed. * denotes the algorithm fails to converge. CPU time by Alpha Station 600 5/333.

degree of matrix	2000		4000		6000		8000		10000	
number of entries	5184		8784		12384		15984		19584	
number of multiplications	589	240	393	180	236	140	393	380	236	80
number of restarts	7	4	5	3	3	2	5	7	3	1
CPU time (sec.)	0.83	0.43	1.24	0.70	1.23	0.85	2.57	2.81	2.14	0.97

Table 7.3. Test problems from TOLOSA extracted from fluid-structure coupling (flutter problem). Size of the basis, degree of the polynomial, and block size are 20, 20, 1, respectively.

Algorithm	$r = 1, m = 8$	$r = 5, m = 20$
EB12	*	98/20930
ARNCHEB	8.6/3233	71/15921
EB13	17/4860	18/4149
ARPACK	3.7/401	2.1/167

Algorithm	$r = 1, m = 12$	$r = 4, m = 20$
EB12	0.6/423	9.1/2890
ARNCHEB	3.4/1401	4.7/1712
EB13	0.4/119	1.3/305
ARPACK	0.5/90	1.3/151

Table 7.4. CPU times by IBM RS/6000 3BT and matrix-vector products for computing the right-most eigenvalues of WEST2021 (left) and PORES2 of degree 1224 (right). * denotes convergence not reached within $2000m$ matrix-vector products. We denote by r the block size and by m the subspace dimension.

the parallel Hessenberg double shifted QR algorithm on distributed memory processors.

The parallelization of non-Hermitian eigenproblem is not commonly studied. A MIMD parallel implementation of the Arnoldi method is implemented and mentioned in Petiton [50] for both tightly coupled as well as loosely coupled memory machines with vector elementary processors and large granularity. This study has already shown that the QR algorithm is the most significant bottleneck on these MIMD architectures. The speed of convergence for such methods usually increases which the subspace size m is chosen larger. The number of floating-point operations, and therefore the time required by the algorithm, rapidly increases with subspace dimension m. Furthermore, m must be taken as small as possible to avoid QR to become a bottleneck.

Henry and van de Geijn [27] show that under certain conditions the described approach is asymptotically 100% efficient. It is impossible to find an implementation with better scalability properties, since for maintaining a given level of efficiency the dimension of the matrix must grow linearly with the number of processors. Therefore, it will be impossible to maintain the performance as processors are added, since memory requirements grow with the square of the dimension, and physical memory grows only with the number of processors. They also show that for the standard implementation of the sequential QR algorithm, it is impossible to find an implementation with better scalability properties.

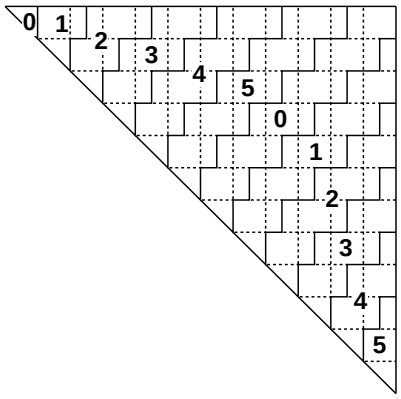

Figure 7.1. The proposed data mapping method

Figure 7.1 shows the data mapping, where the number of the processors $p = 6$. This method is based on the partition of the matrix into $2p \times 2p$ blocks. The mapping is similar to the block Hankel-wrapped storage scheme in that the matrix is partitioned into $2p$ *strips* along the subdiagonal, and

that each processor owns two strips at an interval of p. However, the strips are shifted left by 1.5 blocks, and this shift makes the loads near the diagonal so light that the lookahead step can be executed at the same time with the updates of the previous block transformation. We use a *half block* as a unit of computation: A half block is the computation of the rotations of a half block. We assume that each computation of the lookahead step and the column rotations of a diagonal block, whose nonzero elements are about a half of a block, is a half block. The time taken to execute the computation of a half block is a *quarter*, because each processor has four half blocks of computations in a block transformation.

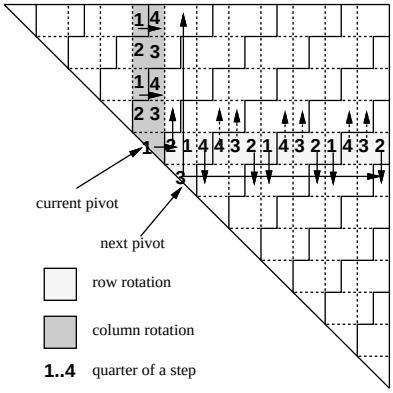

Figure 7.2. Allocation of the computations

Figure 7.2 shows the schedule of the computations in the fourth block transformation. Each processor has four half blocks of computations and the order of the computations is shown with the number 1 to 4. The arrows depict the required communication. The long arrows from the diagonal block stand for the broadcast of the transformations. The lookahead step is executed by the processor 5 in the third quarter. Therefore, there is time of a quarter from the end of a lookahead step to the beginning of the transformations that use the results of the lookahead step, and it becomes possible to hide the latency of the broadcast of the transformations. The column rotation of the diagonal block was done in the first quarter. The row rotations in a processor are executed from right to left and the column rotations in a processor are executed from bottom to top, because the results of the half blocks at the right and the bottom must be sent to the next processors. With this ordering, at least two quarters of time are available to hide the latency of each communication.

We suppose here infinitely large n and p, and ignore lower order terms. In our data mapping, two strips have been allocated to each processor cyclically. The strips can be subdivided into some narrower strips, which will be allocated to the processors in cyclic or block mapping. Assume that the matrix is partitioned into $2bp$ strips and $B = n/2bp$ transformations are bundled. The subdivided strips are allocated to the processors in the block-cyclic fashion, and each processor owns $2h$ block-strips. Note that $1 \leq h \leq b \leq n/2p$. In our algorithm, the following five overheads becomes significant in larger problems:

1. Load Imbalance. The major load imbalance of the parallel double shift QR method is the first lookahead step, which takes $\mathcal{O}(B^2) = \mathcal{O}(n^2/b^2p^2)$ time.

2. Broadcasts. Since the results of the look-ahead steps are broadcasted, the transfer time is larger than the startup time for larger problems. Assuming the circular broadcast method, the overhead time is $\mathcal{O}(hn)$.

3. Border Data Exchanges. A data exchange occurs at border when a rotation requires the data allocated to two different processors. The total amount of the exchanged data is $\mathcal{O}(hn)$ for a processor, which is the order of the total border length.

4. Loop Overhead. The core routine of the QR algorithm is a double loop and the performance is affected by the inner loop length. This overhead is evaluated as $\mathcal{O}(hn)$, which is the outer loop count.

5. Data Redistribution. Data redistribution is required before executing the Francis steps. In the worst case, the time consumption for the data redistribution is $\mathcal{O}(n^2)$, which is the number of the matrix elements. Since the double shifted QR algorithm requires $\mathcal{O}(n)$ Francis iterations, the overhead per Francis iteration is $\mathcal{O}(n)$.

From the above considerations, the overhead per Francis step of our method is $\mathcal{O}(n^2/b^2p^2+hn)$, which is minimized to $\mathcal{O}(n)$ by letting $h = \mathcal{O}(n)$ and $b = \mathcal{O}(n/p)$, where the subdivided strips are allocated in a block fashion and the size of the subdivided blocks are constant. Since the overhead is $\mathcal{O}(n)$ and the load per processor is $\mathcal{O}(n^2/p)$, constant parallel efficiency is obtained with $p = \mathcal{O}(n)$. Therefore, our scheme attains the best possible scalability of the double shifted QR algorithm.

The graph in Figure 7.3 shows the parallel performance of our program without matrix size reduction on a Fujitsu AP1000+, a distributed memory multicomputer system with 256 SuperSparc10 processors (50 MHz). The graph shows the relation between Mflops per processor and n/p with several values for p. The peak performance of the Hessenberg double shift QR algorithm on a single processor of AP1000+ is about 20.8 Mflops, using unrolling and tiling. Therefore, the parallel efficiency of 50% is attained with $n/p < 40$, and the parallel efficiency becomes 90% with $n/p \approx 150$. Such high parallel efficiency has rarely been observed in preceding researches on the parallel double shifted QR algorithms (see Henry and van de Geijn [27]), or the parallel multishift QR algorithms (see Henry, Watkins and Dongarra [28]), considering its minimum parallelizing overhead of $\mathcal{O}(n^{5/4}/p^{1/2})$ for $n \geq p$ and $\mathcal{O}(n/p^{1/4})$ for $n < p$, from which we can see that our algorithm will be faster for $n > \mathcal{O}(p^2)$.

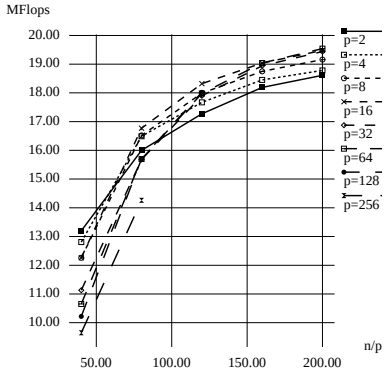

Figure 7.3. The Mflops per processor versus n/p for first iterations. The broken lines in the left figures indicate the boundaries of the blocks, and the solid lines show the boundaries of the elements allocated to different processors. The numbers indicate to which processor each region should be allocated.

Chapter 8

Conclusion

In this thesis, we proposed a least squares based accelerating method for the restarted Arnoldi iteration. In our method, an accelerating polynomial is chosen to minimize an L_2 norm of the polynomial on the boundary of the convex hull, with respect to the Chebyshev weight function. The proof of the minimum property of the orthogonal polynomials defined on the norm was given in Chapter 6.

We estimated the complexity of the least squares Arnoldi in Chapter 7. The complexity of the least squares based acceleration is given by $\mathcal{O}(\mu k^2)$ flops, which is less than those of the Chebyshev acceleration, where the solutions of simultaneous nonlinear equations are required, and the Saad's approach, which needs $\mathcal{O}(k^3)$ additional cost. The validity of our method was confirmed by the experiments using a set of standard test matrices for sparse matrix problems, such as the Harwell-Boeing Sparse Matrix Collection, in Chapter 7 and Appendix A.

The number of floating point operations rapidly increases with the size of the subspace dimension m and it indicates that we need to take m as small as possible if we want to avoid QR to become a bottleneck, as shown in Chapter 7. We proposed a new data mapping method with best possible scalability for the parallelization of the double shifted QR algorithm, in which the loads including the lookahead step are balanced, and the computations are pipelined by hiding the communication latency. Our implementation on a Fujitsu AP1000+ attains parallel efficiency higher than 90% without matrix size reduction, and 70–80% for the whole process including the matrix size reduction. The integration of the these two approaches is the current problem.

Appendix A

Numerical Results

This chapter reports the results of the numerical experiments of our method and evaluates its performance.

A.1 Treatment of the Computational Error

The shortcoming of the method is expected to be the computational error of the eigenvalues distributed closely. We propose several countermeasures for the difficulty. Numerical results are reported in the subsequent chapter.

A.1.1 Computing Complex Eigenvectors

Suppose that the components of $x_2, \cdots, x_n$ has been eliminated and we have

$$u_s = \alpha_1 x_1 + \overline{\alpha_1 x_1}, \quad v_{s+1} = Au_s = \alpha_1 \lambda_1 x_1 + \overline{\alpha_1 \lambda_1 x_1}. \tag{A.1}$$

If we write

$$\alpha_1 x_1 = z_1 + iw_1, \quad \lambda_1 = \xi_1 + i\eta_1 \tag{A.2}$$

then

$$u_s = 2z_1, \quad v_{s+1} = 2\xi_1 z_1 - 2\eta_1 w_1, \tag{A.3}$$

$$z_1 + iw_1 = \frac{1}{2}[u_s + i(\xi_1 u_s - v_{s+1})/\eta_1]. \tag{A.4}$$

Apart from a normalizing factor we have therefore

$$x_1 = \eta_1 u_s + i(\xi_1 u_s - v_{s+1}). \tag{A.5}$$

A.1.2 The Re-orthogonalization

One of the main source of the computational error of the projection method is the orthogonalization process. We consider the vector b_{r+1} defined by

$$b_{r+1} = Ac_r - h_{1r}c_1 - h_{2r}c_2 - \cdots - h_{rr}c_r. \tag{A.6}$$

When the components of b_{r+1} is very small compared with $\parallel Ac_r \parallel_2$, b_{r+1} will not be orthogonal to the c_i. We re-orthogonalize the computed vector b_{r+1} with respect to $c_1, \cdots, c_r$:

$$b'_{r+1} = b_{r+1} - \varepsilon_{1r}c_1 - \varepsilon_{2r}c_2 - \cdots - \varepsilon_{rr}c_r, \tag{A.7}$$

where

$$\varepsilon_{ir} = c_i^T b_{r+1}/c_i^T c_i = c_i^T b_{r+1}, \tag{A.8}$$

for orthogonality. c_{r+1} is computed by

$$c_{r+1} = b'_{r+1}/ \parallel b'_{r+1} \parallel_2 . \tag{A.9}$$

Because b_{r+1} has already been orthogonalized once with respect to c_i, we can be sure that ε is of the same order as in the normal cases.

A.1.3 The Multiplication

We can not always obtain the eigenvalue with the largest real part by the Arnoldi method, especially when there are close eigenvalues. We propose an amplification process defined by

$$\hat{b}_{r+1} = A^n c_r - \sum_{i=1}^{r} h_{ir}c_i, \quad h_{ir} = (A^n c_r, c_i), \quad i = 1, ..., r, \tag{A.10}$$

where

$$h_{r+1,r} = \parallel \hat{b}_{r+1} \parallel . \tag{A.11}$$

The eigenvalues of larger absolute values are made dominant by this method. Note that the convex hull constructed by the nth power of the unwanted eigenvalues is different from the original one.

A.1.4 The Deformation of the Convex Hull

The complexity of the computation of the polynomial on the convex hull is proportional to the number of the edges. We can consider the rectangular area which consists of the unwanted eigenvalues.

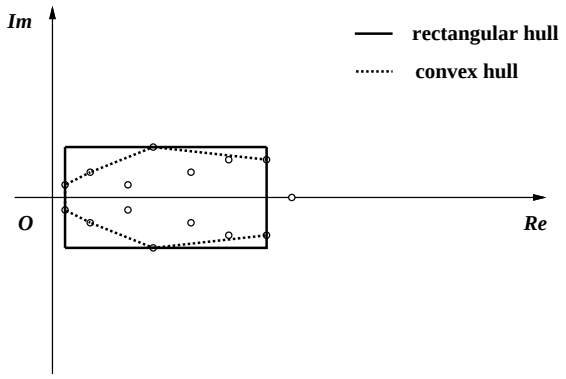

Figure A.1. The concept of the rectangular hull

A.1.5 The Deflation

If the obtained eigenvalues are not the wanted one or if we want to compute another eigenvalue with smaller absolute value, we can remove it by the orthogonalization.

We consider the case with complex eigenvalues. Suppose we have a complex pair of eigenvectors x and $\bar{x}$ which correspond to eigenvalues λ and $\bar{\lambda}$. We remove the components of x and $\bar{x}$ from the approximate vector w obtained through the polynomial iteration. The new vector w' is defined by

$$w' = w - (w, x)x - (w, \bar{x})\bar{x} \tag{A.12}$$

$$= w - 2(u, \operatorname{Re} x) \operatorname{Re} x - 2(u, i \operatorname{Im} x) i \operatorname{Im} x, \tag{A.13}$$

which indicates that the vector $(w, x)x + (w, \bar{x})\bar{x}$ is complex in general.

A.2 Condition

We start from the decision of each element of the matrix given in the problem. In this section, the scaled sequences of random numbers are assigned respectively to the real and the imaginary parts of the eigenvalues except for those which are to be selected. The matrices are block diagonals with 2×2 or 1×1 diagonal blocks. Each block is of the form

$$\begin{pmatrix} a & b/2 \\ -2b & a \end{pmatrix} \tag{A.14}$$

to prevent the matrix to be normal and has eigenvalues $a \pm bi$. It is transformed by an orthogonal matrix generated from a matrix with random elements by the Schmidt's orthogonalization method.

Several values of $n_{\rm ot}$, the number of transformations, are tested in Table A.1 since they may influence the condition of the transformed matrix. m and k denote the iteration number of the Arnoldi method and the maximum degree of the Chebyshev polynomials respectively. We compare this algorithm with the double-shifted QR algorithm and the power method. The number of the iterations of the power method is expressed by $n_{\rm p}$.

A.3 The Iterative Arnoldi Method

We examine two types of the Arnoldi algorithm, i.e., the ordinary one and the iterative one, in the least squares Arnoldi method. The iterative version is proposed by Saad [53]. They are compared with the other projection methods, such as the Arnoldi method, the power method, and the QR algorithm. The re-orthogonalization technique is not used in this section.

A.3.1 The Ordinary Arnoldi Method

We test some values of the maximum eigenvalue $\lambda_{\rm max}$ =2, 1.5, and 1.1 in order to evaluate the effect of the close distribution of the wanted eigenvalues. The distribution of the other eigenvalues is given by $\mathrm{Re}\,\lambda \in [0,1]$ and $\mathrm{Im}\,\lambda \in [-1,1]$.

In the least squares Arnoldi method, the $\lambda_{\rm max}$ is computed by

$$\tilde{\lambda} = \parallel \tilde{x}_{i+1} \parallel_2 / \parallel \tilde{x}_i \parallel_2, \tag{A.15}$$

where

$$\tilde{x}_{i+1} = A\tilde{x}_i, \tag{A.16}$$

since we suppose that the maximum eigenvalue is a positive real number. Another way of computing the approximate eigenvalue using the right-handed eigenvector is performed by

$$\tilde{\lambda} = \frac{(\tilde{x}_{i+1}, \tilde{x}_i)}{(\tilde{x}_i, \tilde{x}_i)} \tag{A.17}$$

using the normal equation. The precision of this technique is inferior to that of the former one, though.

The error is computed by the L_2 norm. The computation time is measured by HP9000/720, where the unit is $\frac{1}{60}$ second.

A.3.1.1 Case 1

The maximum eigenvalue is 2. We test the influence of the number of the orthogonal transformations n_{ot} here. There seems to be no effect on the error of each method. We use $n_{\mathrm{ot}} = 3$ in the rest of the experiments.

As the degree of the matrix A increases, the least squares Arnoldi method gets a clear advantage over the QR algorithm. The complexity of the least squares Arnoldi method can be seen to be roughly $\mathcal{O}(n^2)$ as our evaluation which we made in the previous chapter indicates, while that of the QR algorithm $\mathcal{O}(n^3)$.

The precision of the power method is far less than that of the least squares Arnoldi method. We can show that it gets worse as the maximum eigenvalue approaches the second eigenvalue in the following experiments.

matrix		least squares Arnoldi				Arnoldi			power method			QR	
degree	n_{ot}	m	k	error	time	m	error	time	n_{p}	error	time	error	time
50	1	5	25	1.6E–10	18	15	2.2E–11	11	25	6.0E–12	11	1.7E–15	116
50	2	5	25	2.0E–13	17	15	4.8E–10	10	25	2.1E–10	10	8.8E–16	113
50	3	5	25	2.0E–13	18	15	2.2E–09	12	25	1.0E–10	11	5.1E–15	112
50	4	5	25	1.2E–12	18	15	2.6E–08	13	25	3.2E–10	11	5.3E–15	111
50	5	5	25	2.6E–12	17	15	1.7E–07	12	25	5.0E–09	11	2.2E–15	116
50	10	5	25	3.3E–11	19	15	4.8E–08	11	25	1.2E–08	11	8.8E–15	116
50	50	5	25	3.6E–13	18	15	2.7E–09	13	25	7.9E–10	10	7.7E–14	109
50	100	5	25	1.4E–13	21	15	1.9E–08	11	25	1.1E–09	12	7.0E–13	118
100	3	5	25	1.6E–12	57	15	7.2E–07	31	25	3.5E–08	44	3.1E–15	786
200	3	5	25	1.8E–11	224	15	2.6E–07	109	25	5.2E–09	186	1.1E–15	5745

Table A.1. $\lambda_{\max} = 2$, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0, 1],\ \mathrm{Im}\,\lambda \in [-1, 1]$.

A.3.1.2 Case 2

The maximum eigenvalue is 1.5. Since the condition of the Arnoldi method gets worse, the degree of the Hessenberg matrix of the Arnoldi method m must be larger. Several patterns of the combination of the parameters are tested. The least squares Arnoldi method gives the best results.

The relation between the error and the parameters of the least squares Arnoldi method is given in Figure A.2. We denote the degree of the polynomial by n and the iteration number of the

Arnoldi method by m. This graph shows that the iteration number of the Arnoldi method has the closer correlation with the error than the degree of the least squares Arnoldi method. This is caused by the fact that the Arnoldi method can not always obtain the eigenvalue of the largest modulus. This problem is discussed subsequently.

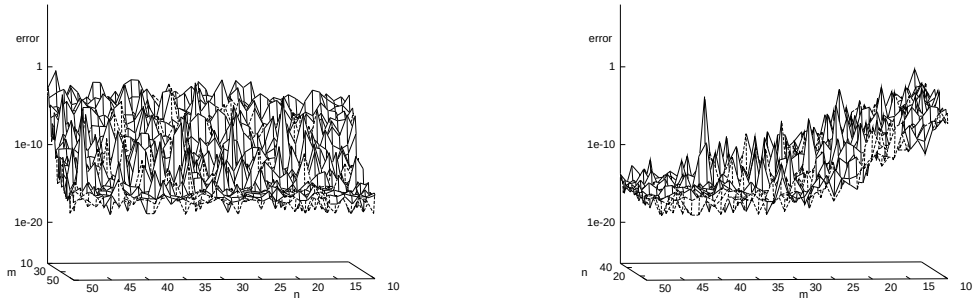

Figure A.2. The relation between the error and the parameters of the least squares Arnoldi method

matrix		least squares Arnoldi				Arnoldi			power method			QR	
degree	n_{ot}	m	k	error	time	m	error	time	n_{p}	error	time	error	time
50	3	5	25	9.4E–06	18	15	1.2E–05	13	25	8.1E–05	12	3.6E–15	111
100	3	5	25	2.5E–04	59	15	2.3E–03	32	25	1.5E–02	43	9.5E–15	830
50	3	10	25	1.0E–07	21	20	8.9E–08	21	25	8.1E–05	11	3.6E–15	113
50	3	10	30	6.2E–10	24	25	1.7E–10	33	30	1.2E–05	13	3.6E–15	124
50	3	10	40	1.2E–11	31	25	1.7E–10	30	40	2.7E–07	17	3.6E–15	109
50	3	10	50	1.3E–14	39	30	3.4E–13	38	50	2.3E–08	22	3.6E–15	113
50	3	20	50	5.1E–11	71	30	3.4E–13	38	50	2.3E–08	21	3.6E–15	115
50	3	10	55	3.1E–15	41	30	3.4E–13	40	55	3.9E–09	23	3.6E–15	113

Table A.2. $\lambda_{\max} = 1.5$, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0, 1],\ \mathrm{Im}\,\lambda \in [-1, 1]$.

A.3.1.3 Case 3

The maximum eigenvalue is 1.1. The convergence of the projection methods is lost. The increase of the degree of the polynomial has no effect.

matrix		least squares Arnoldi				Arnoldi			power method			QR	
degree	n_{ot}	m	k	error	time	m	error	time	n_{p}	error	time	error	time
50	3	10	100	1.0E–03	100	40	1.6E–07	82	100	3.9E–02	45	5.2E–15	117
50	3	15	100	2.6E–05	102	40	1.6E–07	82	100	3.9E–02	44	5.2E–15	115
50	3	20	100	1.6E–01	179	40	1.6E–07	84	100	3.9E–02	45	5.2E–15	118
50	3	30	100	1.4E–01	204	40	1.6E–07	82	100	3.9E–02	43	5.2E–15	112
50	3	15	100	2.6E–05	105	40	1.6E–07	82	200	1.3E–01	93	5.2E–15	118
50	3	15	50	2.3E–03	46	40	1.6E–07	82	50	1.1E–01	24	5.2E–15	114

Table A.3. $\lambda_{\max} = 1.1$, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,1],\ \mathrm{Im}\,\lambda \in [-1,1]$.

A.3.2 The Iterative Arnoldi Method

The iterative Arnoldi method is examined by Saad [53] and it enables us to compute the eigenvalues of a rather ill-conditioned matrix with limited memory space. We adopt this method instead of the ordinary Arnoldi method in the least squares Arnoldi method. The Arnoldi method performed for reference is also made iterative.

In this section we test the five variations of the distribution of the eigenvalues. The cases of $\lambda_{\max} = 2, 1.5$, and 1.1 while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,1]$, and $\mathrm{Im}\,\lambda \in [-1,1]$, and $\lambda_{\max} = 3$ and 2.5 while the distribution of the others is $\mathrm{Re}\,\lambda \in [0,2],\ \mathrm{Im}\,\lambda \in [-1,1]$. The power method is omitted since its inferiority is clear.

We denote the number of the iterations by n_{iter}.

A.3.2.1 Case 1

$\lambda_{\max}$ is 2, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,1],\ \mathrm{Im}\,\lambda \in [-1,1]$. The effect of the iteration is significant, especially for the least squares Arnoldi method. This tendency becomes sharper as the maximum eigenvalue gets closer to the second eigenvalue.

matrix		least squares Arnoldi					Arnoldi				QR	
degree	n_{ot}	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
50	3	1	5	15	1.1E–10	11	1	15	2.3E–09	18	5.1E–15	112
50	3	2	5	15	3.6E–15	23	2	15	8.9E–16	34	5.1E–15	112

Table A.4. $\lambda_{\max} = 2$, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,1],\ \mathrm{Im}\,\lambda \in [-1,1]$.

A.3.2.2 Case 2

The maximum eigenvalue is 1.5, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,1],\ \mathrm{Im}\,\lambda \in [-1,1]$. Some variations of the combination of the parameters n_{iter} and k are examined. The best combination of the parameters is not trivial and the consideration on this problem is given in the last section.

matrix		least squares Arnoldi					Arnoldi				QR	
degree	n_{ot}	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
50	3	1	5	15	5.0E–05	15	1	15	1.2E–05	17	3.6E–15	111
50	3	2	5	15	1.1E–08	25	2	15	5.3E–11	33	3.6E–15	111
50	3	3	5	15	1.9E–11	32	3	15	3.7E–15	49	3.6E–15	111
50	3	4	5	15	5.6E–14	42	4	15	3.3E–15	63	3.6E–15	112
50	3	5	5	15	3.4E–15	54	5	15	2.8E–15	79	3.6E–15	108
50	3	3	5	20	3.0E–15	42	3	15	3.7E–15	47	3.6E–15	110
50	3	3	5	19	8.9E–16	38	3	15	3.7E–15	49	3.6E–15	111
50	3	3	5	18	4.5E–13	38	3	15	3.7E–15	46	3.6E–15	111
50	3	1	5	60	1.3E–14	47	3	15	3.7E–15	47	3.6E–15	111

Table A.5. $\lambda_{\max} = 1.5$, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,1],\ \mathrm{Im}\,\lambda \in [-1,1]$.

A.3.2.3 Case 3

The maximum eigenvalue is 1.1, while the distribution of the other eigenvalues is: $\mathrm{Re}\,\lambda \in [0,1],\ \mathrm{Im}\,\lambda \in [-1,1]$. In this test we examine the relation between the parameter n_{iter} and the iteration number of the Arnoldi method m. The table shows that it is more effective to decrease the iteration number of the Arnoldi method than to decrease the number of the Arnoldi iteration.

A.3.2.4 Case 4 and case 5

The maximum eigenvalues are 3 and 2.5, respectively, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0,2],\ \mathrm{Im}\,\lambda \in [-1,1]$. They are similar experiments as the former ones except for the distribution of the smaller eigenvalues. It can be seen that it is not always effective to compute the polynomial of a higher degree.

matrix		least squares Arnoldi					Arnoldi				QR	
degree	n_{ot}	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
50	3	1	50	10	3.2E–15	240	1	50	7.5E–13	235	5.2E–15	113
50	3	1	45	20	6.9E–15	206	1	45	4.1E–10	191	5.2E–15	115
50	3	2	30	20	3.2E–15	161	2	50	7.7E+00*	531	5.2E–15	113
50	3	3	20	15	6.3E–12	111	1	50	7.5E–13	234	5.2E–15	115
50	3	4	15	20	3.5E–13	112	1	50	7.5E–13	235	5.2E–15	114
50	3	5	10	20	2.9E–14	96	1	50	7.5E–13	236	5.2E–15	112
50	3	6	10	20	6.1E–13	116	1	50	7.5E–13	237	5.2E–15	113
50	3	6	10	25	3.2E–15	147	1	50	7.5E–13	235	5.2E–15	112

Table A.6. $\lambda_{\max} = 1.1$, while the distribution of the other eigenvalues is $\operatorname{Re}\lambda \in [0,1],\ \operatorname{Im}\lambda \in [-1,1]$. *)The Arnoldi iteration fails to converge in some cases, where the wanted eigenvalues are not included in the Ritz values.

matrix		least squares Arnoldi					Arnoldi				QR	
degree	n_{ot}	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
50	3	1	5	10	3.9E–08	8	1	10	1.5E–05	7	2.7E–15	107
50	3	2	5	10	1.3E–11	15	2	10	6.9E–11	14	2.7E–15	110
50	3	3	5	10	3.4E–15	25	3	10	3.1E–15	22	2.7E–15	112
50	3	1	5	25	4.4E–14	20	3	10	3.1E–15	23	2.7E–15	109

Table A.7. $\lambda_{\max} = 3$, while the distribution of the other eigenvalues is $\operatorname{Re}\lambda \in [0,2],\ \operatorname{Im}\lambda \in [-1,1]$.

matrix		least squares Arnoldi					Arnoldi				QR	
degree	n_{ot}	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
50	3	1	5	10	2.2E–06	8	1	10	1.5E–03	8	6.6E–15	107
50	3	2	5	10	1.9E–06	16	2	10	1.2E–06	15	6.6E–15	113
50	3	2	10	10	1.0E–12	26	4	10	1.7E–12	29	6.6E–15	110
50	3	2	10	13	2.4E–14	29	4	10	1.7E–12	29	6.6E–15	107
50	3	2	10	15	6.9E–15	33	4	10	1.7E–12	30	6.6E–15	107

Table A.8. $\lambda_{\max} = 2.5$, while the distribution of the other eigenvalues is $\operatorname{Re}\lambda \in [0,2],\ \operatorname{Im}\lambda \in [-1,1]$.

A.4 The Computational Error of Close Eigenvalues

In this section we take up the problem of the close eigenvalues. As we have seen in the previous experiments, the performance of the least squares Arnoldi method deteriorates when the maximum eigenvalue is close to the second eigenvalue. We examine the techniques proposed in the previous chapter by the following experiments.

A.4.1 The Re-orthogonalization

We begin with the results of the effect of the re-orthogonalization. The process can be written as

$$b_{r+1} = Ac_r - h_{1r}c_1 - h_{2r}c_2 - \cdots - h_{rr}c_r, \tag{A.18}$$

$$b'_{r+1} = b_{r+1} - \varepsilon_{1r}c_1 - \varepsilon_{2r}c_2 - \cdots - \varepsilon_{rr}c_r, \tag{A.19}$$

$$c_{r+1} = b'_{r+1}/ \parallel b'_{r+1} \parallel_2, \quad \varepsilon_{ir} = c_i^T b_{r+1}/c_i^T c_i = c_i^T b_{r+1}. \tag{A.20}$$

We examine the case where the maximum eigenvalue is 2.1, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0, 2],\ \mathrm{Im}\,\lambda \in [-1, 1]$. It can be seen that the effect of the re-orthogonalization in the least squares Arnoldi method is more remarkable than that in the Arnoldi method. Note that the iteration number of the Arnoldi method has a strong influence on the total complexity of the least squares Arnoldi method. The re-orthogonalization is used in the subsequent experiments.

matrix	least squares Arnoldi					Arnoldi				QR	
degree	n_{iter}	m	k	error	time	n_{iter}	m	error	time	error	time
50	1	10	15	5.4E–03	17	1	10	5.5E–02	11	7.4E–15	105
50	2	10	15	1.3E–03	36	2	10	1.1E+00	16	7.4E–15	107
50	1	20	15	2.7E–03	43	1	20	1.8E–03	33	7.4E–15	110
50	2	20	15	1.9E–08	83	2	20	3.4E–05	71	7.4E–15	110
50	3	20	15	2.2E–10	131	3	20	4.3E–05	98	7.4E–15	107
50	4	20	15	3.8E–15	168	4	20	4.3E–05	130	7.4E–15	105

Table A.9. $\lambda_{\max} = 2.1$, while the distribution of the other eigenvalues is $\mathrm{Re}\,\lambda \in [0, 2],\ \mathrm{Im}\,\lambda \in [-1, 1]$.

A.4.2 The Multiplication

The validity of the multiplication of the Arnoldi process is also tested. The algorithm is as follows:

$$b_{r+1} = A^n c_r - \sum_{i=1}^{r} h_{ir} c_i, \tag{A.21}$$

$$h_{ir} = (A^n c_r, c_i), \quad i = 1, ..., r, \quad h_{r+1,r} = \| b_{r+1} \|, \tag{A.22}$$

$$c_{r+1} = b_{r+1}/h_{r+1,r}. \tag{A.23}$$

The process of the multiplication of the matrix A can be considered as the rotation on the origin in the complex plane as illustrated in Figure A.3. The separation of the second eigenvalue is lost when the argument of the rotated eigenvalue is close to $2\pi n$ where n is an arbitrary integer, considering the fact that the eigenvalue $\lambda = ae^{i\theta}$ corresponds to the eigenvalue $a^n e^{in\theta}$ of the matrix A^n.

We denote the number of multiplications by n_{m}.

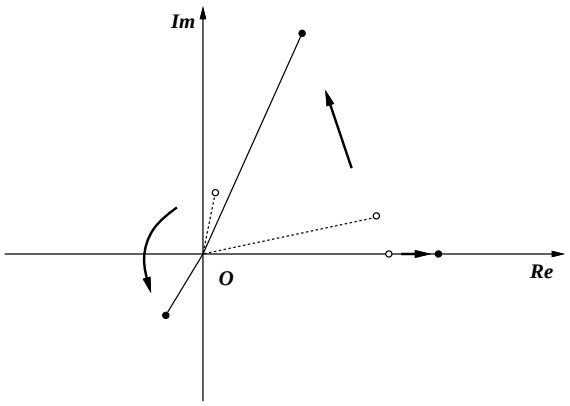

Figure A.3. The concept of the multiplication

The numerical results shown in Table A.10 support this consideration. The matrix solved in this experiment is of degree 50 and has the following distribution of the eigenvalues:

1.

$$\lambda_{\max} = 2.01, \quad \lambda_{\mathrm{neighbor}} = 2.0 \pm 2.0i. \tag{A.24}$$

As for the others,

$$\mathrm{Re}\,\lambda \in [0, 2.0], \quad \mathrm{Im}\,\lambda \in [-1.0, 1.0]. \tag{A.25}$$

2.

$$\lambda_{\max} = 2.01, \quad \lambda_{\mathrm{neighbor}} = 2.0 \pm 0.2i, \tag{A.26}$$

and as for the others,

$$\mathrm{Re}\,\lambda \in [0, 2.0], \quad \mathrm{Im}\,\lambda \in [-1.0, 1.0]. \tag{A.27}$$

The argument of the $\lambda_{\mathrm{neighbor}}$ is $\pm\frac{\pi}{2}$ where $\lambda_{\mathrm{neighbor}} = 2.0 \pm 2.0i$. The precision of $\lambda_{\max}$ deteriorates as the number of multiplications gets closer to 8. Although it is not realistic that we can predicate the distribution of the eigenvalues, this technique is effective when the real parts of all the eigenvalues are positive and the number of multiplications is less than three. The case where $\lambda_{\max} = 2 + 10^{-7}$ and $\lambda_{\mathrm{neighbor}} = 2.0 \pm 0.2i$ indicates that the multiplication technique can separate the considerably close eigenvalues.

A.4.3 The Validity of Rectangular Hull in the Least Squares Arnoldi Method

This subsection discusses the validity of the rectangular hull as shown in Figure A.5. The algorithm to compute the rectangular hull is described as follows:

Algorithm A.4. 1 (Least Squares Arnoldi with Rectangular Hull)

1. Find λ_l, λ_s and λ_i, i.e., the eigenvalues with the largest real part, the smallest real part, and the largest imaginary part, respectively, in the eigenvalue estimates obtained by the Arnoldi process
2. Define the vertices that construct the hull from $(\mathrm{Re}\,\lambda_l, \mathrm{Im}\,\lambda_i)$, $(\mathrm{Re}\,\lambda_s, \mathrm{Im}\,\lambda_i)$, $(\mathrm{Re}\,\lambda_s, -\mathrm{Im}\,\lambda_i)$, and $(\mathrm{Re}\,\lambda_l, -\mathrm{Im}\,\lambda_i)$

The method has an advantage over the former one in that it enables us to deal with only four edges which construct the hull to compute the polynomial. It is also important from the viewpoint of the computational error. We examine this method using the matrix of degree 50. The distribution of the eigenvalues is

$$\lambda_{\mathrm{neighbor}} = 2.0 \pm 1.0i \tag{A.28}$$

$\lambda_{\max}$	$\lambda_{\text{neighbor}}$	conditions				least squares Arnoldi	
		n_{iter}	m	k	n_{m}	error	time
2.01	$2.0 \pm 2.0i$	3	15	30	1	8.5E–01	125
2.01	$2.0 \pm 2.0i$	3	15	30	2	3.0E–04	140
2.01	$2.0 \pm 2.0i$	3	15	30	3	2.6E–02	148
2.01	$2.0 \pm 2.0i$	3	15	30	4	1.7E–08	182
2.01	$2.0 \pm 2.0i$	3	15	30	5	1.0E–11	174
2.01	$2.0 \pm 2.0i$	3	15	30	6	1.8E–14	194
2.01	$2.0 \pm 2.0i$	3	15	30	7	1.3E+00	202
2.01	$2.0 \pm 2.0i$	3	15	30	8	1.2E+00	224
2.01	$2.0 \pm 0.2i$	3	15	30	1	1.3E–01	123
2.01	$2.0 \pm 0.2i$	3	15	30	2	3.6E–01	142
2.01	$2.0 \pm 0.2i$	3	15	30	3	4.6E–01	162
2.01	$2.0 \pm 0.2i$	3	15	30	4	3.1E–05	170
2.01	$2.0 \pm 0.2i$	3	15	30	5	4.0E–10	187
2.01	$2.0 \pm 0.2i$	3	15	30	6	4.3E–14	197
2.01	$2.0 \pm 0.2i$	3	15	30	7	2.0E–15	233
2.0000001	$2.0 \pm 0.2i$	3	15	30	8	2.0E–15	213

Table A.10. The degree of matrix is 50, while the distribution of the other eigenvalues is $\operatorname{Re}\lambda \in [0, 2.0]$, $\operatorname{Im}\lambda \in [-1.0, 1.0]$.

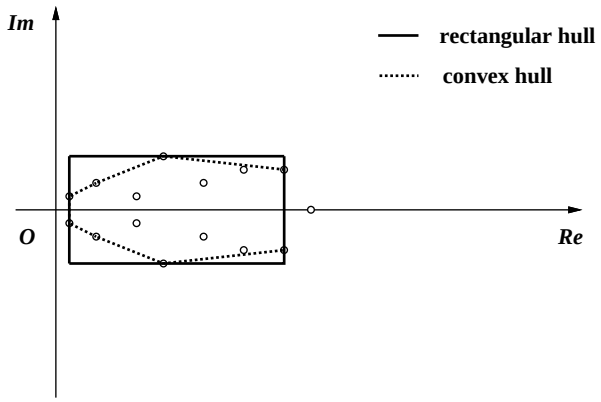

Figure A.4. The concept of the rectangular hull

and as for the others,

$$\mathrm{Re}\,\lambda \in [0, 2.0], \quad \mathrm{Im}\,\lambda \in [-1.0, 1.0]. \tag{A.29}$$

The following table shows that the effect of this method on the total complexity is not so remarkable, since the complexity of this process is not so large in comparison with that of the Arnoldi method to compute the eigenvalue estimates.

$\lambda_{\max}$	conditions			convex hull		rectangular hull	
	n_{iter}	m	k	error	time	error	time
3.0	2	5	10	6.1E–13	19	2.5E–11	17
2.5	2	10	15	2.6E–11	34	2.8E–13	34
2.1	4	15	25	1.1E–10	141	1.5E–15	130
2.01	4	20	25	1.8E–01	197	4.8E–12	178
2.05	4	20	25	2.7E–09	198	1.5E–15	183
2.001	2	35	10	1.7E–13	230	2.0E–12	245
2.0001	2	40	5	3.3E–13	321	5.3E–13	318

Table A.11. The degree of matrix is 50, $\lambda_{\mathrm{neighbor}}$ is $2.0 \pm 1.0i$ and the distribution of the others is $\mathrm{Re}\,\lambda \in [0, 2.0]$, $\mathrm{Im}\,\lambda \in [-1.0, 1.0]$.

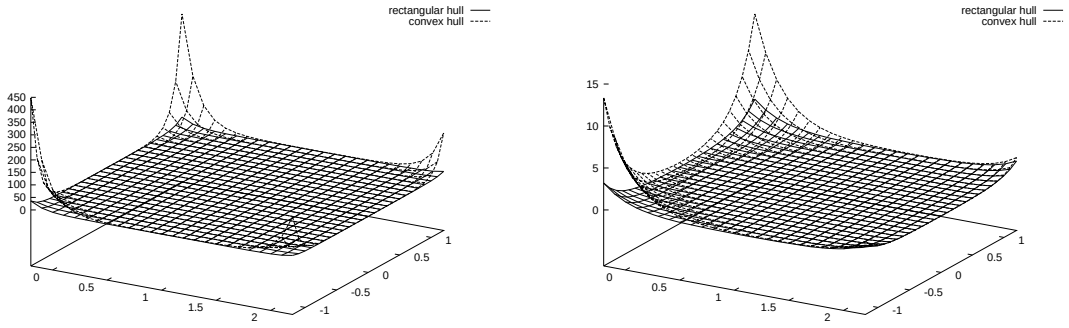

Figure A.5. The values of $|p_n(z)|$ on the complex plane where $\lambda_{\max} = 2.01$ and $\lambda_{\max} = 2.001$, using the convex and the rectangular hull

A.5 Consideration

A.5.1 Recapitulation

We tested the validity of the iterative least squares Arnoldi method with the various combinations of the parameters, and several techniques to extract the wanted eigenvalues from the cluster of the eigenvalues.

The re-orthogonalization is generally valid to keep the orthogonality of the eigenvectors. The side effect of the rectangular hull is the enlargement of the enclosed area, which is not essential in this method since the rightmost point of the hull is invariable.

k is the most effective parameter to improve the precision with small complexity. It requires much memory storage for the coefficients of the Chebyshev polynomials, though. This problem is solved by the iteration of the Arnoldi method.

The increase of m is not so effective compared with the other parameters, considering its complexity of roughly $\mathcal{O}(m^3)$. Hence the iteration number of the Arnoldi method should be as small as possible, while the degree of the polynomial is to be made large. The extraction of the wanted eigenvalues is not guaranteed, though, if the iteration number of the Arnoldi method is too small. This difficulty may be surmounted by the deflation technique, which is not examined in this paper yet.

A.5.2 The Close Eigenvalues

The Arnoldi method computes the eigenvalues with the largest moduli. If the superfluous eigenvalues have large imaginary parts, we will not be able to obtain the appropriate eigenvalue estimates. We argue here the several measures proposed for the power method.

1. Suppose $|\lambda_1| \doteq |\lambda_2| > |\lambda_3|$. After sufficient iterations, we have the approximation

$$x_k \approx \lambda_1^k (c_1 v_1 + c_2 (\lambda_2/\lambda_1)^k v_2. \tag{A.30}$$

It can be extended as

$$\begin{aligned} x_k &= d_1 v_1 + d_2 v_2 && \text{(A.31)} \\ x_{k+1} &= d_1 \lambda_1 v_1 + d_2 \lambda_2 v_2 && \text{(A.32)} \\ x_{k+2} &= d_1 \lambda_1^2 v_1 + d_2 \lambda_2^2 v_2, && \text{(A.33)} \end{aligned}$$

and we have the following relation

$$x_{k+2} + \beta x_{k+1} + \gamma x_k = 0. \tag{A.34}$$

The eigenvalues are computed from

$$\lambda^2 + \beta\lambda + \gamma = 0. \tag{A.35}$$

The coefficients β and γ is the solutions of the normal equation

$$\begin{pmatrix} (x_{k+1}, x_{k+1}) & (x_{k+1}, x_k) \\ (x_k, x_{k+1}) & (x_k, x_k) \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} (x_{k+1}, x_{k+2}) \\ (x_k, x_{k+2}) \end{pmatrix}. \tag{A.36}$$

2. When the other case where $|\lambda_1| > |\lambda_2|$, we finally have the relation

$$x_{k+1} \doteq \alpha x_k \tag{A.37}$$

and α is computed similarly by

$$\alpha = \frac{(x_{k+1}, x_k)}{(x_k, x_k)}. \tag{A.38}$$

Although the case with the eigenvalues with larger moduli is not covered, these techniques are also valid for the Arnoldi method, which is a variation of the power method.

A.6 Comparison with Other Methods

Some test problems from the Harwell-Boeing sparse matrix collection (see Duff, Grimes and Lewis [14]), the computed spectra of which are shown in Figure A.6 and Figure A.7, are solved using the block Arnoldi method. Ho's algorithm is used for reference.

The stopping criterion is based on the maximum of all computed residuals $\max_{1\leq i\leq r} \parallel Ax_i - \lambda_i x_i \parallel_2 / \parallel x_i \parallel_2 \equiv \max_{1\leq i\leq r} \parallel H_{m+1,m} Y_{m,r,i} \parallel_2 / \parallel Y_{m,i} \parallel_2 \leq \varepsilon$. $Y_{m,r,i}$ and $Y_{m,i}$ stand for the i-th column of the $Y_{m,r}$ and Y_m, described in Chapter 4.

Table A.12 and Table A.13 indicate that Ho's algorithm shows better performance than the least squares Arnoldi method in most conditions except for the cases where the moduli of the wanted eigenvalues are much larger than those of the unwanted eigenvalues. We may derive from the result the poor optimality of the convex hull despite its low computation cost.

Lehoucq and Scott [35] presented a software survey of large-scale eigenvalue methods and comparative results. The Arnoldi-based software included the following three packages ARNCHEB

package by Braconnier [9], the ARPACK software package by Lehoucq and Sorensen [36], and the Harwell Subroutine Library code EB13 by Scott [63] and Sadkane [60].

The ARNCHEB package provides the subroutine ARNOL, which implements an explicitly restarted Arnoldi iteration. The code is based on the deflated polynomial accelerated Arnoldi iteration and uses Chebyshev polynomial acceleration. The Harwell Subroutine Library code EB13 implements the similar algorithm and also uses Ho's Chebyshev polynomial acceleration. The ARPACK provides subroutine DNAUPD that implements the implicitly restarted Arnoldi iteration.

Some findings are reported on these methods:

1. ARNCHEB gives reasonable results for computing a single eigensolution but it can struggle on problems for which several eigenvalues are requested.

2. ARPACK displays monotonic consistency and is generally the fastest and most dependable of the codes studied, especially for small convergence tolerances and large departures from normality. It uses dramatically fewer matrix-vector product than ARNCHEB. However, its restarting strategy can be more expensive.

Moreover, from the results of Table A.14 and Table A.15, we can derive the strong dependency of the polynomial acceleration on the distribution of spectrum. Figure A.6 and A.7 indicate that the non-clustered distribution of spectra causes the slow convergence, since the approximate spectra may completely differ from the accurate ones.

problem	WEST0497		WEST0655		WEST0989		WEST2021	
degree of matrix	497		655		989		2021	
number of entries	1727		2854		3537		7353	
number of multiplications	924	440	275	120	13751	*	767	320
number of restarts	14	10	3	2	162	*	12	7
CPU time (sec.)	0.37	0.22	0.17	0.12	8.71	*	1.28	0.67

Table A.12. Test problems from CHEMWEST, a library in the Harwell-Boeing Sparse Matrix Collection, which was extracted from modeling of chemical engineering plants. The results by Manteuffel's algorithm (right) versus those by the least squares Arnoldi method (left), with size of the basis 20, degree of the polynomial 20, and block size 1, respectively, are listed. * denotes the algorithm fails to converge.

degree of matrix	2000		4000		6000		8000		10000	
number of entries	5184		8784		12384		15984		19584	
number of multiplications	589	240	393	180	236	140	393	380	236	80
number of restarts	7	4	5	3	3	2	5	7	3	1
CPU time (sec.)	0.83	0.43	1.24	0.70	1.23	0.85	2.57	2.81	2.14	0.97

Table A.13. Test problems from TOLOSA extracted from fluid-structure coupling (flutter problem). Size of the basis, degree of the polynomial, and block size are 20, 20, 1, respectively.

Algorithm	$r = 1, m = 8$	$r = 5, m = 20$
EB12	*	98/20930
ARNCHEB	8.6/3233	71/15921
EB13	17/4860	18/4149
ARPACK	3.7/401	2.1/167

Table A.14. Evaluation by Lehoucq and Scott. CPU times (in seconds) by IBM RS/6000 3BT and matrix-vector products for computing the right-most eigenvalues of WEST2021 from CHEMWEST (* denotes convergence not reached within $2000m$ matrix-vector products). We denote by r the block size and by m the subspace dimension.

Algorithm	$r = 1, m = 12$	$r = 4, m = 20$
EB12	0.6/423	9.1/2890
ARNCHEB	3.4/1401	4.7/1712
EB13	0.4/119	1.3/305
ARPACK	0.5/90	1.3/151

Table A.15. CPU times (in seconds) and matrix-vector products for computing the right-most eigenvalues of PORES2, matrix of degree 1224 with 9613 entries, which was extracted from reservoir simulation.

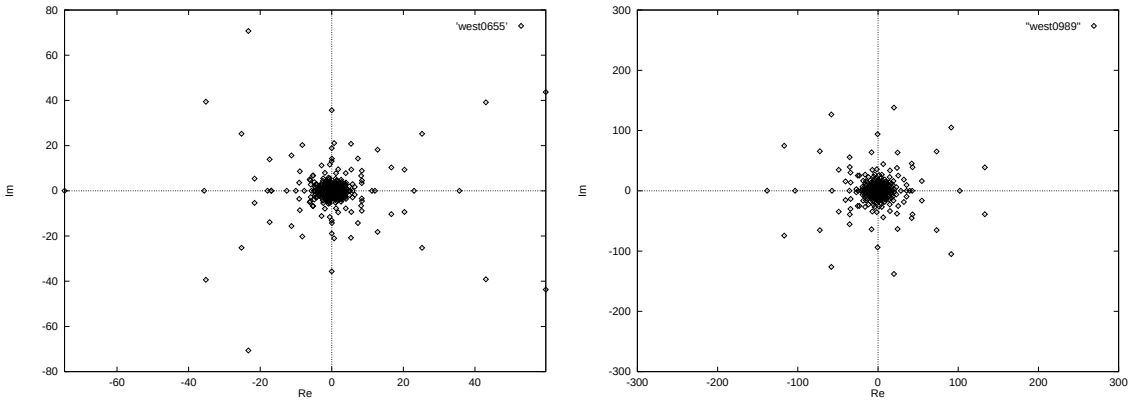

Figure A.6. Computed spectra of WEST0655 and WEST0989 from CHEMWEST.

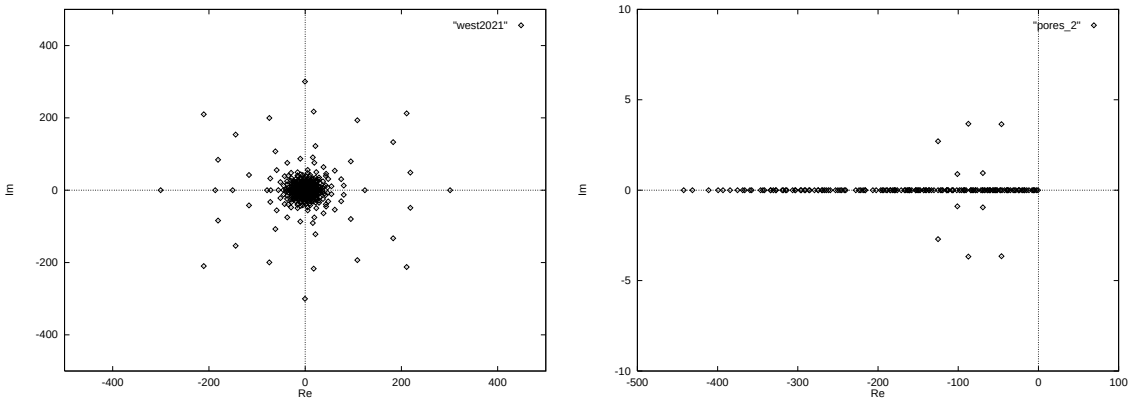

Figure A.7. Computed spectra of WEST2021 and PORES2

Appendix B

Orthonormalization Techniques

This appendix introduces the algorithms for orthonormalizing a given subspace. A set of vectors $\mathcal{G} = \{x_1, x_2, ..., x_n\}$ is called orthogonal if

$$(x_i, x_j) = 0, \quad \text{if } i \neq j \tag{B.1}$$

holds and orthonormal if every vector of $\mathcal{G}$ has a 2-norm equal to unity. A vector which is orthogonal to all the vectors in a subspace S is said to be orthogonal to this subspace and the set of these vectors, which is denoted by $\mathsf{S}^{\perp}$, is called the *orthogonal complement* of S. The projector which maps a vector x into its component in the subspace S is called the *orthogonal projector* onto S.

The orthonormalization of any subspace can be achieved by the method known as the Gram-Schmidt's orthogonalization, which can be described as follows:

Algorithm B.0. 1 (Gram-Schmidt)

1. Compute $r_{11} = \| x_1 \|_2$. If $r_{11} = 0$ then stop
2. Compute $q_1 = x_1 / r_{11}$
3. For $j = 2, ..., r$, do
4. Compute $r_{ij} = (x_j, q_i)$
5. For $i = 1, ..., j-1$, do
6. $\hat{q} = x_j - \sum_{i=1}^{j-1} r_{ij} q_i$
7. $r_{jj} = \| \hat{q} \|_2$
8. If $r_{jj} \neq 0$ then $q_j = \hat{q} / r_{jj}$

The relation

$$x_j = \sum_{i=1}^{j} r_{ij} q_i \tag{B.2}$$

is called the QR decomposition of the $n \times r$ matrix $X = [x_1, x_2, ..., x_r]$.

The above method can be modified for better numerical properties,

Algorithm B.0. 2 (Modified Gram-Schmidt)

1. Compute $r_{11} = \| x_1 \|_2$. If $r_{11} = 0$ then stop
2. Compute $q_1 = x_1 / r_{11}$
3. For $j = 2, ..., r$, do
4. $\quad \hat{q} = x_j$
5. $\quad$ For $i = 1, ..., j-1$, do
6. $\quad\quad r_{ij} = (\hat{q}, q_i)$
7. $\quad\quad \hat{q} = \hat{q} - r_{ij} q_i$
8. $\quad r_{jj} = \| \hat{q} \|_2$
9. $\quad$ If $r_{jj} \neq 0$ then $q_j = \hat{q} / r_{jj}$

Another alternative is the Householder algorithm, which uses the Householder reflectors

$$P = I - 2ww^T, \tag{B.3}$$

where w is a vector of 2-norm unity and the vector Px represents a mirror image of x with respect to the hyperplane span $\{w\}^{\perp}$. For any vector x, the vector w for the Householder transformation (B.3) is selected in such a way that

$$Px = \alpha e_1, \tag{B.4}$$

where α is a scalar. This yields

$$2w^T x w = x - \alpha e_1, \tag{B.5}$$

which shows that w is a multiple of the vector $x - \alpha e_1$, i.e.,

$$w = \pm \frac{x - \alpha e_1}{\| x - \alpha e_1 \|_2}. \tag{B.6}$$

Therefore, we have the condition

$$2(x - \alpha e_1)^T x = \| x - \alpha e_1 \|_2, \tag{B.7}$$

which gives $\alpha = \pm \| x \|_2$. Given a $n \times m$ matrix X, its first column can be transformed to a multiple of the column e_1 by premultiplying it by a Householder matrix P_1, that is,

$$X_1 = P_1 X, \quad \text{where } X_1 e_1 = \alpha e_1. \tag{B.8}$$

Assume here that the matrix X has been transformed in $k-1$ successive steps into

$$X_k \equiv P_{k-1} \cdots P_1 X_1, \tag{B.9}$$

which is upper triangular up to column number $k-1$. To advance by one step, the next Householder reflector matrix must be

$$P_k = I - 2 w_k w_k^T, \quad w_k = \frac{z}{\| z \|_2}, \tag{B.10}$$

where the components of the vector z are given by

$$z_i = \begin{cases} 0 & \text{for} \quad i < k \\ \text{sign}(x_{kk}) \times \left(\sum_{i=k}^{n} k_{ik}^2 \right)^{1/2} + x_{ii} & \text{for} \quad i = k \\ x_{ik} & \text{for} \quad i > k \end{cases} . \tag{B.11}$$

Algorithm B.0. 3 (Householder Orthogonalization)

1. Define $X = [x_1, ..., x_m]$
2. For $k = 1, ..., m$, do
3. If $k > 1$ then compute $r_k = P_{k-1} P_{k-2} \cdots P_1 x_k$
4. Compute w_k
5. Compute $r_k = P_k r_k$ with $P_k = I - 2 w_k w_k^T$
6. Compute $q_k = P_1 P_2 \cdots P_k e_k$

References

[1] L. V. Ahlfors, *Complex Analysis*, McGraw-Hill, 1979.

[2] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[4] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems*, Tech. Report 95-04, Department of Mathematics, University of Kentucky, 1995.

[5] Z. Bai and J. Demmel, *On a block implementation of Hessenberg multishift QR iteration*, International Journal of High Speed Computing, 1 (1989), pp. 97–112.

[6] R. G. Bartle, *Elements of Real Analysis*, Wiley, 1964.

[7] M. Bennani, *A propos de la stalilité de la résolution d'équations sur ordinateur*, PhD thesis,INP,Toulouse, (1991).

[8] D. Boley, R. Maier, and J. Kim, *A parallel QR algorithm for the nonsymmetric eigenvalue problem*, Computer Physics Communication, 53 (1989), pp. 61–70.

[9] T. Braconnier, *The Arnoldi-Tchebycheff algorithm for solving nonsymmetric eigenproblems*, Tech. Report TR/PA/93/25, CERFACS, Toulouse, 1993.

[10] F. Chatelin, *Valeurs Propres de Matrices*, Masson, Paris, 1988.

[11] A. Clayton, *Further results on polynomials having least maximum modulus over an ellipse in the complex plane*, Technical Report AEEW-7348,UKAEA, (1963).

[12] B. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience Publishers, New York, 1963.

[13] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalue and corresponding eigenvectors of large real symmetric matrices*, J. Comp. Phys., 17 (1975), pp. 87–94.

[14] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Softw., 15 (1989), pp. 1–14.

[15] N. Dunford and J. T. Schwartz, *Linear Operators*, Wiley, New York, 1988.

[16] A. M. Erisman, R. G. Grimes, J. G. Lewis, W. G. P. Jr., and H. D. Simon, *Evaluation of orderings for unsymmetric sparse matrices*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 600–624.

[17] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils*, Tech. Report 941, Department of Mathematics, Utrecht University, 1996.

[18] J. G. F. Francis, *The qr method — a unitary analogue to the lr transformation*, Comput.J., 4 (1961/62), pp. 265–271.

[19] V. Fraysse, L. Giraud, and V. Toumazou, *Parallel computation of spectral portrait on the Meiko CS2*, in High-Performance Computing and Networking, H. Liddell, A. Colbrook, B. Hertzberger, and P. Sloot, eds., vol. 1067, Springer-Verlag, New York, pp. 312–318.

[20] R. W. Freund, G. H. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, Acta Numerica, (1991), pp. 57–100.

[21] R. W. Freund and N. M. Nachtigal, *Qmr: a quasi-minimal resudual method for non-hermitian linear systems*, Numer. Math, 60 (1991), pp. 315–339.

[22] T. J. Garatt, G. Moore, and A. Spense, *Two methods for the numerical detection of Hopf bifurcations*, in Bifurcation and chaos: analysis, algorithms and applications, R. Seidel, F. W. Schneider, and H. Troger, eds., Birkhauser, 1991, pp. 119–123.

[23] G. A. Geist and G. J. Davis, *Finding eigenvalues and eigenvectors of unsymmetric matrices using a distributed-memory multiprocessor*, Parallel Computing, 13 (1990), pp. 199–209.

[24] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.

[25] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second-order Richardson iterative methods, Part I and II*, Numer. Math, 3 (1961), pp. 147–156, 157–168.

[26] P. R. Halmos, *Finite-Dimensional Vector Spaces*, Springer Verlag, New York, 1958.

[27] G. Henry and R. van de Geijn, *Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problems: myth and reality*, SIAM. J. Sci. Comput., 17 (1996), pp. 870–883.

[28] G. Henry, D. Watkins, and J. Dongarra, *A parallel implementation of the nonsymmetric QR algorithm for distributed memory architectures*, Tech. Report 121, LAPACK Working Note, 1997.

[29] E. Hille, *Analytic Function Theory*, Ginn, Boston, 1962.

[30] D. Ho, *Tchebychev acceleration technique for large scale nonsymmetric matrices*, Numer. Math, 56 (1990), pp. 721–734.

[31] D. Ho, F. Chatelin, and M. Bennani, *Arnoldi-Tchebychev procedures for large scale nonsymmetric matrices*, Mathematical Modeling and Numerical Analysis, 24 (1990), pp. 53–65.

[32] C. G. J. Jacobi, *Ueber ein leichtes Verfahren, die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Journal für die reine und angewandte Mathematik, (1846), pp. 51–94.

[33] T. Kato, *Perturbation Theory for Linear Operators*, Springer Verlag, New York, 1980.

[34] R. B. Lehoucq, *Restarting an Arnoldi Reduction*, Tech. Report MCS-P591-0496, Argonne National Laboratory.

[35] R. B. Lehoucq and J. A. Scott, *An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices*, Tech. Report MCS-P547-1195, Argonne National Laboratory.

[36] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM Journal on Matrix Analysis and Applications, 17 (1996).

[37] T. A. Manteuffel, *An iterative method for solving nonsymmetric linear systems with dynamic estimation of parameters*, Tech. Report Digital Computer Laboratory Reports, Rep. UIUCDS-R-75-758, University of Illinois, 1975.

[38] ———, *The Tchebychev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.

[39] K. J. Maschhoff and D. C. Sorensen, *A Portable Implementation of ARPACK for Distributed Memory Parallel Architectures*, in Proceedings of Workshop on Applied Parallel Computing in Industrial Problems and Optimization, Technical University of Denmark, Aug. 1996.

[40] A. Nishida, *Least Squares Arnoldi for Large Nonsymmetric Eigenproblems*, in Proceedings of 5th Copper Mountain Conference on Iterative Methods, vol. 2, Copper Mountain, Apr. 1998.

[41] ———, *Polynomial Acceleration for Large Nonsymmetric Eigenproblems*, PhD thesis, the University of Tokyo, Tokyo, Mar. 1998.

[42] A. Nishida and Y. Oyanagi, *A New Acceleration of the Projection Method in Nonsymmetric Eigenvalue Problems*, in Proceedings of 1995 International Conference on High-Performance Computing in Asia-Pacific Region, Sep. 1995.

[43] ———, *A Polynomial Acceleration of the Projection Method for Large Nonsymmetric Eigenvalue Problems*, in Kokyuroku, vol. 944, Research Institute for Mathematical Sciences, Kyoto University, 1996, pp. 135–146.

[44] ———, *A Polynomial Acceleration of the Projection Method for Large Nonsymmetric Eigenvalue Problems*. 1996 SIAM Annual Meeting, Kansas City, USA, Jul. 1996.

[45] ———, *A Polynomial Acceleration of the Projection Method for Large Nonsymmetric Eigenvalue Problems*, in Proceedings of 1996 SIAM Conference on Sparse Matrices, Coeur d'Alene, USA, Oct. 1996, Society for Industrial and Applied Mathematics, p. 117.

[46] ———, *Analysis of Accelerating Algorithms for the Restarted Arnoldi Iteration*, in 15th IMACS World Congress on Scientific Computation, Modeling and Applied Mathematics, A. Sydow, ed., vol. 2, Wissenschaft und Technik Verlag, Berlin, 1997, pp. 279–284.

[47] ———, *Evaluation of Acceleration Techniques for the Restarted Arnoldi Method*, in Kokyuroku, vol. 990, Research Institute for Mathematical Sciences, Kyoto University, 1997, pp. 41–51.

[48] A. Nishida, R. Suda, and Y. Oyanagi, *Polynomial Acceleration for Restarted Arnoldi Iteration and its Parallelization*, in Iterative Methods in Scientific Computation, J. Wang, M. B. Allen III, B. M. Chen, and T. Mathew, eds., vol. 4 of IMACS Series in Computational and Applied Mathematics, International Association for Mathematics and Computers in Simulation, New Brunswick, Jun. 1998, ch. Iterative Methods in Computational Linear Algebra, pp. 45–52.

[49] J. M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988.

[50] S. G. Petiton, *Parallel subspace method for non-Hermitian eigenproblems on the Connection Machine (CM2)*, Applied Numerical Mathematics, 10 (1992), pp. 19–35.

[51] H. Rutishauser, *Computational aspects of f. l. bauer's simultaneous iteration method*, Numer. Math., 13 (1969), pp. 4–13.

[52] Y. Saad, *On the Rates of Convergence of the Lanczos and the Block-Lanczos Methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[53] ———, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra and its Applications, 34 (1980), pp. 269–295.

[54] ———, *Iterative solution of indefinite symmetric linear systems by methods using orthogonal polynomials over two disjoint intervals*, SIAM J. Numer. Anal., 20 (1983), pp. 784–811.

[55] ———, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.

[56] ———, *On the condition number of some Gram matrices arising from least squares approximation in the complex plane*, Numer. Math., 48 (1986), pp. 337–347.

[57] ———, *Least squares polynomials in the complex plane and their use for solving nonsymmetric linear systems*, SIAM J. Numer. Anal., 24 (1987), pp. 155–169.

[58] ———, *Numerical Methods for Large Eigenvalue Problems*, Wiley, New York, 1992.

[59] ———, *Iterative Methods for Sparse Linearsystems*, PWS Publishing, Boston, 1995.

[60] M. Sadkane, *A block Arnoldi-Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numer. Math., 64 (1993), pp. 181–193.

[61] ———, *Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems*, Numer. Math., 64 (1993), pp. 195–211.

[62] D. H. Sattinger, *Bifurcation of periodic solutions of the Navier Stokes equations*, Arch. Rational Mech. Anal., 41 (1971), pp. 68–80.

[63] J. A. Scott, *An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices*, ACM Trans. Mathematical Software, (1995), pp. 432–475.

[64] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996).

[65] D. C. Smolarski and P. E. Saylor, *Optimum parameters for the solution of linear equations by Richardson's iteration*, BIT, 28 (1982), pp. 163–178.

[66] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[67] D. C. Sorensen and C. Yang, *A Truncated RQ-iteration for Large Scale Eigenvalue Calculations*, Tech. Report TR96-06, Rice University, 1996.

[68] G. W. Stewart, *Simultaneous Iteration for Computing Invariant Subspaces of Non-Hermitian Matrices*, Numer. Math, 25 (1976), pp. 123–136.

[69] ———, *SRRIT - A FORTRAN subroutine to calculate the dominant invariant subspaces of a real matrix*, Tech. Report TR-514, University of Maryland, 1978.

[70] R. Suda, A. Nishida, and Y. Oyanagi, *A New Data Mapping Method for Parallel Hessenberg QR Method and its Efficient Implementation on AP1000+*, in Proceedings of Joint Symposium on Parallel Processing 1997, Kobe, 1997, pp. 377–384. in Japanese.

[71] G. Szegö, *Orthogonal Polynomials*, AMS, 1975.

[72] The Mathematical Society of Japan, *Encyclopedic Dictionary of Mathematics*, MIT Press, 2nd ed., 1987.

[73] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, 1997.

[74] R. A. van de Geijn, *Storage schemes for Parallel Eigenvalue Algorithms*, in Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms, G. Golub and P. V. Fooren, eds., Springer-Verlag, New York, 1988, pp. 639–648.

[75] H. van der Vorst, *Subspace Iteration for Eigenproblems*, CWI Quarterly, 9 (1996), pp. 151–160.

[76] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, 1962.

[77] D. S. Watkins, *Shifting Strategies for the Parallel QR Algorithm*, SIAM. J. Sci. Comput., 15 (1994), pp. 953–958.

[78] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[79] J. H. Wilkinson and C. Reinsch, *Linear Algebra*, Springer-Verlag, New York, 1971.

[80] H. E. Wrigley, *Accelerating the Jacobi method for solving simultaneous equations by Chebyshev extrapolation when the eigenvalues of the iteration matrix are complex*, Comput. J., 6 (1963), pp. 169–176.

[81] L. Wu and E. Chu, *New Distributed-memory Parallel Algorithms for Solving Nonsymmetric Eigenvalue Problems*, in Proceedings of 7th SIAM Conference on Parallel Processing for Scientific Computing, 540-545, 1995.

[82] D. P. Young, *Iterative Solution of Large Linear Systems*, Academic Press, 1971.